

UNIVERZITA PARDUBICE
FAKULTA ELEKTROTECHNIKY
A INFORMATIKY

DISERTAČNÍ PRÁCE

2020

Ing. Jaroslav Moravec

Universita Pardubice

Fakulta elektrotechniky a informatiky

Nástroje a algoritmy pro identifikaci osob s využitím kontury lidské ruky

Disertační práce

2020

Ing. Jaroslav Moravec

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 31.10.2020

Jaroslav Moravec

ANOTACE

Student se v rámci disertační práce komplexně zabýval oblastí biometrické identifikace osob s využitím kontury lidské ruky a systematické korekce vad distorze optických soustav biometrických scannerů pro účely pořizování obrazů rukou. Cíl disertační práce byl návrh nových, případně inovovaných, metod a postupů v oblasti identifikace osob s využitím kontury lidské ruky v návaznosti na technické vybavení potřebné pro snímání obrazů lidské ruky. Předmětem výzkumu bylo sestavit návrh nových metod pro odstraňování distorze optických soustav biometrických scannerů. Byly uvažovány různé metody polynomiální korekce centrické distorze s využitím pokročilých evolučních algoritmů. Další oblastí výzkumu byly nové postupy a metody pro identifikaci osob s využitím kontury lidské ruky založené na pokročilých evolučních algoritmech.

Výsledky návrhu korektorů distorze vad optických soustav byly ověřeny s využitím komparativních obrazů ze zvolené optické soustavy biometrického scanneru a také na komparativních obrazech obecného charakteru. Výsledky návrhu algoritmů pro identifikaci osob s využitím kontury lidské ruky byly prakticky ověřeny na veřejně dostupné komparativní databázi obrazů kontur rukou. Všechny navržené postupy byly porovnány s výsledky dosahovanými v rámci state-of-the-art v dané vědní oblasti.

KLÍČOVÁ SLOVA

Klasifikace kontury ruky, evoluční algoritmy, korekce distorze obrazu, biometrická identifikace osob

TITLE

The Tools and algorithms for hand contour identification with use a human hand contour

ANNOTATION

The student (a problem solver) dealt in the framework of his doctoral thesis with a method of a biometric identification of the persons with use the human hand contours. The systematic correction of the centric distortion defects of the biometric scanner was also part of the thesis. Such system (biometric scanner) serves to capture the hand images. Aim of the thesis was to propose the new or innovative methods in the area of the persons identification in connection with the technical equipment which is necessary in the hand images capturing process. Topic of the research was to create the proposals of the new methods enabling to repair the images which are burdened with the distortion defect. Both barrel or pillow. Various methods which are based on so called polynomial centric correction algorithm were used. All proposed methods utilize advanced evolutionary optimizers. Next research area was presented by the new techniques and methods which enable to identify the persons with use of the hand contour. All these methods are also based on advanced evolutionary optimizers.

The results of proposals of the correctors of the optical defects was verified with use of a set of comparative images which were obtained with use of the optical system of the biometric scanner. A set of images from publicly accessible database were used as well and this set served as unique source of valuable information. All proposed methods were compared with the results reached by other authors in accordance with the current state-of-the-art in the given scientific area.

KEYWORDS

Hand contour classification, evolutionary optimizers, an image distortion correction, biometrical identification of persons

Obsah

| | |
|---|----|
| Seznam Obrázků | 8 |
| Seznam Tabulek | 10 |
| Matematické Značení | 11 |
| 1 Úvod | 14 |
| 1.1 Obecný pohled do historie | 14 |
| 1.2 Uspořádání kapitol a základní přehled řešených témat v disertaci | 15 |
| 2 Cíle disertace | 17 |
| 3 Přehled historie a aktuální stav | 18 |
| 3.1 Část první | 18 |
| 3.2 Část druhá | 20 |
| Literatura | 23 |
| 4. Konstrukce biometrického scanneru | 25 |
| 4.1 Precizní laboratorní biometrický scanner | 26 |
| 4.2 Další možnosti využití scanneru | 27 |
| 5 Evoluční algoritmy pro biometrickou identifikaci | 30 |
| 5.1 Diferenciální evoluce a algoritmus jDE | 32 |
| 5.2 Algoritmus EPSDE | 35 |
| 5.3 CMA evoluční strategie | 38 |
| Literatura | 39 |
| 6 Metody kalibrace obrazu biometrického scanneru | 43 |
| 6.1 Kalibrace obrazu založená na metodě back-mappingu | 43 |
| 6.1.1 Úvod | 43 |
| 6.1.2 Související publikace | 44 |
| 6.1.3 Kaskádové uspořádání algoritmů | 45 |
| 6.1.4 Algoritmus pro odstranění distorze obrazu využívající polynomiální reprezentaci | 46 |

| | | |
|---------|--|-----|
| 6.1.5 | Navržená metoda pro korekci obrazu | 50 |
| 6.1.6 | Experimentální výsledky | 58 |
| 6.1.6.1 | Časová náročnost korekce obrazu | 58 |
| 6.1.6.2 | Nelineární dvoudimenzionální účelová funkce | 58 |
| 6.1.6.3 | jDE a CMA vs ostatní metaheuristiky | 59 |
| 6.1.6.4 | N-dimenzionální kaskádový evoluční optimalizátor | 65 |
| 6.1.6.5 | Reprezentace míry zkreslení | 68 |
| 6.1.7 | Diskuze | 72 |
| 6.1.8 | Závěr | 73 |
| | Literatura | 74 |
| 6.2 | Metoda korekce obrazu založená na algoritmu direct-mappingu | 81 |
| 6.2.1 | Úvod | 81 |
| 6.2.2 | Metoda pro korekci obrazu | 82 |
| 6.2.3 | Experimentální výsledky | 88 |
| 6.2.4 | Závěr | 89 |
| | Literatura | 90 |
| 6.3 | Kalibrace obrazu založená na algoritmu division-model | 91 |
| 6.3.1 | Úvod | 91 |
| 6.3.2 | Související publikace | 92 |
| 6.3.3 | Použitý evoluční optimalizátor | 92 |
| 6.3.4 | Navržená korekční metoda | 93 |
| 6.3.5 | Nastavení pracovních parametrů | 94 |
| 6.3.6 | Experimentální výsledky | 98 |
| 6.3.6.1 | Chování testovaných division-modelů při odstranění distorze u uměle poškozených obrazů – sada distorzních polynomů s jedním koeficientem | 98 |
| 6.3.6.2 | Chování testovaných division-modelů při odstranění distorze u uměle poškozených obrazů – sada distorzních polynomů s více koeficienty | 99 |
| 6.3.7 | Závěr a diskuze | 109 |
| | Literatura | 110 |
| 7. | Navržené metody klasifikace kontur rukou | 113 |
| 7.1 | Klasifikace kontury ruky využívající algoritmus EPSDE | 113 |
| 7.1.1 | Úvod | 113 |
| 7.1.2 | Související publikace | 115 |
| 7.1.2.1 | Vybrané metody autentifikace osob s využitím kontury ruky | 115 |
| 7.1.2.2 | Vybrané evoluční optimalizátory | 117 |
| 7.1.2.3 | Algoritmus ICP | 118 |
| 7.1.3 | Popis navržené metody | 119 |
| 7.1.3.1 | Obraz z kamery, transformace RGB obrazu na B&W obraz a filtrace B&W obrazu | 120 |
| 7.1.3.2 | Odhad délky ruky, nalezení kontury ruky, sestavení RDD | 122 |
| 7.1.3.3 | Popis kontury ruky, Radial Distance Diagram, segmentace kontury | 125 |
| 7.1.3.4 | Výpočet os prstů a pozic kloubů prstů | 127 |
| 7.1.3.5 | Úprava kontury ruky pro evoluční algoritmus | 130 |

| | | |
|----------|---|-----|
| 7.1.3.6 | Prostor možných řešení účelové funkce | 130 |
| 7.1.3.7 | Chromozom, reprezentace fitness funkce, evoluční proces | 131 |
| 7.1.3.8 | Složitost navrženého algoritmu a její minimalizace s ohledem na praktickou realizaci, výpočet fitness | 134 |
| 7.1.3.9 | Algoritmus ICP pro zarovnávání kontur \mathcal{M}, S | 137 |
| 7.1.3.10 | Praktická realizace navrženého algoritmu eaICP | 139 |
| 7.1.4 | Experimentální výsledky | 140 |
| 7.1.4.1 | Použité databáze THID, GPDS, HGDB a značení obrazů rukou | 140 |
| 7.1.4.2 | Časové nároky použitých evolučních optimalizátorů | 141 |
| 7.1.4.3 | Volba vhodného evolučního optimalizátoru | 143 |
| 7.1.4.4 | Volba pracovních parametrů, přesnost výpočtu | 146 |
| 7.1.4.5 | Způsob klasifikace a zvolená nastavení estimátorů | 149 |
| 7.1.4.6 | Výsledky klasifikace s využitím estimátoru eaICP a porovnání s ostatními pracemi | 152 |
| 7.1.4.7 | Přesnost algoritmu eaICP při klasifikaci identických snímků z databáze THID | 161 |
| 7.1.5 | Diskuze | 162 |
| 7.1.6 | Závěr | 163 |
| | Literatura | 164 |
| 7.2 | Evoluční estimátor pro klasifikaci kontury ruky využívající množinu různých geometrických primitiv | 173 |
| 7.2.1 | Úvod | 173 |
| 7.2.2 | Související publikace | 174 |
| 7.2.2.1 | Obecký pohled na problematiku | 174 |
| 7.2.2.2 | Evoluční optimalizátor EPSDE | 174 |
| 7.2.2.3 | Algoritmus ICP | 176 |
| 7.2.3 | Navržený algoritmus | 176 |
| 7.2.4 | Experimentální výsledky | 182 |
| 7.2.4.1 | Databáze snímků GPDS a způsob hodnocení výsledků | 182 |
| 7.2.4.2 | Volba vhodných pracovních parametrů optimalizátoru EPSDE | 183 |
| 7.2.4.3 | Experimenty s databází GPDS využívající různou kombinaci příznaků a časová náročnost použitých estimátorů | 184 |
| 7.2.5 | Závěr a diskuze | 189 |
| | Literatura | 190 |
| 7.3 | Klasifikátor kontury ruky využívající 1-DIM optimalizační funkci | 194 |
| 7.3.1 | Úvod | 194 |
| 7.3.2 | Jednodimenzionální klasifikátor | 194 |
| 7.3.3 | Závěr | 198 |
| | Literatura | 199 |
| 8 | Dosažené cíle | 200 |
| 8.1 | Část první | 201 |
| 8.2 | Část druhá | 201 |
| 8.3 | Část třetí - závěry pro další rozvoj a praktickou realizaci | 202 |

| | | |
|---|--------------------------------|-----|
| 9 | Závěr | 206 |
| | Dodatky | |
| | Softwarový balík Handwork (HW) | 208 |
| | Seznam vlastních publikací | 209 |
| | Summary | 210 |

Seznam obrázků

| Č.o. | Popis | Str. |
|----------|---|------|
| (3).1 | Mechanický a elektrooptický systém identifikace rozměrů lidské ruky. | 18 |
| (3).2 | Biometrický scanner fi. Allegion. | 19 |
| (3).3 | Umístění pravé ruky v reálném biometrickém scanneru. | 19 |
| (3).4 | Kontura ruky a popis primitiv podle (Barra et al., 2019). | 21 |
| (4).1 | Některé z méně vhodných testovaných materiálů pro jednoduché prahování. | 25 |
| (4).2 | Výkres navrženého biometrického scanneru. | 26 |
| (4).3 | Sestavený biometrický scanner a jeho vnitřní schéma. | 27 |
| (4).4 | Detailní fotografie biometrického scanneru. | 28 |
| (4).5 | Další možnosti umístění ruky v adjustážním přípravku biometrického scanneru. | 29 |
| (6.1).1 | Vliv centrického soudkového (C) a polštářového (B) zkreslení na obraz. | 46 |
| (6.1).2 | Výsledek práce korekčního polynomiálního algoritmu. | 49 |
| (6.1).3 | Kalibrační obrazy. | 51 |
| (6.1).4 | Schematický diagram zvolených a testovaných kombinací CEE. | 53 |
| (6.1).5 | Schematický diagram auto-kalibračního procesu. | 53 |
| (6.1).6 | Časové nároky na korekci soudkovité distorze obrazu na různých typech procesorů. | 58 |
| (6.1).7 | 3D zobrazení tvarů primární $fitness_1$ funkce. | 59 |
| (6.1).8 | Optimalizátor jDE. Nejlepší nalezená odchylka. | 59 |
| (6.1).9 | Výsledek konvergence optimalizátorů SGA a DE (rand2best1/bin). | 60 |
| (6.1).10 | Výsledek evoluce DE, nejlepší hodnota dosažená v jednotlivých iteracích. | 60 |
| (6.1).11 | Výsledek konvergence optimalizátorů PSO a RPSO. | 61 |
| (6.1).12 | Optimalizátory JADE a jDE. | 61 |
| (6.1).13 | Optimalizátory SADE a CMA. | 62 |
| (6.1).14 | Časové nároky testovaných EA. | 63 |
| (6.1).15 | Finální výsledek kalibrace, optimalizátor jDE+jDE. | 64 |
| (6.1).16 | Vliv hodnot P_{op} a G_{en} na $fitness_1$ funkci u optimalizátoru jDE a CMA. | 66 |
| (6.1).17 | Finální výsledek kalibrace, optimalizátor jDE+CMA. | 66 |
| (6.1).18 | Finální výsledek kalibrace, optimalizátor CMA+CMA. | 67 |
| (6.1).19 | Záznam konvergence pro $fitness_2$ funkci u jednotlivých CEE. | 67 |
| (6.1).20 | Dosažená přesnost pro různé kombinace stupňů polynomu \mathcal{F} ; typ CMA+CMA. | 68 |
| (6.1).21 | Několik reálných příkladů korekce distorze. | 69 |
| (6.1).22 | Několik příkladů, které prezentují účinnost navrženého CEE (jDE+jDE). | 70 |
| (6.2).1 | Schématický diagram navrženého algoritmu DMAM. | 83 |
| (6.2).2 | Příklad práce algoritmu DMAM. | 87 |
| (6.2).3 | Detail z práce algoritmu DMAM. | 88 |
| (6.3).1 | Obraz kalibračních značek bez distorze a obraz zatížený distorzí a získaný jedním ze 136 navržených distorzních polynomů. | 95 |
| (6.3).2 | Interpretace výsledných experimentálních dat. | 97 |
| (6.3).3 | Výsledné hodnoty ε pro distorzní polynomy S1, S2, S3. | 101 |
| (6.3).4 | Výsledné hodnoty ε pro distorzní polynomy S4, S5, S6. | 102 |
| (6.3).5 | Grafy A, B, C zobrazují hodnoty Δ pro jednotlivé distorzní polynomy S1, S2, S3. | 103 |
| (6.3).6 | Grafy D, E, F zobrazují hodnoty Δ pro jednotlivé distorzní polynomy S4, S5, S6. | 104 |
| (6.3).7 | Doplňující informace pro polynomy (S1, S2, S3). | 105 |
| (6.3).8 | Experimentální výsledky s polynomy: (M1)..(M17). | 106 |
| (6.3).9 | Statistické ukazatele: AVGROW pro polynomy (S1)..(S6) | 107 |
| (6.3).10 | Statistické ukazatele: AVGCOL a STDDEV pro polynomy (S1)..(S3) | 107 |
| (7.1).1 | Postupný proces zpracování obrazu z digitálního fotoaparátu. | 119 |
| (7.1).2 | Možné nastavení počátečního směru D_{ir1} . | 123 |
| (7.1).3 | Kontura ruky s vyznačenými důležitými body a popisem. | 124 |
| (7.1).4 | Jednotlivé obrazy ruky při procesu klasifikace. | 129 |
| (7.1).5 | Evoluční proces a jeho vliv na zarovnání kontur. | 138 |
| (7.1).6 | Časová náročnost testovaných optimalizátorů. | 141 |
| (7.1).7 | Časové nároky nutné k dosažení optimální hodnoty. | 142 |
| (7.1).8 | Optimalizátor CMA-ES. | 144 |
| (7.1).9 | Optimalizátor EPSDE. | 144 |

| | | |
|----------|--|-----|
| (7.1).10 | Optimalizátor JADE. | 144 |
| (7.1).11 | Optimalizátor jDE. | 145 |
| (7.1).12 | Optimalizátor SADE. | 145 |
| (7.1).13 | Optimalizátor SANSDE. | 145 |
| (7.1).14 | Optimalizátor SPDE. | 145 |
| (7.1).15 | Vliv hodnoty L_p^{EPSDE} na schopnost konvergence. | 146 |
| (7.1).16 | Záznam konvergence populace pro THID. | 147 |
| (7.1).17 | Schéma postupu při odhadu přesnosti navržené metody pro různé hodnoty κ_M, κ_S . | 147 |
| (7.1).18 | Dosažené hodnoty <i>fitness</i> pro postupně se zvyšující hodnoty κ_M, κ_S . | 148 |
| (7.1).19 | Vliv hodnot $N_{pop}^{EPSDE}, G_{en}^{EPSDE}$ na výslednou hodnotu <i>FRR</i> %. | 149 |
| (7.1).20 | Křivky <i>FAR, FRR</i> pro různé hodnoty λ_{ERR} a pro různé hodnoty κ_S . | 154 |
| (7.1).21 | ROC křivky pro estimátor A1 a různé hodnoty κ_S . | 155 |
| (7.1).22 | ROC křivky pro estimátor A8 a různé hodnoty κ_S . | 158 |
| (7.1).23 | Různě dlouhé prsty u identické osoby na různých snímcích v databázi HGDB. | 161 |
| (7.2).1 | Schéma procesu získání kontury. | 180 |
| (7.2).2 | Křivky konvergence populace. | 184 |
| (7.2).3 | Křivky <i>FAR, FRR</i> a <i>ROC</i> pro jednotlivé experimenty. | 187 |
| (7.3).1 | Dva Radial Distance Diagramy (RDD) vytvořené z bodu P_{wc} na zápěstí ruky. | 195 |
| (7.3).2 | Výsledek klasifikace prvních 10ti obrazů kontur rukou z databáze THID. | 197 |

Seznam tabulek

| Č.t. | Popis | Str. |
|----------|--|------|
| (5.1).1 | Odchylkové vektory (perturbation vectors). | 33 |
| (5.2).1 | Odchylkové vektory použité v optimalizátoru EPSDE. | 36 |
| (6.3).1 | Výsledné hodnoty z Obr. (6.3).3A v detailu. | 98 |
| (7.1).1 | Segmenty c_i kontury \mathbf{C} a jejich vztah k bodům P_C | 127 |
| (7.1).2 | Pomůcka pro výpočet krajních bodů segmentu c_i^{LR} | 128 |
| (7.1).3 | Vazby mezi dimenzemi x_j chromozomu \mathbf{X}_i a konturami a zvolená sestavení kontur. | 132 |
| (7.1).4 | Výsledek konvergence, vzorky THID: P001-03-M vs P057-08-S. | 148 |
| (7.1).5 | Zvolené kombinace rozlišení obrazu, kontur ruky, klasifikačního kritéria. | 150 |
| (7.1).6 | Hodnoty κ_S v jednotlivých generacích. | 151 |
| (7.1).7 | Hodnoty FAR, FRR, EER , A1: THID, 1280x960, $c_{4-8}^{M_{co}}, c_{4-8}^{S_{ea}}$, 4 prsty, $R_H + R_I, P_{K3-6}, D_{im} = 7, E9$. | 154 |
| (7.1).8 | Hodnoty FAR, FRR, EER , A2: THID, 1280x960, $c_{4-8}^{M_{ea}}, c_{4-8}^{S_{ea}}$, 4 prsty, $R_H + R_I, P_{K3-6}, D_{im} = 7, E9$. | 155 |
| (7.1).9 | Hodnoty FAR, FRR, EER , A3: THID, 1280x960, $c_{4-7}^{M_{ea}}, c_{4-7}^{S_{ea}}$, 4 prsty, $R_I, P_{K3-6}, D_{im} = 7, E5$. | 155 |
| (7.1).10 | Hodnoty FAR, FRR, EER , A4: THID, 640x480, $c_{4-8}^{M_{co}}, c_{4-8}^{S_{ea}}$, 4 prsty, $R_H + R_I, P_{K3-6}, D_{im} = 7, E9$. | 156 |
| (7.1).11 | Hodnoty FAR, FRR, EER , A5: THID, 640x480, $c_{4-7}^{M_{co}}, c_{4-7}^{S_{ea}}$, 4 prsty, $R_H + R_I, P_{K3-6}, D_{im} = 7, E10$. | 156 |
| (7.1).12 | Hodnoty FAR, FRR, EER , A6: THID, 640x480, $c_{1-8}^{M_{ea}}, c_{1-8}^{S_{ea}}$, 5 prstů, R_I, P_{K1-6}, P_{K2} fixed, $D_{im} = 8, E11$. | 156 |
| (7.1).13 | Hodnoty FAR, FRR, EER , A7: THID, 640x480, $c_{1-8}^{M_{co}}, c_{1-8}^{S_{ea}}$, 5 prstů, $R_H + R_I, P_{K1-6}, D_{im} = 9, E8$. | 156 |
| (7.1).14 | Hodnoty FAR, FRR, EER , A8: GPDS, 1403x1021, $c_{4-8}^{M_{co}}, c_{4-8}^{S_{ea}}$, 4 prsty, $R_H + R_I, P_{K3-6}, D_{im} = 7, E9$. | 157 |
| (7.1).15 | Hodnoty FAR, FRR, EER , A9: GPDS, 640x480, $c_{1-8}^{M_{ea}}, c_{1-8}^{S_{ea}}$, 5 prstů, $R_I, P_{K1-6}, D_{im} = 9, E3$. | 158 |
| (7.1).16 | Hodnoty FAR, FRR, EER , A10: GPDS, 640x480, $c_{1-8}^{M_{ea}}, c_{1-8}^{S_{ea}}$, 5 prstů, R_I, P_{K1-6}, P_{K2} fixed, $D_{im} = 8, E11$. | 158 |
| (7.1).17 | Hodnoty FAR, FRR, EER , A11: GPDS, 640x480, $c_{4-8}^{M_{co}}, c_{4-8}^{S_{ea}}$, 4 prsty, $R_H + R_I, P_{K3-6}, D_{im} = 7, E9$. | 159 |
| (7.1).18 | Hodnoty FAR, FRR, EER , A12: HGDB, 1754x1275, $c_{4-8}^{M_{co}}, c_{4-8}^{S_{ea}}$, 4 prsty, $R_H + R_I, P_{K3-6}, D_{im} = 7, E9$. | 159 |
| (7.1).19 | Porovnání s ostatními pracemi. | 160 |
| (7.1).20 | Hodnoty FAR, FRR, EER dosahované různými autory a různými metodami. | 162 |
| (7.2).1 | Příznaky použité při klasifikaci. | 178 |
| (7.2).2 | Vliv N_{POP} a N_{GEN} na přesnost výsledků. | 184 |
| (7.2).3 | Hodnoty FAR, FRR, EER , (A13) příznaky $f_{1, \dots, 18}$, 4 prsty + všechny Příznaky, bez palce. | 185 |
| (7.2).4 | Hodnoty FAR, FRR, EER , (A14) příznaky $f_{1, \dots, 6} + f_{11, \dots, 13} + f_{14, \dots, 16}$, žádný prst, není použit EA. | 185 |
| (7.2).5 | Hodnoty FAR, FRR, EER , (A15) Všech 5 prstů + všechny příznaky, tj. $f_{1, \dots, 18}$ + palec. | 185 |
| (7.2).6 | Hodnoty FAR, FRR, EER , (A16) 4 prsty a žádné příznaky, tj. jen f_7, f_8, f_9, f_{10} . | 186 |
| (7.2).7 | Hodnoty FAR, FRR, EER , (A17) 5 prstů a žádný příznak, tj. f_7, f_8, f_9, f_{10} + palec + kontura ruky. | 186 |
| (7.2).8 | Nejlepší a nejhorší výsledky FAR, FRR, EER , pro vybrané hodnoty κ_M, κ_S a jednotlivé experimenty (A13) – (A17). | 187 |

Matematické značení

Seznam důležitých proměnných, značení klasifikátorů v experimentech.

| Označení | Popis |
|---|--|
| I | Grafický obraz. |
| I_D'' | Obraz obsahující kalibrační značky M_C . |
| I_D | Filtrovaný obraz získaný z obrazu I_D'' . |
| I_w, I_h | Šířka a výška obrazu I . |
| $P_1, P_2, P_d(x_d, y_d)$ $P_{dc}(x_{dc}, y_{dc}), P_w$ | Body v E^2 , označuje vždy nějaký pixel obrazu. |
| $D(P_d, P_{dc}), D(P_w, P_{uc}), \rho$ | Euklidovská metrika v E^2 , vzdálenost pro body P_d, P_{dc} a P_w, P_{uc} a P_1, P_2 . |
| φ_d | Úhel, který svírá přímka, na které leží body P_d, P_{dc} . |
| $\mathcal{F}(r_d), \mathcal{F}_{odd}(r_d),$ $\mathcal{F}_{even}(r_d), \mathcal{F}_{NEW}$ | Vyrovňovací polynomy stupně N . |
| V_u, V_d | Pomocné vektory, $V_u(x_u - x_{uc}, x_u - x_{uc}), V_d(x_d - x_{dc}, y_d - y_{dc})$. |
| I_D, I_U | Obraz I zatížený distorzí a obraz I nezatížený distorzí – korigovaný. |
| C_f, \widehat{C}_f | Korekční koeficienty polynomu $\mathcal{F}, \mathcal{F}_{odd}, \mathcal{F}_{even}$. |
| $L(r)$ | Korekční polynom pro tzv. „division-model“. |
| c_i | Člen korekčního polynomu C_f, \widehat{C}_f . |
| $k_1, k_2, k_i, \lambda_1, \lambda_2$ | Členy korekčního polynomu pro „division-model“. |
| M_C, M_R, M_F | Kalibrační značky, referenční značky a značky korigované s využitím polynomu \mathcal{F} . |
| M_{FX}, M_{FY} | Počet kalibračních značek v ose X a Y . |
| $\mathcal{H}_u, \mathcal{H}_d$ | Prostor souřadnic korigovaného obrazu (undistorted), a nekorigovaného (distorted). |
| $(\tilde{x}, \tilde{y})_{i,j}, (\hat{x}, \hat{y})_{i,j}, (x, y)_{i,j}$ | Souřadnice kalibrační značky na nekorigovaném obraze, korigovaném obraze a pozice referenčních značek. |
| A_S, A_w, A_h | Filtrovaná maska o rozměrech $A_w \times A_h$. |
| $fitness_1, fitness_2$ | Označení účelové funkce svázané s evolučním algoritmem. |
| $\mathcal{F}_{CMA}, \mathcal{F}_{jDE}$ | Evoluční proces využívající optimalizátor CMA. |
| \widehat{C}, B, D | Kovarianční matice \widehat{C} algoritmu CMA a další matice použité v algoritmu CMA. |
| P_{op}^{\dots} | Populace jedinců evolučního algoritmu, ... - značí zvolený evoluční algoritmus. |
| $X_i^{\dots}, X_S^{\dots}, X_{rj}$ | Chromozom jedince v populaci, ... - značí zvolený evoluční algoritmus. |
| N^{\dots} | Počet jedinců v populaci, ... - značí zvolený evoluční algoritmus. |
| T, T_1, T_2 | Ukončovací podmínka. Obvykle počet generací. |
| M_{CX}, M_{CY} | Počet kalibračních značek v osách X, Y . |
| X_t | Trial vektor vypočítaný s využitím $\overline{V}_p, X_S^{jDE}, X_{rj}, X_{best}^{jDE}$. |
| \overline{V}_p | Odchylkový (perturbation) vektor pro algoritmus DE a jeho deriváty. |
| X_{best}^{\dots} | Chromozom nejlepšího jedince v populaci, ... - značí zvolený evoluční algoritmus. |
| G_{en} | Počet generací použitého evolučního algoritmu, ... - značí zvolený evoluční algoritmus. |
| G, G_{en} | Někáká konkrétní generace použitého evolučního algoritmu. |
| $actualG_{en}^{\dots}$ | Aktuální generace použitého evolučního algoritmu, ... - značí zvolený evoluční algoritmus. |
| SGA, CMA, DE, PSO, JADE, jDE, RPSO, SADE, EPSDE, SANSDE | Označení pro použité evoluční optimalizátory a jejich slovní význam tam, kde existuje: SGA-Simple Genetic Algorithm, CMA-Covariance Matrix Adaptation Evolution Strategy, PSO-Particle Swarm Optimization (rojení částic), RPSO-Repulsive Particle Swarm Optimization, EPSDE-algoritmus diferenciální evoluce se souborem parametrů a křížením a mutací (Differential Evolution Algorithm with Ensemble of Parameters and Mutation and Crossover). |
| EA | Obecné označení Evolučního Algoritmu (EA) tam, kde nebylo vhodné popisovat nějaký konkrétní evoluční algoritmus. |
| $MSE, PSNR, RMSE$ | Hodnoty MSE (Mean Square Error) a PSNR (Peak Signal-to-Noise Ratio), RMSE – Root Mean Square Error – chyba založená na Euklidovské metrice. |
| $F, F_l, F_u, P_{cr}, \tau_1, \tau_2$ | Pomocné proměnné algoritmu DE, jDE, EPSDE a obecně všech použitých derivátů DE. |

| | |
|--|---|
| D_{im} | Počet dimenzí chromosomu X_i^{\dots} . |
| $center_x, center_y$ | Střed kalibračního obrazu. |
| D_{om} | Prostor možných řešení dané evoluční úlohy pro každou dimenzi. |
| cr, cr_{mean}, CR_{mem} | Dva vektory a matice, interní pracovní proměnné algoritmu EPSDE. |
| \mathcal{H} | Prostor možných řešení dané optimalizační úlohy, obvykle reprezentovaný jako $D_{om}^1 \times D_{om}^2 \times \dots \times D_{om}^k k \in \mathbb{N}^+$. |
| $\mathcal{F}_{primary-jDE}, \mathcal{F}_{secondary-jDE}$ | Účelové funkce algoritmu pro korekci distorze obrazu. Algoritmus DMAM. |
| λDM | Matematický model určený pro korekci distorze obrazu. |
| $\varepsilon_{DM}, \varepsilon_{\lambda DM}, \Delta$ | Měřená chyba s definovanou metrikou např. RMSE - „Root Mean Square Error“. |
| AVGROW, AVGCOL, STDDEV | Zvolené statistické ukazatele. Průměr z řádku, sloupce a směrodatná odchylka. |
| \mathcal{M}, \mathcal{S} | Modelový vzorek kontury ruky a datový vzorek identifikované osoby. |
| \mathcal{M}_{co} | Modelový vzorek kontury \mathcal{C} , který nebyl ořezán pro účely výpočtu s EA. |
| $\mathcal{M}_{ea}, \mathcal{S}_{ea}$ | Modelový vzorek a datový vzorek, oba vhodně ořezané pro účely výpočtu s využitím EA. |
| $I_{RGB}, I_{HSB}, I_{B\&W}^*$ | Obrazy kontury ruky. Obraz získaný z digitálního fotoaparátu v modelu RGB, HSB. * - značí obraz B&W s artefakty, které budou odfiltrovány. |
| $I_{B\&W}$ | Obraz B&W bez artefaktů, tj. filtrovaný některým z filtrů \mathcal{F}_{\dots} . |
| B_{lim}, H_{lim} | Hodnoty „Brightness“ a „Hue“ použité při transformaci T_{HSB} , H-barva, S-sytost, B-jas. |
| $T_{HSB}, T_{B\&W}$ | Označení transformace z obrazu I_{RGB} do I_{HSB} a z I_{HSB} do $I_{B\&W}^*$. |
| T_{YCrCb} | Transformace obrazu RGB resp. I_{RGB} do modelu $YCrCb$. |
| $P_{x,y}^{RGB}, P_{x,y}^{HSB}$ | Bod v obraze I_{RGB} a bod v obraze I_{HSB} . |
| $I_{B\&W}^* \xrightarrow{\mathcal{F}} I_{B\&W}$ | Filtrace obrazu $I_{B\&W}^*$ s využitím filtru \mathcal{F}_{\dots} , kde „...“ značí číslo filtru. |
| \mathcal{F}_{\dots} | Filtry určené pro filtraci šumu v obraze $I_{B\&W}^*$, kde „...“ značí číslo filtru. |
| γ, σ | Pracovní oblasti filtru \mathcal{F}_{\dots} . |
| L_{HAND}, L_{MF} | Délka ruky a délka prostředníčku. |
| P_{CSTART}, P_{HSTART} | První bod na kontuře ruky zleva. Je to bod na zápěstí, ze kterého je vypočítávána kontura ruky. |
| $D_{ir\dots}$ | Směr, jakým bude algoritmus výpočtu kontury pracovat, „...“ značí číslo směru ve smyslu 8mi okolí, čísla 0-7. |
| \mathcal{C} | Kontura ruky získaná z obrazu $I_{B\&W}$. |
| P_j^C, P_j^{RDD} | Bod náležející kontuře \mathcal{C} a křivce R_{DD} . |
| R_{DD} | Křivka Radial Distance Diagram (RDD), jednodimenzionální křivka. |
| P_{RDD} | Bod, ze kterého je vypočítáván RDD. |
| \mathcal{C}_{LR} | Modifikovaná kontura ruky určená pro výpočet s využitím Lineární Regrese (LR). |
| $P_{C1} - P_{C14}$ | Klíčové body na kontuře R_{DD} . |
| $P_{K1} - P_{K6}$ | Body definující polohy kloubů prstů. |
| $A_1 - A_6$ | Osy jednotlivých prstů ruky získané využitím algoritmu lineární regrese a kontury \mathcal{C}_{LR} . |
| P_{WC} | Bod středu zápěstí, je to bod v polovině vzdálenosti mezi body P_{C1} a P_{C14} . |
| L_{RDD} | Vzdálenost na ose X od bodu P_{WC} k bodu P_{RDD} . |
| \mathcal{C}_{RDD} | Kontura získaná z RDD, křivka ve 2D. |
| $c_i^{RDD}, c_i^{LR}, c_i^M, c_i^S$ | Segmenty bodů, náležející jednotlivým prstům a částem ruky. Segmenty jsou součástí kontur $\mathcal{C}_{RDD}, \mathcal{C}_{LR}, \mathcal{M}, \mathcal{S}$. |
| \mathfrak{R} | Označení regresní funkce. |
| $A, B, P, R, \bar{n}_{AB}, \bar{\mu}_{AB}$ | Pomocné body při výpočtu segmentů kontury ruky, normálový a směrový vektor úsečky AB. |
| Γ | Prostor možných řešení účelové funkce evolučního algoritmu. |
| γ_i | Počet možných řešení v dané dimenzi i prostoru Γ . |
| P_{CMASS} | Těžiště kontury ruky. |
| $P_{\Omega}(x_{\Omega}, y_{\Omega})$ | Bod ležící uprostřed obrazu $I_{B\&W}$. |
| \mathfrak{Z}_{EA} | Evoluční proces s využitím evolučního optimalizátoru. |
| $\mathcal{E}, \mathcal{E}_{opt}$ | Výsledek evolučního procesu. Celé nebo reálné číslo udávající podobnost kontur \mathcal{M}, \mathcal{S} ve smyslu použité metriky. Obvykle Euklidovská metrika. |
| P_j^S, P_i^M | Body na konturách \mathcal{M}, \mathcal{S} . |
| P_{min}^M | Bod na kontuře \mathcal{M} , který nejbližší zvolenému bodu P_j^S kontury \mathcal{S} . |
| $fitness, fitness_p$ | Označení účelové funkce svázané s použitým evolučním algoritmem. |

| | |
|--|--|
| $D_{ist}(x, y)$ | Matice vzdáleností bodů P_i^M a bodů obrazu $I_{B\&W}$. |
| \mathcal{L} | Klasifikační funkce s ohledem na D_{om} . |
| κ_M, κ_S | Udává, kolikátý bod kontury \mathcal{M} nebo \mathcal{S} bude použit pro klasifikaci. |
| η_M, η_S | Udává počet bodů kontury \mathcal{M} nebo \mathcal{S} s ohledem na κ_M, κ_S . |
| t, R | Parametry algoritmu ICP, translační vektor v rovině XY a rotační matice 3×3 . |
| $P\langle xxx \rangle - \langle yy \rangle - M.BMP$ $P\langle xxx \rangle - \langle yy \rangle - S.BMP$ | Označení obrazu \mathcal{M} příp. \mathcal{S} , který je ve formě souboru na disku. M -značí datový soubor modelové kontury ruky a S -značí datový soubor vzorku od identifikované osoby. |
| $\lambda_{EER}, \lim, \Sigma G_{OL}, \Sigma I_{UL},$ $\Sigma G_{TOT}, \Sigma I_{TOT}$ | Pomocné proměnné při výpočtu hodnot FAR, FRR, EER . ΣG_{OL} - celkový počet všech pravých (genuine) vzorků, které odhadl klasifikátor jako pravé (tzv. genuine). ΣG_{TOT} - celkový (skutečný) počet všech pravých (tzv. genuine) vzorků v testovací databázi. Např. pro databázi THID je $G_{TOT} = 104 \times 10 = 1040$, 104 osob, 10 snímků od každé osoby a dále $\Sigma I_{TOT} = 108160 - 1040 = 107120$, kde 108160 je celkový počet kombinací porovnávaných obrazů tj. 1040×104 . |
| $CR_{mem}, cr_{mean},$ cr | Pomocné proměnné při výpočtu s využitím algoritmu EPSDE, řídí systém křížení jedinců populace. |
| FAR, FRR, EER, ROC TPR, FPR | Standardní statistické ukazatele používané v oblasti strojového učení, FAR-false acceptance rate, FRR-false rejection rate, TPR-true positive rate, FPR-false positive rate, EER-equal error rate neboli bod, kde křivka ROC protíná hlavní diagonálu v ROC diagramu. Bod EER je obvykle udáván v procentech v rozmezí $(0.0, 1.0)$. V některých případech je doplněna i hodnota <i>fitness</i> v pixelech. ROC-receiver operating characteristics. Všechny použité ukazatele jsou detailně popsány pod jednotným heslem „confusion matrix“ neboli „matice záměn“. Křivka ROC a především bod EER je nutné vypočítat iterativně numericky z hodnot TPR a FPR a zjistit, v kterém okamžiku se bod ROC křivky nejvíce přiblíží hlavní diagonále v ROC diagramu. Dále platí, že $FAR = \frac{\Sigma I_{UL}}{\Sigma I_{TOT}}$ a $FRR = \frac{\Sigma G_{OL}}{\Sigma G_{TOT}}$ a oba koeficienty se udávají v procentech. |
| u, v | Vektory vstupních hodnot u jednodimenzionálního klasifikátoru. Jedná se o hodnoty jednotlivých bodů z Radial Distance Diagramu. |
| u^*, v^* | Modifikované vektory u, v vhodné pro další výpočet. |
| l_{min}, l_{max} | Minimální a maximální hodnota z vektorů u, v . |
| C, D, E | Pomocné matice při výpočtu optima u jednodimenzionálního optimalizátoru. |
| z | Pomocný vektor při výpočtu optima u jednodimenzionálního optimalizátoru. |

V textu disertační práce je velmi často odkazováno na různé typy metod/algoritmů, u nichž neexistuje český ekvivalent k dané anglické terminologii vzhledem k novosti daných metod. V některých případech by překlad byl poměrně krkolomný. V tom případě je použita zkratka ve formátu, jak ji definoval autor metody ve své originální práci nebo je použita zkratka odvozená od názvu metody. V některých případech neexistuje slovní reprezentace zkratky a pak je používáno jen dané označení metody např. algoritmus jDE, nebo jen jDE. Někde je celý, nezkrácený název metody příliš dlouhý a nepoužívají ho ani autoři metody a místo něj se používá jen zkratka např. optimalizátor EPSDE nebo jen EPSDE. Názvy některých metod jako např. „direct-mapping“ nebo „back-mapping“ nebo „division-model“ jsou ponechány v původním formátu v anglickém jazyce, a to z toho důvodu, že tato označení se v anglické odborné literatuře využívají již téměř 100 let. Čeština je poměrně ohebná a v průběhu let bylo různými autory „vynalezeno“ více ekvivalentů pro nějaký anglický termín – příkladem jsou ekvivalentní pojmy „soudkové“, „soudkovité“, „barelové“, „sudové“ zkrácení obrazu. Stěžejní díla v dané vědecké oblasti jsou však vždy publikována s anglickými výrazy ve významných impaktovaných časopisech, a proto je v této disertační práci ponecháme také.

1 Úvod

1.1 Obecný pohled do historie

Tvorbou nástrojů si člověk odpradávně snažil ulehčit život, zjednodušit práci a je jisté, že některé nástroje měly sloužit i k jeho obveselení. Prvotní nástroje sloužily k lovu, obraně, ale i útoku na nepřítele. Je vždy jednodušší použít dobře sestavený nástroj k urychlení práce a snížení fyzické náročnosti, než se lopotit znovu a znovu s notoricky se opakujícími úkony. Člověk objevil důležitou vlastnost strojů, a to, že jsou zdrojem menšího počtu chyb, než když stejnou rutinní práci odvádí on a stejně tak stroje podstupují menší riziko a jsou relativně snadno obnovitelné v případech poruch nebo havárií. V době kamenné to byl pazourek a kamenná sekera, jako vhodný nástroj. Člověk uměl rozdělat oheň. Později se člověk naučil zpracovávat kovy a začal v širších souvislostech chápat dění kolem sebe. To, jak věci fungují, jak se vzájemně ovlivňují a také jaké mohou být důsledky lidského jednání. To byl začátek nové éry, která ho přivedla k obdělávání půdy, stavbě měst, kolonizaci, chápání času a k obecnému rozmachu vědy.

Cíleně využívaná antropometrická identifikace je rodu Homo Sapiens známá již mnoho tisíc let. Dokonce i lidský mozek běžně používá „antropometrické“ metody, aniž si to uvědomujeme např. při interakci s ostatními lidmi. Jsme schopni odhadnout proporce postavy, obličej, rukou nebo nohou a uvědomit si, zdali zkoumaný objekt známe či nikoli. Otisk ruky jako identifikátor humanoidní formy života je znám již z prehistorických dob, kdy lidé ještě žili v jeskyních, kde se tyto znaky se dochovaly dodnes. Ve starověkém Egyptě před 4-5tis. let se běžně využíval otisk ruky jako identifikátor osob při podpisu smluv. První doložené písemné zprávy o širším využívání antropometrické identifikace osob na základě otisku prstů jsou k dispozici z let 1900-2000 p.ř.n.l. jako ochrana proti padělání smluv ve starověkém Babylonu a také pro identifikaci zločinců. Otisk palce tehdy nahrazoval podpis. Rozsáhlejší informace využívání otisku prstů pro identifikaci osob je k dispozici z období vlády Babylonského krále Chammurapiho z let 1792-1750 p.ř.n.l.

V prvních stoletích, které následovaly po narození Krista, se jak otisk jednotlivých prstů, tak i otisk celé ruky běžně používal v mnoha Evropských i Asijských kulturách a po dlouhá století novověku zůstávala tato metoda identifikace osob prakticky neměnná. Otisk se provedl např. na pergamen, do jílu nebo pečetního vosku. Samotná identifikace osoby pak probíhala pouze díky rozpoznávacím schopnostem člověka, který otisk ověřoval. Detailní popis charakteristik lidské kůže a také otisků prstů byl oficiálně podán až v roce 1686 italským lékařem Marcellem Malpighinim, který pracoval na universitě v Bologni. Podrobný anatomický popis kontur kůže na prstech podal také Německý anatom Govard Bidloo ve své knize "Anatomy of the Human Body" z roku 1685. V roce 1858 Sir William James Herschel (syn astronoma Williama Herschela) jakožto civilní úředník Britského impéria v Indii, použil otisk celé ruky na smlouvě s Indickým podnikatelem jménem Rajyadhar Konai na dodávky stavebního materiálu na nově vznikající pozemní komunikace. Nápad využívat otisky ruky se ujal a na základě vlastního výzkumu Herschel zjistil, že pro jednoznačnou identifikaci je dostačující využít pouze otisku jednotlivých prstů. Francouzský kriminalista Alphonse Bertillon vypracoval a v roce 1882 prezentoval metodu na identifikaci zločinců podle měření charakteristických znaků na těle člověka – 11 různých tělesných rozměrů. Bertillon svou metodu nazval Antropometrie, avšak později byla přejmenována na Bertillonáž. Zavedl také využívání fotografií pro identifikaci zločinců.

V průběhu 19. století se antropometrické identifikační metody rozvíjely jen tak, jak to umožňovala tehdejší technická úroveň. Běžně se využíval otisk ruky, otisk jednoho nebo více prstů, rozměry částí lidského těla, kreslené obrázky lidí a po příchodu fotografie také fotografická identifikace člověka. Až do nástupu počítačů v průběhu 80. let byly metody biometrické identifikace závislé čistě na lidském faktoru a také na primitivních mechanických strojích v případě měření tělesných rozměrů. Postupem 80. a 90. let byly vypracovány sofistikované metody pro identifikaci osob např. podle otisku prstů,

fotografie sítnice nebo duhovky, geometrie tváře, geometrie ucha, stylu chůze, stylu písma, 2D a 3D geometrie ruky nebo pěsti, rentgenový snímek zubů, otisk chodidla, struktura DNA, struktura žil na zápěstí a mnoho dalších.

1.2 Uspořádání kapitol a základní přehled řešených témat v disertaci

Předkládaná disertace je výsledkem dlouhodobé práce. Zpracovávaná témata spadají primárně do poměrně atraktivní oblasti umělé inteligence a strukturálních metod rozpoznávání. Práce je rozdělena na několik částí. V úvodu je podán standardní přehled historie v oblasti antropometrie s důrazem na identifikaci osob s využitím kontury ruky. Další kapitolou je detailnější popis navrženého a sestaveného biometrického scanneru, jehož účelem je pořizování snímků kontur rukou. Velmi důležitou součástí všech navržených metod jsou různé druhy evolučních algoritmů. Další kapitola je proto věnována detailnějšímu popisu vybraných evolučních algoritmů, a to především derivátů diferenciální evoluce. Je zde také popsán algoritmus označovaný jako cma-es, který je použit jen v kapitole korekcí optických vad, protože v dalších úlohách podával velmi špatné výsledky.

Při pořizování snímků z biometrického scanneru pomocí digitálních fotoaparátů dochází nutně k různým aberacím pořizovaného obrazu. Velmi důležitou částí disertace je šestá kapitola nesoucí název 'Metody kalibrace obrazu biometrického scanneru'. Jsou zde popsány zcela nové metody pro odstraňování soudkové i polštářové distorze obrazu které současně využívají pokročilé evoluční optimalizátory. Jsou diskutovány jak čistě polynomiální korekční metody, tak i poměrně užší vědní oblast využívající metody označované jako polynomiální korekční metody s lomeným výrazem. Dále budou tyto metody označovány jako tzv. „division methods“ podle originálních zahraničních prací. V oblasti čistě polynomiálních korekčních metod byl navržen nový typ tzv. kaskádového evolučního estimátoru, který umožňuje velmi efektivně a s vysokou přesností korigovat aberaci centrické polštářové i soudkové distorze obrazu. V oblasti „division methods“ byla navržena a na velmi rozsáhlém souboru dat otestována nová korekční metoda, která vykazuje výrazně lepší vlastnosti než dříve používané korekční metody.

V sedmé kapitole, která tvoří pomyslnou třetí třetinu obsahu disertační práce jsou předkládány nově navržené metody, které umožňují provádět identifikaci osob pomocí kontury lidské ruky. Podobně jako v předchozí kapitole i zde jsou jako hlavním hnacím motorem navržených estimátorů pokročilé evoluční algoritmy. Kontura ruky je uvažována jako pětiprstá nebo čtyřprstá, přičemž jednotlivé prsty se při identifikaci pohybují v kloubech. Palec má v případě pětiprsté kontury dva klouby. Na základě výzkumu v dané oblasti bylo zjištěno, že tzv. modelové kontury rukou, které reprezentují uchovávaný vzorek, kterým je každá osoba jednoznačně identifikována, musí nést co možná nejvíce velmi přesných informací. V průběhu procesu identifikace jsou pak modelové kontury porovnávány vůči konturám rukou identifikovaných osob. Tyto kontury již nemusí být tak přesné. Naopak je výhodné, aby některé části kontur byly při identifikaci zcela vynechány. Bylo také zjištěno, že díky použití vybraných evolučních algoritmů není nutné řešit kolize kontur jednotlivých prstů při simulaci. Vynechání striktních omezení kolizí naopak značným způsobem pomáhá při klasifikaci. Při porovnávání kontur, které nejsou stejné, nepoužití detekce kolizí zvyšuje výslednou hodnotu fitness a naopak pro kontury, které jsou si velmi podobné neboli pocházejí od stejného člověka, podporuje rychlejší konvergenci k očekávané limitní hodnotě účelové funkce. Navržené metody pro identifikaci osob jsou zaměřené primárně na identifikaci osob využívající kontury lidské ruky. Mnoho state-of-the-art metod navržených pro tento účel využívá kromě právě zmíněné kontury ruky také různé kombinace geometrických primitiv jako je např. šířka prstů, délka prstů, množiny úhlů, které svírají důležité body na kontuře ruky a mnoho dalších primitiv. Navržená metoda identifikace osob byla také rozšířena tak, aby bylo možné provádět identifikaci osob s využitím výše zmíněných primitiv. Tento způsob identifikace je popisován v druhé části sedmé kapitoly. Pro klasifikaci bylo zvoleno několik různých

základních primitiv jako např. úhly, které svírají přímky procházející důležitými body na kontuře ruky, délka prstů apod.

V poslední části sedmé kapitoly je pak proveden spíše teoretický popis jednodimenzionálního klasifikátoru kontury ruky, který pro klasifikaci využívá Radial Distance Diagramu, který je z kontury ruky vytvořen/vypočten. Klasifikátor nevyužívá evoluční algoritmy a je postavený na modifikované konvoluční funkci a L1-normě (základní metrika). Pro svoji správnou funkci vyžaduje, aby při snímkování byla ruka vložena do adjustážního přípravku biometrického scanneru. Klasifikátor je do disertace zahrnut z toho důvodu, že se jedná o jeden z nejjednodušších způsobů klasifikace kontury ruky. Pro svoji správnou funkci však vyžaduje speciální adjustážní přípravek tzv. „pegs“. Navržený biometrický scanner samozřejmě obsahuje sklopný adjustážní přípravek, který unáší stavící kolíky pro správnou pozici ruky. Nevýhodou tohoto klasifikátoru je to, že neumožňuje virtuálně pohybovat s prsty kontury ruky při klasifikaci a ani celá kontura ruky se nemůže pohybovat a natáčet v rovině XY. Celý klasifikátor je díky tomu velmi citlivý na změnu polohy ruky při snímkování a ruka se do adjustážního přípravku musí vkládat vždy stejně, což je značně problematické především pokud je zde velký časový rozestup mezi snímkováním. Velkou výhodou je pak relativně jednoduchá hardwarová implementace klasifikátoru a z toho vyplívající i velmi vysoká rychlost klasifikace při zachování maximální dosažitelné přesnosti, pokud je k dispozici adekvátní technické vybavení.

V závěrečných kapitolách disertace je pak provedeno shrnutí a jsou zde rozepsány jednotlivé dosažené cíle. Je zde uveden seznam publikací autora této disertace do zahraničních impaktovaných žurnálů a příspěvky ve sbornících mezinárodních konferencí, ze kterých je de-facto celá disertační práce sestavená. Je zde také kapitola, ve které je diskuze dalšího možného rozvoje nebo možné praktické realizace sestavených metod. Nechybí ani odstavec popisující doprovodný software obsahující jednotlivé popisované klasifikátory, který umožnil sestavit veškeré výsledky, které byly poté prezentovány na mezinárodním vědeckém fóru.

2 Cíle disertace

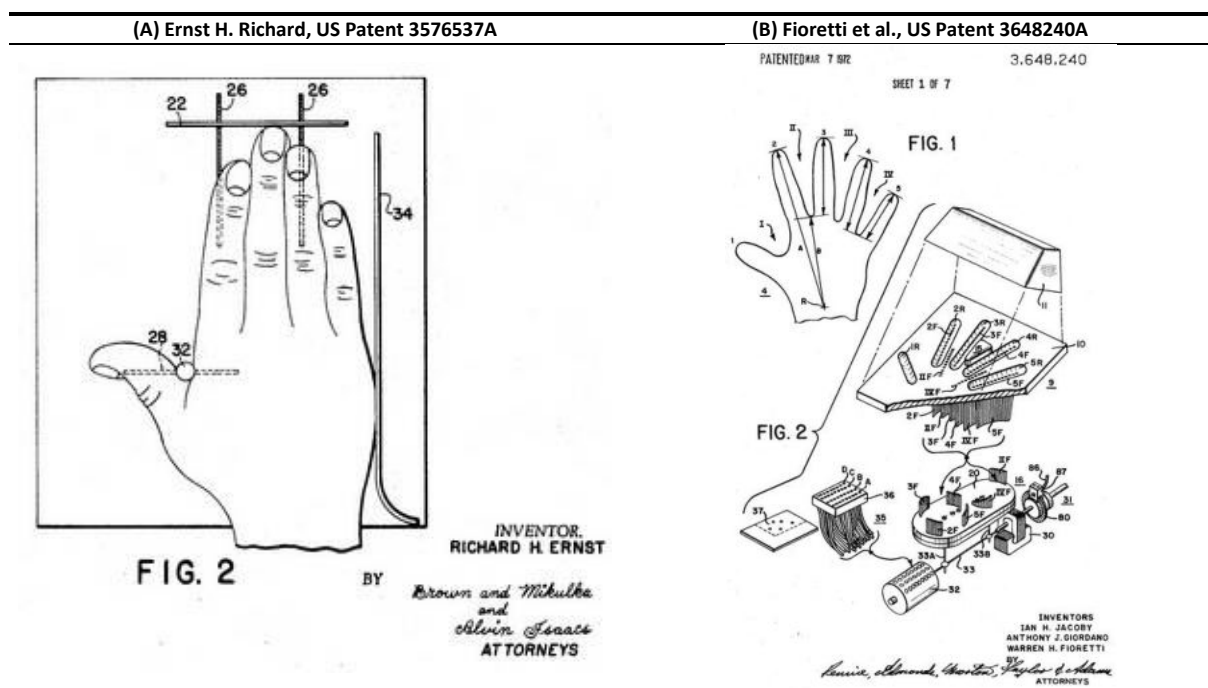
Cílem této disertační práce je:

1. Shrnutí současného stavu poznání v oblasti biometrické autentizace osob s využitím kontury lidské ruky. Podat základní historický přehled v dané oblasti. Podat základní přehled o používaných senzorických systémech, dostupných navržených metodách.
2. Sestavit vhodný nástroj – biometrický scanner universálnějšího charakteru, který by umožnil provádět scanování lidské ruky za účelem biometrické identifikace osob podle 2D kontury lidské ruky. Scanner by měl obsahovat běžně dostupné technické zařízení pro snímání obrazu.
3. Navrhnout základní metodu pro snímání kontury lidské ruky pomocí klasického kompaktního snímacího zařízení (digitální fotoaparát). Navrhnout, sestavit a otestovat algoritmy související s metodologií snímání obrazu kontury lidské ruky jako např. korekce centrické distorze obrazu, formát obrazových dat, filtrace obrazových dat pro dosažení optimálního výsledku.
4. Navrhnout, sestavit a otestovat metodu pro autentizaci osob podle 2D scanu kontury lidské ruky. Využít dostupnou komparativní databázi, která by obsahovala kontury rukou a podle které by bylo možné ověřit, zda jedince, jehož ruka je scanována, již má záznam v databázi.
5. Experimentálně ověřit, funkčnost všech výše uvedených metod v laboratorních a případně i reálných podmínkách. Presentovat výsledky výzkumu na tuzemském nebo zahraničním vědeckém fóru. Biometrický scanner je určen pro identifikaci dospělých osob ve vnitřních prostorech.

3 PŘEHLED HISTORIE A AKTUÁLNÍ STAV

3.1 Část první

Biometrie rozhodně nepatří mezi nově objevená odvětví. Už ve starověkém Egyptě před 4-5tis. lety se používal otisk ruky jako identifikátor při podpisu smluv. Počítače tehdy ještě neexistovaly a otisk se pro účely porovnávání zkoumal pouhým okem. Je známo využití otisku prstu za vlády Babylonského krále Chammurapiho z let 1792-1750 p.ř.n.l. V novodobé historii je pak znám např. Sir William James Herschel (syn astronoma Williama Herschela), který v roce 1858 používal otisk celé ruky na smlouvách o dodávce stavebního materiálu. Později jen otisk palce. Teprve pokračování průmyslové revoluce v éře po II. světové válce přineslo další vývoj (Ernst 1971; Jacoby 1970) a především masivní vývoj počítačů v 80. a 90. letech minulého století umožnil rozmach dané vědní oblasti (Ross & Jain 2007; Duta 2009; Bakshe & Patil 2014; Pavešic et al. 2004). Biometrie současného věku samozřejmě využívá výsledky práce z mnoha vědních odvětví a v mnoha vědních odvětvích je také uplatňována. Je silně závislá na rozvoji a dostupnosti nejmodernějších technologií. Budoucnost biometrie je stoprocentně skvěle zajištěná mnoha společenskými potřebami a objem trhu s biometrickými systémy roste soustavně rok od roku.

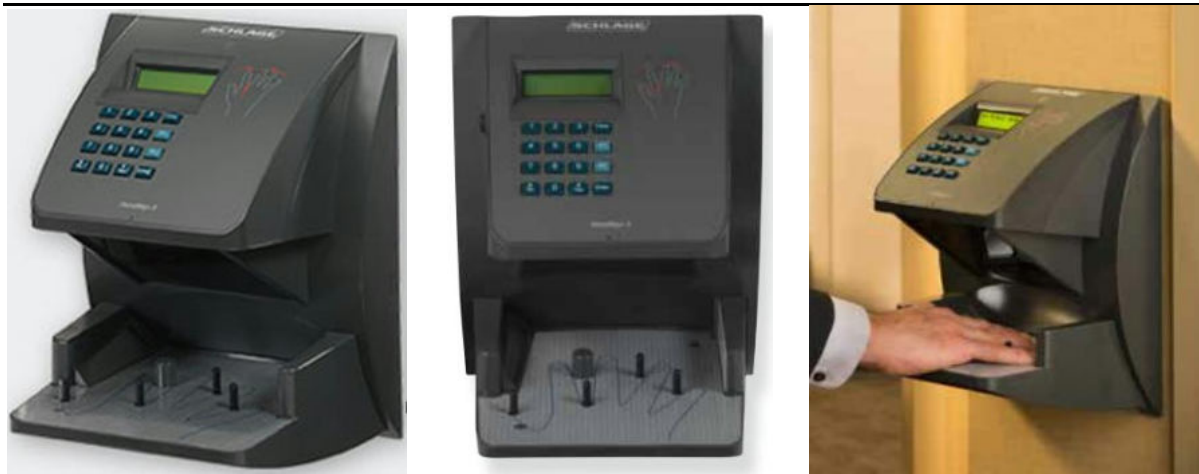


Obrázek (3).1 **Vlevo**-Mechanický systém identifikace rozměrů lidské ruky US Patent 3576537A viz (Ernst 1971), **Vpravo**-Elektrooptický systém pro identifikaci rozměrů lidské ruky US Patent 3648240A.

Pátého prosince 1968 podal Ernst H. Richard, pracovník fi. Polaroid corp. patentovou přihlášku na technické zařízení, které umožňovalo identifikaci osoby na základě antropometrické metody měření délky a šířky lidské ruky (Ernst 1971). Jednalo se o komerčně dostupné zařízení, které sloužilo jako doplňkový systém pro identifikační karty osob (ID karta - obdoba občanského průkazu). Odpovídající rozměry ruky byly vždy zaznamenány na korespondující ID kartě. Patent US3576537A byl udělen v roce 1971 – viz Obr. (3).1A. Ernst H. Richard se díky tomu stal prvním, kdo v novodobé historii úspěšně aplikovat již dříve využívané metody antropometrické identifikace osob – podle rozměrů lidské ruky. O pouhý jeden rok později byl publikován US Patent 3648240A skupinou výzkumníků: Warren H. Fioretti, Anthony J. Giordano a Ian H. Jacoby z firmy Identification corp., který Richardovy metody výrazně zdokonalil (Jacoby 1970). Navržený elektrooptický systém měřil také délku jednotlivých prstů. V následujících letech bylo publikováno několik desítek patentů, ve kterých byly navrženy různé

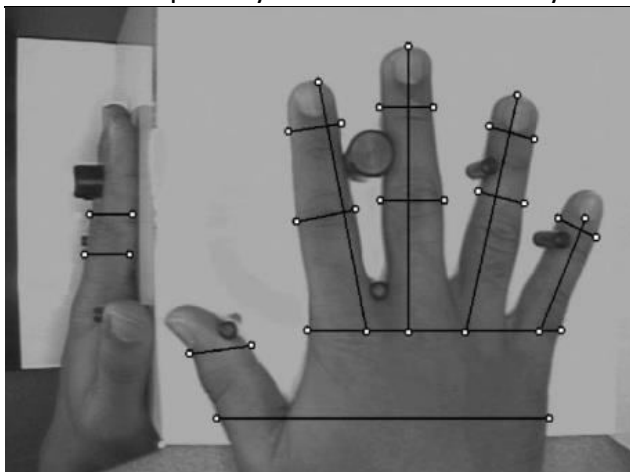
metody antropometrické identifikace. Tyto metody byly založeny nejen na detekci rozměrů ruky, ale také na otisku prstů nebo otisku celé dlaně.

Biometrický scanner fi. Allegion



Obrázek (3).2 Biometrický scanner fi. Allegion (Allegion 2020). Verifikace podle 2D geometrie lidské ruky. Scanner využívá prakticky pouze geometrii čtyřech prstů ruky.

Umístění pravé ruky ve scanneru se středícími kolíčky



Obrázek (3).3 Umístění pravé ruky v reálném biometrickém scanneru. Jsou dobře patrné středící kolíčky (Jain & Duta 1999).

V červnu 1985 podal patentovou přihlášku metody na rozpoznávání 3D kontury lidské ruky David P. Sidlauskas z firmy Recognition Systems Inc. (Allegion 2020; Ibia 2020). Patent byl přijat a publikován v dubnu 1988 US Patent 4736203 - „3D hand profile identification apparatus“. O rok později byl tento první tří-dimenzionální identifikační biometrický scanner komerčně dostupný a v roce 1996 byly 3D scannery dokonce použity pro identifikaci osob na Olympijských hrách v Atlantě. Metoda rozpoznávání geometrie ruky je již dnes standardizovaná podle normy ANSI/INCITS 396-2005 - Hand Geometry Format for Data Interchange. Odpovídající Evropská norma ISO/IEC 19794-10/2007 (Iso 2020) byla schválena v roce 2007. Metody na autentizaci osoby pomocí 2D nebo 3D geometrie ruky jsou však stále ve vývoji viz např. (Cortes et al. 2009; Ibia 2020). Norma ISO/IEC 19794-10/2007 popisuje pouze základní aspekty – viz také (Iso 2020; Incits 2020; Ibia 2020; Dhs 2020). Obecně se prozatím považuje biometrická metoda využívající geometrii ruky za méně spolehlivou v porovnání např. s testem DNA. Mnoho autorů proto prezentuje navržené metody využívající geometrii ruky jako metody autentizační neboli metody, které pouze ověřují deklarovanou identitu jedince. K precizní identifikaci je proto nutné použít další metodu např. identifikační kartu nebo heslo/pin.

3.2 Část druhá

V této části budou jen krátce shrnuty některé zajímavé práce, které se svou tematikou přímo vztahují k tématu korekce distorze obrazu a také identifikace osob s využitím kontury ruky. Detailnější přehled literatury je podán vždy v dané kapitole, kde se tematika řeší. Metody zde popisované, a především některé aspekty v nich, jsou později použity pro řešení úlohy identifikace osob a při kalibraci obrazu z biometrického scanneru.

Jain A.K., Duta N.

(Jain & Duta 1999) publikovali zajímavou studii, ve které řeší identifikaci osob s využitím kontury ruky. Autoři využívají poznatky z předchozích prací např. (Zunkel 1998; Jain et al. 1999). Obraz ruky je získán z biometrického scanneru, který je vybaven adjustážními kolíčky tzv. pegs. Obraz je převeden na šedo-tónovou reprezentaci. Je provedena korekce deformace prstů v místech, kde se nacházejí adjustážní kolíčky. Metodou adaptivního prahování a výpočtem kontury je získána kontura ruky. Postupně jsou extrahovány jednotlivé prsty z kontury ruky. Klasifikace probíhá na základě transformace popsané (Duta et al. 1999; Besl & McKay 1992) a tato transformace využívá kvadratické kritérium a tzv. střední chybu zarovnání (Mean Alignment Error - MAE). Klasifikátor poté využívá reálné číslo, na základě kterého, je rozhodnuto, zda identifikovaná osoba má záznam v databázi nebo nemá.

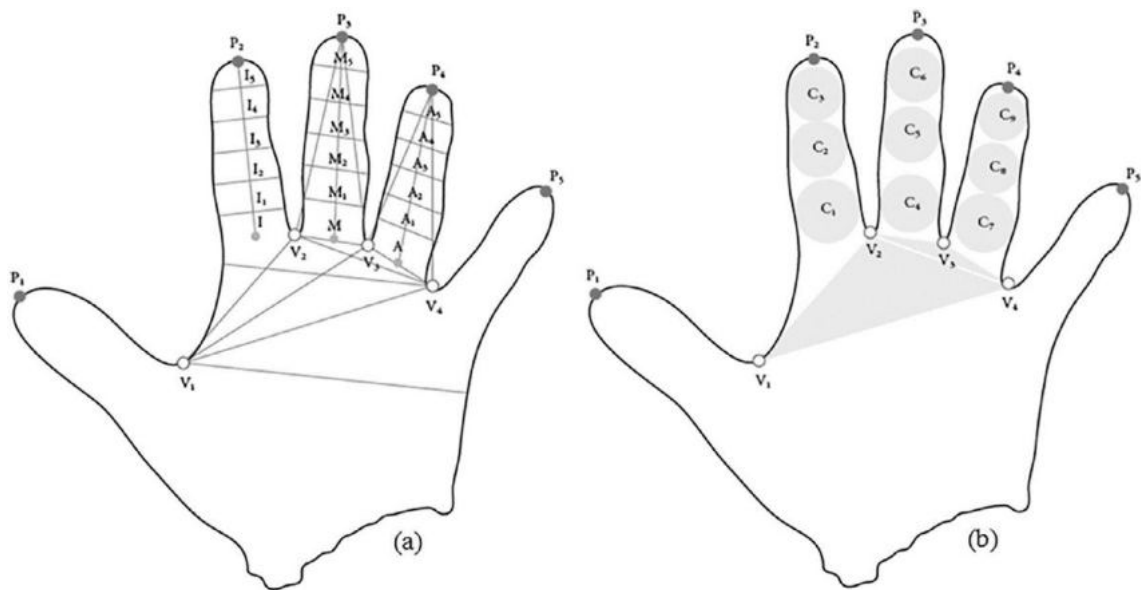
Yörük E., Konukoglu E., Sankur B., Darbon J.

(Yörük et al. 2006) navrhli metodu, která umožňuje s využitím modifikované Hausdorffovy vzdálenosti (Dubuisson & Jain 1994) měřit podobnost kontury ruky dvou osob. Pro klasifikaci je pak využit algoritmus Independent Component Analysis (ICA) (Bartlett et al. 1998), což je výpočetní metoda, která umožňuje oddělit užitečná data z multivariantních dat. Předpokládá se, že ony komponenty/subkomponenty v algoritmu ICA jsou vzájemně statisticky zcela nezávislé a ideálně negaussovské. Pro získání obrazů byl využit scanner HP Scanjet 5300c. Obrazy v rozlišení 383x526 pixelů. Dalším zpracováním se rozlišení upravilo na 200x200 pixelů. Autoři měli k dispozici databázi 1374 obrazů od 458 osob. 3 obrazy od každé osoby. Skenována byla pravá ruka. Věk osob byl 20-50 let a lidé různých národností. Extrakce kontury ruky je pak založena na segmentaci šedotónového obrazu. Pro nalezení kontury prstů je využit Radial Distance Diagram (RDD) a autory navržená unikátní metoda „pendula“. Dále je odhadnuta pozice kloubů prstů v dlani, které se nacházejí mezi záprstní kostí (splint bone) a prvním článkem každého prstu. Pozice je dána prodloužením osy každého prstu o 20 procent. V případě palce je kloub v místě zápěstí. Díky tomu je pak možné při procesu komparace měnit polohu každého prstu samostatně. Výsledná úspěšnost klasifikace se pohybovala v rozmezí 93-97 procent.

Barra S., Marsico M., Nappi M., Narducci F., Riccio D.

(Barra et al. 2019) publikovali práci založenou na vlastním dřívějším výzkumu (Barra et al. 2015, 2013, 2014, 2017). Využívali také poznatky z mnoha prací různých výzkumných týmů např. (Dai & Yuen 2003; Sharma et al. 2015). Cílem této studie bylo navrhnout algoritmus, který by umožnil identifikovat osobu na základě kontury ruky. Zvolili metodu identifikace pro mobilní zařízení, jako jsou např. mobilní telefony, tablety apod. Všechna tato zařízení mají k dispozici digitální fotoaparát nebo kameru s dostatečně vysokým rozlišením, jehož využití je pro daný účel ideální. Autoři nejprve testovali metody využívající geometrická primitiva a Euklidovskou metriku. Později práci rozšířili o algoritmus Lineární Diskriminanční Analýzy (Linear Discriminant Analysis) (LDA) (Faktorová Analýza) (Fisher 1936; Fukunaga 1990). Pro identifikaci osoby využívají sadu příznaků označených jako f_1, \dots, f_{57} a jejich různé kombinace, které popisují geometrii ruky, avšak kontura ruky samotná využita není – viz Obr. (3).4. Autoři sestavili poměrně rozsáhlou komparativní studii. Porovnávali různé typy metod založených čistě na kombinaci různých geometrických primitiv s Euklidovskou metrikou a algoritmem LDA.

Navrženým algoritmem využívající LDA dosáhli nejlepšího výsledku koeficientem EER=0.52% a dále pak hodnoty EER=3.5% pro algoritmus využívající standardizovanou Euklidovskou vzdálenost.



Obrázek (3).4 Kontura ruky a popis primitiv f_1, \dots, f_{57} . Převzato viz (Barra et al., 2019).

Sharma S., Dubey S.R., Singh S.K., Saxena R., Singh R.K.

(Sharma et al. 2015) publikovali metodu nazvanou multimodální biometrický systém, který pro identifikaci osob využívá konturu ruky a také další geometrická primitiva vypočítaná z kontury ruky jako např. vzdálenosti mezi významnými body na kontuře ruky a uvnitř kontury ruky a také úhly, které svírají přímky proložené významnými body. Autoři použili jednodimenzionální Waveletovu dekompozici na všechny extrahované prvky tj. vzdálenosti i úhly (orientaci). Vstupem do navrženého algoritmu je snímek z klasického kancelářského scanneru. Snímek je prahován a je získána kontura a centroid celé ruky včetně prstů a je také odhadnuto natočení ruky podle pozice prostředního prstu. V případě, že kontura není spojitá, což se stává v oblasti, kde se prsty napojují na dlaň, je kontura uměle dopočtena a opravena tak, aby byla spojitá. Dále je vypočten bod na zápěstí, který bude sloužit jako referenční bod pro další výpočty. Z kontury ruky je vytvořen Radial Distance Diagram (RDD) a z něj jsou pak extrahovány důležité body na špičkách prstů a také v tzv. údolí mezi prsty. V tomto okamžiku je k dispozici kontura a důležité body jako vstupní data pro klasifikátor.

Bera A., Bhattacharjee D.

(Bera & Bhattacharjee 2017) navrhli metodu, která umožňuje klasifikaci kontury ruky s využitím "k-nearest neighbours" (kNN) klasifikátoru převzatého z (Luque-Baena et al. 2013), kde je využit také genetický algoritmus a dále je použit klasifikátor označovaný jako "Random Forest" převzatý z (Breiman 1996, 2001). Oba klasifikátory pracují se sadou příznaků na místo kontury ruky a vyhýbají se tak problémům, které plynou z využití klasifikátorů pracujících přímo s konturou ruky. Je proveden výpočet, jak pro celou ruku, která je složená z 5 prstů tak i pro ruku bez palce. Autoři využívají Bosphorus Database, 638 osob v databázi. RGB snímek ze scanneru je nejprve konvertován na šedo-tónovou reprezentaci a je odfiltrován šum s využitím nelineární filtrace. Poté je proveden převod na černobílý obraz pomocí prahování a dále s využitím Cannyho detektoru jsou vypočteny hrany - resp. kontura ruky. Kontura ruky je při předzpracování rozdělena na levostranný profil (konturu) a pravostranný profil. Díky tomu je možné extrahovat jen kontury prstů, které jsou pak použity při

klasifikaci. Z jednotlivých kontur prstů jsou extrahovány sady příznaků jako např. délka prstů, šířka prstů ve zvolených místech, úhly, které svírají přímky proložené důležitými body na kontuře ruky atd. Klasifikována je zvláště levá a pravá ruka a jsou provedeny experimenty s různým počtem osob v databázi a to 400, 458, 500, 600 a 638. Dosažená hodnota EER při využití jen 4 prstů je EER=6.6-8.0% pro pravou ruku a EER=6.2-8.25% pro levou ruku podle použitého počtu osob z databáze. Při zvyšujícím se počtu osob klesá i přesnost klasifikace.

Klonowski M., Plata M., Syga P

(Klonowski et al. 2018) navrhli metodu založenou na geometrii ruky. Autoři sestavili vlastní databázi 60 osob ve věku 20 až 50 let, 5-10 snímků od každé osoby, polovina testovaných osob jsou ženy. Využit je klasický stolní scanner s rozlišením 300dpi, 2500x3500 pixelů. Osoby si při snímkování nesundávali prsteny ani náramky apod. Navržená metoda se skládá z několika kroků. Nejprve se RGB snímek převede na YCrCb obrazovou reprezentaci. Poté je provedeno prahování a zjištěna konvexní obálka celé ruky a je vypočtena kontura ruky. Je provedena korekce obrazu v tom smyslu, že jsou odstraněny kontury v oblasti nehtů a kontura nehtu je nahrazena obloukem. Následuje výpočet důležitých bodů na kontuře ruky, jako jsou špičky prstů, délka prstů a šířka prstů. Celkem je extrahováno 62 příznaků jako je např. délka prstů, šířka prstů podél osy každého prstu, rozměry dlaně apod. Jako metriku pro porovnávání sad příznaků použili autoři absolutní hodnotu z rozdílů jednotlivých příznaků vynásobenou váhovými koeficienty. Autoři také zjišťovali, jaký vliv na výslednou přesnost mají sady příznaků u jednotlivých prstů při klasifikaci. Při použití sad příznaků je výsledná hodnota FAR závislá na počtu rukou, které byly zahrnuty do tzv. template - vzoru. Pro geometrické příznaky je tato hodnota v rozmezí FAR=0.79% pro template sestavený z jedné ruky a FAR=0.03% pro template sestavený ze 3 kontur rukou stejné osoby. Dosažená přesnost v celé databázi je FAR=0.0%, FRR=1.19%, EER=0.59% v případě využívání tzv. vzoru (template) složeného ze 3 vzorků od každé osoby.

LITERATURA

- Allegion. 2020 [online] Allegion. ©Allegion 2020 [cit. 8.4.2020]. Dostupné z: <https://us.allegion.com/en/home/products/categories/biometrics.html>
- Bakshe R.C, Patil A.M. 2014. Hand Geometry Techniques: A Review. *International Journal of Modern Communication Technologies & Research*. vol. 2, no. 11, p.7.
- Barra S., Casanova M., Marsico M., Riccio D. 2014. Babies: Biometric authentication of newborn identities by means of ear signatures. In: *IEEE Workshop on Biometric measurements and system for security and medical application (BIOMS)*. pp. 1-7.
- Barra S., Casanova A., Narducci F., Ricciardi S. 2015. Ubiquitous iris recognition by means of mobile devices. *Pattern recognition Letters*. vol. 57, pp. 66-73.
- Barra S., Marsico M., Galdi C., Harry W. 2013. FAME: face authentication for mobile encounter. In: *IEEE Workshop on biometric measurements and systems for security and medical applications*. BioMS.
- Barra S., Marsico M., Nappi M., Narducci F. 2017. MOHAB: Mobile Hand-based Biometric recognition. In: *International conference on green pervasive and cloud computing*. Springer. pp. 105-115.
- Barra S., Marsico M., Nappi M., Narducci F., Riccio D. 2019. A hand-based biometric system in visible mobile environments. *Information sciences*. vol. 479, pp. 472-485.
- Bera A, Bhattacharjee D. 2017. Human Identification Using Selected Features From Finger Geometric Profiles. *IEEE Transactions on system, man and cybernetics: Systems*. pp. 1-15.
- Besl P.J., McKay H.D. 1992. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vol. 14, no. 2, pp. 239-256.
- Breiman L. 1996. Bagging predictors. *Machine learning*. vol. 24, no. 2, pp. 123–140.
- Breiman L. 2001 Random forests. *Machine learning*. vol. 45, no. 1, pp. 5–32.
- Cortes J.M.R., Gil P.G., Perez G.S., Castro P.C. 2009. Shape-based hand recognition approach using the morphological pattern spectrum. *Journal of Electronic Imaging*. vol. 18, no. 1, pp. 0130121/1 - 0130121/6.
- Dai D.Q., Yuen P.C. 2003. Regularized discriminant analysis and its application to face recognition. *Pattern recognition*. vol. 36, no. 3, pp. 845-847.
- Dhs 2020 [online] [cit. 8.4.2020]. Dostupné z: <https://www.dhs.gov/biometrics>
- Dubuisson M.P., Jain A.K. 1994. A modified Hausdorff distance for object matching. In: *12th International Conference on Pattern Recognition*. pp. 566-568.
- Duta N., Jain A.K., Jolly M.P. 1999. Learning 2D Shape Models, In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR99*, p. 7.
- Duta N. 2009. A survey of biometric technology based on hand shape. *Pattern recognition*. vol. 42, pp 2797-2806.
- Ernst R.H. Hand ID system. USA. Patent US3576537A, 27.4.1971.
- Fisher R. 1936. The use of multiple measures in taxonomic problems. *Annals of Eugenics*. vol. 7, no. 2, pp. 179–188.

Fukunaga K. 1990. *Introduction to Statistical Pattern Recognition*. 2nd ed., New York Academic, p. 591, ISBN 0-12-269851-7.

Ibia. 2020 [online] [cit. 8.4.2020]. Dostupné z: <https://www.w3.org/2008/08/siv/Slides/Daon/CBEFF-Tilton-2009-short.pdf>

Incits. 2020. [online] [cit. 8.4.2020]. Dostupné z: https://standards.incits.org/apps/group_public/download.php/24527/National_Standards_Published_as_of_09_08_2010.pdf

Iso. 2020 [online] [cit. 8.4.2020]. Dostupné z: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=43638

Jacoby I.H., Giordano A.J., Fioretti W.H. Personnel identification apparatus. USA, Patent US3648240A, 15.1.1970.

Jain A.K., Duta N. 1999. Deformable matching of hand shapes for user verification. In: *Proceedings 1999 International Conference on Image Processing*, [cit. 8.4.2020] DOI: 10.1109/ICIP.1999.823019.

Jain A.K., Ross A., Pankanti S. 1999. A prototype hand geometry-based verification system. In: *2nd international conference on audio and video based biometric person authentication*. Washington D.C., pp.166-171.

Klonowski M., Plata M., Syga P. 2018. User authorization based on hand geometry without special equipment. *Pattern Recognition*. vol. 73, pp. 189–201.

Luque-Baena R.M., Elizondo D., López-Rubio E., Palomo E.J., Watson T. 2013. Assessment of geometric features for individual identification and verification in biometric hand systems. *Expert Systems with Applications*. vol. 40, no. 9, pp. 3580–3594.

Pavešić N., Ribarič S., Ribarič D. 2004. Personal authentication using hand-geometry and palm-print features – the state of the art.

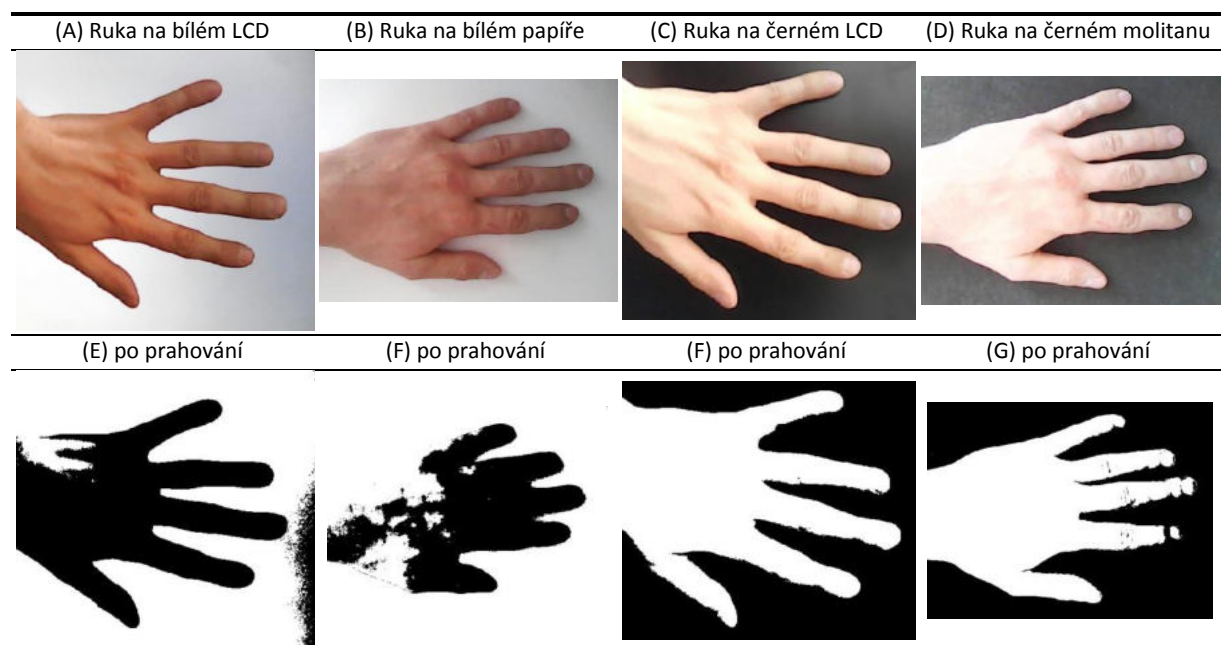
Ross A., Jain A.K. 2007. Human Recognition Using Biometrics: An Overview. *Annals of Telecommunications*, vol. 62, no. 1-2, pp. 11-35.

Sharma S., Dubey S.R., Singh S.K., Saxena R., Singh R.K. 2015. Identity verification using shape and geometry of human hands. *Expert Systems with Applications*. vol. 42, no. 2, pp. 821-832.

Zunkel R. 1998. Hand geometry-based verification. In: *Biometrics: Personal identification in networked society*, Jain, A.K., Bolle, Ruud M., Pankanti, Sharath (Eds.), Kluwer Academic, pp. 87-101, ISBN 978-0-387-28539-9.

4 Konstrukce biometrického scanneru

Biometrická identifikace osob má dlouhou historii. První metody biometrické identifikace jsou známe již z dob vlády faraonů. Tehdy se využíval např. otisk ruky. Novodobá biometrie využívá nejnovější poznatky z mnoha vědních oborů a široce se opírá o počítačové technologie. Biometrika rozhodně nepatří mezi jednoduché vědní obory. Je to dáno tím, že rozmanitost variant, byť jen jediného biometrického ukazatele je v lidské populaci značná. Moderní počítače obvykle pracují s nějakou jednoznačnou sekvencí bytů nebo bitů. Takovouto informaci lze dobře porovnávat a také klasifikovat. U biometrických ukazatelů však není možné hovořit o nějaké jednoznačnosti. Prvním krokem identifikace bývá obvykle scanování a pak následuje předzpracování a teprve potom klasifikace. Úkolem předzpracování je provést nezbytné korekce a do jisté míry i normalizaci získaného obrazu. Krok předzpracování má za úkol také provádět opravy poškozených vstupních obrazů, pokud je to žádoucí a pokud je to vůbec možné. Nejdůležitějším krokem se tak stává korektní získání digitálního obrazu daného biometrického ukazatele v co nejlepším stavu. V případě identifikace osob, která využívá konturu lidské ruky, je to obvykle barevný obrázek ruky. Aby byl takový obraz co možná nejkvalitnější a bez parazitického šumu, jako je nežádoucí přesvětlení nebo stín, je vhodné sestavit zařízení, které usnadní získávání obrazů a zajistí, aby obrazy byly v co možná nejvyšší kvalitě.

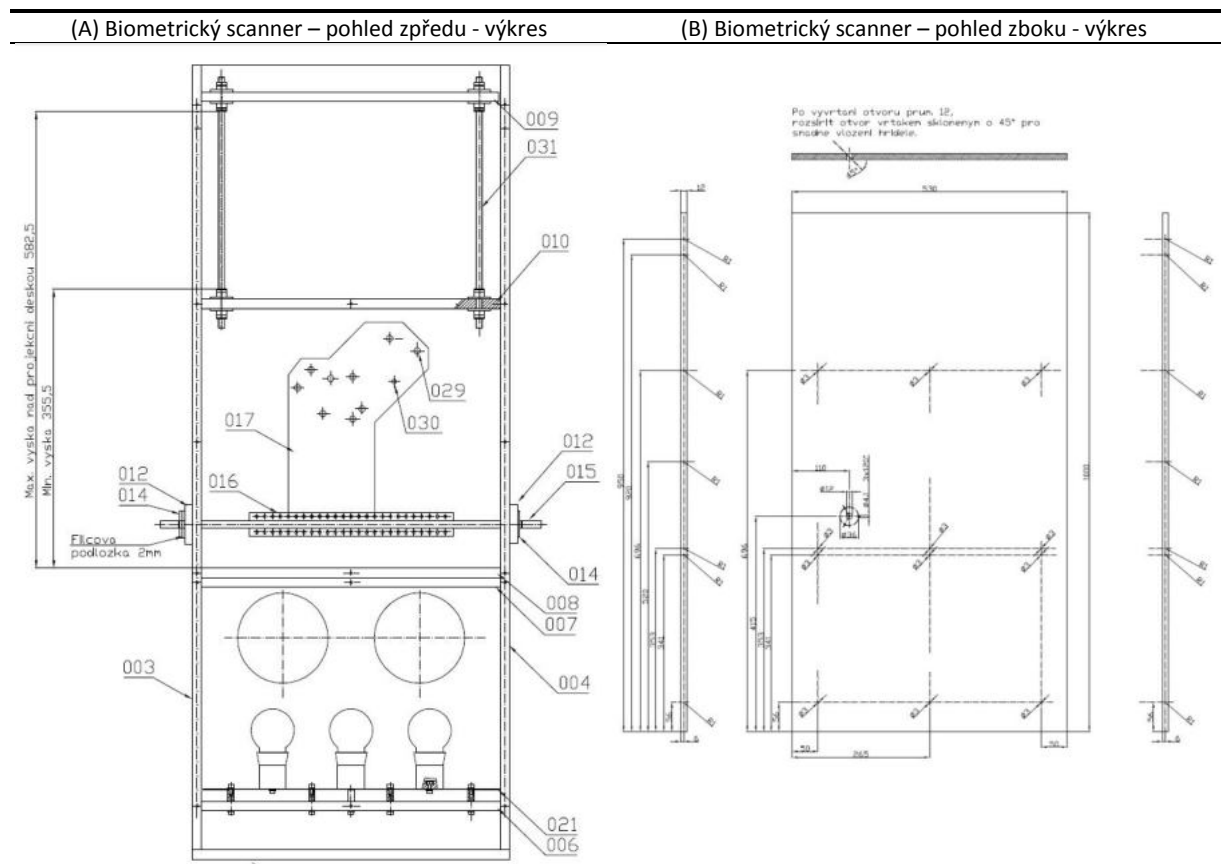


Obrázek (4).1 Některé z méně vhodných testovaných materiálů pro jednoduché prahování - kombinace osvětlení a podkladu.

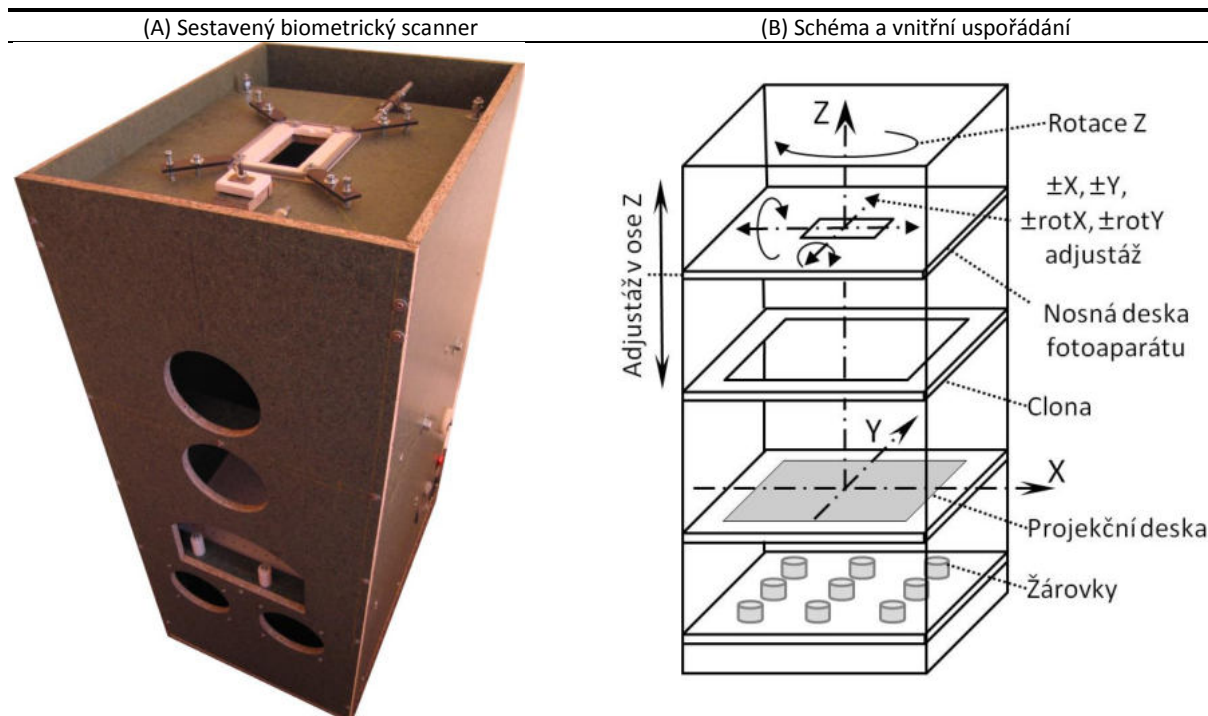
Cílem disertační práce je navržení biometrické metody identifikace lidské ruky. Bylo provedeno značné množství experimentů – viz např. Obr. (4).1, jejichž cílem bylo zjistit, jak co nejefektivněji a současně co možná nejjednodušším způsobem provádět snímání tvaru lidské ruky s využitím běžně dostupných fotografických zařízení bez ztráty přesnosti a pokud možno s co nejmenším množstvím optických vad a deformací výsledného obrazu. Důležitým kritériem bylo také to, aby předzpracování obrazu z biometrického scanneru bylo zatíženo minimální ztrátou užitečných dat. Byla zvolena metoda, kdy snímaná ruka je položena na průsvitné skleněné desce (bílá, matná, neprůhledná). Zespoda je vedeno osvětlení desky a nad deskou je umístěn fotoaparát. Aby bylo zajištěno, že do snímku nebude zasahovat parazitní osvětlení, byla projekční deska uzavřena v komoře. Takto uspořádaný systém umožňuje získat černobílý obraz lidské ruky přímo z barevného RGB snímku z fotoaparátu pouhým primitivním prahováním. Jedná se o rychlou a spolehlivou metodu s minimální ztrátou užitečných dat.

4.1 Precizní laboratorní biometrický scanner

Pro účely experimentů byl sestaven **precizní biometrický scanner** se stíněnou fotografickou komorou – viz. Obr. (4).2, (4).3, (4).4. Díky tomu je možné zajistit, aby do biometrického snímku nezasahovalo okolní parazitní světlo – viz. Obr. (4).3. Základní rozměry scanneru jsou (výška x hloubka x šířka): 102 x 56 x 41. Scanner je sestaven z vodoodpudivých (nenasákavých) dřevotřískových desek (typ Durelis) tloušťka 12mm. Váha cca 35Kg. Smontován je vruty 3x50mm. V případě potřeby je možné jej rozložit. Biometrický scanner umožňuje připevnit libovolný snímací prvek – např. klasický kompaktní fotoaparát, videokameru, web kameru apod. Min. ohnisková vzdálenost nad projekční deskou je 355.5mm, maximální potom 582.5mm. Rozdíl je 227mm. To dává značnou variabilitu při výběru snímacího elementu. Scanner má v dolní části umístěnou soustavu žárovek, která zajišťuje kvalitní, konstantní, a především rovnoměrné osvětlení spodní části projekční desky. Projekční deska je tvořena dvěma skleněnými deskami. Mezi nimi je umístěn difuzor a clona, která zajistí snížení intenzity světelného toku na požadovanou úroveň. Aby bylo možné zajistit dostatečně rovnoměrné osvětlení, byly použity klasické vláknové žárovky v počtu 3x3 – viz. Obr. (4).4.



Obrázek (4).2 Výkres navrženého biometrického scanneru – sestava, pohled zepředu a z boku. Není v měřítku.



Obrázek (4).3 **Vlevo**-Sestavený biometrický scanner, **Vpravo**-schéma bio-scanneru a vnitřní uspořádání.

Žárovky v rozích jsou 40Wattů, ostatní pak 25Wattů. Nosná deska fotoaparátu – viz. Obr. (4).2, (4).3 a (4).4, umožňuje nastavení fotoaparátu v osách X, Y, Z a rotaci okolo os X a Y viz. Obr. (4).3 vpravo. V dolní zadní části biometrického scanneru jsou pak umístěny 2 odtahové ventilátory, které zajišťují dostatečné proudění vzduchu pod projekční deskou. Výkon je 224 metrů krychlových za hodinu pro oba součtově, což je dostačující na chlazení při teplotě v místnosti do +35°C. Na Obr. (4).2. vlevo je zakreslený adjustážní přípravek (pozice 017 na výkrese), který vystředí snímanou ruku do požadované pozice. Je sestaven tak, že může být sklopen a nezasahuje do snímacího prostoru kamery.

Nosná hřídel adjustážního přípravku má na obou stranách bioscanneru ovládací páku. Bioscanner obsahuje 2 snímací kamery. První je náhledová – viz. Obr. (4).4B, s velmi nízkým rozlišením a získaný obraz je viditelný na monitoru ovládacího počítače. Druhá kamera je pak snímací a je umístěna na nosné desce – viz. Obr. (4).3 vlevo nahoře. Uspořádání se dvěma kamerami umožňuje pohodlnou obsluhu bez nutnosti kontrolovat pozici ruky přímo v projekční komoře scanneru, kam je špatně vidět. Předpokládané rozlišení fotoaparátu je alespoň 2592x1944x24bit(RGB) pixelů ve formátu BMP/RAW – nekomprimovaný, a fotoaparát musí mít zabudovanou podporu knihovny MS WIA. Výhodou použití fotoaparátu místo např. jednoduché web kamery jako hlavního snímacího elementu je to, že při každém snímku provádí fotoaparát přesné zaostření a mnoho dalších automatických korekcí. Díky tomu je možné dosáhnout velmi kvalitní obraz.

4.2 Další možnosti využití scanneru

Precizní biometrický scanner je primárně určen pro snímání lidské ruky dospělého člověka. Ruka se vkládá na projekční desku, má roztažené prsty a k přesnému ustavení slouží adjustážní přípravek. Předpokládá se levá ruka v případě použití adjustážního přípravku. Je samozřejmě možné snímat levou i pravou ruku dospělého i dítěte. Kovový adjustážní přípravek je vyměnitelný. Lze jej zcela odstranit a nahradit klasickou průsvitnou deskou (plexisklo, tloušťka cca 10 mm) se středícími kolíčky. Stejně tak je možné umístit i adjustážní přípravek na snímání kontury pěsti levé nebo pravé ruky – tj. jeden velký středící kolík – viz Obr. (4).5A. Biometrický scanner s výše popsaným uspořádáním byl sestaven proto,

aby bylo možné získat dostatečně kvalitní „normativní“ databázi vzorků. Tato databáze pak slouží pro identifikaci nebo verifikaci jedince.



Obrázek (4.4) **Vlevo nahoře**-adjustážní přípravek pro přesné umístění ruky. **Vpravo nahoře**-nosná deska fotoaparátu. **Vlevo dole**-osvětlovací soustava tvořená 9-ti žárovkami (40W a 25W). **Vpravo dole**-snímací komora.

(A) Ruka držící kolík



(B) Ruka na projekční desce scanneru – sepnuté prsty a oddělený palec



(C) Ruka na projekční desce scanneru – sepnuté prsty včetně palce



(D) Ruka na projekční desce scanneru – sepnuté prsty, detail na středící lišty mezi prsteníčkem a ukazováčkem



Obrázek (4).5 **Vlevo nahoře**-Ruka držící kolík na projekční desce. **Vpravo nahoře**-Ruka se sepnutými prsty na projekční desce s vykloněným palcem. **Vlevo dole**-Ruka se sepnutými prsty na středící liště mezi prsteníčkem a ukazováčkem. **Vpravo dole**-Detail na středící lišty na projekční desce s rukou se sepnutými prsty.

5 Evoluční algoritmy pro biometrickou identifikaci

Evoluční algoritmy (EA) tvoří podstatnou část celé disertační práce. Téměř všechny předkládané metody využívají některé druhy populačních evolučních algoritmů. Pro řešení problému korekce distorze je zvolen derivát diferenciální evoluce označovaný jako jDE společně s algoritmem označovaným jako evoluční strategie, který také spadá do oblasti EA. Pro řešení problému biometrické identifikace osob s využitím EA byl pak zvolen opět moderní a pokročilý derivát algoritmu diferenciální evoluce označovaný jako EPSDE. Evoluční algoritmy jsou použity vždy jako velmi výkonný optimalizátor, jehož úkolem je najít optimální řešení dané n-dimenzionální úlohy. Toto řešení spočívá buď v minimalizaci nějaké účelové funkce nebo naopak nalezení nějaké konkrétní hodnoty, pokud vůbec existuje, nebo alespoň přiblížení se k ní. Často je tedy dostačující, aby zvolený evoluční optimalizátor dosáhl na určitou limitní hodnotu.

Oblast EA má základy v Darwinově teorii přirozeného výběru. Počátky vývoje spadají do 40. a 50. let minulého století. První dohledatelné zmínky jsou datované okolo roku 1954 z prací Nils Aall Barricelliho (Barricelli 1954, 1957, 1961, 1963). Nelze samozřejmě opominout i dřívější práce např. (Metropolis & Ulam 1949), kde je využit algoritmus MonteCarlo (MC), ale tento algoritmus není v jádru typickým evolučním algoritmem. V jeho jádru nelze nalézt operace typické pro EA jako jsou např. operace křížení nebo mutace. Původním cílem návrhu evolučních algoritmů bylo simulovat podmínky přirozeného výběru za časově přijatelných okolností a dostat se tak k optimálnímu řešení výrazně dříve než při použití jakékoliv jiné metody. K mohutnému rozvoji EA přispěl i vývoj v oblasti počítačových věd v 80. a 90. letech 20. století a široké rozšíření cenově dostupné výpočetní a senzorové techniky. Komplexní přehled o EA je možné nalézt např. na (Pardalos 2020) a jeho publikace. Postupným vývojem se oblast EA rozdělila na tři hlavní proudy:

- Genetické Algoritmy (GA/SGA) popsáné J. Hollandem (Holland 1962; Goldberg 1987, 1989), a také Genetické Programování (GP) popsáné J. Kozou (Koza 1994, 1996)
- Evoluční Programování (EP), popsáné L. Fogelem (Fogel 1966; Fogel 1999; Fogel & Chellapilla 1998)
- Evoluční Strategie (ES) popsáná autory I. Rechenberg, H.P. Schwefel a P. Bienert (Rechenberg 1973, 1994; Schwefel 1995; Beyer & Schwefel 2002)

Všechny tři větve využívají základů položených v 50. letech. Jsou zde však výrazné rozdíly ve způsobu, jakým je nakládáno např. s reprezentací jedinců v populaci. GA si obecně ponechávají možnost kódování jedinců formou binárního řetězce, ES využívá reálná čísla a EP pak stromové struktury, tzv. S-výrazy. Na základě Hollandova GA bylo v průběhu 90. let navrženo mnoho tzv. metaheuristických iterativních optimalizátorů (Glover 1986; Glover & Kochenberger 2003) - algoritmů, které podobně jako ES využívají reálná čísla pro kódování jedinců populace. Tyto stochastické optimalizátory jako např. Simulované žíhání (Simulated Annealing) (SA) (Kirkpatrick et al. 1983), Diferenciální Evoluce (Differential Evolution) (DE), Mravenčí Kolonie (Ant Colony Optimization) (ACO) (Dorigo 1992), Rojení Částic (Particle Swarm Optimization) (PSO) (Kennedy & Eberhart 1995), Tabu Search (TS) (Glover 1986), Memetické Algoritmy (MA) (Chen et al. 2011) se dnes již běžně využívají nejen v oblastech výzkumu. Pomáhají řešit i reálné problémy každodenního života v oblastech ekonomie, lékařství, zemědělství a v mnoha dalších průmyslových odvětvích. Do kategorie stochastických optimalizátorů jistým způsobem spadá i algoritmus MonteCarlo (MCL), i když je prakticky nejstarší z nich a nemá žádné prvky převzaté z přírodního evolučního vývoje.

Při řešení mnoha reálných úloh není nezbytně nutné najít to 'nejlepší' globální optimum. Přibližné řešení dané úlohy je mnohdy postačující. Z výše jmenovaných optimalizátorů je MC nejjednodušší na

algoritmizaci, ale neposkytuje dostatečný výkon při řešení obtížných nízko-dimenzionálních i vysoko-dimenzionálních problémů. SGA nebo DE je naopak nepoměrně složitější na implementaci a je také nesrovnatelně výkonnější. Nevýhodou je, že korektní nastavení pracovních parametrů je nesmírně důležité, a i malá změna může vést k úplnému selhání optimalizátoru. SGA algoritmus byl mnohokrát diskutován v odborné literatuře a bylo navrženo nespočet různých variant, více či méně robustních oproti původnímu SGA. Totéž platí o algoritmu DE. Algoritmus Diferenciální Evoluce je z pohledu složitosti nastavení pracovních parametrů identický jako SGA.

Díky rozsáhlému výzkumu v 70. a 80. letech, který byl založený na pracích (Goldberg 1987; Holland 1975; Rechenberg 1973; Schwefel 1995; Rechenberg 1994) je dnes možné využívat pokročilé stochastické optimalizátory při řešení úloh v mnoha vědních oborech. Obecný přehled lze najít např. v (Pardalos 2020) a doplňující informace k použitým evolučním algoritmům, které budou popsány v následujících kapitolách pak v (Barrera & Coello 2009; Brest et al. 2010; Igel et al. 2006; Jastrebski & Arnold 2006; Kennedy & Eberhart 1995; Montes et al. 2006; Pošík & Huyer 2012; Pošík et al. 2012; Pošík & Kubalík 2012; Urfalioglu 2004; Wang et al. 2011; Whitley et al. 1996; Wolpert & McReady 1995).

Kromě EA, které budou popisovány v následujících odstavcích této kapitoly, jsou v disertační práci využity další různé druhy EA, kterým je však věnována jen malá pozornost a jsou použity jen pro účely porovnání, a proto nebudou detailně popisovány v této kapitole. V drtivé většině případů se jedná o EA, které nepodávají dobré výsledky a na řešení diskutovaných úloh se nehodí. Je proto proveden jen krátký popis vždy v příslušné kapitole, kde slouží výhradně pro účely porovnání/demonstraci výkonnosti zvolených EA. Svět EA je opravdu velmi rozsáhlý a existují tisíce různých derivátů původních EA z 50. - 90. let a některé z nich jsou velmi náročné jak na algoritmizaci, tak i na popis.

Použití EA se může zpočátku zdát svízelné a je pravda, že rozhodujícím prvkem při jejich nasazení jsou rozsáhlé a dlouholeté tzv. expertní znalosti a samozřejmě dostatek volného času a relativně kvalitní technické vybavení na testování. Velmi důležitý je také vhodný software jako je např. Matlab, Maple, Wolfram Mathematica, geatbx.com, HeuristicLab a stovky dalších nebo jen běžně dostupný MS-VS-C#/C++ nebo jiný vhodný programovací jazyk. Při volbě vhodného EA pro řešení nějaké praktické úlohy jsou jako prvním vodítkem vždy tzv. benchmarkové funkce. Ty samozřejmě nezohledňují specifický reálný stav. V žádném případě nelze očekávat, že např. SGA, který se s oblibou vyučuje na školách poskytne obecný a ideální nástroj pro všechny možné typy úloh. Spíše naopak. V mnoha případech poskytne jen velké zklamání. Dalším krokem je tedy testování desítek/stovek různých EA pro vhodnost a výkonnost a hledání jejich vhodných pracovních parametrů. Tento krok je značně časově náročný. Nalezení vhodného EA, který efektivně, rychle a spolehlivě optimalizuje danou úlohu, se proto může občas zdát spíše jako zázrak. EA jsou obecně velmi časově a výpočetně náročné a cesta jejich praktického využití vede primárně přes reprogramovatelné hardwarové akcelerátory jako např. grafické karty Nvidia a AMD, karty Xilinx Alveo, Intel XeonPhi, multiprocesorové systémy Cerebras.net a podobná zařízení včetně superpočítačů.

5.1 Diferenciální evoluce a algoritmus jDE

V roce 1997 byl v článku (Storn & Price 1997) presentován algoritmus Diferenciální evoluce (DE), který spadá do širší skupiny algoritmů označovaných jako Evoluční Algoritmy (EA). Díky své výkonnosti si velmi rychle získal oblibu mnoha autorů. DE má pouze tři základní pracovní parametry označované jako F, P_{cr} a odchylkový vektor. Výhodou DE je to, že je rychlá a je schopna řešit i multimodální problémy velmi efektivně. V průběhu prací s algoritmem DE se v pozdějších letech bohužel jednoznačně prokázalo, že přes svoji výkonnost je DE značně závislá na korektním nastavení pracovních parametrů. Tím se DE stala výrazně složitější, než se původně myslelo viz (Eiben et al. 1999; Gampeler et al. 2002; Liu & Lampinen 2002; Smit & Eiben 2009; Smit & Eiben 2010a,b; Das & Suganthan 2010). Hledaly se proto nové možnosti. Původní algoritmus DE byl mnohokrát přepracován a rozšířen tak, aby byl schopen řešit různé typy úloh, aby se dále zvýšila jeho výkonnost, a především aby se eliminoval problém s pracným nastavením pracovních parametrů, které musí být velmi často měněny v průběhu výpočtu, jinak se může také stát, že optimalizátor selže. Mezi nejznámější modifikace patří např. algoritmy jDE (Brest et al. 2006), SaDE (Qin & Suganthan 2005), FADE (Liu & Lampinen 2005), SPDE (Abbas 2002), DESAP (Teo 2005), JADE (Zhang & Sanderson 2009), DERSF+DETVSF (Das et al. 2005). Všechny si zachovávají původní vlastnosti DE a výrazně je rozšiřují. Obecně se DE dělí na DE s pevným nastavením parametrů (Storn & Price 1997) podle modelu EA převzatých z (Goldberg 1987; Holland 1992), adaptivní DE např. (Noman et al. 2011) a self-adaptivní např. SPDE (Teo 2005) nebo i self-adaptive-jDE (Brest et al. 2006; Brest & Maučec 2011). Komplexní přehled o EA je možné nalézt např. (Floudas & Pardalos 2009; Pardalos 2020).

Algoritmus původního optimalizátoru DE podle (Storn & Price 1997; Price et al. 1999) je uveden v Algoritmu (5.1).1.

Algoritmus (5.1).1 Diferenciální evoluce (DE)

```

1  Nastav  $N, F, P_{cr}, \bar{V}_p, T$ ; Dáno  $D_{im}, D_{om}$ 
2  Vytvoř populaci jedinců  $P_{op} = (X_1, \dots, X_N)$ ,  $X_i = [x_j | j \in \langle 0, D_{im} \rangle]$ 
3  for  $i = 0$ ;  $i < N$ ;  $i=i+1$ 
4    for  $j = 0$ ;  $j < D_{im}$ ;  $j=j+1$ 
5      Inicializuj  $x_j$  náhodně v  $D_{om}$ 
6    endfor
7  endfor
8  Ohodnoť  $P_{op}$ , Najdi nelepššího jedince  $X_{best}$ 
9  while (  $T$  není splněna )
10   for  $i = 0$ ;  $i < N$ ;  $i=i+1$ 
11     Vyber náhodně  $\{X_{r1}, X_{r2}, \dots, X_{rk}\}, rj \neq i, X_{rj} \in P_{op}$ 
12     Vypočítej trial vektor  $X_t$  s využitím  $\bar{V}_p, X_i, X_{rj}, X_{best}$ 
13     Ohodnoť  $X_i$  a  $X_t$  - vypočítej fitness
14      $X_i = \text{nejlepší}_z(X_i, X_t)$ 
15   endfor
16   Najdi nejlepší individuum  $X_{best}$  v  $P_{op}$ 
17 endwhile
18  $X_{best}$  reprezentuje nejlepší nalezené řešení.
```

P_{op} -populace individuí; N -počet individuí v populaci; F, P_{cr} -pracovní parametry; \bar{V}_p , - odchylkový vektor; D_{im} -počet dimenzí každého individua; D_{om} -domain pro každou jednotlivou dimenzi D_{im} ; T -ukončovací podmínka; X_t -chromozom každého jedince neboli jedno možné řešení dané úlohy; platí, že $X_{r1} \neq X_{r2} \neq \dots \neq X_{rk} \neq X_t$, počet vektorů X_{rk} je dán $\bar{V}_p \cdot D_{im}$, D_{om} -je dáno charakterem úlohy.

Podobně jako mnoho jiných evolučních algoritmů, DE pracuje s populací P_{op} jedinců $X_i | i \in \langle 0, N \rangle$. Počet jedinců může, ale nemusí být konstantní po dobu výpočtu. Každý jedinec X_i reprezentuje jedno

možné řešení optimalizované úlohy. Je sestaven z chromozomu, který představuje množinou vstupních proměnných (dimenzí) $\mathbf{X}_i = [x_j | j \in \langle 0, D_{im} \rangle]$. Využívají se reálné kódovaná čísla. Prvním krokem je inicializace populace \mathbf{P}_{op} . Jedinci jsou umístěni na náhodné pozice v celém prostoru možných řešení $\mathcal{H} = D_{om}^1 \times D_{om}^2 \times \dots \times D_{om}^k | k \in \mathbb{N}$. D_{om} je tzv. domain pro k -tou dimenzi, $D_{om} | D_{om} \in \langle min, max \rangle$, každá dimenze může mít jiné hodnoty min, max . Druhý krok je ohodnocení všech individuí a nalezení individua s nejlepším fitness \mathbf{X}_{best} . Následuje cyklus složený z tzv. generací G , ve kterých populace konverguje k určitému řešení úlohy – viz ř. 9-17 Alg. (5.1).1. V každé generaci G je pro každého jedince \mathbf{X}_i z populace \mathbf{P}_{op} vytvořen tzv. trial vector \mathbf{X}_t operacemi mutace a křížení. \mathbf{X}_t má stejné složení chromosomu jako jedinec \mathbf{X}_i . Zapišme několik nejčastěji používaných vzorců neboli tzv. odchylkových vektorů (perturbation vectors) $\overline{\mathbf{V}}_p$ určených pro operaci mutace - viz (Price et al. 1999):

| Tabulka (5.1).1 Odchylkové vektory (perturbation vectors) | |
|---|---|
| $\overline{\mathbf{V}}_{p1}$ | DE/rand/1: $\mathbf{X}_{mut} = \mathbf{X}_{r1} + F(\mathbf{X}_{r2} - \mathbf{X}_{r3})$ |
| $\overline{\mathbf{V}}_{p2}$ | DE/best/1: $\mathbf{X}_{mut} = \mathbf{X}_{best} + F(\mathbf{X}_{r1} - \mathbf{X}_{r2})$ |
| $\overline{\mathbf{V}}_{p3}$ | DE/rand/2: $\mathbf{X}_{mut} = \mathbf{X}_{r1} + F(\mathbf{X}_{r2} - \mathbf{X}_{r3}) + F(\mathbf{X}_{r4} - \mathbf{X}_{r5})$ |
| $\overline{\mathbf{V}}_{p4}$ | DE/best/2: $\mathbf{X}_{mut} = \mathbf{X}_{best} + F(\mathbf{X}_{r1} - \mathbf{X}_{r2}) + F(\mathbf{X}_{r3} - \mathbf{X}_{r4})$ |
| $\overline{\mathbf{V}}_{p5}$ | DE/rand-to-best/1: $\mathbf{X}_{mut} = \mathbf{X}_i + F(\mathbf{X}_{best} - \mathbf{X}_i) + F(\mathbf{X}_{r1} - \mathbf{X}_{r2})$ |

\mathbf{X}_{mut} je tzv. mutant, \mathbf{X}_{best} je prozatím nejlepší jedinec v \mathbf{P}_{op} , \mathbf{X}_{rj} jsou náhodně zvolení jedinci z populace \mathbf{P}_{op} pro které platí, že $\mathbf{X}_{r1} \neq \mathbf{X}_{r2} \neq \dots \neq \mathbf{X}_{rk} \neq \mathbf{X}_i$. Koeficient F je konstantní po celou dobu výpočtu a je stejný pro všechny jedince. Další operací je křížení. Existují dva druhy označované jako *bin* a *exp* viz (Zaharie 2007). Uvedme častěji využívanou metodu *bin*, která je současně využita i v algoritmu pro odstranění distorze obrazu:

$$\mathbf{X}_t = \begin{cases} \mathbf{X}_{mut} & \text{if } rand(0,1) < P_{cr} \vee j = jrand \\ \mathbf{X}_i & \text{jinak} \end{cases}, j \in \langle 0, D_{im} \rangle \quad (5.1.1)$$

$jrand$ je náhodné číslo, uniformní rozložení v rozsahu $\langle 0, D_{im} \rangle$. P_{cr} je pracovní parameter DE a platí, že $P_{cr} \in \langle 0.0; 1.0 \rangle$, je to tzv. pravděpodobnost křížení. Pokud \mathbf{X}_t překročí hranice dané hodnotami D_{om} , je možné jej vygenerovat zcela náhodně kdekoli v prostoru možných řešení \mathcal{H} nebo na jeho okraji ve směru od středu D_{om} k jeho původní pozici pro j -tou dimenzi. Poslední operací je selekce:

$$\mathbf{X}_i^{(G+1)} = \begin{cases} \mathbf{X}_t & \text{if } fitness(\mathbf{X}_t) \text{ je lepší než } fitness(\mathbf{X}_i^{(G)}) \\ \mathbf{X}_i^{(G)} & \text{jinak} \end{cases} \quad (5.1.2)$$

při které je rozhodnuto, zda zkušební (trial) vektor \mathbf{X}_t je lepší než jedinec \mathbf{X}_i v i -té generaci. Pokud ano, je \mathbf{X}_i nahrazen \mathbf{X}_t jinak je \mathbf{X}_i ponechán. Výpočet pak pokračuje generací $G + 1$. Další užitečné informace viz např. (Zaharie 2007; Das & Suganthan 2010; Jeyakumar & Shanmugavelayutham 2011; Neri & Tironnen 2010).

V předkládaném článku je použita modifikace DE, označená jako jDE (Brest et al. 2006). jDE je využit nejen jako sekundární, ale i jako primární optimalizátor (spojený do kaskády). Výpočet proto musí být rychlý a výsledek přesný. Musí také umožňovat volitelnou přesnost výpočtu. Všechny tyto požadavky jDE splňuje. jDE byl pro řešení dané úlohy zvolen čistě na základě rozsáhlých praktických experimentů. V naší úloze optimalizuje např. dvojdímní funkci, která je omezená, spojitá, neseperabilní, nelineární a dobře podmíněná. jDE používá modifikovaný chromozome definovaný jako:

$$\mathbf{X}_i = [x_j | j \in \langle 0, D_{im} \rangle, F, P_{cr}] \quad (5.1.3)$$

Ten obsahuje nejen hodnoty možných řešení x_j , ale i hodnoty F, P_{cr} . Každý jedinec má tedy svoje vlastní pracovní parametry. Viz Algoritmus (5.1).2, řádek 2. Na řádku 11 je další modifikace. Před vytvořením trial vektoru \mathbf{X}_t jsou hodnoty F, P_{cr} každého jedince \mathbf{X}_i dočasně uschovány. Jsou vytvořeny nové hodnoty podle následujících vzorců – ř.12 Alg. (5.1).2:

$$F^{(G+1)} = \begin{cases} F_l + rand_1 \times F_u & \text{if } rand_2 < \tau_1 \\ F^{(G)} & \text{if } rand_2 \geq \tau_1 \end{cases} \quad (5.1.4)$$

$$P_{cr}^{(G+1)} = \begin{cases} rand_3 & \text{if } rand_4 < \tau_2 \\ P_{cr}^{(G)} & \text{if } rand_4 \geq \tau_2 \end{cases}, \quad (5.1.5)$$

kde τ_1, τ_2 jsou konstanty, podle kterých jsou nastavovány hodnoty parametrů F a P_{cr} . $rand_i | i \in \{1,2,3,4\}$, $rand_i = \langle 0,1 \rangle$ jsou náhodné proměnné, uniformní rozložení. F_l a F_u jsou volené konstanty, které ovlivňují hodnotu $F^{(G+1)}$ u každého jedince. Podobně jako u algoritmu DE i zde platí doporučení, že výsledná hodnota $F^{(G+1)}$ by měla být v rozsahu $\langle 0.0; \sim 2.0 \rangle$.

Algoritmus (5.1).2 jDE optimalizátor

```

1 Zvol  $N, F_l, F_u, \tau_1, \tau_2, \overline{V}_p, T$ ; dáno  $D_{im}, D_{om}$ 
2 Vytvoř populace jedinců  $\mathbf{P}_{op} = (\mathbf{X}_1, \dots, \mathbf{X}_N)$ ,  $\mathbf{X}_i = [x_j | j \in \langle 0, D_{im} \rangle, F, P_{cr}]$ 
3 for  $i = 0$ ;  $i < N$ ;  $i=i+1$ 
4   for  $j = 0$ ;  $j < D_{im}$ ;  $j=j+1$ 
5     Inicializuj  $x_j$  náhodně v  $D_{om}$ ;  $F = rand$ ;  $P_{cr} = rand$ 
6   endfor
7 endfor
8 Ohodnot jedince populace  $\mathbf{P}_{op}$ , Najdi nejlepšího jedince  $\mathbf{X}_{best}$ 
9 while ( $T$  není splněna)
10  for  $i = 0$ ;  $i < N$ ;  $i=i+1$ 
11     $F_{old} = \mathbf{X}_i.F$ ;  $P_{cr-old} = \mathbf{X}_i.P_{cr}$ 
12    Použij rovnice (4,5) a vypočítej nové hodnoty  $\mathbf{X}_i.F$ ,  $\mathbf{X}_i.P_{cr}$ 
13    Vyber náhodně  $\{\mathbf{X}_{r1}, \mathbf{X}_{r2}, \dots, \mathbf{X}_{rk}\}, rj \neq i, \mathbf{X}_{rj} \in \mathbf{P}_{op}$ 
14    Vypočítej trial vektor  $\mathbf{X}_t$  s využitím  $\overline{V}_p, \mathbf{X}_i, \mathbf{X}_{rj}, \mathbf{X}_{best}$ 
15    if  $fitness(\mathbf{X}_i)$  je lepší než  $fitness(\mathbf{X}_t)$ 
16       $\mathbf{X}_i.F = F_{old}$ ;  $\mathbf{X}_i.P_{cr} = P_{cr-old}$ 
17    else
18       $\mathbf{X}_i.x_j = \mathbf{X}_t.x_j | j \in \langle 0, D_{im} \rangle$ ;
19    endif
20  endfor
21  Najdi nejlepšího jedince  $\mathbf{X}_{best}$  v  $\mathbf{P}_{op}$ 
22 endwhile
23  $\mathbf{X}_{best}$  reprezentuje nejlepší nalezené řešení.

```

\mathbf{P}_{op} -populace individuí; N -počet individuí v populaci; F, P_{cr} -pracovní parametry; \overline{V}_p , -odchylkový vektor; D_{im} -počet dimenzí každého individua; D_{om} -domain pro každou jednotlivou dimenzi D_{im} ; T -ukončovací podmínka; \mathbf{X}_i -chromozom každého jedince neboli jedno možné řešení dané úlohy; platí, že $\mathbf{X}_{r1} \neq \mathbf{X}_{r2} \neq \dots \neq \mathbf{X}_{rk} \neq \mathbf{X}_i$, počet vektorů \mathbf{X}_{rk} je dán $\overline{V}_p \cdot D_{im}, D_{om}$ -je dáno charakterem úlohy. F_l, F_u, τ_1, τ_2 -pracovní parametry optimalizátoru jDE.

Podobně jako u DE, je i zde, pro jDE vytvořen tzv. trial vektor \mathbf{X}_t . Pokud je jeho fitness lepší než fitness \mathbf{X}_i , je vektor \mathbf{X}_i nahrazen vektorem \mathbf{X}_t a hodnoty F, P_{cr} jsou přepsány hodnotami vytvořenými v kroku 12. Další výpočet je stejný jako u algoritmu DE.

5.2 Algoritmus EPSDE

Algoritmus EPSDE (Mallipeddi & Suganthan 2010) je založen na podobném principu, jako DE (Storn & Price 1997; Price et al. 1999) nebo jDE (Brest et al. 2006) a vychází také z algoritmu JADE (Zhang & Sanderson 2009). Hlavní devizou EPSDE je však to, že zcela minimalizuje počet pracovních parametrů zadávaných uživatelem. Jedinými volitelnými parametry jsou tak N_{pop} – počet jedinců populace, ukončovací podmínka a hodnota L_p , což je doba v generacích, která udává, po jakou dobu od počátku evoluce nebude aktualizována hodnota \mathbf{cr}_{mean} , která je využita při operaci křížení. Algoritmus EPSDE je zobrazen na Alg. (5.2).2. Na rozdíl od jDE má EPSDE navíc několik interních pracovních parametrů. Jsou to vektory \mathbf{cr} , \mathbf{cr}_{mean} a matice \mathbf{CR}_{mem} . Tyto interní parametry ovlivňují operátor křížení. Parametry lze zapsat následovně:

$$\mathbf{cr} = [a_i | i \in \langle 0, N_{srt} \rangle] \quad (5.2.1)$$

$$\mathbf{cr}_{mean} = [b_i | i \in \langle 0, N_{srt} \rangle]$$

$$\mathbf{CR}_{mem} = \begin{pmatrix} c_{0,0} & \cdots & c_{0,N_{srt}} \\ \vdots & \ddots & \vdots \\ c_{L_p,0} & \cdots & c_{L_p,N_{srt}} \end{pmatrix}$$

\mathbf{cr} je tzv. četnost křížení (crossover rate) a platí, že $0 < a_i, \forall a_i \in \mathbf{cr}$. Vektor \mathbf{cr}_{mean} uchovává střední hodnoty z prvků $c_{i,j}, i \in \langle 0, N_{srt} \rangle$, které se nacházejí na řádcích matice \mathbf{CR}_{mem} – viz ř. 14 Alg. (5.2).2. Odchylkový vektor $\overline{\mathbf{V}}_p$ je volen zcela náhodně na základě hodnoty $u, u \sim U_{uniform} \langle 0, N_{srt} \rangle$, kde $N_{srt} = 4$ je počet strategií – viz ř. 34 Alg. (5.2).2 a také Tab. (5.2).1.

Podobně jako u DE je i zde prvním krokem algoritmu EPSDE inicializace populace \mathbf{P}_{op} jedinců $\mathbf{X}_i | i \in \langle 0, N_{pop} \rangle$, kteří jsou reprezentováni vektorem:

$$\mathbf{X}_i = [x_j | j \in \langle 0, D_{im} \rangle, F] \quad (5.2.2)$$

Každý jedinec \mathbf{X}_i reprezentuje jedno možné řešení úlohy. \mathbf{X}_i je D_{im} -dimenzionální vektor a uchovává konkrétní hodnoty pro jedno řešení v aktuální generaci G_{en} . Parametr F je označován jako mutační (váhová) konstanta a platí, že $0 < F \leq 2$ (Price & Storn 1997). Hodnota F může být pro některé typy úloh i větší např. 5. $\overline{\mathbf{V}}_p$ je tzv. odchylkový vektor, určený pro operaci mutace. V prvním kroku algoritmu jsou všichni jedinci umístěni na náhodné pozice v celém prostoru možných řešení:

$$\mathcal{H} = D_{om}^1 \times D_{om}^2 \times \dots \times D_{om}^k | k \in \mathbb{N}^+ \quad (5.2.3)$$

D_{om} je tzv. domain pro k -tou dimenzi, $D_{om} | D_{om} \in (min, max)$, každá dimenze může mít jiné hodnoty min, max resp. omezení prostoru možných řešení. Vektory \mathbf{cr} , \mathbf{cr}_{mean} a matice \mathbf{CR}_{mem} jsou naplněny hodnotami 0.5. Dalším krokem je ohodnocení populace \mathbf{P}_{op} – výpočet *fitness* funkce a nalezení individua s nejlepším *fitness* \mathbf{X}_{best} . Následuje evoluční cyklus, ve kterém dochází ke konvergenci populace. V každé generaci G_{en} je nejprve provedena kontrola podmínky $G_{en} > L_p^{EPSDE}$, kde $L_p^{EPSDE}, L_p^{EPSDE} \in \mathbb{N}^+$ a pokud je splněna, je aktualizován vektor \mathbf{cr}_{mean} .

Tabulka (5.2).1 Odchylkové vektory použité v optimalizátoru EPSDE

| | |
|--|---|
| \overline{V}_{p1} | rand/1/bin: $\mathbf{X}_{mut} = \mathbf{X}_{r1} + \mathbf{X}_{r1} \cdot F(\mathbf{X}_{r2} - \mathbf{X}_{r3})$ |
| \overline{V}_{p2} | rand-to-best/2/bin: $\mathbf{X}_{mut} = \mathbf{X}_i + \mathbf{X}_i \cdot F(\mathbf{X}_{best} - \mathbf{X}_i) + \mathbf{X}_i \cdot F(\mathbf{X}_{r1} - \mathbf{X}_{r2}) + \mathbf{X}_i \cdot F(\mathbf{X}_{r3} - \mathbf{X}_{r4})$ |
| \overline{V}_{p3} | rand/2/bin: $\mathbf{X}_{mut} = \mathbf{X}_{r1} + \mathbf{X}_{r1} \cdot F(\mathbf{X}_{r2} - \mathbf{X}_{r3}) + F(\mathbf{X}_{r4} - \mathbf{X}_{r5})$ |
| \overline{V}_{p4} | current-to-rand/1: $\mathbf{X}_{mut} = \mathbf{X}_i + N_{ormal}(0.0,1.0)(\mathbf{X}_{r1} - \mathbf{X}_i) + \mathbf{X}_i \cdot F(\mathbf{X}_{r2} - \mathbf{X}_{r3})$ |
| $N_{ormal}(0.0,1.0)$ značí náhodné číslo s normálním rozložením s omezením na rozsah hodnot (0.0,1.0). | |
| \overline{V}_{p4} viz (Iorio & Li 2004), $\overline{V}_{p1,2,3}$ viz (Zhang & Sanderson 2009) | |

Následuje krok mutace resp. výpočet vektoru \mathbf{X}_t s využitím odchylkového vektoru \overline{V}_p a poté krok křížení. (Storn & Price 1997; Price et al. 1999) odvodili několik vektorů \overline{V}_p a některé z nich se používají i v EPSDE. Jejich tvar je v Tab. (5.2).1. \mathbf{X}_{mut} je tzv. mutant, \mathbf{X}_{best} je prozatím nejlepší jedinec v \mathbf{P}_{op} , \mathbf{X}_{rj} jsou náhodně zvolení jedinci z populace \mathbf{P}_{op} pro které platí, že $\mathbf{X}_{r1} \neq \mathbf{X}_{r2} \neq \dots \neq \mathbf{X}_{rk} \neq \mathbf{X}_i$. Na rozdíl od původní DE hodnoty F nejsou konstantní. Další operací je křížení. Existují dva druhy označované jako *bin* a *exp* viz (Price et al. 1999; Zaharie 2009, 2007; Jeyakumar & Shanmugavelayutham 2011; Neri & Tironnen 2010). EPSDE optimalizátor používá jen verzi *bin*. Operace křížení je uvedena na ř. 20-27 Alg. (5.2).2 a nepoužívá se odchylkový vektor current-to-rand/1 - viz (Iorio & Li 2004). Operace křížení je definována jako:

$$\neg \left(\begin{array}{c} N_{ormal}(0.0,1.0) < cr[\overline{V}_p] \\ \vee \\ j \equiv U_{niform}(0, D_{im}) \end{array} \right) \equiv true \Rightarrow \mathbf{X}_t \cdot x_j = \mathbf{X}_i \cdot x_j, j \in \langle 0, D_{im} \rangle \quad (5.2.4)$$

Následuje výpočet fitness funkce pro jedince \mathbf{X}_i v populaci \mathbf{P}_{op} . Pokud je hodnota fitness funkce pro \mathbf{X}_t lepší než u jedince \mathbf{X}_i , je jedinec \mathbf{X}_i nahrazen jedincem \mathbf{X}_t a dále je odstraněn první řádek hodnot matice \mathbf{CR}_{mem} . Následně jsou pak všechny řádky matice posunuty o 1 řádek výše a do matice je přidán nový, poslední řádek a naplněn hodnotami vektoru \mathbf{cr} . V opačném případě je náhodně vygenerován nový vektor \overline{V}_p a stejně tak i vektor \mathbf{cr} je opětovně naplněn náhodnými čísly:

$$\mathbf{cr} = [a_i \sim N_{ormal}(\bar{x} = \mathbf{cr}_{mean}[i], \sigma = 0.01)], i \in \langle 0, N_{srt} \rangle \quad (5.2.5)$$

Na rozdíl od DE nebo jDE, vektor \overline{V}_p není u EPSDE uživatelsky volitelný. EPSDE pracuje se stejnými vektory jako JADE (Zhang & Sanderson 2009) a navíc používá vektor current-to-rand/1 navržený v (Iorio & Li 2004). Pro náš algoritmus byly zvoleny stejné vektory – viz Tab. (5.2).1. Tyto vektory se volí náhodně – viz ř. 34 Alg. (5.2).2.

Pokud \mathbf{X}_t překročí hranice hodnot min, max pro danou D_{om} , je možné jej vygenerovat zcela náhodně kdekoli v prostoru možných řešení \mathcal{H} . Poslední operací algoritmu jDE je selekce:

$$\mathbf{X}_i^{Gen+1} = \begin{cases} \mathbf{X}_t & \text{if } fitness(\mathbf{X}_t) \text{ je lepší než } fitness(\mathbf{X}_i^{Gen}) \\ \mathbf{X}_i^{Gen} & \text{jinak} \end{cases} \quad (5.2.6)$$

při které je rozhodnuto, zda trial vektor \mathbf{X}_t je lepší než jedinec \mathbf{X}_i v i -té generaci. Pokud ano, je \mathbf{X}_i nahrazen \mathbf{X}_t jinak je \mathbf{X}_i ponechán. Výpočet pak pokračuje generací $Gen + 1$ dokud není nalezeno optimální řešení úlohy nebo dokud hodnota G_{en} nedosáhne požadované hodnoty.

Algoritmus (5.2).2 EPSDE optimalizátor

```

1 Zvol  $N_{pop}, L_p^{EPSDE}, T$ ; Dáno  $D_{im}, D_{om}, N_{srt} = 4$ 
2 Vytvoř  $P_{op} = (X_1, \dots, X_N)$ ,  $X_i = [x_j | j \in \langle 0, D_{im} \rangle, F]$ ,  $i \in \langle 1, N_{pop} \rangle$ 
3 Inicializuj všechny  $X_i$  náhodně v  $D_{om}$ ;  $F = rand$ ;  $P_{cr} = rand$ 
4  $CR_{mem}[i].strategy(j) = 0.5$ ,  $\forall i, i \in \langle 0, L_p \rangle, \forall j, j \in \langle 0, N_{srt} \rangle$ 
5  $cr_{mean}[i] = 0.5$ ;  $\forall i, i \in \langle 0, N_{srt} \rangle$ 
6  $cr[i] = 0.5$ ,  $\forall i, i \in \langle 0, N_{srt} \rangle$ 
7 Ohodnot  $P_{op}$ , Najdi nejlepší ind  $X_{best}$ ,  $G = 1$ 
8 while (  $T$  není splněno )
9   if  $G_{gen} > L_p^{EPSDE}$ 
10     for  $i = 0$ ;  $i < N_{srt}$ ;  $i=i+1$ 
11       for  $j = 0$ ;  $j < CR_{mem}.count$ ;  $j=j+1$ 
12          $R1 = R1 + CR_{mem}[j].Strategy[i]$ 
13       endfor
14        $cr_{mean}[i] = R1/CR_{mem}.count$ 
15     endfor
16   endif
17   for  $i = 0$ ;  $i < N$ ;  $i=i+1$ 
18     Zvol náhodně  $\{X_{r1}, X_{r2}, \dots, X_{rk}\}, rj \neq i, X_{rj} \in P_{op}$ 
19     Vypočti trial vektor  $X_t$  užitím  $\bar{V}_p, X_i, X_{rj}, X_{best}$ 
20     if  $\bar{V}_p \neq Current - to - rand/1$  - vektor 4., Tab. (5.2) .1
21       for  $j = 0$ ;  $j < D_{im}$ ;  $j=j+1$ 
22          $R1 = Uniform(0, D_{im})$ ;  $R2 = Normal(0.0, 1.0)$ 
23         if  $\neg (R2 < cr[\bar{V}_p] \text{ or } j \equiv R1)$ 
24            $X_t.x_j = X_i.x_j$ 
25         endif
26       endfor
27     endif
28     if  $fitness(X_t)$  je lepší než  $fitness(X_i)$ 
29        $X_t.x_j = X_i.x_j, \forall j, j \in \langle 0, D_{im} \rangle$ 
30       Odstraň  $CR_{mem}[0]$  z pole
31       Přidej nový prvek nakonec  $CR_{mem}$  pole
32        $CR_{mem}[L_p - 1].Strategy(j) = cr[j], \forall j, j \in \langle 0, D_{im} \rangle$ 
33     else
34        $\bar{V}_p = Uniform(0, N_{srt})$ ;  $X_i.F = Normal(\bar{x} = 0.3, \sigma = 0.5)$ 
35       for  $j = 0$ ;  $j < N_{srt}$ ;  $j=j+1$ 
36         do  $cr[j] = Normal(\bar{x} = cr_{mean}[j], \sigma = 0.01)$ 
37         while ( $cr[j] < 0$  or  $cr[j] > 1.0$ )
38       endfor
39     endif
40   endfor
41   Najdi nejlepší ind  $X_{best}$  v  $P_{op}$ 
42 endwhile
43  $X_{best}$  reprezentuje nejlepší odhadnuté řešení

```

P_{op} – populace individuí; N_{pop} – počet individuí v populaci; \bar{V}_p – odchylkový vektor; L_p^{EPSDE} – perioda učení; D_{im} – počet dimenzí pro každé individuum; D_{om} – domain pro každé individuum a pro každou dimenzi D_{im} ; T – ukončovací podmínka; X_i – chromozom každého jedince neboli jedno možné řešení dané úlohy; platí, že $X_{r1} \neq X_{r2} \neq \dots \neq X_{rk} \neq X_i$, a počet X_{rk} vektorů je dán \bar{V}_p . D_{im}, D_{om} – je dáno charakterem řešené úlohy, cr – vektor uchovávající hodnoty pravděpodobnosti křížení s ohledem na vektor \bar{V}_p , cr_{mean} – střední hodnota pro cr , N_{srt} – počet strategií, platí, že $N_{srt} = 4$, CR_{mem} – paměť pro jednotlivé strategie, ukládá se zde cr_{mean} úspěšných strategií. Koncová pozice pole CR_{mem} znamená pozici $[L_p - 1]$ protože pozice jsou počítány od nuly, $Normal(0.3, 0.5)$ – značí náhodné číslo s normálním rozložením se střední hodnotou $\bar{x} = 0.3$ a směrodatnou odchylkou $\sigma = 0.5$. $Uniform(0, N_{srt})$ – náhodné číslo s uniformním rozložením v rozsahu hodnot $\langle 0, N_{srt} \rangle$, $[L_p - 1]$ – značí index a jeho hodnota je využita na ř. 32., $count$ – značí počet prvků vektoru. Na ř. 34 jsou pracovní parametry \bar{x}, σ de-facto volitelné a mohou výrazně ovlivnit kvalitu výsledku. Při algoritmicizaci byl použit následující příkaz:

`g_population[k].F = MathNet.Numerics.Distributions.Normal.Sample(g_EApar_NormDist_mean, g_EApar_NormDist_stddev);`

 kde `g_EApar_NormDist_mean=0.5` a `g_EApar_NormDist_stddev=0.3`. Hodnoty jsou tedy odlišné od nastavení originálního algoritmu na ř. 34, a to z toho důvodu, aby bylo možné dosáhnout co nejlepších výsledků, a především zajistit stabilní konvergenci.

5.3 CMA evoluční strategie

Algoritmus CMA neboli CMA-Evoluční Strategie (CMA, CMA-ES) využívající adaptaci kovarianční matice (Hansen et al. 1995; Hansen & Ostermeier 1996, 1997, 2001; Hansen et al. 2003; Hansen & Kern 2004; Hansen 2006; Hansen 2013a,b) je velmi populární populační stochastický optimalizátor. Je určený pro numerickou optimalizaci nelineárních nekonvexních a spojitých optimalizačních problémů. Po dobu téměř jednoho desetiletí byl považován za jeden z nejmodernějších optimalizačních nástrojů pro řešení problémů spojitě optimalizace. V dnešní době existuje bezpočet jeho modifikací. CMA náleží do skupiny algoritmů μ, λ -ES. ES (Beyer & Schwefel 2002) se dělí na tři skupiny označované jako 1+1-ES, μ, λ -ES a $\mu + \lambda$ -ES. Zjednodušeně lze popsat tyto strategie tak, že v prvním případě se každý jedinec pohybuje při evoluci k jinému jedinci (v podstatě lokální prohledávání), v dalších dvou případech se skupina jedinců pohybuje k jiné skupině jedinců. Poslední dvě skupiny se liší tak, že první z nich se při evoluci vybírá μ nejlepších jedinců z původní populace a generuje λ testovacích řešení (tzv. offsprings), druhý z nich pak pracuje tak, že vybírá μ nejlepších jedinců z množiny jedinců λ . Podobně jako mnoho jiných evolučních algoritmů je i CMA založen na principech biologické evoluce, tj. selekce, křížení a mutace. CMA je velmi odolný proti uváznutí v lokálním optimu s ohledem na typ optimalizované funkce a pracovní parametry. Při běhu generuje CMA množinu jedinců v prohledávaném prostoru podle multivarietního normálního rozložení a hledá optimální řešení s využitím aktualizace průměrné hodnoty a kovarianční matice. Hodnoty kovarianční matice a průměru jsou pak použity pro výpočet pozic nových jedinců, resp. kandidátských řešení v dalších generacích; je tedy vypočítáno nové rozdělení distribuční funkce. CMA od jiných evolučních algoritmů odlišuje to, že tvar distribuční funkce se počítá podle hodnot získaných v kroku 'mutace' distribuční funkce. Např. DE, jDE, SaDE apod. vypočítávají pozice nových jedinců podle hodnot účelové funkce nejlepších jedinců. Základní algoritmizace pro testovací účely je převzata podle (Hansen 2013a,b) a má následující formu:

Algoritmus (5.3).1 CMA Optimizer

```
1 Inicializuj CMA-ES:
2     Uživatelsky definované konstanty
3     Pracovní parametry pro operátor Selektce
4     Pracovní parametry pro operátor Adaptace
5     Interní parametry
6 while ( T není splněna )
7     Vygeneruj nová řešení 'lambda' a ohodnoť je
8     Setřídění populace, výpočet váženého průměru
9     Výpočet tzv. evoluční cesty
10    Výpočet adaptace kovarianční matice
11    Výpočet parametru sigma
12    Diagonalizace kovarianční matice,  $B \cdot \text{diag}(D.^2) \cdot B'$ 
13 endwhile
```

Algoritmus CMA je využit jako primární a také jako sekundární evoluční optimalizátor v úloze korekce distorze obrazu s využitím kaskádového evolučního optimalizátoru. V druhém případě stojí v kaskádě na druhém místě a používá výsledky dosažené algoritmem jDE. Výsledek výpočtu CMA plně závisí na výsledcích poskytovaných algoritmem jDE. CMA pracuje v až 7-mi dimenzionálním prostoru, účelová funkce je uvažována jako neomezená, silně nelineární, multimodální a neseperabilní (což je to nejhorší co může existovat). CMA bohužel nepatří mezi algoritmicky jednoduché optimalizátory, spíše naopak. Její kompletní praktická realizace např. v jazyce C nebo C++ je poměrně značně zdlouhavá a jen malá nepřesnost pak vede k celkovému selhání optimalizátoru.

LITERATURA

- Abbass A.H. 2002. The self-adaptive pareto differential evolution algorithm. In: *Proceedings of the 2002 Congress on Evolutionary Computation*. vol. 1, pp. 831 – 836.
- Auger A., Hansen N. 2005. A Restart CMA Evolution Strategy with increasing population size. In: *IEEE Congress on Evolutionary Computation*. CEC 2005, pp. 1769-1776.
- Auger A., Hansen N., Zepa P., Ros J.M.R., Schoenauer M. 2009. Experimental comparisons of several derivative free optimization algorithms. In: *Experimental Algorithms, Lecture Notes in Computer Science*, Jan Vahrenhold ed., Springer Heidelberg, Berlin.
- Barricelli N. 1954. Esempi numerici di processi di evoluzione. *Methodos*, pp. 45-68.
- Barricelli N. 1957. Symbiogenetic evolution processes realized by artificial methods. *Methodos*. pp. 143–182.
- Barricelli N. 1961. Numerical testing of evolution theories. Part I: Theoretical introduction and basic tests, Department of Biology, Division of Molecular Biology, Vanderbilt University, Nashville, Tennessee, Acta Biotheoretica, Springer Netherlands, ISSN 0001-5342.
- Barricelli N. 1963. Numerical testing of evolution theories. Part II: Preliminary tests of performance, symbiogenesis and terrestrial life, Department of Biology, Division of Molecular Biology, Vanderbilt University, Nashville, Tennessee, Acta Biotheoretica, Springer Netherlands, ISSN 0001-5342.
- Barrera J., Coello C.A.C. 2009. A Review of Particle Swarm Optimization Methods Used for Multimodal Optimization. *Innovations in Swarm Intelligence*, pp. 9-37.
- Beyer H.G., Schwefel H.P. 2002. Evolution Strategies a Comprehensive Introduction. *Journal of Natural Computing*, vol. 1, no. 1, pp. 3-52.
- Brest J., Greiner S., Boškovič B., Mernik M., Žumer V. 2006. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*. vol. 10, no. 6, pp. 646-657.
- Brest J., Maučec M.S. 2011. Self-adaptive differential evolution algorithm using population size reduction and three strategies. *Soft computing*. vol. 15, no. 11, pp. 2157-2174.
- Brest J., Zamuda A., Fister I., Maucec M.S. 2010. Large Scale Global Optimization using Self-adaptive Differential Evolution Algorithm. In: *IEEE World Congress on Computational Intelligence*. vol. 1, pp. 1-8.
- Das S., Konar A., Chakraborty U. K. 2005. Two improved differential evolution schemes for faster global search. In: *Genetic and Evolutionary Computation Conference (GECCO)*. pp. 991-998.
- Das S., Suganthan P.N. 2010. Differential Evolution: A Survey of the State-of-the-Art. *IEEE Transactions on Evolutionary Computation*. vol. 15, no 1., pp. 4-31.
- Dorigo M. 1992. *Optimization, Learning and Natural Algorithms*. PhD thesis, Politecnico di Milano, Italy.
- Eiben A.E., Hinterding R., Michalewicz Z. 1999. Parameter control in evolutionary algorithms. *IEEE Transaction on evolutionary computation*. vol. 3, no. 2, pp. 124-141.
- Floudas Ch.A., Pardalos P. M. 2009. *Encyclopedia of Optimization*. Springer, 2nd ed., 7 volumes. p. 4626,
- Fogel L.J. 1966. *Artificial Intelligence through Simulated Evolution*. New York, USA. ISBN-13: 978-0471265160.
- Fogel L.J. 1999 *Intelligence through Simulated Evolution: Forty Years of Evolutionary Programming*. John Wiley Eds.
- Fogel D.B., Chellapilla K. 1998 Revisiting Evolutionary Programming. *Applications and Science of Computational Intelligence*. vol. 3390, pp.2-11.
- Gamperle R., S.D. Muller, Koumoutsakos. P. 2002. A parameter study for differential evolution. In: *WSEAS International Conference on Advances in Intelligent Systems*. pp. 293-298.
- Glover F. 1986. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*. vol. 13, pp. 533-549.

- Glover F., Kochenberger G. A. 2003. *Handbook of Metaheuristics*. Springer Verlag Eds. ISBN 978-3-319-91086-4.
- Goldberg D.E. 1987. Simple genetic algorithms and the minimal deceptive problem. In: *Genetic Algorithms and Simulated Annealing*. London, Pitman, pp. 74-88.
- Goldberg D.E. 1989. *Genetic Algorithms in search, optimization, and machine learning*. Addison-Wesley Publishing Corporation, ISBN-13: 978-0201157673.
- Hansen N., Ostermeier A. 1996. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In: *IEEE International Conference on Evolutionary Computation*. pp. 312-317.
- Hansen N. 2013a. [online]. Hansen N. ©2013 [cit. 20.5.2020]. Dostupné z: <http://www.lri.fr/~hansen/>
- Hansen N. 2013b [online]. Wikipedie ©2013 [cit. 20.5.2020]. Dostupné z: [xhttp://en.wikipedia.org/wiki/CMA-ES](http://en.wikipedia.org/wiki/CMA-ES)
- Hansen N. 2006. The CMA Evolution Strategy: A Comparing Review. In: *Towards a new evolutionary computation. Advances in estimation of distribution algorithms*. J.A. Lozano, P. Larrañaga, I. Inza and E. Bengoetxea (Eds.), Springer, pp. 75-102.
- Hansen N., Ostermeier A., Gawelczyk A. 1995. On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation. In: *Proceedings of the Sixth International Conference on Genetic Algorithms*. L. Eshelman (Ed.), Pittsburgh, Morgan Kaufmann, pp. 57-64.
- Hansen N., Ostermeier A. 1996. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation, In: *IEEE International Conference on Evolutionary Computation*. pp. 312-317.
- Hansen N., Ostermeier A. 1997. Convergence properties of evolution strategies with the derandomized covariance matrix adaptation: The $(\mu/\lambda, \lambda)$ -ES. In: *EUFIT'97, 5th Europ. Congr. on Intelligent Techniques and Soft Computing*. pp. 650-654.
- Hansen N., Ostermeier A. 2001. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*. vol. 9, no. 2, pp. 159-195.
- Hansen N., Müller S.D., Koumoutsakos P. 2003. Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation*. vol. 11, no. 1, pp. 1-18.
- Hansen N., Kern S. 2004, Evaluating the CMA Evolution Strategy on Multimodal Test Functions. In: *8th International Conference on Parallel Problem Solving from Nature PPSN VIII*. pp. 282-291.
- Holland J.H. 1962. Outline for a Logical Theory of Adaptive Systems. *Journal of the ACM*. vol. 9, no. 3, pp. 297-314.
- Holland J.H. 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, p. 232. ISBN: 9780262581110.
- Holland J.H. 1992. *Adaptation in natural and artificial systems: an introductory analysis with application to biology, control, and artificial intelligence*. Cambridge, MIT Press. p. 211. ISBN: 9780262275552.
- Chen X.S., Ong Y.S., Lim M.H., Tan K.C. 2011 A Multi-Facet Survey on Memetic Computation. *IEEE Transactions on Evolutionary Computation*. vol. 15, no. 5, pp. 591-607.
- Igel Ch., Suttorp T., Hansen N. 2006. A computational efficient covariance matrix update and a (1+1)-CMA for evolution strategies. In: *GECCO '06 Proceedings of the 8th annual conference on Genetic and evolutionary computation*. pp. 453-460.
- Iorio A., Li X. 2004. Solving rotated multi-objective optimization problems using differential evolution. In: *Australian Conference on Artificial Intelligence*. Cairns, Australia, pp. 861-872.
- Jastrebski G.A., Arnold D.V. 2006. Improving evolution strategies through active covariance matrix adaptation. In: *IEEE Congress on Evolutionary Computation CEC 2006*. pp. 2814 – 2821.
- Jeyakumar G., Shanmugavelayutham C. 2011. Experimental Study on Recent Advances in Differential Evolution Algorithm. *International Journal of Applied Evolutionary Computation*. vol. 2, no. 2, pp. 58-81.
- Kennedy J., Eberhart R.C. 1995. Particle swarm optimization. In: *IEEE International Conference on Neural Networks*. vol. 4, pp. 1942-1948.
- Kirkpatrick S., Gelatt C.D., Vecchi M.P. 1983. Optimization by Simulated Annealing, *Science*. vol. 220, no. 4598, pp. 671-680.
- Koza J. R. 1994. *Genetic Programming – 2*, MIT Press, Cambridge, ISBN: 9780262111898.

- Koza J. 1996. *Genetic Programming: On the Programming of the Computers by means of Natural Selection*. Fifth edition, MIT Press Cambridge, Massachusetts, London, England. ISBN-13: 978-0262111706.
- Liu J., Lampinen J. 2002. On setting the control parameter of the differential evolution method. In: *8th International Conference Soft Computing*, pp. 11-18.
- Liu J., Lampinen J. 2005. A Fuzzy Adaptive Differential Evolution Algorithm. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*. vol. 9, no. 6, pp. 448-462.
- Mallipeddi R., Suganthan P.N. 2010. Differential Evolution Algorithm with Ensemble of Parameters and Mutation and Crossover Strategies. In: *International Conference on Swarm, Evolutionary, and Memetic Computing SEMCCO 2010*, pp. 71-78.
- Metropolis N., Ulam S. 1949. The Monte Carlo method. *Journal of the American Statistical Association*. vol. 44, pp. 335-341.
- Montes E.M., Reyes J.V., Coello C.A.C. 2006. A Comparative Study of Differential Evolution Variants for Global Optimization. In: *GECCO '06 Proceedings of the 8th annual conference on Genetic and evolutionary computation*. pp. 485-492.
- Neri F., Tirronen V. 2010. Recent advances in differential evolution: a survey and experimental analysis. *Artificial Intelligence Review*. vol. 33, no. 1-2, pp. 61-106.
- Noman N., Bollegala D., Iba H. 2011. An Adaptive Differential Evolution Algorithm. In: *IEEE Congress on Evolutionary Computation*. pp. 2229-2236.
- Pardalos P.M. 2020 [online]. Dr. Panos M. Pardalos ©2020 [cit. 20.5.2020]. Dostupné z: Web1: <http://www.ise.ufl.edu/pardalos/>
- Pošik P., Huyer W. 2012. Restarted Local Search Algorithms for Continuous Black Box Optimization. *Evolutionary Computation*. vol. 20, pp 575-607.
- Pošik P., Huyer W., Pál L. 2012. A Comparison of Global Search Algorithms for Continuous Black Box Optimization. *Evolutionary Computation*. vol. 20, pp 509-541.
- Pošik P., Kubalik J. 2012. Experimental Comparison of Six Population-Based Algorithms for Continuous Black Box Optimization. *Evolutionary Computation*. vol. 20, pp. 483-508.
- Price K., Come D., Dorigo M., Glover F. 1999. *An Introduction to Differential Evolution*, Eds. London, U.K. McGraw-Hill, 79-108.
- Qin A.K., Suganthan P.N. 2005. Self-adaptive differential evolution algorithm for numerical optimization. In: *IEEE Congress on Evolutionary Computation*. vol. 2, pp. 1785-1791.
- Qin A. K., Huang V. L., Suganthan P. N. 2009. Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization. *IEEE Transactions on evolutionary computation*. vol. 13, no 2, pp. 398-417.
- Rechenberg I. 1973. *Evolutionsstrategien: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, Frommann-Holzboog Eds., Stuttgart, Germany. ISBN-13: 978-3772803734.
- Rechenberg I. 1994. *Evolutionsstrategie '94*. Frommann-Holzboog Ed., Stuttgart. ISBN 13: 9783772816420.
- Schwefel H.P. 1995. *Evolution and Optimum Seeking*. JohnWiley&Son Eds., New York, USA. ISBN:978-0-471-57148-3.
- Smit S.K., Eiben A.E. 2009. Comparing Parameter Tuning Methods for Evolutionary Algorithms. In: *IEEE Congress on Evolutionary Computation*. pp. 399-406.
- Smit S.K., Eiben A.E. 2010a. Parameter Tuning of Evolutionary Algorithms: Generalist vs. Specialist. *Applications of Evolutionary Computation*. vol. 6024, pp. 542-551.
- Smit S.K., Eiben A.E. 2010b. Parameter Tuning for Configuring and Analyzing Evolutionary Algorithms. *Swarm and Evolutionary Computation*. vol. 1, no. 1, pp. 19-31.
- Storn R., Price K. 1997. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*. vol. 11, pp. 341-359.

- Teo J. 2005. Exploring dynamic self-adaptive populations in differential evolution. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*. vol. 10, no. 8, pp. 673-686.
- Urfahoglu O. 2004. Robust Estimation of Camera Rotation, Translation and Focal Length at High Outlier Rates. In: *1st Canadian Conference on Computer and Robot Vision*. pp. 464-471.
- Zaharie D. 2007. A Comparative Analysis of Crossover Variants in Differential Evolution. In: *International Multi-conference on Computer Science and Information Technology*. pp. 171-181.
- Zaharie D. 2009. Influence of crossover on the behavior of differential evolution algorithms. *Applied Soft Computing*. vol. 9, no. 3, pp. 1126-1138.
- Zhang J., Sanderson A.C. 2009. *JADE: Adaptive Differential Evolution with Optional External Archive*. *IEEE Transactions on Evolutionary Computation*. vol. 13, no. 5, pp. 945 – 958.
- Wang Y, Cai Z., Zhang Q. 2011. Differential Evolution with Composite Trial Vector Generation Strategies and Control Parameters. *IEEE Transactions on evolutionary computation*. vol 15. no. 1, pp. 55 – 66.
- Whitley D., Mathias K., Rana S., Dzubera J. 1996. Evaluating Evolutionary Algorithms. *Artificial Intelligence*. vol. 85, pp. 245–276.
- Wolpert D.H., McReady W.G. 1995. *No free-lunch theorems for search*, Technical Report 95-02-010, SantaFe Institute.

6 Metody kalibrace obrazu biometrického scanneru

Všechny optické soustavy jsou zatíženy do jisté míry jednou nebo více optickými vadami. Soudková distorze obrazu patří mezi ně. V předkládané práci je s využitím inovativní metody řešen problém odstranění centrické soudkové distorze statického obrazu, který slouží pro biometrickou identifikaci osob pomocí 2D kontury lidské ruky. Navržená metoda využívá kaskádového uspořádání dvou algoritmů - klasické metaheuristiky, označované jako jDE-diferenciální evoluce a algoritmus označovaný jako „Covariance Matrix Adaptation Evolution Strategy“ (CMA-ES nebo jen CMA). Optimalizátory řeší metodami inverzního inženýrství a numerické matematiky způsob, jak najít korektní parametry algebraické polynomiální rovnice n -tého stupně, jejíž aplikací je možné z obrazu zatíženého soudkovou distorzí získat obraz touto vadou nezatížený. Navržená metoda poskytuje kvalitní a časově přijatelný způsob optimalizace a možnost volby přesnosti aproximace. S využitím získaných koeficientů je pak metodou označovanou jako back-mapping možné trvale korigovat vadu centrické soudkovitosti u navrženého biometrického scanneru. Rozsáhlé experimenty presentované v následujících odstavcích umožňují lépe pochopit souvislosti, dosahovanou přesnost a možnosti použitých evolučních optimalizátorů v širších souvislostech.

6.1 Kalibrace obrazu založená na metodě back-mappingu

6.1.1 Úvod

Při práci s optickými soustavami tedy čočkou nebo zrcadlem se setkáváme s mnoha vadami zobrazení (Welford 1986; Sasián 2012). Pět nejznámějších základních vad zobrazení neboli aberací bylo poprvé popsáno v roce 1857 německým matematikem Ludwigem von Seidelem (1821-1896†) pro monochromatické světlo. Jedná se o sférickou aberaci, komu, astigmatismus, sklenutí pole (curvature of field) a distorzi. Často je k těmto aberacím přidávána ještě jedna, a to chromatická aberace (vada barevnosti). U většiny běžně dostupných komerčních optických soustav, které se používají např. ve fotoaparátech nebo kamerách jsou výše popsané vady do jisté míry kompenzovány výrobcem již při návrhu optické soustavy. Některé jako např. distorze (Brown 1956) se výrazně projeví v okamžiku, kdy má optická soustava zobrazit takový obraz, který je snímán např. objektivem s velkým zorným polem (FOV - Field Of View) a malou vstupní pupilou (Haneishi et al. 1995). Typickým zástupcem jsou objektivy označované jako rybí oko (fish-eye) (Basu & Licardie 1995; Schwable 2005). Dokonalá korekce všech vad za všech okolností není z fyzikálních důvodů realizovatelná. Pro specifické účely se tak používají různé druhy optických soustav - např. výměnné objektivy fotoaparátů. V oblasti digitálního zpracování obrazu např. u fotoaparátů a videokamer je nutné optické vady, které nebyly korigovány soustavou čoček nebo i zrcadel korigovat jiným vhodným způsobem. V dnešní době je běžné, že obraz z fotoaparátu nebo videokamery se zpracovává na počítači. Zpracování tzv. digitálního obrazu je náročné především z důvodu obrovského množství vstupních dat, které v případě digitální korekce optických vad reprezentuje současně i objem výstupních dat. To se týká např. korekce distorze obrazu u biometrického scanneru, což je náš případ. V presentované práci jsou využity informace ze tří různých oblastí:

- 1) Korekce vad optických soustav (Virendra 1991; Fischer & Tadic 2000; Sasián 2012), konkrétně korekce soudkového zkreslení obrazu (Brown 1965); polynomiální metoda.
- 2) Vybrané moderní evoluční algoritmy (Floudas & Pardalos 2009; Pardalos 2020), které jsou zde použity jako standardní optimalizační prostředky pro optimalizaci až sedmi-dimenzionální účelové funkce.
- 3) A také některé poznatky z oblasti kaskádových estimátorů (Patnaik et al. 1996).

Navržený algoritmus, který umožňuje takovýmto způsobem odstranit nechtěné aberace obrazu, v našem případě soudkovité nebo polštářové zkreslení malého rozsahu, budeme označovat jako Kaskádový Evoluční Estimátor (CEE). První část článku je věnována obecnému přehledu literatury v oblasti korekce soudkové distorze obrazu, jsou zde také podrobněji popsány použité evoluční optimalizátory, následuje krátký odstavec věnovaný tématice kaskádových estimátorů a pak také odstavec věnovaný teoretickému rozboru možností při korekci soudkového zkreslení obrazu a používané metody. Další kapitola popisuje CEE. Závěr této kapitoly je věnován volbě pracovních parametrů a samozřejmě i experimentálními výsledkům, které popisují chování navrženého CEE za různých pracovních podmínek.

6.1.2 Související publikace

Způsob, jak korigovat distorzi obrazu ať už centrickou nebo necentrickou navrhl již (Conrady 1919). (Brown 1971) s využitím poznatků v (Magill 1955) odvodil a experimentálně ověřil vzorec, který vysvětluje rozdíly v distorzi při měnící se ohniskové vzdálenosti. Velmi známé jsou také práce (Brown 1956) a především (Brown 1965). S nástupem počítačů ke konci 80. let se Brownův výzkum začal běžně využívat především v armádě a v průmyslu. Později i v astronomii, letectví a v mnoha dalších vědních oborech a v nezměněné podobě je využíván dodnes. Publikace (Brown 1956; 1965) se do jisté míry staly referenční v dané vědní oblasti. Korekce centrické distorze obrazu nemusí vždy vyhovovat náročným armádním podmínkám, proto byly publikovány práce (Brown 1964, 1965, 1966), kde byl kromě jiného také detailně rozpracován postup odstranění necentrické distorze obrazu.

Postupy pro odstranění distorze obrazu navržené v odborné literatuře je možné rozdělit do různých skupin např. podle toho, jaký je použit kalibrační obrazec nebo pro jaký typ objektivu je metoda navržena apod. Shrňme nyní některé z nejznámějších prací, které současně roztřídíme do různých kategorií podle využití: např. lékařství (Haneishi et al. 1995; Schwartz 1980; Asari et al. 1999; Zhang et al. 2000; Asari 2004; Pierre et al. 2009; Chen et al. 2011; Melo et al. 2011; Melo et al. 2012) nebo v oblasti strojového vidění (machine vision) (Fitzgibbon 2001; Lenz & Tsai 1988), vojenská fotogrammetrie a letectví (Brown 1965; Tang 2012; Tang et al. 2012), průzkum (Kim & Kim 2009), robotika (Tsai 1987a,b; Štěpán 2001), průmysl obecně (Butler 1991; Fernandes et al. 1997; Sun et al. 2008) atd. Kompletní přehled výše popsaných metod, které se zabývají radiální a necentrickou distorzi obrazu u čoček představil (Fryer 1986) a zohlednil také vědecký pokrok v dané oblasti a nové technologické možnosti. Historický přehled je možné najít např. v (Lenz & Tsai 1988; Clarke & Fryer 1998). Další přehledové studie novějšího data z oblasti makro-fotogrammetrie a počítačového vidění byly uvedeny v (McGlone 2004; Remondino & Fraser 2006) a zajímavá je také publikace (Slama 1980).

Metoda korekce distorze je obvykle založena na kalibračním obrazci, který je sestaven buď z bodů (Tsai 1987; Shah & Aggarwal 1994; Asari et al. 1999; Zhang et al. 2000; Tang et al. 2012), úseček/mřížky (Devernay 1995; Prescott & McLean 1997; Devernay & Faugeras 2001; Thormahlen et al. 2003; Tommaselli et al. 2012; Das & Patil 2006; Romero & Gomez 2007) nebo z jiných kombinovaných geometrických obrazců (Fernandes et al. 1997; Wang & Tseng 2000; Wang et al. 2009) nebo také hledáním vhodných struktur ve zkoumaném obraze (Brito et al. 2013). (Prescott & McLean 1997) prokázali, že metody využívající jako kalibrační obrazec body nebo úsečky poskytují shodné výsledky co do přesnosti.

Tzv. klasické polynomiální metody neboli metody založené na polynomiálním radiálním modelu (PM) (Brown 1965) jsou reprezentovány např. pracemi (Weng et al. 1992; Asari et al. 1999). Nepředstavují však jediný způsob, jak popsat distorzi obrazu. Existují další postupy např. tzv. nepolynomiální metody (Kim & Kim 2009), které se vyznačují především svojí rychlostí. (Dhane et al. 2012) navrhli tzv. „nepolynomiální nelineární radiální rozpínací metodu“. (Basu & Licardie 1995) navrhli logaritmický distorzní model nazývaný „Fish-Eye Transform“ (FET) a také polynomiální transformaci pro objektivu typu rybí oko „Polynomial Fish-eye Transform“ (PFET) a na základě experimentů došli k závěru, že PFET pracuje lépe než FET model. (Fitzgibbon 2001) navrhl modifikaci polynomiálního modelu (Brown 1965) a nazval jej model s lomeným výrazem. V anglické odborné literatuře se všude používá

výraz „Division Model – DM“. Jeho výhodou je to, že je schopen odstranit i velkou distorzi obrazu lépe než klasický (PM). Zjednodušený DM je také použit v (Wang et al. 2009). Komparativní studie výkonnosti polynomiálních modelů (PM) – lichý řád, PFET a DM pro objektivu typu rybí oko je prezentována v (Huges et al. 2009).

Potřeba korekce distorze obrazu vzniká za určitých okolností téměř u všech optických soustav. Samostatnou kapitolu tvoří objektivy označované jako Fish-eye, které jsou schopné zobrazit zorné pole do rozsahu až $\sim 180^\circ$ a které jsou velmi oblíbené v oblasti digitálního zpracování obrazu. Bylo navrženo nepřeberné množství metod pro odstranění distorze u takovýchto objektivů např. (Shah & Aggarwal 1994; Dhane et al. 2012) atd. Tyto metody mohou být buď zcela automatické označované jako „self/auto calibrating“ (Wang & Tseng 2000; Brito et al. 2013) nebo založené na čistě iterativním „ručním“ odhadu parametrů (Haneishi et al. 1995; Cucchiara et al. 2003; Das & Patil 2006). Pro běžné komerční digitální fotoaparáty s kvalitním objektivem, u nichž distorze s ohledem na použitou ohniskovou vzdálenost není tak velká, je možné použít metodu navrženou v (Das & Patil 2006). Skupina výzkumníků (Asari et al. 1999) navrhla iterativní algoritmus, který vychází z dřívějších výzkumů (Smith et al. 1992; Haneishi et al. 1995) a řeší problém soudkové distorze obrazu z endoskopu. Předpokládají, že střed distorze je umístěn ve středu obrazu. Problém 2D distorze obrazu je chápán jako čistě nelineární. (Asari et al. 1999) převádí 2D problém nelineární transformace obrazu na řešení 1D problému s využitím iterativní metody.

Práce (Brown 1965, 1971; Haneishi et al. 1995; Asari et al. 1999) se kromě dalších zde citovaných staly primární předlohou pro metodu korekce distorze s využitím CEE prezentovanou v této kapitole.

6.1.3 Kaskádové uspořádání algoritmů

Myšlenka kaskádových estimátorů (CE) je dobře známá především z oblasti počítačového vidění a digitalizovaného zpracování dat obecně. V žádném případě se nejedná o hojně se vyskytující metody. Spíše naopak. CE tvoří vzácnější, řídké se vyskytující téma. Je to dáno především značnou složitostí algoritmizace. Praktické využití CE je mnohem širší a zasahuje do mnoha vědních oborů. Nasazení CE je vhodné všude tam, kde není možné danou úlohu řešit s využitím jednoduchého optimalizátoru. Některé CE mohou být sestaveny z jednoho druhu/typu jednoduchého optimalizátoru – jedna nebo více metod, které jsou obvykle uspořádány sériově nebo paralelně. Nyní popíšeme několik zajímavých publikací, kde se kaskádové estimátory používají k úspěšnému řešení dané úlohy.

V práci (Dollar et al. 2010) byl navržen velmi rychlý a přesný algoritmus pro výpočet/odhad 2D pozice objektů označený autory jako „kaskádová polohová regrese“ (Cascaded Pose Regression) (CPR), která umí zpřesnit počáteční odhad a každé vylepšení přitom využívá jiný regresor. Jednotlivé regresory jsou vázány sériově a výstup jednoho regresoru tvoří vstup dalšího regresoru. Presentované výsledky prokázaly, že CPR je rychlá, přesná a jednoduše implementovatelná metoda. Autoři v (Coroneos et al. 1996) použili množinu kaskádových nelineárních optimalizátorů (Cascaded Nonlinear Optimizers) (CNO), kde jeden je vždy následovaný dalším v určité předem definovaném sledu. Optimalizátory řeší problém designu a konfigurace nadzvukových a podzvukových letadel a také proudových motorů pro vysokorychlostní civilní leteckou přepravu. V článku (Verschae & Ruiz-del-Solar 2010) autoři navrhli rozšíření kaskádového klasifikátoru do tvaru multi-objektového klasifikátoru s postupným vylepšováním výsledku v jednotlivých kaskádách („multiclass Coarse-To-Fine nested cascades“)(CTF). Na základě dosažených výsledků autoři prokázali, že jimi navržený systém je schopen adekvátně zpracovávat požadovanou úlohu s různým počtem zahnížděných tříd klasifikátoru, a to jak v trénovacích cyklech, tak i v ostrém provozu. V publikaci (Chevallier et al. 2013) je navržen velmi přesný a efektivní detektor rysů lidského obličeje, který je založený na kaskádě vzájemně se podporujících regresorů s libovolným počtem úrovní. Každý regresor je trénován separátně a počet regresorů je identický počtu klasifikovaných rysů obličeje. Navržená metoda používá regresi podpořenou gradientním algoritmem a při výpočtu jsou použity tři různé druhy jednoduchých elementárních regresorů. Každý regresor pracuje s tzv., Haarovými charakteristikami objektů (Haar-like features) (Wikipedia 2020c). Autoři prokázali, že jejich metoda je schopná směle konkurovat aktuální úrovni state-of-the-art s ohledem na přesnost,

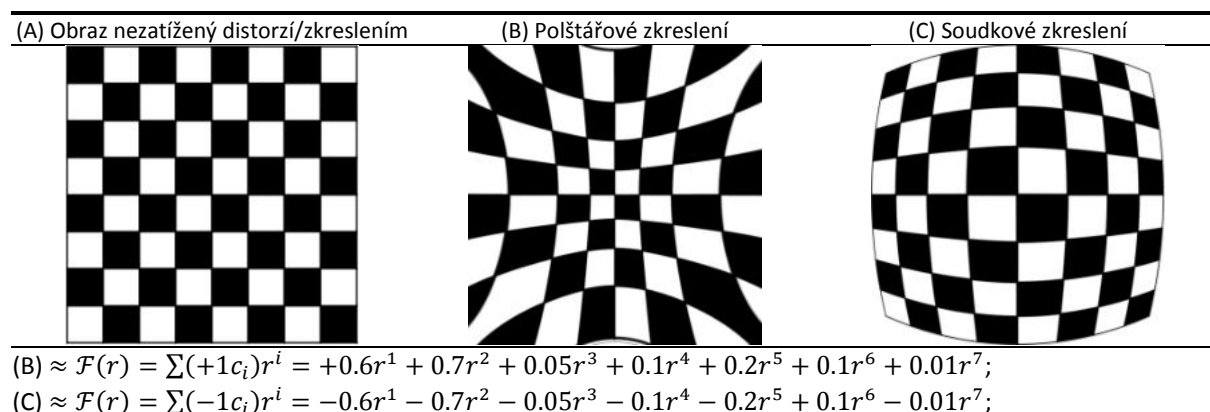
rychlost a robustnost. (Solar et al. 2010) navrhli pracovní rámec pro robustní a rychlou vizuální detekci u chodících robotů s kloubovými nohama (legged robots), která používá estimátory uspořádané do kaskád. Jako klasifikátor je použit algoritmus „Adaboost“ (Freund & Schapire 1995) společně se skupinou tzv. slabých „weak“ klasifikátorů viz (Freund & Schapire 1995). Použitý robot je AIBO. Navržený algoritmus má poměrně vysokou účinnost – téměř 90 procent. Autoři v publikaci (Susnijk et al. 2012) navrhli metodu, která je přímo aplikovatelná na již existující multi-objektové „boosting“ algoritmy pro sestavení kaskádových klasifikátorů. Metoda je pouze intuitivním rozšířením dříve navržené metody označované jako „Adaboost“ (Freund & Schapire 1995). Kaskádový estimátor slouží jen pro sestavení tzv. silného klasifikátoru na základě skupiny „weak“ klasifikátorů viz (Freund & Schapire 1995). Výhodou takového přístupu je podstatná redukce času při trénování klasifikátoru tím, že při trénování se klasifikátoru předkládají jen obtížné úlohy. Experimentální výsledky byly získány s využitím šesti různých datových množin a bylo jednoznačně prokázáno, že doba trénování a stejně tak i doba potřebná pro úspěšnou klasifikaci je výrazně nižší.

6.1.4 Algoritmus pro odstranění distorze obrazu využívající polynomiální reprezentaci

Grafický obraz I – viz Obr. (6.1).1A je obvykle definován jako množina prvků v E^2 s Euklidovskou metrikou ρ definovanou pro libovolné dva prvky resp. body neboli pixely $P_1(x_1, y_1) \in \mathbb{R}^2$ a $P_2(x_2, y_2) \in \mathbb{R}^2$ jako:

$$\rho = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (6.1.1)$$

a je výsledkem zachycení reálného stavu světa v určitý časový okamžik t .



Obrázek (6.1).1 Vliv centrického soudkového (C) a polštářového (B) zkreslení na obraz (A) nezatížený chybou zkreslení.

Rozměry obrazu I resp. šířka a výška jsou definovány jako $I_w \in \mathbb{N}^0, I_h \in \mathbb{N}^0$ a souřadnice x a y s ohledem na zavedený grafický souřadný systém v počítači nabývají hodnot $x \in \langle 0, I_w - 1 \rangle, y \in \langle 0, I_h - 1 \rangle$. Počátek souřadného systému je v počítači vlevo nahoře. Vodorovná osa je $+X$, svislá $+Y$ (shora dolů). Pro všechny následující výpočty předpokládáme, že vada soudkovitosti je čistě centrická. Střed nelineární deformace je ve středu obrazu I . Dále budeme značit I_D (distorted) jako obraz postižený distorzí a I_U (undistorted) jako obraz korigovaný a prostý vady soudkové distorze. Bod P_d obrazu I_D o souřadnicích x_d, y_d , který je zatížen vadou soudkovitosti označme jako $P_d(x_d, y_d)$. Střed deformace $P_{dc}(x_{dc}, y_{dc})$. Poloměr distorze u deformovaného obrazu I_D je pro bod P_d vyjádřen jako:

$$D(P_d, P_{dc}) = r_d = \sqrt{(x_d - x_{dc})^2 + (y_d - y_{dc})^2} \quad (6.1.2)$$

odpovídající úhel φ_d , který svírá přímka, na které leží body P_d, P_{dc} s poloosou $+X$ je pak dán vztahem:

$$\varphi_d = \arctan\left(\frac{(y_d - y_{dc})}{(x_d - x_{dc})}\right) \quad (6.1.3)$$

Bod P_u obrazu I_D o souřadnicích x_d, y_d , který je zbavený vady soudkovitosti bude značen $P_u(x_u, y_u)$ a odpovídající střed $P_{uc}(x_{uc}, y_{uc})$. Poloměr distorze u deformovaného obrazu I_D je pro bod P_u vyjádřen jako:

$$D(P_u, P_{uc}) = r_u = \sqrt{(x_u - x_{uc})^2 + (y_u - y_{uc})^2} \quad (6.1.4)$$

Odpovídající úhel φ_u , který svírá přímka, na které leží body P_u, P_{uc} s poloosou $+X$ je pak dán vztahem:

$$\varphi_u = \arctan\left(\frac{(y_u - y_{uc})}{(x_u - x_{uc})}\right) \quad (6.1.5)$$

Vzhledem k tomu, že vada distorze bude uvažována jako čistě centrická, platí, že $P_{uc} \equiv P_{dc}$. Tuto úvahu lze přijmou na základě skutečnosti, že snímací čip je umístěn nad fotografovanou oblastí tak, že optická osa čočkové soustavy (objektivu) protíná střed snímané oblasti a současně je kolmá na fotografovanou rovinu. Předpokládáme, že optická soustava takovou úvahu umožňuje. Definujme dále vektory V_d a V_u jako $V_d(x_d - x_{dc}, y_d - y_{dc})$ o velikosti r_d svírající s poloosou $+X$ úhel φ_d a $V_u(x_u - x_{uc}, y_u - y_{uc})$ o velikosti r_u svírající s poloosou $+X$ úhel φ_u . Pro uvažovaný zjednodušený případ čistě radiální funkce platí, že $\varphi_d = \varphi_u$. Cílem je najít vztah mezi vektory V_d a V_u resp. funkční transformaci, která převádí r_d na r_u .

$$\mathcal{F}: r_d \rightarrow r_u, \quad (6.1.6)$$

V ideálním případě hledáme takovou funkci f , aby platilo, že $f: \mathbb{R}^1 \rightarrow \mathbb{R}^1$. Poloha pixelu $P_u(x_u, y_u)$ s využitím znalosti r_u a φ_d je definována jako:

$$x_u = x_{uc} + r_u \sin \varphi_d \quad y_u = y_{uc} + r_u \cos \varphi_d \quad (6.1.7)$$

Vztah mezi V_d a V_u lze vyjádřit s využitím polynomiálního modelu PM (Brown 1965; Asari et al. 1999; Wang et al. 2009). Výpočet koeficientů PM je nejčastěji využívanou metodou modelování distorze obrazu. Rovnice, která aproximuje vztah mezi V_d a V_u je vyjádřena polynomem:

$$\mathcal{F}(r_d) \equiv r_u = \sum_{n=1}^N k_n r_d^n = k_1 r_d^1 + k_2 r_d^2 + k_3 r_d^3 + k_4 r_d^4 + \dots + k_N r_d^N \quad (6.1.8)$$

kde k_n jsou koeficienty polynomu. Koeficienty je možné získat např. využitím metody nejmenších čtverců (Asari et al. 1999). Pro malé hodnoty distorze obrazu je u kvalitních fotoaparátů postačující pouze první kvadratický člen $k_2 r_d^2$. k_2 nabývá velmi malých hodnot např. -1×10^{-8} - viz (Das & Patil 2006; Fraser & Al-Ajlouni 2006). Podle (Chen et al. 2011) je polynom inverzního perspektivního mapování možné uvažovat tak, že se použijí pouze členy u lichých mocnin nebo u sudých mocnin. Identickou informaci poskytuje i (Brown 1965). Je samozřejmě možné využít celý polynom. Pokud jsou použity pouze členy s lichým mocnitelem, pak je možné dosáhnout přesnost aproximace v průměru 97.13%, pokud jsou použity jen členy se sudými mocninami, pak je to jen 58.53% - viz (Chen et al. 2011). (Brown 1965) využívá mocninnou řadu s lichými mocniteli. Polynom pak lze zkrátit na:

$$\mathcal{F}_{odd}(r_d) \equiv r'_u = \sum_{n=2m+1, m \in \mathbb{N}^0}^{N_{odd}} k_n r_d^n = k_1 r_d^1 + k_3 r_d^3 + k_4 r_d^5 + \dots + k_{N_{odd}} r_d^{N_{odd}} \quad (6.1.9)$$

nebo v případě sudých mocnitelů na:

$$\mathcal{F}_{even}(r_d) = r'_u = \sum_{n=2m, m \in \mathbb{N}^+}^{N_{even}} k_n r_d^n = k_1 r_d^2 + k_3 r_d^4 + k_4 r_d^6 + \dots + k_{N_{even}} r_d^{N_{even}} \quad (6.1.10)$$

Pro odstranění distorze obrazu s využitím polynomu (6.1.8) existují dva dobře známé přístupy – viz ř.13,14 Alg. (6.1).4. První je tzv. direkt-mapping, který využívá přímý výpočet souřadnic I_U ze souřadnic I_D . Tato metoda je intuitivní a rychlá, ale ne všechny pixely obrazu I_U jsou při transformaci správně obarveny – viz (Cucchiara et al. 2003) a Obr. (6.1).2.

Algoritmus (6.1).4 Algoritmus korekce obrazu, direct-mapping a back-mapping

Input: I_D, I_U, I_w, I_h, C_f

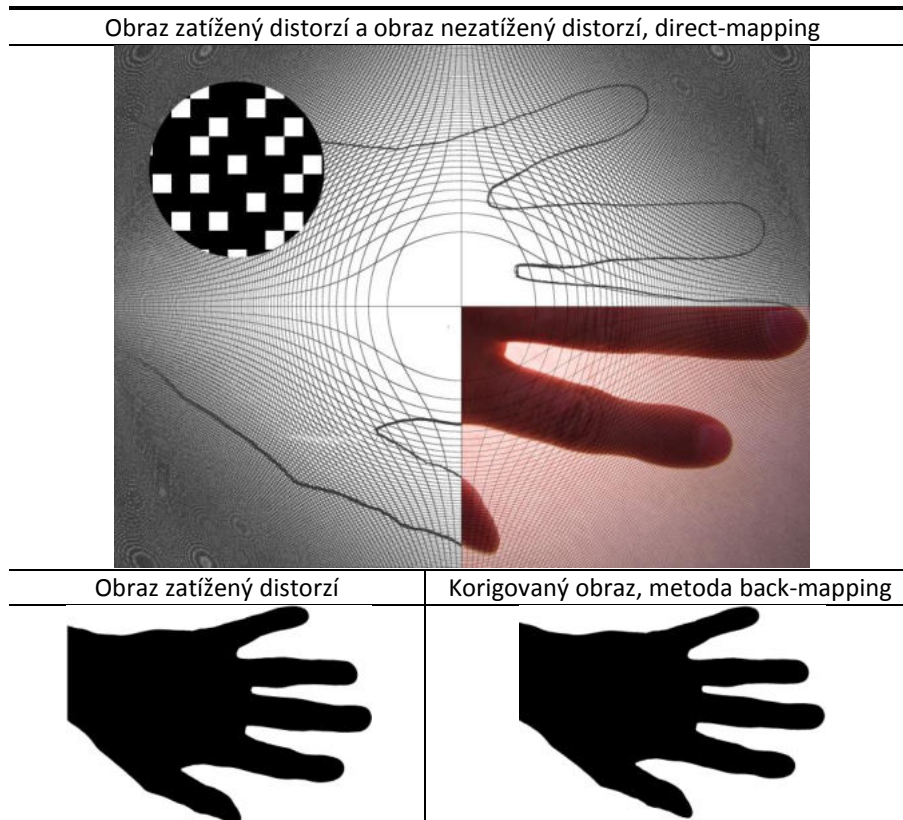
```

1  dividier =  $\min(I_w, I_h)$ ;   $\max(I_w, I_h)$  může být použito taktéž
2   $center_x = 0.5 \times (I_w - 1)$ ;   $center_y = 0.5 \times (I_h - 1)$ ;
3  for  $i=0$ ;  $i < I_w$ ;  $i=i+1$ 
4    for  $j=0$ ;  $j < I_h$ ;  $j=j+1$ 
5       $\Delta_x = \frac{(i-center_x)}{dividier}$ ;   $\Delta_y = \frac{(j-center_y)}{dividier}$ 
6       $R_d = \sqrt{\Delta_x^2 + \Delta_y^2}$ 
7       $R_s = \sum_{i=1}^{i=k} c_i \times R_d^i$ 
8       $factor = \frac{R_d}{R_s}$ 
9       $X_s = center_x + (factor \times \Delta_x \times dividier)$ 
10      $Y_s = center_y + (factor \times \Delta_y \times dividier)$ 
11      $X_s = interp(X_s, Y_s)$ ;   $Y_s = interp(X_s, Y_s)$ 
12     if  $X_s > 0 \wedge Y_s > 0 \wedge X_s < I_w \wedge Y_s < I_h$ 
13        $I_U(i, j) = I_D(X_s, Y_s)$  - BACK MAPPING
14        $I_U(X_s, Y_s) = I_D(i, j)$  - DIRECT MAPPING
15     endif
16   endfor
17 endfor

```

I_D -obraz zatížený distorzí; I_U -korigovaný obraz bez distorze; I_w -šířka obrazu; I_h -výška obrazu; C – vektor koeficientů vyrovnávacího polynomu; I_w, I_h -je obvykle počítáno od nuly. *interp* – pomocná interpolační funkce - je možné vynechat.

Barvy zbylých bodů je nutné dopočítat např. s využitím tzv. Kardanových polynomů (Cardan multinomial) (Cucchiara et al. 2003) nebo aproximovat např. metodou nejbližšího souseda (Ngo & Asari 2005), která však není použitelná vždy, protože jak je vidět na Obr. (6.1).2, chybějící pixely mohou tvořit rozsáhlé plochy. Mnohem horší jsou však kruhové artefakty viz Obr. (6.1).2 vpravo nahoře, které vypadají jako kruhy na vodní hladině po vhození např. kamene do vody. Zde už pak metoda aproximace nejbližšího souseda samozřejmě nepomůže.



Obrázek (6.1).2. Výsledek práce polynomiálního algoritmu. **Vlevo dole**–B&W obraz zatížený distorzí z biometrického scanneru; **Vpravo dole**–obraz po korekci soudkové distorze – metoda back-mapping. **Nahoře**–metoda direct-mapping po odstranění vady soudkové distorze. Obraz je ve formátu B&W. Je zakreslena kontura ruky. V pravém dolním kvadrantu je ponechán původní RGB obraz + zakreslené křivky (chybějící pixely) po odstranění distorze, kruh vlevo nahoře je pak zvětšenina oblasti B&W obrazu poblíž levého horního rohu.

Druhá metoda je označována jako back-mapping nebo také inverse-mapping (Asari et al. 1999; Shah & Aggarwal 1994). Výhodou tohoto přístupu je to, že díky inverznímu mapování souřadnic jsou vždy všechny pixely obrazu I_U obarveny tak, že tvoří spojitý obraz bez vynechaných pixelů jako je tomu na Obr. (6.1).2 – alespoň pro malé hodnoty zkreslení. Obraz může být pochopitelně zmenšen nebo zvětšen. Oba přístupy lze také vzájemně kombinovat.

6.1.5 Navržená metoda pro korekci obrazu

Navržená metoda je určena pro odstranění centrické vady soudkovitosti (radial distortion). Je možné ji využít i k odstranění polštářového zkreslení malého rozsahu. Umožňuje korekci obrazu získaného z biometrického scanneru kontury lidské ruky. Obraz je získáván pomocí klasického kompaktního fotoaparátu. Snímaný objekt – lidská ruka – je položen na projekční desce v tmavé komoře biometrického scanneru. Navržená metoda využívá dva velmi výkonné evoluční optimalizátory jDE a CMA spojené do kaskády. Možné kombinace a vnitřní uspořádání jednoho CEE jsou zobrazeny na Obr. (6.1).4, (6.1).5. Primární optimalizátor pracuje tak, že metodou inverzního inženýrství hledá polohu kalibračních značek v obraze, který je zatížený distorzí a který je současně metodou back-mappingu transformován na obraz bez dané aberace podle koeficientů předkládaných sekundárním optimalizátorem. Sekundární optimalizátor pak hledá odpovídající koeficienty polynomu (6.1.12), které zajistí co možná nejlepší shodu transformovaných značek s referenčními značkami, které reprezentují ideální obraz. Druhý EA je plně závislý na výsledku prvního EA. Přesnost kalibrace se určuje především pracovními parametry optimalizátoru jDE nebo CMA.

Popišme nyní navržený CEE podle Obr. (6.1).4, (6.1).5. Vzhledem k tomu, že zvolené optimalizátory je možné vzájemně zaměnit, jak bude diskutováno v sekci experimentálních výsledků, popíšeme detailně druhé uspořádání na Obr. (6.1).4 tj. CMA je sekundární optimalizátor - funkce $fitness_2$, jDE je primární optimalizátor - funkce $fitness_1$. Další dvě uspořádání jsou již pouze analogií zvolené sestavy.

Nejprve je vytvořen a vytisknut kalibrační výkres CD_A (36x30cm) s referenční sítí a pravítky – viz Obr. (6.1).3A, který umožní:

- 1) vycentrovat osu fotoaparátu v rovině XY na počátek souřadného systému (osa fotoaparátu musí být kolmá k ose fotografické desky) a zajistí správnou rotaci okolo osy Z.
- 2) nastavení rotace okolo os X a Y a výšku fotoaparátu nad fotografickou deskou – nad rovinou XY – viz kapitola Biometrický scanner.

Výkres je vložen mezi skleněné desky (resp. projekční desku) a je průběžně fotografován. Ustavení fotoaparátu se provádí ručně a musí být provedeno velmi precizně. Na výkresu CD_A jsou zobrazeny i kalibrační značky M_C , což jsou malé čtverce cca 20x20 pixelů v rastru 2x2cm v průsečících souřadné sítě – viz Obr. (6.1).3A. Fotoaparát je ručně ustaven tak, aby byly vidět vždy jen celé kalibrační značky a aby tvořili úplnou matici a nezasahovaly do okrajů fotografie. Podle počtu viditelných kalibračních značek M_C - např. 15x12, je vytvořen a vytisknut druhý kalibrační obraz CD_B – viz Obr. (6.1).3B, který zobrazuje již jen viditelné a stejně rozmístěné kalibrační značky v počtu např. 180 a je vložen na projekční desku. Obraz je vyfotografován – viz Obr. (6.1).3C, je zde dobře patrná vada soudkovitosti. Dále je ručně v počítači vytvořena síť referenčních značek M_R , které reprezentují ideální stav bez vady soudkové distorze. Hlavním parametrem pro vytvoření sítě M_R je offset první značky v ose X a Y a relativní rozestupy mezi značkami M_C . Tyto hodnoty jsou doměřeny ručně podle pravítek. Využito je především toho, že je znám bod, kudy prochází optická osa – tj. střed výkresu, velikost zorného pole, relativní rozestupy mezi značkami M_C v ose X a Y a offset počátku první značky od okraje výkresu – tyto informace jsou získány z výkresu CD_A .

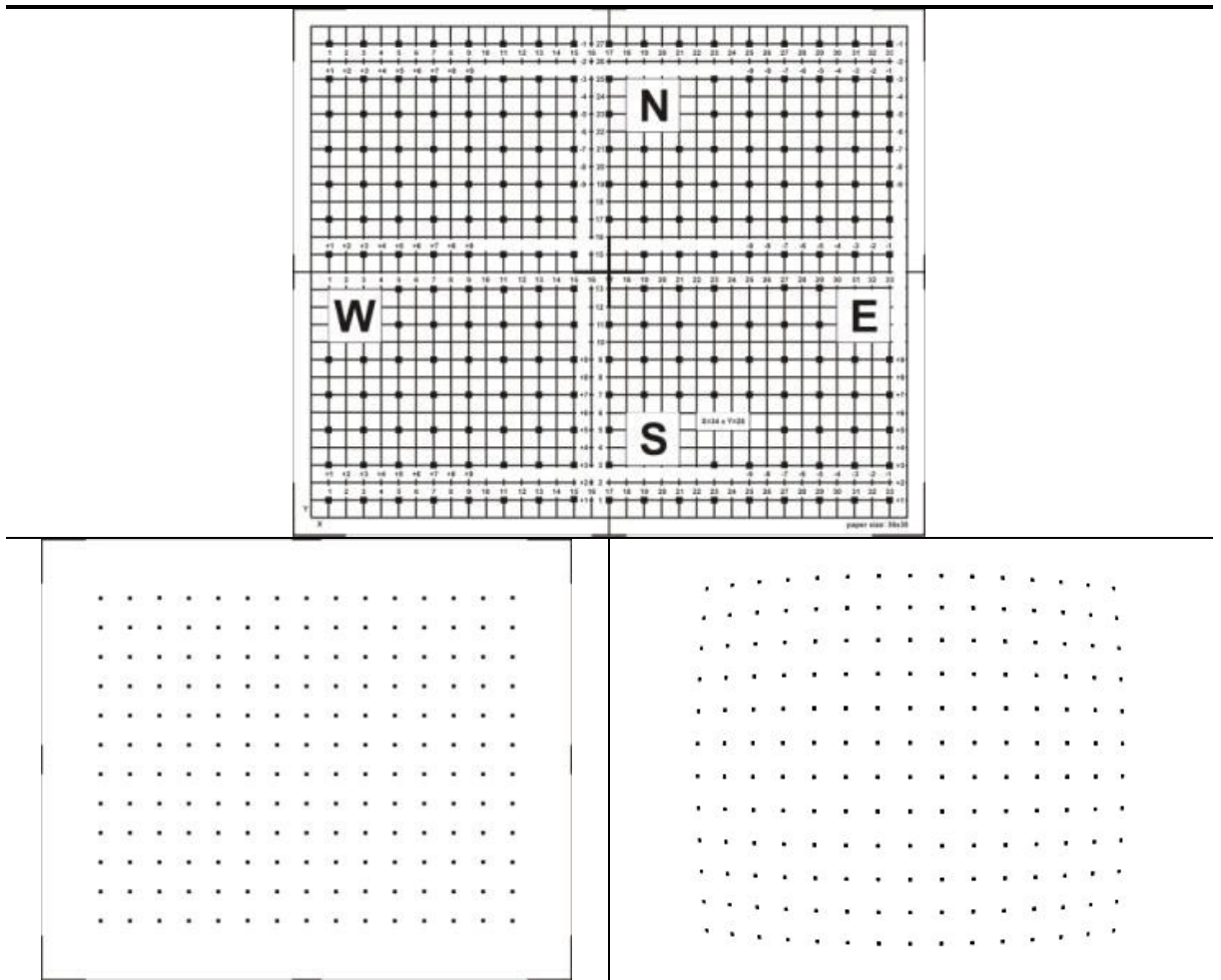
Vstupním údajem pro CEE je 24bppRGB obraz I_D'' zobrazující množinu kalibračních značek M_C , matice M_R referenčních značek, parametry algoritmu pro vyhledání kalibračních značek M_C a pracovní parametry použitých EA. Obraz I_D'' je v průběhu zpracování transformován na černobílý obraz I_D se stejným rozlišením. Rozlišení fotoaparátu a současně i rozlišení obrazu I_D'' je v našem případě 2592x1944x24bppRGB pixelů. Výstupem kalibrátoru je pak vektor C_f resp. koeficienty polynomu \mathcal{F} - viz (6.1.13) a matice M_F obsahující korektní pozice kalibračních značek v obraze I_U . V ideálním případě

je pak $\mathbf{M}_F \equiv \mathbf{M}_R$. Matice \mathbf{M}_R je naplněna celočíselnými hodnotami. Matice \mathbf{M}_C nesoucí pozice kalibračních značek pracuje s reálnými čísly a stejně tak i matice \mathbf{M}_F . Jednotlivé matice jsou definovány následovně:

$$\mathbf{M}_F = \begin{bmatrix} (\hat{x}, \hat{y})_{0,0} & \dots & (\hat{x}, \hat{y})_{0,j} \\ \dots & \dots & \dots \\ (\hat{x}, \hat{y})_{i,0} & \dots & (\hat{x}, \hat{y})_{i,j} \end{bmatrix} \quad \mathbf{M}_R = \begin{bmatrix} (x, y)_{0,0} & \dots & (x, y)_{0,j} \\ \dots & \dots & \dots \\ (x, y)_{i,0} & \dots & (x, y)_{i,j} \end{bmatrix} \quad (6.1.11)$$

$$\mathbf{M}_C = \begin{bmatrix} (\check{x}, \check{y})_{0,0} & \dots & (\check{x}, \check{y})_{0,j} \\ \dots & \dots & \dots \\ (\check{x}, \check{y})_{i,0} & \dots & (\check{x}, \check{y})_{i,j} \end{bmatrix} \quad \begin{aligned} \hat{x}, \hat{y} &\in \mathcal{H}_U; x, y \in \mathcal{H}_U; \check{x}, \check{y} \in \mathcal{H}_D \\ i &\in \langle 0, I_w - 1 \rangle, j \in \langle 0, I_h - 1 \rangle \\ x, y &\in \mathbb{N}^0; \hat{x}, \hat{y} \in \mathbb{R}^+; \check{x}, \check{y} \in \mathbb{R}^+ \end{aligned}$$

kde \mathcal{H}_U značí prostor souřadnic obrazu bez distorze (korigovaný obraz), \mathcal{H}_D – značí prostor souřadnic obrazu zatíženého distorzí. $(\check{x}, \check{y})_{0,0}$ jsou souřadnice první kalibrační značky na obraze zatíženém distorzí, $(\hat{x}, \hat{y})_{0,0}$ jsou souřadnice první kalibrační značky v korigovaném obraze bez distorze a $(x, y)_{0,0}$ jsou souřadnice první referenční značky v korigovaném obraze bez distorze.



Obrázek (6.1).3. Kalibrační obrazy. (A) Nahoře-kalibrační obraz CD_A o rozměrech 36x30cm s pravítky a kalibračními značkami. Kalibrační značky jsou malé čtverce (20x20pixelů), v rastru 2x2cm v průsečících sítě. (B) Vlevo dole – kalibrační obraz CD_B (výkres) připravený pro odstranění soudkové distorze, 15×12 kalibračních značek. (C) Vpravo dole – vyfotografovaný kalibrační obraz CD_B (B), tak jak jej vidí použitý fotoaparát včetně soudkové distorze.

Aby byl algoritmus pro odstranění distorze – viz Alg. (6.1).6 dostatečně výkonný, je polynom \mathcal{F} volen stupně sedm:

$$\mathcal{F}(r_d) = r_u = \sum_{n=1}^{n=7} k_n r_d^n \quad (6.1.12)$$

Hodnotu max. stupně polynomu \mathcal{F} je možné zvětšit podle možností optimalizátoru CMA a pracovních parametrů použitých EAs. Při výpočtu jsou použity jak sudé, tak i liché stupně polynomu. Je však ponechána možnost nezahrnout jeden nebo více stupňů polynomu do výpočtu nebo využít pouze liché členy polynomu, podle rozhodnutí operátora. RGB obraz z fotoaparátu s šířkou I_w a výškou I_h je nejprve převeden na šedotónový (grayscale) obraz a současně na B&W obraz pomocí prahování – viz schéma Obr. (6.1).6 a ř.1,2 Alg. (6.1).6. K vylepšení kvality obrazu je možné provést vhodnou filtraci B&W obrazu. V černobílém bitmapovém obrazu I_D mohou být nyní nalezeny kalibrační značky M_C . Algoritmus 5 obsahuje popis mechanismu pro nalezení kalibračních značek. Je využívána čtvercová bitmapová maska A_S velikosti např. $A_w \times A_h = 40 \times 40$ pixelů, která se posouvá pixel po pixelu po celém obrazu v rozsahu souřadnic $X: \langle 0, I_w - A_w \rangle$; $Y: \langle 0, I_h - A_h \rangle$ zleva doprava a shora dolů. Kalibrační značka je nalezena, pokud na obvodu masky A_S jsou rozmístěny pouze bílé pixely a uvnitř masky je nalezen alespoň předem definovaný minimální počet černých pixelů – např. 190. Do matice M_C se pak zapíše souřadnice těžiště (\bar{x}, \bar{y}) množiny nalezených černých pixelů, která reprezentuje k -tou kalibrační značku. Podmínkou dobrého výsledku je to, že kalibrační značky na okrajích obrazu I_D se následkem distorze nepřekrývají a je mezi nimi mezera v pixelech alespoň o velikosti použité masky A_S . Podobný způsob práce s bitmapovým obrazem je možné nalézt i v (Wang et al. 2009), kde autoři pro detekci kruhových objektů v obraze zatíženém distorzí využili bitmapový filtr.

Algoritmus (6.1).5 Způsob nalezení kalibračních značek M_C

Input: I_D

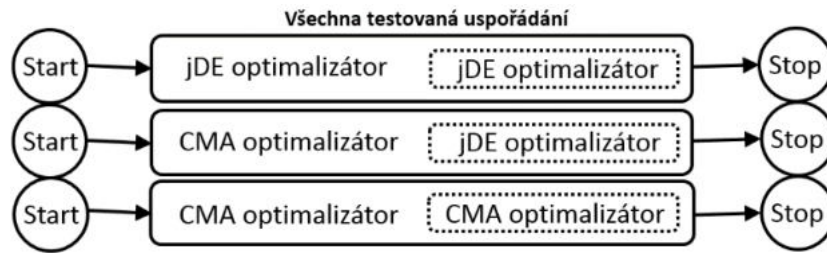
```

1 Inicializace masky  $A_S$  o velikosti  $A_w \times A_h$ ;  $k = 0$ 
2 for  $i=0$ ;  $i < I_w - A_w$ ;  $i=i+1$ 
3   for  $j=0$ ;  $j < I_h - A_h$ ;  $j=j+1$ 
4     Přesun  $A_S$  na pozici  $i, j$ ; (počáteční bod  $A_S$  je lokální pozice 0,0)
5     R = počet černých pixelů na čtyřech stranách  $A_S$ 
6     if  $R = 0$ 
7       R = počet černých pixelů uvnitř masky  $A_S$ 
8       if  $R > \text{limit}$ 
9         Vypočítej těžiště  $k$ -tého shluku bodů
10        Ulož  $k$ -tou kalibrační masku do  $M_C$ ;  $k = k + 1$ 
11      endif
12    endif
13  enfor
14 enfor

```

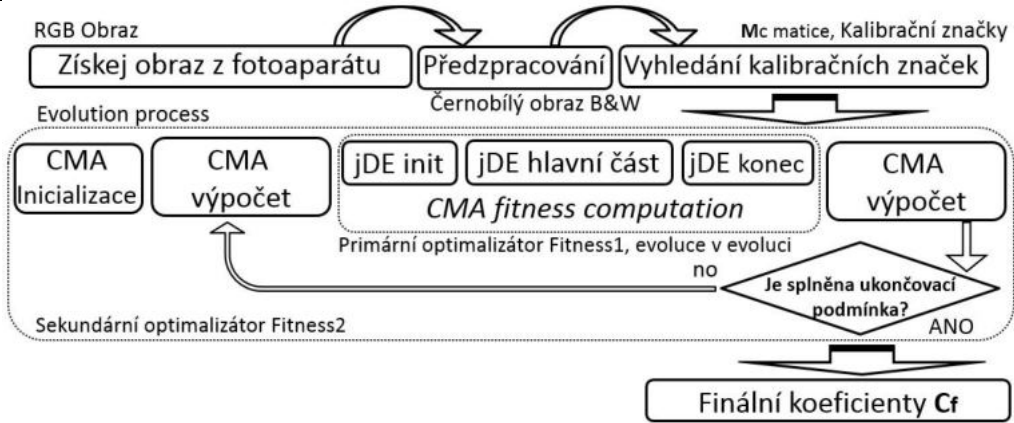
A_S -je čtvercová maska t.j. $A_w = A_h$.

Algoritmus pro odhad hodnot C_f je rozdělen na 4 části. Alg. (6.1).5 popisuje způsob nalezení kalibračních značek. Alg. (6.1).6 popisuje hlavní program. Alg. (6.1).7 výpočet $fitness_1$ funkce a Alg. (6.1).8 způsob výpočtu $fitness_2$ funkce. Z důvodu lepší numerické vyčíslitelnosti a stability testovaných EA při praktické realizaci v jazyce C++/CLI je použit tzv. *divider* – viz ř. 6,10,11 Alg. (6.1).7. Pokud je tento koeficient vynechán, pak se hodnoty korekčních koeficientů c_i vektoru C_f běžně pohybují v řádech 1×10^{-10} .

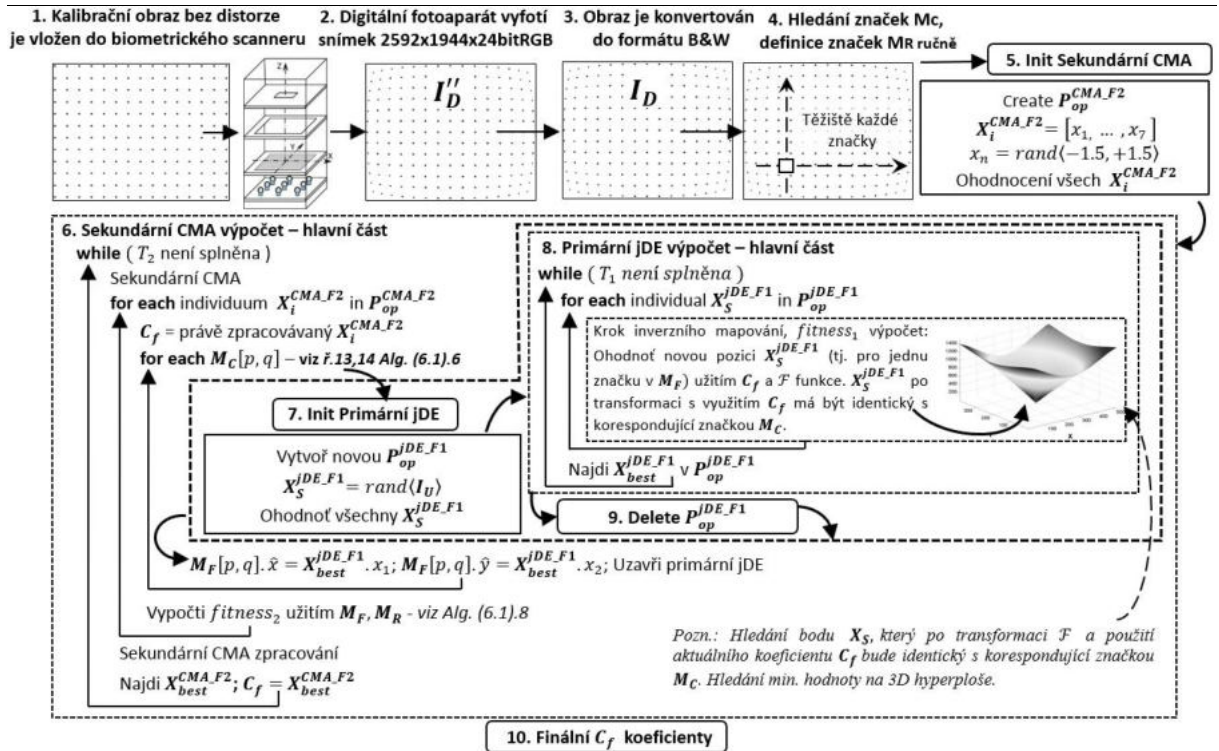


Obrázek (6.1).4 Schematický diagram zvolených a testovaných kombinací CEE.

(A) Základní schéma autokalibračního procesu, optimalizátory jDE a CMA



(B) Detailní schéma autokalibračního procesu, optimalizátory jDE a CMA



Obrázek (6.1).5 Schematický diagram auto-kalibračního procesu, viz také Alg. (6.1).6. (A) – základní schéma, (B) – detailní schéma s rozepsanými jednotlivými kroky algoritmu kaskádového estimátoru. Algoritmizace provedena v MS VS 2008 SP1 C++/CLI x32, cca 15272 řádek ručně psaného kódu.

Získané souřadnice kalibračních značek \mathbf{M}_C , reprezentované těžištěm každé kalibrační značky \mathbf{M}_C , jsou využity pro výpočet tzv. primární účelové funkce značené jako $fitness_1$ v algoritmu jDE. Po inicializaci populace jedinců \mathbf{P}_{op}^{CMA} je spuštěn optimalizátor CMA – ř. 6 Alg. (6.1).6. Hodnoty \mathbf{X}_i^{CMA} , x_j jsou při vytváření populace naplněny náhodnými čísly, normální rozdělení: $x_j \sim N(-1.5; +1.5)$. Na ř. 9 Alg. (6.1).6 je uvedena omezovací podmínka, která zajistí, aby jedinci populace byly vždy v požadovaném prostoru. Tento krok však komplexně narušuje proces konvergence – nepříznivě ovlivňuje proces adaptace kovarianční matice $\hat{\mathbf{C}}$. Při experimentech nebyl použit. Naopak byla využita výrazná penalizace při překročení hranice u $fitness_1$ viz ř. 4 Alg. (6.1).7 a také (6.1.16). CMA při kalibraci optimalizuje až 7-mi dimenzionální účelovou funkci $fitness_2$ zde označenou jako sekundární účelová funkce (objective function), která je vyjádřena vztahem:

$$(\hat{\mathbf{C}}_f) = \arg \operatorname{opt}_{(\mathbf{M}_F, \mathbf{M}_R) \in \mathcal{H}_U} \mathcal{F}_{CMA}(\mathbf{M}_F, \mathbf{M}_R) \quad (6.1.13)$$

a hledá finální ‘optimální’ množinu koeficientů $\hat{\mathbf{C}}_f = [c_i | c_i \in \langle -1.5; +1.5 \rangle]$, s ohledem na $\mathbf{M}_F, \mathbf{M}_R$. \mathcal{F}_{CMA} značí optimalizační proces – funkci CMA. Účelová funkce $fitness_2$ je vypočítávána jako:

$$fitness_2 = \|\mathbf{M}_F - \mathbf{M}_R\|^2 \quad (6.1.14)$$

viz Alg. (6.1).8. Kompletní matice \mathbf{M}_F , která zohledňuje parametry \mathbf{C}_f vždy pro jeden vektor \mathbf{X}_i^{CMA} je k dispozici na ř. 42 Alg. (6.1).6, po ukončení chodu optimalizátoru jDE. Řádky 11–43 Alg. (6.1).6 představují primární EA, pro který byl zvolen optimalizátor jDE. V cyklech na ř. 13, 14–40, 41 je hledána odpovídající pozice každé kalibrační značky \mathbf{M}_C s hodnotou \mathbf{C}_f neboli hodnotou právě zpracovávaného chromozomu jedince \mathbf{X}_i^{CMA} algoritmem CMA. Prostor souřadnic digitálního obrazu \mathbf{I}_U prostého vady soudkovitosti lze popsat jako $\mathcal{H}_U = \mathbf{I}_w \times \mathbf{I}_h$, $\mathbf{I}_w, \mathbf{I}_h \in \mathbb{N}^0$. V našem případě jsou rozměry obrazů \mathbf{I}_D a \mathbf{I}_U stejné. Libovolný bod $P(\hat{x}, \hat{y})$ obrazu \mathbf{I}_U nabývá možných souřadnic $\hat{x}, \hat{x} \in \langle 0, \mathbf{I}_w - 1 \rangle$, $\hat{y}, \hat{y} \in \langle 0, \mathbf{I}_h - 1 \rangle$ – s ohledem na prostředky při počítačovém zpracování. Algoritmus jDE hledá takové souřadnice (\hat{x}, \hat{y}) v prostoru \mathcal{H}_U obrazu \mathbf{I}_U , jejichž transformací při konstantním \mathbf{C}_f resp. pro jednoho jedince \mathbf{X}_i^{CMA} z populace \mathbf{P}_{op}^{CMA} , získáme pozice (\ddot{x}, \ddot{y}) kalibračních značek \mathbf{M}_C v obrazu \mathbf{I}_D . jDE optimalizuje dvoudimenzionální účelovou funkci v prostoru souřadnic \mathbf{I}_U a pozice (\hat{x}, \hat{y}) v obraze \mathbf{I}_D , která koresponduje se značkou v obraze \mathbf{I}_D je vyjádřena jako:

$$(\hat{x}, \hat{y}) = \arg \operatorname{opt}_{\substack{(t_x, t_y) \in \mathcal{H}_U \\ \hat{x}, \hat{y} \in \mathbf{I}_U, x, y \in \mathbf{I}_D}} \mathcal{F}_{jDE}(t_x, t_y, P_W(x, y), \mathbf{C}_f), \quad (6.1.15)$$

kde $\mathbf{C}_f, P_W(x, y), t_x, t_y \in \mathbb{R}^+$. Primární účelová funkce neboli $fitness_1$ je vypočítávána podle vzorce:

$$fitness_1 = \begin{cases} (\|(t_x, t_y), center(x, y)\|^2)^4 & \text{if } (t_x, t_y) \notin \mathbf{I}_U \\ \|P(X_S, Y_S) - P_W(x, y)\|^2 & \text{if } (t_x, t_y) \in \mathbf{I}_U \end{cases} \quad (6.1.16)$$

$center(x, y)$ značí souřadnice středu obrazu \mathbf{I}_U - viz Alg. (6.1).7, bod $P(X_S, Y_S)$ reprezentuje aktuální odhad souřadnic $P_W(x, y)$ v k -té generaci optimalizátoru jDE pro jednoho jedince \mathbf{X}_s^{jDE} z populace \mathbf{P}_{op}^{jDE} , (t_x, t_y) je pak chromozome \mathbf{X}_s^{jDE} nebo \mathbf{X}_t^{jDE} . \mathcal{F}_{jDE} značí optimalizační mechanismus jDE, pro který de-facto platí, že $\mathcal{F}_{jDE}: \mathbf{M}_F \rightarrow \mathbf{M}_C$. Protože je použita metoda back-mappingu, pak na souřadnici $P(\hat{x}, \hat{y})$ v obraze \mathbf{I}_U je zpětně přenesen pixel z obrazu \mathbf{I}_D ze souřadnic $P(\ddot{x}, \ddot{y})$ neboli ze souřadnic $P(X_S, Y_S)$ - viz Alg. (6.1).7, získaných polynomiální transformací s parametry \mathbf{C}_f . Platí, že

$(X_S, Y_S) \in \mathbb{R}^+$, aby bylo možné dosáhnout co možná největší přesnosti. Použití hrubé síly je v tomto případě výpočetně naprosto nereálné.

Při výpočtu je chromozom jedinců X_s^{JDE} kódován reálnými čísly $X_s^{JDE}: [x_j | x_j \in \mathbb{R}, j \in \langle 0, D_{im} \rangle, F \in \mathbb{R}, P_{cr} \in \mathbb{R}]$, aby výsledná odchylka neboli hodnota $fitness_1$ byla co možná nejmenší. Ukončovací podmínky T_1 a T_2 jsou pro optimalizátor jDE nastaveny vždy na pevný počet generací G_{en}^{JDE} . Zdůvodnění této volby je v odstavci ‘experimentální výsledky’. U optimalizátoru CMA je pro T_1 volena kombinovaná ukončovací podmínka a pro T_2 je volen pevný počet generací G_{en}^{CMA} . Kombinovaná ukončovací podmínka má následující tvar:

```
if  $fitness_1 \leq limit$  v  $actualG_{en}^{CMA} \geq G_{en}^{CMA}$   
    ukonči optimalizátor; tj. podmínka  $T_1$  je splněna  
else  
    continue;
```

Hodnota $limit$ je nastavena na $\sim 1 \times 10^{-4}$, $actualG_{en}^{CMA}$ udává aktuální generaci při evoluci CMA.

Algoritmus (6.1).6 Navržený kaskádový evoluční kalibrátor obrazu, typ jDE+CMA

Input: I_D, M_R, T_2, T_1 , jDE a CMA parametry, A_S maska

Output: C_f

```

1 Získej obraz  $I_D'$  z fotoaparátu
2 Konverze  $I_D'$  na šedotónovou reprezentaci + prahování;  $I_D' \rightarrow I_D$ 
3 Vyhledání  $M_C$  v  $I_D'$ 
4 Vytvoř  $P_{op}^{CMA}$  + ohodnocení  $X_i^{CMA}$ 
5 Inicializace CMA pracovních parametrů +  $\hat{C}, B, D$  matic
6 while (  $T_2$  není splněna )
7   for  $i=0$ ;  $i < N^{CMA}$ ;  $i=i+1$ 
8     Vytvoř nové řešení  $X_i^{CMA}$  (neboli 'lambda' jedinec)
9     Aplikuj omezení s ohledem na  $D_{om}$ , pokud je to potřeba
10  endfor
11  for  $i=0$ ;  $i < N^{CMA}$ ;  $i=i+1$ ; tj. vyčíslí  $X_i^{CMA}$  s využitím jDE
12    Inicializuj jDE vstupy:  $M_C, X_i^{CMA} \equiv (c_1, \dots, c_7) = C_f, T_1$ 
13    for  $p=0$ ;  $p < M_{CX}$ ;  $p=p+1$ 
14      for  $q=0$ ;  $q < M_{CY}$ ;  $q=q+1$ 
15         $P_W.x = M_C[p, q].\ddot{x}$ ;  $P_W.y = M_C[p, q].\ddot{y}$ 
16        Inicializace jDE pracovních parametrů
17        Vytvoř  $P_{op}^{jDE}$ 
18        for  $s=0$ ;  $s < N^{jDE}$ ;  $s=s+1$ 
19           $X_S^{jDE}.x_1 = rand(0, l_w)$ ;  $X_S^{jDE}.x_2 = rand(0, l_h)$ 
20           $X_S^{jDE}.F = rand(0,1)$ ;  $X_S^{jDE}.P_{cr} = rand(0,1)$ 
21        endfor
22        while (  $T_1$  není splněna)
23          for  $s=0$ ;  $s < N^{jDE}$ ;  $s=s+1$ 
24             $F_{old} = X_S^{jDE}.F$ ;  $P_{cr-old} = X_S^{jDE}.P_{cr}$ 
25            Vypočítej nové  $X_S^{jDE}.F$ ;  $X_S^{jDE}.P_{cr}$ 
26            Vyber  $\{X_{r1}, X_{r2}, \dots, X_{rk}\}, rj \neq s, X_{rj} \in P_{op}^{jDE}$ 
27            Vypočítej  $X_t$  užitím  $V_p, X_S^{jDE}, X_{rj}, X_{best}^{jDE}$ 
28             $R_1 = fitness_1(X_S^{jDE}.x_1, X_S^{jDE}.x_2)$ 
29             $R_2 = fitness_1(X_t.x_1, X_t.x_2)$ 
30            if  $R_1 < R_2$ 
31               $X_S^{jDE}.F = F_{old}$ ;  $X_S^{jDE}.P_{cr} = P_{cr-old}$ 
32            else
33               $\forall j, j \in \langle 0, D_{im} \rangle$ :  $X_S^{jDE}.x_j = X_t.x_j$ 
34            endif
35          endfor
36          Find  $X_{best}^{jDE}$  in  $P_{op}^{jDE}$ 
37        endwhile
38         $M_F[p, q].x_1 = X_{best}^{jDE}.x_1$ ;  $M_F[p, q].x_2 = X_{best}^{jDE}.x_2$ 
39        Delete  $P_{op}^{jDE}$ 
40      endfor
41    endfor
42    Vypočítej  $fitness_2$  užitím  $M_F, M_R$  - viz Alg. (6.1).8
43  Endfor
44   $P_{op}^{CMA}$  třídění, nejlepší fitness první
45  Vypočítej vážený průměr
46  Aktualizace evoluční cesty
47  Adaptace kovarianční matice  $\hat{C}$ 
48  Vypočítej parametr  $\sigma$ 
49  Dekompozice  $\hat{C}$  do formy  $B \times diag(D.^2) \times B^T$ 
50  Postprocessing
51  Najdi  $X_{best}^{CMA}$ 
52 endwhile
53 Finální výsledek  $C_f = X_{best}^{CMA}$ 

```

M_F -matice korigovaných (bez distorze) pozic x, y kalibračních značek M_C ; M_{CX}, M_{CY} -velikost matice M_F ; M_R -matice pozic x, y referenčních značek; M_{RX}, M_{RY} -velikost matice M_R .

Algoritmus (6.1).7 Výpočet primární $fitness_1$, je využit jDE optimalizátor

Vstup: t_x, t_y Pomocný vstup: $I_w, I_h, c_i, P_w \cdot x, P_w \cdot y$

```
1 dividier = max( $I_w, I_h$ )
2  $center_x = 0.5 \times (I_w - 1)$ ;  $center_y = 0.5 \times (I_h - 1)$ ;
3 if  $t_x < 0 \vee t_x \geq I_w \vee t_y < 0 \vee t_y \geq I_h$ 
4    $fitness_1 = \left[ \sqrt{(t_x - center_x)^2 + (t_y - center_y)^2} \right]^4$ 
5 else
6    $\Delta_x = \frac{(t_x - center_x)}{dividier}$ ;  $\Delta_y = \frac{(t_y - center_y)}{dividier}$ 
7    $R_d = \sqrt{\Delta_x^2 + \Delta_y^2}$ 
8    $R_s = \sum_{i=1}^{i=k} [c_i \times (R_d)^i]$ 
9    $factor = \frac{R_d}{R_s}$ 
10   $X_s = center_x + (factor \times \Delta_x \times dividier)$ 
11   $Y_s = center_y + (factor \times \Delta_y \times dividier)$ 
12   $fitness_1 = \sqrt{(X_s - P_w \cdot x)^2 + (Y_s - P_w \cdot y)^2}$ 
13 endif
```

I_D –obraz zatížený distorzi; I_U –korigovaný obraz; I_w –šířka obrazu; I_h –výška obrazu; C_f –vektor koeficientů vyrovnávacího polynomu; I_w, I_h –hodnoty jsou počítány od nuly.

Algoritmus (6.1).8 Výpočet sekundární $fitness_2$ funkce pomocí CMA algoritmu

Vstup: M_F, M_{FX}, M_{FY}, M_R

```
1  $fitness_2 = 0$ 
2 for  $i=0$ ;  $i < M_{FX}$ ;  $i=i+1$ 
3   for  $j=0$ ;  $j < M_{FY}$ ;  $j=j+1$ 
4      $fitness_2 = fitness_2 + \sqrt{(M_F[i, j] \cdot x_1 - M_R[i, j] \cdot x_1)^2 + (M_F[i, j] \cdot x_2 - M_R[i, j] \cdot x_2)^2}$ 
5   enfor
6 enfor
```

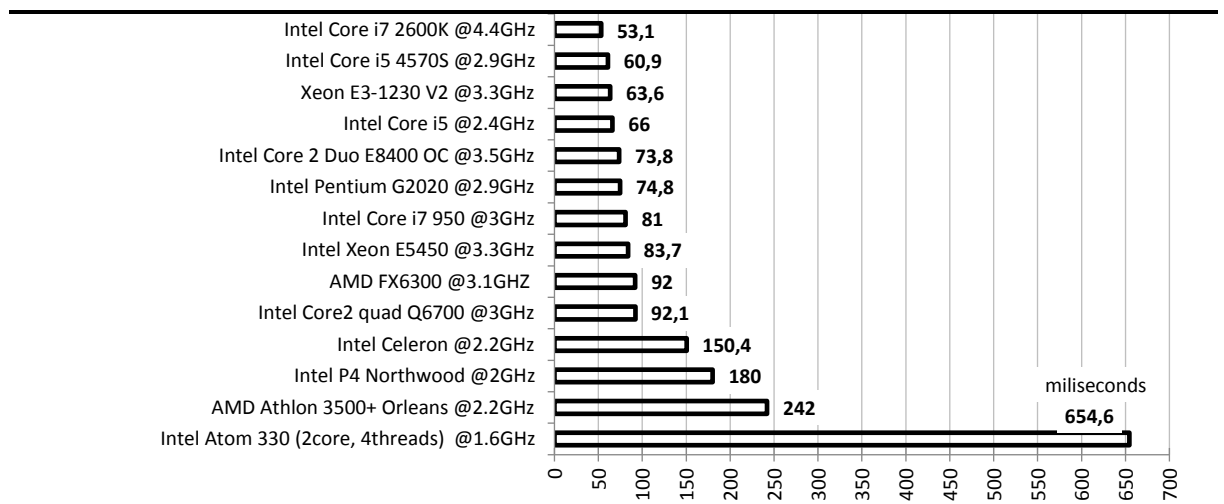
M_F –matice pozic \hat{x}, \hat{y} v obraze, který je nezatížený vadou distorze, jedná se o pozice kalibračních značek M_C ; M_R –matice pozic x, y referenčních značek.

6.1.6 Experimentální výsledky

Odstavec experimentálních výsledků je rozdělen na **A.** část popisující časové nároky na odstranění vady distorze při známých koeficientech polynomu (6.1.12), **B.** charakteristiky nelineární dvoudimenzionální $fitness_1$ funkce, **C.** analýza výhodnosti vybraných EA jako optimalizátoru v úloze korekce distorze a konečně **D.** experimentální ověření tří druhů navržených CEE.

6.1.6.1 Časová náročnost korekce obrazu

Na Obr. (6.1).6 jsou uvedeny časové nároky pro korekci soudkové distorze jednoho obrazu v rozlišení 2592x1944x24bppRGB při použití polynomu (6.1.12) stupně 7 a při známých koeficientech c_i . Podle (Chen et al. 2011) je pro takový účel vhodné použít PLA. S ohledem na pokrok v oblasti mikroprocesorů (2020) lze konstatovat, že u špičkových komerčních procesorů je možné dosáhnout stabilní rychlosti konverze na jedno μC jádro méně než 50ms/snímek/procesorové jádro, tj. 15-20 snímků za sekundu. Tato rychlost je již dostačující pro převod videa alespoň u jednodušších WEB kamer. Pro účely korekce obrazu u biometrického scanneru jsou tedy dostačující i starší typy procesorů. Nejdůležitější je však rychlost, s jakou je možné nalézt správné koeficienty c_i polynomu (6.1.12). Pro takový úkol je již jednoznačně nutné zařízení, jako je např. již zmíněné PLA nebo kombinace CPU+GPU (Melo et al. 2012; Jeught et al. 2012).

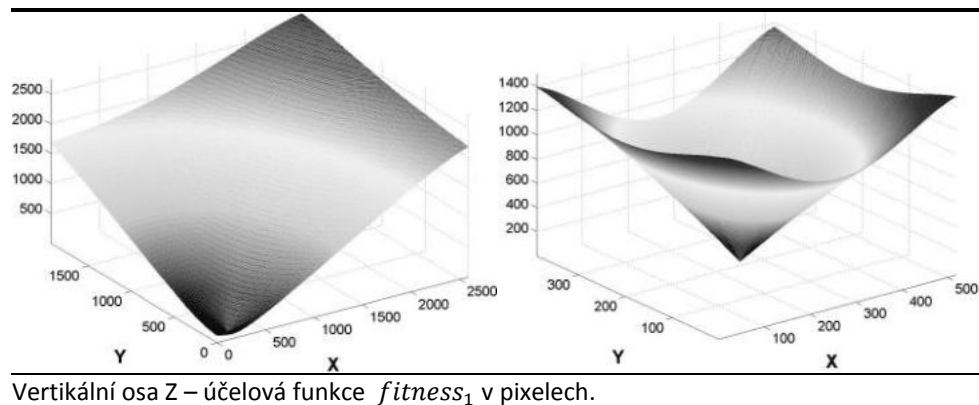


Obrázek (6.1).6 Časové nároky na korekci soudkové distorze obrazu na různých typech procesorů, rozměry obrazu 2592x1944x24bppRGB. Polynom stupně 7. Jedno-vláknová 32bit aplikace. Algoritmizace v MS VS 2008 SP1 C/C++ x32, WinAPI.

6.1.6.2 Nelineární dvoudimenzionální účelová funkce

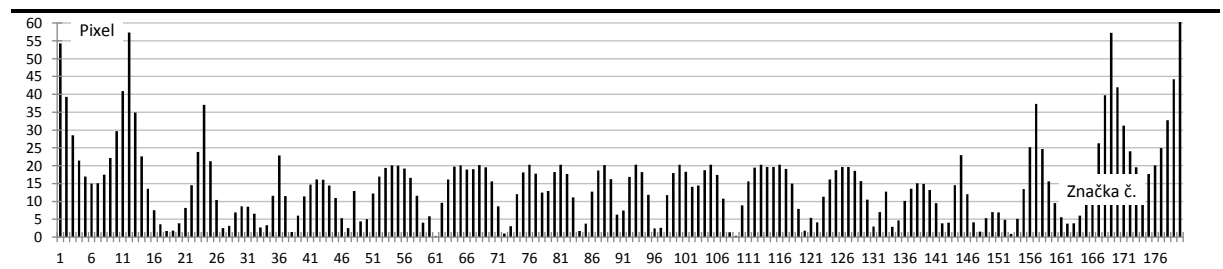
Optimalizace primární $fitness_1$ představuje nalezení optima dvoudimenzionální funkce, která je spojitá, unimodální, neseperabilní a nelineární. Na Obr. (6.1).7 jsou zobrazené některé tvary hyperploch pro účelovou funkci $fitness_1$. Reálná pozice minima se posunuje v pomyslné oblasti I_U podle právě zvolené značky M_C . Pro jednu značku jsou optimalizátorem jDE hledání takový jedinci X_S^{jDE} resp. hodnoty (t_x, t_y) - viz Alg. (6.1).7 a ř. 29 Alg. (6.1).6, kteří se co možná nejpřesněji ztotožní s pozicí (\bar{x}, \bar{y}) . Jedinci X_S^{jDE} populace P_{op}^{jDE} jsou kódováni reálnými čísly. Podle volby bodu M_C se pochopitelně mění i tvar hyperplochy. Primární $fitness_1$ funkce nabývá hledané 'malé' hodnoty ($\sim 1 \times 10^{-4}$) pouze tehdy, když optimalizátor pracuje s ideální sadou koeficientů C_f polynomu (6.1.12) resp. právě tehdy, když sekundární optimalizátor již našel optimální sadu hodnot C_f . Např. na Obr. (6.1).7 je dosažená hodnota $fitness_1$ pro obrázek vlevo 1.884 pixelu, pro obrázek vpravo pak 0.834

pixelu; jDE optimalizátor pracoval s hodnotami $P_{op}^{jDE} = 20$; $G_{en}^{jDE} = 40$. Pozice značky byla odhadnuta přibližně, ale pro běžné lidské oko s dostatečnou přesností.



Obrázek (6.1).7 3D zobrazení tvarů primární $fitness_1$ funkce pro: Vlevo-kalibrační značku M_C na pozici $(x = 265, y = 155)$ a Vpravo-značku na pozici $(x = 1312, y = 1062)$.

Pro každý kalibrační bod M_C je pochopitelně klasifikační hyperplocha jiná. To samozřejmě nijak nezjednodušuje klasifikaci. Primární účelová funkce $fitness_1$ je pouze dvoudimenzionální, prostor souřadnic je dán rozměrem obrazu I_U . S ohledem na možné hodnoty C_f dochází ve velkém počtu případech k tomu, že souřadnice (X_S, Y_S) padnou mimo oblast I_U . V tom případě je od výpočtu upuštěno a hodnota $fitness_1$ je nahrazena penalizační funkcí – viz ř. 4 Alg. (6.1).7. Pro optimalizaci byl použit C++/CLI, kde numerická přesnost při opakovaném výpočtu je cca 0.00001 - viz (Betelgeuse 2020; Kimichi 2020; Dawson 2020; SunMicrosystems 2020). Na Obr. (6.1).9 je zobrazen graf, který udává vzdálenost jednotlivých bodů (t_x, t_y) vůči korespondujícím značkám M_R v k -tém kroku optimalizace. Každý sloupec grafu představuje jeden běh jDE – resp. úplnou konvergenci při $P_{op}^{jDE} = 20$; $G_{en}^{jDE} = 40$. Součet je 2712.211 pixelů. Tato hodnota je vložena do sekundární $fitness_2$ funkce příslušného jedince X_i^{CMA} .

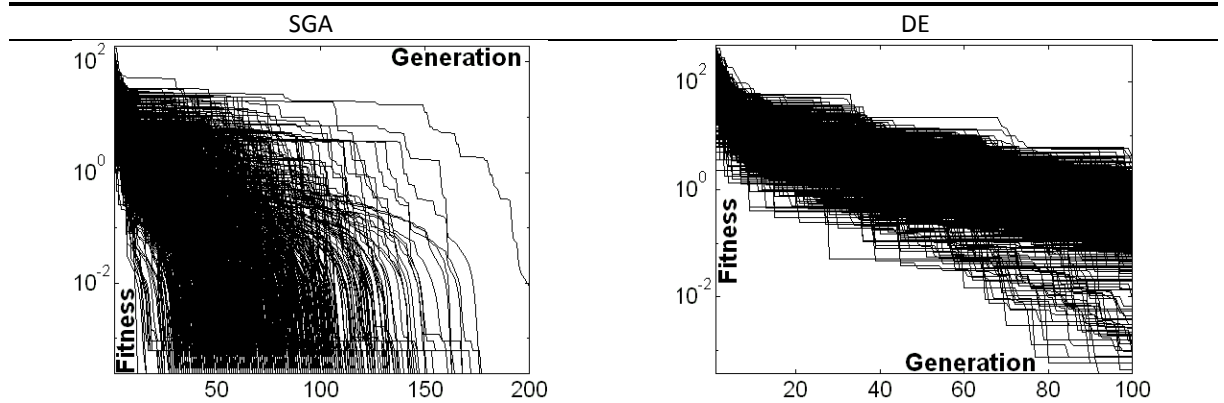


Obrázek (6.1).8 Optimalizátor jDE. Nejlepší nalezená odchylka – vzdálenost – pro jednotlivé body (t_x, t_y) - viz Alg. (6.1).7, vůči referenčním značkám M_R pro jednu sadu C_f resp. pro jednoho náhodného jedince X_i^{CMA} z P_{op}^{CMA} .

6.1.6.3 jDE a CMA vs ostatní metaheuristiky

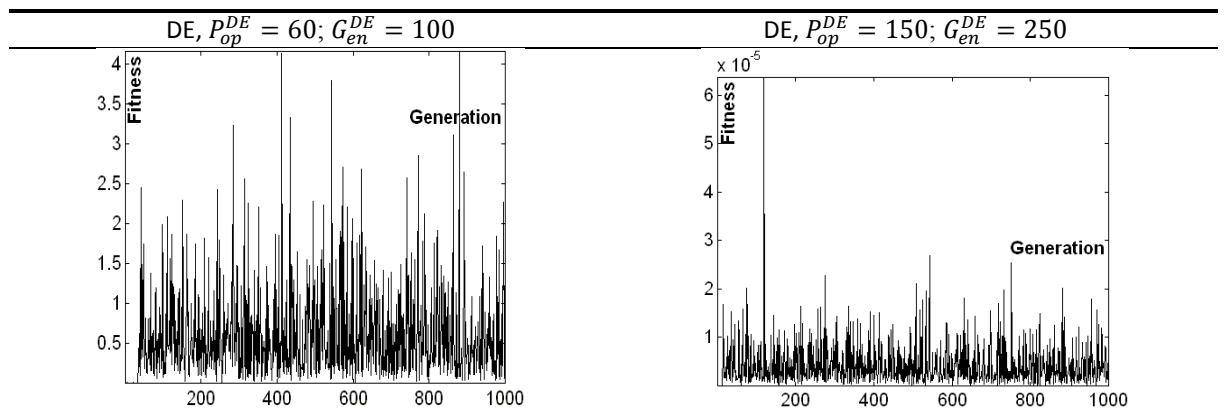
Tzv. „No-free lunch theorem“ (Wolpert & Macready 1995) poskytuje krutý, avšak pravdivý popis podivuhodného světa evolučních algoritmů. Na výsledku, který je dosažen primárním EA je závislý celý CEE. Zvolený EA musí být rychlý a především spolehlivý. Výběr optimalizátoru představoval poměrně zdoluhavý proces založený především na experimentálním ověřování dlouhé řady dostupných EA.

Čistě na základě experimentů bylo zjištěno, že pro odstranění distorze malého rozsahu – viz Obr. (6.1).3C a při použití polynomu stupně 7, je zapotřebí: pro sestavení CEE CMA+CMA cca $(6 \times 60) \times (9 \times 200) = 648 \times 10^3$ vyčíslení $fitness_1$ funkce – v ideálním případě a v nejhorším případě cca $(6 \times 300) \times (9 \times 200) = 3.24 \times 10^6$. Pro optimalizátor jDE+jDE je to v případě ideální konvergence cca $(20 \times 40) \times (30 \times 80) = 1.92 \times 10^6$ a v nejhorším případě pak $(60 \times 100) \times (30 \times 80) = 14.4 \times 10^6$, příp i více.



Obrázek (6.1).9 Výsledek konvergence optimalizátorů SGA a DE (rand2best1/bin)

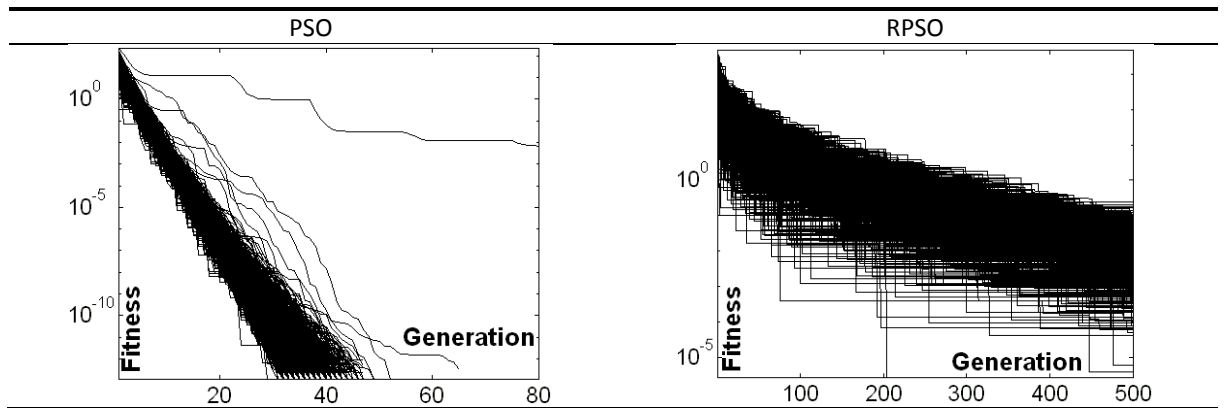
V případě, že by při výsledném odhadu - viz ř. 37 a ř. 42 Alg. (6.1).6, zvolený primární EA selhal následkem neúspěšné konvergence, pak by mohl selhat i celý kalibrační mechanismus. To samozřejmě není možné připustit. Proto byly provedeny analytické testy výkonnosti některých EA. Optimalizátor měl za úkol najít optimum na hyperploše – viz Obr. (6.1).7 pro jedinou značku. Byla vypočítávána $fitness_1$ funkce. Očekávaný výsledek by měl být cca 5.5×10^{-3} v každé iteraci, což je přepočtená součtová chyba velikosti 1 pixel na 180 značek M_C . Parametry testovaných EA jsou nastaveny tak, aby byla dosažena alespoň přibližně požadovaná přesnost. Počet opakování celého výpočtu je 1000 ve všech případech. Grafy na Obr. (6.1).9 - (6.1).13 zachycují celou konvergenci.



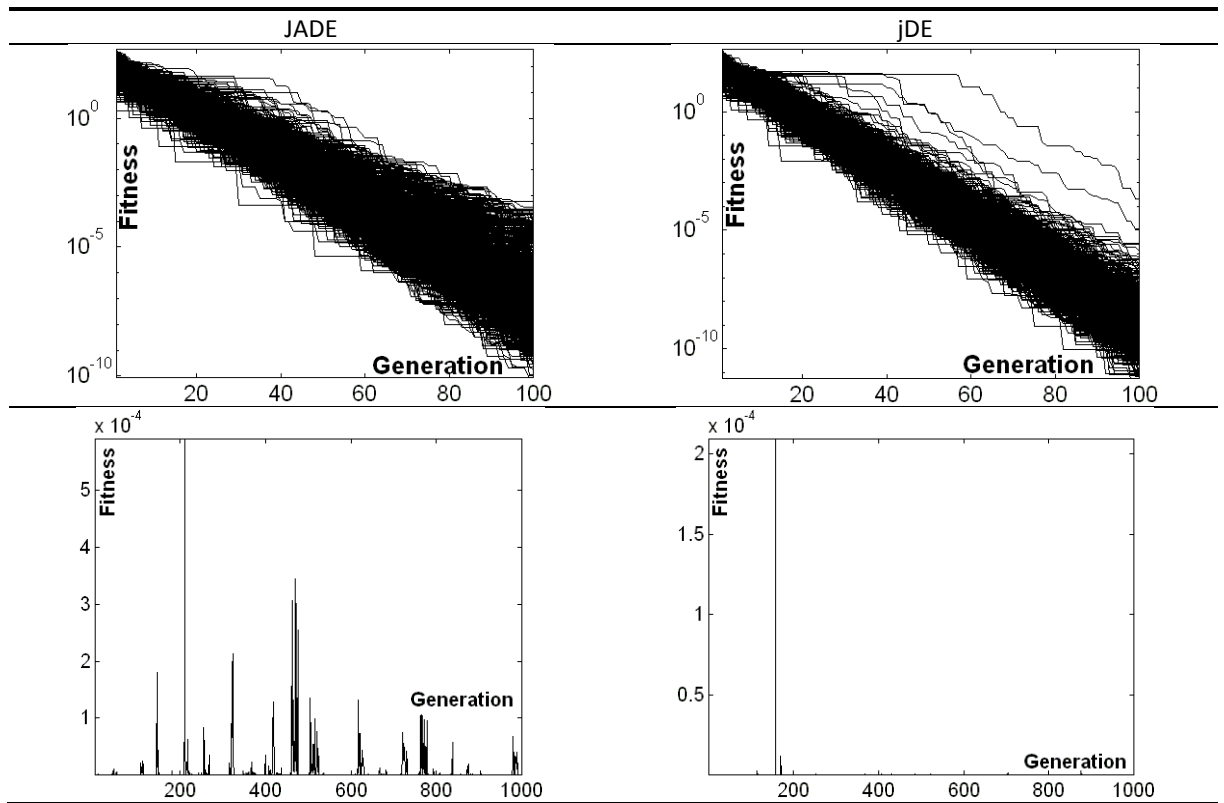
Obrázek (6.1).10 Výsledek evoluce DE, nejlepší hodnota dosažená v jednotlivých iteracích pro různé hodnoty P_{op}^{DE}, G_{en}^{DE} . Vlevo - nepoužitelné nastavení, Vpravo - dobrý výsledek, ale za cenu vysokých výpočetních nároků.

Mezi prvními testovanými EA byl nejstarší existující a to **SGA** (Holland 1962, 1975, 1992; Goldberg 1987, 1989). Pro úspěšný běh vyžadoval SGA minimálně $P_{op}^{SGA} = 250; G_{en}^{SGA} = 200$; 2-bodové křížení, elitistická mutace, binárně kódovaný chromozom 44-bitů dlouhý, identifikovatelná přesnost 1×10^3 . Při menším počtu jedinců a generací je SGA nepoužitelný. Bylo provedeno 1000 iterací evoluce, celkem 5×10^7 vyčíslení. Výsledek je na Obr. (6.1).9. Je vidět, že SGA poskytuje velmi špatné výsledky, často dochází k uváznutí v lokálním optimu. Seleční mechanismy jako např. ruletové kolo nebo turnajová selekce dávají ještě horší výsledky. SGA použitý např. v (Guermeur & Louchet 2003) je tedy velmi

málo účinný, nicméně použitelný. Další testovaný optimalizátor byla **DE** (Storn & Price 1997; Price et al. 1999) – viz Obr. (6.1).9. Bylo testováno všech 10 základních odchylkových vektorů. Jako nejučinnější se ukázaly rand2best1/bin,exp.



Obrázek (6.1).11 Výsledek konvergence optimalizátorů PSO a RPSO.



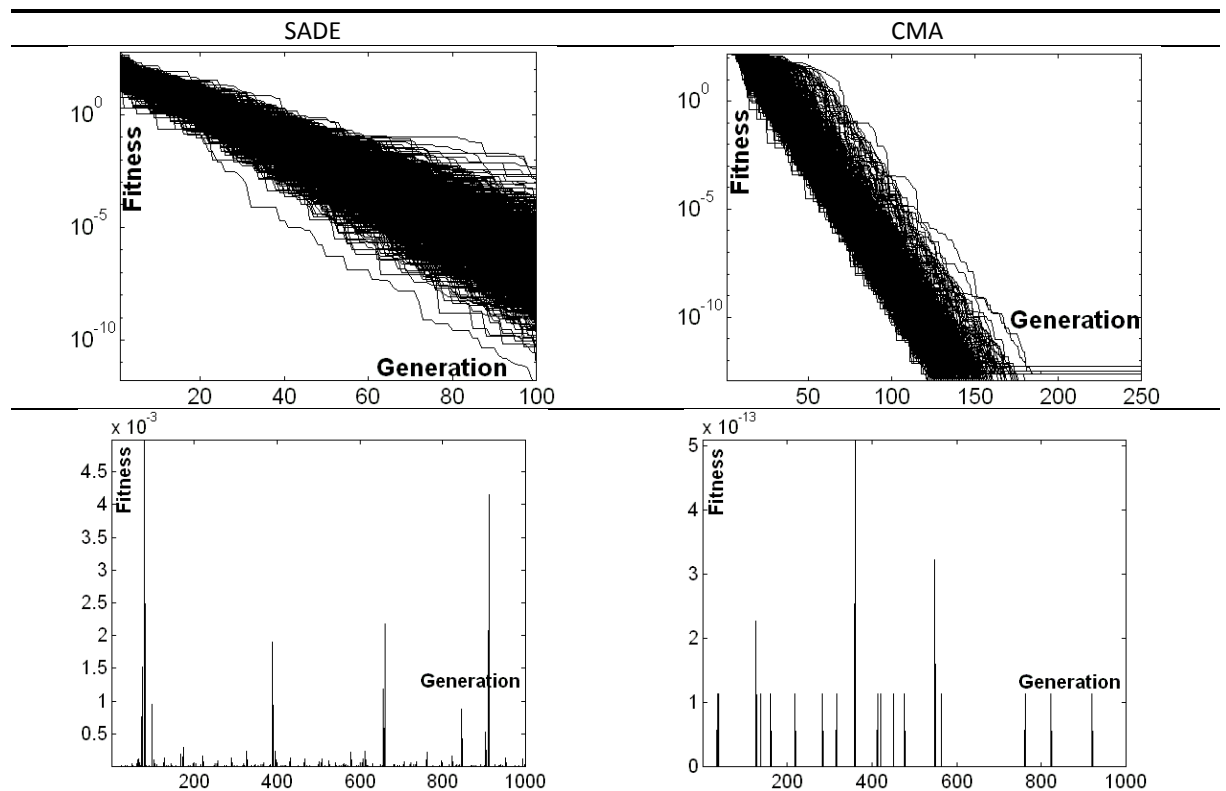
Obrázek (6.1).12 Optimalizátory JADE a jDE. Nahoře-záznam konvergence populace v iteracích pro optimalizátory JADE a jDE. Dole-nejlepší dosažený výsledek v iteracích při $P_{op}^{JADE,jDE} = 60$, $G_{en}^{JADE,jDE} = 100$.

U DE byly použity parametry $P_{op}^{DE} = 60$; $G_{en}^{DE} = 100$; $F = 0.4$; $P_{cr} = 0.9$. Bohužel při daném počtu generací trpí DE náhodným výskytem nedokonalé konvergence – viz Obr. (6.1).10. Původní DE je tedy i přes svoji značnou výkonnost obtížně použitelná. Bylo by nutné nastavit hodnoty přibližně $P_{op}^{DE} = 150$; $G_{en}^{DE} = 250$, což je z výpočetního hlediska neúnosné. Mezi testovanými byl rovněž **PSO** (Kennedy and Eberhart 1995). Je všeobecně známo, že PSO vyžaduje pro svůj úspěšný běh cca 2-10x více jedinců než např. SGA nebo DE. Zvolené parametry $P_{op}^{PSO} = 250$; $G_{en}^{PSO} = 80$; $\Omega = 0.05$, $A = B = 1$. Pokud jsou použity menší hodnoty např. jako u DE, algoritmus není schopen úspěšné konvergence. Přes vysoký

počet vyčíslení účelové funkce je četnost selhání cca 2-3 na 1000 iterací. Celkový počet vyčíslení 2×10^7 . Výsledek je na Obr. (6.1).11. **RPSO** (Urfalioglu 2004) – viz Obr. (6.1).11, dosáhl výrazně horšího výsledku. $P_{op}^{RPSO} = 250$; $G_{en}^{RPSO} = 500$; $\Omega = 0.05$; $A = 1.5$; $B = -1.5$; $C = 0.5$. Změna pracovních parametrů bohužel nevede k lepším výsledkům a RPSO je tak nepoužitelný.

Optimalizátor **JADE** (Zhang & Sanderson 2009) patří mezi pokročilé adaptivní metaheuristiky. $P_{op}^{JADE} = 60$; $G_{en}^{JADE} = 100$; $c = 0.1$; $p = 0.05$; verze s archivem. JADE poskytuje velmi dobré výsledky a je také velmi rychlý. Pokud je použita varianta „bez archivu“ (without archive) je celkový výsledek horší jen o cca 3 procenta. Na Obr. (6.1).12 jsou zobrazeny výsledky.

Podobný optimalizátor jako JADE je i **jDE** (Brest et al. 2006; Brest et al. 2007) – viz Obr. (6.1).12. Byl testován se stejnými parametry P_{op}^{jDE} , G_{en}^{jDE} jako JADE. Dále bylo zvoleno $\tau_1 = 0.6$; $\tau_2 = 0.4$; $f_u = 0.7$; $f_l = 0.3$; rand2best1/bin, na základě experimentů. Rychlost obou algoritmů je přibližně stejná, jDE je nepatrně rychlejší. Poskytuje ale výrazně lepší výsledky s ohledem na dokonalost konvergence – viz Obr. (6.1).12 dole. Platí to však pouze pro zvolený odchylkový vektor. Ostatní ‘klasické’ odchylkové vektory (Storn & Price 1997) tak dobré výsledky neposkytují. Při volbě hodnot $P_{op}^{jDE} = 60$, $G_{en}^{jDE} = 100$ “selhání” následkem nedokonalé konvergence nebylo pozorováno, ani pokud počet iterací byl 10000. Tyto hodnoty lze prohlásit za ‘bezpečné’ pro použití v CEE. Přesnost lze u obou optimalizátorů dále zvyšovat tím, že se zvýší hodnota P_{op}^{jDE} a obzvláště pak G_{en}^{jDE} , ovšem za cenu vyšších výpočetních nároků.



Obrázek (6.1).13 Optimalizátory SADE a CMA.

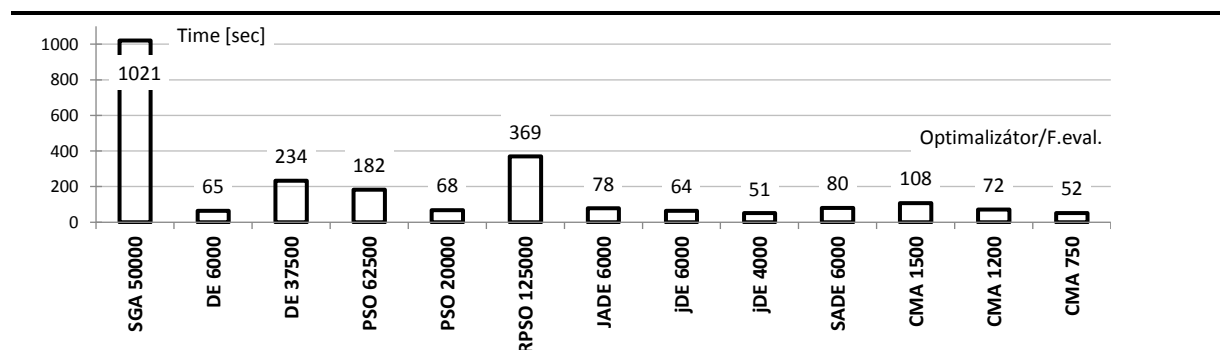
Poslední testované EA jsou SADE a CMA – viz Obr. (6.1).13. **SADE** (Qin & Suganthan 2005) – Obr. (6.1).13, poskytoval za stejných podmínek téměř shodné výsledky jako JADE. Parametr $L_P = 10$ – tzv. learning period. SADE je jen nepatrně pomalejší, než jDE. Na Obr. (6.1).13 dole je vidět, že úspěšnost konvergence v iteracích je horší, než u jDE. **CMA** (Hansen 2006) poskytuje podle očekávání skvělé výsledky. $P_{op}^{CMA} = 6$, $G_{en}^{CMA} = 250$. CMA je možné jednoznačně použít místo jDE – to je dobře patrné z Obr. (6.1).12 a (6.1).13. Pro dosažení očekávaného výsledku je dostačující $P_{op}^{CMA} = 6$, $G_{en}^{CMA} = 150$ což

je pouhých 900 vyčíslení $fitness_1$ funkce. Tj. cca 6.6x méně, než jDE. CMA je však výpočetně a tím i časově výrazně náročnější, než jDE a to cca 5-6x. Ve výsledku je tak možné oba algoritmy na místě výpočetního optimalizátoru primární $fitness_1$ funkce zaměnit a výsledná přesnost CEE přitom zůstane stejná – s ohledem na numerickou vyčíslitelnost při výpočtu. Při experimentech se ukázalo, že u CMA je vhodné kombinovat dvě ukončovací podmínky pro $fitness_1$. Jednak limit počtu generací a pak limit fitness na hodnotě cca 1×10^{-4} . Podle Obr. (6.1).4 je možné na místě primárního optimalizátoru $fitness_1$ funkce použít, jak jDE, tak i CMA algoritmus. Podobně je tomu i u sekundárního optimalizátoru.

Na základě výše uvedených experimentálních výsledků byly zvoleny dva optimalizátory, a to jDE a CMA, kterým byla věnována zvýšená pozornost a ze kterých je možné CEE sestavit. Optimalizátor jDE byl do této studie zvolen na základě experimentálních výsledků v publikacích (Brest et al. 2010; Montes et al. 2006; Quin et al. 2009; Wang et al. 2011) a především na základě vlastních praktických experimentů s několika vybranými EA, z nichž právě jDE poskytoval velmi dobré, a především stabilní výsledky. Algoritmus CMA byl zvolen na základě rozsáhlého výzkumu, který poskytují publikace (Hansen et al. 1995, 1996, 2003; Hansen & Ostermeier 1996,1997, 2001; Hansen & Kern 2004; Auger & Hansen 2005; Jastrebski & Arnold 2006; Hansen 2006; Igel et al. 2006; Auger et al. 2009; Pošík et al. 2012; Pošík & Kubalík 2012; Hansen 2020; Wikipedia 2016; Hansen 2020).

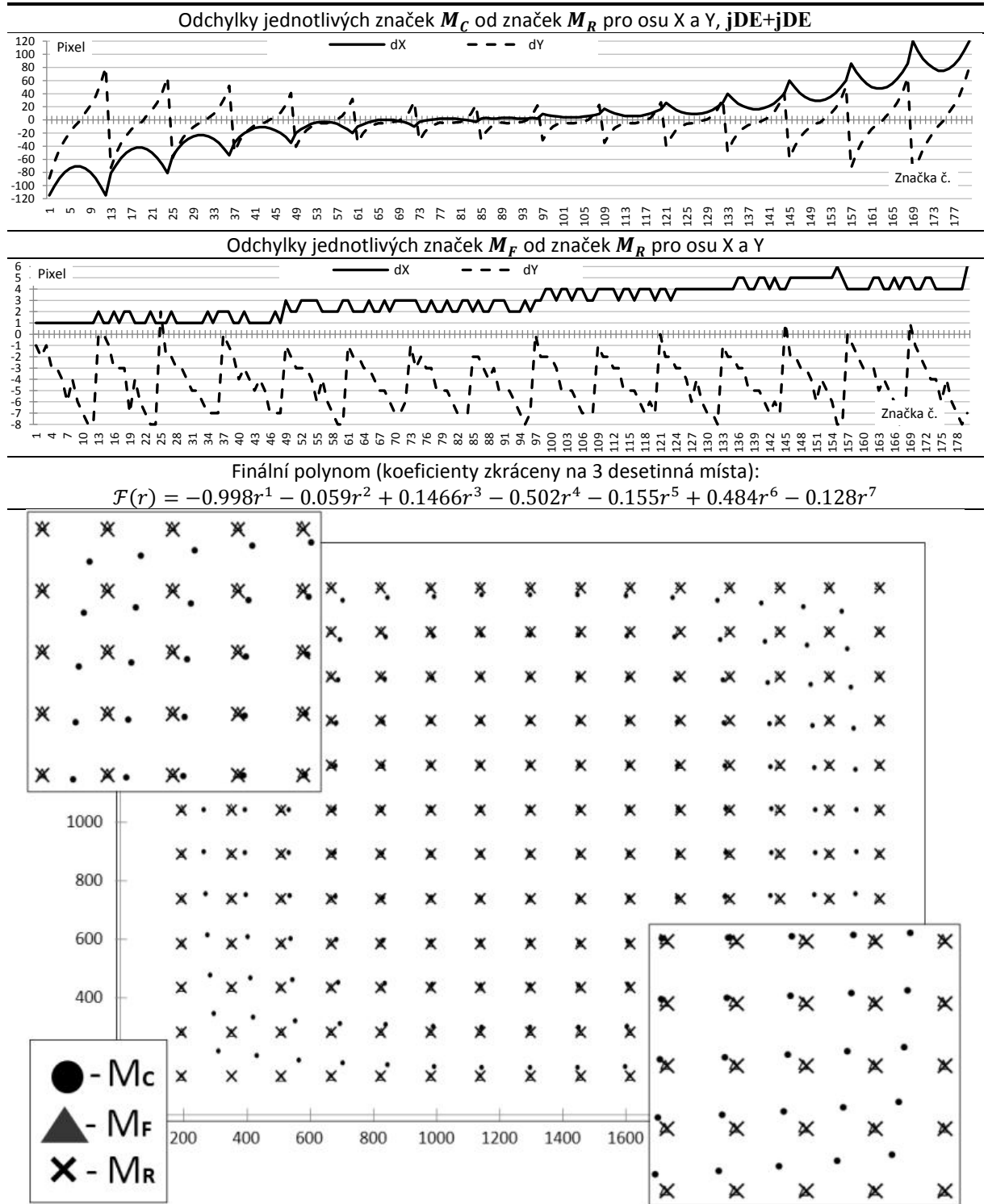
Účelová funkce $fitness_2$ – viz Alg. (6.1).8, je založena na výpočtu $fitness_1$ účelové funkce a pracuje až v 7-mi dimenzionálním prostoru, který je tvořen parametry c_i polynomu \mathcal{F} - viz (6.1.12). Koeficienty c_i nabývají v našem případě hodnot v rozmezí cca $(-1.5; +1.5)$, prohledávaný prostor však není explicitně omezen jako je tomu u $fitness_1$ funkce. Počet dimenzí, resp. počet použitých parametrů c_i je v presentovaném CEE libovolný v počtu 1 až 7. Je možné využít jak sudé, tak i liché koeficienty. Koeficient c_1 bude v případě presentovaného CEE využit vždy, protože obraz biometrického scanneru, resp. z fotoaparátu je vždy menší nebo větší, než je referenční mřížka, tj. nebude nikdy nastaven na fixní hodnotu např. $c_i = 1$.

Čistě až na základě praktických experimentů bylo zjištěno, že konvergence CMA není vždy zcela bezproblémová. Zde použitou verzi CMA neboli „pureCMA“ je možné nahradit modifikací CMA označovanou jako „restartCMA“ (Auger & Hansen 2005; Pošík & Huyer 2012). Konvergence se pak při opakovaném výpočtu může protáhnout až na 250-300 generací nebo i více. Může pak být dosažena lepší hodnota účelové funkce. Při hladkém běhu je dostačujících 60-80 generací, velmi zřídka i méně. Při experimentech byla ověřena také konvergence CMA a jDE (rand2best1/bin) na různých benchmarkových funkcích např. (Whitley et al. 1996) s počtem dimenzí 7. V tomto případě si optimalizátor jDE ($f(x) = 0.1679$) si vedl lépe než CMA ($f(x) = 19.22$).



Obrázek (6.1).14 Časové nároky testovaných EA – SGA, PSO, RPSO, JADE, jDE, SADE a CMA. Údaj za názvem EA udává počet vyčíslení účelové funkce, potřebných k úspěšné konvergenci. ‘F.eval’-počet vyčíslení účelové funkce potřebný k dosažení dobrého výsledku.

Časové nároky testovaných EA při výpočtu $fitness_1$ funkce jsou na Obr. (6.1).14. jDE a CMA jednoznačně poskytují nejrychlejší výsledek s ohledem na počet vyčíslení $fitness_1$ účelové funkce. Proto také byly zvoleny do CEE. JADE, SADE, PSO nebo i SGA optimalizátory by bylo možné využít také, ale jsou složitější na algoritmizaci a pomalejší, i když např. JADE jen o málo. JADE a SADE mají horší schopnost rychlé konvergence – v této úloze.



Obrázek (6.1).15 Finální výsledek kalibrace, optimalizátor jDE+jDE. Nahoře-odchylky kalibračních značek M_C od referenčních značek M_R . Uprostřed-odchylky značek M_F od referenčních značek M_R a polynom s finálními koeficienty odhadnutými pomocí CEE. Dole-výsledný korigovaný obraz se značkami M_C, M_F, M_R + zvětšenina levého horního a pravého dolního rohu.

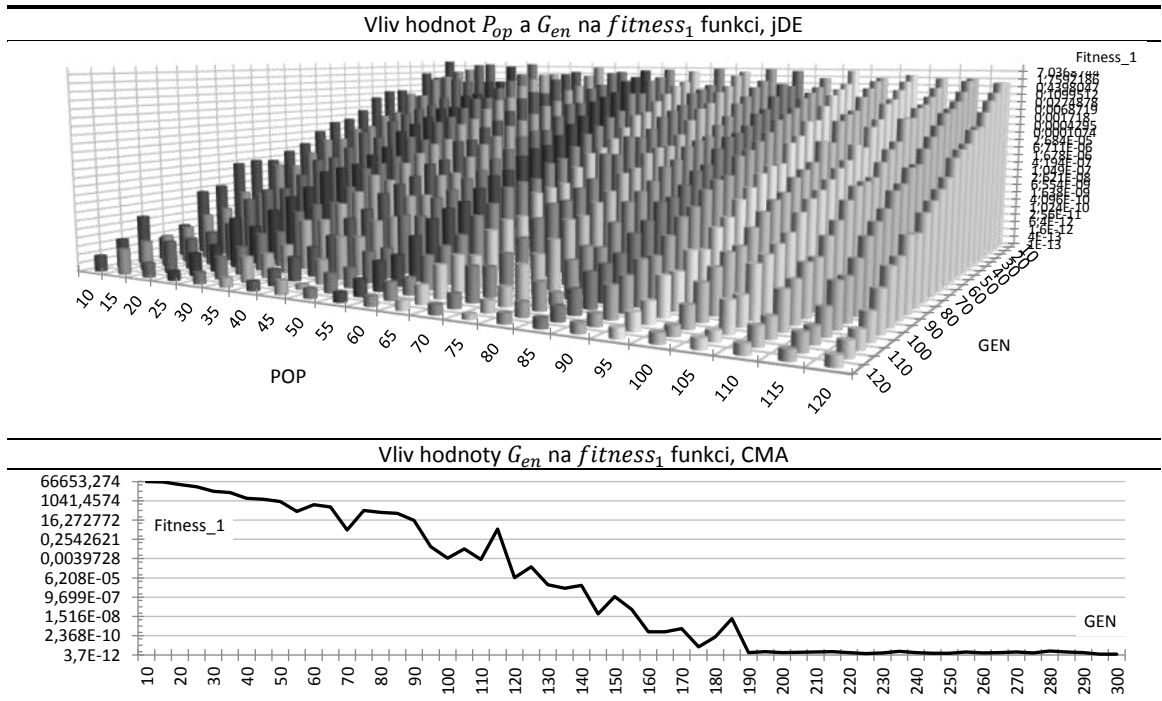
Překvapivě optimalizátory DE a RPSO jsou téměř nepoužitelné. Nejvíce překvapující je to u originální DE (randtobest2/bin), ale výsledky experimentů jsou přesné. SGA potřebuje nesmírně dlouhou dobu na úspěšnou konvergenci.

6.1.6.4 N-dimenzionální kaskádový evoluční optimalizátor

Experimentální ověření navržených CEE, tj. kombinací jDE+jDE, jDE+CMA a CMA+CMA – viz Obr. (6.1).4, bylo provedeno tak, že evolučnímu estimátoru byl předložen obraz zatížený distorzi viz Obr. (6.1).3C, na kterém byly zobrazeny čtvercové značky \mathbf{M}_C o rozměru 20×20 pixelů v počtu $15 \times 12 = 180$. Úkolem CEE bylo s využitím kalibračních značek provést korekci soudkové distorze malého rozsahu, tj. nalézt koeficienty c_i polynomu \mathcal{F} - viz (6.1.8). Výpočet probíhal podle diagramu Obr. (6.1).5. Použitý fotoaparát byl CanonPowerShotS60, 2592x1944x24bppRGB; výstup ve formátu BMP. Použitý procesor ADM3500+/Orleans/2.2GHz. Biometrický scanner byl pečlivě seřízen a byl nastaven střed kalibračního obrazu a změřeny min/max okraje na poloosách $\pm X, \pm Y$. Pak byl pořízen snímek + transfer obrazu přes USB do PC. Sestavený CEE nejprve převedl RGB obraz na šedotónovou reprezentaci a současně prahováním na B&W obraz. Sít referenčních značek \mathbf{M}_R byla sestavena na základě kalibračního obrazu Obr. (6.1).3C následovně: offset první značky byl $X = 190; Y = 130$ (počátek vlevo nahoře), grid spacing $\Delta X = 158; \Delta Y = 152$ pixelů. Podle Alg. (6.1).5 byly nalezeny všechny značky \mathbf{M}_C a vypočtena jejich těžiště. Tyto informace slouží jako vstupní údaje pro všechny testované verze CEE.

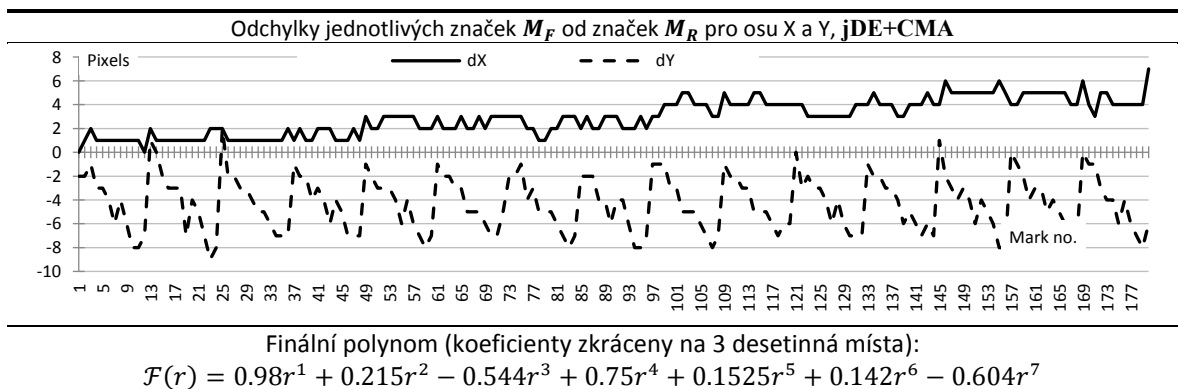
Nejprve popíšeme výsledky pro první testovaný CEE typ jDE+jDE. Parametry prvního jDE pro $fitness_1$: $P_{op}^{jDE-F1} = 60$; $G_{en}^{jDE-F1} = 120$; $\tau_1 = 0.7$; $\tau_2 = 0.3$; $F_u = 0.5$; $F_l = 0.5$. jDE pro $fitness_2$: $P_{op}^{jDE-F2} = 80$; $G_{en}^{jDE-F2} = 200$; $\tau_1 = 0.7$; $\tau_2 = 0.3$; $F_u = 0.5$; $F_l = 0.5$. Korigujeme soudkovou distorzi obrazu z Obr. (6.1).3C. Výsledek kalibrace je na Obr. (6.1).15. Nahoře na obrázku je graf vzdáleností kalibračních značek \mathbf{M}_C a referenčních značek \mathbf{M}_R neboli stav před korekcí. Uprostřed je graf vzdáleností mezi korigovanými značkami \mathbf{M}_F a referenčními značkami \mathbf{M}_R , resp. výsledek procedury odstranění distorze a také zkrácený polynom. Součet vzdáleností jednotlivých značek \mathbf{M}_R a \mathbf{M}_F je $fitness_2 = 999.05$ pixels. Součtová vzdálenost mezi značkami \mathbf{M}_C a \mathbf{M}_R je 7024 pixelů. Zůstatková chyba vzhledem k původní součtové vzdálenosti je tedy $\frac{100}{7024} * 999.05 = 14.2\%$. Tato hodnota je dostatečně nízká a rozdíl mezi 'ideálním' obrazem a obrazem korigovaným pomocí CEE lze jen velmi obtížně rozpoznat pouhým okem. Chyba této velikosti je téměř stejná u všech testovaných CEE. Na Obr. (6.1).15 dole je pak obraz se značkami $\mathbf{M}_{C,F,R}$. V rozích jsou zvětšeniny obrazu, kde je možné vidět, že trojúhelníky značek \mathbf{M}_F jsou nepatrně posunuté ze středů (křížků – viz Obr. (6.1).15) značek \mathbf{M}_R . Rozdíly v poloze u značek ve zvětšeninách v levém a pravém rohu jsou větší, což je dáno nedokonalým vycentrováním obrazu o cca 2-3 pixely v ose X. Odchylku takovéto velikosti již není možné korigovat ručně posouváním kalibračního obrazu po projekční desce a pro účely našeho biometrického scanneru je přijatelná. Výpočet korekčních koeficientů trval 1168 minut – viz Obr. (6.1).19. Přesnost je nastavena pracovními parametry optimalizátoru jDE – viz Obr. (6.1).18 nahoře, vyšší hodnoty P_{op}, G_{en} zaručují lepší výsledek. Obdobně je tomu u CMA.

Druhý experiment se týkal evolučního estimátoru v sestavení jDE+CMA. Výpočet trval jen 240minut a parametry jDE byly stejné jako u předchozího experimentu. Pro CMA byly ponechány automaticky generované parametry podle (Hansen 2013; Hansen 2020) tj. $P_{op}^{CMA-F2} = 9$; $G_{en}^{CMA-F2} = 200$. Počet jedinců populace je počítán podle vzorce $P_{op}^{CMA-F2} = 4 + \lfloor 3 * \log_2 N_{dim} \rfloor$, kde N_{dim} je počet dimenzí úlohy, v našem případě $N_{dim} = 7$ a pro experimenty na Obr. (6.1).20 je pak N_{dim} pro každou úlohu jiný.

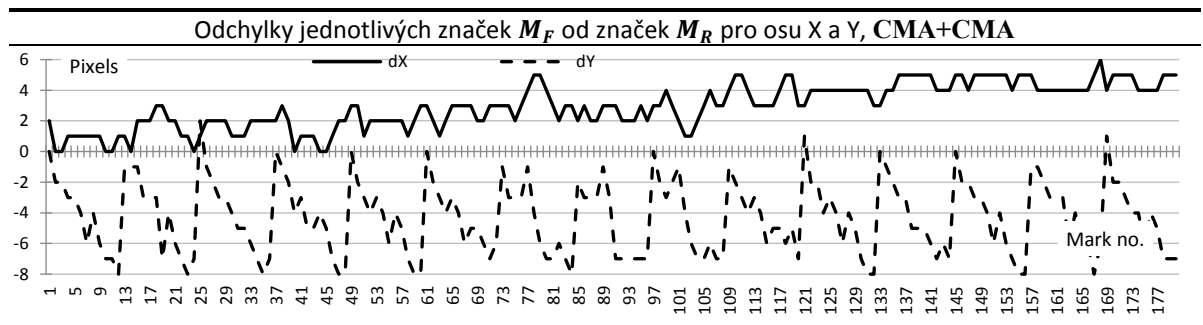


Obrázek (6.1).16 Vliv hodnot P_{op} a G_{en} na $fitness_1$ funkci u optimalizátoru **jDE** a **CMA**.

Výsledek optimalizace je 1001.42 pixelů. To je téměř identická hodnota jako u předchozího testovaného CEE. Výsledný obraz jednotlivých $M_{C,F,R}$ je naprosto shodný jako u předchozího případu na Obr. (6.1).15. Pouhým okem je výsledná chyba nerozlišitelná. Výhodou sestavení **jDE+CMA** je tedy jednoznačně rychlost. Běh tohoto sestavení CEE byl mnohokrát opakován, vždy se stejným výsledkem. Maximální odchylka finálních značek M_F od ideální pozice definované maticí M_R se pohybovala v rozmezí $\langle +8; -8 \rangle$ pixelů – viz Obr. (6.1).17. Stejná odchylka byla změřena u CEE **jDE+jDE**. Je poměrně dobře patrné, že téměř u poloviny značek M_F se odchylka od ideální pozice pohybovala v rozmezí $\langle +4; -4 \rangle$. Byly také provedeny opakované experimenty s trojnásobným počtem generací u CMA, ale lepších výsledků se bohužel nepodařilo dosáhnout.



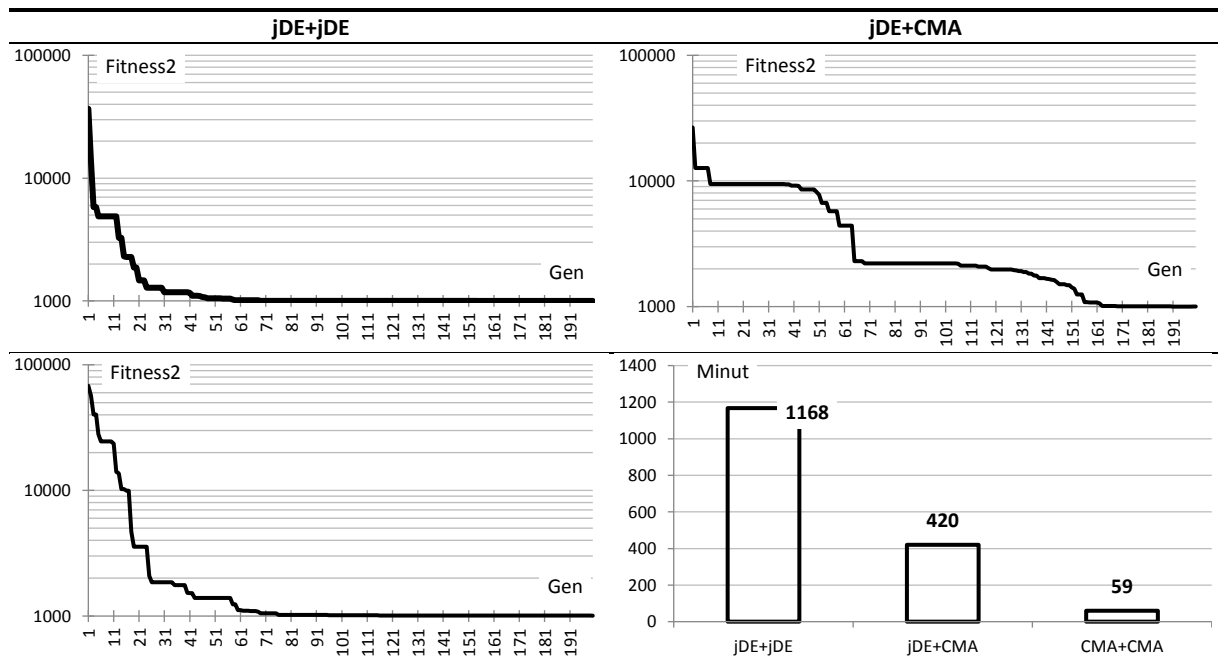
Obrázek (6.1).17 Finální výsledek kalibrace, optimalizátor **jDE+CMA**, ostatní značení jako na Obr. (6.1).15.



Finální polynom (koeficienty zkrácené na 3 desetinná místa):

$$\mathcal{F}(r) = 1,034r^1 - 0,249r^2 + 0,6018r^3 + 0,22r^4 - 0,472r^5 - 0,712r^6 + 0,97r^7$$

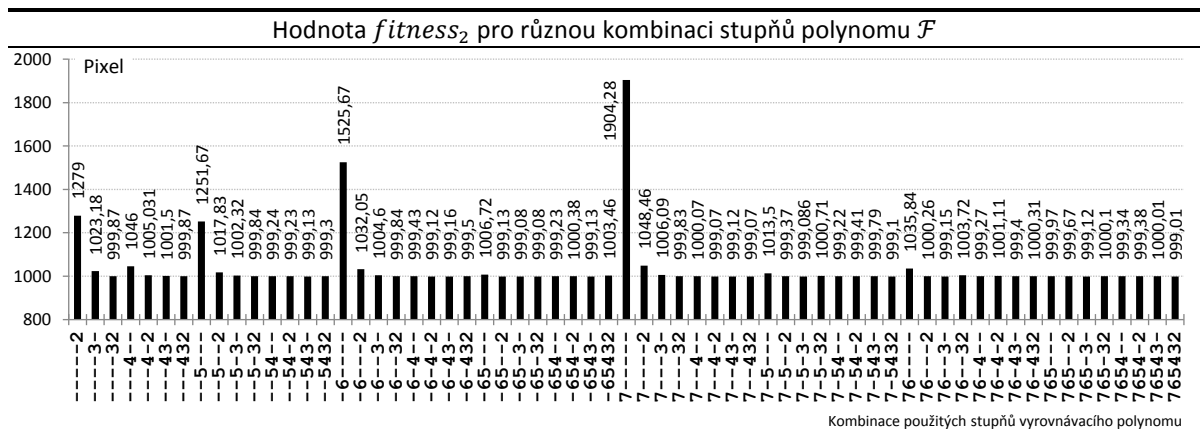
Obrázek (6.1).18 Finální výsledek kalibrace, optimalizátor CMA+CMA, ostatní značení jako na Obr. (6.1).15.



Obrázek (6.1).19 Záznam konvergence pro $fitness_2$ funkci u jednotlivých CEE a graf časové náročnosti testovaných CEE.

Poslední testovaná kombinace je CEE v sestavení **CMA + CMA**. Pracovní parametry $p_{op}^{CMA-F1} = 6$; kombinovaná ukončovací podmínka $G_{en}^{CMA-F1} = 150$ nebo $fitness_1 < 1 \times 10^4$, $p_{op}^{CMA-F2} = 9$; $G_{en}^{CMA-F2} = 200$. Výsledek optimalizace je opět téměř identický jako u předchozích dvou CEE. Dosažená hodnota je 1000.42 pixelů a graf odchylek je na Obr. (6.1).19. Doba potřebná pro výpočet je pouhých 59 minut. Tento výsledek byl dosažen díky použití kombinované ukončovací podmínce u $fitness_1$ funkce. Požadovanou přesnost aproximace lze velmi dobře nastavit pomocí pracovních parametrů – viz Obr. (6.1).17 dole, podobně jako u předchozích CEE. Závislost pracovních parametrů nemá u CMA lineární průběh jako je tomu u jDE, je možné pozorovat časté náhodné výkyvy. Výhodou **CMA + CMA** je tedy jednoznačně výpočetní rychlost. Na Obr. (6.1).19 jsou zobrazeny křivky konvergence $fitness_2$ funkce pro jednotlivé CEE a také graf již zmiňovaných časových nároků. Konvergenční křivky jsou ve všech případech velmi strmé především v počátku evoluce. Dobrého výsledku je možné dosáhnout již po cca 70 generacích, za předpokladu, že konvergence neuvázne v lokálním optimu. Křivka **jDE+CMA** – viz Obr. (6.1).19, je pozvolnější. Nejedná se o pravidlo, ale

pouze o ukázkou méně vydařené konvergence. Přesto však očekávaná finální hodnota byla hladce dosažena díky zvolenému počtu generací 200.



Obrázek (6.1).20 Dosažená přesnost pro různé kombinace stupňů polynomu \mathcal{F} ; typ CMA+CMA. Stupeň 1 použit vždy.

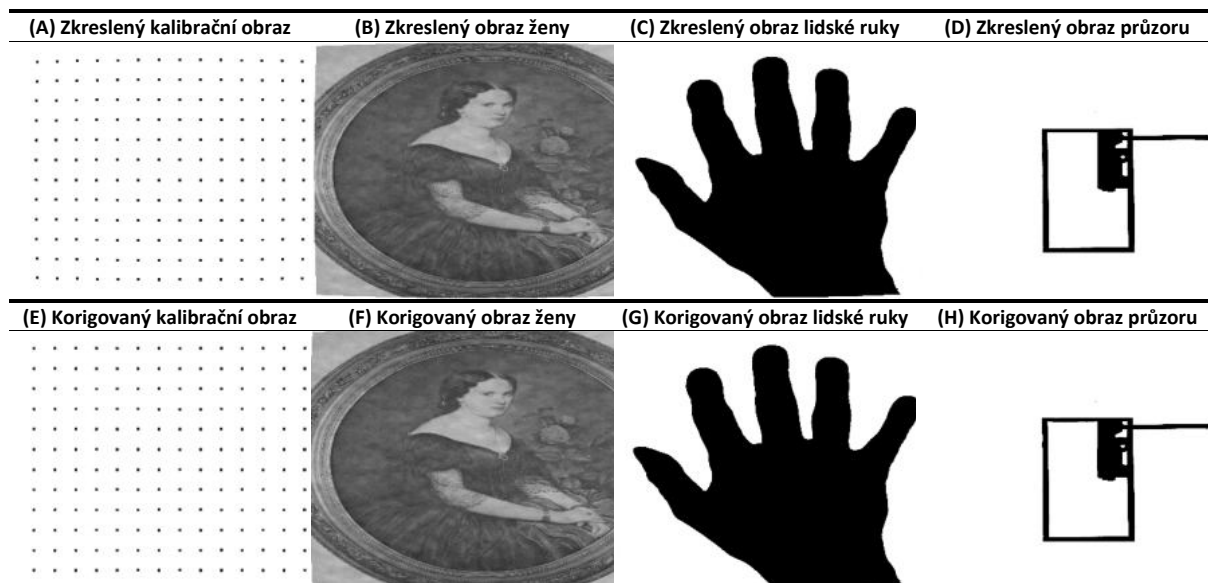
Výše popsané experimenty byly prováděny s plným počtem stupňů polynomu (6.1.12) tedy 7. Všechny 3 typy CEE je možné využít s libovolnou kombinací stupňů např. jen liché nebo jen sudé členy polynomu – viz Obr. (6.1).20. Provedené experimenty prokázaly, že pokud jsou využity jen všechny liché členy, pak dosažená hodnota $fitness_2$ je stejná jako v případě, kdy jsou použity jen všechny sudé členy a člen první. To je odlišný výsledek než např. v (Chen et al. 2011). Pokud je použit menší počet členů polynomu např. pouze první a poslední, pak je výsledek samozřejmě horší.

Bylo testováno celkem 64 různých kombinací – viz Obr. (6.1).20 v sestavení CEE CMA+CMA. Ve všech případech byl použit první člen polynomu se stupněm jedna. Dobrých výsledků bylo možné dosáhnout s využitím mnoha různých kombinací stupňů polynomu \mathcal{F} . Pro korekci distorze obrazu malého rozsahu je možné použít např. kombinaci 1,5,4 – viz Obr. (6.1).20.

Minimální dosažená hodnota $fitness_2$ funkce byla v průběhu experimentů cca 999 pixelů. Lepší hodnotu se nepodařilo dosáhnout v žádném z testů. Značný podíl na tom má nepřesné ruční ustavení kalibračního výkresu CD_B , chyba v určení středu distorze, chyba ustavení náklonu unášecí desky digitálního fotoaparátu atd. Bohužel ruční ustavení biometrického scanneru neumožňuje všechny tyto chyby korigovat lépe. Proto je dosažená hodnota uvažována jako nejlepší možná za daných podmínek.

6.1.6.5 Re prezentace míry zkreslení

Navržená metoda pro korekci obrazu je součástí projektu autentifikace osob s využitím 2D kontury lidské ruky. Po sestavení biometrického scanneru bylo zjištěno, že výsledný obraz získaný z digitálního fotoaparátu je deformován jako následek vady distorze obrazu – soudkové zkreslení. Zkreslení je malé, ale viditelné. Obzvláště, pokud je použita krátká optická vzdálenost – fotoaparát je blízko projekční desky. Digitální fotoaparát může být adjustován v ose Z. Aby bylo možné nějak efektivně korigovat tuto nechtěnou vadu obrazu, byla navržena korekční metoda, která využívá EA. Právě EA byly využity v předchozích projektech a díky tomu byly známy jejich schopnosti. EA poskytují velmi výkonný optimalizační nástroj. Bohužel samotná kombinace EA a CEE a správné nastavení pracovních parametrů je nesmírně složitá samo o sobě. Čistě na základě experimentů bylo zjištěno, že pro korekci obrazu, který je deformován soudkovým zkreslením je dostačující metoda využívající polynomiální algoritmus (Brown 1965).



Obrázek (6.1).21 Několik reálných příkladů korekce distorze. Pro získání všech těchto obrazů byl použit sestavený biometrický scanner. (A, E)-kalibrační obraz, (B, F)-portrét ženy (neprůsvitný, neprůhledný obrázek z kalendáře), (C, G)-lidská ruka, (D, H)-součást ze sériové výroby, kontrolní průzor s pomocným LED displejem. Je prováděna kontrola správného umístění těsnění na skleněné součásti a samozřejmě i její celkový tvar s využitím kontury. Je kontrolována pozice, natočení, průhlednost, průsvitnost atd., tedy mnoho různých parametrů najednou. Navržený kaskádový estimátor samozřejmě provádí pouze korekci distorze obrazu. Obrazy (A, B, C, D) jsou zkreslené – polštářové zkreslení. Obrazy (E, F, G, H) jsou korigované s využitím CEE ($jDE+jDE$) a jsou tedy bez distorze. Jak je možné vidět na obraze (B), distorze je malá, ale viditelná obzvláště v rozích obrazu (B).

Navržený algoritmus pro identifikaci osob pracuje pouze s B&W obrazovou reprezentací, resp. vektorovou reprezentací kontury lidské ruky; kontura je reprezentována pomocí cca 11-15tis. úseček. Osa optického systému biometrického scanneru, resp. digitálního fotoaparátu je kolmá na projekční desku biometrického scanneru a při vytváření databáze kontur je nezbytné zachovat konstantní ohniskovou vzdálenost. Proto je možné kalibrační proces provést pouze jednou a korekční koeficienty pak zůstávají stejné. Podstatnou výhodou je také to, že biometrický scanner má tzv. temnou komoru s konstantním osvětlením. Takovéto uspořádání poskytuje vysoce přesné snímky kontury lidské ruky. Biometrický scanner může být samozřejmě využit i pro další výzkum např. v oblasti klasifikace kontur předmětů na základě kontury. Typicky se jedná o klasifikace strojních součástí, které jsou vyřezávané laserem, vysekávané na lisech, různé typy plochých součástí jako výsledek práce vstřikování plastů, sčítání jednoduše separovatelných součástí na výrobních linkách, kontrola pozice a natočení součástek, kontrola počtu součástí, kontrola průhlednosti, průsvitnosti a také jestli součásti nejsou ušpiněné a mnoho a mnoho dalších možností.



(I) Vybraná zkraslená a k nim odpovídající korekční polynomy; Je využit CEE, typ jDE+jDE, pracovní parametry jsou identické s pracovními parametry v kap. 7.4.

1. Image (A), distozní polynom: $\mathcal{F}(r_u) = +1.00000r^1 - 0.00100r^2 - 0.00100r^3 - 0.00100r^4 - 0.00100r^5 - 0.00100r^6 - 0.00100r^7$
2. Image (E), korekční polynom: $\mathcal{F}(r_d) = +1.01337r^1 - 0.09806r^2 + 0.42649r^3 - 0.74902r^4 + 0.48362r^5 + 0.33673r^6 - 0.40291r^7$
3. Image (B), distozní polynom: $\mathcal{F}(r_u) = +1.00000r^1 - 0.01000r^2 - 0.00200r^3 - 0.00200r^4 - 0.00200r^5 - 0.00200r^6 - 0.00200r^7$
4. Image (F), korekční polynom: $\mathcal{F}(r_d) = -1.00254r^1 - 0.09956r^2 + 0.02746r^3 - 0.11779r^4 - 0.06537r^5 + 0.11867r^6 + 0.00904r^7$
5. Image (C), distozní polynom: $\mathcal{F}(r_u) = +1.00000r^1 - 0.01000r^2 - 0.01000r^3 - 0.00200r^4 - 0.00200r^5 - 0.00200r^6 - 0.00300r^7$
6. Image (G), korekční polynom: $\mathcal{F}(r_d) = +0.99747r^1 + 0.12676r^2 + 0.04710r^3 + 0.00186r^4 + 0.10258r^5 + 0.02167r^6 - 0.09264r^7$
7. Image (D), distozní polynom: $\mathcal{F}(r_u) = +1.00000r^1 - 0.02000r^2 - 0.02000r^3 - 0.00200r^4 - 0.00300r^5 - 0.00300r^6 - 0.00300r^7$
8. Image (H), korekční polynom: $\mathcal{F}(r_d) = +0.98177r^1 + 0.34843r^2 - 0.12887r^3 + 0.03530r^4 + 0.44997r^5 - 0.08758r^6 - 0.30127r^7$

Obrázek (6.1).22 Několik příkladů, které prezentují účinnost navrženého CEE (jDE+jDE). (A, B, C, D) – obraz Leny (Po 2020) uměle deformovaný distorzi zvolených parametrů. (E, F, G, H) – korigovaný obraz Leny. MSE (Střední Kvadratická Chyba) a PSNR koeficient (Wikipedia 2020a) je použit pro měření schopnosti CEE korigovat optickou chybu. Korespondence obrazů pro výpočet hodnot MSE: (A → E), (B → F), (C → G), (D → H). Šedotónový obraz Leny byl získán z originálu 24 bit RGB obrazu (Wikipedia 2020b) s využitím klasické konverze RGB obrazu na B&W reprezentaci s koeficienty R=0.299, G=0.587, B=0.114. (I)-Distorzní polynomy a vypočtené korekční polynomy použité pro obrazy (A, B, C, D), zkráceno na 5 desetinných míst. $\mathcal{F}(r_u)$ – byla použita funkce (6.1.12) ke zkraslení originálního obrazu; $\mathcal{F}(r_d)$ – korekční polynom, který je inverzní k polynomu $\mathcal{F}(r_u)$.

Při kontrole procházejí tyto součásti temnou komorou, která je umístěna na výrobním páse nebo naopak temná komora může být umístěna na vhodný pohyblivý držák. Další možností je právě identifikace osob využívající kontury lidské ruky, která má roztažené nebo naopak sepnuté prsty, nebo také autentifikace na základě ruky zaťaté v pěst nebo držící kolík. Na Obr. (6.1).21A, B, C, D, E, F, G, H je zobrazeno několik reálných příkladů, které popisují, jak také navržený CEE (jDE+jDE) pracuje a jak je úspěšný při odstraňování distorze obrazu. Obr. (6.1).21A, E snímky s a bez soudkového zkraslení. Obr. 21B, F ukazují náhodně zvolený obraz ženy z kalendáře. Obr. (6.1).21C, G ukazuje fotografii kontury lidské ruky. Takovýto obrázek je již připraven pro klasifikaci nebo pro detekci kontury, která bude později klasifikována samostatně. Obr. (6.1).21D, H ukazuje běžnou strojní montovanou součást složenou z více částí. Součást je složena ze skleněné desky, plastového těsnění, LED displeje a plochého kabelu. CEE je schopen provést korekci obrazu a ten je pak již připraven ke klasifikaci. Obr. (6.1).20 ukazuje diagram možných kombinací mocnin korekčního polynomu. Je zřejmé, že pro malá zkraslení obrazu je dostačující použít polynom se dvěma nebo třemi mocninnými členy a výsledek je stejný, jako když je použito všech sedm členů korekčního polynomu. Díky tomu může být korekční proces, resp. výpočet výrazně rychlejší – viz Obr. (6.1).6.

Na obrázcích (6.1).22 A, B, C, D, E, F, G, H, I je zobrazen výsledek práce navrženého CEE (**jDE+jDE**) pro šedotónový obraz (Wikipedia 2020b). Pro vyjádření kvality korekce bylo využito koeficientů MSE (Mean Squaree Error) (6.1.17) a PSNR (Peak Signal-to-Noise Ratio) (6.1.18) – viz také (Wikipedia 2020a). Rovnice pro výpočet MSE a PSNR jsou vyjádřeny jako:

$$MSE = \frac{1}{nm} \sum_{i=0}^{i=n} \sum_{j=0}^{j=m} [I_{original}(i,j) - I_{corrected}(i,j)]^2 \quad (6.1.17)$$

$$PSNR = 20 * \log_{10} \left(\frac{MAX}{\sqrt{MSE}} \right) \quad (6.1.18)$$

kde $n = I_w - 1$, $m = I_h - 1$, I_w je šířka obrazu, I_h -výška obrazu, $I_{original}$ je obraz, který není zatížen chybou distorze, $I_{corrected}$ je korigovaný obraz, MAX je max. úroveň šedi v šedotónovém obraze a v našem případě je to $MAX = 255$; tj. 8bit šedotónový obraz., $I_{...}(i,j)$ reprezentuje úroveň odstínu šedi v obraze v rozsahu $\langle 0,255 \rangle$ pro libovolný pixel na pozici (i,j) .

Velmi dobře známý obraz Leny (Po 2020) byl zvolen pro tyto experimenty. Experiment probíhal tak, že ideální obraz Lena (Po 2020) (2592 x 1944 x 8 bit šedotónový obraz, 256 odstínů šedi) byl uměle zatížen soudkovou vadou s využitím polynomu – viz \mathcal{F} (6.1.12) a také Obr. (6.1).22I – polynomy 1, 3, 5, 7 korespondují s obrazy na Obr. (6.1).22A, B, C, D. Byl také zvolen kalibrační obraz CD_B – viz Obr. (6.1).5B. Tímto způsobem byly získány 4 různé obrazy I''_D zatížené vadou soudkovitosti a současně vhodné pro zpracování s využitím CEE. Všechny 4 kalibrační obrazy I''_D byly postupně (jeden po druhém) vloženy do CEE (**jDE+jDE**) a CEE našel/odhadl nejlepší řešení resp. koeficienty korekčního polynomu C_f – polynom (6.1.8) o stupni 7. Získané hodnoty – viz Obr. (6.1).22I polynomů č. 2, 4, 6, 8, byly použity pro korekci obrazů Leny s využitím polynomiálního korekčního Algoritmu (6.1).4 (Brown 1956) (back-mapping schéma) a byl získán obraz prostý soudkové vady – viz Obr. (6.1).22E, F, G, H. Zkorigované obrazy Leny a korespondující vypočtené hodnoty MSE, PSNR koeficientů jsou zobrazeny na Obr. (6.1).22E, F, G, H. CEE běžel přibližně 1169 minut pro každý obraz. Na Obr. (6.1).22B, F je záměrně ukázaná poněkud horší konvergence a tím pádem jsou i výsledné hodnoty koeficientů MSE, PSNR vyšší; MSE=111.05; PSNR=27.67dB. Takovýto horší výsledek občas nastane právě díky využití EA, ale pokud je výpočet opakován a jsou zvoleny lepší pracovní parametry EA neboli hodnoty $N_{pop}^{jDE_F1}, G_{en}^{jDE_F1}, N_{pop}^{jDE_F2}, G_{en}^{jDE_F2}$, je získán výrazně lepší výsledek. Výrazně vyšší hodnoty pro všechny parametry $N_{pop}^{jDE_F1}, G_{en}^{jDE_F1}$ and $N_{pop}^{jDE_F2}, G_{en}^{jDE_F2}$ – viz Obr. (6.1).4 – nebo jen pro některé parametry.

6.1.7 Diskuze

Jak již bylo zmíněno v předcházejících kapitolách, v (Asari et al. 1999) je navržen iterativní algoritmus, který vychází z dřívějších výzkumů (Smith et al. 1992; Haneishi et al. 1995). (Asari et al. 1999) prozkoumal a navrhl korekční algoritmus, který byl schopen pracovat s polynomem až do stupně 5 a jako vhodný optimalizátor zvolil metody nejmenších čtverců (LSQ). CEE pracuje s polynomy stupně až 7. Je samozřejmě možné metodu rozšířit tak, aby pracovala i s polynomy vyššího stupně, ale výpočetní a s tím spojené časové nároky např. při využívání polynomů stupně 12 jsou značné. Výkonnost (LSQ) v porovnání s EA (zde CEE) je diskutována a analyzována v (Moravec 2015; Moravec & Pošík 2014a,b). Obzvláště pak (Moravec 2015) předkládá hlubší analýzu účinnosti navržených algoritmů, časové nároky a chování LSQ metod v porovnání s EA – účelová funkce má identické vlastnosti jako účelová funkce popsána v (6.1.14), (6.1.15) – viz Obr. (6.1).7. EA zde vykázaly výrazně lepší výsledky. Překvapivě stejných výsledků bylo dosaženo i v práci (Vahdat et al. 2007). Veškerý dříve presentovaný výzkum výrazně přispěl při sestavování navrženého CEE a podal nezbytné odpovědi na otázky, zda se EA hodí nebo nehodí k řešení dané úlohy a zda mohou soupeřit s klasickými metodami založenými na estimátoru, který využívá (LSQ) jak je uvedeno např. v (Asari et al. 1999). Odpověď je samozřejmě jednoznačná: ANO, evoluční algoritmy mohou být využity pro řešení daného problému a podávají skvělé výsledky.

Při experimentech byla použita řada evolučních algoritmů. Je poměrně paradoxní, že všechny testované EA je možné využít k řešení dané úlohy viz Obr. (6.1).9 - (6.1).13 a např. také (Yang and Illingworth 1994; Gao et al. 2008; Farga 2009; Song et al. 2009; Gao et al. 2009; Li et al. 2015). Některé z testovaných EA však mají výrazně nižší účinnost. Pokud jsou takové algoritmy použity, trvá dosažení očekávaného výsledku delší dobu nebo výsledek nemusí být optimální nebo může být dosažen naprosto nevyhovující výsledek. EA prošly poměrně dlouhou cestou od dob svého vzniku (Metropolis & Ulam 1949; Barricelli 1962, 1963; Holland 1962). Avšak teprve v posledních 2 dekadách zaznamenaly masivní rozvoj díky široké dostupnosti výpočetní techniky. V předkládané studii jsou využity některé EA, které byly navrženy právě v posledních dvou dekadách a které zároveň poskytují dobré výsledky při řešení daného typu úloh viz Obr. (6.1).7 a (6.1.14), (6.1.15). Skutečnost, že starší optimalizační postupy tzv. „Gradient Descent Algorithms“ (GDA) poskytují výrazně horší výsledky je popsán a analyzován již v (Yang & Illingworth 1994). Detailní popis výhod a nevýhod je pak uveden v (Salomon 1998). (Fraga 2009) použil algoritmus Diferenciální Evoluce při kalibraci kamery. DE však neposkytuje ty nejlepší možné výsledky s ohledem na současný stav poznání v oblasti EA. V práci (Brest et al. 2006) a později v (Brest & Maučec 2011) byl publikován nový druh EA, který pro některé typy úloh poskytuje výrazně lepší výsledky – s ohledem na počet selhání a dosaženou přesnost výsledku. Různých modifikací algoritmu DE (Storn & Price 1997) bylo od roku 1997 publikováno obrovské množství. Ne všechny se však na řešení daného problému hodí. Bohužel obecně neexistuje nějaká „rovnice“ nebo „předpis“, který by jednoznačně definoval, který EA je nejlepší pro řešení dané úlohy. Striktně zde platí „No free lunch theorem“ (Wolpert & Macready 1995). Lze tedy říci, že všechny dříve navržené postupy, které využívají algoritmus DE jako např. (Li et al. 2015), budou vždy poskytovat horší výsledky, než kdyby byl použit novější optimalizátor jDE. Stejně pravidlo platí i pro (Yang and Illingworth 1994; Gao et al. 2008; Farga 2009; Song et al. 2009; Gao et al. 2009). S ohledem na stále platné výsledky výzkumu v (Eiben et al. 1999; Gamperle et al. 2002; Liu & Lampinen 2002; Liu & Lampinen 2005; Smit & Eiben 2009; Smit & Eiben 2010a,b; Das & Suganthan 2010) však nelze vyloučit, že původní algoritmus DE (Storn & Price 1997; Price 1999) může být stejně výkonný jako jDE pouhou úpravou odchylkového vektoru, který však doposud nebyl objeven. SGA je také hojně využívaným EA pro řešení problému kalibrace kamery viz např. (Ji & Zhang 2001). (Ji & Zhang 2001) na základě svého výzkumu popsali, že „SGA svou výkonností překonává klasické fotogrammetrické metody...“. Je nutné brát v úvahu, že SGA se pro řešení dané úlohy hodí snad nejméně za všech testovaných EA a to z důvodu, že často uvázne v lokálním optimu; jinými slovy selže. Přesto však SGA podává výrazně lepší výsledky než např. algoritmus popsáný v (Tsai 1987).

Cílem předkládané studie v této kapitole bylo najít nejvhodnější evoluční optimalizátor, který by byl schopen řešit danou úlohu co nejefektivněji s ohledem na state-of-the-art v oblasti evolučních optimalizátorů v současné době.

6.1.8 Závěr

Distorze obrazu ať už soudková nebo polštářová značně narušuje geometrii zobrazených objektů. Jedná se tedy o nežádoucí aberaci obrazu. Pokud je ohnisková vzdálenost konstantní po celou dobu provozu zařízení, jako je tomu v našem případě – biometrický scanner, je dostačující nalézt správné koeficienty pro korekční polynom a každý snímaný obraz upravit s pomocí těchto koeficientů. Metoda, která zde byla navržena, je obvykle označována jako tzv. ‘pasivní’ metoda pro korekci obrazu při konstantní ohniskové vzdálenosti. Je samozřejmě možné ji rozšířit např. tak, aby matice kalibračních značek byla nahrazena maticí s LED diodami se zpětnou vazbou do počítače. Takový systém lze označit jako ‘aktivní’ korekční systém. V případě rozšíření o laserový dálkoměr by pak systém korekce byl kompletní a mohl by pracovat zcela automaticky.

Navržený CEE a především použité EA poskytují stabilní a precizní metodu jak provádět korekci distorze. Nevýhodou EA bývá obvykle to, že u nich není zaručená stoprocentní konvergence za všech okolností. CEE tedy může při odhadu koeficientů selhat a pak je nutné výpočet opakovat. Dobrý výsledek při opakovaném výpočtu dosáhnout lze. Zvýší se tím pouze časová náročnost. Na ověření funkčnosti navržených CEE bylo provedeno obrovské množství experimentů a ani jednou nedošlo k tomu, že by použitý EA, resp. CEE zcela selhal. Použité EA patří v současné době mezi skupinu nejvýkonnějších při řešení určitého okruhu úloh.

V presentovaném článku byly navrženy a otestovány 3 typy CEE. Typ jDE+jDE prokázal v testech přesnou konvergenci a minimum ‘nedokonalých’ výsledků ovšem za cenu velmi pomalé konvergence. Naopak typ CMA+CMA prokázal schopnost rychlé konvergence, avšak za cenu, že se ne vždy podaří dosáhnout optimálního výsledku, resp. primární optimalizátor sklouzne do nějakého lokálního optima a pak je nutné výpočet opakovat. Výhodou však je, že pro tento typ CEE nebylo pozorováno totální selhání. Typicky pro sestavení CEE CMA+CMA s koeficienty $\alpha=0.67$ je možné dosáhnout nějakou hodnotu fitness větší než 1000pixelů v cca 1/5 případů. Správná hodnota je cca 995-999. To je daň za rychlost a hranice, kterou v současné době a s daným vybavením lze jen obtížně překročit. Typ jDE+CMA se pohybuje někde na pomezí jDE+jDE a CMA+CMA, ale i tak poskytuje dobré výsledky.

LITERATURA

- Asari K.V. 2004. Design of an efficient VLSI architecture for non-linear spatial warping of wide-angle camera images. *Journal of System Architecture*. vol. 50, no. 12, pp. 743-755, 2004.
- Asari K.V., Kumar S., Radhakrishnan D. 1999. A new approach for nonlinear distortion correction in endoscopic images based on least squares estimation. *IEEE Transaction on medical imaging*. vol. 18, no. 4, pp. 345-354.
- Auger A., Hansen N. 2005. A Restart CMA Evolution Strategy With Increasing Population Size. In: *IEEE Congress on Evolutionary Computation*. CEC 2005, pp. 1769-1776.
- Auger A., Hansen N., Zerpa P., Ros J.M.R., Schoenauer M. 2009. Experimental comparisons of several derivative free optimization algorithms. In: *Experimental Algorithms, Lecture Notes in Computer Science*, Jan Vahrenhold ed., Springer Heidelberg, Berlin 2009.
- Barricelli N.A. 1962. *Numerical testing of evolution theories: Part I Theoretical introduction and basic tests*. Acta Biotheoretica, vol. 16, no. 1-2, pp. 69-98. ISSN 0001-5342.
- Barricelli N. 1963. *Numerical testing of evolution theories, Part II: Preliminary tests of performance, symbiogenesis and terrestrial life*. Department of Biology, Division of Molecular Biology, Vanderbilt University, Nashville, Tennessee, Acta Biotheoretica, Springer Netherlands, ISSN 0001-5342.
- Basu A., Licardie S. 1995. Alternative models for fish-eye lenses. *Pattern recognition letters*. vol. 16, pp. 433-441.
- Betelgeuse 2020. [online]. Betelgeuse LLC. ©2010-2015 [cit. 20.5.2020]. Dostupné z: http://www.softwareandfinance.com/Cpp/FAQ_Floating_Point.html
- Brest J., Greiner S., Bošković B., Mernik M., Žumer V. 2006. Selfadapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*. vol. 10, no. 6, pp. 646-657.
- Brest J., Bošković B., Greiner S., Žumer V., Maučec M.S. 2007. Performance comparison of self-adaptive and adaptive differential evolution algorithms. *Soft computing*. vol. 11, no. 7, pp. 617-629.
- Brest J., Maučec M.S. 2010. Self-adaptive differential evolution algorithm using population size reduction and three strategies, *Soft computing*. vol. 15, no. 11, pp. 2157-2174.
- Brest J., Zamuda A., Fister I., Maucec M.S. 2010. Large Scale Global Optimization using Self-adaptive Differential Evolution Algorithm. In: *IEEE World Congress on Computational Intelligence*, vol. 1, pp. 1-8.
- Brito J.H., Angst R., Koser K., Pollefeys M. 2013. Radial Distortion Self-Calibration. In: *CVPR 2013*. pp. 1368-1375.
- Brown D.C. 1956. The simultaneous determination of the orientation and lens distortion of a photogrammetric camera, Air Force Missile Test Center Report no 50-20, Patrick AFB, Florida.
- Brown D.C. 1964. An advanced plate reduction for photogrammetric cameras, AF Cambridge Research Laboratories technical report no 64-40, Patrick AFB, Florida.
- Brown D.C. 1965. Advanced methods for the calibration of metric cameras, Technical report, U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia 22060.
- Brown D.C. 1966. Decentering distortion of lenses, D.Brown Associates, Inc. Eau Gallie, Florida.
- Brown C.D. 1971. Close range camera calibration, In: *Symposium on close range photogrammetry*, Urbana, Illinois, DBA Systems, Inc. Melbourne.
- Butler D.A. 1991. A distortion-correction scheme for industrial machine-vision applications. *IEEE Transactions on Robotics and Automation*. vol. 7, no. 4, pp. 546 – 551,
- Clarke T.A., Fryer J.G. 1998. The Development of Camera Calibration Methods and Models. *The photogrammetric record*. vol. 16, no. 91, pp. 51-66.
- Conrady A.E. 1919 Decentred lens-systems. *Monthly Notices of the Royal Astronomical Society*. Vol. 79, p.384-390.

- Coroneos R.M., Hopkins D.A., Berke L. 1996. A cascade optimization strategy for solution of difficult multidisciplinary design problems, *NASA Technical Memorandum 107189*, AIAA-96-1628.
- Cucchiara R., Grana C., Prati A. Vezzani R. 2003. A Hough Transform-based method for Radial Lens Distortion Correction, In: *12th International Conference on Image Analysis and Processing*, pp. 182-187.
- Dawson B. 2020 [online]. Dawson B. ©2012 [cit. 20.5.2020]. Dostupné z: <https://randomascii.wordpress.com/2012/02/25/comparing-floating-point-numbers-2012-edition/>
- Das A., Patil D. 2006. Efficient Measurement and Correction techniques for Lens Distortion, Technical report, EE362 Project CANON co., Stanford Center for Image Systems Engineering.
- Das S., Suganthan P.N. 2010. Differential Evolution: A Survey of the State-of-the-Art. *IEEE Transaction on Evolutionary Computation*. vol. 15, no 1., pp. 4-31.
- Devernay F. 1995. A non-maxima suppression method for edge detection with sub-pixel accuracy. Technical report, INRIA.
- Devernay F., Faugeras O. 2001. Straight Lines Have to Be Straight. *Machine Vision and Applications*. vol. 13, no. 1 pp. 14-24.
- Dhane P., Kutty K., Bangadkar S. 2012. A Generic Non-Linear Method for Fisheye Correction. *International Journal of Computer Applications*. vol. 51, no. 10, pp. 58-65.
- Dollar, P., Welinder, P., Perona, P. 2010. Cascaded Pose Regression. In: *IEEE Conference on Computer Vision and Pattern Recognition*. pp. 1078-1085.
- Eiben, A.E., Hinterding R., Michalewicz Z. 1999. Parameter control in evolutionary algorithms. *IEEE Transaction on evolutionary computation*. vol. 3, no. 2, pp. 124-141.
- Fraga L.G. 2009. Self-calibration from Planes Using Differential Evolution. In: *14th Iberoamerican Conference on Pattern Recognition*. vol. XIII, pp. 724-731.
- Fernandes J.C.A., Ferreira M.J.O., Neves J.A.B.C., Couto C.A.C. 1997. Fast correction of lens distortion for image applications. In: *IEEE International Symposium on Industrial Electronics*. vol. 2, pp.708-712.
- Fischer R.F., Tadic B. 2000. *Optical System Design*. McGraw Hill, p. 599. ISBN-13: 978-0071349161.
- Fitzgibbon A.W. 2001. Simultaneous linear estimation of multiple view geometry and lens distortion. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. vol. 1, pp. 125-132.
- Floudas Ch.A., Pardalos P. M. 2009. *Encyclopedia of Optimization*. Springer, 2nd ed., p. 4626, 7 volumes. ISBN 978-0-387-74759-0.
- Fraser C.S., Al-Ajlouni S. 2006. Zoom-Dependent Camera Calibration in Digital Close-Range Photogrammetry. *Photogrammetric Engineering & Remote Sensing*. vol. 72, no. 9, pp. 1017-1026.
- Freund Y., Schapire R.E. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*. vol. 55, no. 1, pp. 119-139.
- Fryer J.G. 1986. Lens distortion for close range photogrammetry. *Photogrammetric Engineering and Remote Sensing*. vol. 52, Jan. 1986, p. 51-58.
- Gamperle R., Muller S.D., Koumoutsakos P. 2002. A parameter study for differential evolution. In: *WSEAS International Conference on Advances in Intelligent Systems*. pp. 293-298.
- Gao H., Ligong S., Li B., Chengdong W., Zhou C. 2008. An improved two-stage camera calibration method based on evolution calculation. In: *7th World Congress on Intelligent Control and Automation*. Chongqing, pp. 8471 - 8476.
- Gao H., Niu B., Yu Y., Chen L. 2009. An Improved Two-Stage Camera Calibration Method Based on Particle Swarm Optimization. In: *Proceedings of the 5th International Conference on Intelligent Computing*. vol. 5755, pp. 804-813.
- Goldberg D.E. 1987. Simple genetic algorithms and the minimal deceptive problem. In: *Genetic Algorithms and Simulated Annealing*, London, Pitman, pp. 74-88.
- Goldberg D.E. 1989. *Genetic Algorithms in search, optimization, and machine learning*. Addison-Wesley Publishing Corporation, ISBN-13: 978-0201157673.

- Guermeur P., Louchet J. 2003. An evolutionary algorithm for camera calibration. In: *ICRODIC 2003 conference*, Rethymnon, Crete, pp. 799-804.
- Haneishi H., Y. Yagihashi Y., Miyake Y. 1995. A new method for distortion correction of electronic endoscope images. *IEEE Transactions on Medical Imaging*. vol. 14, no. 3, pp. 548–555.
- Hansen, N., Ostermeier A. 1996. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In: *IEEE International Conference on Evolutionary Computation*. pp. 312-317.
- Hansen N. 2013 [online] [cit. 8.4.2020]. Dostupné z: <http://en.wikipedia.org/wiki/CMA-ES>
- Hansen N. 2020. [online]. Hansen N. ©2020 [cit. 20.5.2020]. Dostupné z: <http://www.lri.fr/~hansen/>
- Hansen N. 2006. The CMA Evolution Strategy: A Comparing Review. In: *J.A. Lozano, P. Larrañaga, I. Inza and E. Bengoetxea (Eds.). Towards a new evolutionary computation. Advances in estimation of distribution algorithms*. Springer, pp. 75-102.
- Hansen N., Ostermeier A., Gawelczyk A. 1995. On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation. In: *L. Eshelman (Ed.), Proceedings of the Sixth International Conference on Genetic Algorithms*, Pittsburgh, pp. 57-64. Morgan Kaufmann.
- Hansen N., Ostermeier A. 1996. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In: *IEEE International Conference on Evolutionary Computation*. pp. 312-317.
- Hansen N., Ostermeier A. 1997. Convergence properties of evolution strategies with the derandomized covariance matrix adaptation: The $(\mu/\mu\lambda)$ -ES. In: *EUFIT'97, 5th Europ. Congr. on Intelligent Techniques and Soft Computing*. pp. 650-654.
- Hansen N., Ostermeier A. 2001, Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*. vol. 9, no. 2, pp. 159-195.
- Hansen N., Müller S.D., Koumoutsakos P. 2003. Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation*. vol. 11, no. 1, pp. 1-18.
- Hansen N., Kern S. 2004. Evaluating the CMA Evolution Strategy on Multimodal Test Functions. In: *8th International Conference on Parallel Problem Solving from Nature PPSN VIII*, pp. 282-291.
- Holland J.H. 1962. Outline for a Logical Theory of Adaptive Systems. *Journal of the ACM*. vol. 9, no. 3, pp. 297-314.
- Holland J.H. 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, p. 232. ISBN: 9780262581110.
- Holland J.H. 1992. *Adaptation in natural and artificial systems: an introductory analysis with application to biology, control, and artificial intelligence*. Cambridge, MIT Press. p. 211. ISBN: 9780262275552.
- Huges C., Jones. E., Glavin M. and Denny P., 2009. Validation of polynomial-based equidistance fish-eye models. In: *Signals and systems conference*. IET Irish, Dublin, pp.1-6.
- Chen S.L., Huang H.Y., Luo Ch.h. 2011. Time Multiplexed VLSI Architecture for Real-Time Barrel Distortion Correction in Video-Endoscopic Images. *IEEE Transactions on circuits and systems for video technology*. vol. 21, no. 11, pp. 1612-1621.
- Chevallier L., Vigouroux J.R., Goguy A., Ozerov A. 2013. Facial Landmarks Localization Estimation by Cascaded Boosted Regression. In: *International Conference on Computer Vision Theory and Applications*. pp. 513-519.
- Igel Ch., Suttorp T., Hansen N. 2006. A computational efficient covariance matrix update and a (1+1)-CMA for evolution strategies. In: *GECCO '06 Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pp. 453-460.
- Jastrebski G.A., Arnold D.V. 2006. Improving evolution strategies through active covariance matrix adaptation. In: *IEEE Congress on Evolutionary Computation CEC 2006*. pp. 2814 – 2821.
- Jeught S.V., Buytaert J., Dirckx J. 2012. Real-time geometric lens distortion correction using a graphics processing unit. *Optical engineering*. vol. 51, pp. 1–5.
- Ji Q., Zhang Y. 2001 Camera Calibration with Genetic Algorithms. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*. vol. 31, No. 2, pp. 120-130.
- Kennedy J., Eberhart R.C. 1995. Particle swarm optimization. In: *IEEE International Conference on Neural Networks*. vol. 4, pp. 1942-1948.

- Kim W., Kim Ch. 2009. An efficient correction method of wide-angle lens distortion for surveillance systems. In: *IEEE International Symposium on Circuits and Systems*. pp. 3206-3209.
- Kimichi Y.M. 2020. [online]. Kimichi Y.M. ©2005-2020 [cit. 20.5.2020]. Dostupné z: <http://www.cs.technion.ac.il/users/yechiel/c++-faq/floating-point-arith.html>
- Lenz R.K., Tsai R.Y. 1988. Techniques for calibration of the scale factor and image center for high accuracy 3-D machine vision metrology. *IEEE Transactions on pattern analysis and machine intelligence*. vol. 10, pp. 713–720.
- Li D., Gen L., Yuying S., Minrui F., Huosheng H. 2015. A novel camera calibration technique based on differential evolution particle swarm optimization algorithm. *Neurocomputing*. vol. 174/A, pp. 456-465.
- Liu J., Lampinen J. 2002. On setting the control parameter of the differential evolution method. In: *8th International Conference Soft Computing*. pp. 11-18.
- Liu J., Lampinen J. 2005. A Fuzzy Adaptive Differential Evolution Algorithm. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*. vol. 9, no. 6, pp. 448-462.
- Magill A.A. 1955. Variation in distortion with magnification. *Journal of Research of the National Bureau of Standards*. vol. 54, no. 3, pp. 135–142.
- McGlone Ch. 2004. *Manual of photogrammetry*. American Society for Photogrammetry and Remote Sensing, 5th edition p. 1151. ISBN-13: 978-1570830716.
- Melo R., Barreto J.P., Falcao G. 2011. A New Solution for Camera Calibration and Real-Time Image Distortion Correction in Medical Endoscopy - Initial Technical Evaluation. *IEEE Transaction on biomedical engineering*. vol. 59, no.3, pp. 634-644.
- Melo R., Falcao G., Barreto J.P. 2012. Real-time HD image distortion correction in heterogeneous parallel computing systems using efficient memory access patterns, *Journal of real-time image processing*, vol. 11, no. 1, pp 83–91.
- Metropolis N., Ulam S. 1949. The Monte Carlo method. *Journal of the American Statistical Association*. vol. 44, pp. 335-341.
- Montes E.M., Reyes J.V., Coello C.A.C. 2006. A Comparative Study of Differential Evolution Variants for Global Optimization. In: *GECCO '06 Proceedings of the 8th annual conference on Genetic and evolutionary computation*. pp. 485-492.
- Moravec J. 2015. A Comparative Study: L1-norm vs. L2-norm, point-to-point vs. point-to-line Metric, Evolutionary Computation vs. Gradient Search. *Applied Artificial Intelligence*. Special edition on robotics research, in print, vol. 29, no. 2, pp. 164 – 210.
- Moravec J., Pošík P. 2014a. A Comparative Study: The Effect of the Perturbation Vector Type in the Differential Evolution Algorithm on the Accuracy of Robot Pose and Heading Estimation. *Evolutionary Intelligence Journal*. vol. 6, no. 3, pp. 171-191.
- Moravec J., Pošík P. 2014b. Global Robot Localization Under Noise Stress Utilizing EA Methods and Semi-semantic Classification of a Known Environment. *Applied Artificial Intelligence*. vol. 28, no. 4, pp. 360 – 417.
- Ngo H.T., Asari K.V. 2005. A pipelined architecture for real-time correction of barrel distortion in wide-angle camera images. *IEEE Transaction on Circuits Systems and Video Technology*. vol.15, no. 3, pp. 436-444.
- Pardalos P.M. 2020 [online]. Dr. Panos M. Pardalos ©2020 [cit. 20.5.2020]. Dostupné z: <http://www.ise.ufl.edu/pardalos>
- Patnaik S.N., Coroneos R.M., Hopkins D.A., Berke L. 1996. A Cascade Optimization Strategy for Solution of Difficult Multidisciplinary Design Problems, 37th Structures, Structural Dynamics and Materials Conference, NASA Lewis Research Center, Cleveland, p.7.
- Pierre S.A., Ferrandino M.N., Simmons W.N., Fernandez C., Zhong P., Albala D.M., Preminger G.M. 2009. High definition laparoscopy: objective assessment of performance characteristics and comparison with standard laparoscopy. *Journal of Endourology*. vol. 23, no. 3, pp. 523–528.
- Po L.M. 2020. [online]. Li-Man Po ©2014-2019 [cit. 20.5.2020]. Dostupné z: <http://www.ee.cityu.edu.hk/~lmpo/lenna/Lenna97.html>

- Pošik P., Huyer W. 2012. Restarted Local Search Algorithms for Continuous Black Box Optimization. *Evolutionary Computation*. vol. 20, pp 575-607.
- Pošik P., Huyer W., Pál L. 2012. A Comparison of Global Search Algorithms for Continuous Black Box Optimization. *Evolutionary Computation*. vol. 20, pp 509-541.
- Pošik P., Kubalík J. 2012. Experimental Comparison of Six Population-Based Algorithms for Continuous Black Box Optimization. *Evolutionary Computation*. vol. 20, pp. 483-508.
- Prescott B., McLean G.I. 1997. Line based correction of radial lens distortion. *Graphical models and image processing*. vol. 59, no. 1, pp. 39-47.
- Price K., Corne D., Dorigo M., Glover F. 1999. *An Introduction to Differential Evolution*. Eds. London, U.K. McGraw-Hill, pp. 79-108.
- Qin A.K., Suganthan P.N. 2005. Self-adaptive differential evolution algorithm for numerical optimization, In: *IEEE Congress on Evolutionary Computation*. vol. 2, pp. 1785-1791.
- Qin A.K., Huang V.L., Suganthan P.N. 2009. Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization. *IEEE Transactions on evolutionary computation*. vol. 13, no 2, pp. 398-417.
- Remondino F., Fraser C. 2006. Digital camera calibration methods: considerations and comparisons. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences.*, vol. XXXVI, no 5.
- Romero L., Gomez C. 2007. Correcting Radial Distortion of Cameras with Wide Angle Lens Using Point Correspondences, *Scene Reconstruction, Pose Estimation and Tracking*. InTech, Rustam Stolkin ed., pp.530, [cit. 20.5.2020]. DOI: 10.5772/4927
- Salomon R. 1998. Evolutionary algorithm and gradient search: similarities and differences. *IEEE Transaction on evolutionary computation*. vol. 2, no.2, pp. 45-55.
- Sasián J. 2012. *Introduction to Aberrations in Optical Imaging Systems*. Cambridge Press, p. 258. ISBN 9780511795183.
- Shah S., Aggarwal J.K. 1994. A simple calibration procedure for fish-eye (high distortion) lens camera. In: *IEEE International Conference on Robotics and Automation*. vol. 4, pp. 3422 – 3427.
- Schwalbe E. 2005. Geometric modeling and calibration of fisheye camera systems, *SPRS Archives*, vol. XXXVI-5/W8.
- Schwartz E.L. 1980. Computational anatomy and functional architecture of striate cortex: a spatial mapping approach to perceptual coding. *Vision Research*. vol. 20, pp. 645-669.
- Slama C.C. 1980. *Manual of Photogrammetry*. American Society of Photogrammetry, fourth edition, p. 1056. ISBN-0-937294-01-2.
- Smit S.K., Eiben A.E. 2009. Comparing Parameter Tuning Methods for Evolutionary Algorithms. In: *IEEE Congress on Evolutionary Computation*. pp. 399-406.
- Smit S.K., Eiben A.E. 2010a. Parameter Tuning of Evolutionary Algorithms: Generalist vs. Specialist. *Applications of Evolutionary Computation*. vol. 6024, pp. 542-551.
- Smit S.K., Eiben A.E. 2010b. Parameter Tuning for Configuring and Analyzing Evolutionary Algorithms. *Swarm and Evolutionary Computation*. vol. 1, no. 1, pp. 19-31.
- Smith W.E., Vakili N., Maislin S.A. 1992. Correction of distortion in endoscopic images. *IEEE Transaction on Medical Imaging*. vol. 11, pp. 117–122.
- Solar J.R., Arenas M., Verschae R., Loncomilla P. 2010. Visual Detection of Legged Robots and its Application to Robot Soccer Playing and Refereeing. *International Journal of Humanoid Robotics*. vol. 7, no. 4, pp. 669.
- Song X., Yang B., Feng Z., Xu T., Zhu D., Jiang Y. 2009. Camera Calibration Based on Particle Swarm Optimization. In: *2nd International Congress on Image and Signal Processing*. Tianjin, pp. 1-5.
- Storn R., Price K. 1997. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*. vol. 11, pp. 341-359.
- Sun H., Zhang Y., Luo F. 2008. A Novel Approach for Nonlinear Distortion Correction of Industrial Endoscope Images. In: *17th World Conference on Nondestructive Testing*, Shanghai China.

- SunMicrosystems 2020 [online]. Sun Microsystems, Inc. ©2012 [cit. 20.5.2020]. Dostupné z: http://download.oracle.com/docs/cd/E19957-01/806-3568/ncg_goldberg.html
- Susnjak T., Barczak A., Reyes, N., Hawick K. 2012. Multiclass Cascades for Ensemble-based Boosting algorithms. *Frontiers in Artificial Intelligence and Applications*. vol. 241, pp. 330-335.
- Štěpán P. 2001. *Inner representation of an environment for autonomous mobile robots*. Ph.D Thesis, CTU Prague, [cit. 20.5.2020]. <http://labe.felk.cvut.cz/~stepan/> nebo <http://labe.felk.cvut.cz/~kulich/>
- Tang R. 2012. A rigorous and flexible calibration method for digital airborne camera systems. In: *XXII Congress of the International Society for Photogrammetry*. p. 6.
- Tang Z., Gioi R.G., Monasse P., Morel J.M. 2012. High-precision camera distortion measurements with a “calibration harp”. *Computer Vision and Pattern Recognition*. vol. 29, no. 10, pp. 2134-2143.
- Tommaselli A.M.G., Marcato J.J., Telles S.S.S. 2012. Camera calibration using straight lines: assessment of a model based on plane equivalence. *The Photogrammetric journal of Finalnd*. vol. 23, no. 1, p. 11.
- Tsai R.Y. 1987a. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal on Robotics and Automation*. vol. RA-3, pp. 323–344.
- Tsai R.Y. 1987b. A new approach for nonlinear distortion correction in endoscopic images based on least squares estimation. *IEEE Journal of robotics and automaion*. vol. RA-3, no. 4, pp. 323-344.
- Thormahlen T., Broszio H., Wassermann I. 2003. Robust line-based calibration of lens distortion from a single view. In: *Proceedings of Mirage 2003*. INRIA Rocquencourt, France, pp. 105–112.
- Urfahoglu O. 2004. Robust Estimation of Camera Rotation, Translation and Focal Length at High Outlier Rates. In: *1st Canadian Conference on Computer and Robot Vision*. pp. 464-471.
- Vahdat A.R., NourAshrafoddin N., Ghidary S.S. 2007. Mobile Robot Global Localization using Differential Evolution and Particle Swarm Optimization. In: *Proceedings of the IEEE congress on evolutionary computation CEC 2007*. pp. 25-28.
- Verschae R., Ruiz-del-Solar J. 2010. Coarse-To-Fine Multiclass Nested Cascades for Object Detection. In: *20th International Conference on Pattern Recognition*. pp. 344-347.
- Virendra N.M. 1991. *Aberration Theory Made Simple*, SPIE Press, p. 166, ISBN: 9780819488251.
- Wang A., Tianshuang Q., Shao L. 2009. A Simple Method of Radial Distortion Correction with Centre of Distortion Estimation. *Journal of Mathematics Imaging and Vision*. vol. 35, pp. 165-172.
- Wang S., Tseng Y.H. 2000. Automatic self-calibration of digital cameras for close-range photogrammetry. In: *Conferece IAPRS*. vol. XXXIII.
- Wang Y, Cai Z., Zhang Q. 2011. Differential Evolution with Composite Trial Vector Generation Strategies and Control Parameters. *IEEE Transactions on evolutionary computation*. vol 15. no. 1, pp. 55 – 66.
- Welford W.T. 1986. *Aberrations of Optical Systems*. CRC Press, p. 284. ISBN-13: 978-0852745649.
- Weng J., Cohen P., Herniou M. 1992. Camera calibration with distortion models and accuracy evaluation. *IEEE Transaction on pattern analysis and machine intelligence*. vol. 14, no. 10, pp. 965-980.
- Whitley D., Mathias K., Rana S., Dzubera J. 1996. Evaluating Evolutionary Algorithms. *Artificial Intelligence*. vol. 85, pp. 245–276.
- Wikipedie 2016. [online]. Wikipedie ©2020 [cit. 20.5.2020]. Dostupné z: <http://en.wikipedia.org/wiki/CMA-ES>
- Wikipedia 2020a. [online]. Wikipedia ©2020 [cit. 20.5.2020]. Dostupné z: https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio
- Wikipedia 2020b. [online]. Wikipedia ©2020 [cit. 20.5.2020]. Dostupné z: <https://en.wikipedia.org/wiki/Grayscale>
- Wikipedia 2020c. [online]. Wikipedia ©2020 [cit. 20.5.2020]. Dostupné z: https://en.wikipedia.org/wiki/Haar-like_feature

Wolpert D.H., Macready W.G. 1995. No free-lunch theorems for search, Technical Report 95-02-010, SantaFe Institute.

Yang D., Illingworth J. 1994. Calibrating a Robot Camera. In: *Proceedings of the British Machine Vision Conference*. pp. 1-10.

Zhang Ch., Helferty J.P., McLennan G., Higgins W.E. 2000. Nonlinear distortion correction in endoscopic video images. In: *International Conference on Image Processing*. vol. 2, pp. 439 – 442.

Zhang J., Sanderson A.C. 2009. JADE: Adaptive Differential Evolution with Optional External Archive. *IEEE Transactions on Evolutionary Computation*. vol. 13, no. 5, pp. 945 – 958.

6.2 Metoda korekce obrazu založená na algoritmu direct-mappingu

Každá optická soustava je do jisté míry zatížena množinou optických vad. Některé z nich jsou korigovány výrobcem optické soustavy, jiné je nutné korigovat dodatečně např. metodami pro korekci digitálního obrazu. Předkládaný článek popisuje metodu, která umožňuje korekci soudkové nebo polštářové distorze obrazu malého rozsahu metodou direct-mappingu. Navržená metoda využívá algoritmus adaptivní metaheuristiky, označovaný jako jDE a polynomiální reprezentaci zkreslení obrazu. Výhodou spojení direct-mappingu a adaptivní metaheuristiky je především velká rychlost nalezení základních korekčních koeficientů polynomu; v našem případě polynomu až stupně 7 a volitelnou kombinací stupňů polynomu. Nevýhodou direct-mappingu je nutnost rekonstrukce chybějících pixelů výsledného obrazu, která je zde velmi efektivně řešena opět s využitím optimalizátoru jDE. Aplikací navržené metody je možné získat kvalitní obraz prostý soudkové nebo polštářové distorze. Při zachování konstantní ohniskové vzdálenosti použitého objektivu zůstávají hodnoty koeficientů korekčního polynomu a polohy jednotlivých rekonstruovaných pixelů konstantní.

6.2.1 Úvod

Nechtěné vady optických soustav (Welford 1986) je možné korigovat mnoha různými způsoby. Převážná většina chyb je korigována již při návrhu a výrobě optické soustavy. Další vady jako např. distorze je možné korigovat v průběhu snímání obrazu, je-li použita digitální technologie zpracování. Korekci provádí buď procesor podle předem připraveného algoritmu např. v případě kamery, nebo programovatelné logické pole např. pro studiová záznamová zařízení, což může být výrazně rychlejší, ale méně flexibilní.

Základní metodologii korekce distorze obrazu, a to jak centrické, tak i necentrické nastínil již v roce 1919 Prof. A.E. Conrady (Conrady 1919). Později byly metody rozšířeny dalšími autory např. (Magil 1955; Brown 1956). Jako referenční v dané oblasti je často uvažována publikace Dunkana C. Browna (Brown 1971) založená na tzv. polynomiálním radiálním modelu, která je využívána dodnes. Kromě metod založených na polynomiálním modelu existují i metody založené na nepolynomiálním (non-polynomial) modelu např. (Kim & Kim 2009). Jejich výhodou je rychlejší výpočet, a proto je možné je využívat i u objektivů s proměnnou ohniskovou vzdáleností bez nároku na hardwarovou akceleraci. Nevýhodou je pak menší univerzálnost a také do jisté míry vazba na konkrétní typ optické soustavy, resp. konkrétní velikost distorze obrazu např. pro korekci obrazu u objektivů typu rybí oko (Fishe-Eye).

Použité metody pro korekci obrazu se dále dělí na tzv. back-mapping (Brown 1971) a direct-mapping (Cucchiara & Grana 2003). Rychlost výpočtu při známých koeficientech korekčního polynomu je u obou metod stejná. Výhodou první je to, že ve výsledném obraze nevznikají prázdná místa – chybějící pixely, jako následek nedokonalého mapování; to platí alespoň pro malé distorze obrazu. Druhá metoda poskytuje stejně dobře korigovaný obraz jako první, ale korekce je zde zatížena nepřesnostmi při mapování a ve výsledném obraze tak chybí značné množství informace – pixely, které se neobarví. Barvy chybějících pixelů je možné buď získat jako průměr barvy v okolních pixelech nebo pro malý stupeň korekčního polynomu (max. 3) vypočítat pomocí Cardanových polynomů – viz (Cucchiara & Grana 2003). Pokud je korekční polynom vyššího stupně (vyšší než 5), pak neexistuje žádná jednoduchá metoda, která by umožňovala nalézt přesnou korespondenci mezi pixely v obraze zatíženém distorzí a ve výsledném zkorigovaném obraze. Jedinou možností je použít metody numerické matematiky.

Součástí navržené metody pro odstranění distorze obrazu je také algoritmus z oblasti evolučních výpočetních technik autory označovaný jako jDE – viz (Brest et al. 2006). Jedná o pokročilý typ evolučního algoritmu odvozený od optimalizátoru diferenciální evoluce (Storn & Price 1997). jDE je velmi mladý, avšak velmi výkonný stochastický optimalizátor – metaheuristika, která byla presentována

teprve v roce 2006. Je použita jak pro zjištění koeficientů korekčního polynomu, tak také pro nalezení barvy chybějících pixelů ve finálním obraze. Metoda pro nezezení správné barvy pixelů využívá techniku inverzního inženýrství – vyhledává správnou pozici pixelu v obraze, který je zatížený distorzí s využitím koeficientů korekčního polynomu a barvu pixelu kopíruje do obrazu, který je korigovaný neboli bez distorze.

Navržená metoda pro kalibraci kamery bude dále označována jako DMAM (**D**irect **M**apping calibration algorithm using Adaptive **M**etaheuristic). Nejprve bude popsán teoretický základ metody a kompletní algoritmus DMAM a dále budou presentovány experimentální výsledky dosažené s využitím algoritmu DMAM.

6.2.2 Metoda pro korekci obrazu

Při sestavování zde presentovaného algoritmu pro korekci obrazu, který je zatížený distorzí, bylo čerpáno z následujících publikací: polynomiální algoritmus je převzat z (Conrady 1919; Magill 1955; Brown 1956; Brown 1971) a samozřejmě i (Welford 1986), úvaha o možnosti využít direkt-mapping metody pro dané účely je presentována např. v (Liu et al. 2014) a také (Cucchiara 2003), kde je současně využito i algoritmus SGA a nakonec byl použit pokročilý evoluční optimalizátor jDE je převzatý z (Brest et al. 2006; Storn & Price 1997) – viz Alg. (6.2).4. Brown (Brown 1956; 1971) použil ve své práci pro korekci distorze obrazu polynom \mathcal{F}_{odd} sestavený z lichých mocnin:

$$\mathcal{F}_{odd}(r_d) \equiv r_u = \sum_{n=2m+1, m \in \mathbb{N}^0}^{N_{odd}} c_n r_d^n = c_1 r_d^1 + c_3 r_d^3 + c_5 r_d^5 + \dots + c_{N_{odd}} r_d^{N_{odd}} \quad (6.2.1)$$

kde r_d je vzdálenost pixelu od středu obrazu u distorted image \mathbf{I}_D a r_u je vzdálenost pixelu od středu obrazu v undistorted image \mathbf{I}_U . Brownův algoritmus má velmi vysokou přesnost aproximace cca 97%. Pro korekci obrazu je pochopitelně možné využít takový polynom \mathcal{F} , který využívá jak sudé, tak i liché mocniny se stejnou nebo vyšší přesností – viz (Brown 1956; 1971). Tento přístup je aplikován v algoritmu DMAM. Polynom \mathcal{F} je pak definován jako:

$$\mathcal{F}(r_d) \equiv r_u = \sum_{n=1}^N c_n r_d^n = c_1 r_d^1 + c_2 r_d^2 + c_3 r_d^3 + c_4 r_d^4 + \dots + c_N r_d^N \quad (6.2.2)$$

Metoda DMAM je primárně navržena pro korekci soudkového nebo polštářového zkreslení obrazu získaného z biometrického scanneru – viz Obr. (6.2).2A,B, který je snímán fotoaparátem na krátkou vzdálenost. Je však možné ji využít pro odstranění soudkové i polštářové distorze malého rozsahu u libovolného obrazu.

Biometrický scanner používá klasický digitální fotoaparát CanonPowerShotS60, který je připevněn na horní části scanneru a je zde možnost adjustace v osách XYZ a rovině XY – viz Obr. (6.2).1C. Nejprve je získán barevný kalibrační obraz \mathbf{I}_D' , který je transformován na B&W obraz \mathbf{I}_D o stejném rozlišení – viz Obr. (6.2).2B. Obraz obsahuje množinu kalibračních značek \mathbf{M}_C , každá o rozměru cca 20x20 pixelů. S využitím znalosti o geometrickém středu množiny značek a vzájemné vzdálenosti jednotlivých značek na originálním kalibračním výkrese je ručně sestavena matice referenčních \mathbf{M}_R - ideální pozice značek v obraze bez distorze. Cílem algoritmu DMAM je najít takové hodnoty koeficientů $c_i, i \in \langle 1, N \rangle$ resp. vektor \mathbf{C}_f polynomu \mathcal{F} , který umožní správnou transformaci obrazu \mathbf{I}_D na \mathbf{I}_U . Výsledkem transformace je pak matice \mathbf{M}_F , která reprezentuje nejlepší odhad koeficientů c_i vektoru \mathbf{C}_f . Matice $\mathbf{M}_C, \mathbf{M}_R$ a \mathbf{M}_F jsou definovány jako:

$$\mathbf{M}_C = \begin{bmatrix} (\ddot{x}, \ddot{y})_{0,0} & \dots & (\ddot{x}, \ddot{y})_{0,j} \\ \dots & \dots & \dots \\ (\ddot{x}, \ddot{y})_{i,0} & \dots & (\ddot{x}, \ddot{y})_{i,j} \end{bmatrix}, \mathbf{M}_F = \begin{bmatrix} (\hat{x}, \hat{y})_{0,0} & \dots & (\hat{x}, \hat{y})_{0,j} \\ \dots & \dots & \dots \\ (\hat{x}, \hat{y})_{i,0} & \dots & (\hat{x}, \hat{y})_{i,j} \end{bmatrix}, \quad (6.2.3)$$

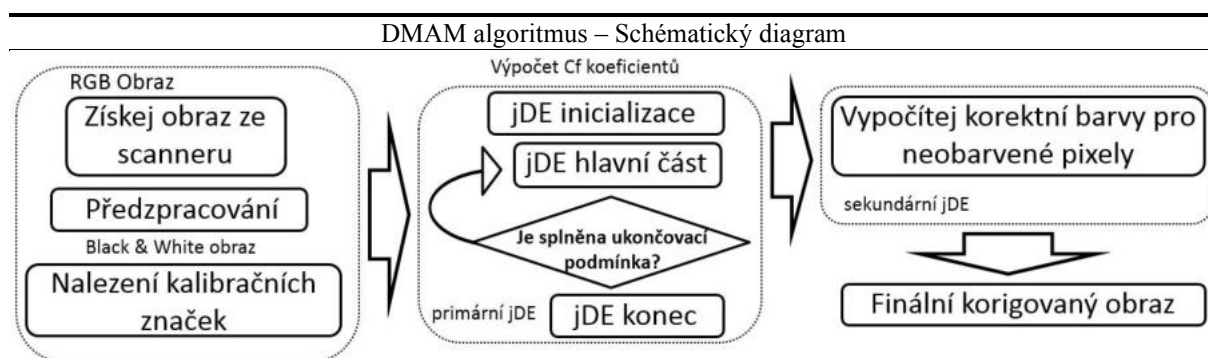
$$\mathbf{M}_R = \begin{bmatrix} (x, y)_{0,0} & \dots & (x, y)_{0,j} \\ \dots & \dots & \dots \\ (x, y)_{i,0} & \dots & (x, y)_{i,j} \end{bmatrix},$$

$$\hat{x}, \hat{y} \in \mathcal{H}_U; x, y \in \mathcal{H}_U; \ddot{x}, \ddot{y} \in \mathcal{H}_D$$

$$i \in \langle 0, I_w - 1 \rangle, j \in \langle 0, I_h - 1 \rangle, x, y \in \mathbb{N}^0; \hat{x}, \hat{y} \in \mathbb{R}^+; \ddot{x}, \ddot{y} \in \mathbb{R}^+,$$

$\mathcal{H}_U, \mathcal{H}_D$ -prostor souřadnic v obraze bez distorze a v obraze, který je zatížený distorzí.

a hodnoty (\ddot{x}, \ddot{y}) reprezentují pozice těžiště kalibračních značek, (\hat{x}, \hat{y}) finální odhad pozic po transformaci s využitím polynomu \mathcal{F} . Pro prvotní nalezení značek \mathbf{M}_C v obraze \mathbf{I}_D je využit algoritmus, který cyklicky hledá očekávaný počet pixelů (např. 350) ve čtvercové oblasti o velikosti 40x40 pixelů. Čtvercová oblast se postupně posouvá po celé oblasti \mathbf{I}_D pixel po pixelu.



Obrázek (6.2).1 Schématický diagram navrženého algoritmu DMAM. Jedná se sériové uspořádání kaskádového estimátoru bez zpětné vazby a bez využití metod inverzního inženýrství.

Navržená metoda DMAM je algoritmicky popsána na Alg. (6.2).1, (6.2).2, (6.2).3. Schématický diagram je na Obr. (6.2).1. Nejprve jsou s využitím optimalizátoru jDE označeného zde jako ‘primární jDE’ (v algoritmech je použit výraz ‘primary-jDE’) nalezeny správné korekční koeficienty c_i vektoru \mathbf{C}_f . Vektor \mathbf{C}_f je pak dosazen do polynomu \mathcal{F} a obraz \mathbf{I}_D je transformován na obraz \mathbf{I}_U – viz Alg. (6.2).1, r. 4-21. Protože je využita metoda ‘direkt-mappingu’ (Cucchiara et al. 2003), jsou některé pixely obrazu \mathbf{C}_f neobarvené – viz Obr. (6.2).3B. V druhé části algoritmu DMAM je opět využit optimalizátor jDE, který pro jednotlivé chybějící pixely obrazu \mathbf{I}_U hledá korespondující pixely v obraze \mathbf{I}_D s využitím vektoru \mathbf{C}_f – viz Alg. (6.2).1, r. 22-45. Výhodou navrženého algoritmu je to, že korekční polynom \mathcal{F} může být libovolného stupně a s libovolnou kombinací stupňů - mocnin. Zde je využit max. stupeň 7. Na Obr. (6.2).2. jsou výsledky práce algoritmu DMAM opět s využitím všech 7 stupňů polynomu \mathcal{F} .

Algoritmus (6.2).1 Navržený algoritmus DMAM

Vstup: I_D, M_R, T_2, T_1 , jDE parametry, maska A_S Výstup: C_f, I_U

```
1 Získej  $I_D'$  z kamery
2  $I_D'$  na šedotónový + prahování;  $I_D' \rightarrow I_D$ 
3 Hledej  $M_C$  v  $I_D$ 
4 Inicializuj primární jDE; Vytvoř  $P_{op}^{jDE}$ 
5 Ohodnot  $P_{op}^{jDE}$ ; setříd  $P_{op}^{jDE}$ ; najdi  $X_{best}^{jDE}$ 
6 while (  $T_1$  není splněna )
7   for  $i=0$ ;  $i < N^{jDE}$ ;  $i=i+1$ 
8      $X_t^{jDE}.F_{old} = X_t^{jDE}.F$ ;  $X_t^{jDE}.P_{cr-old} = X_S^{jDE}.P_{cr}$ 
9     if  $rand(0,1) < \tau_1$ :  $X_t^{jDE}.F = F_l + F_u * rand(0,1)$ 
10    if  $rand(0,1) < \tau_2$ :  $X_t^{jDE}.P_{cr} = rand(0,1)$ 
11    Zvol  $\{X_{r1}, X_{r2}, \dots, X_{rk}\}, r_j \neq i, X_{rj} \in P_{op}^{jDE}$ 
12    Vypočti  $X_t$  s využitím  $\bar{V}_p, X_t^{jDE}, X_{rj}, X_{best}^{jDE}$ 
13     $R_1 = fitness_1(X_t^{jDE}.x_1, X_t^{jDE}.x_2)$ 
14     $R_2 = fitness_1(X_t.x_1, X_t.x_2)$  - trial vektor
15    if  $R_1 < R_2$ :  $X_t^{jDE}.F = F_{old}$ ;  $X_t^{jDE}.P_{cr} = P_{cr-old}$ 
16    if  $R_1 \geq R_2$ :  $\forall j, j \in \langle 0, D_{im} \rangle$ :  $X_t^{jDE}.x_j = X_t.x_j$ 
17  endfor
18  Setříd  $P_{op}^{jDE}$ ; najdi  $X_{best}^{jDE}$  v  $P_{op}^{jDE}$ 
19 endwhile
20 Smaž  $P_{op}^{jDE}$ ;  $C_f = X_{best}^{jDE}$  - finální koeficienty
21 Použij  $C_f$  a vypočítej  $I_U$ ; uzavři primární jDE
22 for  $u=0$ ;  $u < I_U\_width$ ;  $u=u+1$ 
23   for  $v=0$ ;  $v < I_U\_height$ ;  $v=v+1$ 
24     if  $I_U[u,v]$  je již obarvený: continue;
25      $P_W.x = u$ ;  $P_W.y = v$  - cílová pozice pixelu
26     Inicializuj sekundární jDE; Vytvoř  $P_{op}^{jDE}$ 
27     Vyčísli  $P_{op}^{jDE}$ ; setříd  $P_{op}^{jDE}$ ; najdi  $X_{best}^{jDE}$ 
28     while (  $T_2$  není splněna )
29       for  $i=0$ ;  $i < N^{jDE}$ ;  $i=i+1$ 
30          $X_t^{jDE}.F_{old} = X_t^{jDE}.F$ ;  $X_t^{jDE}.P_{cr-old} = X_S^{jDE}.P_{cr}$ 
31         if  $rand(0,1) < \tau_1$  then  $X_t^{jDE}.F = F_l + F_u * rand(0,1)$ 
32         if  $rand(0,1) < \tau_2$  then  $X_t^{jDE}.P_{cr} = rand(0,1)$ 
33         Zvol  $\{X_{r1}, X_{r2}, \dots, X_{rk}\}, r_j \neq i, X_{rj} \in P_{op}^{jDE}$ 
34         Vypočítej  $X_t$  s využitím  $\bar{V}_p, X_t^{jDE}, X_{rj}, X_{best}^{jDE}$ 
35          $R_1 = fitness_2(X_t^{jDE}.x_1, X_t^{jDE}.x_2)$ 
36          $R_2 = fitness_2(X_t.x_1, X_t.x_2)$  - trial vektor
37         if  $R_1 < R_2$ :  $X_t^{jDE}.F = F_{old}$ ;  $X_t^{jDE}.P_{cr} = P_{cr-old}$ 
38         if  $R_1 \geq R_2$ :  $\forall j, j \in \langle 0, D_{im} \rangle$ :  $X_t^{jDE}.x_j = X_t.x_j$ 
39       endfor
40       Setříd  $P_{op}^{jDE}$ ; najdi  $X_{best}^{jDE}$  v  $P_{op}^{jDE}$ 
41     endwhile
42      $I_U[u,v] = I_D[X_{best}^{jDE}.x_1, X_{best}^{jDE}.x_2]$ 
43     Smaž  $P_{op}^{jDE}$ 
44   endfor
45 endfor
```

Algoritmus (6.2).2 Výpočet $fitness_1$

 Vstup: $M_C, M_R, C_f \equiv X_i^{jDE} \cdot C_f$ nebo $X_t \cdot C_f$

```

1  $fitness_1 = 0; d = \max(I_w, I_h)$ 
2 for  $i=0; i < M_{CX}; i=i+1$ 
3   for  $j=0; j < M_{CY}; j=j+1$ 
4      $center_X = 0.5 \times (I_w - 1); center_Y = 0.5 \times (I_h - 1);$ 
5      $\Delta_X = \frac{(M_C[i,j].x - center_X)}{d}; \Delta_Y = \frac{(M_C[i,j].y - center_Y)}{d}$ 
6      $R_d = \sqrt{\Delta_X^2 + \Delta_Y^2}; R_s = \sum_{i=1}^{i=k} [c_i \times (R_d)^i]; f = \frac{R_d}{R_s}$ 
7      $X_S = center_X + (f \times \Delta_X \times d); Y_S = center_Y + (f \times \Delta_Y \times d)$ 
8      $fitness_1 = fitness_1 + \sqrt{(X_S - M_R[i,j].x)^2 + (Y_S - M_R[i,j].y)^2}$ 
9   endfor
10 endfor

```

Algoritmus (6.2).3 Výpočet $fitness_2$

 Input: $I_w, I_h, C_f, P_W, t_R \equiv X_i^{jDE}$ or X_t

```

1  $center_X = 0.5 \times (I_w - 1); center_Y = 0.5 \times (I_h - 1);$ 
2  $d = \max(I_w, I_h); \Delta_X = \frac{(t_R.x - center_X)}{d}; \Delta_Y = \frac{(t_R.y - center_Y)}{d}$ 
3  $R_d = \sqrt{\Delta_X^2 + \Delta_Y^2}; R_s = \sum_{i=1}^{i=k} [c_i \times (R_d)^i]; f = \frac{R_d}{R_s}$ 
4  $X_S = center_X + (f \times \Delta_X \times d); Y_S = center_Y + (f \times \Delta_Y \times d)$ 
5  $fitness_2 = \sqrt{(X_S - P_W.x)^2 + (Y_S - P_W.y)^2}$ 

```

Algoritmus (6.2).4 Originální algoritmus jDE

```

1 Zvol  $N, F_l, F_u, \tau_1, \tau_2, \bar{V}_p, T$ ; Dáno  $D_{im}, D_{om}$ 
2 Vytvoř  $P_{op}^{jDE} = (X_1, \dots, X_N), X_i^{jDE} = [x_j | j \in \langle 0, D_{im} \rangle, F, P_{cr}]$ 
3 for  $i = 0; i < N; i=i+1$ 
4   for  $j = 0; j < D_{im}; j=j+1$ 
5     Inicializuj  $X_i^{jDE}.x_j$  náhodně v  $D_{om}$ 
6      $X_i^{jDE}.F = rand; X_i^{jDE}.P_{cr} = rand$ 
7   endfor
8 endfor
9 Ohodnot  $P_{op}^{jDE}$ ; najdi nejlepšího  $X_{best}^{jDE}$ 
10 while (  $T$  není splněna )
11   for  $i = 0; i < N; i=i+1$ 
12      $F_{old} = X_i^{jDE}.F; P_{cr-old} = X_i^{jDE}.P_{cr}$ 
13     Vypočti nový  $X_i^{jDE}.F, X_i^{jDE}.P_{cr}$ 
14     Svol náhodně  $\{X_{r1}, X_{r2}, \dots, X_{rk}\}, r_j \neq i, X_{rj} \in P_{op}$ 
15     Vypočti trial vektor  $X_t$  s využitím  $\bar{V}_p, X_i, X_{rj}, X_{best}$ 
16     if  $fitness(X_i^{jDE})$  je lepší než  $fitness(X_t)$ 
17        $X_i^{jDE}.F = F_{old}; X_i^{jDE}.P_{cr} = P_{cr-old}$ 
18     Else
19        $X_i^{jDE}.x_j = X_t.x_j | j \in \langle 0, D_{im} \rangle;$ 
20     endif
21   endfor
22   Najdi nejlepšího jedince  $X_{best}^{jDE}$  in  $P_{op}^{jDE}$ 
23 Endwhile
24  $X_{best}^{jDE}$  reprezentuje nejlepší odhadnuté řešení

```

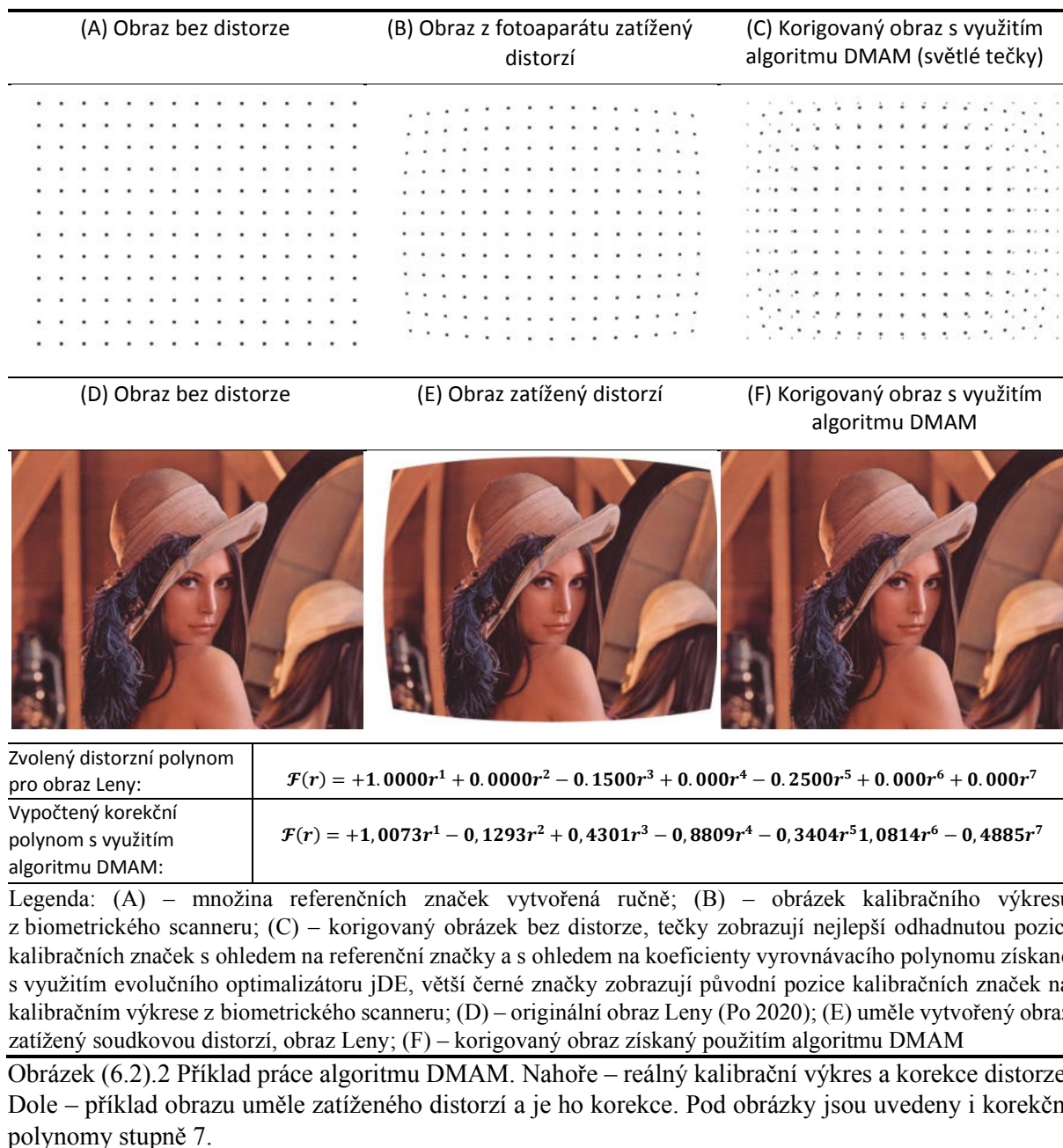
Legenda pro algoritmy (6.2.).1, (6.2.).2, (6.2.).3, (6.2.).4: P_{op}^{jDE} – populace individuí; N – počet individuí v populaci pro primární optimalizátor jDE, individua $X_i^{jDE}(x_1, \dots, x_7) \in \mathbb{R}$, $D_{im} = 7$, x_i je v rozsahu $(-2.0, +2.0)$ a pro sekundární optimalizátor jDE je to $X_i^{jDE}(x_1, x_2) \in I_D$, $D_{im} = 2$; $F, P_{cr}, \tau_1, \tau_2, F_l, F_u$ – jDE pracovní parametry; \bar{V}_p – odchylkový vektor; D_{im} – počet dimenzí každého individua; D_{om} – domain (rozsah použitelných hodnot) pro každou jednotlivou dimenzi D_{im} ; T, T_1, T_2 – ukončovací podmínky; X_i^{jDE} -chromozom každého individua neboli jedno možné řešení dané úlohu; platí, že $X_{r1} \neq X_{r2} \neq \dots \neq X_{rk} \neq X_i^{jDE}$, počet vektorů X_{rk} je dán zvoleným odchylkovým vektorem \bar{V}_p ; X_t – tzv. trial vektor (zkušební vektor) neboli individuum získané aplikací \bar{V}_p ; X_{best}^{jDE} – nejlepší nalezené individuum v každé generaci; I_D'' – originální RGB obraz z digitálního fotoaparátu; I_D – předzpracovaný černobílý obraz zatížený distorzí (B&W obraz); I_U – obraz bez distorze neboli korigovaný; $P_W(x, y)$ – pomocná proměnná; M_C – matice pozic kalibračních značek, která je získána z obrazu I_D ; M_R – matice pozic referenčních značek získaná na základě ručního měření; M_{CX}, M_{CY} – velikosti matic M_C a M_R , velikost M_C a M_R je identická; I_w – šířka obrázku; I_h – výška obrázku; C_f – vektor koeficientů vyrovnávacího polynomu; I_w, I_h – je obvykle počítáno od 0; velikost I_U a I_D je samozřejmě identická; T_1 a T_2 jsou definovány jako pevný počet generací (iterací) použitého evolučního algoritmu, obvykle $T_1 = 120, T_2 = 60$. Tak zvaná ‘tečková notace’ je v algoritmu použita vždy, když je odkazováno na nějakou konkrétní položku záznamu např. X_i^{jDE} . F protože vektory použité v algoritmu jsou uvažovány jako množina záznamů, kde každý záznam se skládá z více položek: $X_i^{primary-jDE} = \{x, fit, F, P_{cr}, F_{old}, P_{cr-old}\}$ kde $x = (x_1, \dots, x_7)$ a podobně pak také pro $X_i^{secondary-jDE}$ a $x = (x_1, x_2)$.

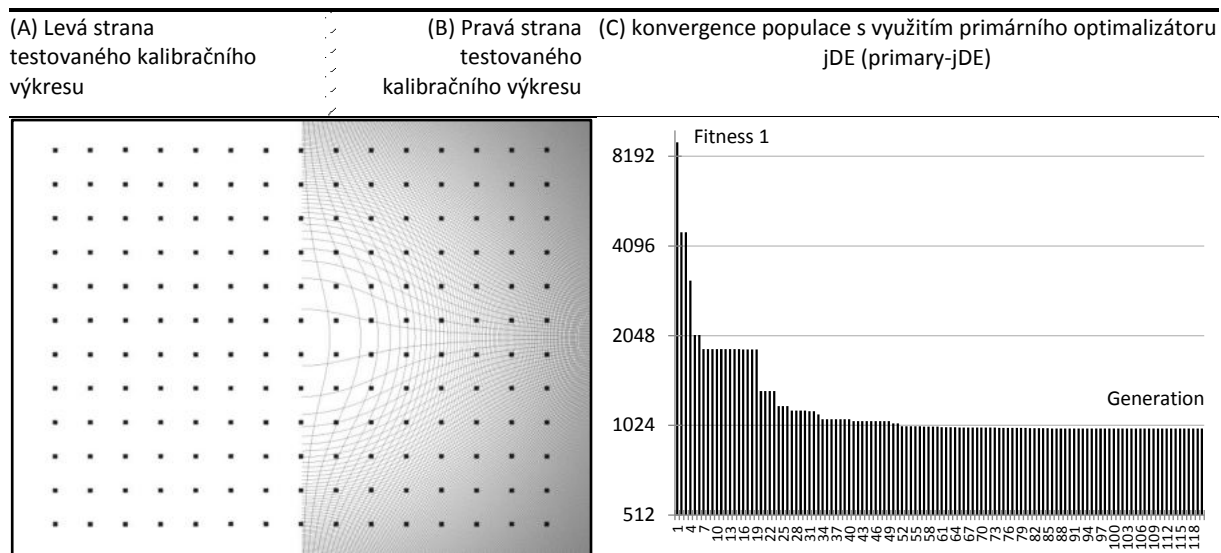
Aby bylo možné rozhodnout, kdy jsou hodnoty vektoru C_f u primary-jDE optimální, byla definována účelová funkce $fitness_1$ – viz Alg. (6.2).2. Podobně byla navržena účelová funkce $fitness_2$ pro druhou část algoritmu DMAM označenou jako ‘sekundární-jDE’ (v algoritmech je použit výraz ‘secondary-jDE’), která definuje míru korespondence mezi pixelem v obraze I_U , který je nutné obarvit a odpovídajícím pixelem v obraze I_D za předpokladu známého vektoru C_f . Obě účelové funkce jsou definovány jako:

$$fitness_1: (\hat{x}, \hat{y}) = \arg \underset{C_f \in \mathbb{R}}{\text{opt}} \mathcal{F}_{primary-jDE}(M_C, M_R) \quad (6.2.4)$$

$$fitness_2: (x', y') = \arg \underset{t_R \in \mathcal{H}_D}{\text{opt}} \mathcal{F}_{secondary-jDE}(t_R, P_W, C_f), P_W \in \mathcal{H}_U$$

(\hat{x}, \hat{y}) pak definuje nejlepší odhad pozice těžiště značky M_F po aplikaci korekčního polynomu na korespondující pozici definovanou značkou M_F s ohledem na pozici referenční značky M_R . (x', y') pak definuje nejlepší odhad pozice pixelu v I_D s ohledem na C_f , jehož barva bude přenesena na pozici $P_W(x, y)$ v undistorted image. Platí tedy, že $X_i^{jDE} \in \mathcal{H}_D$.





Obrázek (6.2.3) Detail z práce algoritmu DMAM. Vlevo – příklad korekce distorze, výsledný korigovaný obraz s použitím metody DMAM v levé části korigovaného kalibračního výkresu a v pravé části bez použití kaskádového estimátoru přímou aplikací algoritmu ‘direkt-mapping’. Vpravo – konvergence populace při odstraňování distorze, kaskádový estimátor DMAM, primární jDE.

6.2.3 Experimentální výsledky

Výsledek práce korekčního algoritmu DMAM je zobrazen na (6.2).2A,B,C. Obrázek (A) zobrazuje pozice referenčních značek M_R , tedy ideální obraz, kterého je potřeba dosáhnout. Obrázek (B) zachycuje B&W snímek získaný z biometrického scanneru, který je zatížený distorzi malého rozsahu. Obrázek (C) zachycuje finální korigovaný obraz – černé puntíky a jsou také zobrazeny kalibrační znaky v obraze, který je zatížený distorzi z obrazu (B). Na Obr. (6.2).3C je zobrazen graf konvergence $fitness_1$ funkce. Výsledné hodnoty v generacích odpovídají součtu vzdálenosti značek M_F a M_R podle aktuálních koeficientů C_f v pixelech. Nejlepší dosažená hodnota byla 999.0 pixelů. S ohledem na ruční ustavení fotoaparátu je výsledek korekce velmi dobrý. Na Obr. (6.2).3 je pak zobrazen korigovaný obraz kalibračních značek, jak vypadá s využitím algoritmu DMAM a s využitím metody ‘direkt-mapping’. Pracovní parametry primárního jDE byla následující: $N = 60$, $T_1 = 120$, $\tau_1 = 0.7$, $\tau_2 = 0.3$, $F_u = 0.5$, $F_l = 0.5$, $\overline{V_p} \equiv RandToBest/1/bin = X1 + F(Xb - X2) + F(X3 - X4)$. Vektor $\overline{V_p}$ byl zvolen čistě na základě praktických experimentů a podával nejlepší výsledky. Pracovní parametry sekundárního jDE byla následující: $N = 20$, $T_2 = 40$, $\tau_1 = 0.7$, $\tau_2 = 0.3$, $F_u = 0.5$, $F_l = 0.5$, Vektor $\overline{V_p}$ byl stejný jako u primárního jDE. Přesnost a především rychlost algoritmu DMAM je možné volit parametry N , T_1 , T_2 . Užitý procesor pro výpočty byl Athlon3500+/Orleans 2.2GHz. Čas potřebný pro výpočet jedné korekce je 58 minut.

Na Obr. (6.2).2D,E,F je výsledek druhého experimentu s běžným bitmapovým obrazem Leny (Po 2020). Ten byl nejprve zkreslen s využitím polynomu \mathcal{F} - viz Obr. (6.2).2. Algoritmem DMAM byl poté získán vektor C_f korekčních koeficientů a byla korekce distorted image a rekonstrukce chybějících pixelů. Výsledek je na Obr. (6.2).2E a je zde uveden polynom použitý pro distorzi obrazu a polynom získaný algoritmem DMAM. Pracovní parametry byly stejné jako v předchozím experimentu. Výsledný obraz – viz Obr. (6.2).2F je zcela perfektní a bez chybějících neobarvených pixelů jako je tomu na Obr. (6.2).3B.

6.2.4 Závěr

Navržená metoda umožňuje provádět odstranění centrické radiální distorze malého až středního rozsahu, a to jak soudkové, tak i polštářové. Metoda je sestavena jako tzv. kaskádový evoluční estimátor a využívá pokročilého evolučního algoritmu označovaného jako jDE, který byl publikován v roce 2006. Korekční metoda je čistě polynomiální a umožňuje volit různou kombinaci stupňů korekčního polynomu např. jen sudé nebo jen liché nebo kombinace obou. Je možné použít i nižší stupeň korekčního polynomu. Maximální stupeň je 7. Tato hodnota vychází čistě z dostupného technického vybavení, a především časových nároků na výpočet. Metoda byla navržena pro korekci obrazů z biometrického scanneru, ale má obecný charakter a lze ji proto využít pro korekci obrazu, který je zatížený soudkovou nebo polštářovou distorzí z různých optických zařízení. Navržená metoda poskytuje relativně rychlý, a především přesný nástroj. Na rozdíl od původního algoritmu ‘direkt-mapping’ nabízí kaskádový estimátor možnost získat výsledný korigovaný obraz, který není postižený strukturálním rozpadem, který se projevuje ztrátou pixelů v celé ploše korigovaného obrazu – viz Obr. (6.2).3 vpravo, pravá polovina výkresu. Nejvíce je tento jev patrný v rozích korigovaného výkresu. Použitý evoluční algoritmus není zcela bezparametrický. Některé pracovní parametry se musí nastavit ručně a tyto parametry musí být nastaveny vhodným způsobem, jinak se může stát, že kaskádový estimátor selže jako celek. Správné hodnoty těchto parametrů je však možné získat poměrně rychlým odhadem – výpočtem v určitých mezích a porovnáním výsledné fitness u primárního algoritmu, který je zařazený v kaskádě.

LITRERATURA

- Conrady A.E. 1919. Decentred lens-systems. *Monthly Notices of the Royal Astronomical Society*. vol. 79, pp 384-390.
- Magill A.A. 1955. Variation in distortion with magnification. *Journal of Research of the National Bureau of Standards*. vol. 54, no. 3, pp. 135–142.
- Brown D.C. 1956. The simultaneous determination of the orientation and lens distortion of a photogrammetric camera, Air Force Missile Test Center – Technical Report, Patrick AFB, Florida. vol. 50/20, p. 23.
- Brown C.D. 1971 Close range camera calibration. In: *Symposium on close range photogrammetry*. DBA Systems Inc., Melbourne.
- Brest J., Greiner S., Boškovič B., Mernik M., Žumer V. 2006. Selfadapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*. vol. 10, no. 6, pp. 646-657.
- Liu Ch., Jia Y., Cai W., Wang T., Song Y., Sun X., Zhang J. 2014. Camera calibration optimization technique based on genetic algorithms. *Journal of Chemical and Pharmaceutical Research*. vol. 6, no. 1, pp. 97-103.
- Kim W., Kim Ch. 2009. An efficient correction method of wide-angle lens distortion for surveillance systems. In: *IEEE International Symposium on Circuits and Systems*. pp. 3206-3209.
- Cucchiara R., Grana C., Prati A., Vezzani R. 2003. A Hough Transform-based method for Radial Lens Distortion Correction. In: *12th International Conference on Image Analysis and Processing*. pp. 182-187.
- Po L.M. 2020. [online]. Li-Man Po ©2014-2019 [cit. 20.5.2020]. Dostupné z: <http://www.ee.cityu.edu.hk/~lmpo/lenna/Lenna97.html>
- Storn R., Price K. 1997. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*. vol. 11, pp. 341-359.
- Welford W.T. 1986. *Aberrations of Optical Systems*. CRC Press. p. 284. ISBN-13: 9781315136530.

6.3 Kalibrace obrazu založená na algoritmu division-model

Jak soudkové, tak i polštářové zkreslení obrazu patří do skupiny distorzí obrazu, které jsou nechtěné a obvykle jsou korigovány nějakým vhodným algoritmem. Pro účely korekce obrazu postiženého distorzí bylo navrženo mnoho různých algoritmů. Některé mají universální charakter a jiné jsou poplatné konkrétnímu typu použité optické soustavy. V této kapitole je provedeno porovnání několika druhů metod určených pro odstranění distorze obrazu, které jsou založené na tzv. ‘Polynomial Division Model’ (PDM). Český překlad zatím není obecně vžitý, ale bude to něco ve smyslu: ‘Polynomiální model s lomeným výrazem’. Celosvětově se však používá originální název. Čistě na základě rozsáhlých experimentálních výsledků byla také navržena nová metoda – korekce stávajícího vzorce PDM, která poskytuje výrazně lepší výsledky při korekci obrazů, které jsou zatíženy soudkovou nebo polštářovou distorzí. Výpočet korekce distorze pomocí klasického polynomiálního modelu je poměrně zdoluhavý. Výpočet korekce podle PDM je rychlý díky tomu, že nevyžaduje velké množství členů polynomu a to, ani pokud je obraz zatížen velkou distorzí např. u objektivů typu rybí oko (Fish-Eye). Nově navržený ‘Division Model’ (DM) zachovává všechny dobré vlastnosti původních algoritmů založených na DM a rozšiřuje je. V experimentální sekci jsou pak presentovány výsledky výpočtů, na jejichž základě byla provedena korekce DM.

6.3.1 Úvod

Původní práce, které popisovaly způsob, jak odstranit vadu distorze obrazu, pocházely převážně z počátku nebo z poloviny minulého století viz (Conrady 1919; Magill 1955; Brown 1956, 1964, 1965, 1966, Brown 1971). Používají se stále. S nástupem počítačů v 80. a především v 90. letech se využívání korekčních metod rozšířilo díky možnostem digitálního zpracování obrazu. Práce jako např. (Brown 1965) sice poskytuje dostatečně obecný a kvalitní nástroj pro odstranění distorze – centrické i necentrické, ale při použití většího množství členů korekčního polynomu je výpočetní náročnost značná. Korekční algoritmus se tak stává nepoužitelným např. při real-time zpracování videa z HD webových kamer. Řešením je buď snížit počet členů polynomu a tím i snížit přesnost korekce, nebo využít nějaký hardwarový akcelerátor – např. programovatelné logické pole (Asari 2004). Další a poměrně elegantní metodou je úprava Polynomiálního Modelu (PM) popsaná v (Brown 1965) tak, aby při zachování výkonnosti a přesnosti poskytoval nový model výrazně rychlejší odezvu. Takové modely existují. Jedním z nich je např. model popsáný v (Kim & Kim 2009). Jedná se o tzv. nepolynomiální model, který zcela opomíjí výsledky práce popsané v (Conrady 1919; Magill 1955; Brown 1956, 1964, 1965, 1966, 1971). Autoři navrhli tuto metodu pro odstranění výrazné vady soudkové distorze z objektivu typu Rybí Oko. Podobně např. (Dhane et al. 2012) navrhli tak zvaný „nepolynomiální nelineární radiální rozšiřovací/roztahovací/expanzní metodu“ (non-polynomial non-linear radial stretching method). (Fitzgibbon 2001) navrhl matematický model, který vychází z práce (Brown 1965), ale upravuje model do tvaru, který autor označil jako Division Model (DM). Je odvozený z klasického PM a jeho výhodou je především rychlost. Je možné jej využít jak pro odstranění vady distorze u Fish-eye objektivů s velkou vadou distorze, tak i u běžných objektivů, kde distorze není tak velká. Řadu dalších alternativních modelů je možné nalézt v (Basu & Licardie 1995). Přehled potom v (Hughes et al. 2008). Obecné informace týkající se identifikace a odstraňování optických vad je možné nalézt v (Welford 1986; Virendra 1991; Fischer & Tadic 2000; Sasián 2012).

V následujících odstavcích bude provedena analýza matematického modelu, který byl navržen v práci (Fitzgibbon 2001). Autor nazval svůj model ‘Division Model’ (DM). Na základě této práce bude dále navržen nový vylepšený DM, který pouze upravuje Fitzgibbonův DM do té míry, že je využitelný jak pro objektivu typu rybí oko, tak i v případě malé distorze obrazu, a to s výrazně lepšími výsledky.

6.3.2 Související publikace

(Fitzgibbon 2001) a také (Burchardt & Voss 2001) navrhli modifikaci klasického PM. Vznikl tzv. Division Model viz např. (Fitzgibbon 2001), jehož model je vyjádřený jako:

$$p = \frac{1}{1 + \lambda \|x\|^2} x \quad (6.3.1)$$

Fitzgibbonův model využívá jeden člen korekčního polynomu. Obecně jej budeme značit jako ‘klasický’ DM. Vznikly také další metody, které využívaly zcela odlišné, přesto však velmi efektivní a rychlé postupy např. (Kim & Kim 2009; Dhane et al. 2012; Devernay & Faugeras 2001). Podobný přístup byl navržen už dříve např. v (Lenz 1987). Výsledky práce popsané v (Fitzgibbon 2001) jsou uplatněny i v publikaci (Flores et al. 2014), kde autoři navrhli metodu označovanou jako metodu bez zpětné vazby, která využívá jednoparametrický DM na korekci radiální centrické distorze zapříčiněné použitím čoček s velkým zorným polem a jen jedním zaznamenaným obrazem. (Cedrés et al. 2015) upravili DM převzatý z (Fitzgibbon 2001) tak, že jsou využity dva členy polynomu:

$$L(r) = \frac{1}{1 + k_1 r^2 + k_2 r^4} \quad (6.3.2)$$

(Brito et al. 2013) podobně jako (Fitzgibbon 2001) použil DM s jedním členem. Svoji volbu zdůvodnili tak, že jeden člen je dostatečný pro odstranění distorze. Vycházeli přitom z výsledků poskytnutých v článcích (Devernay & Faugeras 2001; Wang et al. 2009; Wang et al. 2008). V praxi je pro korekci distorze např. u kompaktních fotoaparátů s ‘běžným’ objektivem naprosto dostačující korekční polynom s jedním členem (Das & Patil 2006). (Bukhari & Dailey 2013) navrhli novou metodu pro automatickou korekci radiální centrické distorze využívající čar z olovnice. Metoda je založená na původním Fitzgibbonově DM modelu (Fitzgibbon 2001) a podobně jako mnoho jiných výše popsaných metod používá DM model jen s jedním parametrem.

6.3.3 Použitý evoluční optimalizátor

Výpočet hodnot koeficientů korekčního polynomu je výpočetně značně náročný. V případě, že korekční polynom je stupně 3 je možné využít např. Cardanových rovnic viz např. (Cucchiara et al. 2003). Rovnice až do stupně 4 lze obecně řešit analyticky. V případě, že stupeň je 5 a vyšší je nutné použít vhodné metody numerické matematiky. Při návrhu korekčního modelu byla zvolena numerická metoda označovaná jako Stochastický Metaheuristický Optimalizátor (SMO). Její původ spadá až do počátků 80. let minulého století, kdy byl navržen Jednoduchý Genetický Algoritmus (Simple Genetic Algorithm) (SGA) (Goldberg 1987). Byl to počátek moderních Evolučních Algoritmů (EA). Evoluční algoritmy (Rechenberg 1973; Holland 1975; Rechenberg 1994; Schwefel 1995) tvoří podstatnou část navrženého korekčního polynomiálního modelu. V roce 1997 byl v článku (Storn & Price 1997; Price et al. 1999) presentován algoritmus označený jako Diferenciální Evoluce (DE). Metoda pro korekci obrazu, která je předkládána v této kapitole používá modifikaci původního algoritmu DE, která nese název jDE a byla publikována o dekádu později v článku (Brest et al. 2006,2010). jDE byl pro řešení dané úlohy zvolen čistě na základě rozsáhlých praktických experimentů (Moravec & Pošík 2014a,b; Moravec 2015).

jDE je využit jako optimalizátor až sedmi-dimezionální objektive function, která je tzv. omezená (constrained), spojitá, neseperabilní, nelineární a dobře podmíněná. jDE používá k řešení úlohy množinu tzv. jedinců resp. kandidátských řešení, které jsou v prvním kroku rozmístěny zcela náhodně v celém prostoru možných řešení. V dalších krocích již pohyb těchto jedinců k optimálnímu řešení řídí algoritmus jDE. Problematika metaheuristických optimalizátorů je velmi rozsáhlá a bližší informace je možné nalézt v (Eiben et al. 1999; Gampeler et al. 2002; Liu & Lampinen 2002; Smit & Eiben 2009;

Smit & Eiben 2010a,b; Das & Suganthan 2010) a také (Floudas & Pardalos 2009). Navržená úprava DM, který presentoval (Fitzgibbon 2001) je čistě záležitostí modifikace vzorce DM navrženého Fitzgibbonem. Algoritmus jDE je využit pouze jako vhodný a velmi efektivní optimalizátor. Např. (Asari et al. 1999) také navrhli vlastní optimalizační metodu s ohledem na tehdy běžně dostupné prostředky.

6.3.4 Navržená korekční metoda

Navrhovaná úprava DM vychází z publikace (Fitzgibbon 2001) a pozdějších výsledků presentovaných v (Flores et al. 2014; Cedrés et al. 2015; Brito et al. 2013; Devernay & Faugeras 2001; Wang et al. 2009; Wang et al. 2008; Bukhari & Dailey 2013). Úprava DM byla provedena čistě na základě rozsáhlých experimentálních měření a jako optimalizátor byl zvolen výše popsany algoritmus jDE.

(Fitzgibbon 2001) navrhl tzv. ‘Division Model‘ (DM), který byl pro jeden člen polynomu formulován jako:

$$\mathcal{F}(r_d) = \frac{r_u}{1 + k_1 r_u^2} \quad (6.3.4)$$

Stejný model je využit i v publikaci (Flores et al. 2014) a v mnoha dalších. Z hlediska stochastické optimalizace s využitím stochastického optimalizátoru jDE je možné použít drobnou modifikaci a zapsat, že:

$$\mathcal{F}(r_d) = \frac{r_u}{1 + k_1 r_u^2} \approx \frac{r_u}{1 - k_1 r_u^2} \quad (6.3.5)$$

Pro praktický výpočet tedy lze použít i výraz za znaménkem \approx , protože oba výrazy dávají identické výsledky při korekci obrazu – soudkové i polštářové – viz sekce experimentálních výsledků. (Cedrés et al. 2015) převzal výsledky publikované v (Fitzgibbon 2001) a použil DM se dvěma parametry:

$$\mathcal{F}(r_d) = \frac{1}{1 + k_1 r_u^2 + k_2 r_u^4} \quad (6.3.6)$$

Podobně jako u jedno-parametrového uspořádání je i zde možné zapsat, že:

$$\mathcal{F}(r_d) = \frac{1}{1 + k_1 r_u^2 + k_2 r_u^4} \approx \frac{1}{1 - (k_1 r_u^2 + k_2 r_u^4)} \quad (6.3.7)$$

Výsledky přesnosti při korekci obrazu jsou opět totožné. Z výše uvedeného je možné úvahou rozšířit DM navržený Fitzgibbonem a definovat nový DM, který má však výrazně lepší výsledky. Tento nový DM bude dále označován jako **λDM** a je definován následujícím způsobem:

$$\mathcal{F}_{NEW}(r_d) = \frac{k_1 r_u}{\lambda_1 + \lambda_2 \sum_{i=2}^n k_i r_u^i} \quad (6.3.8)$$

kde $\lambda_1, \lambda_2 \in \mathbb{R}$ jsou pro výpočet uvažovány jako konstanty a koeficienty k_1 a $k_i, i \in (2, n)$ je nutné určit výpočtem. Při experimentech se osvědčilo nastavení $\lambda_1 \in \{0, +1.0, -1.0\}$, $\lambda_2 \in \{+1.0, -1.0\}$. Ideálně pak $\lambda_1, \lambda_2 = 1$. Model **λDM** byl odvozen čistě na základě experimentů při použití stochastického

evolučního optimalizátoru. Na rozdíl od klasického DM definovaného Fitzgibbonem se jedná o model, který využívá vždy alespoň dva parametry k_1 a k_2 . Jedná se tedy o čistě racionální lomenou funkci.

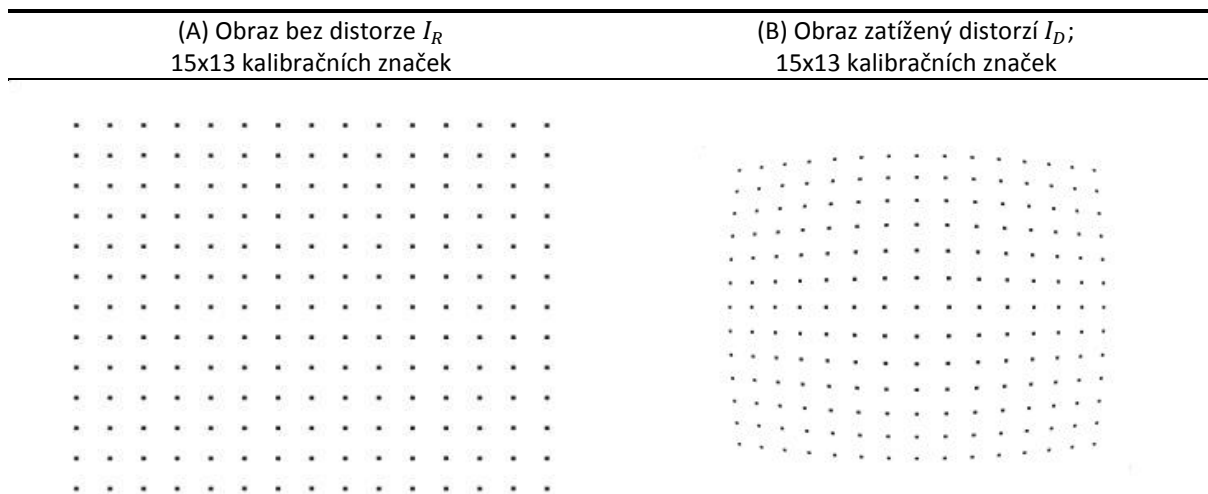
Pro vyčíslení optimálních hodnot koeficientů k_i u polynomu (6.3.8) byl zvolen evoluční estimátor – metaheuristika jDE. jDE minimalizuje účelovou funkci *fitness*, definovanou jako:

$$(\hat{x}, \hat{y}) = \arg \operatorname{opt}_{k_i \in \mathbb{R}} \mathcal{F}_{\text{primary_jDE}}(\mathbf{M}_C, \mathbf{M}_R) \quad (6.3.9)$$

kde \mathbf{M}_C je matice definující pozice kalibračních značek – viz Obr. (6.3).1B, k_i je množina koeficientů polynomu $\mathcal{F}_C(r)$. Pozice libovolné značky je definovaná jako (\hat{x}, \hat{y}) . Hodnoty k_i odpovídají chromozomu jedince \mathbf{X}_{best} v populaci \mathbf{P}_{op} . \mathbf{M}_R je matice referenčních pozic značek u obrazu nezatíženém distorzí – viz Obr. (6.3).1A. Vztah (6.3.9) definuje pouze skutečnost, že jsou hledány optimální koeficienty k_i takové, které minimalizují hodnotu ε ve vzorci (6.3.10). Výsledkem je korigovaný obraz bez distorze (polštářové nebo soudkové) s množinou značek \mathbf{M}_F , které by v ideálním případě měly být totožné s množinou značek \mathbf{M}_R resp. by mělo platit, že $\mathbf{M}_F \equiv \mathbf{M}_R$. Pozice značek \mathbf{M}_F jsou získány s využitím korekčního polynomu $\mathcal{F}_C(r)$ s určitou množinou koeficientů k_i . Kompletní algoritmus použitý pro korekci distorze je detailně popsán v (Moravec & Hub 2015). Článek je volně k dispozici včetně algoritmu. Je poměrně rozsáhlý, a proto zde nebude znovu detailně popisován. Nejdůležitější část navrženého korekčního modelu λDM je především definovaná vzorcem (6.3.8).

6.3.5 Nastavení pracovních parametrů

Experimentální ověření navrženého λDM modelu probíhalo ve dvou fázích. Nejprve byly uměle vytvořeny obrazy zatížené polštářovou nebo soudkovou distorzí s využitím polynomu, kde vždy jen jeden koeficient u mocniny větší, než jedna měl hodnotu jinou než nula. Úkolem použitých DM pak bylo tuto distorzi odstranit. Ve druhé fázi byly vytvořeny obrazy zatížené distorzí za použití polynomu, kde žádný z koeficientů u mocnin 1 až 7 nebyl nulový. Úkolem testovaných DM bylo opět co nejlépe odstranit distorzi. Pro odstranění distorze byly použity jednotlivé testované DM s polynomy, které budou označovány jako $\mathcal{F}_C(r)$ s proměnným počtem členů polynomu až do stupně 7. Byly testovány všechny možné kombinace stupňů; celkem 127 – viz (6.3.8). Porovnáním výsledků u jednotlivých DM pak bylo možné získat informace o tom, který z DM a za jakých podmínek poskytuje lepší výsledky. Popíšme tedy způsob vytvoření obrazů s vadou distorze a klasifikaci výsledků podrobněji.



Použitý distorzní polynom pro obrázek vpravo (B):

$$\mathcal{F}(r) = +1.0r^1 - \mathbf{0.5}r^2 + 0.0r^3 + 0.0r^4 + 0.0r^5 + 0.0r^6 + 0.0r^7$$

Obrázek (6.3).1 Vlevo – obraz kalibračních značek bez distorze a Vpravo – obraz zatížený distorzí a získaný jedním ze 136 distorzních polynomů, které byly vytvořeny pro experimentální účely. Obraz bez distorze obsahuje sadu značek \mathbf{M}_R . Obraz zatížený distorzí obsahuje značky označované jako \mathbf{M}_C . Úkolem opravného mechanismu je ztotožnit značky \mathbf{M}_F s ohledem na značky \mathbf{M}_R , pomocí korekčního algoritmu a kalibračních značek \mathbf{M}_C v obraze, který je zatížený distorzí. Značky se při výpočtu uvažují jako jeden bod korespondující s těžištěm dané značky. \mathbf{M}_F jsou značky generované navrženým korekčním algoritmem v průběhu výpočtu.

Aby bylo možné porovnat jednotlivé testované DM, byla vytvořena sada 136 (119+17) testovacích obrazů, které byly uměle zatíženy soudkovou nebo polštářovou distorzí menšího rozsahu – v rámci možností použitého technického vybavení; resp. byl použit biometrický scanner - viz (Moravec & Hub 2015). Základem testovací série se stal obraz v rozlišení 2592x1944x24rgb – viz Obr. (6.3).1A, který obsahuje 15x13 čtvercových kalibračních značek M_C ; každá značka o rozměru 20x20 pixelů. Testovací obrazy – viz např. Obr. (6.3).1B byly vytvořeny s využitím následujících polynomů (S1)..(S6):

$$(S1) \mathcal{F}(r) = +1.0r^1 + \langle -\mathbf{0.6}; +\mathbf{0.2} \rangle r^2 + 0.0r^3 + 0.0r^4 + 0.0r^5 + 0.0r^6 + 0.0r^7$$

$$(S2) \mathcal{F}(r) = +1.0r^1 + 0.0r^2 + \langle -\mathbf{0.8}; +\mathbf{0.2} \rangle r^3 + 0.0r^4 + 0.0r^5 + 0.0r^6 + 0.0r^7$$

$$(S3) \mathcal{F}(r) = +1.0r^1 + 0.0r^2 + 0.0r^3 + \langle -\mathbf{1.2}; +\mathbf{0.3} \rangle r^4 + 0.0r^5 + 0.0r^6 + 0.0r^7$$

$$(S4) \mathcal{F}(r) = +1.0r^1 + 0.0r^2 + 0.0r^3 + 0.0r^4 + \langle -\mathbf{2.0}; +\mathbf{0.3} \rangle r^5 + 0.0r^6 + 0.0r^7$$

$$(S5) \mathcal{F}(r) = +1.0r^1 + 0.0r^2 + 0.0r^3 + 0.0r^4 + 0.0r^5 + \langle -\mathbf{2.5}; +\mathbf{0.4} \rangle r^6 + 0.0r^7$$

$$(S6) \mathcal{F}(r) = +1.0r^1 + 0.0r^2 + 0.0r^3 + 0.0r^4 + 0.0r^5 + 0.0r^6 + \langle -\mathbf{3.0}; +\mathbf{0.4} \rangle r^7$$

Tato sada rovnic bude označována jako série „S“, kde velikost distorze je pro polynomy (S1)..(S6) měněna úpravou příslušného koeficientu k_i ve zvoleném rozmezí $\langle -min; +max \rangle$, které je současně jiné pro každý polynom. Krok je vždy 0.1. U jednotlivých distorzních polynomů (S1)..(S6) je použit vždy jen jeden člen polynomu s mocninou vyšší než 1 a člen první s mocninou 1. Vzniklo tak 119 nových obrazů {S1-8;S2-10;S3-15;S4-23;S5-29;S6-34}.

Cílem testů v **první fázi** bylo využít jednotlivé DM s proměnným počtem a uspořádáním členů korekčního polynomu k odstranění distorze u těchto 119 obrazů ze série S resp. cílem bylo získat zpět co možná nejlepší podobu obrazu na Obr. (6.3).1A. Chyba ε mezi korigovaným obrazem a ideálním obrazem, který je na Obr. (6.3).1A bude udávána v pixelech a definuje přesnost a schopnost použitého DM korigovat vadu distorze na kalibračním výkrese. Chyba ε je daná jako součet vzdáleností těžišť korespondujících kalibračních značek v obraze bez distorze I_R a na korigovaném obraze I_C a je vyjádřena vztahem:

$$\varepsilon = \sum_{i=1}^n \sqrt{(x_r - x_c)^2 + (y_r - y_c)^2} \text{ resp. RMSE} \quad (6.3.10)$$

kde (x_r, r_r) je těžiště značky na obraze bez distorze a (x_c, r_c) je těžiště značky na korigovaném obraze a n je počet kalibračních značek je $15 \times 13 = 195$. RMSE značí „Root Mean Square Error“.

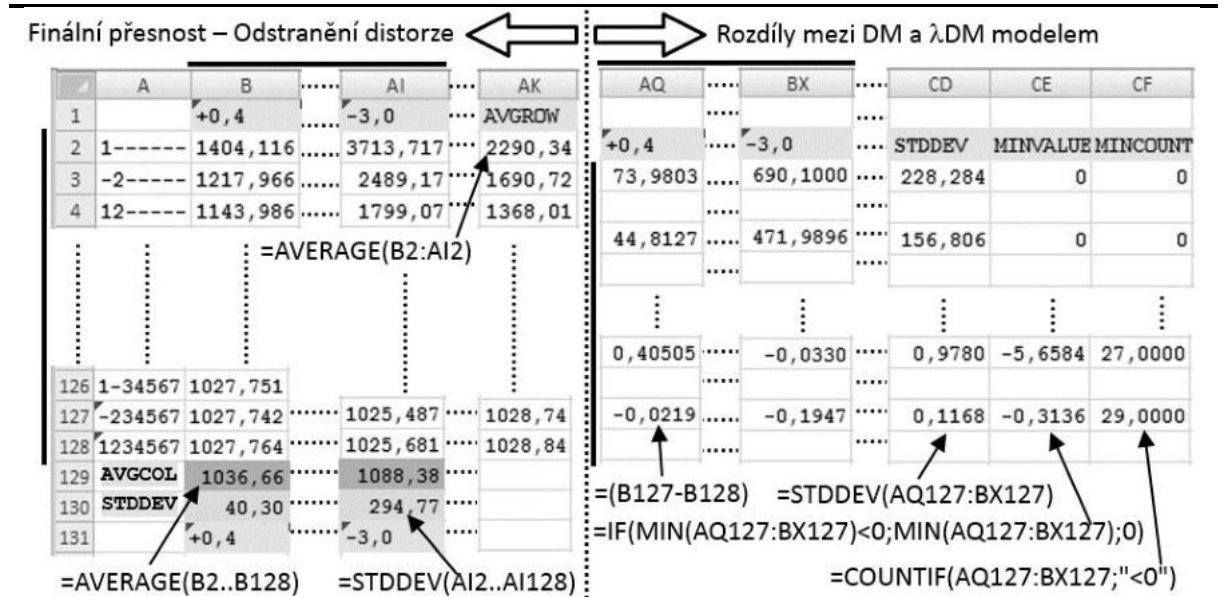
V druhé fázi experimentů bylo cílem ukázat chování testovaným DM v případě, že algoritmus musí korigovat obraz, který vznikl umělou distorzí s využitím polynomu $\mathcal{F}(r)$, kde jsou využity všechny členy tohoto polynomu. Tato varianta je bližší reálnému stavu zatížení obrazu distorzí; soudkovou nebo polštářovou. Podobně jako v první fázi i zde byla využita hodnota ε pro hodnocení kvality korekce. 17 obrazů I_D pro druhou fázi experimentů vzniklo opět distorzí obrazu I_R tentokrát s využitím polynomů označených jako (M1)..(M17) s konstantními koeficienty k_i :

| |
|--|
| (M1) $\mathcal{F}(r) = +1.0r^1 + 0.05r^2 + 0.05r^3 + 0.05r^4 + 0.05r^5 + 0.05r^6 + 0.05r^7$ |
| (M2) $\mathcal{F}(r) = +1.0r^1 + 0.05r^2 + 0.05r^3 + 0.05r^4 + 0.05r^5 + 0.05r^6 + 0.10r^7$ |
| (M3) $\mathcal{F}(r) = +1.0r^1 - 0.05r^2 - 0.05r^3 - 0.05r^4 - 0.05r^5 - 0.05r^6 - 0.05r^7$ |
| (M4) $\mathcal{F}(r) = +1.0r^1 - 0.10r^2 - 0.10r^3 - 0.10r^4 - 0.10r^5 - 0.10r^6 - 0.10r^7$ |
| (M5) $\mathcal{F}(r) = +1.0r^1 - 0.15r^2 - 0.15r^3 - 0.15r^4 - 0.15r^5 - 0.15r^6 - 0.15r^7$ |
| (M6) $\mathcal{F}(r) = +1.0r^1 - 0.15r^2 - 0.20r^3 - 0.20r^4 - 0.20r^5 - 0.20r^6 - 0.20r^7$ |
| (M7) $\mathcal{F}(r) = +1.0r^1 - 0.15r^2 - 0.20r^3 - 0.25r^4 - 0.25r^5 - 0.25r^6 - 0.25r^7$ |
| (M8) $\mathcal{F}(r) = +1.0r^1 - 0.15r^2 - 0.20r^3 - 0.30r^4 - 0.30r^5 - 0.30r^6 - 0.30r^7$ |
| (M9) $\mathcal{F}(r) = +1.0r^1 - 0.15r^2 - 0.20r^3 - 0.30r^4 - 0.35r^5 - 0.35r^6 - 0.35r^7$ |
| (M10) $\mathcal{F}(r) = +1.0r^1 - 0.20r^2 - 0.20r^3 - 0.30r^4 - 0.40r^5 - 0.40r^6 - 0.40r^7$ |
| (M11) $\mathcal{F}(r) = +1.0r^1 - 0.15r^2 - 0.20r^3 - 0.30r^4 - 0.45r^5 - 0.45r^6 - 0.45r^7$ |
| (M12) $\mathcal{F}(r) = +1.0r^1 - 0.15r^2 - 0.20r^3 - 0.30r^4 - 0.50r^5 - 0.50r^6 - 0.50r^7$ |
| (M13) $\mathcal{F}(r) = +1.0r^1 - 0.15r^2 - 0.20r^3 - 0.30r^4 - 0.50r^5 - 0.55r^6 - 0.55r^7$ |
| (M14) $\mathcal{F}(r) = +1.0r^1 - 0.15r^2 - 0.20r^3 - 0.30r^4 - 0.50r^5 - 0.60r^6 - 0.60r^7$ |
| (M15) $\mathcal{F}(r) = +1.0r^1 - 0.15r^2 - 0.20r^3 - 0.30r^4 - 0.50r^5 - 0.60r^6 - 0.65r^7$ |
| (M16) $\mathcal{F}(r) = +1.0r^1 - 0.15r^2 - 0.20r^3 - 0.30r^4 - 0.50r^5 - 0.60r^6 - 0.70r^7$ |
| (M17) $\mathcal{F}(r) = +1.0r^1 - 0.15r^2 - 0.20r^3 - 0.30r^4 - 0.50r^5 - 0.60r^6 - 0.75r^7$ |

Ve všech případech byly pro odstranění distorze využity následující metody: klasický PDM a nově navržený model λDM . Oba modely byly použity s proměnným počtem členů polynomu v rozmezí 1 až 7. DM byl tedy použit ve své klasické podobě s jedním nebo dvěma členy polynomu a také byl rozšířen až na stupeň polynomu 7. Ve všech případech bez prvního členu polynomu. Celkem bylo provedeno 17272 korekcí distorze (obrazy S: 15113 + obrazy M: 2159). Takto velký soubor dat je již dostatečný pro hlubší analýzu chování jednotlivých testovaných DM. Výsledná statistická data z porovnávání jednotlivých DM byla přehledně zaznamenána do 7 tabulek. 6 tabulek pro první fázi, jedna tabulka pro druhou fázi – viz Obr. (6.3).2. V levé části každé tabulky jsou hodnoty ε pro jednotlivé kombinace hodnot koeficientů k_i polynomu $\mathcal{F}(r)$. Sloupec A zobrazuje kombinace stupňů korekčního polynomu. Je-li použita hodnota 1, pak se jedná o λDM jinak o DM. V pravé části tabulky jsou pak již finální rozdíly mezi jednotlivými DM resp. mezi DM a λDM - sloupce AQ až BX. Rozdílová hodnota Δ vznikla prostým odečtením hodnoty ε pro model DM a hodnoty ε pro model λDM a je vyjádřena jako:

$$\Delta = \varepsilon_{DM} - \varepsilon_{\lambda DM} \quad (6.3.11)$$

viz např. buňka AQ127 obsahuje hodnotu rozdílu buněk A127 (kombinace stupňů polynomu $\mathcal{F}(r)$ je 234567) a A128 (kombinace je 1234567). Pokud je hodnota kladná, pak je λDM účinnější než DM. Pokud je hodnota záporná, pak je tomu naopak.



Obrázek (6.3).2 Interpretace výsledných experimentálních dat. Levá část tabulky obsahuje výsledky přesnosti při odstranění distorze, pravá pak přesnost jednotlivých testovaných DM. Zobrazená tabulka je pro jeden polynom: $(S6) \mathcal{F}(r) = +1.0r^1 + 0.0r^2 + 0.0r^3 + 0.0r^4 + 0.0r^5 + 0.0r^6 + (-3.0; +0.4)r^7$, neboli $k_7 \in (-3.0; +0.4)$, krok 0.1, 34 hodnot – rozsah buněk B2 až A1128. Celkem 34 sloupců x 127 řádků = 4318 hodnot ε . Celkem bylo takto získáno 6 tabulek s různým počtem hodnot pro jednoduché distorzní polynomy (S1)..(S6) a jedna tabulka pro distorzní polynomy (M1)..(M17) s konstantními koeficienty k_i .

Pro porovnání výkonnosti je důležitý také počet selhání λDM , resp. hodnoty ve sloupci CF. Další zobrazované hodnoty jsou směrodatná odchylka – sloupec CD a minimální hodnota v řádku – sloupec CE. Protože počet experimentálních dat je značný, budou zobrazeny pouze výsledné 3D grafy získané z těchto tabulek. Separátně pro pravou a levou stranu každé tabulky a pak také pro sloupce CD, CE, CF.

6.3.6 Experimentální výsledky

Sekce experimentálních výsledků je rozdělena na dvě části.

A. Chování testovaných DM při odstranění distorze u uměle poškozených obrazů – sada distorzních polynomů s jedním koeficientem. **B.** Chování testovaných DM při odstranění distorze u uměle poškozených obrazů – sada distorzních polynomů s více koeficienty.

6.3.6.1 Chování testovaných division-modelů při odstranění distorze u uměle poškozených obrazů – sada distorzních polynomů s jedním koeficientem

V první fázi bylo využito 119 obrazů – viz Obr. (6.3).1B, které byly uměle zatíženy distorzí s využitím polynomů S1-S6. Celkem 15113 hodnot ε poskytl dostatečný statistický soubor pro ohodnocení testovaných modelů. Na Obr. (6.3).3 a (6.3).4 jsou finální hodnoty ε v 3D grafech. Separátně pro každý distorzní polynom (S_x). Grafy udávají schopnost použitých DM opravit chybu distorze ať už soudkovou (hodnoty mínus) nebo polštářovou (hodnoty plus). Z grafů je dobře patrné, že testované DM podávají téměř shodné výsledky především tam, kde je poškozený obraz zatížen vadou s využitím distorzního polynomu $\mathcal{F}(r)$ s vysokou hodnotou mocniny a současně pokud korekční polynom $\mathcal{F}_C(r)$ resp. použitý DM obsahuje velké množství mocninných členů – viz Obr. (6.3).4F. Toto pravidlo má překvapivě obecných charakter. Pokud je použit distorzní polynom s nízkou hodnotou mocniny, tj. polynomy S1, S2 a S3, pak schopnost korigovat distorzi mají opět nejvíce polynomy s vyšším počtem mocninných členů. Čím více tím lépe. V Tabulce 1 je detail výsledků hodnot ε pro distorzní polynom S1. Je poměrně dobře patrné, že schopnost perfektně odstranit vadu distorze záleží nejen na zvoleném korekčním polynomu $\mathcal{F}_C(r)$, ale také na velikosti a typu distorze.

| st.p./úroveň | +0,2 | +0,1 | -0,1 | -0,2 | -0,3 | -0,4 | -0,5 | -0,6 |
|--------------|--------|--------|--------|--------|--------|--------|--------|---------|
| -2----- | 3129,3 | 1653,7 | 1357,5 | 3129,3 | 2517,6 | 2928,2 | 3220,4 | 3424,6 |
| 12----- | 1168,7 | 1046,2 | 1038,5 | 1168,7 | 1107,9 | 1129,5 | 1148,3 | 1140,8 |
| --3----- | 5073,3 | 2482,1 | 2028,5 | 5073,3 | 4612,3 | 5589,0 | 6382,4 | 7033,1 |
| 1-3----- | 1397,1 | 1095,3 | 1081,1 | 1397,1 | 1359,4 | 1482,0 | 1581,2 | 1661,6 |
| -23----- | 1316,3 | 1084,2 | 1041,6 | 1316,3 | 1157,2 | 1227,7 | 1284,3 | 1300,7 |
| 123----- | 1035,5 | 1023,6 | 1021,5 | 1035,5 | 1026,8 | 1029,6 | 1041,6 | 1019,2 |
| ---4----- | 6619,0 | 3172,5 | 2632,4 | 6619,0 | 6336,4 | 7776,0 | 8984,3 | 10011,7 |
| 1--4----- | 1624,7 | 1152,2 | 1134,3 | 1624,7 | 1628,5 | 1852,6 | 2040,9 | 2208,9 |
| -2-4----- | 1517,4 | 1130,9 | 1059,6 | 1517,4 | 1265,9 | 1372,7 | 1454,3 | 1489,6 |
| 12-4----- | 1045,0 | 1025,7 | 1022,0 | 1045,0 | 1030,5 | 1034,1 | 1047,0 | 1025,4 |
| -2----- | 3129,3 | 1653,7 | 1357,5 | 3129,3 | 2517,6 | 2928,2 | 3220,4 | 3424,6 |

Tabulka (6.3).1. Výsledné hodnoty z Obr. (6.3).3A v detailu. Liché řádky-klasická metoda DM, Sudé řádky- λDM metoda. st.p. – vybrané stupně korekčního polynomu.

Toto pravidlo platí jednoznačně pro všechny testované DMs. Z obecného hlediska byl DM (Fitzgibbon 2001) navržen především proto, aby bylo možné korigovat vadu distorze v co možná nejkratším čase. Pokud se opět podíváme detailněji na některé dosažené hodnoty ε z grafů na Obr. (6.3).3A zobrazené v Tab. (6.3).1, je vidět, že klasický DM koriguje kulovou vadu dobře, s ohledem na schopnosti běžného lidského zraku, ale λDM poskytuje lepší výsledek. Na Obr. (6.3).5 a (6.3).6 jsou výsledky hodnot Δ pro jednotlivé polynomy S_x a testované modely. Všude tam, kde je hodnota Δ větší než nula, podával λDM lepší výsledky než Fitzgibbonův model DM a ostatní testované DM. V téměř 99% případů poskytuje

λDM lepší výsledky než klasický DM . V mnoha případech jsou rozdíly malé, do 10 pixelů. To se týká především korekčního polynomu $\mathcal{F}_C(r)$, kde je použit velký počet mocninných členů; obvykle větší než 3. Toto pravidlo je platné pro všechny typy testovaných obrazů – viz Obr. (6.3).5A,B,C a (6.3).6A,B,C. Platí, že čím nižší je stupeň použitého distorzního polynomu, tím jsou rozdíly mezi DM a λDM větší. To je patrné především na Obr. (6.3).5A. Lze tedy říci, že λDM se vyplatí použít především tam, kde je distorted obraz zatížený velkou kulovou vadou – polynom S1. Dále pak všude tam, kde je vyžadován malý počet členů korekčního polynomu a současně kvalitní výsledek. Počet testovaných korekčních polynomů $\mathcal{F}_C(r)$ byl 127. Byly tedy testovány všechny kombinace polynomu s max. mocninou 7. Bylo zjištěno, že pro korekci vady distorze s využitím polynomů S1 až A6 je možné použít značné množství kombinací mocninných členů korekčního polynomu se stejným nebo téměř stejným výsledkem. To platí jak pro λDM tak i pro ostatní DM .

Z hlediska výkonnosti testovaných modelů je důležitý především počet selhání nově navrženého λDM . Výsledky jsou zobrazeny na Obr. (6.3).7 a (6.3).8 opět pro jednotlivé distorzní polynomy S1 až S6. V každém grafu je zobrazen jednak počet selhání, resp. počet záporných hodnot Δ , dále minimální nalezená hodnota ε a pak také hodnota std. dev. (směrodatná odchylka). Vše pro jednotlivé testované korekční polynomy. Počet selhání je počítán pro měnící se hodnoty jednotlivých koeficientů u distorzních polynomů S1 až S6. Selhání korekčního algoritmu λDM má dva důvody: první je ten, že selhal evoluční estimátor, resp. algoritmus jDE nebyl schopen úspěšně nalézt optimum účelové funkce – viz (6.3.9). Pokud k tomu dojde, je možné výpočet opakovat a očekávat zlepšení. Druhý důvod je ten, že λDM jednoduše poskytuje horší výsledek než např. Fitzgibbon DM . Počet selhání λDM je pro distorzní polynomy S1 až S4 minimální. Řádově se jedná o jednotky selhání. Výrazná změna nastává u distorzního polynomu S6, kde DM poskytuje lepší výsledky především při větším počtu použitých mocninných členů u (6.3.8). Tento jev se však týká pouze některých kombinací; celkem u 7 různých kombinací mocninných členů korekčního polynomu $\mathcal{F}_C(r)$ – viz Obr. (6.3).8-S1. Rozdíly jsou však minimální; řádově v jednotkách pixelů. V případě opakování výpočtu jDE je možné dosáhnout lepšího výsledku, ale jen o málo. λDM i DM poskytují v takovéto konfiguraci téměř totožné výsledky. Aby byl výčet měřených hodnot kompletní, jsou na Obr. (6.3).9 a (6.3).10 zobrazeny hodnoty AVGCOL, AVGCOL a STDDEV – viz také Obr. (6.3).2.

Obrázek (6.3).9 udává především schopnost zvoleného korekčního polynomu $\mathcal{F}_C(r)$ korigovat distorzi bez ohledu na velikost koeficientů u distorzního polynomu S. Na Obr. (6.3).10 jsou pak doplňkové hodnoty průměru a směrodatné odchylky. Je poměrně zajímavé, že zatímco hodnoty std.dev. (směrodatná odchylka) logicky klesají při zmenšující se hodnotě koeficientů k_i , průměrná hodnota je pro všechny koeficienty k_i téměř neměnná.

Nicméně jedná se pouze o statistický ukazatel, který nemá vypovídací hodnotu o kvalitě konkrétní zvolené kombinace mocninných členů u korekčního polynomu $\mathcal{F}_C(r)$.

6.3.6.2 Chování testovaných division-modelů při odstranění distorze u uměle poškozených obrazů – sada distorzních polynomů s více koeficienty

V předchozí kapitole byly popsány výsledky testovaných DM s pro velmi specifická poškození obrazu. Každý výrobce se snaží co možná nejvíce korigovat optické vady, tak aby objektiv byl co nejlepší. V praxi je tedy spíše možné se setkat s vadami, které nebudou přímo odpovídat distorzním polynomům S1 až S6. Naopak vada bude odpovídat distorznímu polynomu, kde všechny koeficienty k_i budou nenulové. Mohou však být velmi malé a podílet se na vadě jen okrajově. Druhá fáze experimentů popisuje chování testovaných DM s v případě, že žádný z koeficientů u mocninných členů r^1 až r^7 nebude nulový. To odpovídá distorzním polynomům M1 až M17. Výsledky jsou zobrazeny na Obr. (6.3).11. Opět je zobrazen graf hodnot ε a Δ a graf udávající počet selhání navrženého modelu λDM .

Pokud se v grafech hodnot na Obr. (6.3).3, (6.3).4 vyskytovaly výrazné peaky – nepřesnosti při korekcích distorze, pak stejné výsledky je možné získat i v případě, že žádný koeficient k_i distorzního polynomu nebude nulový.

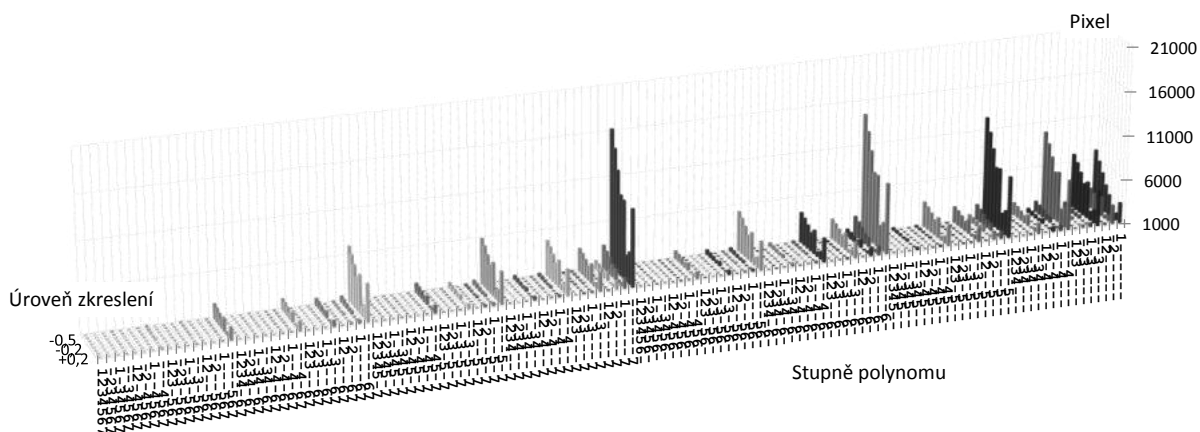
Na Obr. (6.3).11 nahoře je graf hodnot ε . Je poměrně dobře patrné, že v mnoha případech je rozdíl v řádu tisíců pixelů ve prospěch λDM . Největší rozdíly jsou všude tam, kde je použit malý počet členů korekčního polynomu. To odpovídá především situaci, kdy je použita (6.3.4) versus (6.3.8). Je zřejmé, že výhody (6.3.8) jsou především v malém počtu členů korekčního polynomu. Jestliže se počet členů zvyšuje např. až na 7, pak oba všechny testované modely podávají přibližně stejné výsledky. λDM je jen o málo lepší. Na Obr. (6.3).3 uprostřed je zobrazen graf diferencí λDM versus DM .

λDM využívá vždy koeficient k_1 a je dobře patrné, že rozdíly jsou největší, právě když je počet členů korekčního polynomu malý. Spojitost mezi velikostí distorze (osa koeficientů k_i u $\mathcal{F}(r)$) a typem korekčního polynomu nebyla prokázána. λDM i DM poskytují stejné výsledky bez ohledu na velikost distorze. V případě, že počet členů korekčního polynomu $\mathcal{F}_C(r)$ je větší, pak jsou rozdíly mezi testovanými DM výrazně menší. Pokud je použito všech 7 stupňů polynomu, pak všechny testované modely poskytují téměř stejné výsledky. Aby bylo zřejmé, jak je navržený λDM účinný, je na Obr. (6.3).11 dole zobrazen graf počtu selhání modelu λDM resp. počet korekcí, kdy DM byl lepší než λDM .

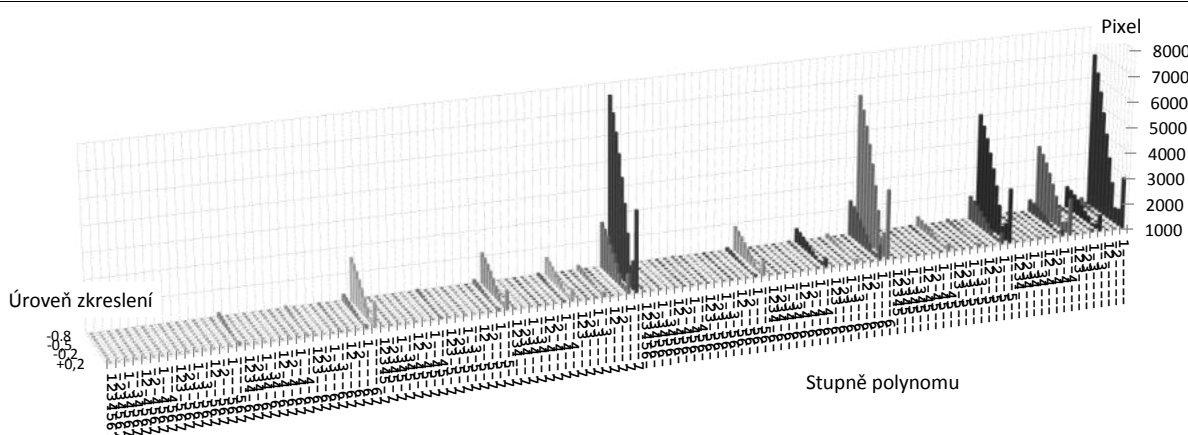
Na Obr. (6.3).11 dole je graf, který udává počet neúspěšných pokusů o korekci distorze pro λDM . V porovnání s (S1) až (S6) – viz Obr (6.3).7, (6.3).8 je schopnost odstranit kombinovanou optickou vadu výrazně lepší i pokud je přítomna složka odpovídající mocnině r^7 . Stále jsou zde kombinace, které pro klasický DM resp. Fitzgibbonův model poskytují lepší výsledek; je jich však jen několik. V tomto případě se spíše jedná o nahodilé případy, a především selhání optimalizátoru jDE. Např. pokud je použita kombinace mocninných členů korekčního polynomu (-234567 a 1234567) a distorzní polynom M4, pak je výsledek pro λDM resp. kombinaci (1234567) při prvním výpočtu 1018.04 a při opakovaném výpočtu 1017.96 pixelů. Pro zvolený DM (-234567) je to při prvním výpočtu 1017.98 a při opakovaném výpočtu 1018.24 pixelů. Tyto hodnoty jsou přímo zaneseny v grafu na Obr. (6.3).11 nahoře. Uprostřed je pak rozdíl z daných hodnot. Řešením těchto malých nepřesností je např. zvýšení počtu jedinců a generací u algoritmu jDE např. $N_{POP} = 300$, $N_{GEN} = 300$. To však výrazně prodlouží výpočet. Dostáváme se tak do stejné situace jako např. (Asari 2004), kdy je nutné použít programovatelné logické pole z důvodu obrovských výpočetních nároků.

Při všech výpočtech byly použity následující parametry optimalizátoru jDE viz také (Moravec & Hub 2015): ‘primary-jDE’: $N_{POP} = 60$, $F_l = 0.7$, $F_u = 0.3$, $\tau_1 = 0.5$, $\tau_2 = 0.5$; $\overline{V_p} = \text{randtobest}/1/\text{bin}$, $N_{GEN} = 120$, $D_{im} = 7$, ‘secondary-jDE’: $N_{POP} = 20$, $F_l = 0.7$, $F_u = 0.3$, $\tau_1 = 0.5$, $\tau_2 = 0.5$; $\overline{V_p} \equiv \text{randtobest}/1/\text{bin}$, $N_{GEN} = 40$, $D_{im} = 2$. Byly testovány i různé hodnoty τ_1, τ_2 v rozsahu (0.1,1.0), ale hodnota 0.5 poskytovala nejlepší výsledky. Čísla v rozsahu (0.45,0.55) jsou vhodná taktéž. Je samozřejmě možné volit i jiná nastavení. Doba výpočtu odstranění distorze u jednoho obrazu na dostupném Athlon64X2-4800+/3.5GBDDR2-800 je 67 minut. Celkem bylo provedeno 17272 korekcí obrazů. Algoritmizace v MSVS 2008, 2019 C++/CLI, 32bitová, jedno vláknová aplikace.

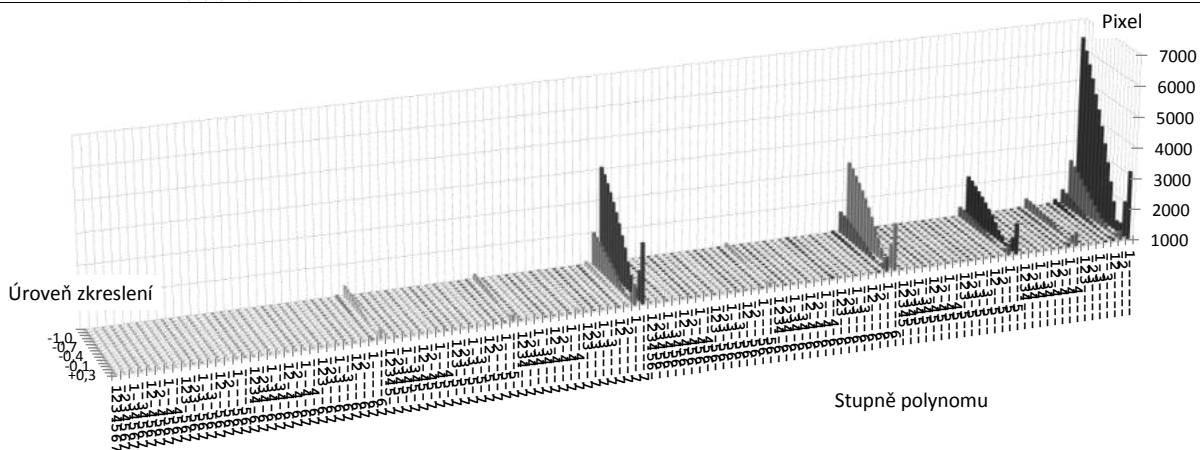
$$(A) (S1) \mathcal{F}(r) = +1.0r^1 + \langle -0.6; +0.2 \rangle r^2 + 0.0r^3 + 0.0r^4 + 0.0r^5 + 0.0r^6 + 0.0r^7$$



$$(B) (S2) \mathcal{F}(r) = +1.0r^1 + 0.0r^2 + \langle -0.8; +0.2 \rangle r^3 + 0.0r^4 + 0.0r^5 + 0.0r^6 + 0.0r^7$$

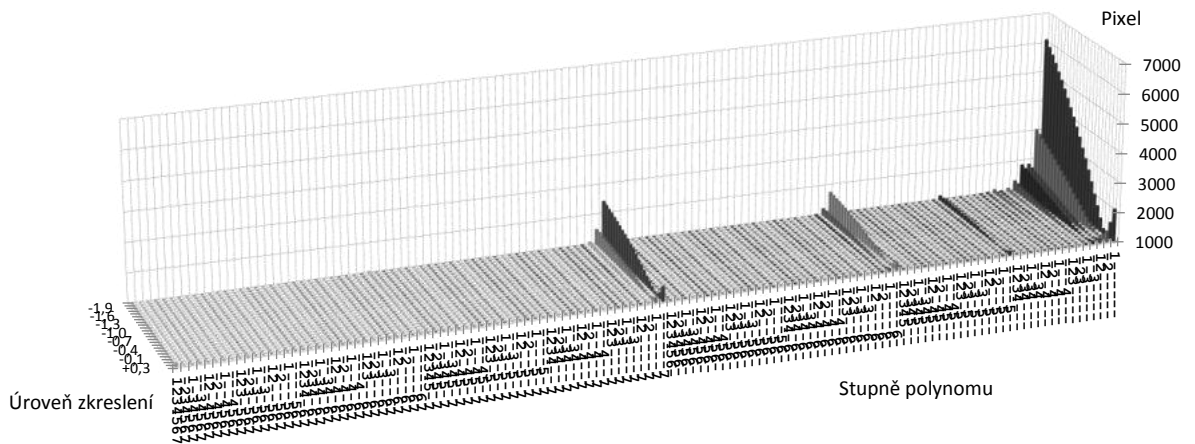


$$(C) (S3) \mathcal{F}(r) = +1.0r^1 + 0.0r^2 + 0.0r^3 + \langle -1.2; +0.3 \rangle r^4 + 0.0r^5 + 0.0r^6 + 0.0r^7$$

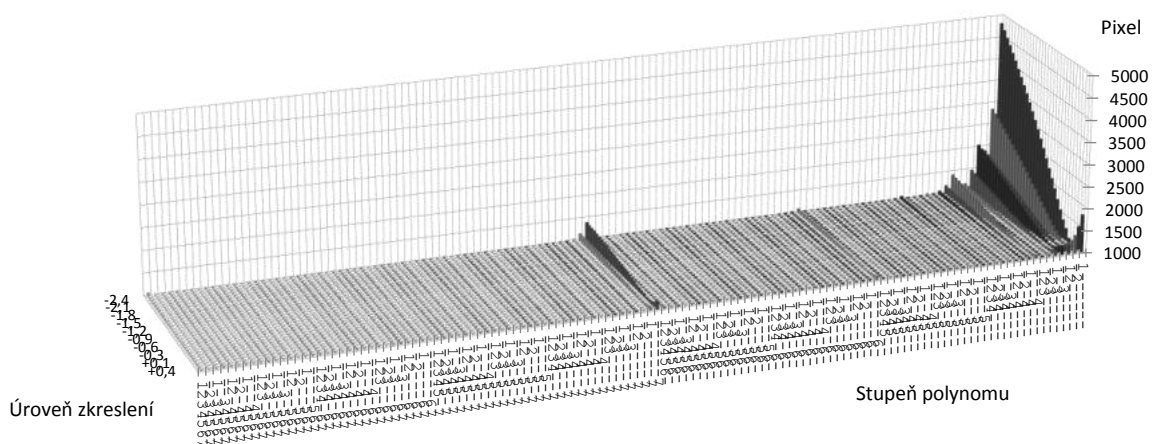


Obrázek (6.3.3) Výsledné hodnoty ε pro distorzní polynomy S1, S2, S3. Korekční polynomy $\mathcal{F}_c(r)$ využívají různý počet a konfiguraci mocninných členů – viz. jednotlivé grafy.

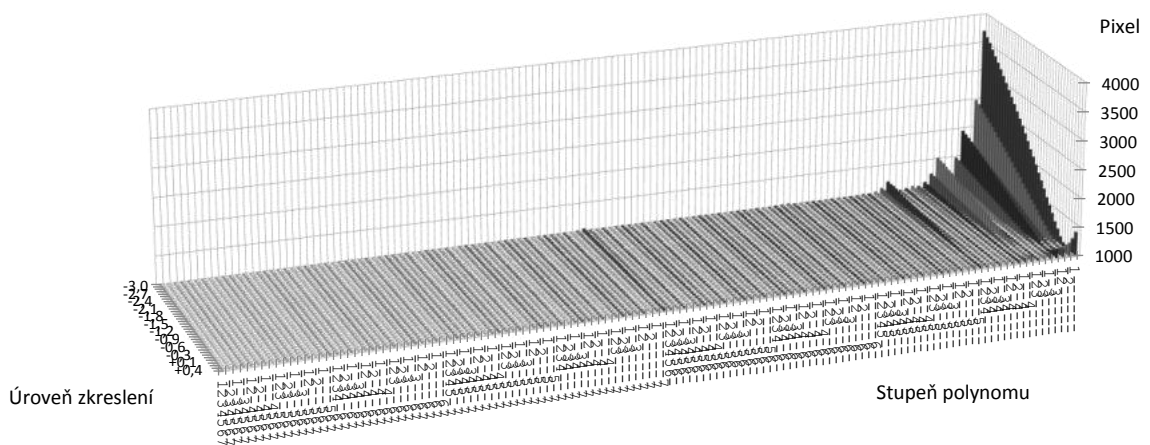
$$(D) (S4) \mathcal{F}(r) = +1.0r^1 + 0.0r^2 + 0.0r^3 + 0.0r^4 + \langle -2.0; +0.3 \rangle r^5 + 0.0r^6 + 0.0r^7$$



$$(E) (S5) \mathcal{F}(r) = +1.0r^1 + 0.0r^2 + 0.0r^3 + 0.0r^4 + 0.0r^5 + \langle -2.5; +0.4 \rangle r^6 + 0.0r^7$$

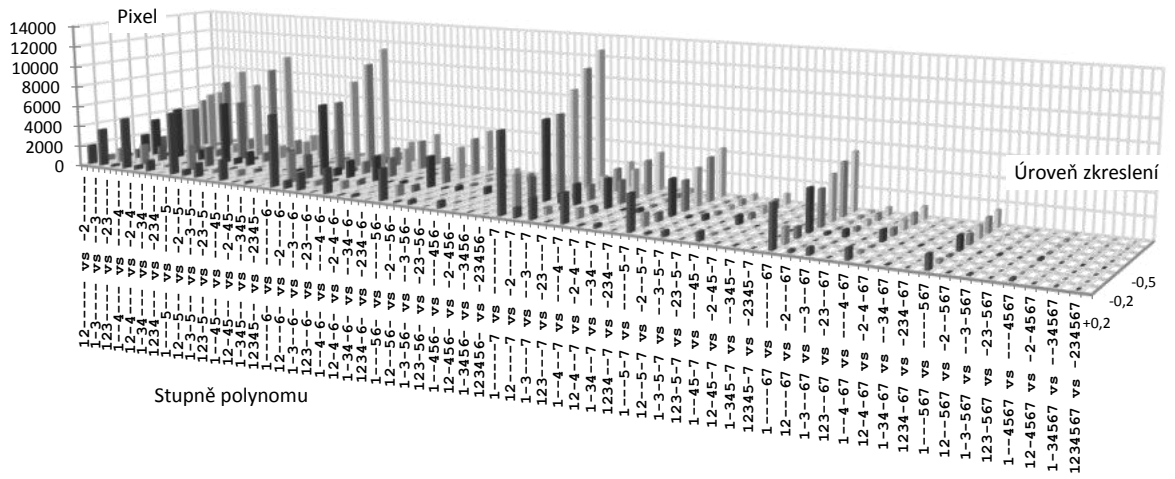


$$(F) (S6) \mathcal{F}(r) = +1.0r^1 + 0.0r^2 + 0.0r^3 + 0.0r^4 + 0.0r^5 + 0.0r^6 + \langle -3.05; +0.4 \rangle r^7$$

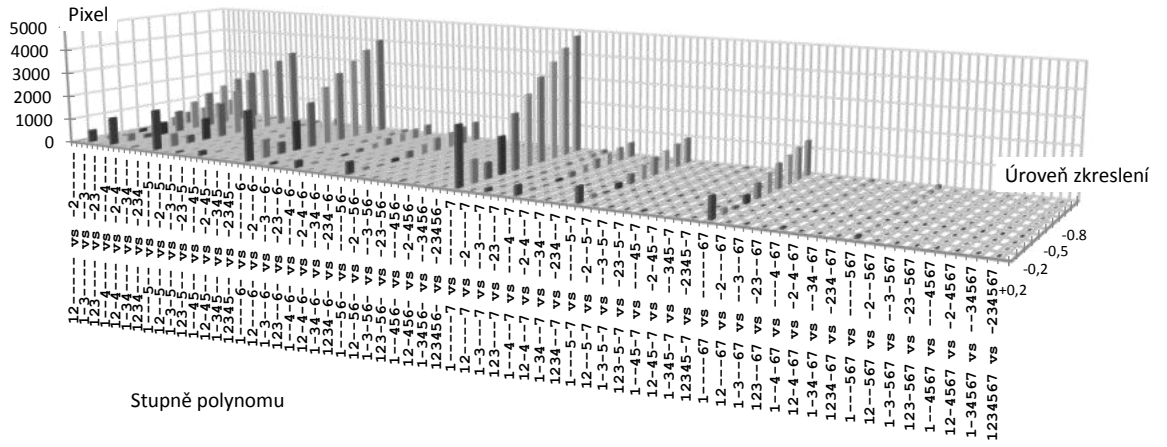


Obrázek (6.3).4 Výsledné hodnoty ε pro distorzní polynomy S4, S5, S6. Korekční polynomy $\mathcal{F}_c(r)$ využívají různý počet a konfiguraci mocninných členů – viz. jednotlivé grafy.

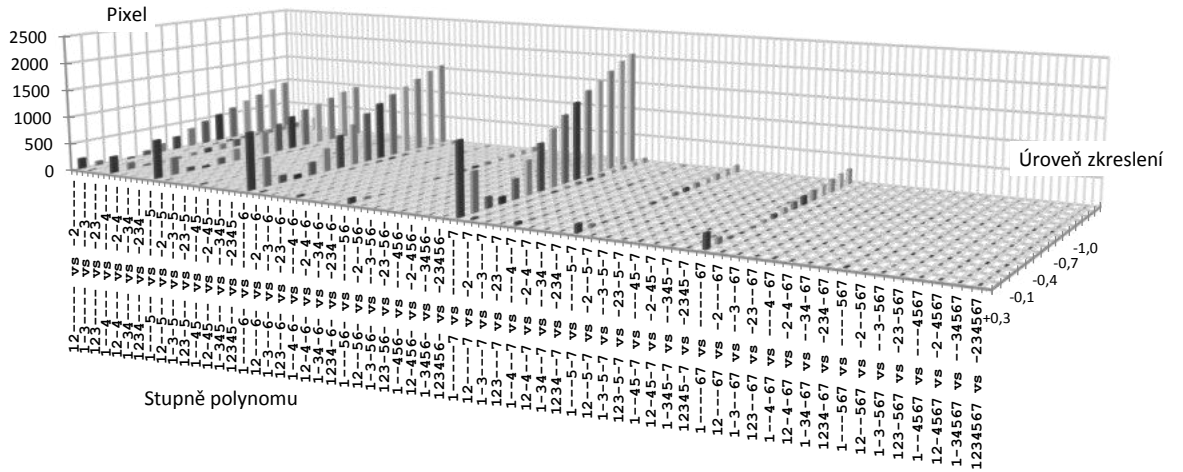
(A) (S1) Rozdíly pro (S1); r^2



(B) (S2) Rozdíly pro (S2); r^3

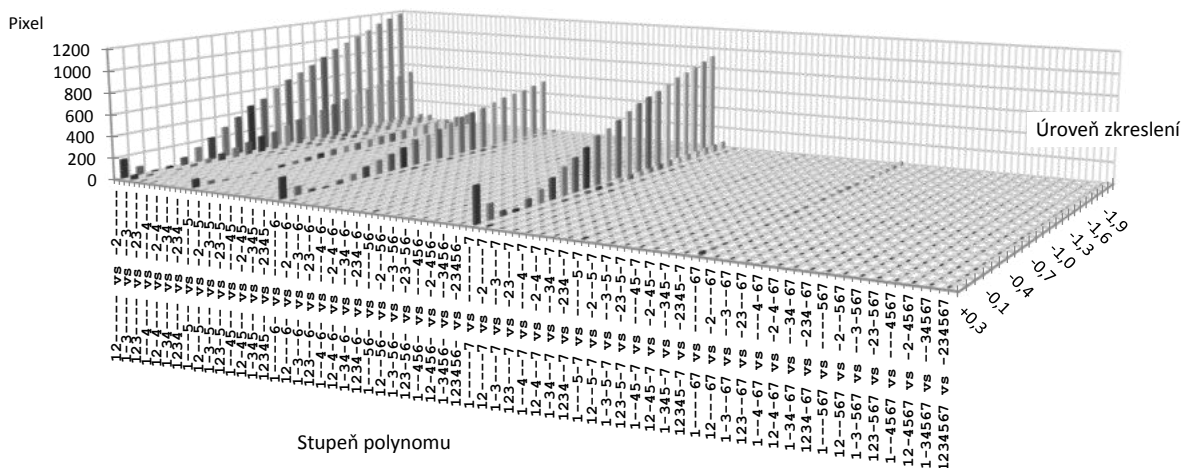


(C) (S3) Rozdíly pro (S3); r^4

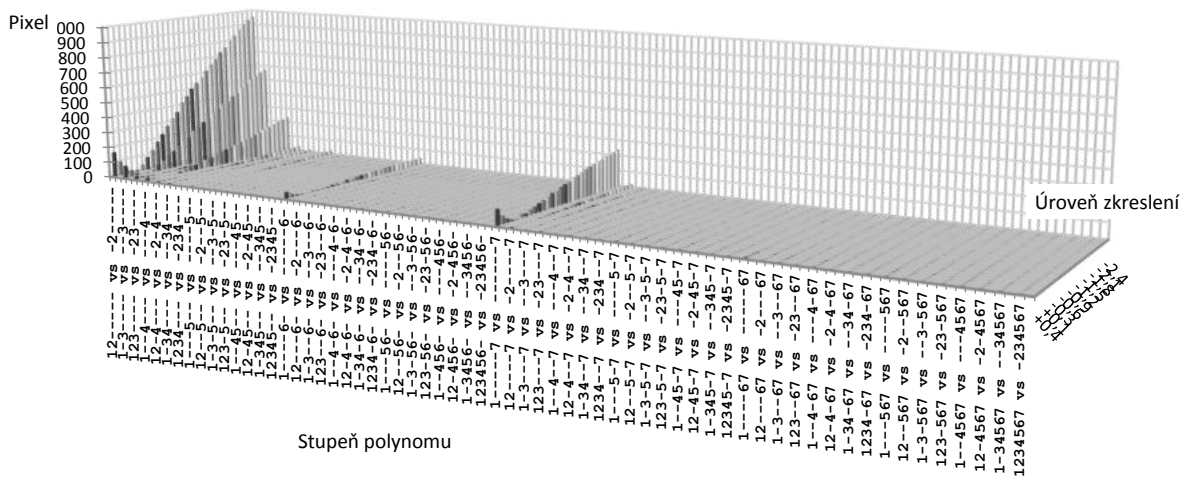


Obrázek (6.3).5 Grafy A, B, C zobrazují hodnoty Δ pro jednotlivé distorzní polynomy S1, S2, S3 a pro zvolené kombinace korekčního polynomu $\mathcal{F}_C(r)$ a velikosti distorze.

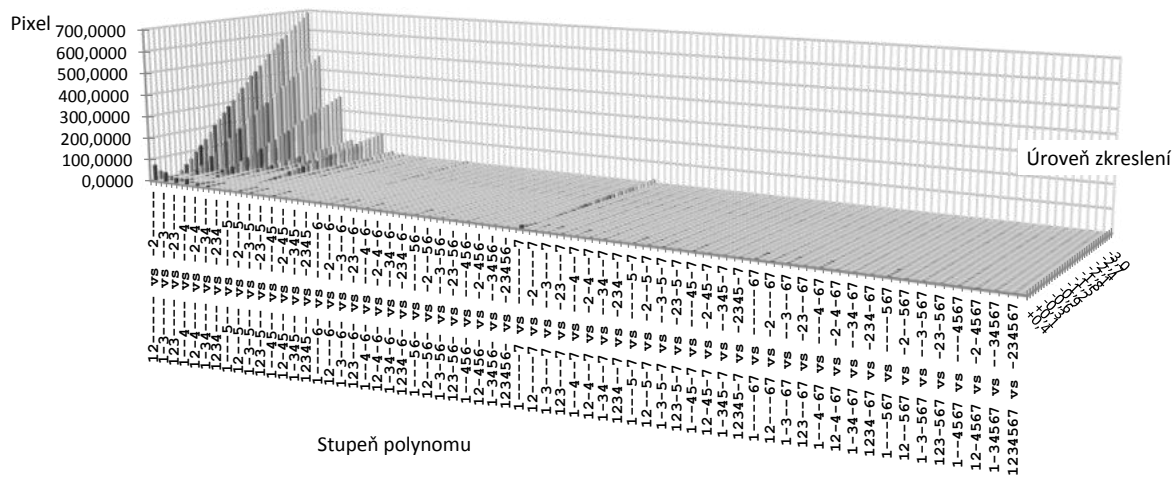
(D) (S4) Rozdíly pro (S4); r^5



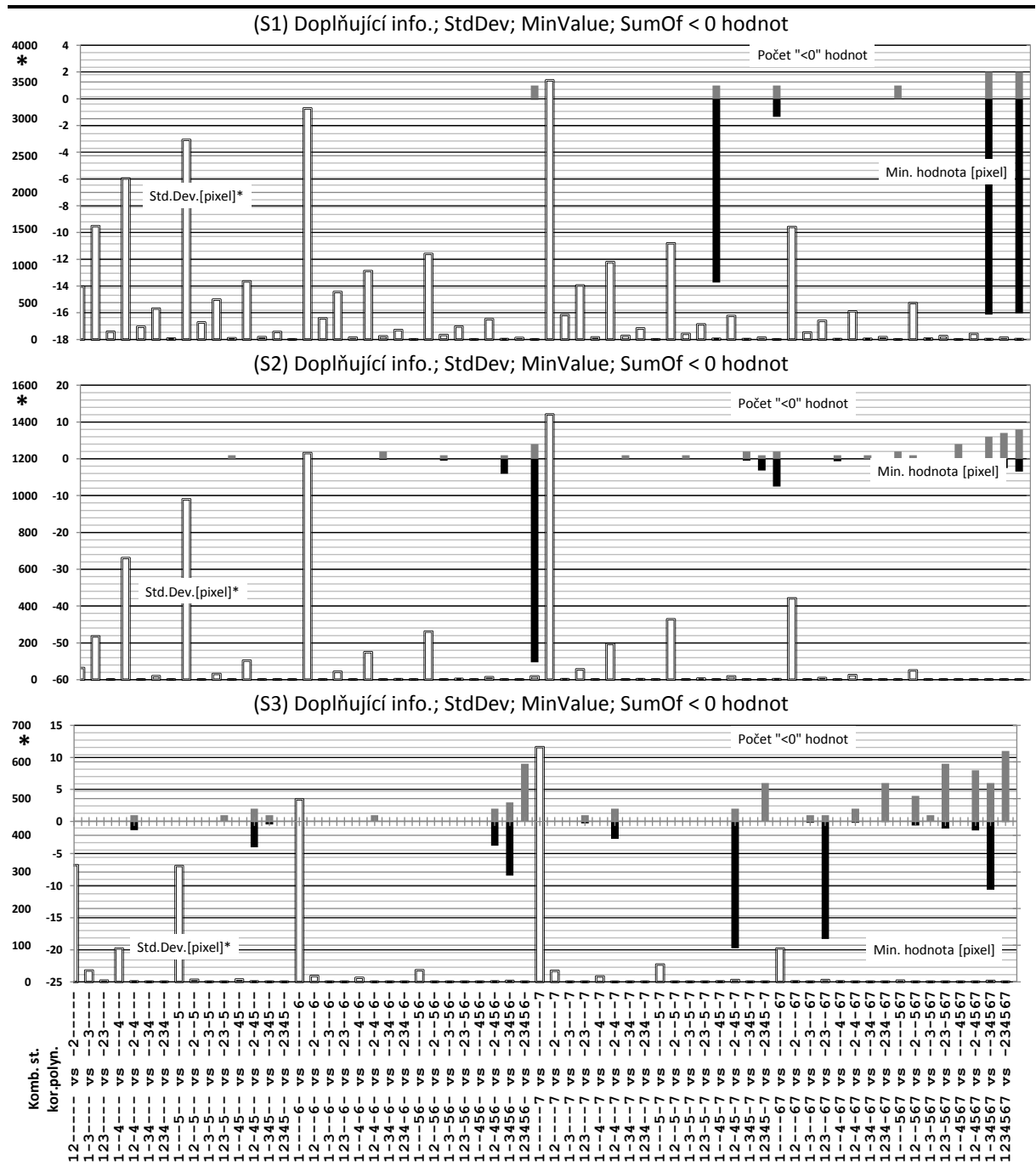
(E) (S5) Rozdíly pro (S5); r^6



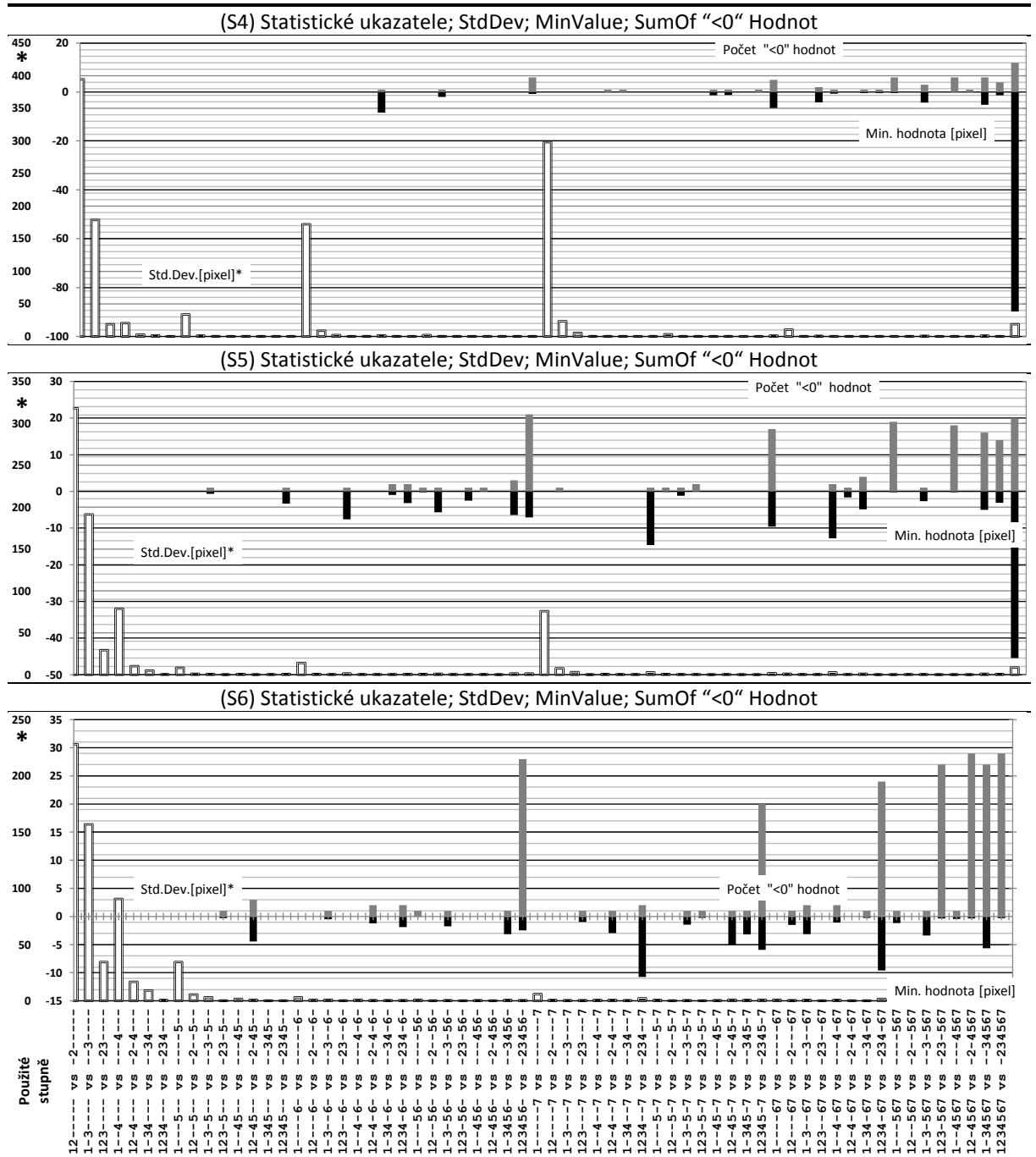
(F) (S6) Rozdíly pro (S7); r^7



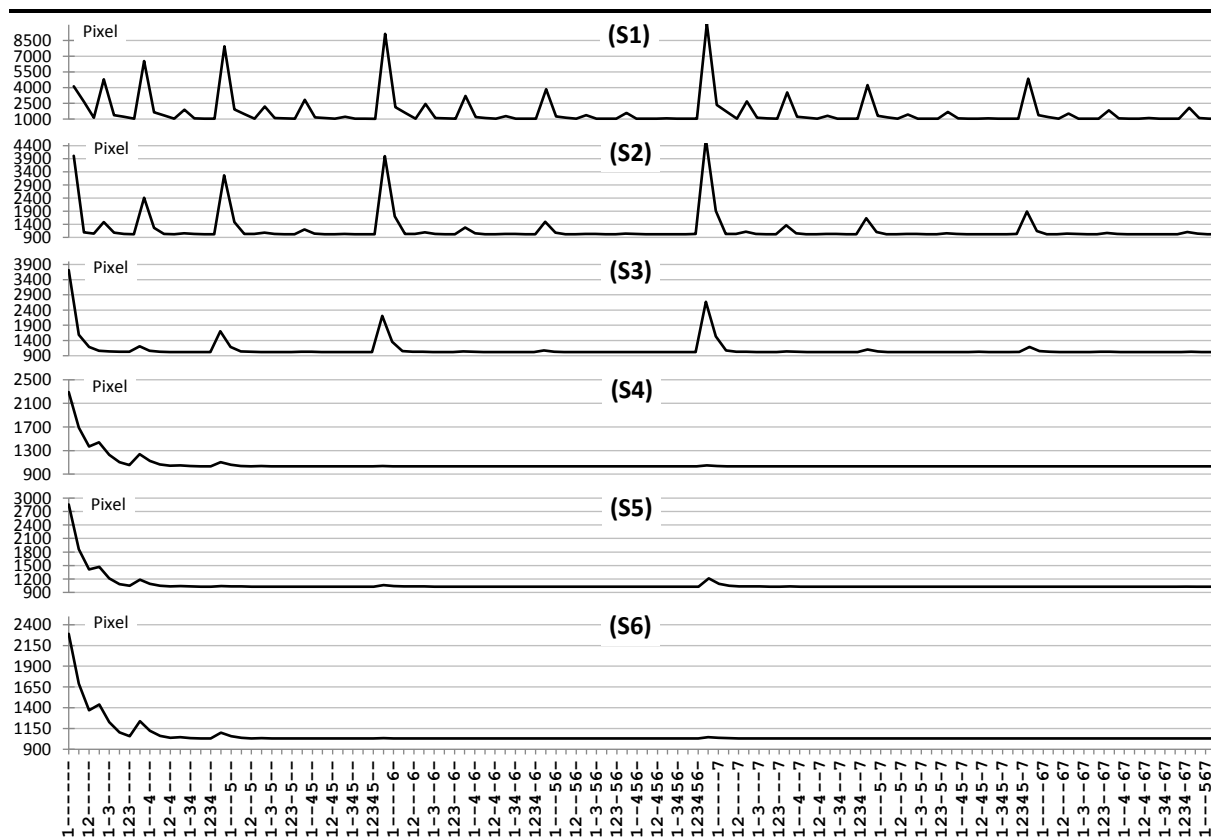
Obrázek (6.3).6 Grafy D, E, F zobrazují hodnoty Δ pro jednotlivé distorzní polynomy S4, S5, S6 a pro zvolené kombinace korekčního polynomu $\mathcal{F}_c(r)$ a velikosti distorze.



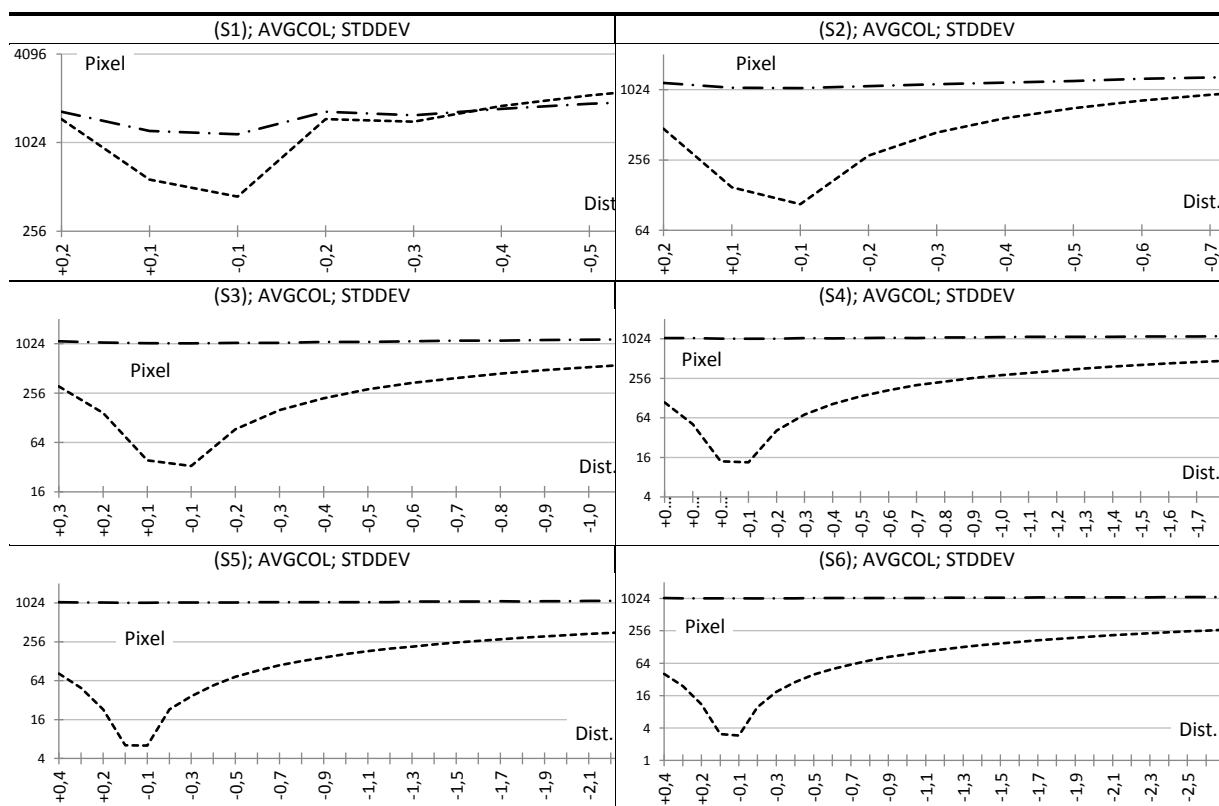
Obrázek (6.3).7 Doplňující informace pro polynomy (S1, S2, S3) – viz Obr. (6.3).3 - (6.3).6. StdDev – udává směrodatnou odchylku. MinValue - udává minimální hodnotu ϵ . SumOf < 0 - udává počet hodnot, které jsou menší než nula. Další informace na Obr. (6.3).2.



Obrázek (6.3).8 Statistické ukazatele pro polynomy (S4, S5, S6) - viz Obr. (6.3).3 - (6.3).6. Počet hodnot, které jsou menší než nula, minimální hodnota pro každé uspořádání ekvalizačního (korekčního) polynomu a směrodatná odchylka. Detailní popis a rovnice, které byly použity při výpočtech viz Obr. (6.3).2.



Obrázek (6.3).9 Statistické ukazatele: AVGGROW pro polynomy (S1)..(S6) – viz Obr. (6.3).3, (6.3).4.



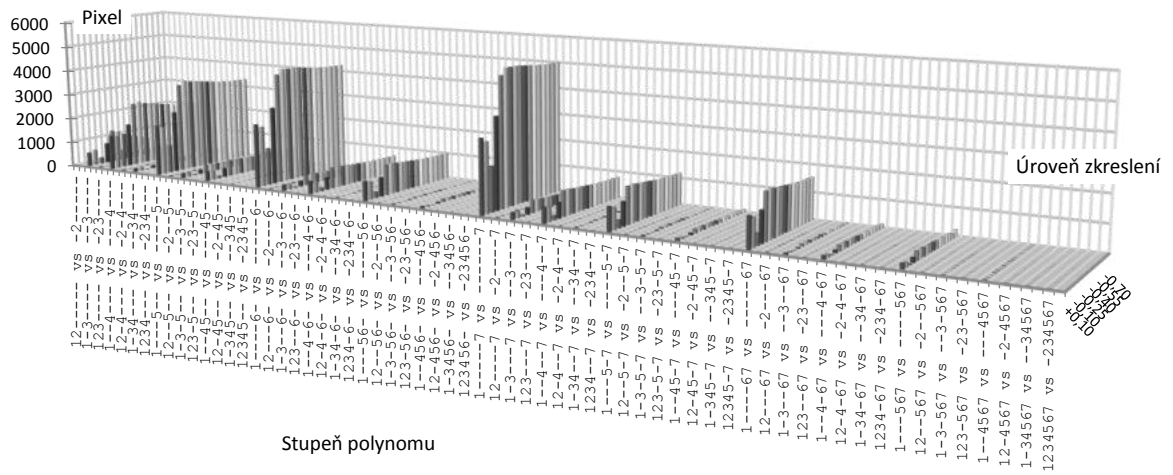
Klíč: — AVGCOL - - - - - STDDEV

Obr. (6.3).10 Statistické ukazatele: AVGCOL a STDDEV pro polynomy (S1)..(S3) - viz Obr. (6.3).3 - (6.3).6.

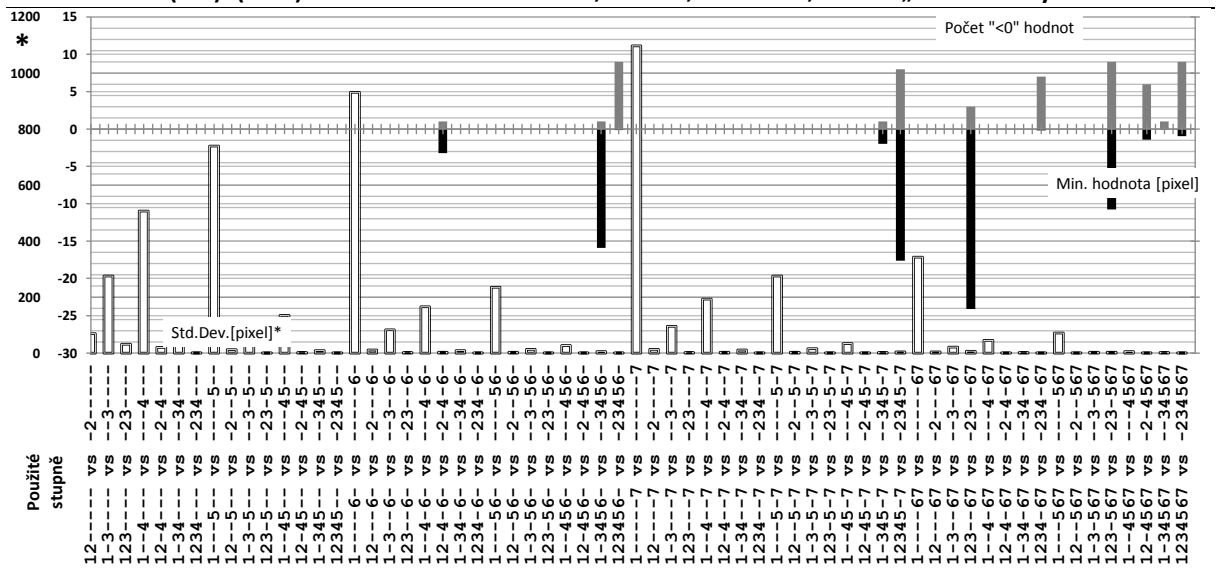
Výsledky přesnosti při odstranění distorze – polynomy (M1)..(M17)



Finální výsledky přesnosti pro polynomy (M1)..(M17)



(M1)..(M17) Statistické ukazatele: Info; StdDev; MinValue; SumOf „<0“ hodnoty



Obrázek (6.3).11 Experimentální výsledky s polynomy: (M1)..(M17). **Nahore**-finální výsledek ε pro všechny testované polynomy M_{xx} . **Uprostřed**-hodnoty Δ . **Dole**-hodnoty směrodatných odchylek, min. hodnoty fitness a počet negativních výsledků.

6.3.7 Závěr a diskuze

V této kapitole byl navržen a otestován nový matematický model, který je určen pro odstranění distorze obrazu u objektivů typu Rybí Oko (Fish-Eye) nebo tzv. ‘pinhole‘ objektivů. Model vychází z dříve publikovaných prací jako je např. Fitzgibbonův ‘division model‘ (Fitzgibbon 2001). Nově navržený model označený jako λDM je vyjádřen jako čistě racionální lomená polynomiální funkce. Nalezení příslušných koeficientů polynomu je provedeno pomocí metaheuristického optimalizátoru, který pracuje až v sedmi-dimenzionálním prostoru s volitelným počtem a kombinací stupňů. Díky rozsáhlému souboru experimentů lze říci, že λDM model podává výrazně lepší výsledky než Fitzgibbonův model. Tento závěr se však podařilo dosáhnout jen díky značnému množství experimentálních dat a poměrně dlouhému výpočetnímu času, který bylo nutné věnovat ověření nově navrženého modelu. Návrh nového korekčního modelu je bohužel spíše alchymistickou a časově nesmírně náročnou záležitostí viz (Fitzgibbon 2001). Neexistuje zde nějaký návod, který by umožnil odvodit novou metodu na základě již existujících. Vždy se jedná o metodu pokus-omyl. Celý model je zaměřen na korekci centrické radiální distorze. Je však možné jej rozšířit i o možnost korekce necentrické radiální distorze tak, že k polynomu přidáme možnost posunu kalibračního obrazu v osách X a Y a případně také natočení vůči ose X. Přidáním dalších 3 dimenzí by se úloha rozšířila na desetidimenzionální. To samé platí i o možnost rozšířit úlohu tak, aby se kalibrační obraz mohl pohybovat v prostoru. Přidáním dalších šesti dimenzí, které by zahrnovali i posun v osách X, Y, Z a rotaci okolo os X, Y, Z by se celá úloha stala třinácti-dimenzionální. Výpočetní náročnost by tak značně narostla, a to až za hranice možností běžné kancelářské výpočetní techniky, která je nyní (2020) k dispozici.

LITERATURA

- Asari K.V., Kumar S., Radhakrishnan D. 1999. A new approach for nonlinear distortion correction in endoscopic images based on least squares estimation. *IEEE Transaction on medical imaging*. vol. 18, no. 4, pp. 345-354.
- Asari K. V. 2004. Design of an efficient VLSI architecture for non-linear spatial warping of wide-angle camera images. *Journal of System Architecture*. vol. 50, no. 12, pp. 743-755, 2004.
- Basu A., Licardie S. 1995. Alternative models for fish-eye lenses. *Pattern recognition letters*. vol. 16, pp. 433-441.
- Brest J., Greiner S., Boškovič B., Mernik M., Žumer V. 2006. Selfadapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*. vol. 10, no. 6, pp. 646-657.
- Brest J., Zamuda A., Fister I., Maucec M.S. 2010. Large Scale Global Optimization using Self-adaptive Differential Evolution Algorithm. *IEEE World Congress on Computational Intelligence*. vol. 1, pp. 1-8.
- Brown D.C. 1964. An advanced plate reduction for photogrammetric cameras. AF Cambridge Research Laboratories technical report, no 64-40.
- Brown D.C. 1956. The simultaneous determination of the orientation and lens distortion of a photogrammetric camera, Air Force Missile Test Center Report no 50-20, Patrick AFB, Florida.
- Brown D.C. 1965. Advanced methods for the calibration of metric cameras, U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia 22060.
- Brown D.C. 1966. Decentering distortion of lenses, D. Brown Associates, Inc. Eau Gallie, Florida.
- Brown C.D. 1971. Close range camera calibration, Symposium on close range photogrammetry, Urbana, Illinois, DBA Systems, Inc. Melbourne.
- Brito J.H., Angst R., Koser K., Pollefeys M. 2013. Radial Distortion Self-Calibration. In: *CVPR 2013*. pp. 1368-1375.
- Bukhari F., Dailey M.N. 2013. Automatic Radial Distortion Estimation from a Single Image. *Journal of Mathematical Imaging and Vision*. vol. 45, no. 1, pp 31-45.
- Burchardt C.B., Voss K. 2001, A new algorithm to correct fish-eye and strong wide-angle-lens-distortion from single images. In: *IEEE ICIP'01*. vol. 1, pp. 225-228.
- Cedrés D.S., Gomez L., Flores M.A., Salgado A., Esclarín J., Mazonra L., Alvarez L. 2015. An Iterative Optimization Algorithm for Lens Distortion Correction Using Two-Parameter Models. *Image Processing On Line*. vol. 6, pp. 326–364. [cit. 8.5.2020] <https://doi.org/10.5201/ipol.2016.130>.
- Conrady A.E. 1919. Decentred lens-systems. *Monthly Notices of the Royal Astronomical Society*. vol. 79, p.384-390.
- Cucchiara R., Grana C., Prati A., Vezzani R. 2003. A Hough Transform-based method for Radial Lens Distortion Correction. In: *12th International Conference on Image Analysis and Processing*. pp. 182-187.
- Das S., Suganthan P.N. 2010. Differential Evolution: A Survey of the State-of-the-Art. *IEEE Transactions on Evolutionary Computation*. vol. 15, no 1., pp. 4-31.
- Das A., Patil D. 2006. Efficient Measurement and Correction techniques for Lens Distortion, Technical report, EE362 Project CANON co., Stanford Center for Image Systems Engineering.
- Devernay F., Faugeras O. 2001. Straight Lines Have to Be Straight. *Machine Vision and Applications*. vol. 13, no. 1 pp. 14-24.
- Dhane P., Kutty K., Bangadkar S. 2012. A Generic Non-Linear Method for Fisheye Correction. *International Journal of Computer Applications*. vol. 51, no. 10, pp. 58-65.
- Eiben, A.E., Hinterding R., Michalewicz Z. 1999. Parameter control in evolutionary algorithms. *IEEE Transaction on evolutionary computation*. vol. 3, no. 2, pp. 124-141.
- Fischer R.F., Tadic B. 2000. *Optical System Design*. McGraw Hill, p. 599. ISBN-13: 978-0071349161.

- Fitzgibbon A.W. 2001. Simultaneous linear estimation of multiple view geometry and lens distortion. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. vol. 1, pp. 125-132.
- Flores M.A., Alvarez L., Gomez L., Cedres D.S. 2014. Automatic Lens Distortion Correction Using One-Parameter Division Models. *Image Processing On Line*. vol. 4, pp. 327–343.
- Floudas Ch.A., Pardalos P. M. 2009. *Encyclopedia of Optimization*. Springer, 2nd ed., p. 4626, 7 volumes. ISBN 978-0-387-74759-0.
- Gamperle R., Muller S.D., Koumoutsakos. P. 2002. A parameter study for differential evolution. In: *WSEAS International Conference on Advances in Intelligent Systems*. pp. 293-298.
- Goldberg D.E. 1987. Simple genetic algorithms and the minimal deceptive problem. In: *Genetic Algorithms and Simulated Annealing*, London, Pitman, pp. 74-88.
- Holland J.H. 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, p. 232. ISBN: 9780262581110.
- Hughes C., Glavin M., Jones E., Denny P. 2008. Review of geometric distortion compensation in Fish-eye cameras. In: *Signals and Systems Conference ISSC 2008*. IET Irish, Galway, Ireland, pp. 162-167.
- Kim W., Kim Ch. 2009. An efficient correction method of wide-angle lens distortion for surveillance systems. In: *IEEE International Symposium on Circuits and Systems*. pp. 3206-3209.
- Lenz R. 1987. Linsenfehlerkorrigierte Eichung von Halbleiterkameras mit Standardobjektiven für hochgenaue 3D Messungen in Echtzeit, In: *Mustererkennung 1987*, Erwin Paulus ed., Informatik-Fachberichte, Springer Berlin Heidelberg, vol. 149, pp. 212-216.
- Liu J., Lampinen J. 2002. On setting the control parameter of the differential evolution method. In: *8th International Conference Soft Computing*. pp. 11-18.
- Magill A.A. 1955. Variation in distortion with magnification. *Journal of Research of the National Bureau of Standards*. vol. 54, no. 3, pp. 135–142.
- Moravec J. 2015. A Comparative Study: L1-norm vs. L2-norm, point-to-point vs. point-to-line Metric, Evolutionary Computation vs. Gradient Search. *Applied Artificial Intelligence*. Special edition on robotics research, vol. 29, no. 2, pp. 164 – 210.
- Moravec J., Hub M., 2015. Camera Calibration Using Direct Mapping and Adaptive Metaheuristic. *International Journal of BioScience and BioTechnology*. vol. 7, no. 3, pp. 111-120.
- Moravec J., Pošík P. 2014a. A Comparative Study: The Effect of the Perturbation Vector Type in the Differential Evolution Algorithm on the Accuracy of Robot Pose and Heading Estimation. *Evolutionary Intelligence Journal*. vol. 6, no. 3, pp. 171-191.
- Moravec J., Pošík P. 2014b. Global Robot Localization Under Noise Stress Utilizing EA Methods and Semi-semantic Classification of a Known Environment. *Applied Artificial Intelligence*. vol. 28, no. 4, pp. 360 – 417.
- Price K., Corne D., Dorigo M., Glover F. 1999. *An Introduction to Differential Evolution*. Eds. London, U.K. McGraw-Hill, pp. 79-108.
- Rechenberg I. 1973. *Evolutionsstrategies: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog Eds., Stuttgart, Germany, ISBN-13: 978-3772803734.
- Rechenberg I. 1994 *Evolutionsstrategie '94*, Frommann-Holzboog Ed., Stuttgart. ISBN 13: 9783772816420.
- Sasián J. 2012. *Introduction to Aberrations in Optical Imaging Systems*. Cambridge Press, p. 258. ISBN 9780511795183.
- Schwefel H.P. 1995. *Evolution and Optimum Seeking*. JohnWiley&Son Eds., New York, USA. ISBN: 978-0-471-57148-3.
- Smit S.K., Eiben A.E. 2009. Comparing Parameter Tuning Methods for Evolutionary Algorithms. In: *IEEE Congress on Evolutionary Computation*. pp. 399-406.
- Smit S.K., Eiben A.E. 2010a. Parameter Tuning of Evolutionary Algorithms: Generalist vs. Specialist. *Applications of Evolutionary Computation*. vol. 6024, pp. 542-551.
- Smit S.K., Eiben A.E. 2010b. Parameter Tuning for Configuring and Analyzing Evolutionary Algorithms. *Swarm and*

Evolutionary Computation. vol. 1, no. 1, pp. 19-31.

Storn R., Price K. 1997. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*. vol. 11, pp. 341-359.

Virendra N.M. 1991. *Aberration Theory Made Simple*. SPIE Press, p. 166. ISBN: 9780819488251.

Wang A., Qiu T., Shao L. 2009. A simple method of radial distortion correction with centre of distortion estimation. *Journal of Mathematical Imaging and Vision*. vol. 35, no. 3, pp. 165–172.

Wang J., Shi F., Zhang J., Liu Y. 2008. A new calibration model of camera lens distortion. *Pattern Recognition*. vol. 41, pp. 607–615.

Welford W.T. 1986. *Aberrations of Optical Systems*, CRC Press, p. 284. ISBN-13: 978-0852745649.

7 Navržené metody klasifikace kontur rukou

Biometrická identifikace osob využívající konturu lidské ruky patří mezi velmi zajímavé a stále ne zcela probádané oblasti a její přesnost a efektivita je do značné míry závislá na technických možnostech při scanování osob. Presentovaná kapitola řeší daný problém s využitím kombinace různých algoritmů. Je využita kontura ruky, topologický popis ruky, evoluční algoritmus, algoritmus lineární regrese pro odhad pozic kloubů. Pro porovnávání geometrických dat je využit algoritmus Iterative Closest Point (ICP) ve své originální podobě, a to především z toho důvodu, že u něj byla opakovaně a jednoznačně prokázána schopnost konvergence k jedinému lokálnímu optimu. Všech 5 prstů ruky je při počítačové klasifikaci plně pohyblivých, palec má 2 klouby. Právě moderní evoluční optimalizátory umožňují od základu změnit přístup k řešení podobných problémů, i když za cenu výrazně vyšších algoritmických nároků. Umožňují však citelné snížení výpočetní náročnosti algoritmu ICP. Experimentální ověření navrhovaných postupů probíhalo na třech různých databázích označovaných jako THID, GPDS a HGBD s osobami různého pohlaví a věku (cca 20-65let) s celkovým počtem osob v jednotlivých databázích 104, 94 a 60. Experimentální výsledky velmi úspěšně ověřili vhodnost použití kombinace metod ICP a evolučního optimalizátoru označovaného jako EPSDE pro řešení dané úlohy s výslednou algoritmickou složitostí $O(N)$ a úspěšností klasifikace danou koeficientem $THID:EER=0.38\%$ a $GPDS:EER=0.35\%$ na reálných snímcích.

7.1 Klasifikace kontury ruky využívající algoritmus EPSDE

Evoluční algoritmy tvoří podstatnou část navrženého algoritmu pro biometrickou identifikaci osob. Evoluční algoritmus je použit vždy jako velmi výkonný optimalizátor, jehož úkolem je najít optimální řešení dané n -dimenzionální úlohy, které je reprezentováno reálným číslem. Zkratka EPSDE (Mallipeddi & Suganthan 2010), zvolená autory, znamená v angličtině ‘Differential Evolution Algorithm with Ensemble of Parameters and Mutation and Crossover’. Do češtiny by bylo možné tento výraz přeložit jako ‘Algoritmus Diferenciální Evoluce se Souborem Parametrů s Mutací a s Křížením’. EPSDE byl zvolen jako hlavní a nejučinnější evoluční optimalizátor na základě rozsáhlých testů různých evolučních algoritmů, a především s různým nastavením pracovních parametrů. Velkou výhodou EPSDE je to, že při řešení dané úlohy, resp. při konvergenci populace k optimálnímu řešení dochází jen k mizivému počtu fatálních selhání na každý milion skutečně konvergenzí – s ohledem na počet individuí v populaci a počet generací.

7.1.1 Úvod

Biometrika a biometrické systémy osob obecně (Ross & Jain 2007) se v posledních dvou dekádách staly centrem zájmu mnoha výzkumných týmů a také soukromých firem (Gemalto 2020). Více či méně úspěšně nacházejí uplatnění v běžném životě. Existuje bezpočet různých způsobů identifikace osob založených na biometrice (Jain et al. 2000; Chauhan et al. 2010), které je možné nalézt např. v oblastech zdravotní péče, bankovníctví, vojenství, spotřební elektroniky, turistiky atd. Automatická identifikace osob s využitím biometrických ukazatelů patří mezi velmi atraktivní vědní obory již po dlouhou dobu. První novodobé praktické aplikace lze najít již na začátku 70. let (Richard 1971; Fioretti et al. 1972). Využití biometriky v praxi je dnes rozsáhlé a trh s těmito technologiemi roste každým rokem. Obecně neexistuje nějaké omezení při využívání biometrických ukazatelů nebo možnost kombinace více biometrických příznaků a metod. Následující odstavce řeší především biometrickou autentifikaci na základě kontury ruky (Jain & Duta 1999; Faundez-Zanuy et al. 2014). Snímání ruky může být provedeno buď tak, že ruka se opírá o podložku (Sanchez-Reillo et al. 2000; Yörük et al. 2006a; Jain et al. 1999) nebo je volně vložena pod kamerou (Hassanat et al. 2015; Kanhangad et al. 2009; Barra et al. 2019). Další možností je využití opěrných kolíček pro prsty (Sanchez-Reillo et al. 2000; Jain & Duta 1999), které ale deformují konturu ruky, a proto jej někteří autoři nevyužívají vůbec viz např. (Cortes et al. 2009; Gross et al. 2007; Wong & Shi 2002; Xiong et al. 2005). Detailní přehledovou studii lze nalézt

v (Duta 2009; Bharathi & Sudhakar 2011; Bakshe & Patil 2014). Úlohu lze řešit jako jednodimenzionální (Samiya et al. 2012) nebo ve 2D prostoru (Santos-Sierra et al. 2011; Yörük et al. 2006a,b), ve 3D (Michael et al. 2010; Kanhangad et al. 2009). Nevýhodou snímkování ruky, která neleží na podkladu, je nejednoznačná poloha celé ruky v prostoru a samozřejmě i prstů, protože se neopírají o nějaký adjustážní přípravek. Dlaň tak může být do jisté míry sevřená. To mění geometrii celé ruky a samozřejmě i konturu jednotlivých prstů. Výhodou je „hygienická“ čistota, protože ruka se nedotýká podkladu. Obecně je k dispozici značné množství různých metod. Ve všech případech presentovaných prací je rozumná přesnost dána hranicemi cca 87-99 procent v časovém intervalu cca 6-12 měsíců – viz také (Duta 2009). Přehled publikací z posledních dvou dekad je možné nalézt v (Duta 2009; Bakshe & Patil 2014; Pavešic et al. 2004).

Jako každá biometrická metoda má identifikace s využitím kontury ruky svoje výhody, nedostatky (Kukula & Elliott 2006) a samozřejmě i perspektivu budoucího využití, která jde ruku v ruce s technickým pokrokem. Právě autentifikace osob s využitím kontury ruky je do značné míry závislá na technickém pokroku. Např. nalezení správné pozice kloubů, které se nacházejí mezi zápěstní kostí (splint bone) a prvním článkem každého prstu je poněkud problematická (Kanhangad et al. 2009; Michael et al. 2010; Santos-Sierra et al. 2011) a bez speciálního vybavení jako je např. rentgen a rentgenový snímek ruky obtížně řešitelná úloha. Obvykle se řeší slepým odhadem pozice (Yörük et al. 2006a; Amayeh et al. 2007; Kanhangad et al. 2009). Další možností je využití speciální přípravku pro ustavení pozice ruky (Jain et al. 1999; Park & Kim 2013; Covavisaruch et al. 2003). Adjustážní přípravek může být i sklopný (Moravec & Hub 2016), ale to vyžaduje plnou spolupráci skenované osoby.

V této kapitole jsou využity informace z několika různých oblastí:

- 1) Algoritmus ICP (Besl & McKay 1992; Chen & Medioni 1991,1992), který spadá do oblasti počítačové grafiky, umožňuje zarovnání dvou mraků bodů a patří mezi nejpoužívanější metody v dané vědní oblasti.
- 2) Evoluční algoritmus (EA) označovaný jako EPSDE (Mallipeddi & Suganthan 2010) a několik dalších efektivních EA jako prostředek pro řešení až 9-ti dimenzionální optimalizační úlohy.
- 3) Poznatky z oblasti autentifikace osob s využitím kontury ruky (Sanchez-Reillo et al. 2000; Yörük et al. 2006a,b; Gross et al. 2007) a obecný přehled v (Duta 2009; Duta et al. 1999; Bakshe & Patil 2014).
- 4) A také některé poznatky z oblasti výpočtu kontury obrazu (Pavlidis 1982; Parker 2010) a počítačového zpracování obrazu obecně (Sural et al. 2002).

Pro účely experimentů byly použity celkem 3 různé databáze:

- 1) Technocampus Hand Image Database (THID) (Faundez-Zanuy et al. 2014; Font-Aragones et al. 2013; SPLab 2020).
- 2) GPDS (Ferrer et al. 2007; Travesio et al. 2004; GPDS 2020) The Grupo de Procesado Digital de la señal, GPDS (Digital Signal Processing Group) with DPDS (División de Procesado Digital de la Señal) from Instituto para el Desarrollo Tecnológico y la Innovación en Comunicaciones IDeTIC, University of Las Palmas de Gran Canaria, Spain.
- 3) Hand Geometry Data Base (HGDB) (Klonowski et al. 2018) Katedra Informatyki, WPPT, Politechnika Wroclawska.

U databáze THID jsou využity pouze obrazy pořízené ve viditelné části spektra.

7.1.2 Související publikace

Biometrická identifikace osob nepatří mezi jednoduché vědní obory. Biometrických ukazatelů má každá osoba značné množství a velká část z nich se mění v průběhu života. Některé z nich je možné i ztratit nebo poškodit. Jejich zpracování je výpočetně i časově značně náročné. V praxi se při identifikaci osob často setkáváme i s kombinací různých biometrických ukazatelů a s nimi pak individuálně koresponduje i úspěšnost jejich využití při identifikaci.

7.1.2.1 Vybrané metody autentifikace osob s využitím kontury ruky

Biometrické údaje člověka mají jednu nespornou výhodu. Obtížně se zapomínají, a ještě hůře se falšují, pokud je testovaná osoba fyzicky k dispozici. Toho lze využít tak, že se při autentifikaci kombinuje více příznaků např. kontura ruky, otisk prstu (fingerprint), otisk dlaně (palmprint) (Nesrine 2017; Liliana & Utaminingsih 2012; Morales et al. 2010), kožní záhyby na prstech (Kumar & Ravikanth 2009), krevní řečiště ruky (Kang & Park 2010; Park & Kim 2013), duhovka oka (iris) (Wildes 1997; Liu-Jimenez et al. 2011; Bhaganagare & Harale 2017), tvar ucha (Bhanu 2007; Abaza et al. 2013), tvar obličeje (Blanz & Vetter 2003) a k tomu je možné vzít v úvahu i klasický pin-code. Takové systémy jsou velmi složité, drahé, vyžadují dostatečný výpočetní výkon a tým programátorů na realizaci a udržování, a proto je možné setkat se s nimi především u specializovaných firem (Gemalto 2020). Jsou chráněny patenty a tvoří cenné firemní know-how. Příkladem využití může být identifikace pracovníků ve firmách podle otisku dlaně a/nebo kontury ruky (Allegion 2020) nebo fotografie obličeje (Convision 2020). Jako dalším příkladem je identifikace osob na letištích s využitím fotografie ucha (Charette 2010; Abaza et al. 2013). Metod založených na biometrice je značné množství a nové stále vznikají. Konturu ruky a otisk struktury dlaně použili (Ribarič et al. 2003; Charfi 2017; Kumar et al. 2003; Angadi & Hatture 2016). Kombinace ukazatelů zajišťuje pochopitelně vyšší bezpečnost, ale je náročnější na zpracování. Přehled úspěšnosti metod lze nalézt např. v (Villegas et al. 2011; Duta 2009; Bakshe & Patil 2014; Bača et al. 2012). Existují i metody, které umí s vysokou úspěšností odvodit pohlaví na základě kontury ruky (Amayeh et al. 2008). Pokročilé metody zpracování počítačové grafiky umožňují rychlou extrakci hrubé kontury ruky, kterou je pak možné používat pro rozpoznávání gest např. (Nguyen et al. 2013). Pro autentifikaci osob je možné využívat jak celou konturu ruky (Gross et al. 2007), tak i jen konturu jednotlivých prstů (Ribarič & Fratric 2005) a to s vysokou úspěšností až 99 procent. Za předpokladu, že jsou k dispozici kvalitní fotografie, je možné použít i strukturu kožních záhybů na hřbetě prstů (Woodard & Flynn 2005). (Faundez-Zanuy et al. 2016, 2014) představili algoritmus umožňující identifikaci osob s využitím kontury ruky a neuronových sítí. Základem je obraz dlaně z klasického scanneru. Celkem se experimentů zúčastnilo 50 osob ve věku 23-30 let a celkový počet získaných obrazů byl 500. Neuronová síť označovaná jako „multi-layer perceptron“ (MLP) (Haykin 1999) byla sestavena z 10 vstupních neuronů a vrstvy 40 skrytých neuronů. Autoři tvrdí, že úspěšnost jejich metody je v rozmezí 99-100 procent. Další využití neuronové sítě lze nalézt v (Alani 2014; Samiya et al. 2012). (Samiya et al. 2012) navíc využili poměrně neobvyklý přístup. Na data z Radial Distance Diagramu (RDD) pohlíželi jako na jednodimenzionální signál a zpracovávali jej pomocí rychlé Fourierovy transformace. Úspěšnost navržené metody je podle autorů v rozmezí 94-96 procent. (Yörük et al. 2006a) navrhli metodu, která umožňuje s využitím modifikované Hausdorffovy vzdálenosti (Dubuisson & Jain 1994) měřit podobnost kontury ruky dvou osob. Měření probíhá tak, že jednotlivé části kontury ruky (kontury prstů) jsou extrahovány a zpracovávány samostatně – viz také (Jain & Duta 1999) a později (Michael et al. 2010). Pro klasifikaci je pak využit algoritmus Independent Component Analysis (ICA) (Bartlett et al. 1998). ICA je výpočetní metoda, která umožňuje separaci multivarietního signálu do aditivních subkomponent. Předpokládá se, že ony subkomponenty jsou vzájemně statisticky zcela nezávislé. Pro získání obrazů byl využit scanner, ruka se dotýká podložky. Obrazy v rozlišení 383x526 pixelů, dalším zpracováním se rozlišení upravilo na 200x200. Autoři měli k dispozici databázi 1374 obrazů od 458 osob. 3 obrazy od každé osoby. Skenována byla pravá ruka. Věk osob byl 20-50 let a lidé různých národností. Výsledná úspěšnost klasifikace se pohybovala v rozmezí 98-99 procent. (Barra et al. 2019) publikovali práci

založenou na vlastním mnohaletém výzkumu (Barra et al. 2013, 2014, 2015, 2017). Využívali také poznatky jiných výzkumných týmů např. (Dai & Yuen 2003; Sharma et al. 2015). Cílem jejich práce bylo navrhnout algoritmus, který by umožnil identifikovat osobu na snímku ruky z mobilního zařízení např. telefonu nebo tabletu. Všechna tato zařízení mají k dispozici digitální fotoaparát nebo kameru s dostatečně vysokým rozlišením, jehož využití je pro daný účel ideální. Autoři testovali celkem 7 různých metod. Nejlepší výsledky dosáhli s použitím algoritmu Lineární Diskriminanční Analýzy (Linear Discriminant Analysis) (LDA) (Fisher 1936; Fukunaga 1990). Pro identifikaci osoby využívají sadu příznaků označených jako f_1, \dots, f_{57} a jejich různé kombinace, které popisují geometrii ruky. Kontura ruky samotná využita není. Navrženým algoritmem využívající LDA dosáhli nejlepšího výsledku koeficientem EER=0.52% a hodnoty EER=3.5% pro algoritmus využívající standardizovanou Euklidovskou vzdálenost. (Sharma et al. 2015) publikovali metodu nazvanou multimodální biometrický systém, který pro identifikaci osob využívá konturu ruky a také další geometrická primitiva vypočítaná z kontury ruky jako např. vzdálenosti mezi významnými body na kontuře ruky a uvnitř kontury ruky a také úhly, které svírají přímky proložené významnými body. Autoři použili jednodimenzionální Waveletovu dekompozici na všechny extrahované prvky, tj. vzdálenosti i úhly (orientaci). Vstupem do navrženého algoritmu je snímek z klasického kancelářského scanneru. Snímek je prahován a je získána kontura a centroid celé ruky včetně prstů a je také odhadnuto natočení ruky podle pozice prostředního prstu. V případě, že kontura není spojitá, což se stává v oblasti, kde se prsty napojují na dlaň, je kontura uměle dopočtena a opravena tak, aby byla spojitá. Dále je vypočten bod na zápěstí, který bude sloužit jako referenční bod pro další výpočty. Z kontury ruky je vytvořen Radial Distance Diagram (RDD) a z něj jsou pak extrahovány důležité body na špičkách prstů a také v tzv. údolí mezi prsty. Nejlepší dosažená hodnota s využitím navržené metody je EER=0.31% se snímky z proprietární databáze a EER=0.52% se snímky z databáze IITD. (Luque-Baena et al. 2013) navrhli metodu nazvanou GA-LDA, která využívá Genetický algoritmus (GA) v kombinaci s Lineární Diskriminanční Analýzou (Fisherova faktorová analýza) (Linear Discriminant Analysis) (LDA). Ze snímku ruky extrahovali 403 významných rysů a jako základní metriku zvolili Mahalanobisovu vzdálenost a Euklidovskou vzdálenost. Při porovnání s dalšími navrženými metodami GA v kombinaci s metodou k-nejbližší sousedé (k-nearest neighbours) (kNN) a GA v kombinaci s algoritmem „Metoda Podpůrných Vektorů“ (Support Vector Machines) (SVM) právě GA-LDA poskytovala nejlepší výsledky s přesností 100 procent pro databázi GPDS a podmnožinou 22 a 34 významných rysů. Pro databázi IITD pak metoda poskytovala přesnost EER=4.6%. (Travesio et al 2004) popsali metodu, která využívá neuronové sítě a Support Vector Machines (SVM). Nejlepší výsledek byl dosažen s využitím SVM a to úspěšnost 99.90 procent. Autoři vytvořili vlastní databázi s obrázky o rozměrech 1403x1021pixelů. Porovnávací vzorový obraz (template) je složen z více testovacích obrazů max. 6. Obrazy ruky jsou snímány pomocí kancelářského scanneru. Popis ruky zahrnuje všech 5 prstů. Kromě extrakce kontury ruky je zpracovávána i kontura kožních záhybů dlaně - palmprint. (Travesio et al. 2012) navrhli metodu, kde kontura ruky je popsána jako množina úhlů a s touto množinou je pak vytvořen Markovův řetězec deskriptorů DHMM. V angličtině označováno jako “angular descriptions forming a Discrete Hidden Markov chain descriptor“ (DHMM). Jako klasifikátor je použit algoritmus Support Vector Machine (SVM). Pro účely testování je použita databáze GPDS. Autorům se podařilo dosáhnout až 100 procentní úspěšnosti na databázi o 60 ti osobách a počtem bodů 750 a počtem trénovačích vzorů 5. (Dale et al. 2012) navrhli metodu, která umožňuje na základě říze geometrie kontury ruky a struktury kožních záhybů dlaně (palmprint) provádět identifikaci jedinců. Vstupem je barevný obraz ruky z digitálního fotoaparátu. Fotografie je nejprve převedena na šedotónovou reprezentaci a je odfiltrován šum. Dále obraz převeden na černobílou reprezentaci a je vypočten Radial Distance Diagram a z něj jsou pak extrahovány jednotlivé segmenty kontur prstů. Autoři pro popis ruky využívají jednoduchá geometrická primitiva jako je např. délka prstů, šířka prstů apod. Klasifikátor využívá Euklidovské kritérium. Databáze obsahuje snímky od 100 osob, 8 snímků od každé osoby. Počet geometrických primitiv při klasifikaci je celkem 53 (17 ruka + 36 otisk dlaně). Porovnávané metody Diskrétní Kosinová Transformace (Discrete Cosine Transformation) (DCT) a pak také Diskrétní Fourierova Transformace (Discrete Fourier Transformation) DFT a Diskrétní Waveletova Transformace (Discrete Wavelet Transformation) (DWT). Nejlepší dosažené hodnoty jsou pro metodu DCT a to GAR=99.5%, EER=1.1198%.

(Klonowski et al. 2018) prezentovali metodu nazvanou „Hand Geometry and Crookedness Identification Algorithm“ (HG CIA). Použitá databáze obsahuje 5 až 10 vzorků od 60 osob. Pro popis ruky využívají geometrická primitiva v počtu 54 a 62 jako je např. délka prstů, tloušťka prstů šířka dlaně apod. Tzv. „template“ je vytvořen ze 2 nebo 3 vstupních obrazů. Nejlepší dosažená přesnost je pro template tvořený ze třech obrazů ruky a dosahuje hodnot FAR=0.0%, FRR=1.19%, IR=100% v identifikačním módu a FAR=0.03%, FRR=1.19% ve verifikačním módu. Bez použití váhových koeficientů pak fantastických FAR=1.19%, FRR=1.19%.

7.1.2.2 Vybrané evoluční optimalizátory

Předchůdcem moderních evolučních algoritmů jsou metody jako je např. algoritmus MonteCarlo (MC) (Metropolis & Ulam 1949). Pro řešení složitých optimalizačních problémů dnešní doby jsou takovéto algoritmy špatně využitelné, protože mají z obecného pohledu nízkou účinnost. Vývoj se však od těchto dob nezastavil a pokračoval krok za krokem kupředu (Barricelli 1962; Holland 1962) a díky rozsáhlému výzkumu v 70., 80. a 90. letech (Rechenberg 1973; Goldberg 1987, 1989; Holland 1975; Schwefel 1995; Rechenberg 1994) je dnes možné využívat pokročilé stochastické optimalizátory při řešení složitých úloh v mnoha vědních oborech (Floudas & Pardalos 2009; Pardalos 2020). Pro výpočty při autentifikaci osob s využitím kontury ruky je využito několik EA. Byly vybrány čistě na základě rozsáhlých praktických experimentů (Moravec 2012; Moravec & Pošík 2014a,b; Moravec 2015; Moravec & Hub 2015, 2016; Moravec 2020a,b) a také (Pošík et al. 2012; Pošík & Kubalík 2012). Další užitečné informace je možné nalézt např. v (Floudas & Pardalos 2009; Pardalos 2020; Neri & Tirronen 2010; Yang et al. 2008).

V experimentální části je v sekci porovnání výkonosti využito několik modifikací algoritmu Diferenciální Evoluce (DE) (Price 1996; Price & Storn 1996, 1997; Storn & Price 1997; Storn 1996) a to: algoritmus jDE (Brest et al. 2006), algoritmus EPSDE (Mallipeddi & Suganthan 2010), DESAP (Teo 2005), JADE (Zhang & Sanderson 2009), SaDE (Qin & Suganthan 2005), a také optimalizátor SPDE (Abbass 2002). Právě optimalizátory jDE (Brest et al. 2006) a EPSDE (Mallipeddi & Suganthan 2010) – viz Alg. (7.1).1, podávaly při řešení úlohy klasifikace kontur rukou velmi dobré výsledky. Byl také testován algoritmus CMA (Hansen 2006; Wikipedia 2020b) neboli CMA-Evoluční Strategie (CMA-ES) využívající Adaptaci Kovarianční Matice (Hansen et al. 1995; Hansen 2006; Hansen & Ostermeier 1996, 1997, 2001).

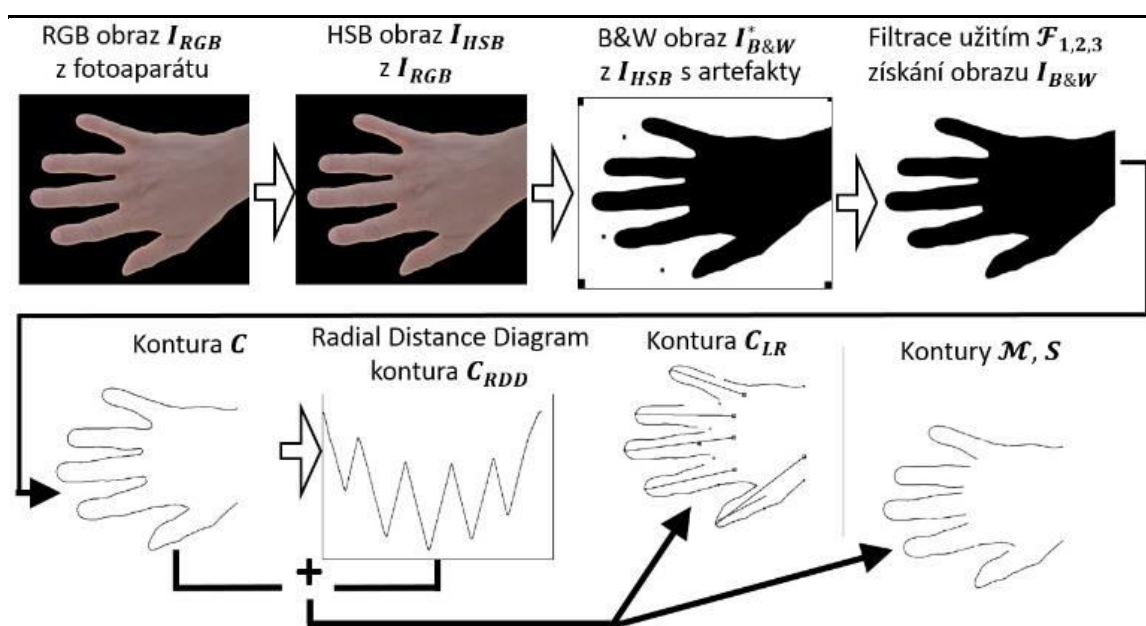
Všechny algoritmy použité v následujících odstavcích této kapitoly optimalizují 7, 8 a 9ti dimenzionální funkci, která je omezená, nespojitá, neseparabilní, silně nelineární a špatně podmíněná. Popíšeme nyní detailněji algoritmus EPSDE, protože právě EPSDE byl zvolen jako nejvhodnější optimalizátor pro řešení dané úlohy a je s ním zpracována největší část experimentálních výsledků. Termíny používané při popisu EPSDE jsou dále využity v popisu algoritmu navrženého pro autentifikaci osob a také v experimentální sekci. Algoritmus EPSDE (Mallipeddi & Suganthan 2010) je založen na podobném principu, jako DE (Storn & Price 1997; Price et al. 1999) nebo jDE (Brest et al. 2006) a vychází také z algoritmu JADE (Zhang & Sanderson 2009). Hlavní devizou EPSDE je však to, že zcela minimalizuje počet pracovních parametrů zadávaných uživatelem z důvodu značné složitosti jejich nastavování u původního algoritmu DE (Eiben et al. 1999; Gamperle et al. 2002; Liu & Lampinen 2002; Smith & Eiben 2009; Smith & Eiben 2010a,b; Das & Suganthan 2010). Jedinými volitelnými parametry jsou tak N_{pop} – počet jedinců populace, G_{en} - počet generací, ukončovací podmínka a hodnota L_p^{EPSDE} . L_p^{EPSDE} udává dobu v generacích, po jakou od počátku evoluce nebude aktualizována hodnota cr_{mean} , která je využita při operaci křížení. Na rozdíl od jDE má EPSDE navíc několik interních pracovních parametrů. Jsou to vektory cr , cr_{mean} a matice CR_{mem} . Tyto interní parametry ovlivňují operátor křížení. Viz detailní popis v kapitole ‘Evoluční algoritmy’.

7.1.2.3 Algoritmus ICP

Za účelem zarovnání dvou kontur ruky byl zvolen algoritmus ICP (Besl & McKay 1992), který vznikl modifikací algoritmu (Horn 1987). ICP (nativeICP) je starší, ale hojně využívaný algoritmus, jehož výsledky jsou ověřené v mnoha pracích. Existuje bezpočet modifikací algoritmu ICP např. (Lu & Milios 1994; Lu & Milios 1997a,b; Bergevin et al. 1996). Nevýhodou těchto algoritmů je, že se u nich nepodařilo prokázat schopnost konvergence jako u originálního ICP. I když tedy vykazují určitá zrychlení původního algoritmu ICP, existuje zde možnost, že proces zarovnání množin bodů bude divergovat a stane se tak nepoužitelný. Původní algoritmus ICP představuje lety ověřenou technologii, která se pro svoje výhody stále častěji používá např. v citlivých oblastech lékařství (Kolekar & Kumar 2017; Farnia et al. 2014). Analýza konvergence algoritmu ICP je uvedena v (Besl & McKay 1992; Pottmann et al. 2006; Maier-Hein et al. 2012). Obecně se má za to, a tyto práce to potvrzují, že na 'přiměřeně malé' oblasti (vzdálenosti dvou mraků bodů) existuje jen jeden extrém (Pottmann et al. 2006; Maier-Hein et al. 2012). Jsou uvažována jen sensorická data bez šumu. Původní algoritmus ICP byl navržen bez schopnosti pracovat s měřítkem objektů. Postupným vývoje byla tato nevýhoda eliminována (Zhu et al. 2017). ICP od autorů (Besl & McKay 1992) pracoval jen s množinami bodů. Tento typ algoritmů je označován také jako Point-to-point (Segal et al. 2009). Je samozřejmě možné zarovnávat i množinu bodů vůči rovině Point-to-plane (Low 2004; Park & Subbarao 2003) nebo úsečce/přímce Point-to-line (Censi 2008). Algoritmus ICP je možné použít jak pro zpracování množin bodů ve 2D (Lu & Milios 1997a,b; Pfister et al. 2002; Minguez et al. 2006; Segal et al. 2009), tak i ve 3D (Chen & Medioni 1991; Amor et al. 2005; Tong et al. 2009). Existují i modifikace ICP využívající doplňková heuristická pravidla a krátkou přehledovou studii je možné nalézt v (Jost & Hügli 2002). Další přehledová studie 2D a 3D variant ICP algoritmu využívaných v oblastech zpracování obrazu a robotiky je k dispozici v (Pomerlau et al. 2013) a detailnější informace o možnostech algoritmu ICP v oblastech zpracování 2D a 3D digitalizovaného obrazu je možné nalézt v (Rusinkiewicz & Levoy 2001; Rodrigues et al. 2002).

7.1.3 Popis navržené metody

Presentovaný systém pro identifikaci osob podle kontury lidské ruky náleží do skupiny bezkontaktních systémů. Ruka se nemusí nutně při snímkování dotýkat podkladu, i když to samozřejmě je možné. Algoritmus neumí pracovat s měřítkem, a proto se předpokládá, že posuv v ose Y je minimální resp. je omezen vhodnými prostředky např. skleněnými deskami. Ruka je vložena do snímací komory nebo do snímkovací oblasti v definovaném směru a úhlu, hřbetem otočeným ke kameře a s plně roztaženými prsty. Osa ruky svírá s osou X co možná nejmenší úhel – viz Obr. (7.1).1 a (7.1).3. Adjustážní přípravek jako např. v (Jain et al. 1999) není použit. Ruka otočená dlaní ke kameře by samozřejmě poskytovala stejný výsledek. Pozice ruky může být proměnlivá v rovině XY v určitém malém rozsahu volného prostoru měřicí komory a samozřejmě s ohledem na zorné pole a zaostření fotoaparátu podobně jako v (Kanhagad et al. 2009). Pozadí scény je černé, matné.



Obrázek (7.1).1 Postupný proces zpracování obrazu z digitálního fotoaparátu od okamžiku, kdy je pořízen snímek až po finální konturu \mathcal{M}, \mathcal{S} vhodnou pro výpočet s využitím evolučního algoritmu a algoritmu ICP.

RGB obraz získaný z kamery je převeden na černobílou reprezentaci transformací z RGB do HSB modelu a pak do B&W obrazu – viz Obr. (7.1).1. Základní schéma je identické, jako je např. v (Travesio et al. 2004, 2012), ale počet „templates“ neboli obrazů, ze kterých je tvořen model \mathcal{M} je roven jedné. Převod na B&W reprezentaci probíhá tak, že homogenní černá barva pozadí je považována za obraz a z tohoto obrazu je odfiltrováno popředí neboli oblast, kterou na snímku zabírá ruka. Odfiltrovaná část pak tvoří černobílý obraz vložené ruky. Poté je vyhledána kontura ruky \mathcal{C} a z ní vytvořen Radial Distance Diagram (RDD) – viz Obr. (7.1).4C,D. Kontura \mathcal{C} je s využitím RDD rozsegmentována tak, aby byly získány samostatné kontury prstů a částí dlaně – viz Obr. (7.1).3. Výsledkem těchto operací je 8 shluků bodů – reprezentujících kontury jednotlivých částí kontury ruky – 5 prstů a 3 části kontury dlaně až po zápěstí. Těchto 8 shluků tvoří vstupní informaci pro evoluční klasifikátor, ve kterém se porovnává referenční kontura s konturou ruky osoby, jejíž identita je prokazována. Dalším krokem je odhad pozice kloubů prstů s využitím algoritmu lineární regrese (LR-LSQ, metoda nejmenších čtverců). Jsou vyhledány klouby všech pěti prstů. Pro palec jsou vyhledány (odhadem) dva ze tří kloubů. Referenční kontury jsou označovány jako model \mathcal{M} a jsou uloženy v databázi ve formě 8-mi shluků bodů + několik dalších podpurných informací. Pro vytvoření kontury \mathcal{M} je použit jen jeden vhodný snímek ruky. Netvoří se statistický soubor dat z více snímků. Vhodný snímek zvolí operátor. Počet bodů kontury je samozřejmě pro každou osobu různý. Každá osoba má v databázi uloženou jednu referenční konturu.

Modelová kontura reprezentuje zcela ideální a nijak nedeformovaný obraz kontury ruky jedné osoby. Znamená to, že scanovaná osoba plně spolupracuje při vytváření databáze modelů kontur. Tato modelová kontura je při klasifikaci plně statická. Pozice a natočení prstů ani pozice a natočení celé modelové kontury se při procesu klasifikace nemění. Vůči modelu \mathcal{M} je v klasifikátoru porovnávána kontura označená jako vzorek – sample \mathcal{S} . Tento vzorek je získán stejným postupem jako model \mathcal{M} a to pokaždé, když je nutné ověřit identitu osoby. Kontura \mathcal{S} je navíc pro účely klasifikace vhodně ořezána tak, aby nepotřebné části nezasahovaly do procesu klasifikace. Při procesu klasifikace je pozice a natočení vzorku \mathcal{S} vůči \mathcal{M} měněna a stejně tak dochází i k změně natočení jednotlivých prstů vzorku \mathcal{S} a to nezávisle, ale v rámci přirozených fyziologických omezení. Výsledkem klasifikace je koeficient podobnosti \mathcal{M} a \mathcal{S} s ohledem na použitou metriku. V případě, že je porovnáván model \mathcal{M} vůči stejnému modelu \mathcal{M} je tento koeficient roven vždy nule. Pokud je \mathcal{M} osoby porovnáván vůči \mathcal{S} stejné osoby, pak se koeficient v ideálním případě blíží nule. Prakticky však nulové hodnoty nedosahuje díky odlišné poloze prstů a celé ruky a mnoha dalším faktorům, především díky nasvětlení scény.

Popišme tedy nyní detailněji jednotlivé kroky navrženého algoritmu. Některé kroky se budou vztahovat jen na databázi THID. Algoritmus budeme označovat jako **eaICP**.

7.1.3.1 Obraz z kamery, transformace RGB obrazu na $B\&W$ obraz a filtrace $B\&W$ obrazu

RGB obraz I_{RGB} z kamery (pravá ruka, hřbetem nahoru, palec směřuje dolů, prsty doleva, jen obrazy z viditelné části spektra, THID databáze) – viz Obr. (7.1).1 a (7.1).3 o rozlišení $I_w \times I_h$. U obrazu je nejprve změněno rozlišení z 640x480 na 1280x960 pixelů s využitím bikubické transformace (Keys 1981; Wikipedia 2020c). V experimentální části je využito různé rozlišení obrazů. Jednak je zachováno původní rozlišení obrazů v databázi 640x480 a pak jsou také využity obrazy s modifikovaným rozlišením 1280x960. Účelem těchto modifikací je získání výrazně lepších výsledků. Obraz je dále převeden na HSB reprezentaci (Hue, Saturation, Brightness) s využitím transformace T_{HSB} podle (Stockman & Shapiro 2001):

$$I_{RGB} = \begin{bmatrix} P_{0,0}^{RGB} & \dots & P_{x,0}^{RGB} \\ \dots & \dots & \dots \\ P_{0,y}^{RGB} & \dots & P_{x,y}^{RGB} \end{bmatrix} \xrightarrow{T_{HSB}} I_{HSB} = \begin{bmatrix} P_{0,0}^{HSB} & \dots & P_{x,0}^{HSB} \\ \dots & \dots & \dots \\ P_{0,y}^{HSB} & \dots & P_{x,y}^{HSB} \end{bmatrix} \quad (7.1.1)$$

$$x, y \in \mathbb{N}^0; x \in \langle 0, I_w - 1 \rangle, y \in \langle 0, I_h - 1 \rangle, I_w, I_h \in \mathbb{N}^0$$

kde $P_{x,y}^{RGB}$ značí pixel na souřadnicích (x, y) . Počátek souřadného systému je u počítačů vlevo nahoře. Horizontální osa je $+X$ (zleva doprava), vertikální osa je $+Y$ (zhora dolů) – viz Obr. (7.1).3. Získaný obraz I_{HSB} je v dalším kroku s využitím transformace $I_{HSB} \xrightarrow{T_{B\&W}} I_{B\&W}^*$ (Sural et al. 2002) převeden na černobílou reprezentaci:

$$I_{HSB} \xrightarrow{T_{B\&W}} I_{B\&W}^*, I_{B\&W}^* = \begin{bmatrix} P_{0,0}^{B\&W} & \dots & P_{x,0}^{B\&W} \\ \dots & \dots & \dots \\ P_{0,y}^{B\&W} & \dots & P_{x,y}^{B\&W} \end{bmatrix} \quad (7.1.2)$$

podle pravidel:

$$P_{x,y}^{B\&W} = \begin{cases} \text{černý} & \text{if } P_{x,y}^{HSB} \cdot H < H_{lim} \wedge P_{x,y}^{HSB} \cdot B < B_{lim} \\ \text{bílý} & \text{jinak} \end{cases} \quad (7.1.3)$$

kde hodnoty $H_{lim} = 315, B_{lim} = 0.39$ jsou voleny čistě na základě praktických experimentů s obrazy z databáze THID a nemají obecný charakter. Dále platí, že $H \in \langle 0.0, 360.0 \rangle, B \in \langle 0.0, 1.0 \rangle$. Parametr

$S \in \langle 0,0,1,0 \rangle$ není využit. Výhodou této transformace je, že hodnoty H_{lim}, B_{lim} mohou zůstat konstantní pro téměř všechny snímky databáze THID. Nevýhodou jsou pak mírně kostrbaté okraje kontur. Byl také testován postup presentovaný v (Klonowski et al. 2018; Gasparini & Schettini 2006) s využitím modelu YCrCb. Byla uvažována pouze složka Y, jiná možnost nebyla. Výhodou tohoto modelu jsou hladké okraje kontur. Nevýhodou pak to, že hodnota Y musí být pro různé snímky v databázi THID měněna v rozmezí $Y \in \langle 60,90 \rangle$. Důvodem je špatné nasvícení snímků. Pro transformaci T_{YCrCb} z RGB do YCrCb byly uvažovány následující vztahy (7.1.4):

$$\begin{aligned} Y &= (0.299 * R) + (0.587 * G) + (0.114 * B) \\ C_B &= 128 - (0.168736 * R) - (0.31264 * G) - (0.5 * B) \\ C_R &= 128 + (0.5 * R) - (0.418688 * G) - (0.081312 * B) \end{aligned} \quad (7.1.4)$$

Pro obraz $I_{B\&W}^*$ samozřejmě platí, že libovolný pixel $P_{x,y}^{B\&W}$ je buď černý $P_{x,y}^{B\&W} = 1$ nebo bílý $P_{x,y}^{B\&W} = 0$ a pro R, G, B platí, že jsou v rozmezí $\langle 0,255 \rangle$. Rozměry obrazu jsou v našem případě $640 \times 480 \times 15bpp$, $I_w = 640, I_h = 480$ a pak také $1280 \times 960 \times 24bpp$, $I_w = 1280, I_h = 960$. Dále je využita filtrace šumu v obraze $I_{B\&W}^*$ s využitím sady digitálních filtrů bitmapového B&W obrazu. Obraz $I_{B\&W}^*$ je filtrován s pomocí sady bitmapových filtračních masek $\mathcal{F} \dots$ neboli $I_{B\&W}^* \xrightarrow{\mathcal{F} \dots} I_{B\&W}$ a tím je získán obraz $I_{B\&W}$ bez nežádoucích artefaktů jako jsou osamocené pixely v obraze nebo jedno pixelové kavity v obraze ruky nebo na jejím okraji. Jsou odstraněny nežádoucí artefakty v rozích obrazu, které vznikají při snímání obrazu kamerou. Odstraněny jsou tak, že pixely ve všech čtyřech rozích jsou obarveny na bílo. Např. pro levý horní roh obrazu jsou obarveny pixely $P_{x,y}^{B\&W}$, $x \in \langle 0,79 \rangle \wedge y \in \langle 0,79 \rangle$. Podobně pro další zbývající rohy obrazu. Šířka čtvercové oblasti o velikosti 80 pixelů je opět volena čistě na základě praktických experimentů, protože v obrazech databáze THID se v rozích některých obrazů nacházejí nežádoucí artefakty (světlá izolepa). Tím samým způsobem je ošetřen i okraj (obvod) celého snímku o šířce okraje 15 pixelů. Pro obecné využití algoritmu tyto operace nejsou nutné a vztahují se jen na obrazy z databáze THID.

Filtrujeme nejprve s využitím (filtru) \mathcal{F}_1 v celé ploše obrazu $I_{B\&W}^*$. Cílem je odstranění osamocených nežádoucích artefaktů na černém pozadí. Masky (filtry) \mathcal{F}_1 je definována jako množina bodů, čtvercová oblast o dané velikosti $\mathcal{F}_1: w \times h, w = h = 21$, která má okraj o šířce 1 pixelu a tento okraj tvoří množina pixelů dále označovaná jako σ a uvnitř oblasti σ je vnitřní část o velikosti $(w - 2) \times (h - 2)$ pixelů dále označovaná jako γ . Platí, že $\forall P_{x,y}^{B\&W}, P_{x,y}^{B\&W} \in \sigma \wedge P_{x,y}^{B\&W} \in \gamma$. Filtrační maska \mathcal{F}_1 se posouvá pixel po pixelu po celém obraze $I_{B\&W}^*$ v rozsahu souřadnic $(I_w - w - 1, I_h - h - 1)$ a provádí následující operaci, která definuje novou barvu každého pixelu:

$$\gamma = \begin{cases} \text{bílý} & \text{jestliže} & \left(\forall P_{x,y}^{B\&W}, P_{x,y}^{B\&W} \in \sigma, P_{x,y}^{B\&W} = 0 \right) \\ \text{žádná operace} & & \left(\exists P_{x,y}^{B\&W}, P_{x,y}^{B\&W} \in \gamma, P_{x,y}^{B\&W} = 1 \right) \\ & & \text{jinak} \end{cases} \quad (7.1.5)$$

Dalším krokem je filtrace maskou \mathcal{F}_2 . Tato maska provádí podobnou operaci jako maska \mathcal{F}_1 , ale má rozměr $w = h = 3$. Jejím úkolem je obarvit všechny osamocené bílé pixely v oblasti γ , které ve svém 8-mi okolí resp. v oblasti v oblasti σ mají alespoň 6 pixelů černých na hodnotu $P_{x,y}^{B\&W} = 1$. Poslední filtrační maska \mathcal{F}_3 má opačnou funkci než \mathcal{F}_2 . Obarvuje všechny černé pixely v oblasti γ , které ve svém 8-mi okolí resp. v oblasti σ mají alespoň 5 bílých pixelů na hodnotu $P_{x,y}^{B\&W} = 0$. Filtrační masky $\mathcal{F}_{1,2,3}$ mají za úkol připravit obraz $I_{B\&W}^*$ pro algoritmus hledání kontury. Filtrace je časově velmi náročná. Jednotlivé po sobě jdoucí kroky neodstraní všechny artefakty, ale pomohou alespoň částečně upravit obraz $I_{B\&W}^*$ pro další zpracování a samozřejmě vytvořit obraz $I_{B\&W}$. Bez filtrace by následující kroky nebyly funkční. V experimentální sekci byl použit větší počet filtrů a to 32, je však možné použít např. 40 nebo 50 filtrů a získat ještě kvalitnější obraz.

7.1.3.2 Odhad délky ruky, nalezení kontury ruky, sestavení RDD

Dalším krokem je nalezení pixelu, který bude označen P_{HSTART} – viz Obr. (7.1).3 a který náleží ke kontuře ruky a současně má nejmenší x -ovou souřadnici. Ruka osoby, jejíž identifikace se provádí, se vkládá do scanovací oblasti vždy pod stejným úhlem \pm malá tolerance cca 5° a ve stejném směru. Pixel na pozici P_{HSTART} leží u většiny 5ti prstých lidí přibližně na špičce prostředníčku. Pro nalezení tohoto bodu je použit jednoduchý Alg. (7.1).2. Bod P_{HSTART} představuje vstupní informaci pro algoritmus, který odhadne délku prostředníčku. Tato metoda odhadu je založena na algoritmu hledání kontury v černobílém obraze viz Alg. (7.1).4, ale na místo řádku 4 jsou vloženy řádky 1-10 z Alg. (7.1).3. Výsledkem je délka prostředníčku L_{MF} . Díky této hodnotě je proveden odhad délky celé ruky – dlaň + prsty. Odhad délky ruky (prsty+dlaň) je pak uvažován jako:

$$\begin{aligned} \mathcal{M}: L_{HAND} &= L_{MF} \times 2.1. \\ \mathcal{S}: L_{HAND} &= L_{MF} \times 2.0. \end{aligned} \quad (7.1.6)$$

Pro \mathcal{M} je kontura vždy delší, než pro \mathcal{S} z důvodu snažší klasifikace. Koeficienty 2.0 a 2.1 byly zvoleny čistě na základě zkoumání fyziologických vlastností lidské ruky a pak také na skutečnosti, že při klasifikaci není nutné mít k dispozici konturu ruky včetně celého zápěstí. Díky znalosti délky ruky L_{HAND} a x -ové souřadnice bodu P_{HSTART} neboli $P_{HSTART}.x$ je možné velmi snadno určit počáteční bod P_{CSTART} na „budoucí“ kontuře ruky, ze kterého se bude pomocí Alg. (7.1).4 vypočítávat kontura ruky. S využitím bodu P_{CSTART} určíme i x -ovou souřadnici bodu středu zápěstí P_{WC} – viz Obr. (7.1).3. Platí, že $P_{CSTART}.x = P_{HSTART}.x + L_{HAND}$. Bod $P_{CSTART}.y$ se pak určí z obrazu $I_{B\&W}$ tak, že z bodu $(P_{CSTART}.x, I_h - 1)$ je vedena přímka směrem k bodu $(P_{CSTART}.x, 0)$ a pozice, kde je nalezen první černý pixel je označena jako P_{C1} – viz Obr. (7.1).3.

| Algoritmus (7.1).2 Nalezení prvního černého pixelu zleva v obraze $I_{B\&W}$. | Algoritmus (7.1).3 Délka prostředníčku v obraze $I_{B\&W}$. |
|--|--|
| Vstup: $I_{B\&W}$; Výstup: bod P_{HSTART} | Vstup: $I_{B\&W}, P_{ACT}, P_{HSTART}.x, c_{limit}$ Výstup: L_{MF} |
| 1 for $i=0; i < I_w; i=i+1$ | 1 $c = 0; x = 0$ |
| 2 for $j=0; j < I_h; j=j+1$ | 2 if $x < P_{ACT}.x$ then |
| 3 if $P_{ij}^{B\&W} = 1$ | 3 $x = P_{ACT}.x ; c = 0$ |
| 4 $P_{HSTART}.x = i$ | 4 else |
| 5 $P_{HSTART}.y = j$ | 5 $c = c + 1$ |
| 6 exit | 6 endif |
| 7 endif | 7 if $c > c_{limit}$ then |
| 8 endfor | 8 $L_{MF} = x - P_{HSTART}.x$ |
| 9 endfor | 9 exit procedure |
| P_{HSTART} – pozice prvního černého pixelu v obraze $I_{B\&W}$ ve směru první zleva. Takových pixelů může být víc. Uvažuje se první ve smysl daného algoritmu. | 10 endif Řádky 2-10 jsou vloženy místo řádku 4 do Alg. (7.1).4. Řádek 1 je vložen do inicializační sekce např. za ř. 1 Alg. (7.1).4. L_{MF} – odhadnutá délka prostředníčku v pixelech. c_{limit} – zvolená hodnota meze, 20 pixelů v našem případě. |

Algoritmus 4 pro výpočet kontury ruky byl navržen drobnou úpravou algoritmu RadialSweep, který je odvozen od algoritmu Moore-Neighbor Tracing (MNT) (Pavlidis 1982; Parker 2010). Pro získání kontury je použit $I_{B\&W}$ obraz, který je zakreslen na Obr. (7.1).4B. Algoritmus (7.1).4 vyžaduje, aby obraz ruky na $I_{B\&W}$ byl spojitý v 8-mi okolí. Korekci obrazu $I_{B\&W}$ provádí filtry $\mathcal{F}_{1,2,3}$, což je minimální možná sestava digitálních bitmapových filtrů, která umožní další zpracování obrazu. Pro lepší, a především stabilnější výsledky je možné nutné použít např. 31 za sebou jdoucích různých druhů filtrů, přičemž některé se mohou i opakovat a pořadí filtrů je důležité. Algoritmus dále vyžaduje určení počátečního směru prohledávání – viz ř. 2 Alg. (7.1).4. Počáteční směr – viz Obr. (7.1).2, je určen na základě rozmístění černých a bílých pixelů v okolí počátečního pixelu P_{CSTART} . Na Obr. (7.1).2 je počáteční pixel P_{CSTART} zakreslen vždy jako křížek. V okolí toho bodu jsou postupně porovnávány

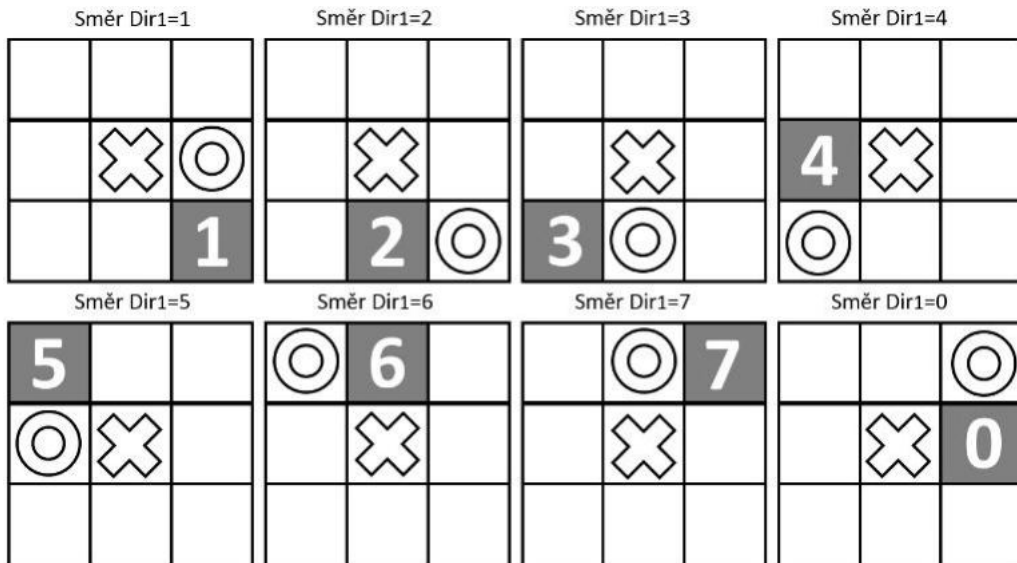
kombinace bílých a černých pixelů v 8-mi okolí podle Obr. (7.1).2. Celkem máme 8 možných kombinací. Podle první nalezené shody, resp. pozice bílého pixelu (kroužek) a současně pozice černého pixelu (černý čtvereček s číslem směru D_{ir1}) je určen směr D_{ir1} viz Alg. (7.1).4. Hodnota D_{ir1} reprezentuje predikci směru v 8-mi okolí, kde by se měl nalézat následující bod kontury vůči bodu P_{CSTART} . Jedná se pouze o drobné urychlení algoritmu RadialSweep o cca 8 procent s využitím „polo-slepé“ predikce, která vychází z tvaru lidské ruky a samozřejmě také posledního směru prohledávání. Alg. (7.1).4 prohledává v rovině XY ve směru hodinových ručiček. Finální kontura C získaná Alg. (7.1).4 je množina černých bodů P_j^C v obraze $I_{B\&W}$ a je definovaná jako:

$$\begin{aligned} C, C \subset I_{B\&W}, \forall P_j^C, P_j^C \in C \wedge P_j^C = 1 \\ C = (P_j^C), j \in \mathbb{N}^+, P_j^C \in I_{B\&W} \end{aligned} \quad (7.1.7)$$

P_j^C samozřejmě koresponduje s odpovídajícím bodem $P_{x,y}^{B\&W}$ v obraze $I_{B\&W}$ na souřadnici (x, y) . Jakmile je nalezena kontura C ruky – viz Obr. (7.1).4C, je sestaven tzv. Radial Distance Diagram (RDD) (7.1.7) resp. křivka R_{DD} (7.1.8) – viz Obr. (7.1).4D a s jeho využitím je pak sestaven popis ruky – viz Obr. (7.1).3. R_{DD} je definovaná jako:

$$R_{DD} = (P_j^{RDD}), P_j^{RDD} = d(P_{RDD}, P_j^C), j \in \mathbb{N}^+ \quad (7.1.8)$$

Křivka R_{DD} je na Obr. (7.1).4D zakreslena do souřadného systému XY , kde X značí pořadí j bodu P_j^{RDD} a Y vzdálenost bodu k P_{RDD} a korespondujícího bodu $P_j^{RDD} = 1$ – viz (7.1.8) a (7.1.9).



Obrázek (7.1).2 Možné nastavení počátečního směru D_{ir1} pro Alg. (7.1).4. (7.1).8 kombinací různých pozic černého a bílého pixelu v 8-mi okolí počátečního pixelu P_{CSTART} , který je zakreslen jako křížek. **Křížek** – pozice prvního/startovacího pixelu P_{CSTART} neboli P_{HSTART} , **Kruh** – pozice bílého pixelu s ohledem na bod P_{CSTART} a P_{HSTART} , **Tmavý čtverec** – pozice černého pixelu s ohledem na P_{CSTART} a P_{HSTART} .

Kontury rukou jsou pro účely klasifikace, která bude popsána v následujících kapitolách rozděleny do dvou skupin \mathcal{M} a \mathcal{S} podle toho, zda kontura je uvažována jako model \mathcal{M} , neboli ideální stav vhodný pro porovnávání a uložený v databázi jako referenční vzor nebo jako datový vzorek \mathcal{S} , který se bude porovnávat s modelem \mathcal{M} s cílem identifikace osoby, která daný datový vzorek poskytla. C_{LR} bude značena kontura, která je získána z R_{DD} úpravou podle Tab. (7.1).1. Tato kontura společně s algoritmem lineární regrese umožní odhad pozic kloubů prstů – viz Tab. (7.1).2.

Algoritmus (7.1).4 Algoritmus pro výpočet kontury (kombinace RadialSweep+Moore)

Vstup: $P_{CSTART} \in I_{B\&W}$, $I_{B\&W}$

Výstup: Kontura ruky C tvořená body $P_{x,y}^{B\&W}$

```

1  $P_{ACT} = P_{CSTART}$  nebo  $P_{ACT} = P_{HSTART}$  v případě výpočtu  $L_{MF}$ 
2 Najdi počáteční směr prohledávání  $D_{ir1}$  podle Obr. (7.1).2
3 while
4   Přidej bod  $P_{ACT}$  do kontury  $C$ 
5   for  $i = (D_{ir1} - 1); i \leq (D_{ir1} + 6); i++$ 
6      $D_{ir2} = i$ 
7     if  $D_{ir2} < 0$  then  $R1=7$ 
8     if  $D_{ir2} > 7$  then  $R1=R1-8$ 
9      $P_{TEST} = L_{TBL}[D_2] + P_{ACT}$ 
10    if  $P_{TEST} \notin I_{B\&W}$  then continue; t.j. GOTO 5 příkaz
11    if  $P_{TEST} = 1$  then t.j. na pozici  $P_{TEST}$  je černý pixel
12     $P_{ACT} = P_{TEST}$ 
13    if  $(D_{ir2} \bmod 2) \neq 0$  then - je  $D_{ir2}$  liché číslo?
14       $D_{ir1} = D_{ir2} - 1$ 
15    else
16       $D_{ir1} = D_{ir2}$ 
17    endif
18  endif
19 endwhile ( $P_{ACT}.x \geq P_{START}.x$ )
20 Přidej  $P_{ACT}$  do kontury  $C$  (poslední bod kontury)

```

P_{START} – startovací černý pixel, první pixel kontury. D_{ir1} – celočíselná hodnota, směr, ve smyslu 8-mi okolí, kterým se bude ubírat sledování kontury; $D_{ir1} \in (0,1,2,3,4,5,6,7)$. D_{ir2} – pomocná proměnná pro dočasné uložení aktuálního směru. P_{ACT} – poslední nalezený černý pixel, který náleží kontuře. P_{TEST} – testovaný bod v obraze $I_{B\&W}$, na pozici P_{TEST} se nalézá buď černý, nebo bílý pixel. Algoritmus předpokládá spojitost v 8-mi okolí, prohledávání probíhá ve 4-okolí. L_{TBL} – tzv. look-up table (viz vpravo), ve které se vyhledá nova pozice P_{TEST} vůči bodu P_{ACT} v 8-mi okolí bodu P_{ACT} .

Hodnoty změny polohy $\Delta x, \Delta y$ v osách X, Y pro pixel look-up tabulce L_{TBL} pro jednotlivé směry prohledávání D_{ir2} :

| | |
|----|---|
| 0: | $L_{TBL}[0].\Delta x = +1; L_{TBL}[0].\Delta y = +0;$ |
| 1: | $L_{TBL}[1].\Delta x = +1; L_{TBL}[1].\Delta y = +1;$ |
| 2: | $L_{TBL}[2].\Delta x = +0; L_{TBL}[2].\Delta y = +1;$ |
| 3: | $L_{TBL}[3].\Delta x = -1; L_{TBL}[3].\Delta y = +1;$ |
| 4: | $L_{TBL}[4].\Delta x = -1; L_{TBL}[4].\Delta y = +0;$ |
| 5: | $L_{TBL}[5].\Delta x = -1; L_{TBL}[5].\Delta y = -1;$ |
| 6: | $L_{TBL}[6].\Delta x = +0; L_{TBL}[6].\Delta y = -1;$ |
| 7: | $L_{TBL}[7].\Delta x = +1; L_{TBL}[7].\Delta y = -1;$ |

Detailní popis struktur ruky využívané při simulaci s EA

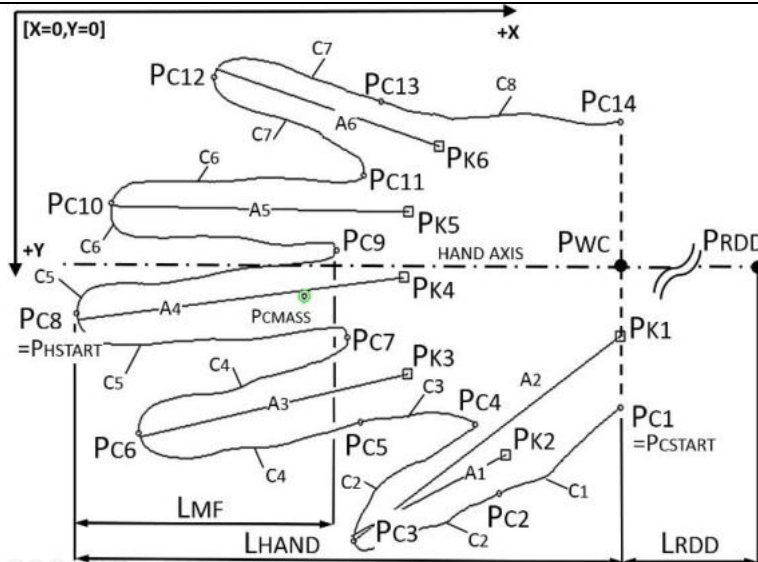


Schéma ruky tak, jak ji vytvoří navržený algoritmus z RGB obrazu. Počátek souřadného systému je vlevo nahoře, X -ová osa je vodorovná, Y -ová svislá (zachováváme souřadný systém počítače). Body $P_{C1} - P_{C14}$ jsou klíčové body na kontuře ruky získané zpracováním RDD podle Alg. (7.1).5 + další pravidla viz Tab. (7.1).1. Body $P_{K1} - P_{K6}$ značí pozice důležitých kloubků prstů, ve kterých bude prováděna simulace natočení prstů v rovině XY . Kontura c_2^- se otáčí kolem kloubu P_{K2} a ten se pak společně s konturou c_2^- otáčí kolem kloubu P_{K1} . Osy prstů A_1, A_3, A_4, A_5, A_6 jsou získány výpočtem pomocí lineární regrese. Osa palce A_2 je získána spojením bodů P_{C3} a P_{K1} . Bod P_{RDD} je bod, ze kterého je vytvářen Radial Distance Diagram. Bod P_{WC} je bod v polovině vzdálenosti mezi body P_{C1} a P_{C14} . Pohyb a natočení celé ruky při simulaci je vztažen k bodu P_{CMASS} .

Obrázek (7.1).3 Kontura ruky s vyznačenými důležitými body a popisem.

7.1.3.3 Popis kontury ruky, Radial Distance Diagram, segmentace kontury

Aby bylo možné konturu ruky dále klasifikovat, je nutné vytvořit její popis. Podle Obr. (7.1).3 vyznačíme na ruce důležité body, úsečky a osu ruky:

1. Body $P_{C1} - P_{C14}$, tyto body leží na kontuře ruky a ohraničují kontury jednotlivých prstů a částí dlaně. Jejich pozice jsou získány z RDD. Osy A_k prstů neprocházejí nutně body P_C .
2. Body $P_{K1} - P_{K6}$, tyto body reprezentují pozice kloubů prstů, které se nalézají vždy mezi záprstní kostí (splint bone) a prvním článkem každého prstu. V těchto bodech (kloubech) bude každý bod pohyblivý (otočný) v rovině XY o určitý specifický úhel – viz (7.1.17), vzhledem ke svému původnímu natočení v kontuře/obrazu S .
3. Bod P_{CMASS} – Těžiště kontury C ruky, která byla získána Alg. (7.1).4.
4. Osy $A_1 - A_6$ – osy prstů. Osy byly získány algoritmem lineární regrese, vůči shlukům bodů c_i^{LR} , který tvoří vždy konturu daného prstu ohraničenou body P_C + drobná korekce viz Tab. (7.1).1. Ke shluku bodů palce se přidává bod P_{C1} , aby se regresní přímka neotočila o 90° .
5. Bod P_{HSTART} je bod kontury C s nejmenší X –ovou souřadnicí – viz Alg. (7.1).2 a Alg. (7.1).4.
6. Bod $P_{CSTART}=P_{C1}$ počáteční bod kontury ruky.
7. Bod P_{RDD} je bod, ze kterého je vytvářen RDD.
8. Bod P_{WC} je bod v polovině vzdálenosti mezi body P_{C1} a P_{C14} . Je to odhad středu zápěstí. Tímto bodem také vede osa ruky rovnoběžně s osou X .
9. Hodnota L_{MF} , která značí přibližný odhad délky prostředního prstu – viz Alg. (7.1).3 a Alg. (7.1).4.
10. Hodnota L_{HAND} značí délku ruky, která se bude využívat při klasifikaci.
11. L_{RDD} je vzdálenost na ose X od bodu P_{WC} k bodu P_{RDD} . Platí, že $d(P_{RDD}, P_{WC}) = 10 \times L_{HAND}$.
12. Dále označíme C_{RDD} jako konturu získanou z Radial Distance Diagramu (RDD), z křivky R_{DD} . C_{RDD} je získána s využitím Alg. (7.1).5 (+ další pravidla v Tab. (7.1).1) a rozdělena na 8 segmentů $c_i^{RDD}, i \in \{1,8\}, c_i^{RDD} = (P_j^{C_{RDD}}), j \in \mathbb{N}^+$ - viz také (7.1.10) a Obr. (7.1).4D,G.
13. C_{LR} budeme značit konturu získanou z kontury C úpravou délky segmentů c_k – viz Tab. (7.1).1 a (7.1.10) s využitím kontury C_{RDD} a dalších heuristických pravidel. C_{LR} je vhodná pro výpočet s využitím algoritmu lineární regrese.
14. Pro kontury M a S platí identický postup – viz Tab. (7.1).1 a (7.1.8). Kontura S je vhodná pro výpočet s využitím EA. Pro S platí totéž co pro M .
15. Kontury M_{co}, M_{ea}, S_{ea} . Tato označení jsou použita v sekci experimentální výsledky. M_{co} značí konturu M , která není pro účely výpočtu s EA ořezána podle hodnot v Tab. (7.1).1. M_{ea} a S_{ea} značí kontury M a S , které jsou pro účely výpočtu s EA je ořezány podle hodnot v Tab. (7.1).1. Účelem ořezání je dosažení výrazně lepší přesnosti výsledku.

Kontura ruky C je nejprve rozsegmentována s využitím C_{RDD} a křivky R_{DD} . R_{DD} se vytváří z bodu P_{RDD} , ze kterého jsou vedeny úsečky k jednotlivým bodům kontury C . Bod P_{RDD} je získán tak, že délka ruky L_{HAND} je zvětšena desetinásobně – viz (7.1.9). Bod P_{RDD} leží na ose ruky – viz Obr. (7.1).3. Pro body $P_{WC}(x, y)$ a $P_{RDD}(x, y)$ tedy platí, že:

$$\begin{aligned}
 & P_{WC}(P_{WC} \cdot x, P_{WC} \cdot y) = & (7.1.9) \\
 & = P_{WC}(P_{HSTART} \cdot x + L_{HAND}, P_{C14} \cdot y + 0.5 \times (P_{C1} \cdot y - P_{C14} \cdot y)) \\
 & P_{RDD}(P_{RDD} \cdot x, P_{RDD} \cdot y) = \\
 & = P_{RDD}(P_{WC} \cdot x + (L_{HAND} \times 10), P_{WC} \cdot y)
 \end{aligned}$$

Bod P_{RDD} daných vlastností byl zvolen proto, že R_{DD} sestavená z bodu P_{WC} dávala velmi špatné

výsledky při segmentaci ruky – při hledání prstů v databázi THID. RDD kopíroval obvod kruhu na špičkách prstů osob s tlustými prsty a bylo tak obtížnější nalézt správnou polohu špičky prstu. Tento postup v navrženém algoritmu funguje dobře, pokud je úhel, který svírá osa palce a osy ruky max. 75 stupňů. Z obecného hlediska je výhodnější umístit bod P_{RDD} poblíž bodu P_{WC} . Jednotlivé body $P_{C1}, P_{C3}, P_{C4}, P_{C6-12}, P_{C14}$ – viz Obr. (7.1).3, byly získány přímo z RDD vyhledáním extrémů v \mathbf{R}_{DD} – viz Obr. (7.1).4D. Algoritmus (7.1).5 zobrazuje způsob nalezení bodu P_{C3} v \mathbf{R}_{DD} . Ostatní body P_C jsou získány analogicky. Bod P_{C2} byl získán tak, že od bodu P_{C3} byl ve směru k bodu P_{C1} odpočítán stejný počet bodů/pixelů kontury \mathbf{C} jaký je mezi body P_{C3} a P_{C4} . Bod P_{C5} byl získán tak, že od bodu P_{C6} byl ve směru k bodu P_{C4} odpočítán stejný počet bodů/pixelů kontury \mathbf{C} jaký je mezi body P_{C6} a P_{C7} . Obdobným způsobem jako byly získány body P_{C2} a P_{C5} , byl získán i bod P_{C13} . Segmentací \mathbf{R}_{DD} jsou získány body, které ohraničují kontury jednotlivých prstů a části kontury dlaně pro další výpočty. Celkem takto získáme 8 shluků bodů pro jednotlivé kontury – viz Obr. (7.1).3 a také (7.1.10), které lze zapsat jako:

$$\begin{aligned} \mathbf{C}_{RDD} &= (c_i^{RDD}), c_i^{RDD}, i \in \langle 1,8 \rangle, c_i^{RDD} = (P_j^{C_{RDD}}), j \in \mathbb{N}^+ \\ \mathbf{C}_{LR} &= (c_i^{LR}), i \in \langle 1,8 \rangle, c_i^{LR} = (P_j^{C_{LR}}), j \in \mathbb{N}^+ \\ \mathbf{M} &= (c_i^{\mathcal{M}}), i \in \langle 1,8 \rangle, c_i^{\mathcal{M}} = (P_j^{\mathcal{M}}), j \in \mathbb{N}^+ \\ \mathbf{S} &= (c_i^{\mathcal{S}}), i \in \langle 1,8 \rangle, c_i^{\mathcal{S}} = (P_j^{\mathcal{S}}), j \in \mathbb{N}^+ \end{aligned} \quad (7.1.10)$$

P_j^{\dots} značí j -tý bod (pixel) i -tého segmentu c_i^{\dots} . Korespondence jednotlivých segmentů c_i^{\dots} kontur \mathbf{C}_{RDD} , \mathbf{C}_{LR} , \mathbf{M} , \mathbf{S} je zaznamenána v Tab. (7.1).1, Obr. (7.1).3 a Obr. (7.1).4G. Každý bod P_j^{\dots} je vztažen k nějaké konkrétní souřadnici (x, y) obrazu $\mathbf{I}_{B\&W}$, která je buď pevná, nebo se mění při výpočtu podle potřeb EA.

Algoritmus (7.1).5 Segmentace RDD (jedna část) – nalezení bodu P_{C3}

Vstup: \mathbf{R}_{DD} – křivka Radial Distance Diagram, I_{VAL} ;
Výstup: P_{C3}

```

1 counter = 0; LMAX = 0
2 for i = 0; i <  $\mathbf{R}_{DD}$ .Count; i++
3   if LMAX <  $\mathbf{R}_{DD}(i)$  – délka i-tého paprsku RDD
4     LMAX =  $\mathbf{R}_{DD}(i)$ 
5     counter = 0;
6   else
7     counter++;
8     if counter > IVAL
9       PC3 =  $\mathbf{R}_{DD}(i - I_{VAL})$ 
10    endif
11  endif
12 endfor

```

\mathbf{R}_{DD} - je to pole hodnot, vzdáleností mezi P_{RDD} a body $P_{x,y}^{B\&W}$, které náleží ke kontuře ruky \mathbf{C} . Segmentací RDD podle navrženého algoritmu získáme konturu \mathbf{C}_{RDD} . I_{VAL} – celé číslo, validační interval, umožňující nalezení extrémů – bodu P_{C3} , hodnota je nastavena na 20 pixelů. \mathbf{R}_{DD} . Count – značí počet bodů kontury.

| Tabulka (7.1).1 Segmenty c_i kontury \mathbf{C} a jejich vztah k bodům P_c | | | | |
|--|--------------------|------------------------|-----------|-------------------------|
| segment | popis | začátek | střed | konec |
| $c_1^{\dots} - P_{K1}$ | část kontury dlaně | $P_{C1} (+0) \{+0\}$ | | $P_{C2} (-0) \{-0\}$ |
| $c_2^{\dots} - P_{K2} + P_{K1}$ | palec [1] | $P_{C2} (+0) \{+0\}$ | P_{C3} | $P_{C4} (-0) \{-15\}$ |
| $c_3^{\dots} - P_{CMASS}$ | část kontury dlaně | $P_{C4} (+0) \{+0\}$ | | $P_{C5} (-0) \{-0\}$ |
| $c_4^{\dots} - P_{K3}$ | ukazováček [25] | $P_{C5} (+0) \{+5\}$ | P_{C6} | $P_{C7} (-15) \{-8\}$ |
| $c_5^{\dots} - P_{K4}$ | prostředníček [30] | $P_{C7} (+7) \{+10\}$ | P_{C8} | $P_{C9} (-7) \{-8\}$ |
| $c_6^{\dots} - P_{K5}$ | prsteníček [58] | $P_{C9} (+17) \{+8\}$ | P_{C10} | $P_{C11} (-42) \{-10\}$ |
| $c_7^{\dots} - P_{K6}$ | malíček [40] | $P_{C11} (+5) \{+8\}$ | P_{C12} | $P_{C13} (-18) \{-4\}$ |
| $c_8^{\dots} - P_{CMASS}$ | část kontury dlaně | $P_{C13} (+0) \{+25\}$ | | $P_{C14} (-0) \{-0\}$ |

c_k^{\dots} značí segment kontury \mathbf{C}_{LR} nebo \mathbf{M} nebo \mathbf{S} . Hodnoty v kulatých závorkách (...) udávají o kolik procent původní délky je segment c_k^{LR} kontury zkrácen od začátku směrem ke středu a od konce segmentu směrem ke středu pro účely lineární regrese vůči kontuře \mathbf{C} – viz Obr. (7.1).4E. Segment c_2^{\dots} je nejprve natočen v kloubu P_{K2} a poté v kloubu P_{K1} společně se segmentem c_1^{\dots} . Nově vzniklá kontura určená pro nalezení os prstů algoritmem lineární regrese je značena \mathbf{C}_{LR} . Vazba segmentů a bodů, kde dochází ke zkrácení a v jakém směru je následující $c_2^{\dots}: P_{C2} \rightarrow P_{C3}, P_{C4} \rightarrow P_{C3}; c_4^{\dots}: P_{C5} \rightarrow P_{C6}, P_{C7} \rightarrow P_{C6}; c_5^{\dots}: P_{C7} \rightarrow P_{C8}, P_{C9} \rightarrow P_{C8}; c_6^{\dots}: P_{C9} \rightarrow P_{C10}, P_{C11} \rightarrow P_{C10}; c_7^{\dots}: P_{C11} \rightarrow P_{C12}, P_{C13} \rightarrow P_{C12}; c_8^{\dots}: P_{C13} \rightarrow P_{C14}$. Hodnoty v hranatých závorkách [...] udávají o kolik procent je aproximační úsečka prstu prodloužena směrem do dlaně. Hodnoty ve složených závorkách {...} mají stejnou funkci jako hodnoty v kulatých závorkách – zkrácení segmentů, ale výsledné zkrácené kontury jsou již použity pro klasifikaci s využitím evolučních algoritmů. Tyto kontury budou dále značeny jako \mathbf{M}, \mathbf{S} . Za názvem segmentu c_k^{\dots} je uvedeno označení kloubu P_K , se kterým je daný segment spojen – viz Obr. (7.1).3.

Při výpočtech pomocí evolučního klasifikátoru **eaICP** jsou uvažovány různé kombinace segmentů $c_k^{\mathbf{M}}, c_k^{\mathbf{S}}$ kontur \mathbf{M} nebo \mathbf{S} a jejich délek. Jedna z testovaných kombinací používá pro konturu \mathbf{M} hodnoty ve složených závorkách rovné nule tj. $\{0\}$ a pro konturu \mathbf{S} jsou hodnoty podle Tab. (7.1).1 – viz sekce experimentálních výsledků. Právě tato kombinace dává skvělé výsledky.

7.1.3.4 Výpočet os prstů a pozic kloubů prstů

Pro výpočet os prstů je využit algoritmus lineární regrese (LR) (Yan & Su 2009; Wikipedia 2020a), kvadratické kritérium – metoda nejmenších čtverců. Regresní funkce \mathfrak{R} je obecně vyjádřena jako:

$$\mathfrak{R} = \sum_{i=0}^n (F(x_i) - y_i)^2 \quad (7.1.11)$$

kde $F(x_i)$ značí aproximační funkci v bodě x_i a y_i zanáčí změřené hodnoty. Pro aproximační funkci $F(x)$ v nějakém bodě x platí, že $F(x) = \sum_{i=0}^n (a_i f_i(x))$, a_i jsou neznámé reálné koeficienty a f_i jsou lineárně nezávislé funkce. Každou množinu bodů c_i^{LR} kontury \mathbf{C}_{LR} aproximujeme individuálně přímkou, takže lze zapsat, že:

$$\mathfrak{R} = \min \sum_{i=0}^n (y_i - (ax_i + b))^2, \quad d\mathfrak{R}/da_k = 0, k = \langle 1, n \rangle \quad (7.1.12)$$

$$a = \frac{n \sum_{i=1}^n y_i x_i - \sum_{i=1}^n x_i \times \sum_{i=1}^n y_i}{n \sum_{i=1}^n x_i x_i - \sum_{i=1}^n x_i \times \sum_{i=1}^n x_i}, \quad b = \frac{\sum_{i=1}^n x_i x_i \times \sum_{i=1}^n y_i - \sum_{i=1}^n x_i \times \sum_{i=1}^n y_i x_i}{n \sum_{i=1}^n x_i x_i - (\sum_{i=1}^n x_i)^2}$$

$$p_1: y = ax + b$$

Rovnice vyrovnávací přímkou má pak tvar $p_1: y = ax + b$. Hodnoty (x_i, y_i) odpovídají jednotlivým bodům $P_j^{C_{LR}}$ ze segmentů c_i^{LR} kontury \mathbf{C}_{LR} . Aproximace přímkou se týká jen segmentů $c_2^{LR}, c_4^{LR}, c_5^{LR}, c_6^{LR}, c_7^{LR}$ – viz Tab. (7.1).1 a Obr. (7.1).3. Z čistě praktických důvodů byly jednotlivé segmenty zkráceny o určitý počet bodů oproti segmentům $c_2^{RDD}, c_4^{RDD}, c_5^{RDD}, c_6^{RDD}, c_7^{RDD}$ – viz Tab. (7.1).1 a Obr. (7.1).4E. Zkrácením získáme konturu \mathbf{C}_{LR} , která je použita jen pro výpočet pozic kloubů.

Díky tomuto zkrácení je jednak možné měnit úhel aproximační přímky vůči ose X a je také oříznuta ta část kontury, která je pro klasifikaci nežádoucí. Je to dobře vidět na Obr. (7.1).4E. Pro další zpracování je aproximační přímka nahrazena aproximační úsečkou \overline{AB} . Aproximační úsečka \overline{AB} leží na aproximační přímce p_1 a její počáteční a koncový bod je zjištěn na základě krajních bodů segmentu c_i^{LR} ve smyslu Euklidovské metriky. Každým bodem $P_j^{CLR}(x, y)$ segmentu c_i^{LR} je vedena přímka p_2 kolmá na vyrovnávací přímku p_1 , čímž získáme průsečíky $R_i, i = \langle 1, n \rangle$ pro jednotlivé přímky p_2 a vyrovnávací přímku p_1 . Pro každý segment c_i^{LR} hledáme vždy takové dva body $A \in R_i, B \in R_i$, které jsou od sebe nejvzdálenější. Tyto body pak tvoří krajní body úsečky \overline{AB} – viz Tab. (7.1).3.

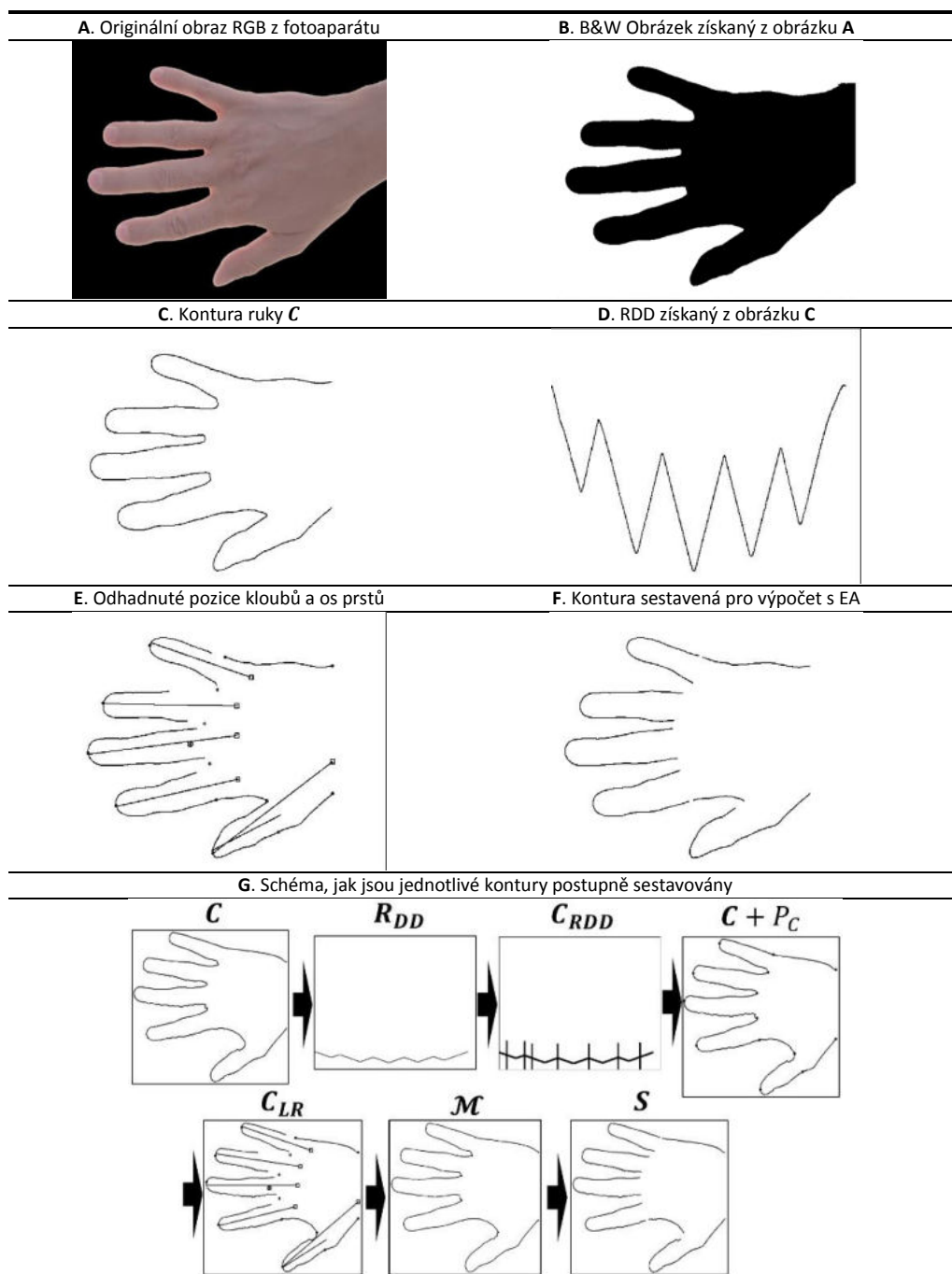
| Tabulka (7.1).2 Pomůcka pro výpočet krajních bodů segmentu c_i^{LR} | |
|--|---|
| $A[x_1, y_1], B[x_2, y_2], P[x_3, y_3], R[x_4, y_4]$ | |
| $p_1: x(y_1 - y_2) + y(x_2 - x_1) + (x_1y_2 - x_2y_1) = 0$ | |
| $\vec{n}_{AB}: [(y_1 - y_2), (x_2 - x_1)], \vec{\mu}_{AB}: [-1 \times (x_2 - x_1), (y_1 - y_2)]$ | |
| $p_2: x(-1 \times (x_2 - x_1)) + y(y_1 - y_2) + C = 0$ | |
| $C = -1 \times (x_3(-1 \times (x_2 - x_1)) + y_3(y_1 - y_2))$ | |
| $\overline{AB}: $ | $\overline{PR}: $ |
| $a_1 = y_1 - y_2$ | $a_2 = -1 \times (x_2 - x_1)$ |
| $b_1 = x_2 - x_1$ | $b_2 = y_1 - y_2$ |
| $c_1 = x_1y_2 - x_2x_1$ | $c_2 = -1 \times (x_3(-1 \times (x_2 - x_1)) + y_3(y_1 - y_2))$ |
| $x_4 = \frac{b_1c_2 - b_2c_1}{a_1b_2 - a_2b_1}, y_4 = \frac{c_1a_2 - c_2a_1}{a_1b_2 - a_2b_1}$ | |
| Průsečík R přímky p_2 jdoucí bodem P , která je současně kolmá na přímku p_1 . | |

Jakmile jsou nalezeny krajní body A, B , je segment prodloužen neboli je změněna pozice bodu B – viz hodnoty v hranatých závorkách v Tab. (7.1).1 a to směrem do dlaně. Bod A leží vždy na špičce prstu. Díky tomuto prodloužení dosáhnou koncové body B jednotlivých aproximačních úseček na pozice kloubů jednotlivých prstů. Pozice kloubů se nachází mezi zápěstí kostí (splint bone) a prvním článkem každého prstu. Jedná se samozřejmě pouze o odhad pozic kloubů jednotlivých prstů. Získání přesné pozice kloubů by znamenalo mít k dispozici např. rentgenový snímek ruky nebo snímek z magnetické rezonance. Na základě experimentů bylo zjištěno, že použití alespoň dvou kloubů palce dává lepší výsledky, než jen použití kloubu palce, který je na zápěstí (je součástí trapézové kosti zápěstí). Je označen jako P_{K1} . Pro výpočet s využitím EA budou použity klouby na pozicích $P_{K1}, P_{K2}, P_{K3}, P_{K4}, P_{K5}, P_{K6}$. Poloha kloubu $P_{K1}(x, y)$ palce je určena následovně:

$$P_{K1}(P_{K1} \cdot x, P_{K1} \cdot y) = \quad (7.1.13)$$

$$= P_{K1} \left(P_{WC} \cdot x, P_{C14} \cdot y + 4.5 \left(\frac{1}{6} (P_{C1} \cdot y - P_{C14} \cdot y) \right) \right)$$

Vhodná poloha kloubu palce P_{K1} byla zjištěna a ověřena čistě experimentálně. Pro drtivou většinu obrazů \mathcal{M}, \mathcal{S} bylo možné dosáhnout ideálního zarovnání i když nebyla známa přesná poloha P_{K1} , která je u každého člověka samozřejmě jiná. Použití algoritmu LR bylo pro obrazy v jednotlivých databázích velmi výhodné. Bohužel na některých obrazech, kde prsty ruky byly zkřivené do tvaru písmene L nebo C např. nevhodným položením na podložku nebo vážnou nemocí, nepracoval algoritmus LR zcela ideálně. Pro nalezení druhé osy palce bylo nutné zahrnout do mraků bodů kontury palce i bod P_{C1} , jinak docházelo k rotaci osy o 90 stupňů díky charakteru algoritmu LR. Výrazně lepších výsledků by bylo možné dosáhnout např. využitím aproximace kontury prstu pomocí b-spline nebo metodami navrženými např. v (Luque-Baena et. al 2013; Yoruk et al. 2006a) apod. Žádná z těchto metod pochopitelně není schopná najít skutečnou polohu kloubů. Kromě metody LR je možné použít i jiné postupy, které jsou velmi efektivní a rychlé, a to např. výpočet pomocí významných geometrických bodů, které kopírují běh kostí ruky a vycházejí ze zápěstí. Stále sice neřeší problém deformovaných prstů ohnutých např. do písmene C vlivem nemocí nebo věkem, ale jedná se jednodušší metodu, než LR a především výrazně jednodušší než parametrické křivky.



A – Originální RGB snímek ruky. B – B&W reprezentace obrazu získaného v A s využitím prahování v HSB barevném modelu (Sural et al. 2002). C – kontura ruky C , získáno výpočtem pomocí Alg. (7.1).4 a z obrazu B. D – čistě pro ilustraci - Radial Distance Diagram získaný z kontury na obr. C tak, že centrální bod ležel v bodě P_{WC} . Pozice palce je vlevo, pozice malíčku vpravo. E – kontura C_{LR} ukazuje části použité pro odhad pozice kloubků prstů s využitím lineární regrese a těžiště kontury z obr. C zobrazené černým vyplněným puntíkem na ose prostředního prstu. Kontura C_{LR} je získána z C_{RDD} vynecháním některých bodů kontury C nevhodných pro výpočet s využitím lineární regrese. Odhad pozice P_K kloubků je zobrazen malými čtverečky na osách prstů – viz také Obr. (7.1).3. Dále jsou zobrazeny osy prstů odhadnuté lineární regrese a klíčové body kontury P_C získané z R_{DD} – viz Obr. (7.1).3. Odhad pozice kloubu palce je proveden tak, že na zápěstí je zvolen vhodný bod a z něj je vedena úsečka na špičku palce. Špička palce je stejně jako u ostatních prstů odhadnuta z C_{RDD} . F – kontura \mathcal{M}_{ea} nebo \mathcal{S}_{ea} , tak jak je použita při výpočtu s využitím evolučního algoritmu. Jedná se pouze o vhodně upravený počet bodů kontury C_{RDD} resp. jsou vynechány části, které se pro výpočet nehodí. G – Jednotlivé kontury $C, R_{DD}, C_{RDD}, C_{LR}, \mathcal{M}, \mathcal{S}$ podle (7.1.7), (7.1.8), (7.1.10) tak jak jsou zpracovávány při procesu klasifikace. R_{DD} i C_{RDD} jsou v reálném tvaru z bodu P_{RDD} . Svislé čáry u C_{RDD} značí nalezené hranice prstů resp. body $P_{C2}, P_{C4}, P_{C5}, P_{C7}, P_{C9}, P_{C11}, P_{C13}$. Podobně jako na D je palec vlevo. Kontura R_{DD} je získána z bodu P_{RDD} .

Obrázek (7.1).4 Jednotlivé obrazy ruky při procesu klasifikace.

7.1.3.5 Úprava kontury ruky pro evoluční algoritmus

Kontura \mathbf{C} získaná pomocí Alg. (7.1).4 není vhodná pro výpočet s využitím EA. Kožní záhyby mezi prsty vytváří na každém obraze z kamery jinou konturu, podle toho, jak identifikovaná osoba ruku položila na podložku nebo do jaké míry jsou prsty ruky natočené vůči zdroji světla. Kontura \mathcal{M} a také \mathbf{S} byla získána úpravou kontury \mathbf{C} s využitím informací z kontury \mathbf{C}_{RDD} . Kontura \mathbf{C}_{RDD} je upravena podle Tab. (7.1).1. Jednotlivé segmenty c_i^{RDD} jsou zkráceny podle hodnot ve složených závorkách v Tab. (7.1).1. Výsledkem je kontura \mathcal{M} nebo \mathbf{S} - viz Obr. (7.1).4F. Všechny segmenty c_i kontury \mathbf{S} jsou pohyblivé kolem korespondujících kloubů a pohybují okolo kloubů P_K . Vazba segmentu na kloub je uvedena v Tab. (7.1).2 v prvním sloupci. Při výpočtech jsou jednotlivé segmenty kontury \mathcal{M} a \mathbf{S} rozdílné co do délky jednotlivých segmentů. Detailní popis je uveden v sekci experimentální výsledky. Pokud mají při výpočtu kontury \mathcal{M} a \mathbf{S} stejnou délku, pak klasifikátor poskytuje velmi špatné výsledky. Jednotlivé segmenty $c_i^{\mathcal{M}}, c_i^{\mathbf{S}}$ jsou tedy odlišné od segmentů c_i^{LR} , které byly použity při lineární regresi. Liší se počtem bodů každého segmentu. Označení $\mathcal{M}_{co}, \mathcal{M}_{ea}, \mathbf{S}_{ea}$ je zavedeno jen pro účely experimentů.

7.1.3.6 Prostor možných řešení účelové funkce

Při výpočtu s využitím evolučních algoritmů je v prvním kroku výpočtu celá kontura ruky \mathbf{S} zarovnána vůči modelu \mathcal{M} tak, že jsou ztotožněna těžiště \mathbf{S} a \mathcal{M} . Model \mathcal{M} má stejné charakteristiky zkrácených segmentů $c_i^{\mathcal{M}}$ jako vzorek \mathbf{S} . Hodnoty uvedené v Tab. (7.1).1 ve složených závorkách mají universální charakter pro všechny \mathbf{S} a \mathcal{M} . Model \mathcal{M} zaujímá při evolučním výpočtu pevnou pozici a natočení v rovině XY , jeho těžiště je umístěno na pozici $P_\Omega(x_\Omega = 0.5I_w, y_\Omega = 0.5I_h)$ resp. do středu zobrazovacího pole kamery. Pozice celé kontury vzorku \mathbf{S} a úhel natočení vůči ose X se v průběhu evolučního procesu mění v závislosti na výpočtu nejlepšího odhadu vzájemné pozice \mathbf{S} a \mathcal{M} - viz Obr. (7.1).5. Úhel natočení jednotlivých prstů se mění také, a to zcela nezávisle pro každý prst, ale prsty jsou pochopitelně svázány s rukou v kloubech P_K . Klouby P_K jsou pak pevně svázány s těžištěm ruky P_{CMASS} . Segmenty $c_3^{\mathbf{S}}$ a $c_8^{\mathbf{S}}$ nejsou svázány s nějakým kloubem a pohybují se pouze s celou rukou - jsou svázány s těžištěm P_{CMASS} . Posun a rotace vzorku \mathbf{S} v rovině XY je vztažen vždy k těžišti P_{CMASS} . Natočení je prováděno v rozsahu reálných čísel s přesností 1×10^{-5} radiánu.

Pro účely výpočtu pomocí evolučního algoritmu je zaváděna tzv. účelová funkce *fitness* (objective function) - viz (7.1.22). Tato funkce ohodnocuje vztah mezi \mathcal{M} a \mathbf{S} s využitím zvolené metriky pomocí reálného čísla - viz (7.1.21) a (7.1.22). *fitness* funkce v našem případě pracuje až v $D_{im} = 9$ -tí dimenzionálním prostoru všech možných řešení Γ :

$$\Gamma: [\gamma_i | \gamma_i \in \mathbb{N}, i \in \langle 0, D_{im} = 9 \rangle] \quad (7.1.4)$$

kde γ_i značí počet možných řešení - viz (7.1.15), (7.1.17) pro i -tou dimenzi. S ohledem na reálnou přesnost při výpočtech cca 1.0×10^{-5} , viz (Betelgeuse 2020; Kimichi 2020; Dawson 2020; SunMicrosystems 2020; Technion 2020) je přibližný počet možných řešení (5 prstů) pro rozlišení obrazu 640x480 pixelů s ohledem na (7.1.17) v prostoru Γ dán vztahem:

$$\begin{aligned} \Gamma &= \prod_{i=0}^{D_{im}} \gamma_i \approx \mathbf{1.38} \times \mathbf{10^{48}} \text{ pro } D_{im} = 9 \\ \gamma_0 &= 19200000, \gamma_1 = 14400000, \\ \gamma_{2,3} &= 80000, \gamma_{4,5,6,7,8} = 60000 \end{aligned} \quad (7.1.15)$$

Indexy u hodnot γ_i korespondují s indexy u jednotlivých genů x_j vektoru \mathbf{X}_i pro jednotlivé dimenze. V prostoru Γ potom použitý evoluční optimalizátor hledá optimální řešení s ohledem na použitou

metriku reprezentovanou *fitness* funkcí. V experimentální části jsou také testovány kontury \mathcal{M} a \mathcal{S} s různým uspořádáním segmentů $c_i^{\mathcal{M}}, c_i^{\mathcal{S}}$, které mají nižší počet dimenzí např. 7 nebo 8. Maximální počet dimenzí a korespondující složitost zůstává zachována. Úloha je tedy 7, 8 nebo 9-ti dimenzionální, podle zvolené konfigurace počtu kloubů.

7.1.3.7 Chromozom, reprezentace fitness funkce, evoluční proces

Jak bylo uvedeno v úvodních odstavcích kapitoly 7.1, optimalizace probíhá s využitím evolučního algoritmu (EA). Při experimentech bylo testováno několik EA. Všechny lze označit jako tzv. populační evoluční optimalizátory neboli výpočet je založen na populaci individuí, kde každé individuum uchovává jedno možné řešení - obvykle vektor reálných čísel. S ohledem na skutečnost, že všechny použité EA mají tuto charakteristiku a všechny EA použité v experimentální části využívají identickou *fitness* funkci a také identický chromosom \mathbf{X}_i , bude další popis proveden obecně pro všechny použité EA.

Po inicializaci populace jedinců \mathbf{P}_{op} je spuštěn evoluční optimalizátor. Při výpočtu je chromosom jedinců \mathbf{X}_i kódován reálnými čísly (7.1.16), aby výpočet byl co nejpřesnější. Geny x_j chromosomu \mathbf{X}_i jsou dány (posun v ose X a Y + rotace kolem X a klouby prstů) podle (7.1.16) jako:

$$x_j, x_j \in \mathbb{R}, j \in \{0, D_{im}\}, D_{im} = \{7, 8, 9\} \quad (7.1.16)$$

Především úhel natočení prstů má největší vliv na výsledek. Hodnoty \mathbf{X}_i, x_j jsou při vytváření populace naplněny náhodnými čísly z rozsahu (7.1.17), včetně pozice ruky resp. kontury \mathcal{S} v rovině XY, která se díky evoluci pochopitelně změní. Rozsah hodnot, kterých může \mathbf{X}_i v jednotlivých dimenzích x_j nabývat označujeme jako D_{om} . Pro každou dimenzi je různý viz (7.1.17). Označme prostor použitelných hodnot \mathcal{H} pro jednotlivé dimenze $D_{om} | D_{om} \in (min, max)$ a hodnoty x_j pak mohou nabývat následujících rozsahů $D_{om}^{j \in \{0,8\}}$.

$$\begin{aligned} D_{om}^{0,1}: x_{0,1} &= [x_{0,1} | x_{0,1} \in \Omega], \\ D_{om}^2: x_2 &= [x_2 | x_2 \in \langle -0.40^{RAD}; +0.40^{RAD} \rangle] \\ D_{om}^3: x_3 &= [x_3 | x_3 \in \langle -0.40^{RAD}; +0.40^{RAD} \rangle], \\ D_{om}^{4-8}: x_{4-8} &= [x_{4-8} | x_{4-8} \in \langle -0.30^{RAD}; +0.30^{RAD} \rangle], \end{aligned} \quad (7.1.17)$$

Povolené hodnoty natočení jednotlivých prstů x_{3-8} a celé ruky x_2 jsou v radiánech a jsou vztaženy k původní poloze daného prstu v kontuře \mathcal{S} jak byla získána Alg. (7.1).4. Viz také Tab. (7.1).2. Geny x_{0-8} chromosomu \mathbf{X}_i jsou při optimalizaci kódovány reálnými čísly. Hodnoty $x_{0,1}$ udávají změnu o $\Delta x, \Delta y$ pozice těžiště $P_{CMASS}^{\mathcal{S}}(x, y)$ kontury \mathcal{M} vůči těžišti $P_{CMASS}^{\mathcal{M}}(x, y)$ kontury \mathcal{M} – viz Obr. (7.1).3 a x_2 udává změnu natočení $\Delta \alpha$ osy kontury ruky vůči ose \mathbf{X} použitého Kartézského souřadného systému s ohledem na bod $P_{CMASS}^{\mathcal{M}}(x, y)$. Kontura \mathcal{S} se natáčí okolo svého těžiště. Platí tedy korespondence mezi $\Delta x, \Delta y, \Delta \alpha$ a geny x_j chromosomu \mathbf{X}_i a to následovně:

$$\Delta x \equiv x_0, \Delta y \equiv x_1, \Delta \alpha \equiv x_2 \quad (7.1.18)$$

Je samozřejmě možné volit větší hodnoty (7.1.17), ale výrazně se tím sníží stabilita použitého evolučního optimalizátoru, který pak nemusí nalézt optimální řešení. Týká se to především překryvů kontur prstů, protože kolize kontur ve 2D prostoru není uvažována. Pokud jsou dodrženy parametry (7.1.17) pak detekce kolizí postrádá jakýkoliv smysl. Celá kontura \mathcal{S} se pohybuje v rovině XY s ohledem na těžiště P_{CMASS} . Pohyb je dán hodnotami genů $x_{0,1}$ chromosomu \mathbf{X}_i v prostoru možných hodnot Ω (souřadnic v rovině XY), jehož středem je bod P_{Ω} .

Lze tedy zapsat, že:

$$P_{\Omega}(x_{\Omega} = 0.5I_w, y_{\Omega} = 0.5I_h) \quad (7.1.19)$$

a prostor Ω je pak dán jako:

$$\Omega = [\langle x_{\Omega} - 0.5I_w, x_{\Omega} + 0.5I_w \rangle, \langle y_{\Omega} - 0.5I_h, y_{\Omega} + 0.5I_h \rangle]. \quad (7.1.20)$$

V Tab. (7.1).3 je možné nalézt vzájemné propojení jednotlivých kontur c_k , kloubů P_K a uspořádání genů x_j chromosomu \mathbf{X}_i .

Tabulka (7.1).3
Vazby mezi dimenzemi x_j chromosomu \mathbf{X}_i a konturami
a zvolená sestavení kontur, kloubů a pravidel (7.1.22) pro experimenty

| gen | Ovlivněná kontura | Kloub | Popis části ruky |
|-------|--|-------------|---|
| x_0 | $c_1^{\cdot\cdot}, c_2^{\cdot\cdot}, c_3^{\cdot\cdot}, c_4^{\cdot\cdot}, c_5^{\cdot\cdot}, c_6^{\cdot\cdot}, c_7^{\cdot\cdot}, c_8^{\cdot\cdot}$ | P_{CMASS} | Hodnota Δx kontury \mathbf{S} v ose X |
| x_1 | $c_1^{\cdot\cdot}, c_2^{\cdot\cdot}, c_3^{\cdot\cdot}, c_4^{\cdot\cdot}, c_5^{\cdot\cdot}, c_6^{\cdot\cdot}, c_7^{\cdot\cdot}, c_8^{\cdot\cdot}$ | P_{CMASS} | Hodnota Δy kontury \mathbf{S} v ose Y |
| x_2 | $c_1^{\cdot\cdot}, c_2^{\cdot\cdot}, c_3^{\cdot\cdot}, c_4^{\cdot\cdot}, c_5^{\cdot\cdot}, c_6^{\cdot\cdot}, c_7^{\cdot\cdot}, c_8^{\cdot\cdot}$ | P_{CMASS} | Natočení $\Delta\alpha$ kontury \mathbf{S} vůči ose X |
| x_3 | $c_1^{\cdot\cdot}, c_2^{\cdot\cdot}$ | P_{K1} | Palec + část kontury ruky |
| x_4 | $c_2^{\cdot\cdot}$ | P_{K2} | Palec, jen kontura $c_2^{\cdot\cdot}$ |
| x_5 | $c_4^{\cdot\cdot}$ | P_{K3} | Ukazováček |
| x_6 | $c_5^{\cdot\cdot}$ | P_{K4} | Prostředníček |
| x_7 | $c_6^{\cdot\cdot}$ | P_{K5} | Prsteníček |
| x_8 | $c_7^{\cdot\cdot}$ | P_{K6} | Malíček |

Pozice a natočení kontury \mathbf{S} v ose X a Y vůči kontuře \mathcal{M} je vztažena vůči bodu P_{CMASS} . $c_i^{\cdot\cdot}$ - značí i -ý segment kontury \mathbf{S} . Natočení $\Delta\alpha$ je vztaženo ose X , protože osa ruky nemusí být nutně rovnoběžná s osou X použitého Kartézského souřadného systému – viz Obr. (7.1).3.

Další částí je popis evolučního procesu a samotné *fitness* funkce. Označme evoluční proces s využitím evolučního optimalizátoru jako \mathfrak{S}_{EA} a zapišme jej jako:

$$\mathcal{E} = \arg \operatorname{opt}_{\mathcal{H} \in \mathbb{R}, \mathcal{S}} \mathfrak{S}_{EA}(\mathcal{M}, \mathcal{S}, \mathcal{H}). \quad (7.1.21)$$

Cílem evolučního procesu (úkolem optimalizátoru) je nalézt takovou hodnotu \mathcal{E} , pro kterou platí, že $\mathcal{E} \cong 0$. To samozřejmě platí pouze v případě, že porovnáváme dvě totožné kontury neboli $\mathcal{M} \equiv \mathcal{S}$. V opačném případě se bude hodnota \mathcal{E} od ideální hodnoty více či méně lišit. Metrika *fitness* funkce byla zvolena na základě praktických experimentů. Využívá jednak Euklidovské vzdálenosti kontur \mathcal{M} a \mathcal{S} a pak také informaci o topologii některých významných bodů $P_{C7}^{\mathcal{M}}, P_{C9}^{\mathcal{M}}, P_{C11}^{\mathcal{M}}$ na kontuře ruky – viz Obr. (7.1).3. Funkce *fitness* je daná podle (7.1.22) následovně:

$$fitness = R_H + \sum_{j=1}^{j=n} R_I \quad (7.1.22)$$

$$fitness \in \mathbb{R}^{+,0}, j \in \mathbb{N}^+$$

n – počet bodů kontury \mathcal{S}

$$R_I = \begin{cases} d(P_j^{\mathcal{S}}, P_{\Omega})^4 & \text{if } \mathcal{L}_1 = true \\ d(P_j^{\mathcal{S}}, P_{min}^{\mathcal{M}}) & \text{if } \mathcal{L}_1 = false \end{cases}, R_I \in \mathbb{R}^{+,0}$$

$$P_{min}^{\mathcal{M}}: d(P_{x,y}^{I_{B\&W}}, P_{min}^{\mathcal{M}}) = \min d(P_{x,y}^{I_{B\&W}}, P_i^{\mathcal{M}}), \forall x, y$$

$$P_{min}^{\mathcal{M}} \in \mathcal{M} \wedge P_{x,y}^{I_{B\&W}} \in I_{B\&W}$$

$$D_{ist}(x, y) = \begin{bmatrix} d(P_{x,y}^{I_{B\&W}}, P_{min}^{\mathcal{M}})_{(0,0)} & \dots & d(P_{x,y}^{I_{B\&W}}, P_{min}^{\mathcal{M}})_{(x,0)} \\ \dots & \dots & \dots \\ d(P_{x,y}^{I_{B\&W}}, P_{min}^{\mathcal{M}})_{(0,y)} & \dots & d(P_{x,y}^{I_{B\&W}}, P_{min}^{\mathcal{M}})_{(x,y)} \end{bmatrix}$$

$$\mathcal{L}_1 = \begin{cases} true, & \text{if } x_3 \notin \langle -0.40^{RAD}; +0.40^{RAD} \rangle \\ true, & \text{if } x_{4-8} \notin \langle -0.30^{RAD}; +0.30^{RAD} \rangle \\ true, & \text{if } x_{0,1} \notin \Omega \\ true, & \text{if } x_2 \notin \langle -0.40^{RAD}; +0.40^{RAD} \rangle \\ false, & \text{jinak} \end{cases}$$

$$R_H = r_{h1} + r_{h2} + r_{h3}, R_H \in \mathbb{R}^{+,0}$$

$$r_{h1} = \begin{cases} M_{ul1} * d(P_{C7}^{\mathcal{M}}, \overline{P_{C7}^{\mathcal{S}}}) & \text{if } \mathcal{L}_2 = true \\ 0 & \text{if } \mathcal{L}_2 = false \end{cases}$$

$$r_{h2} = \begin{cases} M_{ul1} * d(P_{C9}^{\mathcal{M}}, \overline{P_{C9}^{\mathcal{S}}}) & \text{if } \mathcal{L}_3 = true \\ 0 & \text{if } \mathcal{L}_3 = false \end{cases}$$

$$r_{h3} = \begin{cases} M_{ul1} * d(P_{C11}^{\mathcal{M}}, \overline{P_{C11}^{\mathcal{S}}}) & \text{if } \mathcal{L}_4 = true \\ 0 & \text{if } \mathcal{L}_4 = false \end{cases}$$

$$\mathcal{L}_2 = \begin{cases} true & \text{if } d(P_{C7}^{\mathcal{M}}, \overline{P_{C7}^{\mathcal{S}}}) > M_{ul2} * (0.01 * d_{7,9,11}) \\ false & \text{jinak} \end{cases}$$

$$\mathcal{L}_3 = \begin{cases} true & \text{if } d(P_{C9}^{\mathcal{M}}, \overline{P_{C9}^{\mathcal{S}}}) > M_{ul2} * (0.01 * d_{7,9,11}) \\ false & \text{jinak} \end{cases}$$

$$\mathcal{L}_4 = \begin{cases} true & \text{if } d(P_{C11}^{\mathcal{M}}, \overline{P_{C11}^{\mathcal{S}}}) > M_{ul2} * (0.01 * d_{7,9,11}) \\ false & \text{jinak} \end{cases}$$

$$d_{7,9,11} = d(P_{C7}^{\mathcal{M}}, P_{C9}^{\mathcal{M}}) + d(P_{C9}^{\mathcal{M}}, P_{C11}^{\mathcal{M}})$$

$$\overline{P_{C...}^{\mathcal{S}}} = \mathbf{R}(P_{C...}^{\mathcal{S}} - P_{CMASS}^{\mathcal{S}}) + \mathbf{t}, \mathbf{t} \equiv P_{CMASS}^{\mathcal{M}} + [x_0 \quad x_1]^T$$

$d(\dots)$ - Euklidovská metrika

kde R_H reprezentuje hodnotu heuristické účelové funkce, která je daná topologií ruky, R_I značí hodnotu účelové funkce, která je daná geometrií kontury \mathcal{S} resp. body $P_j^{\mathcal{S}}$ a je vypočítávaná podle pravidel algoritmu ICP, n je počet bodů kontury \mathcal{S} , $P_j^{\mathcal{S}}$ je j -tý bod kontury \mathcal{S} , $P_{min}^{\mathcal{M}}$ značí bod kontury \mathcal{M} , který je ve smyslu použité metriky nejbližší k bodu $P_j^{\mathcal{S}}$ kontury \mathcal{S} s ohledem na hodnoty genů chromosomu \mathbf{X}_i , $\overline{P_{C...}^{\mathcal{S}}}$ značí pozice bodů $P_{C7}^{\mathcal{S}}$, $P_{C9}^{\mathcal{S}}$, $P_{C11}^{\mathcal{S}}$ s ohledem na aktuální hodnoty genů $x_{0,1,2}$ chromosomu \mathbf{X}_i . \mathbf{R} je rotační matice daná hodnotou genu x_2 chromosomu \mathbf{X}_i , \mathbf{t} je translační vektor daný hodnotami genů $x_{0,1}$ s ohledem na $P_{CMASS}^{\mathcal{M}}$ a $P_{CMASS}^{\mathcal{S}}$. Body $P_{C7,9,11}^{\mathcal{M}}$ a $P_{C7,9,11}^{\mathcal{S}}$ jsou významné body na kontuře \mathcal{M} a \mathcal{S} – viz Obr. (7.1).3, zvolené na základě rozsáhlých experimentů. Hodnoty $M_{ul1} = 50$ a $M_{ul2} = 20$ byly zvoleny čistě na základě rozsáhlých praktických experimentů. Ideální hodnoty lze samozřejmě získat

výpočet např. v rozsahu hodnot 0-100 pro obě konstanty. Prakticky to znamená přepočítat hodnoty *fitness* pro celou databázi THID celkem 10tis. krát. To je časově neúnosné. Proto byly hodnoty experimentálně odhadnuty. Heuristické pravidlo R_H se skládá ze tří částí r_{h1}, r_{h2}, r_{h3} a udává, kdy dochází k penalizaci *fitness* funkce jako následek neshody zvolených topologických vlastností porovnávaných kontur. Aktivace pravidla je daná rozhodovací podmínkou značenou jako $\mathcal{L}_2, \mathcal{L}_3, \mathcal{L}_4$. V případě, že v průběhu evoluce je bod P_j^S kontury \mathcal{S} umístěn mimo prostor povolených hodnot D_{om} , je zvolena penalizace s využitím pravidla \mathcal{L}_1 , které je dáno jako $d(P_j^S, P_\Omega)^4$, pro každý takový bod P_j^S kontury \mathcal{S} . Výpočet podle (7.1.22) probíhá pro všechny body v jednotlivých zvolených segmentech c_i^S kontury \mathcal{S} . Hodnoty matice \mathbf{D}_{ist} jsou vypočítány pro každou konturu \mathcal{M} a pro každý bod $P_{x,y}^{I_{B\&W}}$ obrazu $\mathbf{I}_{B\&W}$ udávají vzdálenost bodu $P_{x,y}^{I_{B\&W}}$ a nejbližším bodem P_i^M kontury \mathcal{M} ve smyslu Euklidovské metriky. Takový bod je označen jako P_{min}^M . Jedná se o podstatně rychlejší variantu v porovnání s originálním ICP algoritmem; v průměru až 1700x. Struktura algoritmu ICP tímto krokem není nijak dotčena.

Kontura \mathcal{M} je při výpočtu nepohyblivá. Pohybuje se kontura \mathcal{S} s ohledem na těžiště P_{CMASS}^S a dále jsou pohyblivé všechny prsty v kloubech. Matice \mathbf{D}_{ist} je využita pro snížení výpočetní náročnosti – viz následující kapitola. Algoritmus ICP volí pro výpočet postupně všechny body kontury \mathcal{S} a ke každému z nich hledá nejbližší bod v kontuře \mathcal{M} resp. pouze čte korespondující hodnoty v matici \mathbf{D}_{ist} . Výraz $d(P_j^S, P_\Omega)^4$ je volen tak, aby nevhodné řešení bylo opuštěno co možná nejrychleji. Omezující podmínky pro \mathcal{L}_1 a způsob výpočtu *fitness* funkce je zvolen s ohledem na \mathcal{H} a Ω a především na základě praktických experimentů. Omezení daná funkcí \mathcal{L}_1 jsou naprosto nezbytná. Např. testovaný algoritmus CMA-ES by bez nich selhal, protože je primárně navržen pro řešení n-dimenzionálních, neomezených funkcí ve smyslu velikosti prohledávaného prostoru. Počet bodů kontur \mathcal{M} a \mathcal{S} je při výpočtu odlišný, kromě případu kdy $\mathcal{M} \equiv \mathcal{S}$. Dokonce ani dvě kontury stejné osoby na dvou různých snímcích nemají stejný počet bodů. Protože algoritmus ICP hledá vždy takový bod z P_j^M , který je nejbližší k bodu P_j^S ve smyslu použité metriky, různý počet bodů kontur nijak nevádí.

7.1.3.8 Složitost navrženého algoritmu a její minimalizace s ohledem na praktickou realizaci, výpočet fitness

Složitost originálního algoritmu ICP - viz (Besl & McKay 1992; Pottmann et al. 2006) je pro nejhorší případ rovna $O(NM)$ resp. $O(N^2)$, kde N je počet bodů shluku \mathcal{S} a M je počet bodů shluku \mathcal{M} . Výrazné snížení výpočetních nároků přináší již samotné použití evolučních algoritmů. Abychom ještě více snížili obrovské výpočetní nároky, byla zavedena matice $\mathbf{D}_{ist}(x, y)$ – viz (7.1.22), do které jsou předem uloženy hodnoty $d(P_{x,y}^{I_{B\&W}}, P_{min}^M)$ pro každý bod obrazu $\mathbf{I}_{B\&W}$ a pro konturu \mathcal{M} zakreslenou do obrazu $\mathbf{I}_{B\&W}$.

Kontury rukou v databázi THID, které jsou získány zde navrženou metodou, obsahují průměrně cca 3400 pixelů pro rozlišení 1280x960 pixelů a cca 1700 pixelů pro rozlišení 640x480 pixelů. Každá kontura má samozřejmě odlišný počet bodů. Zvolený počet bodů kontury budeme dále značit jako η_M a η_S pro kontury \mathcal{M} a \mathcal{S} . Pro jednu čtvrtinu pixelů tedy platí, že $\eta_M = \eta_S = 850$. Podobný přístup je možné nalézt v (Turk & Levoy 1994; Masuda et al. 1996; Godin et al. 1994). Současně zavedeme značení κ_M, κ_S , které udává kolikátý bod kontury \mathcal{M} a \mathcal{S} je při výpočtu použit. Hodnoty κ_M, κ_S plně korespondují s hodnotami η_M a η_S . Např. kontura \mathcal{S} má 3290 bodů a pro $\kappa_S = 20$ je $\eta_S \cong 164$. Velikost prostoru Γ se tím nijak nezmenší, ale výrazně se tak zrychlí konvergence k optimálnímu řešení. Závislost počtu pixelů kontury na přesnosti výsledku je diskutována v sekci experimentálních výsledků. Bylo testováno postupné snižování počtu bodů η_M a η_S obou kontur \mathcal{M} a \mathcal{S} vzhledem k počtu generací $G_{en} = 400$ např. lineárně podle rovnice $p_a: \kappa_S = [-0.098G_{en} + 40]$ a to se změnou v každé generaci. Tento

postup však nedával dobré výsledky. Jako výrazně lepší řešení se na základě „alchymistických“ experimentů ukázala skoková změna parametrů aproximativně vůči p_a . Např. pro cílovou hodnotu $\kappa_S = 1$ a při rozměru obrazu 1280x960 pixelů je vhodné volit skoky podle následujícího schématu:

$$\begin{aligned}\kappa_M = \kappa_S = 40, G_{en} &= \langle 0,39 \rangle \\ \kappa_M = \kappa_S = 20, G_{en} &= \langle 40,79 \rangle \\ \kappa_M = \kappa_S = 5, G_{en} &= \langle 80,129 \rangle \\ \kappa_M = \kappa_S = 1, G_{en} &= \langle 130,400 \rangle\end{aligned}\tag{7.1.23}$$

Pro úspěšnou konvergenci je nejdůležitější poslední část s parametry $\kappa_S = 1, G_{en} = \langle 130,400 \rangle$. Tento úsek musí být dlouhý alespoň 60 procent z celkového počtu generací G_{en} . Toto pravidlo platí pro všechny optimalizátory odvozené od algoritmu DE. Je tedy využít κ_M -tý bod kontury \mathcal{M} a/nebo κ_S -tý bod kontury \mathcal{S} . V experimentální části je především měněna hodnota κ_S . Hodnota κ_M zůstává zafixována s ohledem na matici \mathbf{D}_{ist} – viz následující odstavec; $\kappa_M = 1$. Počet generací G_{en} je volen čistě na základě výpočetní náročnosti a také s ohledem na to, že při zvyšování počtu generací nad určitou hodnotu již výsledek nijak výrazně nemění – viz záznam konvergence Obr. (7.1).16. Při každé změně η_M, η_S dochází i k resetu *fitness* hodnoty ve vektoru \mathbf{X}_{best} na max. hodnotu pro daný formát čísla (t.j. $\mathbf{X}_{best} \cdot fitness = IEEE754(double.maxvalue)$), jinak by použité EA nefungovaly.

Nyní popíšeme čistě implementační záležitosti, které dále výrazně upravují časovou náročnost a numerickou složitost algoritmu **eaICP**. Základní mechanismus popsany v předchozích odstavcích zůstává zcela zachován.

Algoritmizace estimátoru **eaICP** využívá toho, že celkový počet bodů obrazu $\mathbf{I}_{B\&W}$ je relativně malý a kontury \mathcal{M} jsou neměnné. Aby se výpočet výrazně urychlil, je pro každý bod $P_{x,y}^{B\&W}$ obrazu $\mathbf{I}_{B\&W}$ vypočtena hodnota $d(P_{x,y}^{B\&W}, P_{min}^{\mathcal{M}})$ pro každou konturu \mathcal{M} předem – viz Alg. (7.1).6. Počet bodů obrazu $\mathbf{I}_{B\&W}$ je dán součinem $I_w \times I_h$ neboli $1280 \times 960 = 1228800$. Soubor těchto dat (seznam vzdáleností) zabírá pro každou modelovou konturu \mathcal{M} přibližně 33MB při rozlišení snímku 1280x960 pixelů a výpočet trvá cca 130 sekund (jedno procesorové jádro z procesoru AMD-FX-6300) pro jednu konturu při $\kappa_M = 1$. EPSDE optimalizátor tak při procesu evoluce již fyzicky nepočítá *fitness* podle vzoru ICP a zvolené metriky, ale pouze pro každý bod $P_j^{\mathcal{S}}$ kontury \mathcal{S} , který se pochopitelně může v procesu evoluce nalézat téměř kdekoli v obraze $\mathbf{I}_{B\&W}$ – v rámci (7.1.10), pouze „přečte“ hodnotu vzdálenosti k nejbližšímu bodu kontury \mathcal{M} z předem vypočtených dat z matice $\mathbf{D}_{ist}(x,y)$ – viz Alg (7.1).6. a také řádek 28 Alg. (7.1).1.

Velikost kontury \mathcal{M} tak již nehraje významnější roli, nicméně v experimentální sekci při výpočtech časových nároků je uvažována, a to s velmi zajímavými výsledky. Algoritmus **eaICP** je tak mnohem rychlejší, než kdyby byl ve spojení s ICP použit např. mechanismus k-d-tree (Bentley 1975; Greenspan & Yurick 2003) – viz Obr. (7.1).7, nebo jen originální ICP (Besl & McKay 1992). Výhodou oproti originálnímu algoritmu ICP je také to, že při výpočtu *fitness* funkce již není časově náročná část, kde je nutné vyčíslit vzorec pro *fitness* podle (7.1.22) vůbec zapotřebí. Další popis je v sekci experimentální výsledky. Výpočet *fitness* na ř. 28 Alg. (7.1).1 je proveden podle Alg. (7.1).7. Žádná z výše uvedených úprav neporušuje pravidlo zaručené konvergence k optimu, která pro algoritmus ICP byla prokázána v (Pottmann et al. 2006; Maier-Hein et al. 2012).

Algoritmus (7.1).6 Výpočet hodnot matice D_{ist} pro všechny pixely v obraze $I_{B\&W}$

Vstup: \mathcal{M} , $I_{B\&W}$ obraz o rozměrech $I_w \times I_h$ Výstup: D_{ist} matice o velikosti $I_w \times I_h$

```
1 for  $y = 0; y < I_h; y=y+1$ 
2   for  $x = 0; x < I_w; x=x+1$ 
3     Pro pixel  $P_{x,y}^{B\&W} \in I_{B\&W}$  na souřadnicích  $(x,y)$ 
       najdi takový bod  $P_{min}^{\mathcal{M}} \in \mathcal{M}$ , pro který je
        $d(P_{x,y}^{B\&W}, P_i^{\mathcal{M}})$  minimální ve smyslu fitness
       funkce - viz (7.1.21)
4      $D_{ist}(x,y) = P_{min}^{\mathcal{M}}$  pro bod  $P_{x,y}^{B\&W}$  na pozici  $(x,y)$ 
5   endfor
6 endfor
```

$P_{x,y}^{B\&W}$ -bod z obrazu $I_{B\&W}$ na souřadnicích (x,y) , $P_{min}^{\mathcal{M}}$ -bod kontury \mathcal{M} , který je nejbližší k bodu $P_{x,y}^{B\&W}$, $D_{ist}(x,y)$ -matice jejíž prvky reprezentují hodnoty $P_{min}^{\mathcal{M}}$ pro korespondující pixel $P_{x,y}^{B\&W}$ na souřadnicích (x,y) .

Algoritmus (7.1).7 Výpočet *fitness* podle (7.1.22)

Vstup: \mathcal{S} , \mathbf{X}_i , $D_{ist}(x,y)$, počet dimenzí úlohyVýstup: *fitness*

```
1 Natočení jednotlivých segmentů  $c_i^{\mathcal{S}}$  kontury  $\mathcal{S}$  v kloubech
    $\mathbf{P}_k$  podle hodnot genů  $x_3, x_4, x_5, x_6, x_7, x_8$  chromosomu  $\mathbf{X}_i$ .
   Segment  $c_2^{\mathcal{S}}$  je natočen v kloubu  $\mathbf{P}_{k2}$  podle hodnoty  $x_4$ 
   a poté společně se segmentem  $c_1^{\mathcal{S}}$  v kloubu  $\mathbf{P}_{k1}$  podle
   hodnoty  $x_3$ .
2 Posunutí a natočení celé kontury  $\mathcal{S}$  podle hodnot genů
    $x_0, x_1, x_2$  chromosomu  $\mathbf{X}_i$ .
3 Výpočet fitness jako součet hodnot  $R_H$  a  $R_I$  získaných
   z matice  $D_{ist}(x,y)$  pro aktualizované polohy bodů
   kontury  $\mathcal{S}$  - podle zvoleného uspořádání estimátoru.
```

\mathcal{S} -Kontura ruky identifikované osoby, \mathbf{X}_i - aktuální stav chromosomu v průběhu evoluce, $D_{ist}(x,y)$ -tabulka hodnot *fitness* pro jednotlivé pixely obrazu $I_{B\&W}$ s konturou \mathcal{M} . Bod 2 algoritmu provádí posun kontury \mathcal{S} na souřadnice $[x = 0, y = 0]$, rotaci podle hodnoty x_2 , která je v radiánech a opětovný posun kontury \mathcal{S} s ohledem na těžiště $P_{CMASS}^{\mathcal{S}}$ a $P_{CMASS}^{\mathcal{M}}$ a hodnoty x_0, x_1 . Bod 3 algoritmu je počítán podle (7.1.22). V experimentální části jsou testovány i úlohy s nižším počtem dimenzí, než 9, např. 7 nebo 8. V tomto případě jsou některé geny chromosomu \mathbf{X}_i neaktivní resp. v bodu 1 se pro neaktivní geny neprovádí natočení kontur a korespondující kontury jsou z výpočtu *fitness* v bodě 3 zcela vynechány. To samé platí pro hodnotu R_H , která také není v některých experimentech uvažována – podle zvoleného uspořádání estimátoru.

Složitost navrženého algoritmu **eaICP**, který s využitím evolučního optimalizátoru, zarovná dvě kontury \mathcal{M} a \mathcal{S} je daná součinem:

$$O(G_{en} \times N_{pop} \times \eta_S) \approx O(N) \quad (7.1.24)$$

kde G_{en} značí počet generací evolučního algoritmu nezbytných pro dosažení optimálního řešení, N_{pop} značí počet jedinců \mathbf{X}_i v populaci \mathbf{P}_{op} a η_S uvažujeme jako konstantu po celou dobu evoluce. Průměrný počet pixelů kontury v databázi THID při rozlišení snímku 1280×960 je cca 3400. Nejdelší kontura

měří 4013 pixelů. Hodnoty G_{en}, N_{pop} nemá smysl zvětšovat nad určitou mez – viz Obr. (7.1).16, protože není možné dosáhnout výrazně lepších výsledků. Jedná se tedy o konstanty. Budeme-li uvažovat nejhorší případ \mathcal{S} tj. $\eta_S = 4013$ pixelů a dále $G_{en} = 400, N_{pop} = 40$, pak lze zapsat, že:

$$O(400 \times 40 \times 4013) = 64\,208\,000,00 \approx \mathbf{6.44210^7} \quad (7.1.25)$$

což je celkový a současně i maximální potřebný počet vyčíslení *fitness* funkce resp. počet čtení z tabulky \mathbf{D}_{ist} , potřebných k dosažení optimálního zarovnání dvou shluků bodů \mathcal{M} a \mathcal{S} v 9-ti dimenzionálním prostoru s odhadnutým počtem možných řešení $\mathbf{\Gamma}$ s využitím evolučního optimalizátoru – v našem případě EPSDE. Se vzrůstajícím počtem bodů η_S kontury \mathcal{S} vzrůstá složitost lineárně podle (30). Pokud však budeme definovat určitou max. hodnotu η_S s hledem na max. velikost $\mathbf{I}_{B\&W}$, která vychází z maximální velikosti lidské ruky, což je dáno čistě lidskou fyziologií, lze říci, že složitost algoritmu **eaICP** je konstantní. Uvažujme délku kontury průměrné lidské ruky dospělého člověka přibližně 1100mm (od zápěstí k zápěstí). Při rozlišení 0.1mm se bude max. hodnota η_S pohybovat okolo hodnoty 11000 bodů. Výpočetní nároky by tak byly šestinásobné tj. 1.76×10^8 . Hodnota 0.1mm je spíše akademická a samozřejmě vyžaduje nějakou speciálně upravenou fotografickou komoru s ideálním a plně homogenním osvětlením. Diskutabilní je také časový interval použitelnosti. To je velmi výhodné pro případnou hardwarovou akceleraci s využitím programovatelných logických polí, grafických karet atd. Takovýto mechanismus je pochopitelně možné použít jen díky tomu, že kontury \mathcal{M} jsou neměnné na určitém časovém intervalu. Stejný přístup však lze uplatit i u podobných problémů z oblasti biometrie např. při identifikaci osob využívající konturu ucha apod.

Pro další urychlení výpočtu je možné využít i výkonné grafické karty, resp. kombinace CPU+GPU nebo také PLA jako je tomu např. v oblasti zpracování obrazu viz (Melo et al. 2012; Jeught et al. 2012). Za předpokladu, že takové zařízení je finančně dostupné. Výsledky přesnosti pro skokové snižování počtu bodů kontury a stejně tak i lineární závislost s ohledem na aktuální generaci je dále diskutováno v sekci experimentální výsledky.

7.1.3.9 Algoritmus ICP pro zarovnávání kontur \mathcal{M}, \mathcal{S}

Nyní jen krátce popíšeme algoritmus ICP s ohledem na navržený algoritmus **eaICP**. Při klasifikaci je nutné kontury \mathcal{M}, \mathcal{S} nějakým způsobem porovnat. Pokud nejsou kontury identické, pak mezi nimi vznikají viditelné rozdíly. Je tedy nutné najít takový vektor \mathbf{X}_i , pro který platí, že $\mathcal{E} \cong 0$ – viz (7.1.26). Algoritmus ICP umožňuje zarovnat vůči sobě dva shluky bodů s využitím operací rotace a translace tak, aby platilo (Besl & McKay 1992), že:

$$\mathcal{E} = \arg \min_{\mathbf{R}, \mathbf{t}, \mathcal{S}} \sum_{j=1}^n \|(\mathbf{R}P_j^{\mathcal{S}} + \mathbf{t}) - P_j^{\mathcal{M}}\|_2^2 \quad (7.1.26)$$

kde $\mathbf{R}, \mathbf{R} \in \mathbb{R}$ je rotační matice 3×3 , \mathbf{t} je translační vektor v rovině XY , $P_i^{\mathcal{M}}$ jsou body kontury modelu a $P_j^{\mathcal{S}}$ je kontura vzorku, který je vůči $P_j^{\mathcal{M}}$ porovnáván. Algoritmus ICP vyžaduje takové počáteční podmínky, pro které platí, že obě kontury \mathcal{M} a \mathcal{S} musí být vůči sobě zarovnány (a natočeny) o co možná nejmenší hodnotu vůči optimálnímu zarovnání. Optimální hodnoty pro \mathbf{R} a \mathbf{t} nejsou apriorně známy a proto se pro oba mraky bodů pouze ztotožní jejich těžiště – využije se toho, že ruka je vkládaná vždy stejným směrem a pod přibližně stejným úhlem. Počáteční natočení obou vzorků je dáno tím, že osy \mathcal{M} a \mathcal{S} svírají malý úhel v rozmezí cca $\pm 10^\circ$. Originální algoritmus ICP (Besl & McKay 1992) se skládá ze třech kroků:

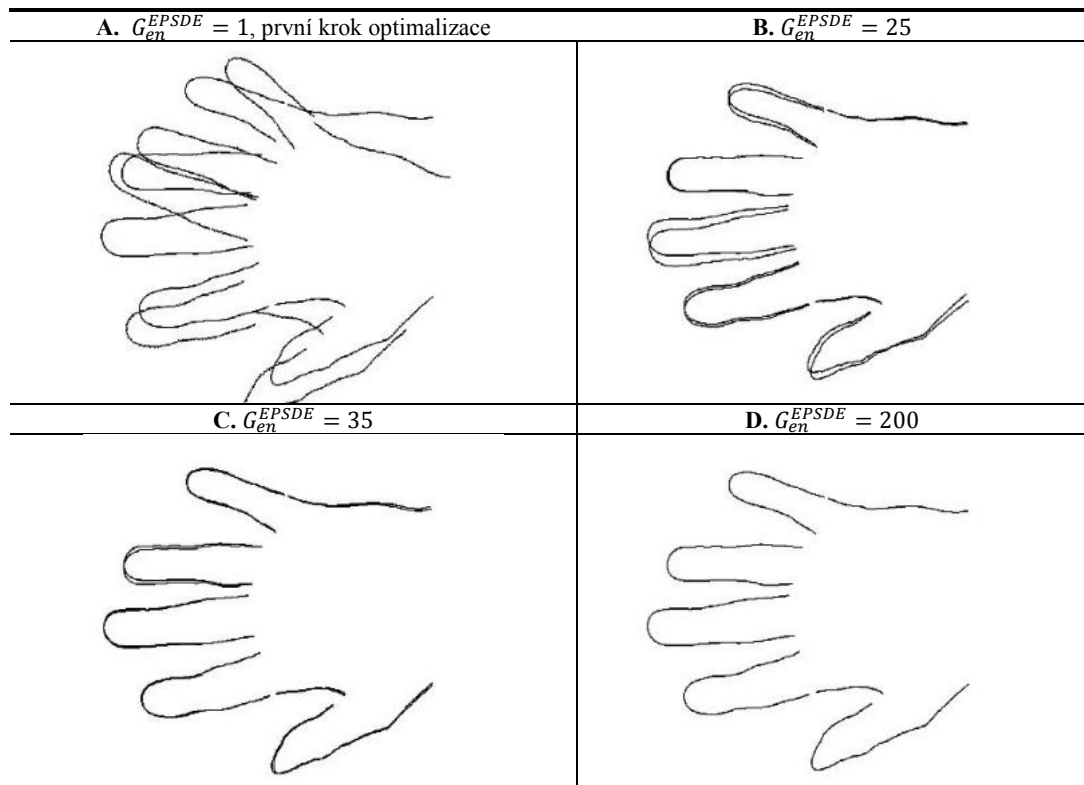
- 1) Nejprve je pro každý bod $P_j^{\mathcal{S}}$ vyhledán nejbližší bod z $P_j^{\mathcal{M}}$ s využitím aktuálních hodnot \mathbf{R} a \mathbf{t} . Tato operace má pro nejhorší případ složitost $O(\eta_M, \eta_S)$.

2) Aktualizace hodnot \mathbf{R} a \mathbf{t} tak, aby platilo, že:

$$(\mathbf{R}, \mathbf{t}) = \min_{\mathbf{R}, \mathbf{t}} \sum_{i=1}^n \|(\mathbf{R}P_j^{\mathcal{S}} + \mathbf{t}) - P_j^{\mathcal{M}}\|_2^2 \quad (7.1.27)$$

3) Pokud hodnota \mathcal{E} je menší, než předem zvolené minimum, algoritmus končí, resp. byly nalezeny optimální hodnoty (\mathbf{R}, \mathbf{t}) .

Opakováním kroků 1,2,3 je dosažen požadovaný výsledek. ICP (Besl & McKay 1992) byl navržen pro zarovnávání shluku bodů ve 3D prostoru za pomoci rotace a translace, měřítko nebylo uvažováno. Algoritmus ICP je pochopitelně možné využít i pro řešení 2D úloh. Pro obě varianty byla prokázána konvergence. Existuje bezpočet modifikací, ale u žádné z nich nepodařilo prokázat schopnost konvergence. Z tohoto důvodu je také využít jen originální ICP a to i za cenu relativně vyšších výpočetních nároků, které jsou však stále v přijatelných mezích. Důležitý je především přesný výsledek.



Kontury \mathcal{M} a \mathcal{S} jsou vůči sobě zarovnané podle těžiště. Hodnoty natočení jednotlivých prstů, posun celé ruky v rovině XY a její natočení vůči ose X je definováno hodnotami genů x_j chromosomu \mathbf{X}_i , které jsou v prvním kroku evoluce náhodné v předem zadaném rozmezí hodnot. Hodnoty x_j nemohou být v prvním kroku evoluce pro některé optimalizátory, jako je např. jDE nulové, jinak by se evoluční proces nerozběhl. Proto je viditelné vychýlení \mathcal{S} vůči \mathcal{M} , které se ale během několika generací zkoriguje. Pracovní parametry pro EPSDE: $\mathbf{P}_{op} = \mathbf{20}$, $L_p = 10$, $G_{en}^{EPSDE} = \mathbf{200}$, $\eta_M = \eta_S = 1861$ resp. $\kappa_M = \kappa_S = 1$ (plný počet bodů kontur při rozlišení 640x480 pixelů). Sub-optimální řešení nalezl EPSDE již v 82. generaci. **B**, **C**, **D** – postupná konvergence ke správnému řešení. V tomto případě jsou shluky \mathcal{M} a \mathcal{S} totožné neboli platí, že $\mathcal{M} \equiv \mathcal{S}$ a tak je $\mathcal{E} = 0$.

Obrázek (7.1).5 A – Evoluční proces a jeho vliv na zarovnání kontur s využitím optimalizátoru EPSDE.

Kontura \mathcal{S} se skládá z 8-mi částí – viz Tab. (7.1).1, (7.1).2. Jednotlivé segmenty $c_i^{\mathcal{S}}$ se natáčejí samostatně v kloubech P_k podle hodnot x_j chromosomu \mathbf{X}_i , ale pro výpočet pomocí ICP se samozřejmě uvažuje celá kontura \mathcal{S} resp. všechny body kontury \mathcal{S} – podle zvoleného uspořádání viz Tab. (7.1).5. Na Obr. (7.1).5 je dobře patrné, že už ve 35. generaci je kontura \mathcal{S} dobře zarovnaná vůči kontuře \mathcal{M} . Tento

stav nastává mezi 20 a 40 generací evolučního procesu. Poté se rychlost konvergence ke správnému řešení zpomalí. Tento jev je shodný u všech použitých EA.

7.1.3.10 Praktická realizace navrženého algoritmu eaICP

Praktická realizace algoritmu **eaICP** je rozvržena do **dvou částí**. V **první části** je kontura ruky klasifikována, je vytvořen její popis a jsou získány kontury \mathcal{M}, \mathcal{S} a jsou vytvořeny matice \mathbf{D}_{ist} pro každou konturu \mathcal{M} v obraze $\mathbf{I}_{B\&W}$. Je zvolena vhodná kombinace jednotlivých segmentů $c_i^{\mathcal{M}}, c_i^{\mathcal{S}}$ a kloubů P_{K1} kontur \mathcal{M}, \mathcal{S} – viz Tab. (7.1).5. Zvolené kombinaci kontur a kloubů odpovídá i počet dimenzí úlohy, kterou **eaICP** řeší. Ve **druhé části** jsou kontury klasifikovány ve smyslu (7.1.22) přesně podle Alg. (7.1).1. Jediný rozdíl je v tom, že hodnota *fitness* funkce na ř. 28 Alg. (7.1).1 není počítána podle (7.1.22), ale je čtena z tabulky \mathbf{D}_{ist} , která byla získána Alg. (7.1).6 a počítá se podle Alg. (7.1).7. Omezující podmínky dané (7.1.17) jsou dodrženy.

7.1.4 Experimentální výsledky

Kapitola experimentálních výsledků je rozdělena do několika částí: **A:** Použité databáze THID, HGDB, GPDS a značení obrazů rukou, **B:** časové nároky použitých evolučních optimalizátorů, **C:** Volba vhodného evolučního optimalizátoru, **D:** Volba pracovních parametrů, přesnost výpočtu, **E:** Způsob klasifikace, zvolená sestavení estimátorů, **F:** Výsledky klasifikace s využitím estimátoru **eaICP** a porovnání s ostatními pracemi, **G:** Přesnost algoritmu **eaICP** při klasifikaci identických snímků z databáze THID.

7.1.4.1 Použité databáze THID, GPDS, HGDB a značení obrazů rukou

Pro účely experimentů byly využity tři různé databáze jmenovitě:

1) THID, (Faundez-Zanuy et al. 2014; Font-Aragones et al. 2013; SPLab 2020), obsahuje 1040 snímků od 104 osob, 10 snímků od každé osoby, rozlišení 640x480. Celkem 108160 porovnání kontur. Použité rozlišením snímků 640x480 není sice ideální, ale malý počet bodů kontury (průměrně cca 1700) nevyžaduje při zpracování extrémně velké výpočetní nároky. Pro účely některých experimentů byly obrázky z databáze THID převzorkovány na rozlišení 1280x960pixelů s využitím bikubické interpolace (Keys 1981; Wikipedia 2020). Databáze THID představuje hlavní zdroj dat při experimentech.

2) GPDS (Travesio et al. 2004; Ferrer et al. 2007), obsahuje 940 snímků od 94 osob, 10 snímků od každé osoby, rozlišení snímků je různé 1403x1021, 1620x1280, 1402x1018 atd. Celkem 88360 porovnání kontur. Pro experimentální účely byly obrázky převzorkovány na rozlišení 1403x1021 a 640x480, podobně jako u databáze THID.

3) Hand Geometry Data Base (HGDB) (Klonowski et al. 2018), obsahuje 300 snímků od 60 osob, 5 snímků od každé osoby, rozlišení snímků 1754x1275pixelů. Celkem 18000 porovnání kontur.

Dále byly použito i několik snímků z vlastního biometrického scanneru (Moravec & Hub 2015; Moravec & Hub 2016).

Některé snímky z jednotlivých databází nebylo možné použít, a to z následujících důvodů:

1. Snímky byly špatně nasvícené. Barva stínu byla identická s barvou pozadí. Např. téměř každý snímek v databázi THID je jinak nasvícen. Většina snímků nevhodně.
2. Testovaná osoba si nevyhrnula rukáv košile/bundy a ta pak zastiňovala konturu ruky.
3. Černě (tmavě) nalakované nehty, kde barva lakování korespondovala s barvou černého pozadí nebo byla ještě tmavší než pozadí.
4. Tmavé široké prsteny (kovové, dřevěné, pletené/háčkované) nejen na prsteníčku, ale na dalších prstech.
5. Prsty (jeden nebo více) nebo část ruky obvázaná po úrazu.
6. Černé nebo tmavě hnědé fleky na nehtech a na prstech, které zasahují do kontury ruky a mají stejnou barvu jako pozadí.
7. Náramky (jeden nebo více) zasahující do kontury ruky.
8. Brva pleti. Snímek ruky osoby s černou pletí, která koresponduje s barvou pozadí, se zpracovává velmi špatně.
9. Stín vržený identifikovanou osobou, která stála na nevhodném místě a zastínila lampu na stropě.
10. Na několika snímcích byl také problém s dlouhými nehty, které za dobu snímkování narostly, což deformovalo konturu prstů. Některé z nehtů byly lakované tmavě modře.

11. Na některých snímcích byly zjištěny různé dlouhé prsty u stejné osoby až o 1cm. To samozřejmě není fyzicky možné.
12. V několika případech patřil snímek S testované osoby nějaké jiné osobě.

Všechny snímky, které byly nějakým způsobem poškozené, jak je uvedeno v bodech 1-12, byly pochopitelně nepoužitelné. Bylo by samozřejmě možné všechny „nevhodné“ části kontury ruky vynechat nebo uměle modifikovat jako např. v (Klonowski et al. 2018; Sharma et al. 2015), ale takto bychom mohli postupovat i v místech, kde jsou např. prsteny (prsten může být na každém prstě) a tak by postupně z kontury ruky nezbylo mnoho. Snímky, které nebylo možné použít (cca 2 procenta snímků) byly nahrazeny jiným náhodným snímkem od dané osoby, aby bylo možné zachovat stejný počet snímků u každé osoby pro jednotlivé databáze. Výrazně lepší výsledky a jednodušší zpracování obrazu z fotoaparátu dává temná komora, která má zesponu podsvícenou matnou projekční desku (Moravec & Hub 2016).

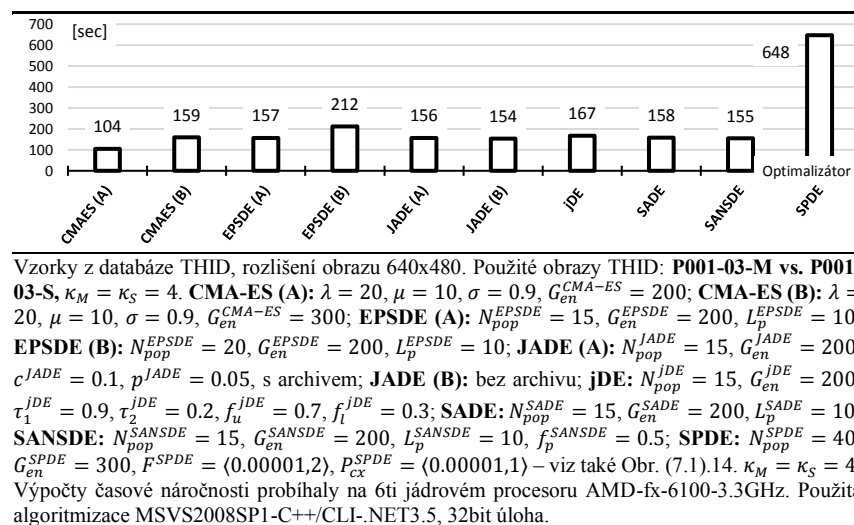
U každé osoby byl vybrán jeden snímek, který vykazoval ideální charakteristiky. Nebyl nijak poškozen, byl dobře osvětlen a ani nedošlo k zástině kontury ruky. Tento snímek byl po předzpracování a nalezení kontury označen jako modelová kontura M . Skupina snímků obsahující jak modely, tak ostatní datové snímky byla po předzpracování a nalezení kontury zařazena do skupiny datových vzorků S . Pro experimentální účely byly originální snímky z kamery označeny následovně:

$$P\langle xxx \rangle - \langle yy \rangle - M.BMP \text{ a } P\langle xxx \rangle - \langle yy \rangle - S.BMP$$

kde $\langle xxx \rangle$ značí číslo osoby, $\langle yy \rangle$ značí číslo snímku, M značí model a S značí datový vzorek (sample). V grafech výsledků budou vždy uváděny názvy snímků bez přípony BMP např. „P001-03-M“ nebo „ M P001-03“.

7.1.4.2 Časové nároky použitých evolučních optimalizátorů

Důležitým faktorem pro výběr vhodného evolučního optimalizátoru je kromě výsledné přesnosti a spolehlivosti také časová náročnost. Pro testy časové náročnosti byly zvoleny snímky z databáze THID o nativním rozlišení 640x480 pixelů. Na Obr. (7.1).6 jsou uvedeny testované optimalizátory.



Obrázek (7.1).6 Časová náročnost testovaných optimalizátorů při porovnání dvou vzorků.

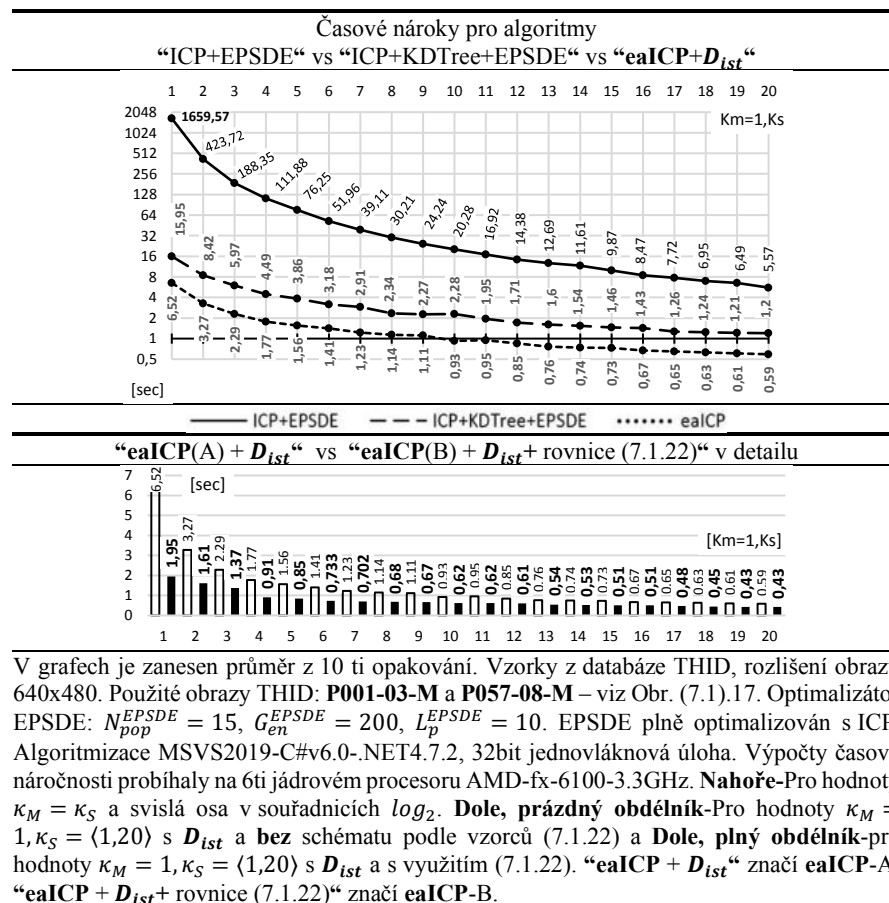
Podle očekávání je nejrychlejší CMAES. Optimalizátory EPSDE, JADE, jDE, SADE a SANDSE vykazují všechny téměř shodnou rychlost zpracování úlohy. Na posledním místě je pak SPDE. Jeho časové nároky téměř 2.65x převyšují nároky algoritmů jako JADE, jDE nebo EPSDE. CMA-ES je

jednoznačným vítězem testů rychlosti konvergence, ale bohužel jeho schopnost nalézt optimální řešení dané úlohy je horší – viz následující kapitola. Výpočet proběhl pro hodnoty $\kappa_M = \kappa_S = 4$ tak, aby všechny testované optimalizátory měly stejné pracovní podmínky a aby bylo možné je vzájemně důvěryhodně porovnat bez ohledu na strukturu algoritmu.

Na Obr. (7.1).7 je zobrazen čas potřebný k zarovnání kontur P001-03-M a P057-08-S pro různé hodnoty κ_M, κ_S v intervalu $\langle 1,20 \rangle$. Porovnávány jsou čtyři kombinace a to:

- 1) ICP+EPSDE (Besl & McKay 1992),
- 2) sestavení ICP+KDTTree+EPSDE, (Bentley 1975)
- 3) **eaICP(A)** s využitím matice D_{ist} a **bez** (7.1.22),
- 4) **eaICP(B)** s využitím matice D_{ist} a **včetně** (7.1.22)

Pro obě testované varianty **eaICP(A)** i **eaICP(B)** bylo zvoleno $\kappa_M = \kappa_S = \langle 1,20 \rangle$ viz Obr. (7.1).7. Hodnota κ_M samozřejmě nehraje díky matici D_{ist} žádnou roli. Je to pouze demonstrace rychlosti. Mechanismus KDTTree použitý u 2) lze nalézt v (Bentley 1975; Greenspan & Yurick 2003) a byl zvolen jen pro demonstraci rychlosti, nic víc.



Obrázek (7.1).7 Časové nároky nutné k dosažení optimální hodnoty v sec. při různé hodnotě κ_M, κ_S v intervalu $\langle 1,20 \rangle$ pro testované algoritmy.

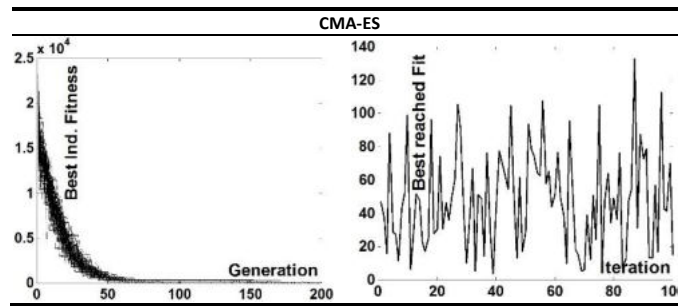
Algoritmus EPSDE, byl pro dané účely plně optimalizován pro spolupráci s ICP. Je dobře vidět, že kombinace ICP+EPSDE bez jakékoliv úpravy a s $\kappa_M = \kappa_S = 1$ má časovou náročnost cca 27 minut a ICP+KDTTree+EPSD cca 16 sekund. Oproti tomu **eaICP(A)** jen cca 6.5 sekundy a **eaICP(B)** skvělých 1.95 sekundy díky všem akceleračním postupům popsaným v předchozích kapitolách. Tj. algoritmus ICP byl zrychlen 851x a přitom dosahuje identické přesnosti. Počet bodů kontury \mathcal{M} resp. vzorku P001-03-M byl 1861 a počet bodů kontury \mathcal{S} vzorku P057-08-S byl 1415. Výpočetní nároky pro klasifikaci

108160 kombinací obrazů pro konfiguraci ICP+EPSDE by byly neúnosné s ohledem na dostupné technické vybavení. Kombinace ICP+KDTree+EPSDE je pro $\kappa_M = \kappa_S = 1$ stále 2x pomalejší než **eaICP(A)**. Je to dáno tím, že procházení stromu KD-Tree není možné nijak obejít. Z hlediska efektivity nemá vůbec smysl KD-Tree mechanismus pro řešení takovéto úlohy používat. Při výpočtech s hlavní databází byl pochopitelně využit jen **eaICP(B)**. KD-tree mechanismus ani samotný algoritmus **eaICP(A,B)** nijak neporušují pravidlo zaručené konvergence popsané v (Pottmann et al. 2006; Maier-Hein et al. 2012). Na Obr. (7.1).18 je zaznamenán úbytek přesnosti při zarovnání kontur pro různé hodnoty κ_M, κ_S vůči přesnosti zarovnání, které poskytuje kombinace $\kappa_M = \kappa_S = 1$ jako referenční hodnota. Úbytek přesnosti je zcela minimální a to cca 646 pixelů na plochu celé dlaně – viz Obr. (7.1).18. Takováto chyba v zarovnání kontur je pouhým okem nerozpoznatelná. Odchylky se samozřejmě týkají všech genů x_j vektoru \mathbf{X}_i . Na Obr. (7.1).7 je časová závislost pro $\kappa_M = \kappa_S$. V případě, že $\kappa_M = \langle 1, 20 \rangle, \kappa_S = 1$, je časová náročnost pro **eaICP(A)** vždy 6.52sec a pro **eaICP(B)** vždy 1.95sec bez ohledu na hodnotu κ_M , protože časová náročnost závisí už jen na hodnotách κ_S – viz (7.1.22). Pokud $\kappa_M = 1, \kappa_S = \langle 1, 20 \rangle$ pak časová náročnost odpovídá hodnotám na Obr. (7.1).7 a těchto časů bylo dosaženo právě díky využití matice \mathbf{D}_{ist} . V praktické implementaci je možné hodnotu κ_S měnit v každé generaci EA podle předem vygenerované tabulky.

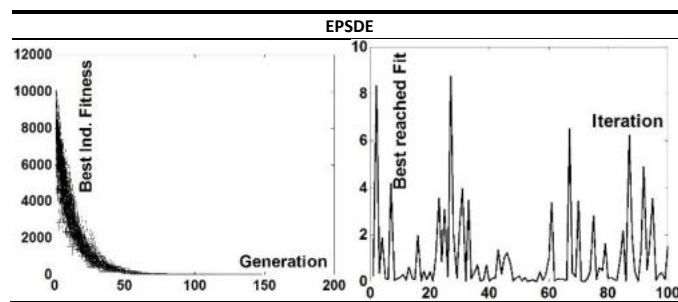
7.1.4.3 Volba vhodného evolučního optimalizátoru

Bylo testováno celkem 10 evolučních algoritmů pro vhodnost použití při řešení dané optimalizační úlohy a to: CMA-ES, EPSDE, JADE, jDE, SADE, SANSDE, SPDE a okrajově také PSO, RPSO a SGA – originál (Holland 1975). Pro účely těchto testů byl zvolen algoritmus využívající výpočet *fitness* bez matice \mathbf{D}_{ist} . V testech byly použity vzorky \mathcal{M} : P001-03-M a \mathcal{S} : P001-03-S, ale u \mathcal{M} byly nepatrně změněny polohy bodů P_K pro všechny prsty. Kontura zůstala identická. Byly nepatrně upraveny zcela náhodně tak, aby výsledná *fitness* nebyla zcela nulová a aby optimalizátor měl úlohu těžší. Např. pro malíček \mathcal{M} : $P_{K6}.x = 346.97181, P_{K6}.y = 151.68178$ a poloha korespondujícího kloubu \mathcal{S} : $P_{K6}.x = 346.87492, P_{K6}.y = 152.33841$. Díky této úpravě vzniklo okolo ideální polohy při zarovnání kontur mnoho ostře ohraničených lokálních optim. Při experimentech se totiž ukázalo, že např. CMA-ES neobyčejně rychle a naprosto přesně zarovná kontury, které jsou ve všech ohledech zcela identické, ale má obrovský problém nalézt optimum, pokud kontury identické nejsou, nebo pokud jsou jen málo odlišné, to samé platí o SGA. Dále byly použity hodnoty $\kappa_M = \kappa_S = 4$ -tý vzorek kontury \mathcal{M} a \mathcal{S} . V ideálním případě pak musí výsledná hodnota *fitness* být rovna nule. Chyba v řádu pixelů je nepodstatná. Aby bylo možné jednotlivé optimalizátory porovnat, byl všude tam, kde to je možné zvolen stejný počet generací a jedinců v populaci a to $G_{gen} = 200, N_{pop} = 15$. Pro CMA-ES pak $\lambda = 20, \mu = 10, \sigma = 0.9$. Hodnota $G_{gen} = 200$ je volena na časové omezení s ohledem na velikost databáze. Na obrázcích je vždy zobrazena konvergence populace pro 100x opakovaný pokus a pak také nejlepší dosažená hodnota *fitness* v jednotlivých iteracích. Protože testované EA mají různý počet pracovních parametrů, byly testovány jejich různé kombinace a bylo vybráno vždy takové nastavení, které poskytovalo nejlepší výsledek. Výsledky testů a nastavení s ohledem na individuální parametry je zaznamenáno na Obr. (7.1).8 – (7.1).14. Je poměrně dobře vidět, že nejlepší výsledky poskytuje poměrně paradoxně optimalizátor EPSDE – viz Obr. (7.1).9. Na druhém místě těsně následuje optimalizátor jDE – viz Obr. (7.1).11. EPSDE a jDE vykazují jen malé rozdíly, ale EPSDE je lepší i když jen o málo cca 5-7 procent. Na dalších místech pak jsou algoritmy SADE – viz Obr. (7.1).12, JADE – viz Obr. (7.1).10 a nebyl nalezen rozdíl mezi JADE s archivem a bez archivu, SANSDE – viz Obr. (7.1).13 a CMA-ES – viz Obr. (7.1).8. Velmi špatné výsledky poskytuje SPDE – viz Obr. (7.1).14, který vykazuje častá uváznutí v lokálním optimu. Algoritmy JADE, jDE, SADE, EPSDE a SANSDE jsou si velmi podobné co do průběhu konvergence k optimálnímu řešení. Rozdíly nastávají až v posledních cca 50ti generacích, kdy nejlepší řešení dané úlohy je schopen nejlépe nalézt algoritmus jDE a EPSDE. Velkým zklamáním je CMA-ES. Rychlost konvergence je značná – viz Obr. (7.1).8. Již po 100-120 generacích dosahuje relativně dobré hodnoty *fitness*, ale pokles *fitness* k nule v dalších

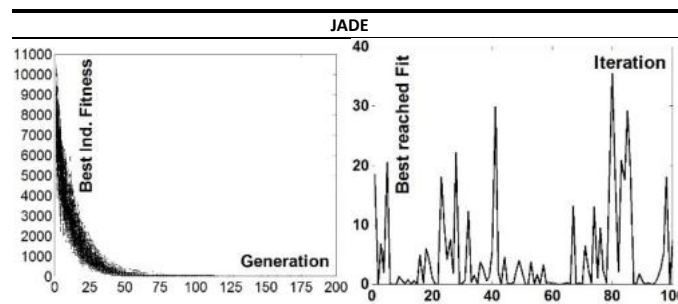
generacích je značný problém. CMA-ES vykazuje opakované uvážnutí v lokálním optimu. Jednoduše se na řešení dané úlohy nehodí. Kaskádovou kombinací CMA-ES a např. jDE nebo EPSDE viz (Moravec & Hub 2016) by se dobrý výsledek nedal získat rychleji, protože použitelné hodnoty *fitness* se pro CMA-ES dosahují až v rozmezí 100-120 generací. Jinak řečeno CMA-ES by se ve spojení např. s jDE stala stejně rychlá a účinná jako jDE samotné. Podle (Hansen 2006) je doporučeno vypočítat počet jedinců v populaci podle vzorce $P_{op}^{CMA-ES} = 4 + \lfloor 3 * \log_2 N_{dim} \rfloor$, kde N_{dim} je počet dimenzí úlohy, v našem případě např. $N_{dim} = 8$. Tento vzorec v dané úloze bohužel nefunguje. Bylo zjištěno, čistě experimentálně, že dobré výsledky poskytuje hodnota $P_{op}^{CMA-ES} = 15 - 20$. Další zvyšování počtu jedinců populace nad hodnotu 20 již k lepším výsledkům nevede.



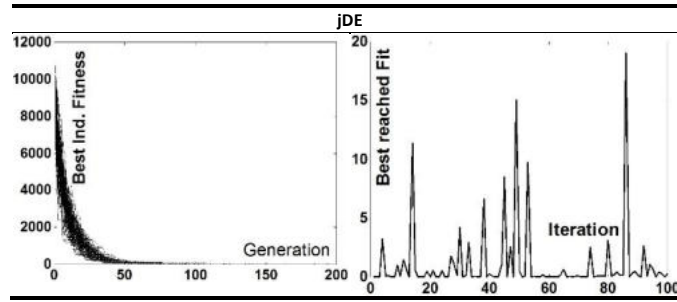
Obrázek (7.1).8 Optimalizátor **CMA-ES**: Pracovní parametry $\lambda = 20$, $\mu = 10$, $\sigma = 0.9$, $G_{en}^{CMA-ES} = 200$. **Vlevo**-konvergence populace pro 100 iterací evoluce, **Vpravo**-nejlepší dosažená hodnota v každé iteraci. $\kappa_M = \kappa_S = 4$.



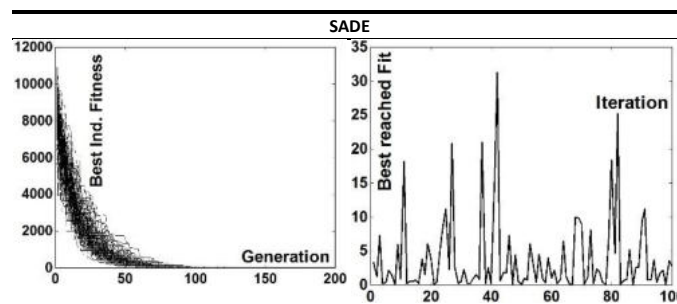
Obrázek (7.1).9 Optimalizátor **EPSDE**. Pracovní parametry $N_{pop}^{EPSDE} = 15$, $G_{en}^{EPSDE} = 200$, $L_p^{EPSDE} = 10$. **Vlevo**-konvergence populace pro 100 iterací evoluce, **Vpravo**-nejlepší dosažená hodnota v každé iteraci. $\kappa_M = \kappa_S = 4$.



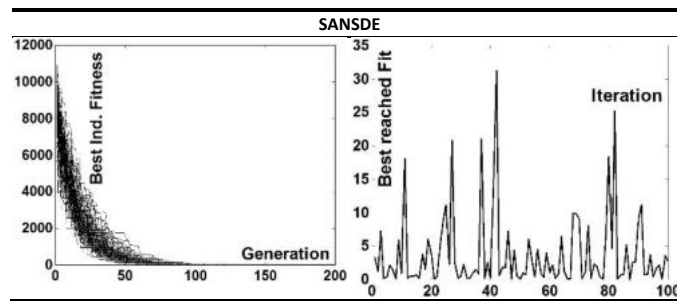
Obrázek (7.1).10 Optimalizátor **JADE**. Pracovní parametry $N_{pop}^{JADE} = 15$, $G_{en}^{JADE} = 200$, $c^{JADE} = 0.1$, $p^{JADE} = 0.05$, s archivem. Téměř identické výsledky poskytuje nastavení „s archivem“ a „bez archivu“. **Vlevo**-konvergence populace pro 100 iterací evoluce, **Vpravo**-nejlepší dosažená hodnota v každé iteraci. $\kappa_M = \kappa_S = 4$.



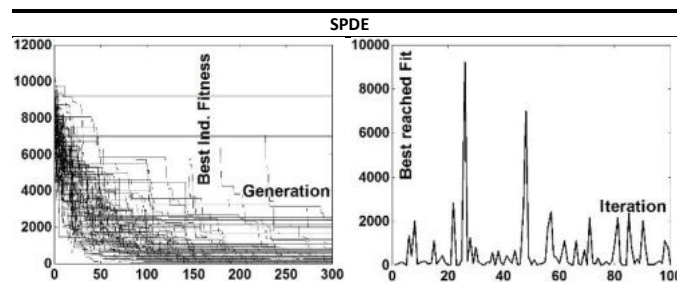
Obrázek (7.1).11 Optimalizátor **jDE**. Pracovní parametry $N_{pop}^{jDE} = 15$, $G_{en}^{jDE} = 200$, $\tau_1^{jDE} = 0.9$, $\tau_2^{jDE} = 0.2$, $f_u^{jDE} = 0.7$, $f_l^{jDE} = 0.3$. **Vlevo**-konvergence populace pro 100 iterací evoluce, **Vpravo**-nejlepší dosažená hodnota v každé iteraci. $\kappa_M = \kappa_S = 4$.



Obrázek (7.1).12 Optimalizátor **SADE**. Pracovní parametry $N_{pop}^{SADE} = 15$, $G_{en}^{SADE} = 200$, $L_p^{SADE} = 10$. **Vlevo**-konvergence populace pro 100 iterací evoluce, **Vpravo**-nejlepší dosažená hodnota v každé iteraci. $\kappa_M = \kappa_S = 4$.



Obrázek (7.1).13 Optimalizátor **SANSDE**. Pracovní parametry $N_{pop}^{SANSDE} = 15$, $G_{en}^{SANSDE} = 200$, $L_p^{SANSDE} = 10$, $f_p^{SANSDE} = 0.5$. **Vlevo**-konvergence populace pro 100 iterací evoluce, **Vpravo**-nejlepší dosažená hodnota v každé iteraci. $\kappa_M = \kappa_S = 4$.



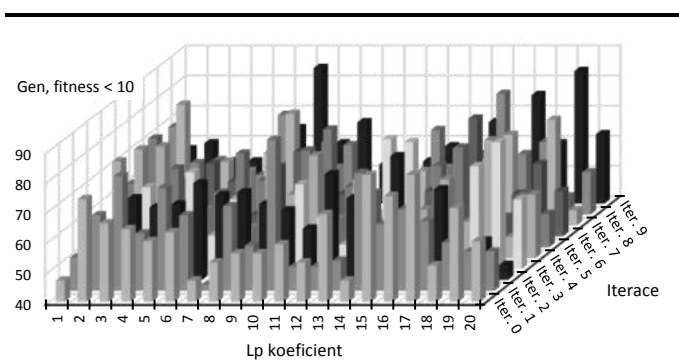
Obrázek (7.1).14 Optimalizátor **SPDE**. Pracovní parametry $N_{pop}^{SPDE} = 40$, $G_{en}^{SPDE} = 300$, $F^{SPDE} = \langle 0.00001, 2 \rangle$, $P_{cx}^{SPDE} = \langle 0.00001, 1 \rangle$. **Vlevo**-konvergence populace pro 100 iterací evoluce, **Vpravo**-nejlepší dosažená hodnota v každé iteraci. $\kappa_M = \kappa_S = 4$.

Okrajově byly testovány také optimalizátory Particle Swarm Optimisation (PSO) (Kennedy & Eberhart 1995), RPSO (Urfalioglu 2004). Oba ve své originální podobě. PSO vykazoval velmi špatné výsledky podobně jako RPSO. K řešení dané úlohy se nehodí. Oba algoritmy navíc vyžadují až 20x vyšší počet jedinců v populaci, což je časově neúnosné. Testován byl také algoritmus SGA (Goldberg 1987; Goldberg 1989; Holland 1992), jako zástupce starších typů optimalizátorů, které pochází z 60. a 70. let a to ve své původní podobě s chromosomem, který je vyjádřen pomocí řetězce bitů o délce chromozomu/genu 136/17 bitů, $N_{pop}^{SGA} = 20$, $G_{en}^{SGA} = 200$, přesnost 1×10^{-5} a s různými typy evolučních operátorů. SGA umožňuje velmi rychlou počáteční konvergenci a chová se podobně jako CMA-ES, ale má velký problém nalézt přesné řešení a často uvázne v lokálním optimu, nebo konvergence selže úplně. Právě náhodná totální selhání konvergence k optimálnímu řešení činí SGA obtížně použitelným.

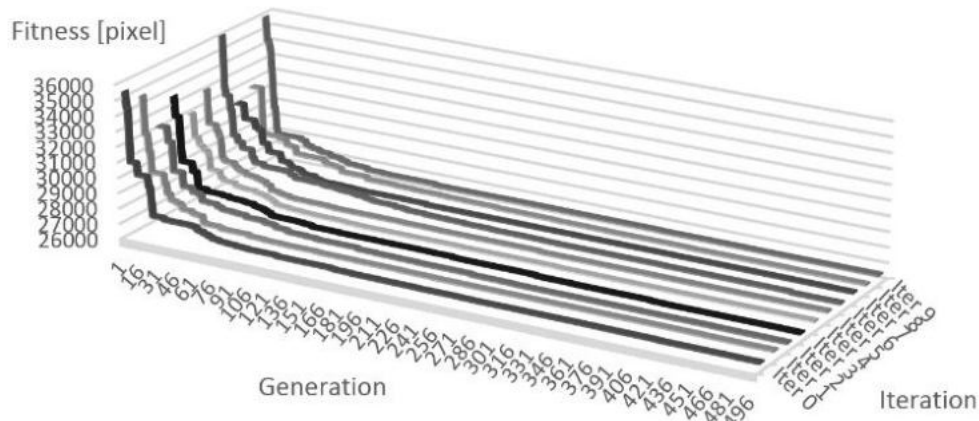
Jako nejvhodnější optimalizátor byl na základě výsledků časové náročnosti a výsledků přesnosti zvolen algoritmus EPSDE (157 sekund). Není sice tak rychlý jako CMA-ES (104 sekund), ale při rozumném počtu generací je podobně jako jDE (167 sekund) schopen dosáhnout očekávaného cíle, a především u něj nedochází k náhodným fatálním selháním konvergence jako např. u SGA. V testech byla použita kombinace ICP+EA. Použitá algoritmizace má spíše obecný charakter a byla vytvořena pomocí MS VS 2008 SP1 C++/CLI .NET3.5, proto se dosažené časy liší od Obr. (7.1).17, ale poměr zůstává stejný.

7.1.4.4 Volba pracovních parametrů, přesnost výpočtu

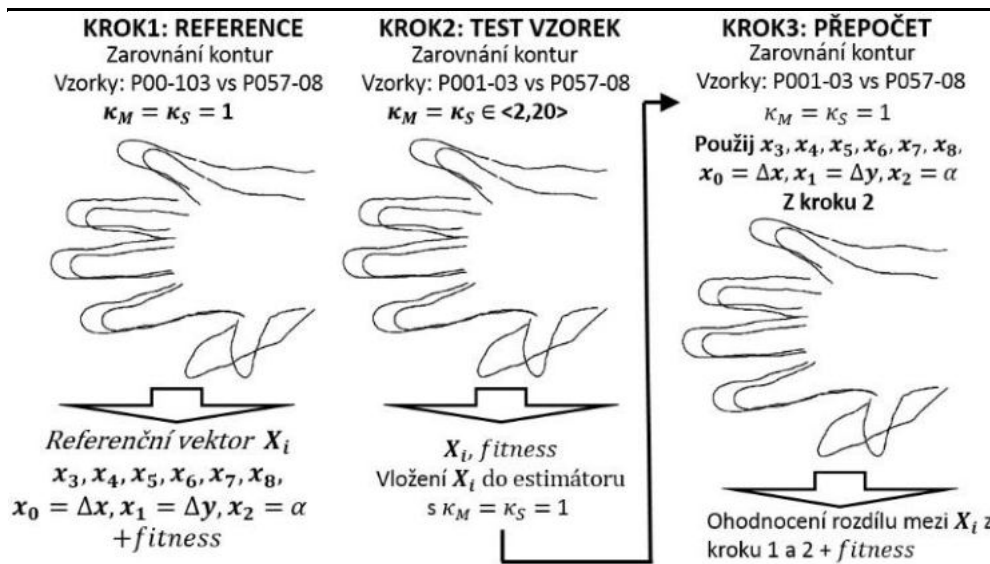
Zvolený optimalizátor EPSDE má pouze jeden pracovní parametr a to L_p^{EPSDE} . Je také nutné určit počet jedinců v populaci a počet generací jako ukončovací podmínku a samozřejmě hodnoty η_M, η_S resp. κ_M, κ_S . Na základě provedených experimentů bylo zjištěno, že parametr L_p^{EPSDE} nemá velký vliv na výsledek optimalizace – viz Obr. (7.1).15 a byl proto při všech výpočtech nastaven na hodnotu 10. Volba hodnot κ_M, κ_S probíhala s ohledem na technické možnosti a přesnost výpočtu. Pro testy byly zvoleny vzorky označené jako P001-03-M a P057-08-S – viz Obr. (7.1).17. Jednalo se kontury větší a menší ruky, kde evidentně neexistuje jednoznačná pozice pro zarovnání. Aby bylo možné zjistit únosné minimální hodnoty κ_M, κ_S použitelné pro další experimenty, byly nejprve obě kontury zarovnané pro $\kappa_M = \kappa_S = 1$ a použitý algoritmus **eaICP(A)** s parametry $N_{pop}^{EPSDE} = 20$, $G_{en}^{EPSDE} = 500$, $L_p^{EPSDE} = 10$ vypočetl hodnoty x_j vektoru \mathbf{X}_i . Počet opakování byl 10 – viz Tab. (7.1).4 a záznam konvergence je na Obr. (7.1).16. Dále byla zjištěna průměrná hodnota *fitness* 26526,50 pixelů z těchto 10ti opakování.



Obrázek (7.1).15 Vliv hodnoty L_p^{EPSDE} na schopnost konvergence. Byly zvoleny hodnoty $L_p^{EPSDE} \in \langle 1, 20 \rangle$. Výsledné hodnoty v grafu udávají, v které generaci bylo poprvé dosaženo hodnoty *fitness* < 10 pixelů. Pokus byl iterativně opakován 10x. Vzorky THID: **P001-03-M** vs. **P001-03-S**. Hodnota L_p^{EPSDE} má jen nepatrný, spíše žádný vliv. $\kappa_M = \kappa_S = 4$. Matice \mathbf{D}_{ist} nebyla použita.



Obrázek (7.1).16 Záznam konvergence populace pro THID: **P001-03-M** a **P057-08-S**, EPSDE, $N_{pop}^{EPSDE} = 20$, $G_{en}^{EPSDE} = 500$, $L_p^{EPSDE} = 10$, $\kappa_M = \kappa_S = 1$. *fitness* viz Tab. (7.1).4. Je dobře vidět, že od 200. generace se *fitness* již příliš nemění.



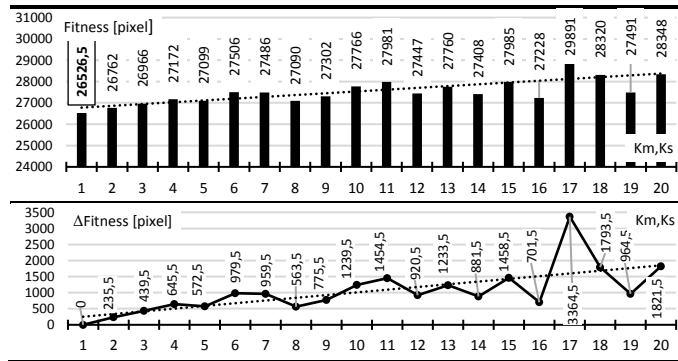
Obrázek (7.1).17 Schéma postupu při odhadu přesnosti navržené metody pro různé hodnoty κ_M, κ_S . V prvním kroku je zjištěna hodnota *fitness* pro $\kappa_M = \kappa_S = 1$, poté je měněna hodnota κ_M, κ_S v daném rozsahu a vektor X_i je vložen do algoritmu **eaICP(A)** s $\kappa_M = \kappa_S = 1$ a vypočítána *fitness* – viz Obr. (7.1).18, (7.1).16. Vzorky THID: **P001-03-M** a **P057-08-S** (menší ruka).

Tabulka (7.1).4

Výsledek konvergence, vzorky THID: **P001-03-M** vs **P057-08-S**, 10 iterací, $\kappa_M = \kappa_S = 1$

| Kloub. | P_{K1} | P_{K3} | P_{K4} | P_{K5} | P_{K6} | P_{CMass} | P_{CMass} | P_{CMass} | - |
|---------|----------|----------|----------|----------|----------|------------------|------------------|-----------------------|----------------|
| Unit | RAD | RAD | RAD | RAD | RAD | pixel | pixel | RAD | pixel |
| It./Gen | x_0 | x_1 | x_2 | x_3 | x_4 | $x_5 (\Delta x)$ | $x_6 (\Delta y)$ | $x_7 (\Delta \alpha)$ | <i>fitness</i> |
| 1 | -0,2568 | -0,0323 | 0,0461 | 0,0859 | 0,0856 | -1,4200 | -1,5222 | -0,1664 | 26460 |
| 2 | -0,1954 | -0,0095 | 0,0799 | 0,1086 | 0,1058 | -0,1682 | 0,2979 | -0,1754 | 26587 |
| 3 | -0,2179 | -0,0098 | 0,0795 | 0,1207 | 0,1306 | -4,5415 | 0,3046 | -0,1926 | 26460 |
| 4 | -0,2605 | -0,0127 | 0,0818 | 0,1139 | 0,1116 | -8,1795 | 0,0923 | -0,1840 | 26447 |
| 5 | -0,2412 | -0,0203 | 0,0818 | 0,1114 | 0,1071 | -8,5342 | -0,4332 | -0,1816 | 26571 |
| 6 | -0,2717 | -0,0117 | 0,0877 | 0,1266 | 0,1332 | -8,0947 | 2,1573 | -0,1627 | 26584 |
| 7 | -0,2277 | -0,0346 | 0,0458 | 0,0860 | 0,0817 | -1,6233 | -2,0762 | -0,1661 | 26509 |
| 8 | -0,2711 | -0,0329 | 0,0526 | 0,1007 | 0,1071 | -7,3600 | -0,4107 | -0,1612 | 26593 |
| 9 | -0,2802 | -0,0339 | 0,0516 | 0,1012 | 0,1059 | -7,2485 | -1,1953 | -0,1675 | 26520 |
| 10 | -0,2129 | -0,0306 | 0,0501 | 0,0912 | 0,0899 | -7,1302 | -0,6741 | -0,1653 | 26534 |
| AVG | -0,2435 | -0,0228 | 0,0657 | 0,1046 | 0,1059 | -5,4300 | -0,3460 | -0,1723 | 26526,50 |
| DEV | 0,0274 | 0,0104 | 0,0167 | 0,0134 | 0,0163 | 3,0539 | 1,1187 | 0,0100 | 53,79 |
| MIN | -0,2802 | -0,0346 | 0,0458 | 0,0859 | 0,0817 | -8,5342 | -2,0762 | -0,1926 | 26447,00 |
| MAX | -0,1954 | -0,0095 | 0,0877 | 0,1266 | 0,1332 | -0,1682 | 2,1573 | -0,1612 | 26593,00 |

Tabulka udává hodnoty genů x_j chromosomu X_i a *fitness* pro 10 iterací. Hodnoty z řádky AVG pak byly použity jako reference pro porovnání přesnosti zarovnání kontur algoritmem EPSDE při nižším počtu bodů kontury $\kappa_M = \kappa_S$ v rozsahu $\langle 2,20 \rangle$. AVG – průměr z iterací, DEV – směrodatná odchylka pro iterace, MIN, MAX – minimální a maximální nalezená hodnota. It.-číslo iterace. Gen.-číslo genu chromosomu X_i . $\Delta x, \Delta y, \Delta \alpha$ - odpovídá pořadí genům x_5, x_6, x_7 . Kloub-korespondující kloub.

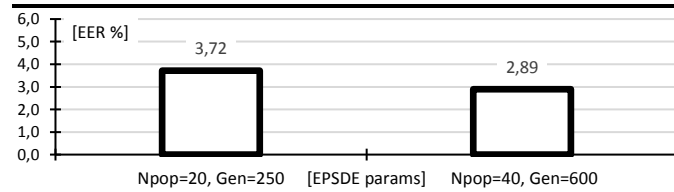


Obrázek (7.1).18 **Nahoře**-Dosažené hodnoty *fitness* pro postupně se zvyšující hodnoty κ_M, κ_S . Vzorky THID: **P001-03-M** a **P057-08-S**. vektor X_i byl po výpočtu zpětně vložen do estimátoru s $\kappa_M = \kappa_S = 1$ a byla odhadnuta *fitness*. $\kappa_M = \kappa_S = 1$ je referenční vzorek. **Dole**-Rozdíl mezi *fitness* = 26526.5 pro $\kappa_M = \kappa_S = 1$ a *fitness*, která byla získána pro aktuálně testovanou hodnotou κ_M, κ_S . (Tečkovanou čarou je zakreslena spojnice trendu.)

Poté byl pokus proveden pro řadu hodnot $\kappa_M = \kappa_S \in \langle 2,20 \rangle$ s krokem 1 s nastavením EPSDE: $N_{pop}^{EPSDE} = 15$, $G_{en}^{EPSDE} = 200$, $L_p^{EPSDE} = 10$. Každý pokus byl opět opakován 10x. Výsledné hodnoty byly zaznamenány a průměrná hodnota – v Tab. (7.1).4 označená jako AVG z genů x_j chromosomu X_i z oněch 10ti opakování pro všechny $\kappa_M = \kappa_S \in \langle 2,20 \rangle$ byla pro jednotlivé výsledky vložena zpět jako správné řešení do algoritmu **eaICP(A)**, kde $\kappa_M = \kappa_S = 1$ a byla vypočtena *fitness* – viz Obr. (7.1).17. Podle předpokladu se hodnota *fitness* pozvolna zvětšovala při zvětšujících se hodnotách κ_M, κ_S . S ohledem na časové nároky – viz Obr. (7.1).6, (7.1).7.

Na základě rozsáhlých experimentů byl ze skupiny evolučních optimalizátorů zvolen EPSDE – vykazuje nejmenší počet selhání konvergence a obsahuje jen minimum pracovních parametrů. Pouze hodnoty N_{pop}^{EPSDE} a G_{en}^{EPSDE} . Tyto hodnoty je nutné nastavit ručně a podle očekávání jsou závislé na rozlišení obrazu a počtu bodů kontury. Pro rozlišení 640×480 pixelů jsou dostačující hodnoty $N_{pop}^{EPSDE} = 15 - 20$, $G_{en}^{EPSDE} = 200$, $L_p^{EPSDE} = 10$ – viz Obr. (7.1).16. Bohužel v průběhu experimentů se jednoznačně ukázalo, že při značném počtu konvergencí v řádech desítitisíců dochází i u EPSDE k nedokonalé konvergenci. Nedojde sice k úplnému selhání konvergence jako např. u SGA, ale výsledná hodnota pro každou cca tisíci evoluci nedosáhne ideální hodnoty zarovnání kontur. To se projeví na horší hodnotě koeficientu *EER*. Tento jev je zachycen na Obr. (7.1).19 a má universální charakter pro všechna rozlišení obrazu. Rozdíl mezi ideálními hodnotami *EER* při velkých počtech evolucí je sice malý cca 0.8%, ale

je viditelný. Pro dosažení ještě lepších hodnot je možné zvýšit počet jedinců v populaci a počet generací např. na $N_{pop}^{EPSDE} = 40$, $G_{en}^{EPSDE} = 600$. Celý proces funguje tak, že pro hodnoty N_{pop}^{EPSDE} , G_{en}^{EPSDE} odhadnuté podle Obr. (7.1).16 dochází pro zdvojnásobení počtu individuí a ztrojnásobení počtu generací ke snížení hodnoty EER o 0.8 procenta. Nastavením takových pracovních hodnot se ovšem výrazně zvýší výpočetní a samozřejmě i časové nároky šestkrát. Naštěstí další zvyšování N_{pop}^{EPSDE} , G_{en}^{EPSDE} již nemá smysl, protože výsledná hodnota EER se již výrazně nesnižuje resp. výsledná hodnota zlepšení koeficientu EER o 0.01% nebo 0.001% postrádá hlubší smysl s ohledem na testované databáze daných velikostí.



Obrázek (7.1).19. Vliv hodnot N_{pop}^{EPSDE} , G_{en}^{EPSDE} na výslednou hodnotu $EER\%$ pro 108160 evolucí resp. zarovnání kontur. Databáze THID, 640x480, kontury $c_{4-7}^{M_{ea}}$, $c_{4-7}^{S_{ea}}$, 4 prsty, R_I , $D_{im} = 7$.

7.1.4.5 Způsob klasifikace a zvolená nastavení estimátorů

Drtivá většina autorů např. (Barra et al. 2019; Sharma et al. 2015; Bera & Bhattacharjee 2017; Klonowski et al. 2018) atd., používá pro klasifikaci jen některé části ruky, obvykle 4 prsty bez palce a pro popis této části ruky používá sadu několika desítek geometrických primitiv a případně další prvky. Vyhýbá se tak značně komplikované oblasti palce, který má 3 klouby. **eaICP** využívá různé části kontury ruky, které jsou složeny ze stovek nebo tisíců bodů. Pro účely experimentů byly zvoleny následující kombinace kontur, rozlišení obrazu a klasifikačního kritéria podle (7.1.22) – viz Tab. (7.1).5 a také Tab. (7.1).2 a Alg. (7.1).7.

Pro klasifikaci výsledků jsou využívány hodnoty FAR (False Acceptance Rate), FRR (False Rejection Rate), bod EER (Equal Error Rate point or Equilibrium) a ROC křivka (Receiver Operating Characteristic) – viz např. (Sanches-Reillo et al. 2000). Podobně jako u (Luque-Baena et al. 2013; Barra et al. 2019; Klonowski et al. 2018), bod EER je uváděn v procentech a korespondující hodnota $fitness$ v pixelech. Klasifikace získaných výsledků, je prováděna na základě hodnot FAR a FRR podle vzorců (7.1.28):

| | | |
|----------------|--|----------|
| | $FAR = \frac{\Sigma I_{UL}}{\Sigma I_{TOT}}, FRR = \frac{\Sigma G_{OL}}{\Sigma G_{TOT}}$ | |
| Databáze THID: | $\Sigma G_{TOT} = 104 \times 10 = 1040$ $\Sigma I_{TOT} = 108160 - 1040 = 107120$ | (7.1.28) |
| Databáze GPDS: | $\Sigma G_{TOT} = 94 \times 10 = 940$ $\Sigma I_{TOT} = 88360 - 940 = 87420$ | |
| Databáze HGDB: | $\Sigma G_{TOT} = 60 \times 5 = 300$ $\Sigma I_{TOT} = 18000 - 300 = 17700$ | |

kde ΣG_{OL} – součet všech pravých (genuine) vzorků, u kterých však byl překročen limit λ_{EER} , ΣG_{TOT} – celkový počet pravých vzorků, ΣI_{UL} – počet podvodníků (impostors), kteří díky chybné klasifikaci dostali pod hodnotu λ_{EER} , ΣI_{TOT} – celkový počet podvodníků. Způsob výpočtu hodnot G_{OL} a I_{UL} a samozřejmě i FAR , FRR , EER byl proveden podle Alg. (7.1).6. Hodnota λ_{EER} reprezentuje v určitém okamžiku nalezený bod equilibria EER v pixelech. Je to $fitness$ vypočtená podle (7.1.22). Aby bylo možné zjistit hodnotu equilibria λ_{EER} je nutné výpočet opakovat pro všechny hodnoty ve zvoleném rozmezí $\lambda_{EER} = \langle 0, lim \rangle$. Hodnota lim je dána maximální hodnotou $fitness$ funkce pro všechny snímky

\mathcal{M} od stejných osob. Bod, kde se křivky FAR a FRR protínají, je definován jako bod EER – viz např. Tab. (7.1).7. V ROC diagramu je pak bod EER definován jako průsečík ROC křivky a hlavní diagonály. Pro tento bod jsou také udávány hodnoty FAR a FRR v jednotlivých tabulkách výsledků.

| Tabulka (7.1).5 | |
|--|---|
| Zvolené kombinace rozlišení obrazu, kontur ruky, klasifikačního kritéria (7.1.21), sestavy kloubů, počtu dimenzí, estimátoru | |
| Usp. | Databáze, rozlišení, kontury, poč. prstů, klasifikátor, klouby, poč.dim., estimátor |
| A1 | THID, 1280x960, $c_{4-8}^{\mathcal{M}co}, c_{4-8}^{Sea}$, 4 prsty, $R_H + R_I, P_{K3-6}, D_{im} = 7, E9$ |
| A2 | THID, 1280x960, $c_{4-8}^{\mathcal{M}ea}, c_{4-8}^{Sea}$, 4 prsty, $R_H + R_I, P_{K3-6}, D_{im} = 7, E9$ |
| A3 | THID, 1280x960, $c_{4-7}^{\mathcal{M}ea}, c_{4-7}^{Sea}$, 4 prsty, $R_I, P_{K3-6}, D_{im} = 7, E5$ |
| A4 | THID, 640x480, $c_{4-8}^{\mathcal{M}co}, c_{4-8}^{Sea}$, 4 prsty, $R_H + R_I, P_{K3-6}, D_{im} = 7, E9$ |
| A5 | THID, 640x480, $c_{4-7}^{\mathcal{M}co}, c_{4-7}^{Sea}$, 4 prsty, $R_H + R_I, P_{K3-6}, D_{im} = 7, E10$ |
| A6 | THID, 640x480, $c_{1-8}^{\mathcal{M}ea}, c_{1-8}^{Sea}$, 5 prstů, $R_I, P_{K1,3-6}, P_{K2}$ fixován, $D_{im} = 8, E11$ |
| A7 | THID, 640x480, $c_{1-8}^{\mathcal{M}co}, c_{1-8}^{Sea}$, 5 prstů, $R_H + R_I, P_{K1-6}, D_{im} = 9, E8$ |
| A8 | GPDS, 1403x1021, $c_{4-8}^{\mathcal{M}co}, c_{4-8}^{Sea}$, 4 prsty, $R_H + R_I, P_{K3-6}, D_{im} = 7, E9$ |
| A9 | GPDS, 640x480, $c_{1-8}^{\mathcal{M}ea}, c_{1-8}^{Sea}$, 5 prstů, $R_I, P_{K1-6}, D_{im} = 9, E3$ |
| A10 | GPDS, 640x480, $c_{1-8}^{\mathcal{M}ea}, c_{1-8}^{Sea}$, 5 prstů, $R_I, P_{K1,3-6}, P_{K2}$ fixován, $D_{im} = 8, E11$ |
| A11 | GPDS, 640x480, $c_{4-8}^{\mathcal{M}co}, c_{4-8}^{Sea}$, 4 prsty, $R_H + R_I, P_{K3-6}, D_{im} = 7, E9$ |
| A12 | HGDB, 1754x1275, $c_{4-8}^{\mathcal{M}co}, c_{4-8}^{Sea}$, 4 prsty, $R_H + R_I, P_{K3-6}, D_{im} = 7, E9$ |

Popis jednotlivých označení Ex, s ohledem na (Moravec 2020b) a sestavení A1 až A12:

Označení eaICP estimátoru E3, 9-ti dim. úloha: $x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8$; výpočtu se účastní všechny kontury $c_1^-, c_2^-, c_3^-, c_4^-, c_5^-, c_6^-, c_7^-, c_8^-$ a korespondující klouby P_{K1-6} tj. všech 5 prstů, je použito jen pravidlo R_I pro výpočet *fitness* podle (7.1.22).

Označení eaICP estimátoru E5, 7-ti dim. úloha: $x_0, x_1, x_2, x_5, x_6, x_7, x_8$; klouby P_{K1}, P_{K2} jsou z výpočtu zcela vynechány a stejně tak i korespondující kontury c_1^-, c_2^-, c_3^- ; výpočtu se účastní jen kontury $c_4^-, c_5^-, c_6^-, c_7^-, c_8^-$ a klouby P_{K3-6} , je použito jen pravidlo R_I pro výpočet *fitness* podle (7.1.22).

Označení eaICP estimátoru E8, 9-ti dim. úloha: $x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8$; výpočtu se účastní všechny kontury $c_1^-, c_2^-, c_3^-, c_4^-, c_5^-, c_6^-, c_7^-, c_8^-$ a korespondující klouby P_{K1-6} tj. všech 5 prstů, jsou použita pravidla $R_H + R_I$ pro výpočet *fitness* podle (7.1.22).

Označení eaICP estimátoru E9, 7-mi dim. úloha: $x_0, x_1, x_2, x_5, x_6, x_7, x_8$; klouby P_{K1}, P_{K2} jsou z výpočtu zcela vynechány a stejně tak i korespondující kontury c_1^-, c_2^-, c_3^- ; výpočtu se účastní jen kontury $c_4^-, c_5^-, c_6^-, c_7^-, c_8^-$ a klouby P_{K3-6} , jsou použita pravidla $R_H + R_I$ pro výpočet *fitness* podle (7.1.22). E9 – globálně nejlepší dosažitelné výsledky.

Označení eaICP estimátoru E10, 7-mi dim. úloha: $x_0, x_1, x_2, x_5, x_6, x_7, x_8$; klouby P_{K1}, P_{K2} jsou z výpočtu zcela vynechány a stejně tak i korespondující kontury c_1^-, c_2^-, c_3^- a dále je vynechána kontura c_8^- ; výpočtu se účastní jen kontury $c_4^-, c_5^-, c_6^-, c_7^-$ a klouby P_{K3-6} , jsou použita pravidla $R_H + R_I$ pro výpočet *fitness* podle (7.1.22).

Označení eaICP estimátoru E11, 8-mi dim. úloha: $x_0, x_1, x_2, x_3, x_5, x_6, x_7, x_8$; kloub P_{K2} je pevný; výpočtu se účastní všechny kontury $c_1^-, c_2^-, c_3^-, c_4^-, c_5^-, c_6^-, c_7^-, c_8^-$ a korespondující klouby $P_{K1,3-6}$ tj. všech 5 prstů, je použito jen pravidlo R_I pro výpočet *fitness* podle (7.1.22).

Označení \mathcal{M}_{co} reprezentuje takovou konturu \mathcal{M} , která pro účely klasifikace není ořezána podle hodnot ve složených závorkách v Tab. (7.1).1., ale je kompletní. Označení \mathcal{M}_{ea}, S_{ea} reprezentuje konturu, která byla pro účely klasifikace ořezána podle hodnot ve složených závorkách v Tab. (7.1).1. Tento drobný detail – vhodné ořezání kontury a porovnávání ořezaných a neořezaných kontur \mathcal{M}_{co} a S_{ea} – přináší značně rozdílné výsledky v přesnosti klasifikace. $c_{1-8}^{\mathcal{M}ea}, c_{4-8}^{Sea}$ – značí čísla kontur podle Tab. (7.1).1. Označení R_H, R_I je podle (7.1.22). V některých experimentech jsou vynechána heuristická pravidla R_H a *fitness* je počítána pouze podle pravidla R_I . P_{K1-6} – značí čísla kloubů prstů, které se účastnily výpočtu podle Obr. (7.1).3, E9 – značí číslo estimátoru.

Byly provedeny experimenty s různými hodnotami κ_S , aby bylo možné zjistit, jak robustní jsou navržené klasifikátory. Jakákoliv hodnota κ_M , větší, než 1 vede vždy k výrazně horším výsledkům. Proto byla ve všech experimentech volena $\kappa_M = 1$. Algoritmizace byla provedena v MSVS-2019-C#v6.0-.NET-

4.7.2, 32bit-ová jedno-vláknová úloha pod Windows7-x32-SP1. Výpočty s hlavní databází THID probíhaly na 6ti jádrovém procesoru AMD-fx-6100-3.3GHz.

Algoritmus (7.1).6 Způsob výpočtu hodnot FAR, FRR, EER , databáze THID

Vstup: lim
Výstup: EER and FAR, FRR at EER point

```

1  $\Sigma G_{TOT} = 1040; \Sigma I_{TOT} = 107120; x \approx "oo"$ 
2 for  $\lambda_{EER}=0; \lambda_{EER} < lim; \lambda_{EER} = \lambda_{EER} + 1$ 
3    $G_{OL} = 0; I_{UL} = 0$ 
4   for all models  $\mathcal{M}$  (104)
5     for all samples  $\mathcal{S}$  (1040)
6       if  $S^{NAME} \equiv \mathcal{M}^{NAME} \wedge \varepsilon > \lambda_{EER}$  then  $G_{OL} = G_{OL} + 1$  (SELHÁNÍ)
7       if  $S^{NAME} \neq \mathcal{M}^{NAME} \wedge \varepsilon \leq \lambda_{EER}$  then  $I_{UL} = I_{UL} + 1$  (SELHÁNÍ)
8       if  $S^{NAME} \equiv \mathcal{M}^{NAME} \wedge \varepsilon \leq \lambda_{EER}$  then (úspěch) ☺
9     next
10  next
11   $FAR = I_{UL} / \Sigma I_{TOT}; FRR = G_{OL} / \Sigma G_{TOT}$ 
12  if  $x > |FAR - FRR|$  then
13     $x = |FAR - FRR|; EER = \lambda_{EER};$  vypočítej a ulož  $FAR, FRR;$ 
14  endif
15 next

```

$S^{NAME} \equiv \mathcal{M}^{NAME}$ - značí, že zkoumaná kontura \mathcal{S} a modelová kontura \mathcal{M} náleží stejnému člověku, identifikace probíhá podle jména vzorků viz (7.1.27). λ_{EER} - udává hodnotu *fitness* pro bod EER . lim -je nutné zjistit experimentálně viz Obr. (7.1).19. x - pomocná proměnná.

Aby se výpočet hodnot *fitness* urychlil, jsou používány proměnlivé hodnoty κ_S podle Tab. (7.1).6 a (7.1.23). Tyto hodnoty nijak neovlivňují výslednou přesnost, ale umožní rychlejší konvergenci. V Tab. (7.1).7 - (7.1).18 jsou zaznamenány důležité statistické ukazatele $\Sigma I_{UL}, FAR\%, G_{OL}, FRR\%$ a $EER\%$ podle (34) a Alg. (7.1).6. Bod EER je uveden v procentech a také v pixelech. Časová náročnost pro úspěšné zarovnání dvou kontur v sekundách a milisekundách [sec.ms] při použití jednoho jádra AMD-fx-6300. Je uvažován čistý čas bez režie, t.j. je uvažována jen doba potřebná pro běh jedné evoluce. Režie činí cca 15 procent a je bohužel variabilní, protože algoritmus běžel pod Windows 7-SP1-x32 v systému .NET. Režii operačního systému nelze ovlivnit.

Tabulka (7.1).6 Hodnoty κ_S v jednotlivých generacích podle (28)
pro obrazy z databáze THID při rozlišení 1280x960 pixelů

| No | $G_{EER}^{EPSDE} / \kappa_S$ | 1 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 |
|----|------------------------------|----|----|----|----|----|----|----|----|----|
| 1 | 0-39 | 40 | 40 | 40 | 40 | 40 | 40 | 40 | 40 | 40 |
| 2 | 40-79 | 20 | 20 | 20 | 30 | 35 | 35 | 40 | 40 | 40 |
| 3 | 80-129 | 10 | 10 | 10 | 15 | 30 | 30 | 30 | 35 | 40 |
| 4 | 130-400 | 1 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 |

Okamžiky přepnutí hodnoty κ_S v dané generaci G_{EER}^{EPSDE} .

Jednotlivá sestavení A1-A12 – viz Tab. (7.1).5, používají různé typy kontur, které nesou označení $\mathcal{M}_{co}, \mathcal{M}_{ea}, \mathcal{S}_{ea}$ – viz Tab. (7.1).1, (7.1).2, (7.1).5. Kontura \mathcal{M}_{co} není ořezaná podle schématu v Tab. (7.1).2, kontury typu $\mathcal{M}_{ea}, \mathcal{S}_{ea}$ jsou ořezány podle schématu podle Tab. (7.1).1. Na základě praktických experimentů bylo zjištěno, že při porovnávání kontur $\mathcal{M}_{co}, \mathcal{S}_{ea}$ je možné dosáhnout mnohem lepších výsledků, než při porovnávání kontur $\mathcal{M}_{ea}, \mathcal{S}_{ea}$ a to i při použití jen jediného vzoru (template) od každé osoby.

7.1.4.6 Výsledky klasifikace s využitím estimátoru eaICP a porovnání s ostatními pracemi

V Tab. (7.1).7 - (7.1).18 jsou zaznamenány výsledky experimentů pro uspořádání A1-A12 estimátoru **eaICP** – viz Tab. (7.1).5. Všechny výpočty samozřejmě probíhaly s využitím matice \mathbf{D}_{ist} – viz (7.1.22). Aby bylo možné zjistit, jak robustní jednotlivé estimátory jsou, byly postupně nastaveny různé hodnoty κ_S z množiny vybraných hodnot $\kappa_S \in \{1,5,10,15,20,25,30,35,40\}$. Hodnoty $\kappa_S = 40$ pro databázi THID a rozlišení 640x480 již odpovídají pouhým 7 bodům kontury \mathbf{S} na prst. I při takových podmínkách je navržený estimátor **eaICP** schopen pracovat bezvadně, i když s horšími výsledky, než při $\kappa_S = 1$ resp. při zachování modelu v Tab. (7.1).6.

Nejprve výsledky pro databázi THID (104 osob) a korespondující sestavení estimátorů A1-A7 a různé kombinace typů kontur a počtu prstů. Nejlepších výsledků algoritmu **eaICP** při práci s databází THID bylo dosaženo pro uspořádání A1: $FAR = 0.38\%$, $FRR = 0.38\%$, $EER = 0.38\%$, $IR = 99.61\%$ - viz Tab. (7.1).7, (7.1).19 a Obr. (7.1).20, (7.1).21. Hodnota A1: $EER = 0.38\%$, není „zcela“ ideální. Důvod je ten, že např. osoba P055 na snímcích P055-09, P055-10 nesprávně položila ruku na podložku – prsty byly nepřírodně ohnuté do úhlu 70 stupňů atd. Pokud je např. hodnota M_{ul2} v (7.1.22) nastavena na nulu, pak estimátor poskytuje výsledek $EER = 2.51\%$. (Barra et al. 2019) dosáhli na identické databázi nejlepší hodnotu $EER = 0.52\%$ s využitím algoritmu LDA a při použití samotného Euklidovského kritéria pouze na hodnotu $EER = 3.5\%$ – viz Tab (7.1).19. V Tab. (7.1).8 jsou uvedeny výsledky pro sestavení A2: $FAR = 0.48\%$, $FRR = 0.48\%$, $EER = 0.48\%$. Tyto výsledky jsou stále lepší, než jakých dosáhli (Barra et al. 2019) za použití algoritmu LDA. Hodnota $EER = 0.48\%$ pro sestavení A2 je oproti A1 horší, protože byly porovnávány kontury $\mathcal{M}_{ea}, \mathcal{S}_{ea}$. Pro sestavení A3, Tab. (7.1).9 poskytl estimátor **eaICP** výsledné hodnoty A3: $FAR = 1.92\%$, $FRR = 1.92\%$, $EER = 1.92\%$. Bylo zde použito pouze Euklidovské kritérium R_I – viz (7.1.22). Důvodem horšího výsledku, než je pro A1 je to, že samotný ICP algoritmus bez doplňkových heuristických pravidel R_H nemá takovou účinnost. Rozdíl mezi sestavením A1 a A3 je ten, že pro A3 byla vynechána část kontury ruky od malíčku k zápěstí. Skutečnost, že právě tato část kontury ruky je pro klasifikaci důležitá, bylo zjištěno až na základě experimentů. To je patrné z výsledků v Tab. (7.1).7 a Tab. (7.1).9. Pro sestavení A4 bylo dosaženo hodnot $FAR = 1.15\%$, $FRR = 1.15\%$, $EER = 1.15\%$, pro A5: $FAR = 1.83\%$, $FRR = 1.82\%$, $EER = 1.83\%$, pro A6: $FAR = 3.49\%$, $FRR = 3.46\%$, $EER = 3.49\%$ a pro A7: $FAR = 1.82\%$, $FRR = 1.82\%$, $EER = 1.82\%$. Sestavení A4 pracuje se 4 prsty a využívá obě klasifikační kritéria. Sestavení A5 je stejné jako A4, ale je vynechána část kontury ruky mezi malíčkem a zápěstím. Sestavení A6 pracuje s nastavením estimátoru s konturou ruky s 5ti prsty, ale palec má jen jeden kloub. Výsledky jsou proto nejhorší. V sestavení A7 má palec klouby dva. Díky tomu je výsledek výrazně lepší oproti A6. Z výsledků v Tab. (7.1).9, (7.1).10, (7.1).11 pro sestavení A3, A4, A5 je dobře vidět, že na výslednou hodnotu EER má velký vliv typ porovnávaných typů kontur $\mathcal{M}_{co}, \mathcal{M}_{ea}, \mathcal{S}_{ea}$. Nejlepších výsledků bylo možné dosáhnout právě pro kombinaci kontur $\mathcal{M}_{co}, \mathcal{S}_{ea}$ a při použití obou klasifikačních pravidel $R_H + R_I$ podle (7.1.22). Estimátor využívající kontury $\mathcal{M}_{ea}, \mathcal{S}_{ea}$ je také méně robustní. To je dobře vidět při porovnání výsledků pro $\kappa_S = 1 - 40$. Pro všechny estimátory, kde je použito jen pravidlo R_I bez R_H , bylo dosaženo horších výsledků. Pravidlo zaručené konvergence algoritmu ICP platí pro identické nebo „téměř“ identické shluky bodů, ale v případě klasifikace množiny bodů, která tvoří konturu ruky je algoritmem ICP upřednostněna pozice zarovnání shluků \mathcal{M} a \mathcal{S} ve směru konců prstů, kde je větší počet bodů, než v oblasti mezi prsty – u kořene prstu. Tento jev byl pozorován bez ohledu na typ použité kontury $\mathcal{M}_{co}, \mathcal{M}_{ea}, \mathcal{S}_{ea}$. To je také důvod, proč použití jen pravidla R_I nevede k tak dobrým výsledkům jako kombinace $R_H + R_I$. Díky pravidlu R_H je algoritmem ICP správně ztotožněna kontura \mathcal{M} a \mathcal{S} . V průběhu experimentů byla také měněna hodnota κ_S v rozsahu $\kappa_S = 1 - 40$ pro celou dobu evoluce v souladu s Tab. (7.1).6. Důvodem testování různých hodnot κ_S je samozřejmě snaha zjistit do jaké míry je estimátor **eaICP** robustní/stabilní při nižším počtu bodů, které konturu tvoří. Pro $\kappa_S = 40$ bylo např. dosaženo hodnot A1: $FAR = 0.63\%$, $FRR = 0.67\%$, $EER = 0.63\%$ - viz Tab. (7.1).7. Pro všechny ostatní sestavení jsou výsledky horší. Navržený estimátor **eaICP** v uspořádání A1 je velmi robustní a

poskytuje velmi dobré výsledky. Na Obr. (7.1).19 jsou zakresleny křivky FAR , FRR pro výsledky v Tab. (7.1).7. Na Obr. (7.1).21 jsou pak ROC (Receive Operating Characteristics) křivky pro jednotlivé testované hodnoty κ_S v rozsahu $\kappa_S = 1 - 40$.

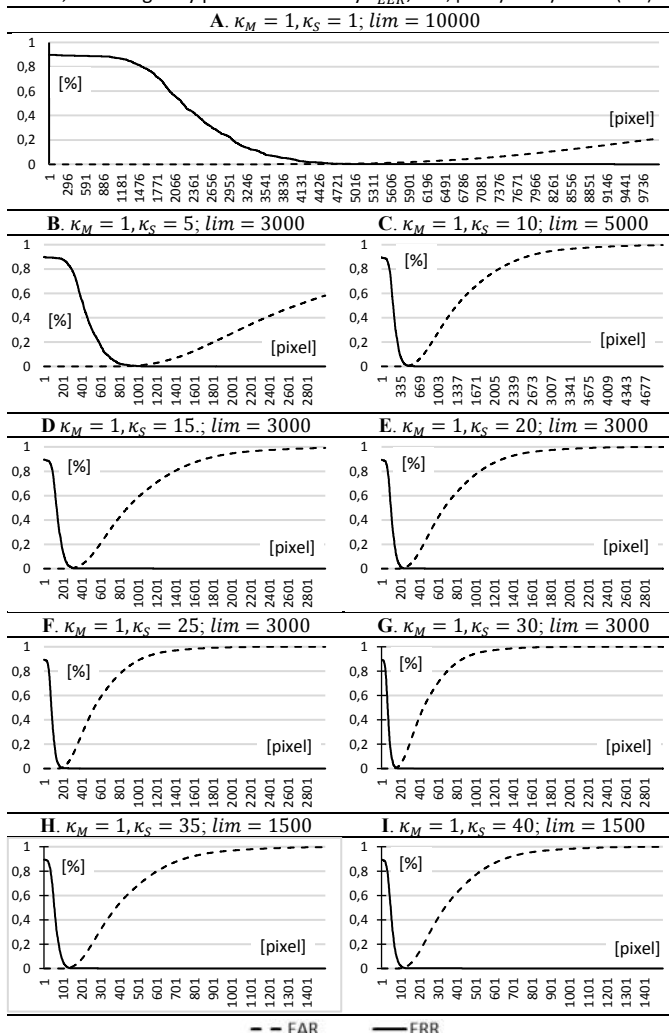
Tabulka (7.1).7 Hodnoty FAR, FRR, EER ,
973 440 evolucí podle Alg. (7.1). 1, (7.1).6, (7.1).7

A1: THID, 1280x960, $c_{4-8}^{M_{co}}$, $c_{4-8}^{S_{ea}}$, 4 prsty, $R_H + R_L, P_{K3-6}, D_{lim} = 7, E9$

| n. | κ_M | κ_S | ΣI_{UL} | $FAR\%$ | ΣG_{OL} | $FRR\%$ | $EER\% / \text{pixel}$ | Tot. time |
|----|------------|------------|-----------------|---------|-----------------|---------|------------------------|-----------|
| 1 | 1 | 1 | 412 | 0.38 | 4 | 0.38 | 0.38 / 4873 | 23.340 |
| 2 | 1 | 5 | 615 | 0.57 | 6 | 0.57 | 0.57 / 965 | 5.240 |
| 3 | 1 | 10 | 614 | 0.57 | 6 | 0.57 | 0.57 / 476 | 2.790 |
| 4 | 1 | 15 | 512 | 0.47 | 5 | 0.48 | 0.47 / 309 | 1.990 |
| 5 | 1 | 20 | 427 | 0.39 | 4 | 0.38 | 0.38 / 226 | 1.540 |
| 6 | 1 | 25 | 613 | 0.57 | 6 | 0.57 | 0.57 / 189 | 1.290 |
| 7 | 1 | 30 | 563 | 0.52 | 5 | 0.48 | 0.52 / 155 | 1.140 |
| 8 | 1 | 35 | 599 | 0.55 | 6 | 0.57 | 0.55 / 133 | 1.000 |
| 9 | 1 | 40 | 680 | 0.63 | 7 | 0.67 | 0.63 / 118 | 0.920 |

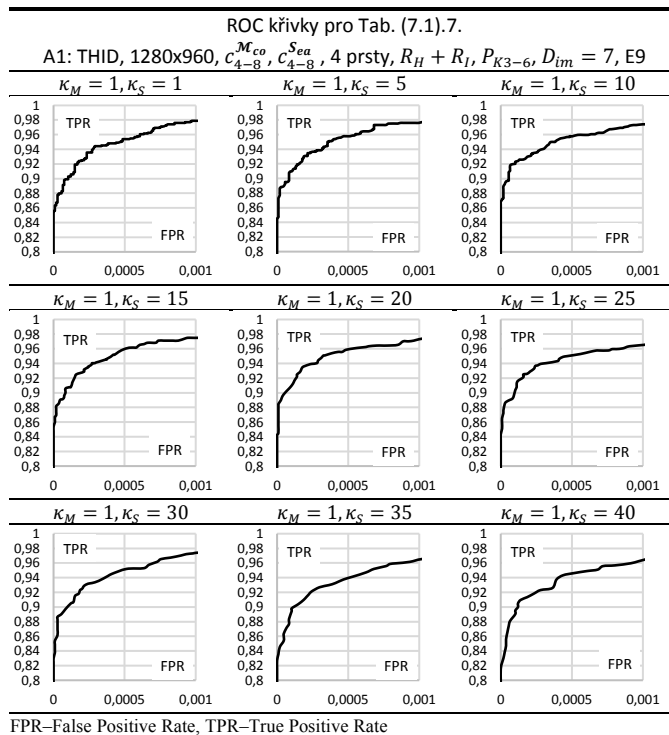
Časové nároky v [sec.ms] jsou uvažovány pro úspěšné zarovnání vzorků THID: **P001-03-M** a **P057-08-S** – vzorky viz Obr. (7.1).17. Celkový počet evolucí je: 9 řádků \times 108160 porovnání \mathcal{M} vs \mathcal{S} = 973 440 evolucí. Pro ř. 1. je hodnota „Identification rate“ (IR) rovna 99.61%. $N_{pop}^{EPSDE} = 40, G_{en}^{EPSDE} = 420$.

FAR, FRR diagramy pro různé hodnoty λ_{EER}, lim , pro výsledky v Tab. (7.1).7



Křivky pro eaICP-B. Svislá osa v procentech v rozmezí (0,0,1,0). Vodorovná osa udává hodnotu $fitness$ v pixelech resp. interval λ_{EER} , na kterém se hodnota EER vyhledává.

Obrázek (7.1).20 Křivky FAR, FRR pro různé hodnoty λ_{EER} a pro různé hodnoty κ_S .



Obrázek (7.1).21 ROC křivky pro estimátor A1 a různé hodnoty κ_S .

Tabulka (7.1).8 Hodnoty FAR , FRR , EER ,
973 440 evolucí podle Alg. (7.1). 1, (7.1).6, (7.1).7

A2: THID, 1280x960, $c_{4-8}^{M_{ea}}$, $c_{4-8}^{S_{ea}}$, 4 prsty, $R_H + R_I$, P_{K3-6} , $D_{im} = 7$, E9

| n. | κ_M | κ_S | ΣI_{UL} | $FAR\%$ | ΣG_{OL} | $FRR\%$ | $EER\%$ / pixel | Tot. time |
|----|------------|------------|-----------------|---------|-----------------|---------|-----------------|-----------|
| 1 | 1 | 1 | 515 | 0.48 | 5 | 0.48 | 0.48 / 4856 | 23.180 |
| 2 | 1 | 5 | 523 | 0.48 | 5 | 0.48 | 0.48 / 963 | 5.160 |
| 3 | 1 | 10 | 591 | 0.55 | 5 | 0.48 | 0.56 / 482 | 2.770 |
| 4 | 1 | 15 | 637 | 0.59 | 6 | 0.57 | 0.59 / 402 | 1.980 |
| 5 | 1 | 20 | 699 | 0.65 | 7 | 0.67 | 0.64 / 373 | 1.530 |
| 6 | 1 | 25 | 782 | 0.73 | 8 | 0.76 | 0.75 / 285 | 1.290 |
| 7 | 1 | 30 | 847 | 0.79 | 9 | 0.86 | 0.79 / 201 | 1.120 |
| 8 | 1 | 35 | 911 | 0.85 | 10 | 0.96 | 0.85 / 157 | 1.000 |
| 9 | 1 | 40 | 1079 | 1.00 | 10 | 0.96 | 1.00 / 132 | 0.900 |

Časové nároky v [sec.ms] jsou uvažovány pro úspěšné zarovnání vzorků THID: **P001-03-M** a **P057-08-S** – vzorky viz Obr. (7.1).17. Celkový počet evolucí je: 9 řádků \times 108160 porovnání \mathcal{M} vs $\mathcal{S} = 973\,440$ evolucí. $N_{pop}^{EPSDE} = 40$, $G_{en}^{EPSDE} = 420$.

Tabulka (7.1).9 Hodnoty FAR , FRR , EER ,
973 440 evolucí podle Alg. (7.1). 1, (7.1).6, (7.1).7

A3: THID, 1280x960, $c_{4-7}^{M_{ea}}$, $c_{4-7}^{S_{ea}}$, 4 prsty, R_I , P_{K3-6} , $D_{im} = 7$, E5

| n. | κ_M | κ_S | ΣI_{UL} | $FAR\%$ | ΣG_{OL} | $FRR\%$ | $EER\%$ / pixel | Tot. time |
|----|------------|------------|-----------------|---------|-----------------|---------|-----------------|-----------|
| 1 | 1 | 1 | 2062 | 1.92 | 20 | 1.92 | 1.92 / 4572 | 21.652 |
| 2 | 1 | 5 | 2078 | 1.93 | 22 | 2.11 | 1.93 / 914 | 4.860 |
| 3 | 1 | 10 | 2130 | 1.98 | 23 | 2.21 | 1.97 / 453 | 2.610 |
| 4 | 1 | 15 | 2243 | 2.09 | 26 | 2.50 | 2.09 / 342 | 1.830 |
| 5 | 1 | 20 | 2560 | 2.38 | 27 | 2.59 | 2.38 / 246 | 1.470 |
| 6 | 1 | 25 | 2782 | 2.59 | 29 | 2.78 | 2.59 / 201 | 1.210 |
| 7 | 1 | 30 | 3078 | 2.87 | 30 | 2.88 | 2.87 / 189 | 0.940 |
| 8 | 1 | 35 | 3200 | 2.98 | 30 | 2.88 | 2.98 / 157 | 0.920 |
| 9 | 1 | 40 | 3610 | 3.37 | 34 | 3.26 | 3.37 / 139 | 0.870 |

Časové nároky v [sec.ms] jsou uvažovány pro úspěšné zarovnání vzorků THID: **P001-03-M** a **P057-08-S** – vzorky viz Obr. (7.1).17. Celkový počet evolucí je: 9 řádků \times 108160 porovnání \mathcal{M} vs $\mathcal{S} = 973\,440$ evolucí. $N_{pop}^{EPSDE} = 40$, $G_{en}^{EPSDE} = 420$.

Tabulka (7.1).10 Hodnoty FAR, FRR, EER ,
973 440 evolucí podle Alg. (7.1). 1, (7.1).6, (7.1).7

A4: THID, 640x480, $c_{4-8}^{Mco}, c_{4-8}^{Sea}$, 4 prsty, $R_H + R_I, P_{K3-6}, D_{im} = 7, E9$

| n. | κ_M | κ_S | ΣI_{UL} | $FAR\%$ | ΣG_{OL} | $FRR\%$ | $EER\%$ / pixel | Tot. time |
|----------|------------|------------|-----------------|-------------|-----------------|-------------|--------------------|--------------|
| 1 | 1 | 1 | 1239 | 1.15 | 12 | 1.15 | 1.15 / 1248 | 1.996 |
| 2 | 1 | 5 | 1288 | 1.20 | 17 | 1.63 | 1.20 / 249 | 0.491 |
| 3 | 1 | 10 | 1302 | 1.21 | 19 | 1.82 | 1.22 / 115 | 0.296 |
| 4 | 1 | 15 | 1478 | 1.37 | 22 | 2.11 | 1.37 / 98 | 0.218 |
| 5 | 1 | 20 | 2077 | 1.93 | 25 | 2.40 | 1.95 / 92 | 0.187 |
| 6 | 1 | 25 | 2498 | 2.33 | 28 | 2.69 | 2.34 / 88 | 0.171 |
| 7 | 1 | 30 | 2870 | 2.67 | 29 | 2.78 | 2.67 / 62 | 0.150 |
| 8 | 1 | 35 | 3132 | 2.92 | 30 | 2.88 | 2.92 / 40 | 0.144 |
| 9 | 1 | 40 | 3538 | 3.33 | 30 | 2.88 | 3.30 / 33 | 0.140 |

Časové nároky v [sec.ms] jsou uvažovány pro úspěšné zarovnání vzorků THID: **P001-03-M** a **P057-08-S** – vzorky viz Obr. (7.1).17. Celkový počet evolucí je: 9 řádků \times 108160 porovnání \mathcal{M} vs $\mathcal{S} = 973\,440$ evolucí. $N_{pop}^{EPSDE} = 20, G_{en}^{EPSDE} = 200$.

Tabulka (7.1).11 Hodnoty FAR, FRR, EER ,
973 440 evolucí podle Alg. (7.1). 1, (7.1).6, (7.1).7

A5: THID, 640x480, $c_{4-7}^{Mco}, c_{4-7}^{Sea}$, 4 prsty, $R_H + R_I, P_{K3-6}, D_{im} = 7, E10$

| n. | κ_M | κ_S | ΣI_{UL} | $FAR\%$ | ΣG_{OL} | $FRR\%$ | $EER\%$ / pixel | Tot. time |
|----------|------------|------------|-----------------|-------------|-----------------|-------------|--------------------|--------------|
| 1 | 1 | 1 | 1963 | 1.83 | 19 | 1.82 | 1.83 / 1153 | 1.903 |
| 2 | 1 | 5 | 1998 | 1.86 | 19 | 1.82 | 1.85 / 230 | 0.483 |
| 3 | 1 | 10 | 2002 | 1.86 | 22 | 2.11 | 1.87 / 164 | 0.280 |
| 4 | 1 | 15 | 2140 | 1.99 | 27 | 2.59 | 1.97 / 130 | 0.218 |
| 5 | 1 | 20 | 2490 | 2.32 | 29 | 2.78 | 2.32 / 98 | 0.171 |
| 6 | 1 | 25 | 2673 | 2.49 | 30 | 2.88 | 2.49 / 74 | 0.156 |
| 7 | 1 | 30 | 2980 | 2.78 | 31 | 2.98 | 2.78 / 41 | 0.150 |
| 8 | 1 | 35 | 3054 | 2.85 | 33 | 3.17 | 2.85 / 32 | 0.144 |
| 9 | 1 | 40 | 3139 | 2.93 | 34 | 3.26 | 2.93 / 27 | 0.140 |

Časové nároky v [sec.ms] jsou uvažovány pro úspěšné zarovnání vzorků THID: **P001-03-M** a **P057-08-S** – vzorky viz Obr. (7.1).17. Celkový počet evolucí je: 9 řádků \times 108160 porovnání \mathcal{M} vs $\mathcal{S} = 973\,440$ evolucí. $N_{pop}^{EPSDE} = 20, G_{en}^{EPSDE} = 200$.

Tabulka (7.1).12 Hodnoty FAR, FRR, EER ,
973 440 evolucí podle Alg. (7.1). 1, (7.1).6, (7.1).7

A6: THID, 640x480, $c_{1-8}^{Mco}, c_{1-8}^{Sea}$, 5 prstů, R_I, P_{K1-6}, P_{K2} fixed, $D_{im} = 8, E11$

| n. | κ_M | κ_S | ΣI_{UL} | $FAR\%$ | ΣG_{OL} | $FRR\%$ | $EER\%$ / pixel | Tot. time |
|----------|------------|------------|-----------------|-------------|-----------------|-------------|--------------------|--------------|
| 1 | 1 | 1 | 3742 | 3.49 | 36 | 3.46 | 3.49 / 2902 | 1.970 |
| 2 | 1 | 5 | 3802 | 3.54 | 39 | 3.75 | 3.54 / 576 | 0.480 |
| 3 | 1 | 10 | 3917 | 3.65 | 42 | 4.03 | 3.65 / 268 | 0.290 |
| 4 | 1 | 15 | 4117 | 3.84 | 47 | 4.51 | 3.84 / 232 | 0.220 |
| 5 | 1 | 20 | 4627 | 3.31 | 50 | 4.80 | 3.31 / 201 | 0.190 |
| 6 | 1 | 25 | 4790 | 4.47 | 53 | 5.09 | 4.47 / 170 | 0.160 |
| 7 | 1 | 30 | 5237 | 4.88 | 57 | 5.48 | 4.88 / 120 | 0.140 |
| 8 | 1 | 35 | 5701 | 5.32 | 59 | 5.67 | 5.32 / 101 | 0.140 |
| 9 | 1 | 40 | 5744 | 5.36 | 59 | 5.67 | 5.36 / 82 | 0.140 |

Časové nároky v [sec.ms] jsou uvažovány pro úspěšné zarovnání vzorků THID: **P001-03-M** a **P057-08-S** – vzorky viz Obr. (7.1).17. Celkový počet evolucí je: 9 řádků \times 108160 porovnání \mathcal{M} vs $\mathcal{S} = 973\,440$ evolucí. $N_{pop}^{EPSDE} = 20, G_{en}^{EPSDE} = 200$.

Tabulka (7.1).13 Hodnoty FAR, FRR, EER ,
973 440 evolucí podle Alg. (7.1). 1, (7.1).6, (7.1).7

A7: THID, 640x480, $c_{1-8}^{Mco}, c_{1-8}^{Sea}$, 5 prstů, $R_H + R_I, P_{K1-6}, D_{im} = 9, E8$

| n. | κ_M | κ_S | ΣI_{UL} | $FAR\%$ | ΣG_{OL} | $FRR\%$ | $EER\%$ / pixel | Tot. time |
|----------|------------|------------|-----------------|-------------|-----------------|-------------|--------------------|--------------|
| 1 | 1 | 1 | 1956 | 1.82 | 19 | 1.82 | 1.82 / 2199 | 2.496 |
| 2 | 1 | 5 | 1960 | 1.82 | 19 | 1.82 | 1.82 / 435 | 0.608 |
| 3 | 1 | 10 | 1958 | 1.82 | 19 | 1.82 | 1.82 / 215 | 0.343 |
| 4 | 1 | 15 | 1970 | 1.83 | 20 | 1.92 | 1.84 / 198 | 0.280 |
| 5 | 1 | 20 | 1982 | 1.85 | 19 | 1.82 | 1.85 / 140 | 0.218 |
| 6 | 1 | 25 | 1989 | 1.85 | 20 | 1.92 | 1.85 / 100 | 0.202 |
| 7 | 1 | 30 | 1998 | 1.86 | 20 | 1.92 | 1.86 / 74 | 0.187 |
| 8 | 1 | 35 | 2008 | 1.87 | 20 | 1.92 | 1.87 / 56 | 0.171 |
| 9 | 1 | 40 | 2010 | 1.87 | 20 | 1.91 | 1.87 / 49 | 0.156 |

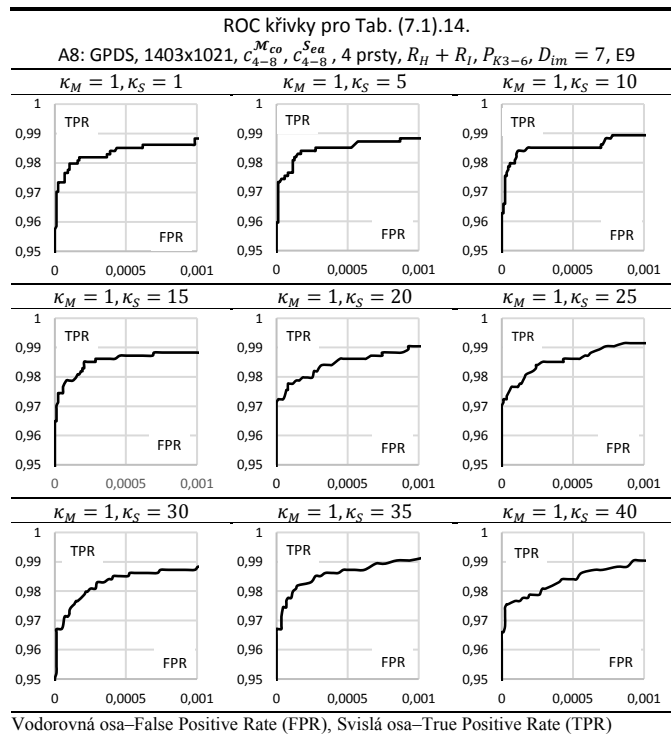
Časové nároky v [sec.ms] jsou uvažovány pro úspěšné zarovnání vzorků THID: **P001-03-M** a **P057-08-S** – vzorky viz Obr. (7.1).17. Celkový počet evolucí je: 9 řádků \times 108160 porovnání \mathcal{M} vs $\mathcal{S} = 973\,440$ evolucí. $N_{pop}^{EPSDE} = 20, G_{en}^{EPSDE} = 200$.

Pro práci s databází GPDS (94 osob) byly zvoleno sestavení estimátorů A8-A11 – viz Tab. (7.1).5 a také Tab. (7.1).14 - (7.1).17, s různým rozlišením obrazů, počtem prstů a kontur. Nejlepších výsledků bylo dosaženo pro sestavení A8: $FAR = 0.35\%$, $FRR = 0.31\%$, $EER = 0.35\%$, $IR = 99.68\%$ při nativním rozlišení obrazu 1403x1021 pixelů – viz Tab. (7.1).14 a Obr. (7.1).22. Na Obr. (7.1).22 jsou zobrazeny ROC křivky pro jednotlivé řádky v Tab. (7.1).14, sestavení A8. V některých případech nebylo možné konturu správně klasifikovat, protože jeden se snímků pořízených od stejné osoby měl např. kratší prsty nebo prsty byly jinak neočekávaně deformované – viz příklad na Obr. (7.1).23. Estimátor **eaICP** je bohužel velmi citlivý na podobné anomálie. (Angadi & Hatture 2016) dosáhli s využitím stejné databáze hodnot $FAR = 1.50\%$, $FRR = 3.5\%$ pro databázi o velikosti 50 osob a $FAR = 1.50\%$, $FRR = 5.95\%$ pro databázi o velikosti 100 osob. V práci (Angadi & Hatture 2018) pak autoři dosáhli hodnot $FAR = 0.50\%$, $FRR = 2.00\%$ pro 100 osob v databázi GPDS. V Tab. (7.1).14, sestavení A8, ř. 8 je uvedena hodnota $EER = 0.26\%$ pro $\kappa_S = 35$. Zpracovávaná úloha a celý mechanismus **eaICP** představuje silně nelineární systém. Výpočet byl několikrát opakován se stejným výsledkem a bylo zjištěno, že se jedná o „šťastnou“ náhodu. Pro rozlišení obrazů 640x480 viz Tab. (7.1).15 - (7.1).17, dosáhl estimátor **eaICP** hodnot A9: $FAR = 0.63\%$, $FRR = 0.63\%$, $EER = 0.63\%$ a pro A10: $FAR = 2.12\%$, $FRR = 2.12\%$, $EER = 2.12\%$. Horší výsledek pro A10 při porovnání s A9 je dán tím, že palec má jen jeden kloub, druhý kloub je fixovaný. Stejných výsledků bylo dosaženo i pro databázi THID. V obou případech A9 a A10 bylo použito jen klasifikační kritérium R_I , bez heuristických pravidel. Pro sestavení A11, Tab. (7.1).17 byly dosaženy hodnoty $FAR = 0.64\%$, $FRR = 0.63\%$, $EER = 0.64\%$. Při porovnání výsledků pro A8 a A11 je dobře vidět, že snížení rozlišení obrazu má značný vliv na výslednou přesnost. Hodnoty pro A11 jsou o polovinu horší, než pro A8 při zachování stejných pracovních parametrů (kromě rozlišení). Rozdíl mezi A9, A10 a A11 je ten, že A11 porovnává kontury typu $\mathcal{M}_{co}, \mathcal{S}_{ea}$ zatímco A9, A10 porovnávají kontury $\mathcal{M}_{ea}, \mathcal{S}_{ea}$. Podle výsledků je tedy dobře vidět, že velký vliv v případě porovnávání 5ti-prsté kontury má právě uspořádání se dvěma klouby palce. Je to samozřejmě dáno i tím, že použité snímky z databáze GPDS jsou velmi kvalitní. Kvalitnější než v databázi THID. Z Tab. (7.1).14 - (7.1).17 je poměrně dobře patrné, že navržené estimátory jsou značně robustní. Pro sestavení A8 a A11 se výsledné hodnoty EER téměř nemění ani při použití koeficientu $\kappa_S = 40$. Pro sestavení A11 – databáze GPDS, to znamená, že byly porovnávány kontury, kde algoritmus ICP pracoval s pouhými cca 10ti body na každý ze 4 prstů. V práci (Travesio et al. 2012) je také použita databáze GPDS, počet osob 100. Autoři bohužel neudávají hodnoty FAR , FRR , EER , ale pro databázi GPDS bylo dosaženo hodnoty $IR = 99.87\%$ pro porovnávací vzor (templates) sestavený ze třech kontur identické osoby. Pro jeden porovnávací vzor pak $IR = 99.42\%$.

Tabulka (7.1).14 Hodnoty FAR, FRR, EER ,
975 240 evolucí podle Alg. (7.1). 1, (7.1).6, (7.1).7
A8: GPDS, 1403x1021, $\mathcal{C}_{4-8}^{\mathcal{M}_{co}}, \mathcal{C}_{4-8}^{\mathcal{S}_{ea}}$, 4 prsty, $R_H + R_I, P_{K3-6}, D_{im} = 7, E9$

| n. | κ_M | κ_S | ΣI_{UL} | $FAR\%$ | ΣG_{OL} | $FRR\%$ | $EER\%$ / pixel | Tot. time |
|----|------------|------------|-----------------|---------|-----------------|---------|-----------------|-----------|
| 1 | 1 | 1 | 310 | 0.35 | 3 | 0.31 | 0.35 / 6006 | 42.470 |
| 2 | 1 | 5 | 311 | 0.35 | 3 | 0.31 | 0.35 / 1190 | 10.400 |
| 3 | 1 | 10 | 331 | 0.36 | 3 | 0.28 | 0.37 / 593 | 5.220 |
| 4 | 1 | 15 | 413 | 0.38 | 4 | 0.38 | 0.42 / 400 | 3.580 |
| 5 | 1 | 20 | 370 | 0.42 | 4 | 0.42 | 0.42 / 297 | 2.680 |
| 6 | 1 | 25 | 276 | 0.36 | 3 | 0.31 | 0.36 / 230 | 2.270 |
| 7 | 1 | 30 | 333 | 0.38 | 4 | 0.42 | 0.38 / 193 | 1.900 |
| 8 | 1 | 35 | 229 | 0.26 | 3 | 0.31 | 0.26 / 160 | 1.650 |
| 9 | 1 | 40 | 368 | 0.42 | 4 | 0.42 | 0.42 / 147 | 1.550 |

Časové nároky v [sec.ms] jsou uvažovány pro úspěšné zarovnání vzorků GPDS: **P001-03-M** a **P057-08-S**. Celkový počet evolucí je: 9 řádků \times 88360 porovnání \mathcal{M} vs $\mathcal{S} = 795\,240$ evolucí. Pro ř.1 je hodnota „Identification rate“ (IR) rovna 99.68%. $N_{pop}^{EPSDE} = 40$, $G_{en}^{EPSDE} = 420$.



Obrázek (7.1).22 ROC křivky pro estimátor A8 a různé hodnoty κ_S .

Tabulka (7.1).15 Hodnoty FAR, FRR, EER ,
975 240 evolucí podle Alg. (7.1). 1, (7.1).6, (7.1).7

A9: GPDS, 640x480, c_{1-8}^{Mca} , c_{1-8}^{Sea} , 5 prstů, R_I , P_{K1-6} , $D_{im} = 9$, E3

| n. | κ_M | κ_S | ΣI_{UL} | $FAR\%$ | ΣG_{OL} | $FRR\%$ | $EER\%$ / pixel | Tot. time |
|----|------------|------------|-----------------|---------|-----------------|---------|-----------------|-----------|
| 1 | 1 | 1 | 559 | 0.63 | 6 | 0.63 | 0.63 / 2455 | 4.210 |
| 2 | 1 | 5 | 1216 | 1.39 | 13 | 1.38 | 1.39 / 533 | 0.970 |
| 3 | 1 | 10 | 1232 | 1.40 | 13 | 1.39 | 1.40 / 250 | 0.630 |
| 4 | 1 | 15 | 1255 | 1.43 | 14 | 1.43 | 1.41 / 212 | 0.450 |
| 5 | 1 | 20 | 1274 | 1.45 | 14 | 1.45 | 1.44 / 189 | 0.380 |
| 6 | 1 | 25 | 1285 | 1.46 | 14 | 1.48 | 1.46 / 115 | 0.310 |
| 7 | 1 | 30 | 1623 | 1.85 | 17 | 1.80 | 1.85 / 87 | 0.270 |
| 8 | 1 | 35 | 1625 | 1.85 | 18 | 1.91 | 1.85 / 87 | 0.260 |
| 9 | 1 | 40 | 1640 | 1.87 | 18 | 1.91 | 1.87 / 76 | 0.255 |

Časové nároky v [sec.ms] jsou uvažovány pro úspěšné zarovnání vzorků GPDS: **P001-03-M** a **P057-08-S**. Celkový počet evolucí je: 9 řádků \times 88360 porovnávan \mathcal{M} vs $\mathcal{S} = 795$ 240 evolucí. $N_{pop}^{EPSDE} = 20$, $G_{en}^{EPSDE} = 200$.

Tabulka (7.1).16 Hodnoty FAR, FRR, EER ,
975 240 evolucí podle Alg. (7.1). 1, (7.1).6, (7.1).7

A10: GPDS, 640x480, c_{1-8}^{Mca} , c_{1-8}^{Sea} , 5 prstů, R_I , P_{K1-6} , P_{K2} fixed, $D_{im} = 8$, E11

| n. | κ_M | κ_S | ΣI_{UL} | $FAR\%$ | ΣG_{OL} | $FRR\%$ | $EER\%$ / pixel | Tot. time |
|----|------------|------------|-----------------|---------|-----------------|---------|-----------------|-----------|
| 1 | 1 | 1 | 1856 | 2.12 | 20 | 2.12% | 2.12 / 3324 | 4.290 |
| 2 | 1 | 5 | 1876 | 2.14 | 20 | 2.12% | 2.14 / 676 | 0.990 |
| 3 | 1 | 10 | 1894 | 2.16 | 21 | 2.23% | 2.16 / 345 | 0.520 |
| 4 | 1 | 15 | 2169 | 2.48 | 23 | 2.44% | 2.48 / 239 | 0.420 |
| 5 | 1 | 20 | 1967 | 2.25 | 22 | 2.34% | 2.25 / 181 | 0.350 |
| 6 | 1 | 25 | 2218 | 2.53 | 25 | 2.65% | 2.53 / 150 | 0.300 |
| 7 | 1 | 30 | 2304 | 2.63 | 23 | 2.44% | 2.63 / 127 | 0.270 |
| 8 | 1 | 35 | 2405 | 2.75 | 26 | 2.76% | 2.75 / 113 | 0.250 |
| 9 | 1 | 40 | 2613 | 2.98 | 28 | 2.97% | 2.98 / 100 | 0.200 |

Časové nároky v [sec.ms] jsou uvažovány pro úspěšné zarovnání vzorků GPDS: **P001-03-M** a **P057-08-S**. Celkový počet evolucí je: 9 řádků \times 88360 porovnávan \mathcal{M} vs $\mathcal{S} = 795$ 240 evolucí. $N_{pop}^{EPSDE} = 20$, $G_{en}^{EPSDE} = 200$.

Tabulka (7.1).17 Hodnoty FAR, FRR, EER ,
975 240 evolucí podle Alg. (7.1). 1, (7.1).6, (7.1).7
A11: GPDS, 640x480, $c_{4-8}^{Mco}, c_{4-8}^{Sea}$, 4 prsty, $R_H + R_I, P_{K3-6}, D_{im} = 7, E9$

| n. | κ_M | κ_S | ΣI_{UL} | $FAR\%$ | ΣG_{OL} | $FRR\%$ | $EER\%$ / pixel | Tot. time |
|----------|------------|------------|-----------------|-------------|-----------------|-------------|--------------------|--------------|
| 1 | 1 | 1 | 561 | 0.64 | 6 | 0.63 | 0.64 / 1371 | 3.280 |
| 2 | 1 | 5 | 561 | 0.64 | 6 | 0.64 | 0.64 / 268 | 0.840 |
| 3 | 1 | 10 | 560 | 0.64 | 6 | 0.64 | 0.64 / 136 | 0.500 |
| 4 | 1 | 15 | 560 | 0.64 | 6 | 0.64 | 0.64 / 115 | 0.360 |
| 5 | 1 | 20 | 561 | 0.64 | 6 | 0.64 | 0.64 / 97 | 0.300 |
| 6 | 1 | 25 | 560 | 0.64 | 6 | 0.63 | 0.64 / 85 | 0.250 |
| 7 | 1 | 30 | 561 | 0.64 | 6 | 0.63 | 0.64 / 67 | 0.240 |
| 8 | 1 | 35 | 561 | 0.64 | 6 | 0.64 | 0.64 / 46 | 0.210 |
| 9 | 1 | 40 | 560 | 0.64 | 6 | 0.63 | 0.64 / 29 | 0.200 |

Časové nároky v [sec.ms] jsou uvažovány pro úspěšné zarovnání vzorků GPDS: **P001-03-M** a **P057-08-S**. Celkový počet evolucí je: 9 řádků \times 88360 porovnání \mathcal{M} vs $\mathcal{S} = 795$ 240 evolucí. $N_{pop}^{EPSDE} = 20, G_{en}^{EPSDE} = 200$.

Velmi špatných výsledků dosáhl **eaICP** s databází HGDB a to $FAR = 4.44\%$, $FRR = 4.44\%$, $EER = 3.00\%$ - viz Tab. (7.1).18. (Klonowski et al. 2018) dosáhli s využitím navrženého HGCIa algoritmu na hodnoty $FAR = 0.00\%$, $FRR = 1.19\%$, $EER = 0.59\%$. Důvodem selhání **eaICP** je to, že téměř třetina obrazů v databázi má pro snímky stejné osoby různé délky a bohužel i šířky někdy i tvar některých prstů ruky – názorný příklad viz Obr. (7.1).23. Je velká záhada, jak jedna a tatáž osoba může mít různé délky, tloušťky a tvar prstů, kde rozdíl v délce činí až 10mm. (Klonowski et al. 2018) řešili tento problém tak, že zavedli váhový koeficient w pro každou měřenou hodnotu a tím eliminovali významný vliv měřených hodnot č.16,33,23,8,1 na celkový výsledek přesnosti pro jednotlivé obrazy. Dále pak uměle upravili špičky prstů u některých snímků z databáze podle vlastního algoritmu. U **eaICP** algoritmu něco takového bohužel není možné, znamenalo by to zcela vynechat části kontur ruky, která náleží špičkám prstů nebo vytvořit špičky prstů uměle podle vlastního algoritmu jako (Klonowski et al. 2018).

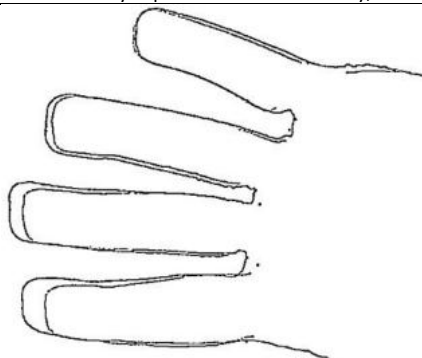
Tabulka (7.1).18 Hodnoty FAR, FRR, EER ,
18000 evolucí podle Alg. (7.1). 1, (7.1).6, (7.1).7
A12, HGDB, 1754x1275, $c_{4-8}^{Mco}, c_{4-8}^{Sea}$, 4 prsty, $R_H + R_I, P_{K3-6}, D_{im} = 7, E9$

| n. | κ_M | κ_S | ΣI_{UL} | $FAR\%$ | ΣG_{OL} | $FRR\%$ | $EER\%$ / pixel | Tot. time |
|----------|------------|------------|-----------------|-------------|-----------------|-------------|---------------------|---------------|
| 1 | 1 | 1 | 792 | 4.44 | 8 | 4.44 | 3.00 / 13005 | 45.310 |

Časové nároky v [sec.ms] jsou uvažovány pro úspěšné zarovnání vzorků HGDB: **P001-01-M** a **P001-05-S**. Celkový počet evolucí je 18000. $N_{pop}^{EPSDE} = 420, G_{en}^{EPSDE} = 420$.

Tabulka (7.1).19 Porovnání s ostatními pracemi

| n. | Autor/autoři | Metoda, databáze, další ukazatele | EER% |
|----|-----------------------------|---|--------------------|
| 1 | (Travieso et al. 2012) | DHMMK + SVM, GPDS db. + UST db. GPDS db. úspěšnost 99.87%, 3 templates UST db. úspěšnost 100%, 99.2% | 0.31 |
| 2 | (Travieso et al. 2004) | SVM, 5 prstů, GPDS db., úspěšnost 99.9% | - |
| 3 | (Dale et al. 2012) | Hand geometry + palmprint, 4 prsty, proprietární db., DCT: GAR=99.5% | 1.1198 |
| 4 | (Sharma et al. 2015) | Shape+Geometry, IITD database (EER=0.52%) Shape+Geometry, proprietární db., (EER=0.31%) | 0.52, 0.31 |
| 5 | (Kanhangad et al. 2009) | 2D+3D Hand geometry, proprietární db. (Kumar 2020) | 0.22 |
| 6 | (Kanhangad et al. 2009) | 2D Hand geometry, proprietární db. (Kumar 2020) | 6.3 |
| 7 | (Ferrer et al 2011) | Geometry + palm texture, proprietární db. | 0.17 |
| 8 | (Santos-Sierra et al. 2011) | SVM, k-NN, HTC db., UST db., IIT db. | 2.5,2.0,1.4 |
| 9 | (Santos-Sierra et al. 2009) | Kontura, parametrická křivka, proprietární db. | 3.7 |
| 10 | (Kang & Wu 2014) | Fourierovy deskriptory, 4 prsty, Bogazici University Hand Database | 3.69, 2.73 |
| 11 | (Varchol & Levicky 2007) | Gauss.mix.model, ED, Hamming vzdálenost, 5 prstů, neveřejná db. FAR=0.27%,FRR=10.42% | 6.45 |
| 12 | (Luque-Baena et al. 2013) | SGA optimalizátor, GPDS db., CASIA db., IITD-TPD db. Casia, GA-LDA: EER=4.64% IITD, GA-LDA: EER=4.51% | 4.51, 4.61 |
| 13 | (Morales et al. 2008) | SVM, 3 prsty, proprietární db. | 3.4 |
| 14 | (Michael et al. 2010) | Korelační koeficienty | 4.2 |
| 15 | (Bera & Bhattacharjee 2017) | k-NN, RF klasifikátor, 4 prsty, 5 prstů Bogazici University Hand Database | 6.0, 8.0 |
| 16 | (Dutagaci et al. 2008) | ICA, levá + pravá ruka, palmprint, říze kontury + textury, proprietární db. | 1.0, 1.16, 0.33 |
| 17 | (Yörük et al. 2006a) | Tvar ruky, ICA, 5 prstů, proprietární db. 458 osob, úspěšnost cca 99.48% | 6.14 |
| 18 | (Kumar et al. 2003) | Geometrie ruky, proprietární db. FAR=5.29%,FRR=8.34% | - |
| 19 | (Angadi & Hatture 2016) | Geometrie + SVM, GPDS db., FAR=1.5%,FRR=3.5% | - |
| 20 | (Angadi & Hatture 2018) | Geometrie, 16 topologických ukazatelů, graf minimálních délek hran, Euklidovská vzdálenost, GPDS db., 50 osob: FAR=0.0%,FRR=1.5% 100 osob: FAR=0.5%,FRR=2.0% | - |
| 21 | (Klonowski et al. 2018) | Sada příznaků, abs.hodnota z rozdílu příznaků + váhové koeficienty, 4 prsty, HGDB, FAR=0.00%, FRR=1.19% | 0.59 |
| 22 | (Barra et al. 2019) | Euklidovská vzdálenost, THID db. | 3.5 |
| 23 | (Barra et al. 2019) | LDA, geometrie ruky + množina parametrů rozhodnutelnosti, THID db. | 0.9 |
| 24 | (Barra et al. 2019) | Geometrie ruky, množina dalších parametrů „Intra/Inter-Class Variability“, LDA, 4 prsty, THID db. | 0.52 |
| 25 | Tato práce, A1 | Geometrie+topologie, 4 prsty, \mathcal{M}_{co}, S_{ea} THID db., 1280x960, IR%=100, E9 | 0.38 |
| 26 | Tato práce, A2 | Geometrie+topologie, 4 prsty, 4 kloubů, \mathcal{M}_{ea}, S_{ea} THID db., 1280x960, E9 | 0.48 |
| 27 | Tato práce, A3 | Geometry, 4 prsty, 4 kloubů, \mathcal{M}_{ea}, S_{ea} THID db., 1280x960, E9 | 1.92 |
| 28 | Tato práce, A4 | Geometrie+topologie, 4 prsty, 4 kloubů, \mathcal{M}_{co}, S_{ea} THID db., 640x480, E9 | 1.15 |
| 29 | Tato práce, A5 | Geometrie+topologie, 4 prsty, 4 kloubů, \mathcal{M}_{co}, S_{ea} THID db., 640x480, E10 | 1.83 |
| 30 | Tato práce, A6 | Geometry, 5 prstů, 5 kloubů, \mathcal{M}_{ea}, S_{ea} THID db., 640x480, E11 | 3.49 |
| 31 | Tato práce, A7 | Geometrie+topologie, 5 prstů, 6 kloubů, \mathcal{M}_{co}, S_{ea} THID db., 640x480, E8 | 1.82 |
| 32 | Tato práce, A8 | Geometrie+topologie, 4 prsty, \mathcal{M}_{co}, S_{ea} GPDS db., 1403x1021, E9 | 0.35 |
| 33 | Tato práce, A9 | Geometry, 5 prstů, 6 kloubů, \mathcal{M}_{ea}, S_{ea} GPDS db., 640x480, E3 | 0.63 |
| 34 | Tato práce, A10 | Geometry, 5 prstů, 5 kloubů, \mathcal{M}_{ea}, S_{ea} GPDS db., 640x480, E11 | 2.12 |
| 35 | Tato práce, A11 | Geometrie+topologie, 5 prstů, 4 kloubů, \mathcal{M}_{co}, S_{ea} GPDS db., 640x480, E9 | 0.64 |
| 36 | Tato práce, A12 | Geometrie+topologie, 4 prsty, \mathcal{M}_{co}, S_{ea} HGDB db., 1754x1275, E9 | 3.00 |



Obrázek (7.1).23 Různě dlouhé prsty u identické osoby na různých snímcích v databázi HGDB, na obrázku jsou vzorky z databáze HGDB: P32-01-M vs P32-05-S. Rozdíl délky je až 1cm. Rozdíl je také v tloušťce a tvaru u některých z 5ti prstů.

7.1.4.7 Přesnost algoritmu eaICP při klasifikaci identických snímků z databáze THID

Při experimentech byla také ověřována schopnost algoritmu **eaICP** dosáhnout hodnoty $\mathcal{E} = 0$ v případě, že $\mathcal{M} \equiv \mathcal{S}$. Celkem 104 vzájemných porovnání jednotlivých modelových kontur \mathcal{M} . Pracovní parametry $N_{pop}^{EPSDE} = 15$, $G_{en}^{EPSDE} = 200$, $L_p^{EPSDE} = 10$. Algoritmus dosahoval hodnoty $\mathcal{E} = 0$ již v rozmezí 70-90 generace pro všechny testované vzorky \mathcal{M} , \mathcal{S} a pro všechny hodnoty $\kappa_S \in (1,40)$ a to i při 1000x opakování pro libovolné dva vzorky, pro které platí, že $\mathcal{M} \equiv \mathcal{S}$. Prakticky to znamená $FAR = 0.0\%$, $FRR = 0.0\%$, $ERR = 0.0\%$. K chybám klasifikace nedochází ani při $\kappa_S = 40$ a **eaICP** vypočítá vždy $\mathcal{E} = 0$. Optimalizátor EPSDE vykazuje v tomto ohledu vynikající vlastnosti. Podobné, i když jen o málo horší vlastnosti má při porovnávání identických vzorků také jDE a JADE.

Velkou výhodou použitého optimalizátoru EPSDE je, že v průběhu experimentů nebylo pozorováno ani jediné fatální selhání konvergence. Hardwarová akcelerace s využitím maticových procesorů, PLA nebo grafických karet typu (4Q/2019) AMD FirePro S9300x2, NVidia TeslaK80, Xilinx Alveo, Cerebras apod. by samozřejmě přenesla navržený algoritmus **eaICP** do oblasti real-time aplikací i pro databáze většího rozsahu. Každý akcelerátor se hodí pro určitý druh oblasti umělé inteligence. Především karty Xilinx Alveo nebo multiprocesorové systémy Cerebras.net představují velkou přidanou hodnotu pro účely klasifikace s využitím evolučních algoritmů. A to buď samotné nebo ve spojení s akcelerátory NVidia apod.

7.1.5 Diskuze

V dnešní době existuje bezpočet různých metod navržených pro klasifikaci kontury lidské ruky. Každé navržené a publikované metodě pochopitelně přináší určitá přesnost a svým způsobem i jedinečnost. V Tab. (7.1).20 jsou zapsány některé z významných publikací, ve kterých bylo dosaženo vysoké úspěšnosti při klasifikaci kontur. Úspěšnost je definovaná hodnotou *EER* (Equal Error Rate). Nejlepší dosahované přesnosti se pohybují okolo $EER = 0.3 - 1.0\%$. Průměrné metody dosahují přesností okolo $EER = 1.1 - 7.0\%$. Existuje zde bohužel značná rozptýlenost co do vstupních dat, a proto je nesporně obtížné porovnat která z navrhovaných prací je ta nejlepší. Autoři také v drtivé většině nezveřejňují algoritmy, použité při klasifikaci, takže pro porovnání je nutné vytvořit kompletně celou práci někoho jiného, která často zabrala i několik let a pracovalo na ni mnoho výzkumníků. Hodnoty *EER* jsou tak jediným měřítkem pro vzájemné porovnání, což může být značně zavádějící. Mnoho autorů také využívá různé veřejně dostupné databáze, které však kvalitou snímků nejsou nejlepší, spíše naopak. Kvalita snímku přitom hraje klíčovou roli pro klasifikaci kontury ruky. Rozdíly v rádech procent při nevhodně nastavených parametrech předzpracování jsou běžné. V některých veřejně dostupných databázích se můžeme setkat i s tím, že snímky jsou záměrně deformované nebo jsou přebarvené určité množiny pixelů poblíž hypotetické kontury, aniž by se o tom autoři v příložené dokumentaci zmiňovali. Jedné a též osobě nemůže ruka na polovině z 10ti snímků v databázi narůst o 1cm a celkově se zúžit o 8mm za dobu scanování. To pak způsobuje pokles výsledné přesnosti při klasifikaci a je to zjevné v podstatě jen až při hodnocení výsledků klasifikace, která představuje např. klasifikaci 100000 snímků.

| Tabulka (7.1).20 Hodnoty <i>FAR</i> , <i>FRR</i> , <i>EER</i> dosahované různými autory a různými metodami | | | | | |
|--|-----------------------------|--|-------------------|---|---------------------------------|
| n. | Publikace | Metoda | Osob/snímků/prvků | Databáze | <i>EER</i> % |
| 1 | (Travieso et al. 2012) | DHMMK + SVM | 287 /10 / 25 | UST databáze, 60 DHMM stavů a 100 hran, které kódují body | 0.31 |
| 2 | (Sharma et al. 2015) | Tvar + Geometrie | 240/5/50+50+21 | IITD | 0.52 |
| 3 | (Sharma et al. 2015) | Tvar + Geometrie | 50/10/50+50+21 | (Sharma et al. 2015) soukromá databáze | 0.31 |
| 4 | (Khangad et al. 2011) | 2D+3D Geometrie ruky + 2D+3D Palmprint + Textura kloubků prstů | 177/5/- | soukromá databáze (GPDS 2020) | 0.22 |
| 5 | (Khangad et al. 2011) | 2D Geometrie ruky | 177/5/- | soukromá databáze (GPDS 2020) | 6.3 |
| 6 | (Santos-Sierra et al. 2011) | SVM, k-NN | 120+235+287/-/- | HTC, UST, IIT | 2.5,2.0,1.4 |
| 7 | (Kang & Wu 2014) | Použity jen 4 prsty, Fourierovy deskriptory | 638/1926/- | Bogazici University Hand Database (Bosphorus hand database) | 3.69, 2.73 |
| 8 | (Luque-Baena et al. 2013) | Genetický algoritmus a další informace | -/-/400+40 | GPDS, CASIA, IITD-TPD | 4.64 |
| 9 | (Morales et al. 2008) | SVM | -/-/- | soukromá databáze | 6.3 |
| 10 | (Michael et al. 2010) | Korelační koeficienty | 50/10/27 | soukromá databáze | 4.2 |
| 11 | (Bera & Bhattacharjee 2017) | k-NN, RF klasifikátor, 4 prsty, 5 prstů | 638/1926/9,12 | Bogazici University Hand Database (Bosphorus hand database) | 6.0, 8.0 |
| 12 | (Klonowski et al. 2018) | Sada příznaků, abs.hodnota z rozdílu příznaků + váhové koeficienty | 60/cca450/54 | Soukromá databáze | 0.59 FAR=0.00%, FRR=1.19% |

Navržená metoda **eaICP** dosahuje přesností v rozmezí $EER = 0.38 - 3.5\%$ což je zcela běžně dosahovaná přesnost metod, které představují state-of-the-art v současné době (2020).

7.1.6 Závěr

V předchozích odstavcích byl navržen algoritmus označený jako **eaICP**, který umožňuje identifikaci osob s využitím znalosti kontury lidské ruky a jehož složitost je $O(N)$. Nejlepší dosažená úspěšnost klasifikace je cca 99.61% pro databázi THID a 99.68% pro databázi GPDS. Koeficient EER pak THID: EER=0.38%, GPDS: EER=0.35% a HGDB: EER=3.00%. Bylo prokázáno, že značný vliv na výslednou přesnost klasifikace má kvalita modelové kontury a počet bodů (obecně entit), ze kterých se skládá. Dále pak vhodný výběr jednotlivých částí kontury ruky a také kloubů palce. Celý proces zpracování informace je silně závislý na moderních technologiích, a především na výpočetním potenciálu současných běžně dostupných procesorů. Při klasifikaci kontur je využito poznatků z mnoha oborů umělé inteligence, které v současné době odráží state-of-the-art v dotčených vědních oblastech. Cílem celé studie bylo zjistit, jestli je možné danou 7, 8 a 9ti dimenzionální optimalizační úlohu úspěšně řešit s využitím moderních evolučních optimalizátorů. Skupina vhodných optimalizátorů byla úspěšně nalezena. Algoritmus **eaICP** využívá nejmodernější klasifikační metody, které se běžně využívají v mnoha citlivých lékařských i vojenských oborech.

Navržený algoritmus (kapitola 7) pracuje s fotografií lidské ruky, která je položena na černou podložku nebo může být držena volně nad tmavou podložkou při běžném osvětlení bez nároků na nějakou fotografickou komoru. Lze použít jak snímky dlaně s prsty, tak i ruku otečenou na kameru hřbetem. V žádném případě se nejedná o ideální podmínky pro zpracování obrazu. To je také citelné na výsledcích klasifikace. Časté a naprosto nahodilé zastínění kontury na různých místech způsobuje, že není možné přesně odhadnout konturu ruky, protože barva stínu odpovídá barvě podložky. Výrazně lepších výsledků by bylo dosaženo, pokud by databáze fotografií byla pořízena s využitím nějaké fotografické komory s vyšším rozlišením. Předkládané výsledky ukazují, že klasifikace osob s využitím kontury ruky stále nepatří do kategorie primárních identifikačních metod a nemá takovou váhu jako např. identifikace s využitím lidské DNA, otisku prstu nebo jen obyčejný a všem dobře známý PIN kód. Je však dobře využitelná na krátkém časovém úseku s malými nároky na bezpečnost.

Na to, aby navržený algoritmus dobře fungoval je nutné, aby modelová kontura \mathcal{M} byla vytvořena z kontury ruky \mathcal{C} tak, aby byla delší, než kontura \mathcal{S} alespoň o 10 procent (viz vzorec 7.1.6) a dále aby kontura \mathcal{S} byla ořezaná v problematických místech mezi prsty, kde je proměnlivá struktura záhybů kůže. Kontura \mathcal{M} být ořezaná nesmí. Je možné také ořezat i menší části špiček prstů kontury \mathcal{S} a vyhnout se tak problémům s dorůstajícími nehty, ale v tomto případě může dojít k selhání ICP algoritmu. Pro dosažení co nejnižšího koeficientu EER je „vhodné“ používat jen 4 prsty bez palce a dále použít část dlaně mezi malíčkem a zápěstím. Klasifikace kontury palce je problematická z důvodu nepřesného určení kloubů z databáze THID při jednom jediném snímku, ze kterého se tvoří kontura \mathcal{M} . Metoda popsaná v této kapitole bohužel neposkytuje dobré výsledky bez definovaných fuzzy-heuristických pravidel. Použitá databáze THID má všechny snímky v rozlišení 640x480x15bpp (příp. 24bpp). Tyto obrázky bylo nutné převzorkovat na 1280x960x24bpp. Bez převzorkování by se nejlepší dosažená hodnota EER pohybovala okolo 1 procenta. Obecně se má za to, že metody, založené na Euklidovské metrice podávají výkon na hranici $EER \approx 3.5 - 5.0\%$, ale ne výrazně lepší; viz (Barra et al 2019) a mnoho dalších autorů. Vhodnou volbou klasifikovaných příznaků a fuzzy-heuristických pravidel je možné dosáhnout až na hodnotu $EER = 0.38\%$ nebo možná i nižší, což zaručuje stejnou nebo lepší přesnost v porovnání s nejlépešími klasifikátory, které jsou založené na metodách faktorové analýzy (lineární diskriminanční analýza, Fisherova analýza, faktorová analýza).

LITERATURA

- Abbass A.H. 2002. The Self-Adaptive Pareto Differential Evolution Algorithm. In: *Proceedings of the 2002 Congress on Evolutionary Computation*. vol. 1, pp. 831–836.
- Abaza A., Ross A., Hebert Ch., Harrison M.A., Nixon M.S. 2013. A Survey on Ear Biometric. *Journal ACM Computing Surveys*. vol. 45, no. 22, p.35.
- Alani H.A. 2014. Enhanced Geometric – Based Hand Recognition Using Neural Network. *International Journal of Scientific & Engineering Research*. vol. 5, no. 6, pp. 168-171.
- Allegion corp. 2020 [online]. Allegion. ©2020 [cit. 8.4.2020]. Dostupné z: <https://us.allegion.com/en/home/products/categories/biometrics.html>
- Amayeh G., Bebis G., Nicolescu M. 2008. Gender Classification from Hand Shape, In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. pp. 1-7.
- Amayeh G., Bebis G., Erol A., Nicolescu M. 2007. A Component-Based Approach to Hand Verification. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. pp. 1-8.
- Amor B.B., Oujj K., Ardabilian M., Chen L. 2005. 3D Face recognition by ICP-based shape matching. In: *ICMI*, p.6.
- Angadi S.A., Hatture M.S. 2016. Biometric Person Identification System: A Multimodal Approach Employing Spectral Graph Characteristics of Hand Geometry and Palmprint. *Intelligent Systems and Applications*. vol. 3, pp. 48-58. [cit. 8.4.2020] DOI: 10.5815/ijisa.2016.03.06
- Angadi S.A., Hatture M.S. 2018. Hand geometry based user identification using minimal edge connected hand image graph. *IET Computer Vision*. vol. 12, no. 5, pp. 744-752.
- Bača M., Grd P., Fotak T. 2012. Basic Principles and Trends in Hand Geometry and Hand Shape Biometrics. In: *New Trends and Developments in Biometrics*. IntechOpen, [cit. 8.4.2020] <http://dx.doi.org/10.5772/51912>
- Barricelli N.A. 1962. Numerical testing of evolution theories: Part I Theoretical introduction and basic tests. *Acta Biotheoretica*. vol. 16, no. 1-2, pp. 69–98.
- Bakshe R.C, Patil A.M. 2014. Hand Geometry Techniques: A Review. *International Journal of Modern Communication Technologies & Research*, vol. 2, no. 11, p.7.
- Barra S., Casanova M., Marsico M., Riccio D. 2014. Babies: Biometric authentication of newborn identities by means of ear signatures. In: *IEEE Workshop on Biometric measurements and system for security and medical application (BIOMS)*. pp. 1-7.
- Barra S., Casanova A., Narducci F., Ricciardi S. 2015. Ubiquitous iris recognition by means of mobile devices. *Pattern recognition Letter*, vol. 57, pp. 66-73.
- Barra S., Marsico M., Galdi C., Harry W. 2013. FAME: face authentication for mobile encounter. In: *IEEE Workshop on biometric measurements and systems for security and medical applications*. BioMS, [cit. 8.4.2020] DOI: 10.1109/BIOMS.2013.6656140.
- Barra S., Marsico M., Nappi M., Narducci F. 2017. MOHAB: Mobile Hand-based Biometric recognition. In: *International conference on green pervasive and cloud computing*. Springer, pp. 105-115.
- Barra S., Marsico M., Nappi M., Narducci F., Riccio D. 2019. A hand-based biometric system in visible mobile environments. *Information sciences*. vol. 479, pp. 472-485.
- Bartlett M.S., Lades H.M., Sejnowski T.J. 1998. Independent component representations for face recognition. In: *Conference on Human Vision and Electronic Imaging III*. San Jose, California, vol. 612, pp. 39-67.
- Bentley J.L. 1975. Multidimensional Binary Search Trees Used for Associative Searching. *Communications of the ACM*. vol. 18, no. 9, pp. 509-517.
- Bera A., Bhattacharjee D. 2017. Human Identification Using Selected Features From Finger Geometric Profiles. *IEEE Transactions on system, man and cybernetics: Systems*. pp. 1-15.
- Bergevin R., Soucy M., Gagnon H., Laurendeau D. 1996. Toward a general multi-view registration technique. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vol. 18, no. 5, pp. 540-547.

- Besl P.J., McKay H.D. 1992. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vol. 14, no. 2, pp. 239-256.
- Betelgeuse LLC, 2020 [online]. Betelgeuse LLC, C++ FAQ - How accurate floating point comparison is? ©2010-2015 [cit. 8.4.2020]. Dostupné z: http://www.softwareandfinance.com/Cpp/FAQ_Floating_Point.html
- Bhaganagare B.B., Harale A.D. 2017. Iris as biometrics for security system. In: *Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*. vol. 3, no. 2.
- Bhanu B. 2007. Ear Shape for Biometric Identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vol. 29, no. 8, pp. 1297-1308.
- Bharathi S., Sudhakar R. 2011. Hand Biometrics: An Overview. *International Journal of Automated Identification Technology*, vol. 3, no. 2, pp. 101-108.
- Blanz V., Vetter T. 2003. Face Recognition Based on Fitting a 3D Morphable Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vol. 25, no. 9, pp. 1063-1074.
- Brest J., Greiner S., Boškovič B., Mernik M., Žumer V. 2006. Selfadapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*. vol. 10, no. 6, pp. 646-657.
- Censi A. 2008. An ICP variant using a point-to-line metric. In: *IEEE International Conference on Robotics and Automation*. pp. 19-25. [cit. 8.4.2020] DOI: 10.1109/ROBOT.2008.4543181.
- Convision 2020 [online]. Convision. ©2020 [cit. 8.4.2020]. Dostupné z: <http://www.convision.cz/>
- Cortes J.M.R., Gil P.G., Perez G.S., Castro P.C. 2009. Shape-based hand recognition approach using the morphological pattern spectrum. *Journal of Electronic Imaging*. vol. 18, no. 1, pp. 0130121/1 - 0130121/6.
- Covavisaruch N., Prateepamornkul P., Ruchikachorn P., Taksaphan P. 2003. Personal Verification and Identification Using Hand Geometry. *ECTI Transaction on computer and information technology*. vol. 1, no.2, pp. 134-140.
- Dai D.Q., Yuen P.C. 2003. Regularized discriminant analysis and its application to face recognition. *Pattern Recognition*. vol. 36, no. 3, pp. 845-847.
- Dale M.P., Joshi M.A., Galiyawala H.J. 2012. A single sensor hand geometry and palm texture fusion for person identification. *International Journal of Computer Applications*. vol. 42, no. 7, pp. 11-16.
- Das S., Suganthan P.N. 2010. Differential Evolution: A Survey of the State-of-the-Art. *IEEE Transactionson Evolutionary Computation*, vol. 15, no 1., pp. 4-31.
- Dawson B., 2020 [online]. Comparing floating point numbers. ©2020 [cit. 8.4.2020]. Dostupné z: <http://www.cygnus-software.com/papers/comparingfloats/Comparing%20floating%20point%20numbers.htm>
- Dubuisson M.P., Jain A.K. 1994. A modified Hausdorff distance for object matching. In: *12th International Conference on Pattern Recognition*. pp. 566-568.
- Duta N. 2009. A survey of biometric technology based on hand shape. *Pattern recognition*, vol. 42, pp 2797-2806.
- Duta N., Jain A. K., Jolly M. P. 1999. Learning 2D Shape Models, In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR99*, p. 7.
- Dutagaci H., Sankur B., Yoruk E. 2008. Comparative analysis of global hand appearance-based person recognition. *Journal of Electronic Imaging*. vol. 17, no. 1, pp. 1-19.
- Eiben A.E., Hinterding R., Michalewicz Z. 1999. Parameter control in evolutionary algorithms. *IEEE Transaction on evolutionary computation*. vol. 3, no. 2, pp. 124-141.
- Farnia P., Ahmadian A., Behnam H., Dadashi N. 2014. Performance Evaluation of the Modified Iterative Closest Point Methods for Intra-operative Ultrasound and Pre-operative MR Image Registration of Brain. *Frontiers in Biomedical technologies*. vol. 1, no. 2, pp. 123-131.
- Faundez-Zanuy M., Elizondo D.A., Ferrer-Ballester M.A., Travieso-González C.M. 2016. Authentication of Individuals using Hand Geometry Biometrics: A Neural Network Approach. *Neural Processing Letters*. vol. 26, pp. 201-216.

- Faundez-Zanuy M., Mekyska J., Font-Aragones X. 2014. A new hand image database simultaneously acquired in visible, near-infrared and thermal spectrums. *Cognitive Computation*. vol. 6, no. 2, pp. 230–240.
- Ferrer M.A., Morales A., Travieso C.M., Alonso J.B. 2007. Low Cost Multimodal Biometric Identification System based on Hand Geometry, Palm and Finger Textures, In: *41st Annual IEEE International Carnahan Conference on Security Technology*. pp. 52-58.
- Ferrer M.A., Vargas F., Morales A. 2011. BiSpectral Contactless hand based biometric identification device. In: *2nd National Conference on Telecommunications*. pp. 1-6, [cit. 8.4.2020] DOI: 10.5772/18096.
- Fioretti W.H., Giordano A.J., Jacoby I.H. USA. Patent US3648240A, 7.3.1972.
- Fisher R. 1936. The use of multiple measures in taxonomic problems. *Annals of Eugenics*. vol. 7, no. 2, pp. 179–188.
- Fukunaga K. 1990. *Introduction to Statistical Pattern Recognition*. 2nd ed., New York Academic, ISBN 0-12-269851-7.
- Floudas Ch.A., Pardalos P. M. 2009. *Encyclopedia of Optimization*. Springer, 2nd ed., p. 4626, 7 volumes, ISBN 978-0-387-74759-0.
- Font-Aragones X., Faundez-Zanuy M., Mekyska J. 2013. Thermal hand image segmentation for biometric recognition. *IEEE Aerospace and Electronic Systems Magazine*. vol. 28, no. 6, pp. 4–14.
- Gemalto corp. 2020 [online]. Gemalto. ©2020 [cit. 8.4.2020]. Dostupné z: www.gemalto.com
- Gamperle R., Muller S.D., Koumoutsakos P. 2002. A parameter study for differential evolution. In: *WSEAS International Conference on Advances in Intelligent Systems*. pp. 293-298.
- Gasparini F., Schettini R. 2006. Skin segmentation using multiple thresholding. In: *Proceedings SPIE 6061*. pp. 60610F–60610F–8. [cit. 8.4.2020] DOI: 10.1117/12.647446.
- Greenspan M., Yurick M. 2003. Approximate k-d tree search for efficient ICP. In: *Fourth International Conference on 3-D Digital Imaging and Modeling*. pp. 442-448.
- Godin G., Rioux M., Baribeau R. 1994. Three-dimensional Registration Using Range and Intensity Information. In: *Proceedings of SPIE: Videometrics III*. vol. 2350.
- Goldberg D.E. 1987. Simple genetic algorithms and the minimal deceptive problem. In: *Genetic Algorithms and Simulated Annealing*, London, Pitman, pp. 74-88.
- Goldberg D.E. 1989. *Genetic Algorithms in search, optimization, and machine learning*. Addison-Wesley Publishing Corporation, ISBN-13: 978-0201157673.
- GPDS, 2020 [online]. GPDS ©2020 [cit. 8.4.2020]. Dostupné z: <http://www.gpds.ulpgc.es/>
- Gross R., Li Y., Sweeney L., Jiang X., Xu W., Yurovsky D. 2007. Robust Hand Geometry Measurements for Person Identification using Active Appearance Models. In: *First IEEE International Conference on Biometrics: Theory, Applications, and Systems*. vol. 1, pp. 6.
- Hansen N. 2006. The CMA Evolution Strategy: A Comparing Review. In: J.A. Lozano, P. Larrañaga, I. Inza and E. Bengoetxea (Eds.). *Towards a new evolutionary computation*. Advances in estimation of distribution algorithms. Springer, pp. 75-102.
- Hansen N., Ostermeier A., Gawelczyk A. 1995. On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation. In: *Proceedings of the Sixth International Conference on Genetic Algorithms*. L. Eshelman (Ed.), Pittsburgh, pp. 57-64.
- Hansen N., Ostermeier A. 1996. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In: *IEEE International Conference on Evolutionary Computation*. pp. 312-317.
- Hansen N., Ostermeier A. 1997. Convergence properties of evolution strategies with the derandomized covariance matrix adaptation: The $(\mu/\mu, \lambda)$ -ES. In: *EUFIT'97, 5th European Congress on Intelligent Techniques and Soft Computing*. pp. 650-654.
- Hansen N., Ostermeier A. 2001. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, vol. 9, no. 2, pp. 159-195.
- Hassanat A., Al-Awadia M., Btousha E., Al-Btousha A., Alhasanata E., Altarawnehb G. 2015. New Mobile phone and Webcam

- Hand Images Databases for Personal Authentication and Identification. *Procedia Manufacturing*. vol. 3, pp. 4060-4067.
- Haykin S. 1999. *Neural nets: A comprehensive foundation*. 2nd ed., Prentice Hall, p. 842. ISBN-13: 978-0132733502.
- Horn B.K.P. 1987. Closest form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*. vol. 4, no. 4, pp. 629-642.
- Holland J.H. 1962. Outline for a Logical Theory of Adaptive Systems. *Journal of the ACM*. vol. 9, no. 3, pp. 297-314.
- Holland J.H. 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, p. 232, ISBN: 9780262082136.
- Holland J.H. 1992. *Adaptation in natural and artificial systems: an introductory analysis with application to biology, control, and artificial intelligence*. Cambridge, MIT Press, p. 211, ISBN: 9780262275552.
- Charette R. 2010. Looking at Your Ears: The New Biometric Security Scan?. [online]. *IEEE Spectrum Journal 2010*. [cit. 8.4.2020]. Dostupné z: <https://spectrum.ieee.org/riskfactor/computing/it/looking-at-your-ears>
- Charfi N. 2017. Biometric recognition based on hand shape and palmprint modalities. [online]. *Image Processing*. [cit. 8.4.2020]. Ecole nationale supérieure Mines-Télécom Atlantique. Dostupné z: <https://tel.archives-ouvertes.fr/tel-01781354/document>
- Chauhan S., Arora A.S., Kaul A. 2010. A survey of emerging biometric modalities. *Procedia Computer science*. vol. 2, pp. 213-218.
- Chen Y., Medioni G. 1991. Object modeling by Registration of Multiple Range Images. In: *IEEE International Conference on Robotics and Automation*. pp. 2724-2729.
- Chen Y., Medioni G. 1992. Object modeling by registration of multiple range images. *Image and Vision Computing*. vol. 10, no. 3, pp. 145-155.
- Jain A.K., Duta N. 1999. Deformable matching of hand shapes for user verification. In: *IEEE International Conference on Image Processing*. pp. 857-861.
- Jain A.K., Ross A., Pankanti S. 1999. A Prototype Hand Geometry-based Verification System. In: *2nd International Conference on Audio and Video-based Biometric Person Authentication*. pp. 166-171.
- Jain A.K., Ross A., Prabhakar S. 2000. An Introduction to Biometric Recognition. *IEEE Transactions on circuits and systems for video technology*. vol. 14, no. 1, pp. 4-20.
- Jeught S.V., Buytaert J., Dirckx J. 2012. Real-time geometric lens distortion correction using a graphics processing unit. *Optical engineering*. vol. 51, pp. 1-5.
- Jeyakumar G., Shanmugavelayutham C. 2011. Experimental Study on Recent Advances in Differential Evolution Algorithm. *International Journal of Applied Evolutionary Computation*. vol. 2, no. 2, pp. 58-81.
- Jost T., Hügli H. 2002. Fast ICP Algorithms for Shape Registration. In: *Joint Pattern Recognition Symposium*. pp. 91-99.
- Kang B.J., Park K.R. 2010. Multimodal biometric method based on vein and geometry of a single finger. *IET Computer Vision*. vol. 4, no. 3, pp. 209-217.
- Kang W., Wu Q. 2014. Pose-invariant hand shape recognition based on finger geometry. *IEEE Transactions on System Man and Cybernetics Systems*. vol. 44, no 11, pp. 1510-1521.
- Kanhangad V., Kumar A., Zhang D. 2009. Combining 2D and 3D Hand Geometry Features for Biometric Verification. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. pp. 39-44.
- Kennedy J., Eberhart R.C. 1995. Particle swarm optimization. In: *IEEE International Conference on Neural Networks*. vol. 4, pp. 1942-1948.
- Keys R. 1981. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*. vol. 29, no. 6, pp. 1153-1160.
- Kimichi Y.M. 2020. [online]. Kimichi Y.M. ©2005-2020 [cit. 20.5.2020]. Dostupné z: <http://www.cs.technion.ac.il/users/yechiel/c++-faq/floating-point-arith.html>

- Klonowski M., Plata M., Syga P. 2018. User authorization based on hand geometry without special equipment. *Pattern Recognition*. vol. 73, pp. 189-201.
- Kolekar M.H., Kumar V. 2017. *Biomedical Signal and Image Processing in Patient Care*. IGI Global ed., p. 312, ISBN13: 9781522528296.
- Kukula E., Elliott S. 2006. Implementation of hand geometry: an analysis of user perspectives and system performance. *IEEE Aerospace and Electronic System Magazine*. vol. 21, no. 3, pp. 3–9.
- Kumar A., 2020 [online]. Kumar A., Hand 3D Database ©2020 [cit. 8.4.2020]. Dostupné z: <https://www4.comp.polyu.edu.hk/~csajaykt/>
- Kumar A., Ravikanth Ch. 2009. Personal Authentication using Finger Knuckle Surface. *IEEE Transactions on Information Forensics and Security*. vol. 4, no. 1, pp. 98-110.
- Kumar A., Wong D.C.M., Shen H.C., Jain A.K. 2003. Personal Verification using Palmprint and Hand Geometry Biometric. In: *International Conference on Audio- and Video-Based Biometric Person Authentication*. pp. 668-678.
- Liliana D.Y., Utaminingsih E.T. 2012. The combination of palm print and hand geometry for biometrics palm recognition. *International Journal of Video & Image Processing and Network Security*. vol. 12, no. 1.
- Liu J., Lampinen J. 2002. On setting the control parameter of the differential evolution method. In: *8th International Conference Soft Computing*. pp. 11-18.
- Liu-Jimenez J., Sanchez-Reillo R., Fernandez-Saavedra B. 2011. Iris Biometrics for Embedded Systems, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. vol 19, no. 2, pp. 274-282.
- Low K.L. 2004. Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration. *Technical Report TR04-004*, Department of Computer Science, University of North Carolina, p.3.
- Lu F., Milios E. 1994. Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. pp. 935–938.
- Lu F., Milios E. 1997a. Robot pose estimation in unknown environments by matching 2D range scans. *Journal of Intelligent and Robotics Systems*. vol. 18, no. 3, pp. 249–275.
- Lu F., Milios E. 1997b. Globally consistent range scan alignment for environment. *Autonomous Robots*. vol. 4, pp. 333-349.
- Luque-Baena R.M., Elizondo D., López-Rubio E., Palomo E.J., Watson T. 2013. Assessment of geometric features for individual identification and verification in biometric hand systems. *Expert Systems and applications*. vol. 40, no. 9, pp. 3580–3594.
- Maier-Hein L., Franz A.M., Santos T.R., Schmidt M., Fangerau M., Meinzer H.P., Fitzpatrick J.M. 2012 Convergent Iterative Closest-Point Algorithm to Accomodate Anisotropic and Inhomogenous Localization Error. *IEEE Transactions on pattern analysis and machine intelligence*. vol. 34, no. 8, pp. 1520-1532.
- Mallipeddi R., Suganthan P.N. 2010. Differential Evolution Algorithm with Ensemble of Parameters and Mutation and Crossover Strategies. In: *International Conference on Swarm, Evolutionary, and Memetic Computing SEMCCO 2010*. pp. 71-78.
- Masuda, T., Sakaue, K., and Yokoya, N. 1996. Registration and Integration of Multiple Range Images for 3-D Model Construction. In: *13th International Conference on Pattern Recognition*. vol. 1., pp. 879-883.
- Melo R., Falcao G., Barreto J.P. 2012. Real-time HD image distortion correction in heterogeneous parallel computing systems using efficient memory access patterns. *Journal of real-time image processing*. vol. 11, no. 1, pp. 83-91.
- Metropolis N., Ulam S. 1949. The Monte Carlo method. *Journal of the American Statistical Association*. vol. 44, pp. 335-341.
- Michael, G.K.O., Connie, T., Hoe L.S., Jin A.T.B. 2010. Locating Geometrical Descriptors for Hand Biometrics in a Contactless Environment. In: *International Symposium in Information Technology*. vol. 1, pp. 1–6.
- Minguez J, Lamiroux F., Montesano L. 2006. Metric-based scan matching algorithms for mobile robot displacement estimation, In: *IEEE International Conference on Robotics and Automation*. pp. 3557-3563.
- Morales A., Ferrer M.A., Díaz F., Alonso J.B., Travieso C.M. 2008. Contact-free hand biometric system for real environments. In: *16th European signal processing conference*. Lausanne, pp. 1-5.

- Morales A., Ferrer M.A., Kumar A. 2010. Improved palmprint authentication using contactless imaging. In: *Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. Washington, pp. 1-6.
- Moravec J. 2012. Cascaded Evolutionary Estimator for Robot Localization. *International Journal of Applied Evolutionary Computation*. vol. 3, no. 3, pp. 33-61.
- Moravec J. 2015. A Comparative Study: L1-norm vs. L2-norm, point-to-point vs. point-to-line Metric. Evolutionary Computation vs. Gradient Search. *Applied Artificial Intelligence*. Special edition on robotics research, vol. 29, no. 2, pp. 164–210.
- Moravec J. 2020a. Hand Contour Classification Using Differential Evolution Algorithm with Ensemble of Parameters and Mutation and Crossover. *Information Technology and Control*. vol. 49, no. 1, pp. 55-79. [cit. 8.4.2020] DOI: <https://doi.org/10.5755/j01.itc.49.1.24140>.
- Moravec J., 2020b [online]. Handwork. *Handwork.4fan.cz* ©2014-2020 [cit. 8.4.2020]. Dostupné z: <http://handwork.4fan.cz/>
- Moravec J., Hub M. 2015. Camera Calibration Using Direct Mapping and Adaptive Metaheuristic. *International Journal of BioScience and BioTechnology*. vol. 7, no. 3, pp. 111-120.
- Moravec J., Hub M. 2016. Automatic Correction of Barrel Distorted Images Using Cascaded Evolutionary Estimator. *Journal Information Sciences*. vol. 366, pp. 70-98.
- Moravec J., Pošík P. 2014a. A Comparative Study: The Effect of the Perturbation Vector Type in the Differential Evolution Algorithm on the Accuracy of Robot Pose and Heading Estimation. *Evolutionary Intelligence Journal*. vol. 6, no. 3, pp. 171-191.
- Moravec J., Pošík P. 2014b. Global Robot Localization Under Noise Stress Utilizing EA Methods and Semi-semantic Classification of a Known Environment. *Applied Artificial Intelligence*. vol. 28, no. 4, pp. 360–417.
- Neri F., Tirronen V. 2010. Recent advances in differential evolution: a survey and experimental analysis. *Artificial Intelligence Review*. vol. 33, no. 1-2, pp. 61-106.
- Nesrine Ch. 2017. *Biometric recognition based on hand schape and palmprint modalities*. Ecole nationale supérieure Mines-Télécom Atlantique, PhD Thesis.
- Nguyen L.T., Thanh C.D., Ba T.N., Viet C.T., Thanh H.L. 2013. Contour Based Hand Gesture Recognition Using Depth Data. *Advanced Science and Technology Letters*. vol.29, pp.60-65.
- Oracle corp. 2020 [online]. What Every Computer Scientist Should Know About Floating-Point Arithmetic. ©1991 [cit. 8.4.2020]. Dostupné z: http://download.oracle.com/docs/cd/E19957-01/806-3568/ngc_goldberg.html
- Pardalos P.M. 2020 [online]. Dr. Panos M. Pardalos ©2020 [cit. 20.5.2020]. Dostupné z: <http://www.ise.ufl.edu/pardalos>
- Park G., Kim S. 2013. Hand Biometric Recognition Based on Fused Hand Geometry and Vascular Patterns. *Sensors*, vol. 28, no. 3, pp. 2895-2910.
- Park S.Y., Subbarao M. 2003. An accurate and fast point-to-plane registration technique. *Pattern Recognition Letters*. vol. 24, pp. 2967–2976.
- Parker J.R. 2010. *Algorithms for Image Processing and Computer Vision*. 2nd Edition, Wiley, ISBN: 978-0-470-64385-3.
- Pavešić N., Ribarič S., Ribarič D. 2004. Personal authentication using hand-geometry and palmprint features – the state of the art. Corpus ID: 13966389.
- Pavlidis T. 1982. *Algorithms for Graphics and Image Processing*. Springer, p. 416, ISBN 978-3-642-93208-3.
- Pfister S., Kriechbaum K., Roumeliotis S., Burdick J. 2002. Weighted range sensor matching algorithms for mobile robot displacement estimation. In: *IEEE International conference on Robotics and Automation*, vol. 2, pp. 1667-1674.
- Pomerleau F., Colas F., Siegwart R., Magnenat S. 2013. Comparing ICP Variants on Real-World Data Sets. *Autonomous Robos*. vol. 34, no. 3, pp 133-148.
- Pošík P., Huyer W., Pál L. 2012. A Comparison of Global Search Algorithms for Continuous Black Box Optimization. *Evolutionary Computation*. vol. 20, pp. 509-541.
- Pošík P., Kubalík J. 2012. Experimental Comparison of Six Population-Based Algorithms for Continuous Black Box

- Optimization. *Evolutionary Computation*. vol. 20, pp. 483-508.
- Pottmann H., Huang Q.X., Yang Y.L., Hu S.M. 2006. Geometry and convergence analysis of algorithms for registration of 3D shapes. *International Journal on Computer Vision*. vol. 67, no. 3, pp. 277-296.
- Price K. 1996. Differential Evolution: A Fast and Simple Numerical Optimizer. In: *NAFIPS'96*. pp. 524–527.
- Price K., Corne D., Dorigo M., Glover F. 1999. An Introduction to Differential Evolution, *New ideas in optimization*, pp. 79-108.
- Price K., Storn R. 1996, Minimizing the Real Functions of the ICEC'96 contest by Differential Evolution. In: *IEEE International Conference on Evolutionary Computation*. pp. 842–844.
- Price K., Storn R. 1997. Differential Evolution – A simple evolution strategy for fast optimization. *Dr. Dobb's Journal*. vol. 22, no. 4, pp.18-24 and 78.
- Qin A.K., Suganthan P.N. 2005. Self-adaptive differential evolution algorithm for numerical optimization. In: *IEEE Congress on Evolutionary Computation*. vol. 2, pp. 1785-1791.
- Rechenberg I. 1973. Evolutionsstrategies: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. *Feddes repertorium*. Frommann-Holzboog Eds., Stuttgart, Germany, vol. 86, no. 5, pp. 337-337.
- Rechenberg I. 1994. Evolutionsstrategie '94, Frommann-Holzboog Ed., Stuttgart, ISBN 13: 9783772816420.
- Ribaric S., Fratric I. 2005. An online biometric authentication system based on eigenfingers and finger geometry. In: *13th European Signal Processing Conference*. pp.1-4.
- Ribarič S., Ribarič D., Pavešič N. 2003. A multimodal biometric-based user-identification system for internet use. In: *IEEE Proceedings - Vision, Image and Signal Processing*. vol. 150, no. 6, pp. 409-416.
- Richard E.H. 1971. Hand ID system. USA. Patent US3576537A. 27.4.1971.
- Rodrigues M., Fisher R., Liu Y., 2002. Special issue on registration and fusion of range images. *Computer Vision and Image Understanding*. vol. 87, pp. 1–131.
- Ross A., Jain A.K. 2007. Human Recognition Using Biometrics: An Overview. *Anal. of Telecommunications*. vol. 62, no. 1-2, pp. 11-35.
- Rusinkiewicz S., Levoy M. 2001. Efficient variants of the ICP algorithm. In: *IEEE Third International Conference on 3-D Digital Imaging and Modeling*. p. 8.
- Samiya S., Abderrahmane B., Abdelkader B. 2012. Recognition of individuals based on hand geometry. *Uncertainty Modeling in Knowledge Engineering and Decision Making*. pp. 1023-1029.
- Sanches-Reillo S.R., Sanches-Avila S.C., Gonzales-Marcos A. 2000. Biometric identification through hand geometry measurement. *IEEE Transactions on pattern analysis and machine intelligence*. vol. 22, no. 10., pp. 1168-1171.
- Santos-Sierra A., Sánchez-Ávila C., Pozo G.B., Guerra-Casanova J. 2011. Unconstrained and Contactless Hand Geometry Biometrics. *Sensors*. vol. 11, pp. 10143-10164.
- Santos-Sierra A., Casanova J., Avila C., Vera V. 2009. Silhouette-based hand recognition on mobile devices. In: *International Carnahan Conference on Security Technology*. pp. 160–166.
- Segal A.V., Haehnel D., Thrun S. 2009. Generalized-ICP. *Robotics: Science and Systems*, p. 8.
- Sharma S., Dubey S.R., Singh S.K., Saxena R., Singh R.K. 2015. Identity verification using shape and geometry of human hands. *Expert Systems and applications*. vol. 42, no. 2, pp. 821–832.
- Schwefel H.P. 1995. Evolution and Optimum Seeking. JohnWiley&Son Eds., New York, USA, p. 456, ISBN:978-0-471-57148-3.
- Smit S.K., Eiben A.E. 2009. Comparing Parameter Tuning Methods for Evolutionary Algorithms. In: *IEEE Congress on Evolutionary Computation*. pp. 399-406.
- Smit S.K., Eiben A.E. 2010a. Parameter Tuning of Evolutionary Algorithms: Generalist vs. Specialist. *Applications of Evolutionary Computation*. vol. 6024, pp. 542-551.

- Smit S.K., Eiben A.E. 2010b. Parameter Tuning for Configuring and Analyzing Evolutionary Algorithms. *Swarm and Evolutionary Computation*. vol. 1, no. 1, pp. 19-31.
- Splab 2020. [online]. SPLAB, Brno University of Technology, ©2011 [cit. 8.4.2020]. Dostupné z: <http://splab.cz/en/download/databaze/tecnocampus-hand-image-database>
- Stockman G., Shapiro L. 2001. *Computer Vision*. Prentice Hall, p. 608, ISBN-13: 978-0130307965.
- Storn R. 1996. On the Usage of Differential Evolution for Function Optimization. In: *NAFIPS'96*. pp. 519–523.
- Storn R., Price K. 1997. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization. *Journal of Global Optimization*. no. 11, pp. 341–359.
- Sural S., Qian G., Pramanik S. 2002. Segmentation and histogram generation using the HSV color space for image retrieval. In: *IEEE International Conference on Image Processing*. pp. 589-592.
- Sun Microsystems 2020 [online]. Sun Microsystems, Inc. ©2012 [cit. 20.5.2020]. Dostupné z: http://download.oracle.com/docs/cd/E19957-01/806-3568/ncg_goldberg.html
- Technion, 2020 [online]. Why don't floating point comparisons work? ©2020 [cit. 8.4.2020]. Dostupné z: <http://www.cs.technion.ac.il/users/yechiel/c++-faq/floating-point-arith.html>
- Teo J. 2005. Exploring dynamic self-adaptive populations in differential evolution. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*. vol. 10, pp. 673–686.
- Tong C., Kamata S., Ahrary A. 2009. 3D face recognition based on fast feature detection and non-rigid iterative Closest Point. In: *IEEE International Conference on Intelligent Computing and Intelligent Systems*. pp. 509-512.
- Travieso C.M., Alonso J.B., David S., Ferrer M.A. 2004. Optimization of a biometric system identification by hand geometry. *Complex systems intelligence and modern technological applications*. Cherbourg, France, pp. 581-586.
- Travieso C.M., Briceo J.C., Alonso J.B. 2012. Transformation of hand-shape features for a biometric identification approach. *Sensors*. vol. 12, no. 1, pp. 987–1001.
- Turk G., Levoy M. 1994. Zippered Polygon Meshes from Range Images Computer Graphics Proceedings. In: *Annual Conference Series, ACM SIGGRAPH*. pp. 311-318.
- Urfalioglu O. 2004. Robust Estimation of Camera Rotation, Translation and Focal Length at High Outlier Rates. In: *1st Canadian Conference on Computer and Robot Vision*. pp. 464-471.
- Varchol P., Levicky D. 2007. Using of hand geometry in biometric security systems. *Radioengineering*. vol. 16, no. 4, pp. 82–86.
- Wikipedia, 2020a [online]. Wikipedia ©2020 [cit. 8.4.2020]. Dostupné z: https://en.wikipedia.org/wiki/Linear_regression
- Wikipedia, 2020b [online]. Wikipedia ©2020 [cit. 8.4.2020]. Dostupné z: <https://en.wikipedia.org/wiki/CMA-ES>
- Wikipedia, 2020c [online]. Wikipedia ©2020 [cit. 8.4.2020]. Dostupné z: https://en.wikipedia.org/wiki/Bicubic_interpolation
- Villegas O.O.V, Domínguez H.J.O, Sánchez V.G.C., Maynez L.O., Orozco H.M. 2011. Biometric Human Identification of Hand Geometry Features Using Discrete Wavelet Transform, Discrete Wavelet Transforms - Biomedical Applications. Hannu Olkkonen, IntechOpen, [cit. 8.4.2020] DOI: 10.5772/19508.
- Wildes R.P. 1997. Iris recognition: an emerging biometric technology. *Proceedings of the IEEE*. vol. 85, no. 9, pp. 1348-1363.
- Woodard D.L., Flynn P.J. 2005. Finger surface as a biometric identifier. *Computer Vision and Image Understanding*. vol. 100, pp. 357–384.
- Wong A.L.N., Shi P. 2002. Peg-Free Hand Geometry Recognition Using Hierarchical Geometry and Shape Matching. In: *IAPR Workshop on Machine Vision Applications*. pp. 281-284.
- Xiong W., Xu Ch., Ong S.H. 2005. Peg-Free human shape analysis and recognition. In: *IEEE International Conference on Acoustics Speech, and Signal Processing*. vol. 2, pp. ii/77–ii/80.
- Yan X., Su X.G. 2009. *Linear Regression Analysis: Theory and Computing*. World Scientific Publishing Company, 1.ed., p. 328, ISBN-13: 978-9812834102.

- Yang Z., Tang K., Yao X. 2008. Self-adaptive differential evolution with Neighborhood search. In: *Proceedings of the 2008 Congress on Evolutionary computation*, IEEE Press, Hong Kong, pp. 1110-1116.
- Yörük E., Konukoglu E., Sankur B., Darbon J. 2006a. Shape Based Hand Recognition. *IEEE Transactions on Image Processing*. vol. 15, no. 7, pp. 1803-1815.
- Yörük E., Dutagaci H., Sankur B. 2006b. Hand biometrics. *Image and Vision Computing*. vol. 24, pp. 483–497.
- Zhang J., Sanderson A.C. 2009. JADE: Adaptive Differential Evolution with Optional External Archive. *IEEE Transactions on Evolutionary Computation*. vol. 13, no. 5, pp. 945–958.
- Zhu J., Xu S., Hou J., Li Y., Wang H. 2017. Effective scaling registration approach by imposing the emphasis on the scale factor. *Computer Vision and Pattern Recognition*, vol. 54, no. 7, pp. 422-424.

7.2 Evoluční estimátor pro klasifikaci kontury ruky využívající množinu různých geometrických primitiv

Biometrika patří dnes mezi všeobecně uznávané obory. Má široké uplatnění především v oblastech zajišťování bezpečnosti, ale také v lékařství, zábavním průmyslu, forenzních vědách apod. Běžně se můžeme setkat s biometrickými systémy, které využívají DNA (Zayaraz et al. 2009), otisk prstu (Borra et al. 2016), strukturu dlaně (Ramteke & Hatkar 2013), konturu ucha (Kumar et al. 2011), strukturu obličeje (Hemery et al. 2008; Zhi et al. 2014), strukturu oční duhovky (Daugman 2004), konturu ruky (Luque-Baena et al. 2013; Barra et al. 2019). Obecně neexistuje nějaké pravidlo, které by omezovalo použití biometrických příznaků. Spíše naopak, je možné se setkat s kombinacemi různých biometrických ukazatelů např. kontura ruky a otisk struktury dlaně (Charfi 2017) nebo kontura ruky a struktura krevního řečiště (Park & Kim 2013). Základní přehled je možné nalézt v (Chauhan et al. 2010; Bakshe & Patil 2014)

7.2.1 Úvod

Existuje bezpočet metod a nástrojů pro klasifikaci kontury ruky. Některé navržené metody vyžadují ustavení ruky v adjustázním přípravu (Allegion 2020a,b; Jain et al. 1999; Sanchez-Reillo et al. 2000; Park & Kim 2013). Některé metody naopak takový přípravek vůbec nevyužívají (Travesio et al. 2004; Yörük et al. 2006a,b; Xiong et al. 2005). Adjustázní přípravek může být i sklopný (Moravec & Hub 2016). Některé metody používají jen fotografii ruky, která je umístěna volně v prostoru (Santos-Sierra et al. 2009; Ferrer et al. 2011; Barra et al. 2019). Velmi oblíbená metoda je využití klasického stolního scanneru (Travesio et al. 2004; Ferrer et al. 2007). Některé metody jsou specializované jen pro použití na mobilních telefonech (Santos-Sierra et al. 2009; Barra et al. 2019). V praxi je možné setkat se s přístroji, které provádí identifikaci bez využití kontury palce (Allegion 2020a,b). Některá rámcová pravidla pro rozpoznávání geometrie ruky jsou dnes standardizovaná podle normy ANSI/INCITS 396-2005 – „Hand Geometry Format for Data Interchange“ a odpovídající Evropská norma ISO/IEC 19794-10/2007 (ISO 2020) byla schválena již v roce 2007, ale popisuje pouze základní aspekty bez jakýchkoliv detailů. Obecně se prozatím považuje biometrická metoda využívající geometrii ruky za méně spolehlivou v porovnání např. s testem DNA. Detailní přehledovou studii o aktuálních možnostech v dané oblasti lze nalézt v (Duta 2009; Bharathi & Sudhakar 2011; Bakshe & Patil 2014).

Pro účely experimentů byla využita databáze GPDS150 (GPDS) (Travesio et al. 2004; Ferrer et al. 2007; GPDS 2020) The „Grupo de Procesado Digital de la señal“, GPDS (Digital Signal Processing Group) with DPDS (División de Procesado Digital de la Señal) from „Instituto para el Desarrollo Tecnológico y la Innovación en Comunicaciones IDeTIC“, both are active at the „University of Las Palmas de Gran Canaria“.

Nyní bude krátce rozepsána sekce zajímavých publikací a poté bude popsán použitý evoluční self-adaptive optimalizátor EPSDE (Mallipeddi & Suganthan 2010), dále algoritmus ICP (Besl & McKay 1992), který je využit pro zarovnání kontur rukou. Další kapitolou je popis navrženého algoritmu, který je schopen efektivní klasifikace kontur rukou. Poslední částí je pak sekce experimentálních výsledků a samozřejmě také závěrečné zhodnocení. Tato kapitola obsahuje především komparativní studii několika různých uspořádání základního estimátoru, kde vstupní informací je nejen kontura ruky, ale také další podpůrné informace – příznaky - extrahované z klasifikovaného snímku ruky.

7.2.2 Související publikace

7.2.2.1 Obecký pohled na problematiku

Vývoj v oblasti autentifikace osob se rozvíjí značným tempem. Některé z metod jsou dnes již běžně komerčně dostupné např. biometrický scanner fi. Allegion (Allegion 2020b), který využívá jen 4 prsty ruky bez palce a adjustážní kolíky (pegs). Nevýhodou kolíků je to, že deformují konturu prstů viz. (Jain et al. 1999; Covavisaruch et al. 2003) a některé části prstů se pak musí zrekonstruovat. Výhodou použití adjustážních kolíků je to, že umožňují jednoznačně ustavit ruku, která se dotýká podložky tak, aby roztažení prstů bylo optimální pro další zpracování. Nedochozí také k deformaci celé kontury tím, že se ruka nevhodně natočí v 3D prostoru jako je tomu u (Santos-Sierra et al. 2011; Barra et al. 2019), kde je možné naklonit ruku jakýmkoliv směrem vůči kameře a tím např. zkrátit konturu prstů nebo celé ruky. Výhodou metod jako je (Barra et al. 2019) je hygienická čistota, protože identifikovaná osoba se nedotýká podložky. Pořizování snímků ruky může být provedeno tak, že je fotografován hřbet ruky (Jain et al. 1999; Faundez-Zanuy et al. 2016) nebo je možné použít i klasický stolní scanner (Yörük et al. 2006a; Ramirez-Cortes et al. 2009). (Yörük et al. 2006a) publikovali zajímavý článek, ve kterém popsali metodu, která umožňuje identifikaci osob podle kontury ruky. Pro získání snímků rukou je využit klasický kancelářský scanner s rozlišením 45dpi. Je skenována dlaň a prsty. Pro účely experimentů získali autoři celkem 1374 snímků od 458 osob. Pro klasifikaci je využit histogram a algoritmus „Independent Component Analysis“ (ICA) (Bartlett et al. 1998). Kontury jednotlivých prstů jsou extrahovány s využitím Radial Distance Diagramu (RDD) a při výpočtu se pak pohybují okolo kloubů. Pozice kloubů jsou určeny odhadem. Základní metrikou je modifikovaná Hausdorffova vzdálenost (Dubuisson & Jain 1994). Výsledky presentované autory ukazují na dosaženou přesnost okolo 98 procent.

Při návrhu zde presentovaného algoritmu byly využity některé dříve publikované poznatky a metody. Jednak je velmi výhodné navrhnout metodu, která nevyžaduje adjustážní kolíčky (pegs), protože kolíčky deformují konturu ruky a korekce kontury v místech, kde jsou kolíčky je poměrně náročná. Další výhodou je schopnost algoritmu pracovat jak s fotografií, na které je ruka otočená dlaní tak i s fotografií, kde je ruka natočená hřbetem. Ruka také může, ale nemusí být položená na nějaké podložce obvykle v biometrickém scanneru nebo na klasickém stolním scanneru. Jediné omezení, které navržený algoritmus má je na homogenní osvětlení a minimální rozlišení snímku přibližně 0.3 mega pixelu.

7.2.2.2 Evoluční optimalizátor EPSDE

Algoritmus EPSDE je detailně popsán v kapitole „Evoluční algoritmy“ a proto jen krátce zopakujeme jeho základní charakteristiku a souvislosti, vztahující se k řešené problematice v této kapitole.

V roce 1996 publikovali pánové Rainer Storn a Kenth Price algoritmus nazvaný Diferenciální Evoluce (DE) (Price 1996; Price & Storn 1996; Storn 2020; Wikipedia 2020b). DE spadá do kategorie tzv. stochastických nelineárních metaheuristických populačních algoritmů. Optimální řešení daného problému je hledáno iterativním opakováním základního schématu algoritmu DE. Nevýhodou DE je to, že nezaručuje nalezení optimálního řešení. Tato poměrně paradoxní vlastnost je však stejná u všech metaheuristických optimalizátorů a z obecného pohledu i u všech evolučních algoritmů. DE se hodí pro řešení jak nízko-dimenzionálních tak i více-dimenzionálních úloh. DE je také vhodným nástrojem pro řešení úloh globální optimalizace (Price & Storn 1996, 1997; Storn & Price 1997; Montes et al. 2006; Price et al. 2005). Algoritmus DE vznikl při společné práci autorů na úloze s Chebychevovými polynomy (the Chebychev Polynomial Fitting Problem) (CPFP) (Storn 2020). DE se velmi rychle stala populární pro svoji výkonnost, spolehlivost a především proto, že v mnoha úlohách převyšovala výkonost do té doby běžně používaných optimalizátorů jako je metoda MonteCarlo (MC) (Metropolis & Ulam 1949), Jednoduchý Genetický Algoritmus (Simple Genetic Algorithm) (SGA) (Holland 1962,

1975, 1992), Particle Swarm Optimization (PSO) (Kennedy & Eberhart 1995) nebo Evoluční Strategie (ES) (Rechenberg 1973, 1994). Po počátečním nadšení z nově objeveného optimalizátoru však přišlo velké zklamání. Zjistilo se, že algoritmus DE vyžaduje poměrně precizní nastavení pracovních parametrů, které se v mnoha úlohách musí měnit při běhu algoritmu viz (Eiben et al. 1999; Gamperle et al. 2002; Liu & Lampinen 2002; Smith & Eiben 2009; Smith & Eiben 2010a,b; Das & Suganthan 2010). Stoprocentně se tak vyplnit krutý, ale pravdivý „No Free Lunch Theorem“ (Wolpert & Macready 1997), který říká, že v oblasti optimalizačních algoritmů nepřichází nic zcela zdarma. Na konci 90. let minulého století tak nastává druhá etapa vývoje v oblasti DE, kdy se mnoho vědců snažilo najít řešení tohoto problému ve smyslu automatického nastavování pracovních parametrů DE. Druhá etapa vývoje DE, která trvá dodnes přinesla mnoho zajímavých optimalizátorů odvozených od původního algoritmu DE, které svojí výkonností v mnoha ohledech převyšují původní DE. Viz přehledové studie (Montes et al. 2006; Neri & Tironen 2010; Das et al. 2016). V počátečních letech výzkumu se pro vylepšení vlastností DE velmi často využívala tzv. hybridizace, což je přímé spojení dvou optimalizačních algoritmů např. (Zhang & Xie 2003; Vesterstrom & Thomsen 2004). Hybridizace z obecného pohledu bohužel nepřinesla očekávané výrazné zlepšení vlastností DE, spíše byla zklamáním. Současně se tak vývoj začal ubírat cestou modifikace původního DE. Již v roce 2002 byl presentován algoritmus SPDE (Abbass 2002) jako nástupce algoritmu PDE (Abbass et al. 2001). SPDE patří do skupiny „self-adaptive“ algoritmů a svou výkonností překonává DE. Výhodou je především to, že alespoň do jisté míry minimalizuje počet pracovních parametrů. Dalšími algoritmy jsou SaDE (Qin & Suganthan 2005), který zcela odstraňuje všechny pracovní parametry a naopak zavádí nový s názvem „learning period“. O několik let později byl navržen algoritmus JADE (Zhang & Sanderson 2009), který má jen dva pracovní parametry a je k dispozici ve dvou implementačních variantách označovaných jako „s archivem a bez archivu“. Zajímavým algoritmem je také „Differential Evolution With Self Adapting Populations“ (DESAP) (Teo 2005). Významným přínosem je „self-adaptive“ algoritmus jDE (Brest et al. 2006). Počet pracovních parametrů jDE je sice pět a je také ponechána volba odchylkového vektoru, ale pracovní parametry původního algoritmu DE, jsou voleny automaticky. jDE dosahuje skvělých výsledků v mnoha typech optimalizačních úloh (Brest et al. 2007) a v pozdějších letech byl autory dokonce vylepšen (Brest et al. 2014). Velmi zajímavý algoritmus je EPSDE (Mallipeddi & Suganthan 2010), který podobně jako SaDE zcela minimalizuje počet pracovních parametrů. Na rozdíl od DE využívá jen 4 odchylkové vektory. Jsou převzaty jednak z původního algoritmu DE, dále pak z (Iorio & Li 2004) a také z algoritmu JADE (Zhang & Sanderson (2009)). Kromě self-adaptive optimalizátorů odvozených od DE existují i tzv. adaptivní optimalizátory např. FADE (Liu & Lampinen 2005) nebo algoritmus (aDE) (Noman et al. 2011). Jejich nevýhodou je, že podobně jako DE ponechávají volbu některých pracovních parametrů na uživateli. Komparativní studii výkonnosti „self-adaptive“ a „adaptive“ algoritmů odvozených od DE lze nalézt např. v (Brest et al. 2007).

Všechny výše jmenované algoritmy byly testovány na vhodnost použití pro úlohu identifikace osob s využitím kontury ruky. Nejlepší výsledky prokázal algoritmus EPSDE. Danou úlohu je schopen řešit velmi efektivně, a proto je také využit jako hlavní optimalizátor. Detailní popis viz sekce experimentálních výsledků.

7.2.2.3 Algoritmus ICP

Algoritmus ICP (Besl & McKay 1992) byl poprvé presentován v roce 1992 a vychází z dříve publikované práce (Horn 1987). Jeho úkolem bylo bezchybně ztotožnit dva shluky bodů v 3-dimensionálním prostoru s využitím operací rotace a translace a to tak, aby euklidovská vzdálenost podle (7.2.2) byla co možná nejmenší. Optimální zarovnání je dané hodnotou $\mathcal{E} = 0$ a této hodnoty lze obvykle dosáhnout jen u dvou totožných shluků bodů. Proces optimalizace je vyjádřen vztahem:

$$\mathcal{E} = \arg \min_{\mathbf{R}, \mathbf{t}} \sum_{i=1}^n \|(\mathbf{R}P_i + \mathbf{t}) - P_j\|_2^2 \quad (7.2.1)$$

kde \mathbf{R} , $\mathbf{R} \in \mathbb{R}$ je rotační matice 3×3 , \mathbf{t} je translační vektor a v (Besl & McKay 1992) je prováděno zarovnávání dvou mraků bodů v 3D prostoru. P_i a P_j představují body obou zarovnávaných shluků. Počet bodů v obou shlucích může být samozřejmě různý. Originální algoritmus ICP (Besl & McKay 1992) se skládá ze třech kroků:

- 1) Nejprve je pro každý bod P_i vyhledán nejbližší bod z P_j s využitím aktuálních hodnot \mathbf{R} a \mathbf{t} .
- 2) Aktualizace hodnot \mathbf{R} a \mathbf{t} tak, aby platilo, že:

$$(\mathbf{R}, \mathbf{t}) = \min_{\mathbf{R}, \mathbf{t}} \sum_{i=1}^n \|(\mathbf{R}P_i + \mathbf{t}) - P_j\|_2^2 \quad (7.2.2)$$

- 3) Pokud hodnota \mathcal{E} je menší, než předem zvolené minimum, algoritmus končí. Byly nalezeny optimální hodnoty \mathbf{R} , \mathbf{t} . V opačném případě se body 1,2,3 opakují.

Nezbytnou počáteční podmínkou algoritmu ICP je, aby oba shluky bodů byly v prvním kroku vůči sobě zarovnány o co možná nejmenší vzdálenost. Tento první krok se obvykle řeší ztotožněním těžišť obou shluků. Velkou výhodou algoritmu ICP je skutečnost, že postupem let byla jednoznačně a opakovaně prokázána jeho schopnost konvergence k jedinému lokálnímu optimu viz (Pottmann et al. 2006; Maier-Hein et al. 2012). Detailnější informace a přehledovou studii lze nalézt např. v (Rusinkiewicz & Levoy 2001; Jost & Hügli 2002; Rodrigues et al. 2002).

7.2.3 Navržený algoritmus

Základem navrženého algoritmu pro identifikaci osob je optimalizátor EPSDE a algoritmus ICP. EPSDE tvoří obal ICP a rozhoduje, jakým způsobem budou kontury rukou vůči sobě zarovnány. Navržený algoritmus pro klasifikaci kontur je podobný algoritmu v kap. 7 a proto popíšeme jen základní kroky:

1/ Nejprve je vytvořena databáze modelových kontur rukou, vůči kterým bude identifikovaná osoba porovnávána. Každá osoba, která má mít „zabezpečený přístup“ má v databázi uloženu jednu modelovou konturu, kterou budeme označovat jako \mathcal{M} . Identifikovaná osoba, která žádá o povolení přístupu, bude značena jako \mathcal{S} . Výsledkem identifikačního procesu je číselně vyjádřená míra shody mezi modely \mathcal{M} , které jsou uloženy v databázi a identifikovaným vzorkem \mathcal{S} . Modelové kontury rukou jsou vytvořeny stejným postupem, jaký je použit pro pozdější identifikaci osob, a proto popíšeme postup univerzálně.

2/ Osoba, která vyžaduje identifikaci nebo vytvoření modelové kontury vloží ruku do biometrického scanneru. Je provedeno snímání a je získán barevný (RGB) snímek v požadovaném rozlišení $I_w \times I_h$ s využitím digitálního fotoaparátu. Ruka je vložena do zorného pole kamery tak, aby se nacházela přibližně uprostřed a osa ruky a osa X pomyslného souřadného systému svírá co možná nejmenší úhel – viz Obr. (7.2).1G. Prsty směřují vlevo a palec dolů. Pro výpočet lze využít jak snímky rukou ze

scanneru (snímek dlaně a prstů) tak i s využitím klasického fotoaparátu (snímek hřbetu ruky nebo dlaně s prsty). Ruka není ničím zastíněná. Základním předpokladem je, že konturu lze snadno extrahovat. Ruka nemusí nutně ležet na podložce. Je možné použít snímky ve směru hřbetu ruky i ve směru dlaně ruky.

3/ Snímek je nejprve převeden na HSB reprezentaci (Stockman & Shapiro 2001) a poté na černobílý snímek B&W (2 barvy, černá a bílá). Snímek v B&W reprezentaci vypadá tak, že pozadí snímku je bílé a popředí resp. ruka je černá. V ideálním případě je pro konverzi z HSB ($H = \langle 0,360 \rangle$, $S = \langle 0,0,1,0 \rangle$, $B = \langle 0,0,1,0 \rangle$) reprezentace na B&W snímek zapotřebí jen kanál intenzity B. Uvažovány jsou pouze pixely s limitem $B \leq 0,4$.

4/ B&W snímek je zatížen artefakty, které znesnadňují další zpracování, a proto je použita filtrace. Všechny bílé pixely, které ve svém 8-mi okolí mají méně, než 7 černých pixelů jsou přebarveny na černo. Všechny černé pixely, které ve svém 8-mi okolí mají méně než 5 bílých pixelů, jsou obarveny na bílo. Jsou odstraněny (přebarveny na bílo) všechny shluky černých pixelů, které se nachází uvnitř čtvercové oblasti o velikosti 21x21 pixelů, která má současně na svém okraji (2x21+2x19 pixelů) nulový počet černých pixelů. Masky velikosti 21x21 pixelů se posouvá po celém obraze z levého horního rohu až po pravý spodní roh. Jsou také odstraněny artefakty jako středící značky v levé části snímků a vyčištěny všechny čtyři okraje snímků o šířce 15 pixelů.

5/ Filtrovaný snímek B&W je nejprve využit pro nalezení délky prostředního prstu. Je nalezen první černý pixel zleva v obraze B&W, který u většiny pětiprstých lidí patří na špičku prostředníčku. Z tohoto bodu je algoritmem sledování kontury odhadnuta délka prostředníčku. Délka celé ruky až po zápěstí je počítána jako 2,1x délka prostředníčku. Jakmile je známa délka ruky, je s využitím algoritmu sledování kontury vypočítána kontura ruky. Pro nalezení kontury ruky je použit algoritmus RadialSweep (Pavlidis 1982; Parker 2010). Stejný algoritmus je použit pro nalezení délky prostředníčku. Počáteční bod pro výpočet kontury je mezi palcem a zápěstím podle toho, jak byla vypočtena délka ruky.

6/ Kontura ruky je využita pro výpočet Radial Distance Diagramu (RDD) – viz Obr. (7.2).1D. RDD je rozsegmentován a to tak, že jsou nalezena maxima a minima v XY diagramu RDD – viz Obr. (7.2).1D. Díky segmentaci RDD získáme body na špičkách jednotlivých prstů a také body u kloubů prstů mezi jednotlivými prsty. RDD je získán z bodu, který se nachází uprostřed zápěstí.

7/ Algoritmus ICP nezarovnává celou konturu ruky, ale pouze 4 prsty bez palce a používá ukazováček, prostředníček, prsteníček a malíček a pak také několik příznaků viz Tab. (7.2).1 a Obr. (7.2).1G. Chybějící části kontury ruky jsou dobře vidět na Obr. (7.2).1G. Kontura palce není uvažována a stejně tak není uvažována kontura dlaně mezi zápěstím a malíčkem a mezi palcem a ukazováčkem a také mezi palcem a zápěstím.

8/ Pro každý prst je vypočtena osa prstu pomocí algoritmu lineární regrese (LR) (Yan & Su 2009; Wikipedia 2020a). Algoritmus LR aproximuje množinu bodům, které náleží danému prstu přímkou. Jsou uvažovány jen body náležející kontuře každého prstu + malá korekce spočívající ve vynechání několika bodů kontury v okolí bodů $P_{1,3,5,7}$. Díky této korekci je možné „vylepšit“ úhel natočení regresní přímky. Ukazováčku náleží kontury $S_1 + S_2$ atd. Osa prstu získaná lineární regrese je v dalším kroku výpočtu nahrazena úsečkou, jejíž koncové body korespondují s body dané kontury prstu. Koncové body kontury prstu jsou nalezeny tak, že je vytvořen kolmý průmět každého bodu kontury na osu prstu a nejvzdálenější body pak určují koncové body úsečky. Aby bylo možné najít kloub každého prstu, je nutné úsečku protáhnout směrem do dlaně. Bohužel universální délka protažení o 20 procent bez ohledu na prst, která je navržena např. v (Yörük et al. 2006) v našem případě nefunguje. Pro každý prst je nutné prodloužit délku aproximační úsečky individuálně, nicméně hodnoty protažení mají universální charakter pro všechny snímky v databázi GPDS. Tímto postupem jsou nalezeny pozice kloubů. Protažení je provedeno o následující počet pixelů pro ukazováček 25, prostředníček 30, prsteníček 58 a malíček 40.

Jedná se samozřejmě pouze o odhad pozic kloubů.

9/ Nyní je připravená kontura ruky a nalezeny významné body pro další klasifikaci – viz. Obr. (7.2).1G. Pokud se jedná o modelovou konturu \mathcal{M} , je vložena do databáze. Pokud je jedná o konturu \mathcal{S} , pak je porovnávána s konturami \mathcal{M} a je rozhodnuto, jestli náleží osobě, která je registrovaná v databázi kontur \mathcal{M} . Rozdělení kontur na \mathcal{M} a \mathcal{S} je samozřejmě vědomý proces řízený lidským operátorem. Při klasifikaci jsou postupně vybírány kontury \mathcal{M} z databáze a jsou porovnávány (klasifikovány) vůči kontuře \mathcal{S} identifikované osoby. Při procesu klasifikace je modelová kontura \mathcal{M} plně statická a vůči ní je porovnávána kontura \mathcal{S} . Jednotlivé prsty kontury \mathcal{S} se při procesu klasifikace natáčí v kloubech P_{K1} , P_{K2} , P_{K3} , P_{K4} v maximálním rozsahu úhlů, který je definován v (7.2.5) a klasifikační algoritmus využívající ICP se snaží ztotožnit shluky bodů jednotlivých prstů kontury \mathcal{M} a \mathcal{S} s využitím klasifikačního kritéria (7.2.3, 7.2.4). Úhel natočení jednotlivých prstů je řízen algoritmem EPSDE. Pozice kloubů jsou svázány s těžištěm kontury \mathcal{S} . V prvním kroku klasifikace jsou těžiště kontury \mathcal{M} a \mathcal{S} ztotožněny. Pozice těžiště kontury \mathcal{S} je posouváno podle potřeb klasifikačního algoritmu vůči těžišti kontury \mathcal{M} a stejně tak se mění i natočení celé kontury \mathcal{S} , která reprezentuje ruku identifikované osoby. Algoritmus EPSDE pracuje tak dlouho, dokud hodnota *fitness* funkce nenabyde požadované velikosti nebo dokud není splněn předem daný počet evolučních cyklů.

10/ Výsledkem klasifikace je míra podobnosti \mathcal{E} mezi jednotlivými konturami \mathcal{M} v databázi a konturou \mathcal{S} identifikované osoby – viz (7.2.3). Míra podobnosti je udávána reálným číslem. V případě, že jsou klasifikovány dvě totožné kontury \mathcal{M} a \mathcal{S} , pak reálné číslo vyjadřující míru podobnosti je rovné nule. V opačném případě je vždy kladné a nenulové. Míra podobnosti vyjádřená reálným číslem je daná jako součet vzdáleností jednotlivých bodů kontury \mathcal{M} k nejbližším bodům kontury \mathcal{S} podle algoritmu ICP a několika dalších příznaků, jejich seznam je uveden v Tab. (7.2).1 a reprezentace *fitness* funkce viz (7.2.4). Pro nalezení nejlepšího zarovnání kontur je použit algoritmus ICP. Jako metrika je uvažována Euklidovská vzdálenost mezi dvěma body.

| Tabulka (7.2).1 Příznaky použité při klasifikaci | | | |
|--|--|-----------------------------|---|
| Vzdálenosti | Kontury | Vzdálenosti | Úhly |
| $f_1: \overline{P_2P_3}$ | $f_7: S_1 + S_2, [P_{K1}], \{x_0\}$ | $f_{11}: \overline{P_3P_5}$ | $f_{14}: \overline{P_3P_5P_7} = \alpha_1^{\mathcal{M}}$ |
| $f_2: \overline{P_3P_4}$ | $f_8: S_3 + S_4, [P_{K2}], \{x_1\}$ | $f_{12}: \overline{P_5P_7}$ | $f_{15}: \overline{P_5P_7P_3} = \alpha_2^{\mathcal{M}}$ |
| $f_3: \overline{P_4P_5}$ | $f_9: S_5 + S_6, [P_{K3}], \{x_2\}$ | $f_{13}: \overline{P_3P_7}$ | $f_{16}: \overline{P_5P_3P_7} = \alpha_3^{\mathcal{M}}$ |
| $f_4: \overline{P_5P_6}$ | $f_{10}: S_7 + S_8, [P_{K4}], \{x_3\}$ | | |
| $f_5: \overline{P_6P_7}$ | | | |
| $f_6: \overline{P_7P_8}$ | | | |

$\overline{P_2P_3}$ - značí Euklidovskou vzdálenost bodů. $S_1 + S_2$ - značí kontury – viz Obr. (7.2).1, které se pohybují okolo společného kloubu $[P_{K1}]$. $\overline{P_3P_5P_7}$ - značí úhel $\alpha_1^{\mathcal{M}}$, který svírají přímky vedené body P_3P_5 a P_5P_7 a které se protínají v bodě P_5 . Analogicky pro ostatní příznaky $f_{...}$ $[P_{K1}]$ - udává kloub, se kterým jsou kontury S_1, S_2 svázány. Všechny klouby P_K jsou pak společně svázány s těžištěm kontury. $\{x_{...}\}$ - značí korespondující gen chromosomu evolučního algoritmu. $\alpha_1^{\mathcal{M}}, \alpha_2^{\mathcal{M}}, \alpha_3^{\mathcal{M}}$ - úhly jsou uvažovány v radiánech. Příznaky $f_{...}$ jsou specifické pro modelovou konturu \mathcal{M} a datový vzorek \mathcal{S} a jsou označovány jako $f_i^{\mathcal{M}}, f_i^{\mathcal{S}}$.

V Tab. (7.2).1. jsou uvedeny některé vybrané měřené vzdálenosti. Poměrně značným problémem je správný odhad bodu P_1 , který se nalézá mezi ukazováčkem a palcem. Poloha bodu se mění až o 1-1.5cm podle toho, jak je palec přimknutý k ukazováčku. To samozřejmě výrazně ovlivňuje klasifikaci. Z tohoto důvodu byly zvoleny jen body $P_{3,5,7}$ a je mezi nimi vypočtena vzdálenost a úhly, které svírají přímky proložené těmito body.

Evoluční proces s využitím optimalizátoru EPSDE lze zapsat jako:

$$\mathcal{E} = \arg \operatorname{opt}_{\mathcal{H} \in \mathbb{R}, \mathcal{S}} \mathfrak{F}_{\text{EA}}(\mathcal{M}, \mathcal{S}, \mathcal{H}) \quad (7.2.3)$$

kde \mathcal{H} značí prostor možných řešení, který je definován proměnnými f_k jejichž správné nastavení je hledáno. Kontura \mathcal{M} je statická, kontura \mathcal{S} je pohyblivá.

Funkce *fitness* se počítá jako součet reálných čísel pro jednotlivé zvolené příznaky – viz Tab. (7.2).1 a (7.2.5). Pro příznaky $f_{1,2,3,4,5,6,11,12,13}$ se uvažuje Euklidovská vzdálenost. Stejně tak i pro vzdálenosti $f_{7,8,9,10}$ bodů kontur \mathcal{M} a \mathcal{S} . V případě příznaků $f_{14,15,16}$, které značí úhly je jako metrika použita absolutní hodnota z rozdílu úhlů – viz (7.2.4).

$$\text{fitness} = \begin{cases} \left(\sum_{i=1}^{i=16} \wp \right)^6 & \text{if } \mathcal{L} = 1 \\ \sum_{i=1}^{i=16} \wp & \text{if } \mathcal{L} = 0 \end{cases} \quad (7.2.4)$$

$$\text{fitness}, \wp \in \mathbb{R}$$

$$\wp = \begin{cases} 10 \times |f_i^{\mathcal{M}} - f_i^{\mathcal{S}}| & \text{if } i = 1,2,3,4,5,6 \\ f_{7,8,9,10} & \text{if } i = 7,8,9,10 \\ 10 \times |f_i^{\mathcal{M}} - f_i^{\mathcal{S}}| & \text{if } i = 11,12,13 \\ 1000 \times |f_i^{\mathcal{M}} - f_i^{\mathcal{S}}| & \text{if } i = 14,15,16 \end{cases}$$

$$f_{1,2,3,4,5,6}^{\mathcal{S}}: d(P_m^{\mathcal{S}}, P_n^{\mathcal{S}}), f_{1,2,3,4,5,6}^{\mathcal{M}}: d(P_m^{\mathcal{M}}, P_n^{\mathcal{M}})$$

$$f_{7,8,9,10}: d(P_j^{\mathcal{S}}, P_{min}^{\mathcal{M}})$$

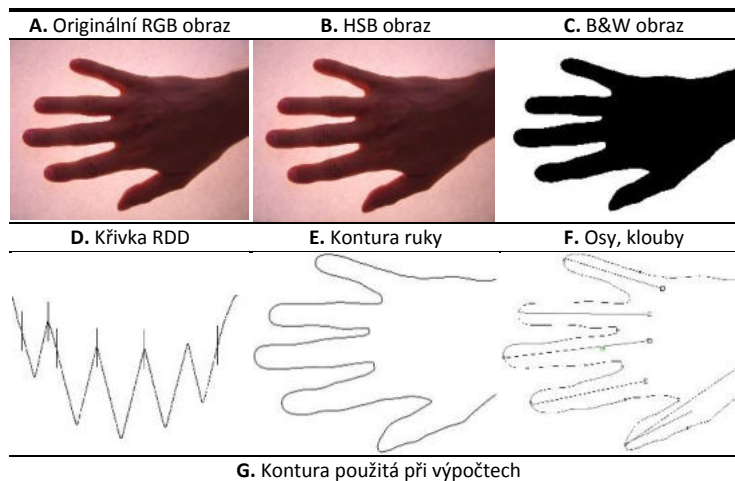
$$d(P_j^{\mathcal{S}}, P_{min}^{\mathcal{M}}) = \min d(P_j^{\mathcal{S}}, P_i^{\mathcal{M}}) |_{i=1}^{i=n}$$

$$f_{11,12,13}^{\mathcal{S}}: d(P_m^{\mathcal{S}}, P_n^{\mathcal{S}}), f_{11,12,13}^{\mathcal{M}}: d(P_m^{\mathcal{M}}, P_n^{\mathcal{M}})$$

$$f_{14,15,16}^{\mathcal{S}}: \alpha_j^{\mathcal{S}}, f_{14,15,16}^{\mathcal{M}}: \alpha_j^{\mathcal{M}}, j \in (1,2,3)$$

kde pro vzdálenost $d(A, B)$ dvou bodů $A(x_A, y_A)$, $B(x_B, y_B)$ platí, že $d(A, B) = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$. Indexy m, n pro jednotlivé body P_{\dots} viz Tab. (7.2).1 a Obr. (7.2).1G. $f_i^{\mathcal{M}}$ reprezentuje značku náležející modelové kontuře \mathcal{M} a $f_i^{\mathcal{S}}$ reprezentuje značku náležející datovému vzorku \mathcal{S} , který je právě identifikován. Násobky $10 \times$, $1000 \times$ jsou naprosto nezbytné, protože např. rozdíly úhlů v radiánech jsou velmi malé a hodnoty by tak nebyly při klasifikaci vůbec užitečné.

Pro výpočet jsou uvažovány jen 4 prsty – viz Obr. (7.2).1. Natočení jednotlivých prstů je nutné omezit vhodnou funkcí, která udává, o kolik je možné natočit prst v daném kloubu vůči původní poloze v kontuře \mathcal{S} , kterou našel algoritmus pro výpočet kontury. Omezovací funkci budeme značit \mathcal{L} .



A. - Originální RGB snímek z digitálního fotoaparátu. **B.** - HSB reprezentace RGB snímku. **C.** - B&W reprezentace připravená k nalezení kontury. **D.** - Radial Distance Diagram, svislémi čarami jsou naznačeny hranice prstů, palec je vlevo. **E.** - Kompletní kontura vypočtená algoritmem pro hledání kontury, který byl aplikován na Obr. (7.2).1C. **F.** - nalezené osy prstů a pozice kloubů. **G.** - popis důležitých částí nalezené kontury prstů, osy prstů, klouby a částí kontur jednotlivých prstů. Prsty se při výpočtu pohybují v kloubech.

Obrázek (7.2).1. Schéma procesu získání kontury.

$$\mathcal{L} = \begin{cases} 1 & \text{if } x_{0,1,2,3} \notin \langle -0.30^{RAD}; +0.30^{RAD} \rangle \\ 1 & \text{if } x_{4,5} \notin \Omega \\ 1 & \text{if } x_6 \notin \langle -0.40^{RAD}; +0.40^{RAD} \rangle \\ 0, & \text{otherwise} \end{cases} \quad (7.2.5)$$

$x_{0,1,2,3,4,5,6}$ značí jednotlivé geny chromosomu \mathbf{X}_i . Přiřazení genů chromosomu k jednotlivým příznakům je dáno následovně: $\{x_0\}$ = ukazováček, $\{x_1\}$ = prostředníček, $\{x_2\}$ = prsteníček, $\{x_3\}$ = malíček, $\{x_4\}$ = ΔX , $\{x_5\}$ = ΔY , $\{x_6\}$ = $\Delta\alpha$. Geny x_4, x_5, x_6 udávají v daném pořadí změnu pozice $\Delta X, \Delta Y$ a natočení $\Delta\alpha$ celé kontury \mathbf{S} v jejím těžišti vůči těžišti kontury \mathcal{M} a ose X souřadného systému – viz Obr. (7.2).1G. Kontura \mathbf{S} se natáčí v bodě svého těžiště. Pořadí genů je odlišné od pořadí v kap. 7.1.

Příznaky $f_{7,8,9,10}$ jsou svázané s konturou prstů a to následovně: f_7 -ukazováček, f_8 -prostředníček, f_9 -

prsteníček, f_{10} -malíček. V případě, že při procesu evoluce je algoritmem EPSDE nastaven úhel nebo pozice mimo dovolené hranice, je použita šestá mocnina na výslednou funkci *fitness* – viz (7.2.4) a (7.2.5). Hodnota Ω značí prostor, ve kterém se při procesu evoluce může pohybovat těžiště kontury \mathcal{S} tedy hodnoty genů x_4, x_5 . Prostor Ω je dán jako obdélníková plocha se středem v pozici P_Ω a to následovně:

$$P_\Omega(x_\Omega = 0.5I_w, y_\Omega = 0.5I_h) \quad (7.2.6)$$

$$\Omega = [\langle x_\Omega - 0.17I_w, x_\Omega + 0.17I_w \rangle, \langle y_\Omega - 0.17I_h, y_\Omega + 0.17I_h \rangle]$$

I_w značí šířku obrázku, I_h značí výšku obrázku. Chromosome \mathbf{X}_i ovlivňuje jen příznaky f_7, f_8, f_9, f_{10} . Ostatní příznaky jsou dány pevně charakterem kontury ruky a samozřejmě se účastní výpočtu *fitness* funkce.

Evoluční optimalizátor pracuje se 7-mi dimenzionální optimalizační *fitness* funkcí $D_{im} = 7$, která je omezená (constrained), nespojitá, neseparabilní, silně nelineární a špatně podmíněná. Součástí identifikačních údajů každého jedince \mathbf{X}_i je i hodnota F , takže správně by bylo lepší zapsat $\mathbf{X}_i = [x_0, \dots, x_6, F]$, ale hodnota F se neúčastní výpočtu podle (7.2.4). Hodnota F je u algoritmu EPSDE volena náhodně a je zahrnuta do jeho „self-adaptive“ mechanismů. Geny x_0, x_1, x_2, x_3 obsahují hodnoty natočení kontur prstů S_{\dots} v kloubech $P_{K_{\dots}}$ o vypočtený úhel (\pm) v radiánech. Výpočtu *fitness* funkce podle (7.2.4) se účastní všechny příznaky f_{\dots} , ale pouze příznaky f_7, \dots, f_{10} jsou přímo ovlivňovány chodem algoritmu EPSDE. Hodnoty pro ostatní příznaky jsou dané pevně každé jednotlivé kontuře resp. není hledána jejich optimální hodnota.

V experimentální sekci jsou použity pro porovnání i sestavy kontury, která má všech 5 prstů a současně jsou využity i další příznaky – příznaky.

7.2.4 Experimentální výsledky

Sekce experimentálních výsledků je rozdělena na několik částí.

A. Databáze snímků GPDS a způsob hodnocení výsledků.

B. Volba vhodných pracovních parametrů optimalizátoru EPSDE.

C. Experimenty s databází GPDS využívající různou kombinaci příznaků a časová náročnost použitých estimátorů.

7.2.4.1 Databáze snímků GPDS a způsob hodnocení výsledků

Databáze GPDS (Ferrer et al. 2007) obsahuje snímky z klasického stolního scanneru od 150 osob, 10 snímků od každé osoby. Jedná se o snímky dlaně včetně prstů. Snímky mají různé rozlišení (1521x1280x8bit-šedotónový obraz, 1402x1018x8bit-šedotónový obraz atd.) a jsou také různě nasvícené a mnoho snímků obsahuje různé středící značky na různých místech. Filtrace všech těchto nepoužitelných a neustále se měnících částí obrazů je poměrně časově náročná. Velkou nevýhodou je také to, že do mnoha snímků zasahuje okolní osvětlení.

| Algoritmus (7.2).1 Způsob výpočtu hodnot FAR, FRR, EER | |
|--|--|
| Vstup: lim | Výstup: EER, FAR, FRR $TPR_{EER}, FPR_{EER}, TNR_{EER}, FNR_{EER}$ $P_{TP-EER}, P_{FN-EER}, N_{FP-EER}, N_{TN-EER}$ |
| 1 | $\Sigma G_{TOT} = 940; \Sigma I_{TOT} = 87420; x \approx "oo"$ |
| 2 | for $\lambda_{EER}=0; \lambda_{EER} < lim; \lambda_{EER} = \lambda_{EER} + 1$ |
| 3 | $P_{TP} = 0; P_{FN} = 0; N_{FP} = 0; N_{TN} = 0$ |
| 4 | for all models \mathcal{M}_{EA} (94) |
| 5 | for all samples S_{EA} (940) |
| 6 | if $S \cong \mathcal{M} \wedge \varepsilon \leq \lambda_{EER}$ then $P_{TP} = P_{TP} + 1$ (SUCCESS)⊙ |
| 7 | if $S \cong \mathcal{M} \wedge \varepsilon > \lambda_{EER}$ then $P_{FN} = P_{FN} + 1$ (FAILURE) |
| 8 | if $S \not\cong \mathcal{M} \wedge \varepsilon \leq \lambda_{EER}$ then $N_{FP} = N_{FP} + 1$ (FAILURE) |
| 9 | if $S \not\cong \mathcal{M} \wedge \varepsilon > \lambda_{EER}$ then $N_{TN} = N_{TN} + 1$ (SUCCESS)⊙ |
| 10 | next |
| 11 | next |
| 12 | $TPR = P_{TP}/(P_{TP} + P_{FN}); FPR = N_{FP}/(N_{FP} + N_{TN})$ |
| 13 | $TNR = N_{TN}/(N_{FP} + N_{TN}); FNR = P_{FN}/(P_{TP} + P_{FN})$ |
| 14 | $R_1 = P_{FN}/\Sigma G_{TOT}; R_2 = N_{FP}/\Sigma I_{TOT}$ |
| 15 | if $x > R_1 - R_2 $ then |
| 16 | $x = R_1 - R_2 ; EER = \lambda_{EER}$ |
| 17 | $P_{TP-EER} = P_{TP}; P_{FN-EER} = P_{FN}; N_{FP-EER} = N_{FP}; N_{TN-EER} = N_{TN}$ |
| 18 | $TPR_{EER} = TPR; FPR_{EER} = FPR$ |
| 19 | $TNR_{EER} = TNR; FNR_{EER} = FNR$ |
| 20 | $FAR = R_2; FRR = R_1$ |
| 21 | $\Sigma I_{UL} = N_{FP}; \Sigma G_{OL} = P_{FN}$ |
| 22 | endif |
| 23 | next |

$S \cong \mathcal{M}$ - značí, že zkoumaná kontura S a modelová kontura \mathcal{M} náleží stejnému člověku. Detekce správné klasifikace byla provedena podle jména obou vzorků. λ_{EER} - udává jednotlivé hodnoty *fitness* ve kterých je hledán bod *EER*. *lim*-je nutné zjistit experimentálně nebo stačí jej uvažovat jako max. *fitness* ze všech hodnot na diagonále ve výsledné tabulce hodnot, pokud je tabulka k dispozici. x – pomocná proměnná. *EER*-equal error rate point, *FAR*-false acceptance rate, *FRR*-false rejection rate, TPR_{EER} -true positive rate at *EER* point, FPR_{EER} -false positive rate at *EER* point, TNR_{EER} -true negative rate at *EER* point, FNR_{EER} -false negative rate at *EER* point, P_{TP-EER} -true positive (positive, genuine), P_{FN-EER} -false negative (positive, genuine), N_{FP-EER} -false positive (negative, impostor), N_{TN-EER} -true negative (negative, impostor), *FAR*-false acceptance rate, *FRR*-false rejection rate. *TNR*-true negative rate, *FNR*-false negative rate. Values *TPR*, *FPR* define Receive operating characteristics (ROC) curve. $FAR = FPR_{EER}$, $FRR = FNR_{EER}$, $\Sigma G_{TOT} = (P_{TP} + P_{FN})$, $\Sigma I_{TOT} = (N_{FP} + N_{TN})$. ΣG_{TOT} -total number of genuines, ΣI_{TOT} -total number of impostors. ΣI_{UL} -sum of impostors under limit λ_{EER} . ΣG_{OL} -sum of genuines over limit λ_{EER} .

Z výše uvedených důvodů musely být některé snímky vyřazeny a nahrazeny náhodným snímkem ze sady dostupných od identické osoby. Pro účely klasifikace bylo zvoleno 94 osob a 10 snímků od každé

osoby, celkem 940 snímků. Jeden snímek od každé osoby byl zvolen jako referenční vzorek \mathcal{M} . Vůči tomuto referenčnímu vzorku je pak provedeno porovnávání s ostatními snímky \mathcal{S} v databázi GPDS. Celkem je provedeno $94 \times 940 = 88360$ porovnání kontur pro každý testovaný algoritmus.

Aby bylo možné výsledky nějak klasifikovat, byly zvoleny standardní statistické ukazatele a to: FAR-False Acceptance Rate, FRR-False Rejection Rate a výpočet bodu EER-Equal Error Rate. Aby bylo možné zobrazit i křivku ROC, byly dopočteny také hodnoty FPR-false positive rate a TPR-true positive rate. Algoritmus pro výpočet hodnot je uveden na Alg. (7.2).1. Horní limit pro vstupní proměnnou lim bylo nutné zjistit experimentálně. Se vzrůstající hodnotou κ_S se zvyšuje i lim z hodnoty $lim = 1000$ pro $\kappa_S = 1$ až na hodnotu $lim = 40000$ pro $\kappa_S = 40$. Hodnota lim je v pixelech. Obecně je hodnota lim rovna max. *fitness* ze všech genuiných vzorků v tabulce výsledků. Pokud pro účely scanování poskytla každá osoba 10 snímků a osob bylo 94, je to max. *fitness* ze 940 hodnot, které se ve výsledné tabulce nalézají obvykle na hlavní diagonále.

Aby bylo možné kontury nějak klasifikovat, byly zvoleny významné body, úhly a vzdálenosti na kontuře ruky a také části kontur – viz Tab. (7.2).1. Všechny tyto příznaky jsou označeny jako $f_{1, \dots, 16}$ a pro účely klasifikace jsou zvoleny určité kombinace těchto příznaků. Pro výpočet byl použit různý počet pixelů kontury ruky a bylo zkoumáno, jak se zhorší výsledná přesnost resp. hodnoty *FAR*, *FRR*, *EER*. Zvolené kombinace jsou označeny jako κ_M, κ_S a reprezentují konturu, kde pro výpočet je použit jen κ_M -tý pixel kontury \mathcal{M} a κ_S -tý pixel kontury \mathcal{S} . Vzhledem k tomu, že přesnost porovnání kontur se značně (rapidně) zhoršuje pro hodnoty $\kappa_M > 1$, bude v experimentech uvažována jen hodnota $\kappa_M = 1$ a hodnota κ_S se bude měnit podle následující množiny hodnot $\kappa_S \in (1, 5, 10, 15, 20, 25, 30, 35, 40)$. Např. pro $\kappa_M = 40, \kappa_S = 1$ je obvykle dosažena hodnota *FAR* = 25%. Takovéto výsledky jsou samozřejmě nepoužitelné. Průměrný počet pixelů kontury ruky v databázi GPDS je cca 1800 a tak při $\kappa_S = 40$ je využito při výpočtu jen 45 pixelů kontury \mathcal{S} a všechny pixely kontury \mathcal{M} . To znamená pouhých cca 6 pixelů na každý jednotlivý prst kontury \mathcal{S} .

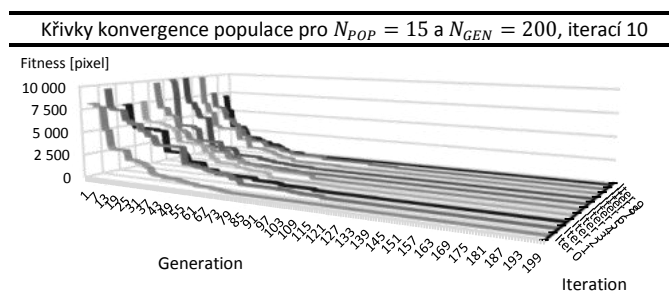
7.2.4.2 Volba vhodných pracovních parametrů optimalizátoru EPSDE

Optimalizátor EPSDE má jen několik pracovních parametrů. N_{POP} , N_{GEN} a L_P . Nejdůležitější je samozřejmě volba optimálního počtu jedinců v populaci a počtu generací. V Tab. (7.2).2 je výsledek, který ukazuje, jaké hodnoty je možné zvolit, aby výsledek byl co nejpřesnější. Porovnávány byly identické kontury \mathcal{M} a \mathcal{S} . S ohledem na kapitolu 7.1 lze tyto kontury označit také jako \mathcal{M}_{ea} a \mathcal{S}_{ea} , ale ponecháme \mathcal{M} a \mathcal{S} . Rozlišení obrazů bylo 640×480 pixelů pro všechny experimenty. Zvýrazněná oblast představuje vhodné hodnoty. S ohledem na časovou náročnost bude při experimentech použita kombinace $N_{POP} = 15$, $N_{GEN} = 200$ a $L_P = 10$. Při těchto hodnotách se již výsledek nemění ani při počtu opakování 10000. Hodnota L_P má jen velmi malý vliv na výsledek. Vliv různých kombinací příznaků podle Tab. (7.2).1 na hodnotu L_P pozorován nebyl. Výsledky v Tab. (7.2).2 jsou získány pro hodnoty $\kappa_M = 1, \kappa_S = 1$ a při zvyšujících se hodnotách κ_S se již oblast použitelných hodnot N_{POP}, N_{GEN} naštěstí nemění.

| n. | $N_{GEN} \rightarrow$ N_{POP} | 50 | 100 | 150 | 200 | 250 | 300 |
|----|------------------------------------|-----|-----|-----|-----|-----|-----|
| 1 | 6 | 645 | 89 | 17 | 16 | 0 | 0 |
| 2 | 10 | 739 | 89 | 11 | 0 | 0 | 0 |
| 3 | 15 | 857 | 114 | 24 | 0 | 0 | 0 |
| 4 | 20 | 741 | 81 | 6 | 0 | 0 | 0 |
| 5 | 25 | 847 | 88 | 78 | 0 | 0 | 0 |
| 6 | 30 | 605 | 30 | 2 | 0 | 0 | 0 |
| 7 | 35 | 681 | 47 | 0 | 0 | 0 | 0 |
| 8 | 40 | 769 | 61 | 0 | 0 | 0 | 0 |

Výpočet proveden pro variantu se zvolenými příznaky $f_{1...18}$, tj. 4 prsty + všechny příznaky. Hodnoty jsou v pixelech a jedná se o průměr z 50ti opakování. Hodnoty $\kappa_M = 1$, $\kappa_S = 1$.

Na Obr. (7.2).2 jsou zobrazeny křivky konvergence populace ke správnému řešení pro $N_{POP} = 15$, $N_{GEN} = 200$, počet iterací 10. Počáteční rychlost konvergence je poměrně značná. Od hodnoty $N_{GEN} = 150$ se však hodnota *fitness* mění již jen minimálně. Zvolené hodnoty N_{POP} , N_{GEN} odpovídají také technickým možnostem resp. časové náročnosti. Samozřejmě by bylo možné zvolit trojnásobné hodnoty.



Obrázek (7.2).2 Křivky konvergence populace.

7.2.4.3 Experimenty s databází GPDS využívající různou kombinaci příznaků a časová náročnost použitých estimátorů

Celkem bylo testováno 5 kombinací příznaků z Tab. (7.2).1. Čistě z důvodu zajímavého porovnání byly připojeny i výsledky experimentů za celou konturou ruky. Experimenty jsou označeny jako (A13) – (A17) a navazují tak na označení experimentů z předchozí kapitoly a mají následující uspořádání a výsledky:

(A13) Příznaky $f_{1...16}$, tj. všechny příznaky podle Tab. (7.2).1., kontura ruky má 4 prsty – ukazováček, prostředníček, prsteníček a malíček. Zarovnání kontur \mathcal{M} a \mathcal{S} je řízeno algoritmem ICP a jako optimalizátor je použit EPSDE. Čtyři prsty bez palce jsou pro klasifikaci využity např. v (Allegion 2020b,a).

Výsledky klasifikace jsou v Tab. (7.2).3. Nejlepší dosažená hodnota $FAR = 1.38\%$, $FRR = 1.38\%$. V publikaci (Kang & Wu 2014) autoři dosáhli hodnoty $EER\langle 0.0,1.0 \rangle = 0.0369\%$. Při zvětšujících se hodnotách κ_S poměrně rychle klesá i přesnost, a to bez ohledu na to, že kromě kontury prstů jsou využity i ostatní příznaky. Rozdíly jsou značné a navržené sestavení rozhodně nelze označit jako dostatečně robustní i když poskytuje relativně dobré výsledky pro $\kappa_S = 1$.

Tabulka (7.2).3 Hodnoty FAR, FRR, EER
(A13) Příznaky $f_{1,..,18}$, 4 prsty + všechny příznaky, bez palce

| n. | κ_M | κ_S | ΣI_{UL} | FAR | ΣG_{OL} | FRR | $EER / EER\%$ | Tot.Time |
|----|------------|------------|-----------------|--------|-----------------|--------|---------------|----------|
| 1 | 1 | 1 | 1207 | 1.38% | 13 | 1.38% | 2360 / 1.38% | 4.102 |
| 2 | 1 | 5 | 5254 | 6.01% | 57 | 6.06% | 1290 / 6.01% | 0.655 |
| 3 | 1 | 10 | 7174 | 8.20% | 77 | 8.19% | 1115 / 8.2% | 0.639 |
| 4 | 1 | 15 | 8090 | 9.25% | 87 | 9.25% | 1054 / 9.25% | 0.561 |
| 5 | 1 | 20 | 8840 | 10.11% | 95 | 10.10% | 1030 / 10.10% | 0.452 |
| 6 | 1 | 25 | 9210 | 10.53% | 99 | 10.53% | 1012 / 10.53% | 0.390 |
| 7 | 1 | 30 | 9436 | 10.79% | 102 | 10.85% | 998 / 10.79% | 0.374 |
| 8 | 1 | 35 | 9585 | 10.96% | 103 | 10.95% | 988 / 10.96% | 0.343 |
| 9 | 1 | 40 | 9639 | 11.02% | 103 | 10.95% | 978 / 11.02% | 0.327 |

Tot.Time-udává celkový čas pro zarovnání jednoho vzorku v sekundách. EER -v pixelech získáno jako průsečík křivek FAR, FRR . EER -je v pixelech. $EER\%$ -získáno z ROC křivky. Celkový počet evolucí je 9 řádků \times 87420 vzorků $S = 786780$.

(A14) Jsou využity pouze příznaky $f_{1,..,6} + f_{11,..,13} + f_{14,..,16}$. Není využita kontura prstů ani žádná jiná část kontury dlaně. Výhodou tohoto uspořádání je nesmírně rychlý výpočet, protože není nutné používat evoluční algoritmus EPSDE ani algoritmus ICP. Z výpočtu $fitness$ podle (7.2.4) je vyňat řádek s příznaky $f_{7,8,9,10}$ resp. \emptyset může nabývat jen 3 hodnot.

Výsledky klasifikace jsou v Tab. (7.2).4. Hodnotu κ_S zde není možné použít, protože není využita žádná část kontury ruky. Dosažená výsledná hodnota přesnosti je $FAR = 2.65\%$, $FRR = 2.65\%$. To jsou poměrně špatné výsledky a takovéto uspořádání klasifikátoru nemá žádný význam.

Tabulka (7.2).4 Hodnoty FAR, FRR, EER ,
(A14) Příznaky $f_{1,..,6} + f_{11,..,13} + f_{14,..,16}$, žádný prst, není použit EA

| n. | κ_M | κ_S | ΣI_{UL} | FAR | ΣG_{OL} | FRR | $EER / EER\%$ | Tot. time |
|----|------------|------------|-----------------|-------|-----------------|-------|---------------|-----------|
| 1 | 1 | 1 | 2312 | 2.65% | 25 | 2.65% | 445 / 2.65% | 0.702 |

Tot.Time-udává celkový čas pro zarovnání jednoho vzorku v sekundách.

(A15) Jsou využity všechny prsty ruky včetně palce a částí dlaně podle Obr. (7.2).1E a všechny ostatní příznaky. T.j. $f_{1,..,16}$ + kontura palce a zbývající části dlaně. Zarovnání kontur \mathcal{M} a \mathcal{S} je řízeno algoritmem ICP a jako optimalizátor je použit EPSDE.

Výsledky experimentů jsou v Tab. (7.2).5. nejlepší dosažené hodnoty jsou $FAR = 2.55\%$, $FRR = 2.55\%$. Výsledky jsou horší než v Tab. (7.2).3. Důvod je ten, že palec má 3 klouby mezi zápěstím a špičkou palce. Jejich přesnou pozici je možné zjistit např. pomocí rentgenu, ale takovéto vybavení není k dispozici. Proto byl uvažován pouze kloub palce na zápěstí. Je zajímavé, že ač metoda (A13) dosahuje lepšího výsledku pro $\kappa_M = 1, \kappa_S = 1$, tak při zvyšujících se hodnotách κ_S klesá přesnost rychleji. Metodu (A13) lze tedy označit za méně robustní i když jen o málo.

Tabulka (7.2).5 Hodnoty FAR, FRR, EER ,
(A15) Všech 5 prstů + všechny příznaky, tj. $f_{1,..,18}$ + palec

| n. | κ_M | κ_S | ΣI_{UL} | FAR | ΣG_{OL} | FRR | EER | Tot. time |
|----|------------|------------|-----------------|-------|-----------------|-------|--------------|-----------|
| 1 | 1 | 1 | 2230 | 2.55% | 24 | 2.55% | 4497 / 2.55% | 5.491 |
| 2 | 1 | 5 | 3908 | 4.47% | 42 | 4.46% | 1727 / 4.47% | 1.341 |
| 3 | 1 | 10 | 5390 | 6.16% | 58 | 6.17% | 1353 / 6.16% | 0.795 |
| 4 | 1 | 15 | 6416 | 7.33% | 69 | 7.34% | 1224 / 7.33% | 0.624 |
| 5 | 1 | 20 | 7116 | 8.14% | 76 | 8.08% | 1160 / 8.14% | 0.514 |
| 6 | 1 | 25 | 7716 | 8.82% | 83 | 8.82% | 1122 / 8.82% | 0.468 |
| 7 | 1 | 30 | 8061 | 9.22% | 87 | 9.25% | 1093 / 9.22% | 0.436 |
| 8 | 1 | 35 | 8331 | 9.52% | 89 | 9.46% | 1073 / 9.52% | 0.405 |
| 9 | 1 | 40 | 8538 | 9.76% | 92 | 9.78% | 1058 / 9.76% | 0.374 |

Tot.Time-udává celkový čas pro zarovnání jednoho vzorku v sekundách. Celkový počet evolucí je 9 řádků \times 87420 vzorků $S_{EA} = 786780$.

(A16) Jsou využity pouze 4 prsty bez palce T.j. jen příznaky $f_{7,8,9,10}$. Zarovnání kontur \mathcal{M} a \mathcal{S} je řízeno algoritmem ICP a jako optimalizátor je použit EPSDE.

Výsledky tohoto uspořádání jsou v Tab. (7.2).6. Nejlepší dosažené hodnoty jsou $FAR = 0.48\%$, $FRR = 0.42$. Jedná se o nejlepší dosažené hodnoty ze všech experimentů. Toto uspořádání umožňuje navrhnout velmi robustní estimátor. Při $\kappa_M = 1, \kappa_S = 40$ jsou hodnoty $FAR = 1.57\%$, $FRR = 1.70\%$. Díky menšímu počtu bodů kontury je výpočet rychlejší. Důvod takovýchto dobrých výsledků je ten, že kontury prstů jsou stále stejné, i když se pohybují v rovině XY v kloubech $P_{K1,2,3,4}$. U palce toto říci nelze. Občasným problémem byly snímky, kde testovaná osoba nepřitlačila prsty až k podložce a díky tomu se délka prstů zkrátila a kontura zdeformovala.

Tabulka (7.2).6 Hodnoty FAR, FRR, EER ,
(A16) 4 prsty a žádné příznaky, tj. jen f_7, f_8, f_9, f_{10}

| n. | κ_M | κ_S | ΣI_{UL} | FAR | ΣG_{OL} | FRR | EER | Tot. time |
|----|------------|------------|-----------------|-------|-----------------|-------|--------------|-----------|
| 1 | 1 | 1 | 425 | 0.48% | 4 | 0.42% | 1225 / 0.48% | 4.710 |
| 2 | 1 | 5 | 751 | 0.85% | 8 | 0.85% | 261 / 0.85% | 1.600 |
| 3 | 1 | 10 | 711 | 0.81% | 8 | 0.85% | 130 / 0.81% | 0.670 |
| 4 | 1 | 15 | 835 | 0.95% | 9 | 0.95% | 88 / 0.95% | 0.514 |
| 5 | 1 | 20 | 1092 | 1.12% | 11 | 1.17% | 69 / 1.24% | 0.452 |
| 6 | 1 | 25 | 967 | 1.10% | 11 | 1.17% | 54 / 1.10% | 0.436 |
| 7 | 1 | 30 | 1063 | 1.21% | 11 | 1.17% | 45 / 1.12% | 0.374 |
| 8 | 1 | 35 | 1157 | 1.32% | 14 | 1.48% | 39 / 1.32% | 0.358 |
| 9 | 1 | 40 | 1331 | 1.52% | 16 | 1.70% | 34 / 1.52% | 0.343 |

Tot. Time-udává celkový čas pro zarovnání jednoho vzorku v sekundách. Celkový počet evolucí je 9 řádků \times 87420 vzorků $S_{EA} = 786780$ evolucí.

(A17) Je využito všech 5 prstů včetně palce a nejsou použity další příznaky. Zarovnání kontur \mathcal{M} a \mathcal{S} je řízeno algoritmem ICP a jako optimalizátor je použit EPSDE. Tato kombinace je zde pouze pro účely porovnání, protože zkoumáme především kombinace podle Tab. (7.2).1.

Výsledky klasifikace jsou v Tab. (7.2).7. Nejlepší dosažené hodnoty jsou $FAR = 2.12\%$, $FRR = 2.12\%$. Metoda je poměrně robustní. Se snižujícím se počtem bodů kontury se dosažená přesnost snižuje jen velmi pozvolna. Při $\kappa_M = 1, \kappa_S = 40$ jsou dosaženy výsledky $FAR = 2.98\%$, $FRR = 2.97\%$. Podobně jako v experimentu (A16), Tab. (7.2).6 ani zde nejsou použity další příznaky $f_{1,\dots,6}$, $f_{11,\dots,13}$ a $f_{14,\dots,16}$. Hodnota $fitness$ je počítána čistě na základě zarovnání a pozic jednotlivých pixelů kontur \mathcal{M} a \mathcal{S} .

Tabulka (7.2).7 Hodnoty FAR, FRR, EER ,
(A17) 5 prstů a žádný příznak, tj. f_7, f_8, f_9, f_{10} + palec + kontura ruky

| n. | κ_M | κ_S | ΣI_{UL} | FAR | ΣG_{OL} | FRR | EER | Tot. time |
|----|------------|------------|-----------------|-------|-----------------|-------|--------------|-----------|
| 1 | 1 | 1 | 1856 | 2.12% | 20 | 2.12% | 3324 / 2.12% | 5.319 |
| 2 | 1 | 5 | 1876 | 2.14% | 20 | 2.12% | 676 / 2.14% | 1.294 |
| 3 | 1 | 10 | 1894 | 2.16% | 21 | 2.23% | 345 / 2.16% | 0.764 |
| 4 | 1 | 15 | 2169 | 2.48% | 23 | 2.44% | 239 / 2.48% | 0.608 |
| 5 | 1 | 20 | 1967 | 2.25% | 22 | 2.34% | 181 / 2.25% | 0.483 |
| 6 | 1 | 25 | 2218 | 2.53% | 25 | 2.65% | 150 / 2.53% | 0.436 |
| 7 | 1 | 30 | 2304 | 2.63% | 23 | 2.44% | 127 / 2.63% | 0.405 |
| 8 | 1 | 35 | 2405 | 2.75% | 26 | 2.76% | 113 / 2.75% | 0.374 |
| 9 | 1 | 40 | 2613 | 2.98% | 28 | 2.97% | 100 / 2.98% | 0.343 |

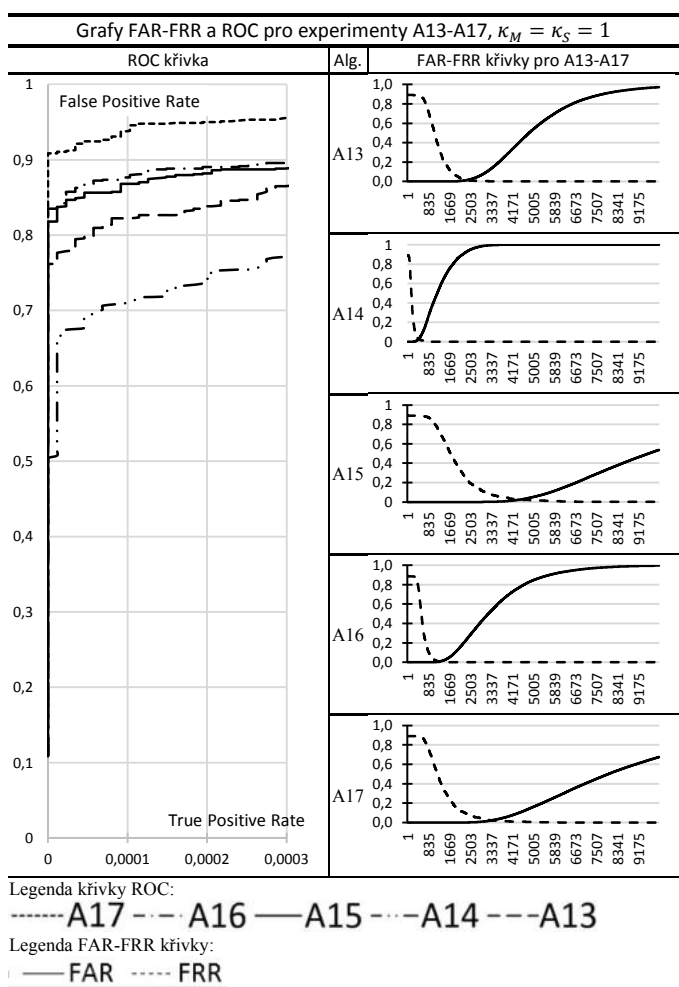
Tot. Time-udává celkový čas pro zarovnání jednoho vzorku v sekundách. Celkový počet evolucí je 9 řádků \times 87420 vzorků $S_{EA} = 786780$.

V Tab. (7.2).8 jsou ve formě shrnutí rozepsány nejlepší a nejhorší dosažené výsledky s ohledem na hodnoty κ_M, κ_S . Je poměrně dobře vidět, že ve všech případech, kde byly použity příznaky, je sice možné dosáhnout lepších výsledků pro $\kappa_M = 1, \kappa_S = 1$, ale o to horších výsledků, když se hodnoty κ_S zvyšují. Metoda, která zahrnuje zvolené příznaky je bohužel méně robustní.

Tabulka (7.2).8 Nejlepší a nejhorší výsledky *FAR*, *FRR*, *EER*,
pro vybrané hodnoty κ_M, κ_S a jednotlivé experimenty (A13) – (A17)

| n. | κ_M, κ_S | Experiment type | b/w | <i>FAR</i> % | <i>FRR</i> % | <i>EER</i> % |
|----|----------------------|----------------------------|----------|--------------|--------------|--------------|
| 1 | 1,1 | A13 + příznaky | nejlepší | 1.38% | 1.38% | 1.38% |
| 2 | 1,1 | A14, jen příznaky | nejlepší | 2.65% | 2.65% | 2.65% |
| 3 | 1,1 | A15 + příznaky | nejlepší | 2.55% | 2.55% | 2.55% |
| 4 | 1,1 | A16, žádné příznaky | nejlepší | 0.48% | 0.42% | 0.48% |
| 5 | 1,1 | A17, no příznaky | nejlepší | 2.12% | 2.12% | 2.12% |
| 6 | 1,40 | A13 + příznaky | nejhorší | 11.02% | 10.95% | 11.02% |
| 7 | 1,1 | A14, jen příznaky | nejhorší | 12.14% | 12.12% | 12.17% |
| 8 | 1,40 | A15 + příznaky | nejhorší | 9.76% | 9.78% | 9.76% |
| 9 | 1,40 | A16, no příznaky | nejhorší | 1.52% | 1.70% | 1.52% |
| 10 | 1,40 | A17, no příznaky | nejhorší | 2.98 | 2.97 | 2.98 |

b/w-nejlepší-nejlepší dosažené výsledky a b/w-nejhorší-nejhorší dosažené výsledky a vše s ohledem na κ_M, κ_S .



Obrázek (7.2).3 Křivky *FAR*, *FRR* a ROC pro jednotlivé experimenty.

Z jednotlivých výsledků v Tab. (7.2).3 - (7.2).7 a příp. i Tab. (7.2).8 a Obr. (7.2).3 je evidentní, že zvolené příznaky nepřinášejí podstatné vylepšení. Výsledky jsou díky nim spíše horší, i když jen o málo. Využití samotných příznaků podle Tab. (7.2).1 poskytuje nejhorší výsledky – viz Tab. (7.2).4. Paradoxně horší, než když jsou zvoleny hodnoty $\kappa_M = 1, \kappa_S = 40$ a porovnávány jen kontury. Je to dáno tím, že polohy jednotlivých prstů a také jejich natočení se při scanování do databáze GPDS samozřejmě měnilo. Velkým problémem je také to, že přesná pozice kloubů jednotlivých prstů není známa. Pozice kloubů se pouze odhadují. Bohužel i na různých snímcích stejné osoby není možné

odhadnout vždy identickou pozici kloubu. Důvodů je několik. Jednak snímky z databáze mají vždy odlišné nasvícení a pak také scanovaná ruka nebyla vkládána do scanneru vždy pod stejným úhlem vůči osvětlení a stín okolo ruky díky tomu koresponduje s barvou pozadí. Při takovémto uspořádání není možné odhadnout konturu zcela přesně. Jednou z možností by bylo zakreslit pro každý snímek z databáze GPDS konturu ručně a stejně tak i odhadnout pozice kloubů.

Na Obr. (7.2).3 jsou zobrazeny výsledné křivky FAR-FRR a ROC pro jednotlivé experimenty (A13) – (A17). Všechny křivky jsou uvažovány pro hodnoty $\kappa_M = \kappa_S = 1$. Křivky ROC jsou na ose X vyneseny jen v rozmezí $TPR \in (0.0, 0.0003)$. Nejlepší výsledky poskytuje sestavení (A16), kde jsou vzaty v úvahu jen kontury prstů bez palce a dalších příznaků. Naopak nejhorší výsledky byly dosaženy v experimentu (A14), kde jsou použity jen vybrané příznaky. Pokud je využita celá kontura ruky včetně palce, pak jsou výsledky velmi podobné uspořádání, které je zvoleno v experimentu (A16). Křivky ROC experimentů (A13), (A15), (A16), (A17) jsou si velmi podobné. Z praktického hlediska je poměrně lhostejné, zda byla dosažena přesnost $FAR\% = 0.48\%$ nebo $FAR\% = 12.14\%$, protože žádný z těchto klasifikátorů s takovými výsledky není použitelný pro citlivé aplikace v oblasti zabezpečení. Lze jej však využít všude tam, kde bezpečnost nehraje primární roli.

7.2.5 Závěr a diskuze

V této kapitole byly presentovány výsledky výzkumu klasifikace kontur lidské ruky s využitím různých kombinací geometrických primitiv tak, jak je to možné nalézt např. v (Barra et al. 2019). Současně byl brán ohled na výsledky dosažené v předchozích kapitolách, kde jsou použity především estimátory založené na porovnávání množin bodů, které náleží kontuře ruky. Nejlepší výsledky, jakých se podařilo dosáhnout, jsou dány hodnotou FAR v rozmezí $FAR\% = 0.48\% - 2.65\%$ při klasifikaci kontur s a bez využití dalších doplňujících informací podle zvoleného uspořádání. Největším problémem bylo odhadnout správné pozice kloubů pro jednotlivé prsty ruky a pak samozřejmě skutečnost, že databáze GPDS nebyla pořízena s příliš velkou pečlivostí. Nasvícení snímků je proměnlivé. Velmi důležitý faktor představuje také konstantní přítlak celé ruky včetně prstů na podložku scanneru. Pokud se mění, což je přirozené, pak se mění i kontura celé ruky. Faktorů, které ovlivňují výsledek, je značné množství. Postupný vývoj a technický pokrok v budoucnu zcela jistě umožní pořizovat lepší snímky, a to včetně snímků kostí a pozic kloubů bez využití rentgenu. Takové snímky by zajistili mnohem přesnější klasifikaci kontur. Rentgenové snímkování osob při každé autentifikaci samozřejmě není možné.

LITERATURA

- Abbass A.H. 2002. The Self-Adaptive Pareto Differential Evolution Algorithm. In: *Proceedings of the 2002 Congress on Evolutionary Computation*. vol. 1, pp. 831 – 836.
- Abbass H.A., Sarkar R., Newton C. 2001. A pareto differential evolution approach to vector optimization problems. In: *The IEEE Congress on Evolutionary Computation*. Seoul, Korea, pp. 971–978.
- Allegion 2020a. [online]. Allegion corp. ©2020 [cit. 8.4.2020]. Dostupné z: <https://us.allegion.com>
- Allegion 2020b [online]. Allegion corp. ©2020 [cit. 8.4.2020]. Dostupné z: <https://us.allegion.com/en/home/products/categories/biometrics.html>
- Bakshe R.C, Patil A.M. 2014. Hand Geometry Techniques: A Review. *International Journal of Modern Communication Technologies & Research*. vol. 2, no. 11, p.7
- Barra S., Marsico M.D., Nappi M., Narducci F., Riccio D. 2019. A hand-based biometric system in visible light for mobile environments. *Information sciences*. vol. 479, pp. 472-485.
- Bartlett M.S., Lades H.M., Sejnowski T.J. 1998. Independent component representations for face recognition. In: *Conference on Human Vision and Electronic Imaging III*. San Jose, California, vol. 3299, pp. 528-539.
- Besl P.J., McKay H.D. 1992. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vol. 14, no. 2, pp. 239-256.
- Bharathi S., Sudhakar R. 2011. Hand Biometrics: An Overview. *International Journal of Automated Identification Technology*. vol. 3, no. 2, pp. 101-108.
- Borra S.R., Reddy G.J., Reddy E.S. 2016. A broad survey on fingerprint recognition systems. *IEEE 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. ISBN: 978-1-4673-9338-6.
- Brest J., Greiner S., Boškovič B., Mernik M., Žumer V. 2006. Selfadapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*. vol. 10, no. 6, pp. 646-657.
- Brest J., Boškovič B., Greiner S., Žumer V., Maučec M.S. 2007. Performance comparison of self-adaptive and adaptive differentialevolution algorithms. *Soft Computing*. vol. 11, pp. 617-629.
- Brest J., Zamuda A., Fister I., Boškovič B. 2014. Some Improvements of the Self-Adaptive jDE Algorithm. In: *IEEE Symposium on Differential Evolution (SDE)*. pp. 1-8.
- Covavisaruch N., Prateepamornkul P., Ruchikachorn P., Taksaphan P. 2003. Personal Verification and Identification Using Hand Geometry. *ECTI Transaction on computer and information technology*. vol. 1, no.2, pp. 134-140.
- Das S., Mullick S.S., Suganthan P.N. 2016. Recent advances in differential evolution – An updated survey. *Swarm and Evolutionary Computation*. vol. 27, pp. 1-30.
- Das S., Suganthan P.N. 2010. Differential Evolution: A Survey of the State-of-the-Art. *IEEE Transactionson Evolutionary Computation*. vol. 15, no 1., pp. 4-31.
- Daugman J. 2004. How iris recognition works. *IEEE Transactions on Circuits and Systems for Video Technology*. vol. 14 no. 1, pp. 21-30.
- Dubuisson M.P., Jain A.K. 1994. A modified Hausdorff distance for object matching. In: *12th International Conference on Pattern Recognition*. pp. 566-568.
- Duta N. 2009. A survey of biometric technology based on hand shape. *Pattern recognition*. vol. 42, pp. 2797-2806.
- Eiben, A.E., Hinterding R., Michalewicz Z. 1999. Parameter control in evolutionary algorithms. *IEEE Transaction on evolutionary computation*. vol. 3, no. 2, pp. 124-141.
- Faundez-Zanuy M., Elizondo D.A., Ferrer-Ballester M.A., Travieso-González C.M. 2016. Authentication of Individuals using Hand Geometry Biometrics: A Neural Network Approach. *Neural Processing Letters*. vol. 26, pp. 201–216.

- Ferrer M.A., Morales A., Travieso C.M., Alonso J.B. 2007. Low Cost Multimodal Biometric Identification System based on Hand Geometry, Palm and Finger Textures. In: *41st Annual IEEE International Carnahan Conference on Security Technology*. pp. 52-58.
- Ferrer M., Vargas-Bonilla J., Morales A. 2011. BiSpectral Contactless hand based biometric identification device. In: *2nd National Conference on Telecommunications*. [cit. 11.5.2020] DOI: 10.5772/18096.
- Gamperle R., Muller S.D., Koumoutsakos P. 2002. A parameter study for differential evolution. In: *WSEAS International Conference on Advances in Intelligent Systems*. pp. 293-298.
- GPDS 2020 [online]. The Grupo de Procesado Digital de la señal, GPDS ©1992-2020 [cit. 8.4.2020]. Dostupné z: <http://www.gpds.ulpgc.es/>
- Hemery B., Mahier J., Pasquet M., Rosenberger C. 2008. Face Authentication for Banking. In: *First International Conference on Advances in Computer-Human Interaction*. [cit. 11.5.2020] DOI: 10.1109/ACHI.2008.17.
- Holland J.H. 1962. Outline for a Logical Theory of Adaptive Systems. *Journal of the ACM*. vol. 9, no. 3, pp. 297-314.
- Holland J.H. 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, p. 232, ISBN: 9780262082136.
- Holland J.H. 1992. *Adaptation in natural and artificial systems: an introductory analysis with application to biology, control, and artificial intelligence*. Cambridge, MIT Press, p. 211, ISBN: 9780262275552.
- Horn B.K.P. 1987. Closest form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*. vol. 4, no. 4, pp. 629-642.
- Charfi N. 2017. Biometric recognition based on hand shape and palmprint modalities. *Image Processing*. Ecole Nationale Supérieure Mines-Télécom Atlantique.
- Chauhan S., Arora A.S., Kaul A. 2010. A survey of emerging biometric modalities. *Procedia Computer science*. vol. 2, pp. 213-218.
- Iorio A., Li X. 2004. Solving rotated multi-objective optimization problems using differential evolution. In: *Australian Conference on Artificial Intelligence*. Cairns, Australia., pp. 861-872.
- ISO 2020 [online]. ISO ©2020 [cit. 8.4.2020]. Dostupné z: <https://www.iso.org/standard/43638.html>
- Jain A.K., Ross A., Pankanti S. 1999. A Prototype Hand Geometry-based Verification System. In: *2nd International Conference on Audio and Video based Biometric Person Authentication*. pp. 166-171.
- Jost T., Hügli H. 2002. Fast ICP Algorithms for Shape Registration. In: *Joint Pattern Recognition Symposium*. pp. 91-99.
- Kang W., Wu Q. 2014. Pose-Invariant Hand Shape Recognition Based on Finger Geometry. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. vol. 44, no. 11, pp. 1510-1521.
- Kennedy J., Eberhart R.C. 1995. Particle swarm optimization. In: *IEEE International Conference on Neural Networks*. vol. 4, pp. 1942-1948.
- Kumar A., Hanmandlu M., Kuldeep M., Gupta H.M. 2011. Automatic Ear Detection for Online Biometric Applications. In: *2011 Third National Conference on Computer Vision*. [cit. 11.5.2010] DOI: 10.1109/NCVPRIPG.2011.69
- Liu J., Lampinen J. 2002. On setting the control parameter of the differential evolution method. In: *8th International Conference Soft Computing*. pp. 11-18.
- Liu J., Lampinen J. 2005. A fuzzy adaptive differential evolution algorithm. *Soft Computing*. vol. 9, no. 6, pp. 448-462.
- Luque-Baenaa R.M., Elizondob D., Lopez-Rubioa E., Palomoa E.J., Watsonb T. 2013. Assessment of Geometric Features for Individual Identification and Verification in Biometric Hand Systems. *Expert Systems with Application*. vol. 40, no. 9, pp. 3580-3594.
- Maier-Hein L., Franz A.M., Santos T.R., Schmidt M., Fangerau M., Meinzer H.P., Fitzpatrick J.M. 2012. Convergent Iterative Closest-Point Algorithm to Accomodate Anisotropic and Inhomogenous Localization Error. *IEEE Transactions on pattern analysis and machine intelligence*. vol. 34, no. 8, pp. 1520-1532.

- Mallipeddi R., Suganthan P.N. 2010. Differential Evolution Algorithm with Ensemble of Parameters and Mutation and Crossover Strategies. In: *International Conference on Swarm, Evolutionary, and Memetic Computing SEMCCO 2010*. pp. 71-78.
- Metropolis N., Ulam S. 1949. The Monte Carlo method. *Journal of the American Statistical Association*. vol. 44, pp. 335-341.
- Montes E.M., Reyes J.V., Coello C.A.C. 2006, A Comparative Study of Differential Evolution Variants for Global Optimization, In: *GECCO '06 Proceedings of the 8th annual conference on Genetic and evolutionary computation*. pp. 485-492.
- Moravec J., Hub M. 2016. Automatic Correction of Barrel Distorted Images Using Cascaded Evolutionary Estimator. *Journal Information Sciences*. vol. 366, pp. 70-98.
- Neri F., Tirronen V. 2010. Recent advances in differential evolution: a survey and experimental analysis. *Artificial Intelligence Review*. vol. 33, no. 1-2, pp. 61-106.
- Noman N., Bollegala D., Iba H. 2011. An Adaptive Differential Evolution Algorithm. In: *IEEE Congress on Evolutionary Computation*. 2229-2236, New Orleans, USA. [cit. 11.5.2020] DOI: 10.1109/CEC.2011.5949891.
- Park G., Kim S. 2013. Hand Biometric Recognition Based on Fused Hand Geometry and Vascular Patterns. *Sensors*. vol. 28, no. 3, pp. 2895-2910.
- Parker J.R. 2010. *Algorithms for Image Processing and Computer Vision*. 2nd Edition, Wiley, ISBN: 978-0-470-64385-3.
- Pavlidis T. 1982. *Algorithms for Graphics and Image Processing*. Springer, p. 416, ISBN 978-3-642-93208-3.
- Pottmann H., Huang Q.X., Yang Y.L., Hu S.M. 2006. Geometry and convergence analysis of algorithms for registration of 3D shapes. *International Journal on Computer Vision*. vol. 67, no. 3, pp. 277-296.
- Price K. 1996. Differential Evolution: A Fast and Simple Numerical Optimizer. In: *NAFIPS'96*. pp. 524–527.
- Price K., Storn R. 1996. Minimizing the Real Functions of the ICEC'96 contest by Differential Evolution. In: *IEEE International Conference on Evolutionary Computation*. pp. 842–844.
- Price K., Storn R. 1997. Differential Evolution – A simple evolution strategy for fast optimization. *Dr. Dobb's Journal*. vol. 22, no. 4, pp.18-24+78.
- Price K., Storn R., Lampinen J.A. 2005. *Differential Evolution: A Practical Approach to Global Optimization*, Springer, p. 439. ISBN-13: 978-3642424168.
- Qin A.K., Suganthan P.N. 2005. Self-adaptive differential evolution algorithm for numerical optimization. In: *2005 IEEE Congress on Evolutionary Computation*. vol. 2, pp. 1785-1791.
- Ramirez-Cortes J.M., Gomez-Gil P., Prieto-Castro G.S.P.C. 2009. Shape-based hand recognition approach using the morphological pattern spectrum. *Journal of Electronic Imaging*. vol. 18, no. 1, pp. 013012-1 - 013012-6.
- Ramteke S.M., Hatkar S.S. 2013. A Survey on Security and Accuracy in Palmprint Recognition. *International Journal of Engineering Research & Technology (IJERT)*. vol. 2, no. 1, p. 6.
- Rechenberg I. 1973. Evolutionsstrategies: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. *Feddes repertorium*. Frommann-Holzboog Eds., Stuttgart, Germany, vol. 86, no. 5, pp. 337-337.
- Rechenberg I. 1994. *Evolutionsstrategie '94*, Frommann-Holzboog Ed., Stuttgart, ISBN 13: 9783772816420.
- Rodrigues M., Fisher R., Liu Y., 2002. Special issue on registration and fusion of range images. *Computer Vision and Image Understanding*. pp. 1–131.
- Rusinkiewicz S., Levoy M. 2001. Efficient variants of the ICP algorithm. In: *IEEE Third International Conference on 3-D Digital Imaging and Modeling*. p. 8.
- Santos-Sierra A., Casanova J.G., Avila C.S. 2009. Silhouette-based Hand Recognition on Mobile Devices. In: *43rd Annual 2009 International Carnahan Conference on Security Technology*. [cit. 11.5.2020] DOI: 10.1109/CCST.2009.5335548
- Sanches-Reillo S.R., Sanches-Avila S.C., Gonzales-Marcos A. 2000. Biometric identification through hand geometry measurement. *IEEE Transactions on pattern analysis and machine intelligence*. vol. 22, no. 10., pp. 1168-1171.

- Santos-Sierra A., Sánchez-Ávila C., Pozo G.B., Guerra-Casanova J. 2011. Unconstrained and Contactless Hand Geometry Biometrics. *Sensors*. vol. 11, pp. 10143-10164.
- Smit S.K., Eiben A.E. 2009. Comparing Parameter Tuning Methods for Evolutionary Algorithms. *IEEE Congress on Evolutionary Computation*. pp. 399-406.
- Smit S.K., Eiben A.E. 2010a. Parameter Tuning of Evolutionary Algorithms: Generalist vs. Specialist. *Applications of Evolutionary Computation*. vol. 6024, pp. 542-551.
- Smit S.K., Eiben A.E. 2010b. Parameter Tuning for Configuring and Analyzing Evolutionary Algorithms. *Swarm and Evolutionary Computation*. vol. 1, no. 1, pp. 19-31.
- Stockman G., Shapiro L. 2001. *Computer Vision*. Prentice Hall, p. 608, ISBN-13: 978-0130307965.
- Storn 2020. [online]. Storn R. ©1996-2020 [cit. 8.4.2020]. Dostupné z: <http://www1.icsi.berkeley.edu/~storn/code.html>
- Storn R., Price K. 1997. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*. vol. 11, no. 4, pp.341–359.
- Teo J. 2005. Exploring dynamic self-adaptive populations in differential evolution. *Soft Computing: A fusion of foundations, methodologies and applications*. vol. 10, no. 8, pp. 673-686.
- Travieso C.M., Alonso J.B., David S., Ferrer M.A. 2004. Optimization of a biometric system identification by hand geometry Complex systems. *Intelligence and modern technological applications*. Cherbourg, France, pp. 581-586,
- Vesterstrom J., Thomsen R. 2004. A comparative Study of Differential Evolution Particle Swarm Optimization and Evolutionary Algorithms on Numerical Benchmark Problems. In: *IEEE Congress on Evolutionary Computation CEC2004*. vol. 3 pp. 1980-1987.
- Wikipedia 2020a. [online]. Wikipedia ©2020 [cit. 8.4.2020]. Dostupné z: https://en.wikipedia.org/wiki/Linear_regression
- Wikipedia 2020b. [online]. Wikipedia ©2020 [cit. 8.4.2020]. Dostupné z: https://en.wikipedia.org/wiki/Differential_evolution
- Wolpert D.H., Macready W.G. 1997. No Free Lunch Theorem for Optimization. *IEEE Transaction on Evolutionary Computation*. vol. 1, no. 1, pp. 67-82.
- Xiong W., Xu Ch., Ong S.H. 2005. Peg-Free human shape analysis and recognition. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing*. [cit. 11.5.2020] DOI: 10.1109/ICASSP.2005.1415345
- Yan X., Su X.G. 2009. *Linear Regression Analysis: Theory and Computing*. World scientific, ISBN: 978-981-283-410-2.
- Yörük E., Konukoglu E., Sankur B., Darbon J. 2006a. Shape Based Hand Recognition. *IEEE Transactions on Image Processing*. vol. 15, no. 7, pp. 1803-1815.
- Yörük E., Dutagaci H., Sankur B. 2006b. Hand biometrics. *Image and Vision Computing*. vol. 24, pp. 483–497.
- Zayaraz G., Vijayalakshmi V., Jagadiswary D. 2009. Securing biometric authentication using DNA sequence and Naccache Stern Knapsack cryptosystem. In: *IEEE 2009 International Conference on Control, Automation, Communication and Energy Conservation*. Print ISBN: 978-1-4244-4789-3.
- Zhang J., Sanderson A.C. 2009. JADE: Adaptive Differential Evolution with Optional External Archive. *IEEE Transactions on Evolutionary Computation*. vol. 13, no. 5, pp. 945 – 958.
- Zhang W.J., Xie X.F. 2003. Hybrid Particle Swarm with Differential Evolution Operator. In: *IEEE International Conference on Systems, Man and Cybernetics*. vol. 4, pp. 3816-3821.
- Zhi-Peng F., Yan-Ning Z., Hai-Yan H. 2014. Survey of deep learning in face recognition. In: *2014 International Conference on Orange Technologies*. [cit. 11.5.2020] DOI: 10.1109/ICOT.2014.6954663.

7.3 Klasifikátor kontury ruky využívající 1-DIM optimalizační funkci

V minulých dekádách bylo navrženo a presentováno značné množství různých klasifikátorů. Přehled je možné nalézt např. (Duta 2009; Barra et al. 2019; Bera & Bhattacharjee 2017; Ferrer et al. 2007; Ferrer et al. 2011; Klonowski et al. 2018; Morales et al. 2008; Morales et al. 2010; Travieso et al. 2004; Travieso et al. 2012; Yörük et al. 2006a,b; Varchol & Levicky 2007) atd. Všechny metody využívají velmi podobné metody pro předzpracování obrazu ruky, která je obvykle snímána klasickým kancelářským scannerem nebo digitálním fotoaparátem. Pro klasifikaci jsou využity opět velmi podobné metody založené buď na statistických algoritmech např. Faktorová analýza – Lineární Diskriminanční Analýza (LDA) (Fisher 1936), neuronové sítě, Waveletova Transformace (WT), Diskrétní Kosinová Transformace (DCT) nebo skupiny geometrických primitiv, kde pro klasifikaci je využita jak L1-norma, tak i L2-norma příp. normy vyšších řádů. Využití metod nijak nesouvisí s tím, zdali je pro ustavení ruky a prstů použit nějaký adjustážní přípravek (tzv. pegs – kolíčky) nebo zdali je ruka volně položená v prostoru nebo na projekční desce scanneru.

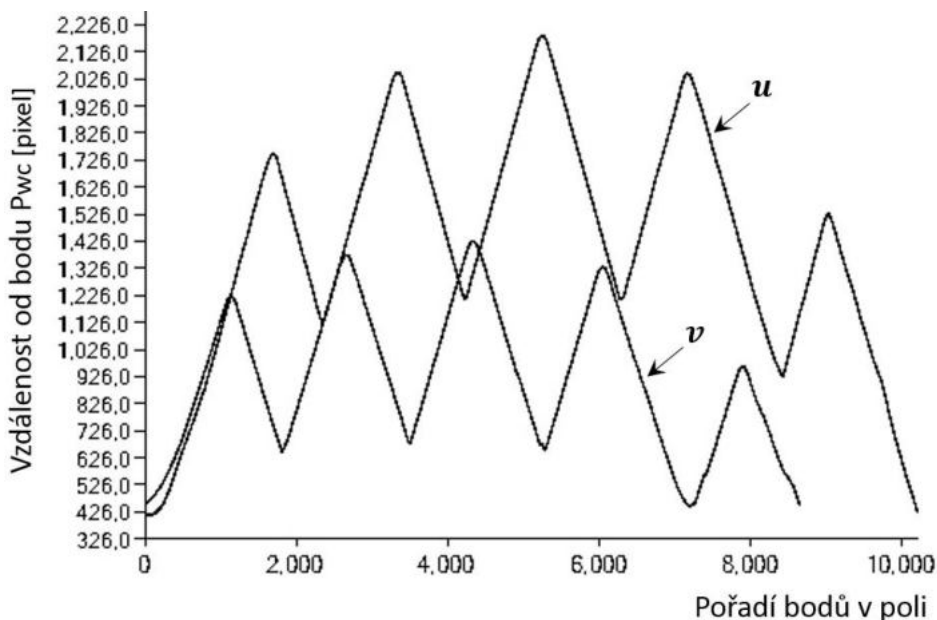
7.3.1 Úvod

V práci (Samiya et al. 2019) je popsán poměrně zajímavý způsob klasifikace kontury ruky, který transformuje 2D konturu ruky na jedno-dimenzionální křivku s využitím transformace v polárních souřadnicích, kde centrální bod je v oblasti zápěstí. Tento mechanismus je podobný vytvoření křivky Radial Distance Diagramu (RDD). 1-D kontura je vytvořena v rozsahu úhlu $\alpha = \left(-\frac{\pi}{3}; +\frac{\pi}{3}\right)$ od osy ruky, která je daná prostředníčkem a centrálním bodem ve středu zápěstí. Křivka je rozdělena na 100 intervalů a pomocí Fourierovy transformace jsou vypočítány koeficienty FFT v počtu 100. Pro další výpočet je využito jen 32 koeficientů – komplexních čísel (32 img. + 32 real.). Křivka RDD je chápána jako nízkofrekvenční signál. Klasifikovány jsou pak množiny koeficientů z FFT. Jsou využity dva klasifikátory a to Support Vector Machine (SVM) podle aplikace použité v publikaci (Bousquet 2001; Cornuéjols 2002) a neuronová síť tzv. Multi Layer Perceptron (MLP) podle aplikace v publikaci (Leray 2001). Nejlepšího výsledku bylo dosaženo pomocí SVM a to přesností identifikace 98 procent v samoučícím módu a 96 procent v testovacím módu a pro MLP pak 92 procent v samoučícím módu a 90 procent v testovacím módu. (Samiya et al. 2019) sice neuvádí hodnoty FAR, FRR, EER, ani není k dispozici kód pro veřejné ověření, ale „papírové“ dosažené výsledky jsou velmi solidní. Celá publikace (Samiya et al. 2019) je velmi krátká a bez jakýchkoliv bližších detailů. Je tak v podstatě nemožné zreplikovat danou metodu. Použitá databáze v (Samiya et al. 2019) je GPDS (GPDS 2020). Z databáze GPDS bylo použito jen 50 vybraných snímků.

7.3.2 Jednodimenzionální klasifikátor

V této kapitole bude podán pouze velmi krátký a spíše teoretický popis metody určené pro identifikaci osob, který je založený na jedno-dimenzionálním klasifikátoru, který zpracovává konturu ruky jako jednodimenzionální signál. Evoluční algoritmy použity nejsou. De-facto se jedná o algoritmus hrubé síly. Výpočetní náročnost je bohužel téměř shodná jako při použití evolučního algoritmu. Podobný přístup byl presentován např. v (Samiya et al. 2019). Není však využita žádná z metod jako např. Support Vector Machine (SVM) (Bousquet 2001; Cornuéjols 2002) nebo neuronová síť. Celý výpočet závisí čistě na modifikované konvoluční funkci. Je použita L1-norma. Algoritmus byl navržen čistě pro biometrický scanner popisovaný v této práci v kapitole 4 a základní podmínkou funkčnosti je použití sklopného přípravku s ustavovacími kolíky. Klasifikátor neumožňuje měnit při výpočtu úhel natočení prstů ani rotaci nebo posun celé kontury ruky.

Nyní bude proveden popis navrženého jednodimenzionálního klasifikátoru, který vychází z (Samiya et al. 2019) a primárně z (Moravec 2001; Moravec & Přeučil 2001), kde byl použit pro lokalizaci autonomní robotické platformy pomocí 2D laserového scanneru. Místo SVM a MLP jako je v (Samiya et al. 2019) je využita modifikovaná konvoluční funkce a při výpočtu klasifikačního kritéria je použita L1-norma místo L2-normy.



Obr. (7.3).1 Dva Radial Distance Diagramy (RDD) vytvořené z bodu P_{wc} na zápěstí ruky. Rozlišení obrazu je 2592x1944 pixelů. Při výpočtu se vektor \mathbf{v} posouvá vůči vektoru \mathbf{u} a hledá se minimum účelové funkce ε .

Základem metody je křivka Radial Distance Diagramu (RDD) – viz Obr. (7.3).1. V prvním kroku je nutné získat křivky RDD z obou kontur rukou \mathcal{M} a \mathcal{S} , které jsou vzájemně porovnávány. Křivka \mathcal{M} představuje referenční konturu ruky, která je uložena v databázi a křivka \mathcal{S} reprezentuje konturu ruky osoby, kterou identifikujeme. Křivka RDD se počítá z vhodného bodu na zápěstí, který budeme označovat jako P_{wc} . Tím jsou získány vektory \mathbf{u} a \mathbf{v} – viz Obr. (7.3).1, a také Alg. (7.3).1 bod 3. Pro účely porovnání je delší z vektorů \mathbf{u} a \mathbf{v} rozšířen o několik prvků zleva i zprava. Přidávané hodnoty jsou v_{max} – Alg. (7.3).1 bod 3. Je to z toho důvodu, že ruka nemusí být vždy vložena do skenovacího přípravku zcela stejně. Tím vzniknou vektory \mathbf{u}^* a \mathbf{v}^* – viz Alg. (7.3).1 bod 4. Prvky vektoru \mathbf{u}^* pak ponese označení $[a_1^*, a_2^*, \dots, a_p^*]$ a prvky vektoru \mathbf{v}^* budou označeny jako $[b_1^*, b_2^*, \dots, b_q^*]$. Z prvků vektoru \mathbf{u}^* vytvoříme matici \mathbf{C} a to tak, že na první řádek matice \mathbf{C} zapíšeme prvky a_i^* vektoru \mathbf{u}^* v pořadí, v jakém jsou zapsány ve vektoru \mathbf{u}^* . Na druhý řádek matice \mathbf{C} zapíšeme prvky vektoru \mathbf{u}^* tak, že první prvek vektoru \mathbf{u}^* rotujeme na poslední místo neboli $[a_2^*, \dots, a_p^*, a_1^*]$ a celý vektor \mathbf{u}^* zapíšeme do matice \mathbf{C} . Na třetí řádek matice \mathbf{C} zapíšeme rotovaný vektor \mathbf{u}^* z druhého řádku, u kterého opět posuneme první prvek a_2^* na poslední pozici $[a_3^*, \dots, a_p^*, a_1^*, a_2^*]$. Tímto způsobem postupujeme, dokud se poslední prvek a_p^* vektoru \mathbf{C} nedostane na první místo neboli $[a_p^*, a_1^*, a_2^*, \dots, a_{p-1}^*]$.

| Algoritmus (7.3).1 Výpočet koeficientu podobnosti ε u dvou RDD – dvou kontur rukou | |
|--|--|
| př. | Popis výpočetního kroku: |
| 1 | Vstup: Vektory \mathbf{u}, \mathbf{v} |
| 2 | Výstup: číslo ε |
| 3 | Vnitřní proměnné: Vektor \mathbf{z} Matice $\mathbf{C}, \mathbf{D}, \mathbf{E}$ Indexy n, m, p, q, t, i, j $\mathbf{u} = [a_1, a_2, \dots, a_n], a_i \in \mathbb{N}^0, i \in (1, n), n \in \mathbb{N}^+, n$ – počet prvků vektoru \mathbf{u} $\mathbf{v} = [b_1, b_2, \dots, b_m], b_j \in \mathbb{N}^0, j \in (1, m), m \in \mathbb{N}^+, m$ – počet prvků vektoru \mathbf{v} $l_{min} = \min(\mathbf{u}, \mathbf{v}) = \min(a_i, b_j) \Big _{i=1}^{i=n} \Big _{j=1}^{j=m}$ $l_{max} = \max(\mathbf{u}, \mathbf{v}) = \max(a_i, b_j) \Big _{i=1}^{i=n} \Big _{j=1}^{j=m}$ $e_{max} \in \langle 20, 100 \rangle$ – uživatelem volená hodnota, velikost offsetu |
| 4 | Výpočet: if $n \geq m$ $\mathbf{u}^* = [a_1^*, a_2^*, \dots, a_p^*] = [a_1^{ext}, a_2^{ext}, \dots, a_t^{ext}, a_1, a_2, \dots, a_n, a_1^{ext}, a_2^{ext}, \dots, a_t^{ext}]$ $\mathbf{v}^* = [b_1^*, b_2^*, \dots, b_q^*] = [b_1, b_2, \dots, b_m]$ $q = m$ $p = (2 \times t) + n, t \in \langle 1, e_{max} \rangle \text{ – viz element } a_p^* \text{ vektoru } \mathbf{u}^*$ $k = (2 \times t) + (n - m) + 1$ $a_i^{ext} \Big _{i=1}^{i=t} = l_{max}$ if $n < m$ $\mathbf{u}^* = [a_1^*, a_2^*, \dots, a_p^*] = [a_1^{ext}, a_2^{ext}, \dots, a_t^{ext}, b_1, b_2, \dots, b_m, a_1^{ext}, a_2^{ext}, \dots, a_t^{ext}]$ $\mathbf{v}^* = [b_1^*, b_2^*, \dots, b_q^*] = [a_1, a_2, \dots, a_n]$ $q = n$ $p = (2 \times t) + m, t \in \langle 1, e_{max} \rangle \text{ – viz } a_p^*$ $k = (2 \times t) + (m - n) + 1$ $a_i^{ext} \Big _{i=1}^{i=t} = l_{max}$ |
| 5 | $\mathbf{C} = \begin{bmatrix} c_{1,1} & \dots & c_{1,j} \\ \dots & \dots & \dots \\ c_{i,1} & \dots & c_{i,j} \end{bmatrix} = \begin{bmatrix} a_1^* & \dots & a_q^* \\ \dots & \dots & \dots \\ a_k^* & \dots & a_{q+k-1}^* \end{bmatrix}, i \in (1, k), j \in (1, q)$ $\mathbf{D} = \begin{bmatrix} d_{1,1} & \dots & d_{1,j} \\ \dots & \dots & \dots \\ d_{i,1} & \dots & d_{i,j} \end{bmatrix} = \begin{bmatrix} b_1^* & \dots & b_q^* \\ \dots & \dots & \dots \\ b_1^* & \dots & b_q^* \end{bmatrix}, i \in (1, k), j \in (1, q)$ |
| 6 | $\mathbf{E} = \mathbf{C} - \mathbf{D} , \text{ L1-norma}$ $\mathbf{E} = \begin{bmatrix} e_{1,1} & \dots & e_{1,q} \\ \dots & \dots & \dots \\ e_{k,1} & \dots & e_{k,q} \end{bmatrix} = \begin{bmatrix} c_{1,1} - d_{1,1} & \dots & c_{1,j} - d_{1,j} \\ \dots & \dots & \dots \\ c_{i,1} - d_{i,1} & \dots & c_{i,j} - d_{i,j} \end{bmatrix} = \begin{bmatrix} a_1^* - b_1^* & \dots & a_q^* - b_q^* \\ \dots & \dots & \dots \\ a_k^* - b_1^* & \dots & a_{q+k-1}^* - b_q^* \end{bmatrix}$ |
| 7 | $\mathbf{z} = \begin{bmatrix} z_1 \\ \dots \\ z_k \end{bmatrix} = [z_1 \quad \dots \quad z_k]^T = \left[\sum_{i=1}^{i=q} e_{1,i} \quad \dots \quad \sum_{i=1}^{i=q} e_{k,i} \right]^T$ |
| 8 | $\varepsilon = \min \mathbf{z} = \min [z_1 \quad \dots \quad z_k]^T = \min (z_i) \Big _{i=1}^{i=k}$ |

Legenda: \mathbf{u}, \mathbf{v} – vektory RDD obou klasifikovaných kontur. ε – reálné číslo udávající míru soběpodobnosti obou vektorů \mathbf{u}, \mathbf{v} ve smyslu použité metriky L1-norm. $\mathbf{u}^*, \mathbf{v}^*$ – pomocné proměnné vytvořené z vektorů \mathbf{u}, \mathbf{v} . \mathbf{C} – matice vytvořená postupnou rotací vektoru \mathbf{u}^* . \mathbf{D} – matice vytvořená z vektoru \mathbf{v}^* . \mathbf{E} – matice vytvořená jako $|\mathbf{C} - \mathbf{D}|$. \mathbf{z} – vektor vytvořený ze součtu hodnot na řádcích matice \mathbf{E} . ε – hledané optimum dané úlohy - minimální prvek vektoru \mathbf{z} .

Matice \mathbf{D} vznikne tak, že na jednotlivé řádky zapíšeme prvky vektoru \mathbf{v}^* bez jakékoliv další změny - viz Alg. (7.3).1 bod 5. Dalším krokem je výpočet matice \mathbf{E} , která vznikne jako absolutní hodnota z

rozdílu prvků matic C a D . Každý prvek matice E je vypočítán jako absolutní hodnota z rozdílu korespondujících prvků matice C a D neboli obecně zapsáno jako $|c_{i,j} - d_{i,j}|$ - viz Alg. (7.3).1 bod 6.

Dalším krokem je výpočet vektoru z . Každý prvek vektoru z vznikne jako součet prvků v korespondujícím řádku matice E - viz Alg. (7.3).1 bod 6. Vektor z má tedy stejný počet prvků, jako je řádků matice E . Posledním krokem je nalezení nejmenšího prvku vektoru z - viz Alg. (7.3).1 bod 7. Nejmenší prvek je pak označen jako ε neboli hledané optimum dané úlohy.

Celý algoritmus je poměrně jednoduchý a relativně rychlý, ale s ohledem na velké množství zpracovávaných dat je bez nějakého vhodného hardwarového akcelerátoru špatně použitelný jako real-time aplikace obzvláště na rozsáhlé databáze, kde je nutné provádět porovnání statisíců obrazů. Jestliže při použití evolučního estimátoru je doba nutná ke klasifikaci dvou vzorků v rozlišení 1280x960 pixelů cca 23 sekund, pak pro obrazy se stejným rozlišením je pro použití 1-Dim klasifikátoru dostačující čas méně než 1 sekunda. Typickým akcelerátorem vhodným pro získání hodnoty ε - viz Alg. (7.3).1 bod 7, jsou např. výpočetní karty Xilinx Alveo, multijádrové procesory fi. Cerebras nebo re-programovatelné maticové procesory apod. Klasické grafické karty NVidia/AMD se bohužel na výpočty tohoto typu příliš nehodí nebo je nutné, aby pracovaly společně s dalším typem akcelerátoru. Nevýhodou navrženého algoritmu je jednoznačně to, že ruka se při skenování musí vždy vložit do stavícího přípravku s kolíky velmi přesně. Posun v ose X nebo Y nebo rotace ruky v rovině XY na projekční desce biometrického scanneru vede k nepřesnostem při výpočtu a celkově tak snižuje účinnost metody. Stavící kolíky sice slouží jako doraz a omezují pohyb ruky v osách X a Y a také rotaci, ale jistá volnost v pohybu stále zůstává.

| | A | B | C | D | E | F | G | H | I | J | K |
|----|------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| 1 | | P000 | P001 | P002 | P003 | P004 | P005 | P006 | P007 | P008 | P009 |
| 2 | P000 | 0 | 399 190,3236 | 411 194,1792 | 409 020,1700 | 491 204,6493 | 652 636,5954 | 291 288,8404 | 201 851,0567 | 397 015,0254 | 304 549,3602 |
| 3 | P001 | 399 190,3236 | 0 | 688 399,8666 | 762 069,8403 | 180 716,0097 | 008 256,2628 | 210 538,3616 | 528 132,3276 | 164 966,5149 | 644 263,6223 |
| 4 | P002 | 411 194,1792 | 688 399,8666 | 0 | 512 561,2094 | 817 058,4143 | 554 999,4156 | 579 825,3447 | 007 694,9426 | 740 766,1129 | 493 552,7285 |
| 5 | P003 | 409 020,1700 | 762 069,8403 | 512 561,2094 | 0 | 794 985,2177 | 293 413,3349 | 632 956,4683 | 903 811,9976 | 710 318,2533 | 199 389,7200 |
| 6 | P004 | 491 204,6493 | 180 716,0097 | 817 058,4143 | 794 985,2177 | 0 | 056 948,5026 | 317 532,1783 | 599 506,8701 | 259 030,9634 | 674 674,9073 |
| 7 | P005 | 652 636,5954 | 008 256,2628 | 554 999,4156 | 293 413,3349 | 056 948,5026 | 0 | 878 727,2543 | 627 190,9069 | 972 506,4877 | 413 542,6982 |
| 8 | P006 | 291 288,8404 | 210 538,3616 | 579 825,3447 | 632 956,4683 | 317 532,1783 | 878 727,2543 | 0 | 404 558,6252 | 221 074,9615 | 525 200,6399 |
| 9 | P007 | 201 851,0567 | 528 132,3276 | 007 694,9426 | 903 811,9976 | 599 506,8701 | 627 190,9069 | 404 558,6252 | 0 | 528 455,1495 | 027 310,1755 |
| 10 | P008 | 397 015,0254 | 164 966,5149 | 740 766,1129 | 710 318,2533 | 259 030,9634 | 972 506,4877 | 221 074,9615 | 528 455,1495 | 0 | 590 450,7960 |
| 11 | P009 | 304 549,3602 | 644 263,6223 | 493 552,7285 | 199 389,7200 | 674 674,9073 | 413 542,6982 | 525 200,6399 | 027 310,1755 | 590 450,7960 | 0 |

Obr. (7.3).2 Výsledek klasifikace prvních 10ti obrazů kontur rukou z databáze THID. V jednotlivých buňkách tabulky je vypočítané minimum účelové funkce ε . Na diagonále jsou výsledky z porovnání stejných kontur. Konvoluční klasifikátor pracuje velmi dobře.

Vzhledem k tomu, že daný klasifikátor vyžaduje, aby pozice ruky v obraze byla vždy stejná a daná sklopným adjustážním přípravkem biometrického scanneru, bylo provedeno testování klasifikátoru jen s omezeným počtem 10ti snímků z databáze THID. 10 snímků od 10 různých osob, 1 snímek od každé osoby. Klasifikátor vypočítává hodnotu optima použitím pravidla každý-s-každým. Na Obr. (7.3).2 je dobře vidět, že klasifikátor vyhodnotil správně jednotlivé kombinace rukou – na diagonále jsou nuly, což znamená, že rozpoznal stejnou ruku při vzájemném porovnávání od kontury ruky cizího vzorku. Je nutné uvést, že kontury rukou nebyly pořízeny z biometrického scanneru a pozice každé ruky je na jednotlivých snímcích databáze THID samozřejmě nepatrně jiná. To do jisté míry ovlivňuje i výsledek. Na rozdíl od evolučního klasifikátoru účelová funkce ε dosahuje podstatně vyšších hodnot až o několik řádů. Použití rozlišení snímků bylo 2592x1944 pixelů. Výsledná hodnota $EER = 0.0\%$ je v tomto případě poněkud zavádějící, protože počet vzorků od každé osoby je roven jedné. Je to samozřejmě ideální hodnota, ale za cenu toho, že zkoumáme právě jeden jediný vzorek od každé osoby.

7.3.3 Závěr

Kromě vysoce efektivních n -dimenzionálních evolučních estimátorů, které byly použity v úvodní části sedmé kapitoly existuje bezpočet jiných metod, které umožňují provádět tutéž klasifikaci, ale jsou založeny na diametrálně odlišných principech. Jako příklad lze uvést Lineární Diskriminanční Analýzu (Faktorová Analýza, Fisherova analýza), nástroj Support Vector Machine, Neuronové Sítě apod. Za ideálních podmínek by mělo být možné dosáhnout všemi těmito metodami solidních výsledků, ale v reálném prostředí, kde existuje mnoho nahodilých nepříznivých vlivů nelze jednoznačně říci, že kvalitní výsledek bude dosažen. V případě klasifikace kontur rukou hraje velmi důležitou roli již samotná metoda pro získání obrazu ruky. Obvykle se používá digitální fotoaparát nebo scanner. Oba tyto nástroje však dávají ve výsledku výrazně odlišné snímky. Např. u snímků z klasického stolního scanneru bývá často problém s nespojitostí kontury po převodu do černobílé reprezentace. Mnoho autorů tak využívá možnost umělé opravy kontury na základě heuristických pravidel. Takto opravená kontura však nereprezentuje reálnou konturu, a to se pak projeví i na výsledcích. Na základě výzkumu (viz předchozí dvě kapitoly 7.1 a 7.2) bylo zjištěno, že se zvětšujícím se počtem snímků v databázi se snižuje i úspěšnost klasifikace. Především to platí u metod, které jako primární senzor používají kancelářský scanner nebo je snímkování prováděno např. fotoaparátem mobilního telefonu. Oním nahodilým a nechtěným faktorem je vždy parazitické světlo nebo naopak stín vržený rukou při nevhodném osvětlení. Jako další velmi nepříjemný faktor je to, že identifikovaná osoba nemusí vždy vložit ruku na scanner stejným způsobem. Nejhorší je, když prsty nejsou správně natažené nebo když je ruka rotovaná o náhodný úhel kolem všech tří os. Výhodou biometrického scanneru, který je prezentován např. fi. Allegion je to, že alespoň zčásti jsou všechny nahodilé jevy eliminovány i když scanner neřeší problém nenatažených prstů. Allegion používá stavící/adjustážní kolíky. U biometrického scanneru, který byl sestaven v rámci této disertační práce bylo dbáno na to, aby se soubor nahodilých jevů eliminoval v co největší míře. Biometrický scanner obsahuje fotografickou komoru s homogenním osvětlením bez parazitických světel z okolí. To pak umožňuje jako prvotní předzpracování RGB snímku ruky zařadit metodu jednoduchého prahování následovanou několika kroky filtrace digitálního obrazu. Výsledný předzpracovaný obraz pak vypadá tak, že ruka je např. černá a okolí bílé. Z tohoto obrazu lze již velmi snadno vypočítat velmi kvalitní konturu, která může být přímo vložena do klasifikátoru, a to jak jednodimenzionálního, tak i 9-ti dimenzionálního nebo případně vícedimenzionálního.

LITERATURA

- Barra S., Marsico M., Nappi M., Narducci F., Riccio D. 2019. A hand-based biometric system in visible mobile environments. *Information sciences*. vol. 479, pp. 472-485.
- Bera A., Bhattacharjee D. 2017. Human Identification Using Selected Features From Finger Geometric Profiles. *IEEE Transactions on system, man and cybernetics: Systems*. vol. 50, no 3, pp. 747 - 761.
- Bousquet O. 2001. Introduction to Support Vector Machines (SVM). Centre of Applied Mathematics, Ecole Polytechnique, Palaiseau Orsay.
- Cornuéjols A. 2002. A learning method: The vast margin separators SVM. *Bulletin of the AFIA*. University of Paris-Sud, Orsay, vol. 51, pp.14-23.
- Duta N. 2009. A survey of biometric technology based on hand shape. *Pattern recognition*. vol. 42, pp 2797-2806.
- Ferrer M.A., Morales A., Travieso C.M., Alonso J.B. 2007. Low Cost Multimodal Biometric Identification System based on Hand Geometry, Palm and Finger Textures. In: *41st Annual IEEE International Carnahan Conference on Security Technology*. pp. 52-58
- Ferrer M.A., Vargas F., Morales A. 2011. BiSpectral Contactless hand based biometric identification device. In: *2nd National Conference on Telecommunications*, DOI: 10.5772/18096.
- Fisher R. 1936. The use of multiple measures in taxonomic problems. *Annals of Eugenics*. vol. 7, no. 2, pp. 179–188.
- GPDS 2020 [online]. GPDS ©2020 [cit. 8.4.2020]. Dostupné z: <http://www.gpds.ulpgc.es/>
- Klonowski M., Plata M., Syga P. 2018. User authorization based on hand geometry without special equipment. *Pattern Recognition*. vol. 73, pp. 189-201.
- Leray P. 2001. Some Types of backpropagation Neural Networks. Technical report. Department ASI, PSI Laboratory, INSA Rouen.
- Morales A., Ferrer M.A., Díaz F., Alonso J.B., Travieso C.M. 2008. Contact-free hand biometric system for real environments. In: *16th European signal processing conference*. Print ISSN: 2219-5491.
- Morales A., Ferrer M.A., Kumar A. 2010. Improved palmprint authentication using contactless imaging. In: *2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. Washington. [cit. 11.5.2020] DOI: 10.1109/BTAS.2010.5634472
- Moravec J. 2001. Continuous Robot Localization in Known Environment Using Genetic Algorithms. In: *The 10th IEEE International Conference on Fuzzy Systems FUZZ IEEE 2001*. Melbourne, Australia, p.6, poster.
- Moravec J., Přeučil L. 2001. A Novel Approach to Accurate Mobile Robot Positioning. In: *Poster 2001: The 5th International Student Conference on Electrical Engineering*, CTU Poster 2001, Prague, p.30.
- Samiya S., Abderrahmane B., Abdelkader B. 2012. Recognition of individuals based on hand geometry. *Uncertainty Modeling in Knowledge Engineering and Decision Making*. pp. 1023-1029.
- Travieso C.M., Alonso J.B., David S., Ferrer M.A. 2004. Optimization of a biometric system identification by hand geometry. *Complex systems intelligence and modern technological applications*. Cherbourg, France, pp. 581-586.
- Travieso C.M., Briceo J.C., Alonso J.B. 2012. Transformation of hand-shape features for a biometric identification approach. *Sensors*. vol. 12, no. 1, pp. 987–1001.
- Varchol P., Levicky D. 2007. Using of hand geometry in biometric security systems. *Radioengineering*. vol. 16, no. 4, pp. 82–86.
- Yörük E., Konukoglu E., Sankur B., Darbon J. 2006a. Shape Based Hand Recognition. *IEEE Transactions on Image Processing*. vol. 15, no. 7, pp. 1803-1815.
- Yörük E., Dutagaci H., Sankur B. 2006b. Hand biometrics. *Image and Vision Computing*. vol. 24, pp. 483–497.

8 Dosažené cíle

8.1 Část první

Nyní budou popsány cíle, kterých se v průběhu výzkumu podařilo dosáhnout s ohledem na cíle vytyčené:

- I. V kapitole '3 - *Přehled historie a aktuální stav*' byl podán základní přehled v dané oblasti biometricky a především pak ve zkoumané oblasti identifikace osob využívající konturu lidské ruky.
- II. V kapitole '4 – *Biometrický scanner*' byl popsán návrh a praktická realizace biometrického scanneru, který umožňuje provádět snímkování rukou v temné komoře. Biometrický scanner je sestaven tak, že umožňuje snímkovat levou i pravou ruku. Ruka může být položena na skleněné podložce nebo může volně viset nad ní. Ruka se může opírat o sklopný adjustážní přípravek (tzv. pegs). Je možné provádět snímkování jak ve směru hřbetu ruky, tak i ve směru dlaně. Je možné snímkování kožních záhybů v dlani nebo na prstech ruky. Výhodou biometrické komory je zajištění výrazně dokonalejšího obrazu kontury ruky v porovnání např. se snímky z klasického stolního scanneru. Jsou omezeny parazitické odlesky, stíny, parazitické osvětlení.
- III. V kapitole '5 – *Evoluční algoritmy*' byl proveden základní popis používaných evolučních optimalizátorů, jejich algoritmická podoba a také soupis nejdůležitějších publikací, které se k danému algoritmu vztahují.
- IV. V kapitole '6 – *Kalibrace obrazu*' byly navrženy a publikovány nové metody (estimátory) umožňující kalibraci optické soustavy snímacího zařízení, které provádí snímkování na biometrickém scanneru. Byly navrženy celkem tři metody, které využívají evoluční optimalizátory a které spadají do všech tří základních kategorií označovaných jako back-mapping, direct-mapping a division-model. První a druhý typ estimátoru je navržen jako kaskádový. Třetí estimátor spadá do kategorie korekčních mechanismů, které se využívají spíše pro korekci obrazu zatíženého velkou distorzí např. u objektivů typu fish-eye. Typicky lékařské endoskopy. Algoritmy založené na modelu division-model spadají stále do oblasti intenzivního výzkumu. Bohužel nalezení lepších a výkonnějších korekčních metod není jednoduché.
- V. V kapitole '7 – *Klasifikace kontury ruky*' byla navržena nová metoda (estimátor) využívající pokročilé evoluční optimalizátory pro klasifikaci kontury lidské ruky. Jsou zde definovány 7, 8 a 9-ti dimenzionální estimátory, které řeší optimalizační úlohu s využitím pokročilého evolučního algoritmu nazývaného EPSDE a několika dalších EA. Počet dimenzí je dán počtem prstů, které jsou při výpočtu použity, resp. jejich konturami. Při výpočtu se samozřejmě jednotlivé prsty pohybují v korespondujících kloubech a pohybuje se i celá kontura ruky v ose X a Y s ohledem na těžiště kontury a celá kontura se natáčí vůči ose X podle potřeb evolučního optimalizátoru. Doplněna je také krátká kapitola, kde je využit jednodimenzionální L1-norm klasifikátor, který zpracovává obrazy z biometrického scanneru vybaveného adjustážními kolíky.

Všechny důležité poznatky a dosažené cíle byly publikovány v zahraničních recenzovaných periodikách, na zahraničních konferencích nebo formou technických zpráv. Obecně bylo ve všech případech dosaženo shody s cíli vytyčenými v rámci dostupných technických možností. Moderní biometrická identifikace osob je relativně mladá oblast, která se rozvíjí díky neustále se zvyšujícím nárokům na bezpečnost. Oblast identifikace osob s využitím kontury ruky je v posledních několika letech intenzivně řešená mnoha zahraničními výzkumnými týmy a počet publikací v dané oblasti velmi rychle roste. Vznikají veřejně dostupné biometrické databáze, které umožňují precizní porovnávání výsledků v mezinárodním měřítku.

8.2 Část druhá

Je zřejmé, že pro tuto vědeckou práci bylo nutné vytvořit ‘vhodné nástroje’ umožňující provádět jednotlivé experimenty a publikovat výsledky, a to jak s využitím softwaru, tak i s využitím kvalitního a nejmodernějšího hardwaru. V oblasti aplikovaných věd, bohužel, je povětšinou výsledek práce/výzkumu do jisté míry závislý na určité dávce finanční podpory. To je značná nevýhoda oproti čistě teoretickému výzkumu, kde jako nástroje stačí pouze tužka a papír. Nyní tedy budou uvedeny i další dosažené cíle, které se této práci týkají a které tvoří značnou část práce, resp. bylo nutné vyvinout značné a dlouhodobé pracovní úsilí k jejich naplnění.

- VI. Pro účely experimentů byl sestaven rozsáhlý softwarový balík, který umožnil administrovat veškeré prováděné experimenty předložené v jednotlivých kapitolách a díky kterému bylo možné tak značné množství dat efektivně zpracovávat, třídit a uchovávat po mnoho let. Softwarový balík se skládá z cca 50ti samostatných modulů. Některé z nich vzájemně spolupracují. Je umožněno snímkování s využitím navržené biometrické komory. Je umožněno předzpracování snímků, filtrace šumu, korekce distorze a samozřejmě i klasifikace s využitím čtyřech základních druhů estimátorů typu: jednodimenzionální L1-norm estimátor a dále pak 7, 8 a 9-ti dimenzionální L1/L2-norm evoluční estimátor. 7, 8 a 9-ti dimenzionální evoluční estimátory využívají pro svoji práci pokročilé evoluční algoritmy, které umožňují výrazně akcelarovat výpočty a dosáhnout tak potřebného výsledku v téměř reálném čase a především potřebné přesnosti, protože prostor možných řešení zpracovávaných úloh je obrovský.
- VII. Pro účely získávání obrazů kontur rukou byl navržen, prakticky sestaven a otestován biometrický scanner sestavený z dřevotřískových desek typu Durelis 12mm o rozměrech cca 60x45x100cm, váha cca 35Kg. Scanner umožňuje velmi přesné snímkování rukou v různých konfiguracích. Je možné použít sklopný přípravek s klasickými adjustážními kolíky (pegs), je možné snímkovat bez adjustážního přípravku. Je možné snímkovat jak ruku s roztaženými prsty, tak i ruku zaťatou v pěst nebo ruku s prsty u sebe (neroztaženými). Je také možné provádět snímkování záhybů kůže na ruce, podle toho, jak je nastavena poloha použitého digitálního fotoaparátu. Biometrický scanner má na sobě také náhledovou web kameru, která sama o sobě také umožňuje snímkování. Použitý digitální fotoaparát je možné připojit k počítači a katalogizovat a zpracovávat jednotlivé snímky pomocí navrženého softwarového balíku

8.3 Část třetí - závěry pro další rozvoj nebo praktickou realizaci

V této kapitole bude diskutována poměrně zajímavá tematika možného dalšího rozvoje navržených metod a závěrů pro praktickou realizaci. Nejprve se budeme věnovat tématice klasifikace kontur. Ve druhé části pak korekci aberací distorze.

Vědní oblast biometrické identifikace osob se v posledních dvou dekáдах stala velmi atraktivní a věnuje se jí mnoho výzkumných týmů po celém světě. Je to dáno nejen narůstajícími společenskými potřebami, ale také dostupností výkonnějších počítačů, které jsou dostatečně výkonné na zpracování náročných úloh v oblasti biometriky. Jako příklad praktické aplikace lze uvést identifikaci osob pomocí struktury krevního řečiště ruky v kombinaci s dalšími identifikačními prvky. Tento způsob identifikace je velmi efektivní a v praxi je využíván fi. Amazon, která si postup nechala patentovat a používá jej k identifikaci zákazníků při prodeji zboží. Mezi běžně používané postupy patří např. identifikace osob pomocí DNA, otisku prstů, otisk celé dlaně, struktury ušního boltce, ale také identifikace podle stylu chůze nebo písma. Již desítky let je využívání hlasová identifikace osob i když přesnost metody je poměrně variabilní. Nevýhodou mnoha biometrických ukazatelů je fakt, že je možné vytvořit poměrně jednoduše falsifikát. Typickým příkladem je kontrola přístupu pomocí otisku prstu na mobilních telefonech. Jiná situace nastává, když identifikovaná osoba je fyzicky k dispozici a je nutné ji identifikovat. Nevýhodou většiny biometrických ukazatelů je také skutečnost, že se s přibývajícím věkem osoby postupně mění. Některé biometrické ukazatele se mění značně a jejich význam z dlouhodobého hlediska je pak sporný.

Biometrická identifikace osob s využitím kontury ruky je velmi specifická. Na rozdíl od jiných biometrických ukazatelů jako je např. otisk prstu nebo struktura duhovky není tak přesná, a je při tom výpočetně značně náročná. Obzvláště pokud je nutné porovnávat zkoumaný vzorek se vzorky v rozsáhlejší databázi. Na základě výzkumů (viz detaily v popisované v předchozích kapitolách) bylo zjištěno, že jako vhodné prostředky při identifikaci osob na základě kontury ruky je možné použít statistické metody jako např. faktorová analýza. Stejnou přesnost lze dosáhnout i při použití klasifikace množin geometrických primitiv jako je šířka prstů, délka prstů a také přímou klasifikací kontury ruky a Euklidovské metriky. Poslední zmiňovaná metoda je náročná v tom, že vyžaduje, aby se při výpočtu jednotlivé kontury prstů pohybovaly. To klade nároky především na správný odhad pozic kloubů. Podstatnou výhodou do všech metod vnáší možnost vytvořit jakýsi „zprůměrovaný“ model ruky každé osoby sestavený z více snímků kontury ruky. To platí pro všechny zmiňované metody. Při experimentech s testovacími databázemi rukou bylo zjištěno, že významnou roli hraje také rozlišení originálního RGB snímku. Testovaná databáze THID poskytuje snímky v rozlišení 640x480 pixelů. Pokud jsou použity obrazy s tímto rozlišením, pak není možné dosáhnout dobrých výsledků na hranici 1.0-0.5 procenta EER nebo menší. RGB obrazy bylo proto nutné převzorkovat na rozlišení 1280x960pixelů s využitím bilineární transformace. Dalším důležitým faktorem je kvalita vstupního obrazu. Vstupní obraz poškozený mnoha stíny a náhodnými světelnými zdroji poskytuje velmi špatnou informaci pro další zpracování a klasifikaci. Jednoznačně je tedy důležité mít k dispozici kvalitní snímávací aparaturu a vhodnou fotografickou komoru bez okolního parazitického světla.

Výpočetní náročnost identifikace osob s využitím kontury ruky je daná především počtem zpracovávaných primitiv, což jsou v našem případě body kontury a samozřejmě i počtem snímků v databázi. Jak již bylo uvedeno v kapitole o výpočetní náročnosti lze říci, že úloha je shora ohraničená maximálním počtem bodů kontury. Nemá smysl výrazně zvyšovat počet bodů kontury nad hranici cca

11 tis. bodů, protože přesnost odhadu se již nezvýší. Je to dáno velikostí ruky dospělého jedince a především tím, že člověk se v průběhu let života mění. Nabírá na váze nebo naopak zhubne. To se projevuje i na velikosti kontury ruky.

Výrazným pomocníkem při výpočtech je samozřejmě klasický a cenově dostupný kancelářský počítač s jedno nebo i více-jádrovým procesorem. Stále se však jedná o sériové zpracování dat i když je úloha rozdělena na více procesorových jader. V dnešní době již existují poměrně výkonné a relativně dostupné výpočetní systémy jako jsou např. multiprocessorové systémy Cerebras (<https://www.cerebras.net/>), které mají 400000 výpočetních jader (46.225 mm² s 1.2 Trillion transistorů) nebo výpočetní karty Xilinx Alveo (<https://www.xilinx.com/products/boards-and-kits/alveo.html>), které jsou specializované právě na výpočty v oblasti strojového učení a hromadného zpracování dat. Takové systémy jsou ale velmi drahé pro běžné použití. V úvahu přichází i kombinace různých typů výpočetních karet. S ohledem na způsob výpočtu se na danou úlohu bohužel příliš nehodí grafické karty typu NVidia/AMD, protože jednotlivé výpočetní bloky nejsou vzájemně rekonfigurovatelné, není možné provést dynamickou relokaci spojů mezi nimi. K výpočtu je samozřejmě možné použít i superpočítače, ale cenová a energetická náročnost je poměrně vysoká. Je tu také aspekt, který není na první pohled viditelný, a to je doba, kterou je potřeba věnovat pochopení principů fungování specializovaných výpočetních systémů. Tato doba rozhodně není zanedbatelná, protože podobné výpočetní systémy jsou nesmírně komplikované.

Navržené algoritmy pro klasifikaci kontur rukou pracují s určitými vstupními parametry. Hodnoty těchto parametrů lze buď odhadnout nebo vypočítat. V našem případě se jedná o parametry fuzzy-heuristických pravidel, která usnadňují výpočet a klasifikaci a tvoří důležitý prvek, bez kterého by klasifikátor nefungoval. Pro „akademicky“ správné nastavení některých hodnot by bylo nutné nechat proběhnout výpočet celé použité databáze THID např. deset tisíckrát. Jedná se především o hodnoty M_{ul1} a M_{ul2} u klasifikátoru kontur rukou v kapitole 7.1. Dalšími parametry jsou např. počet jedinců v populaci použitého evolučního optimalizátoru a počet generací a také střední hodnota a směrodatná odchylka generátoru náhodných čísel u algoritmu EPSDE. Tady je již situace příznivější. Dostačující je např. 10 výpočtů celé databáze THID, což prakticky znamená 30 měsíců kontinuálního výpočetního času 24 × 7 při použití dostupného procesoru AMD6300 (6 jader).

Budeme-li hovořit o čistě o možnostech dalšího rozvoje biometrické identifikace s využitím kontury ruky, pak jednoznačně zvítězí cesta fúze dat z více informačních zdrojů např. kombinace otisku prstu a kontury ruky nebo krevního řečiště ruky a kontury ruky. To ovšem znamená mít k dispozici výkonnější hardware a lepší a specializované technické vybavení, které je velmi drahé. Zvýší se tím nejen výpočetní náročnost, ale také pracnost sestavení daných metod, která může hravě přesáhnout časové možnosti jednotlivce.

Prozatím byla diskuze v této kapitole věnována výhradně biometrické identifikaci osob, ale značnou část disertační práce tvoří i metody korekce aberací obrazu při snímkování za účelem klasifikace. Jedná se o korekci vady distorze (soudkové i polštářové). Zvolené korekční metody jsou výhradně určeny pro korekci radiální centrické vady distorze. Všechny aspekty zmiňované v předchozích odstavcích samozřejmě platí i v případě korekce distorze. Metoda samotná využívá evoluční algoritmy, které výrazně pomáhají k nalezení koeficientů polynomiálních rovnic. Opět jsou zde množiny různých pracovních parametrů, na jejichž správném nastavení závisí nejen přesnost výpočtu, ale především stabilita použitých algoritmů. Navržené metody pro korekci vad distorze spadají do oblasti poloautomatických korekčních metod. Na to, aby metoda fungovala musí uživatel poměrně přesně odhadnout/definovat korekční pozice kalibračních značek. Navržená metoda pak pouze dopočítá

nezbytné koeficienty korekčního polynomu. Existuje samozřejmě i možnost automatické korekce, ale ta je závislá opět na dostupném technickém vybavení. Mezi základní možnosti vylepšení navržené metody patří rozšíření max. 7-mi dimenzionální úlohy o další dimenze definující např. posun kalibračního obrazu v osách X, Y a jeho natočení vůči ose X. Tím se z úlohy stane 10ti dimenzionální optimalizační problém, který je samozřejmě výrazně více výpočetně náročný. Další možností je rozšíření úlohy o možnost úplné tří-dimenzionální prostorové korekce. Úloha se tak rozšíří o souřadné osy X,Y,Z a samozřejmě o úhly α, β, γ rotace okolo základních os XYZ. Počet dimenzí díky tomu naroste až na 13 a změní se tím i charakter úlohy, a především tvar definiční hyperplochy, nad kterou zvolený evoluční algoritmus pracuje/optimalizuje. To znamená, že evoluční algoritmus zvolený pro optimalizaci 7-mi dimenzionální úlohy, což je v našem případě jDE, již nemusí být efektivní pro zpracování mnohem těžší 13ti dimenzionální úlohy. Naopak, může být naprosto nevhodný. Časová náročnost nalezení nového efektivního optimalizátoru je bohužel značná a znamená to rozsáhlé testování různých typů derivátů evolučních algoritmů, kterých dnes existují tisíce.

Na rozdíl od spíše akademicky pojaté kapitoly klasifikace kontur rukou je oblast využití korekce distorze obrazu poměrně značná. Stačí připomenout jen oblast lékařství, kde korekce distorze u endoskopů tvoří důležitou součást moderního vybavení. Stejně uplatnění je možné najít i v oblasti bezpečnosti – korekce distorze z bezpečnostních kamer nebo také v oblasti vojenského výzkumu a vývoje. Další poměrně velkou oblastí jsou komerčně dostupné digitální fotoaparáty a mobilní telefony s fotoaparátem. Klasifikace kontur rukou s dynamicky proměnlivou konfigurací tvoří poměrně samostatnou kapitolu. Sama identifikace a klasifikace dynamicky proměnlivých struktur patří mezi velmi žhavé oblasti strojového vidění, ale její výpočetní náročnost stále zůstává za hranicemi běžně dostupného technického vybavení.

Z výše uvedeného poměrně snadno odvodíme, že aplikované metody umělé inteligence nepatří mezi jednoduché oblasti výzkumu. Na druhou stranu stále narůstá všeobecný hlad po nových metodách, které jsou schopné velmi efektivně řešit složité praktické úlohy, a to ve všech oblastech praktického života a samozřejmě také ve výzkumu. Technologický rozvoj a pokrok přináší nové a nové výzvy a pokud společnost nechce zaostávat, je nucena nalézat řešení, která nemusí být vždy jen triviální.

9 ZÁVĚR

Systémy automatické identifikace osob patří mezi velmi atraktivní témata současné vědy. Možností, jak identifikovat jedince s využitím biometrických ukazatelů jsou tisíce. Od jednoduchých antropometrických metod jako je např. měření rozměrů částí těla vhodným měřítkem, až po ty nejsložitější jako je např. analýza DNA, která vyžaduje velmi sofistikované laboratorní zařízení. Mezi běžnější metody patří např. analýza struktury sítnice nebo duhovky, tvar ucha, otisk prstu, otisk struktury dlaně, nebo také styl chůze, styl písma apod. Identifikace osob s využitím biometrických ukazatelů je z obecného hlediska poměrně značně výpočetně náročná. Obzvláště když je zkoumaný vzorek porovnáván se vzorky v nějaké větší databázi, což bývá téměř vždy.

V předkládané práci byl podán souhrnný přehled nejběžnějších senzorických systémů používaných v oblasti antropometrie, a to od prehistorických dob až po současnost. Největší rozmach ve zkoumané oblasti byl zaznamenán od začátku 80. let minulého století díky nástupu výpočetní techniky a většímu rozšíření cenově dostupných senzorických systémů. Je poměrně zajímavé, že po dlouhá tisíciletí lidských dějin nevznikla prakticky žádná písemně zdokumentovaná a zároveň sofistikovanější metoda, která by nějak definovala možnost, byť jen jednoduché antropometrické identifikace založené na obyčejném měřicím zařízení jako je metr, a to i přesto, že znalost, jak měřit délku nebo objem je lidem známa již několik tisíc let. Detailní analytický postup, jak využít měření při identifikaci osob byl popsán až v 19. století Alphonsem Bertillonem pro účely kriminalistiky a z biologického hlediska je znám popis kůže ze 17. století italským lékařem Marcellem Malpighinim. Malpighiniho postupy se však nevyužívaly pro nějakou hromadnou identifikaci osob. V dřívějších dobách se využívala především metoda porovnávání dvou obrazů pouhým okem bez hlubšího analytického popisu a rozsáhlé katalogizace založené na vědeckých poznatcích, jaký vytvořil např. Bertillon.

V rámci předkládané disertační práce byl navržen, sestaven a prakticky otestován biometrický scanner, jehož účelem je získávání kvalitních obrazů kontury ruky. Ruka může být položena na projekční desce nebo může být držena volně nad projekční deskou. Může mít roztážené prsty nebo může být i zaťatá v pěst. Možností různých konfigurací je poměrně velké množství. Primárním důvodem sestavení biometrického scanneru byla možnost získávat kvalitní obrazy ruky pomocí digitálního fotoaparátu, které by nevyžadovaly nesmírně komplikované metody filtrace před dalším zpracováním pro účely biometrické identifikace osob. Jako vhodná metoda předzpracování obrazu je plně dostačující pouhé prahování černobílého obrazu získaného z digitálního fotoaparátu, který je připojený na biometrickou komoru. Výsledek návrhu biometrického scanneru byl zpracován formou technické zprávy.

V rámci předkládané práce byla navržena, otestována a publikována nová metoda, která umožňuje provádět korekce aberací obrazu, který je získán z biometrického scanneru. Jedná se korekci vady distorze, a to soudkové i polštářové. Metoda spadá do oblasti poloautomatických metod a je založena na pokročilém evolučním optimalizátoru, který nese označení EPSDE. Metoda je navržena jako tzv. polynomiální korekční mechanismus v několika různých variantách. Konkrétně je využit postup označovaný jako direct-mapping a také back-mapping ve spojení s polynomiálními korekčními metodami s jednoduchým korekčním polynomem a také korekční metoda využívající lomený výraz označovaná jako polynomial-division-method nebo krátce division-method. Ve všech případech bylo dosaženo poměrně unikátních výsledků, které byly současně publikovány v zahraničních impaktovaných časopisech.

V rámci předkládané práce byla navržena, otestována a publikována nová metoda pro identifikaci osob. Tato metoda je založená čistě na strukturálních metodách rozpoznávání ve vědecké oblasti umělé inteligence a využívá pokročilý evoluční optimalizátor. Metoda využívá algoritmus označovaný jako ICP neboli Iterative Closest Point jako výpočetní jádro účelové funkce, která je dále zpracovávána evolučním algoritmem. Primární struktura, která je identifikována je tvořena konturou ruky sestavenou z jednotlivých bodů ve vhodném rastru např. 640x480 pixelů nebo 1280x960 pixelů s průměrným počtem bodů na konturu cca 1700-2300 pixelů podle rozlišení obrazu. Kontura má při výpočtu plně pohyblivé prsty, palec má 2 klouby. Celý evoluční estimátor je navržen v mnoha různých konfiguracích, které umožňují testovat konturu ruky zahrnující jen 4 prsty nebo i 5 prstů, palec s jedním nebo i se dvěma klouby. Ověření výkonnosti a přesnosti metody bylo provedeno na veřejně dostupných komparativních databázích označovaných jako THID, GPDS a HGDB. Primární výsledky poskytuje databáze THID. Výsledky experimentů byly porovnávány vůči výsledkům ostatních autorů a dosažená přesnost navržených metod se pohybuje v mezích, které jsou definovány pracemi jiných autorů, které v současné době (2020) reprezentují pomyslný state-of-the-art v dané vědní oblasti. Výsledky práce byly publikovány v zahraničním impaktovaném časopise.

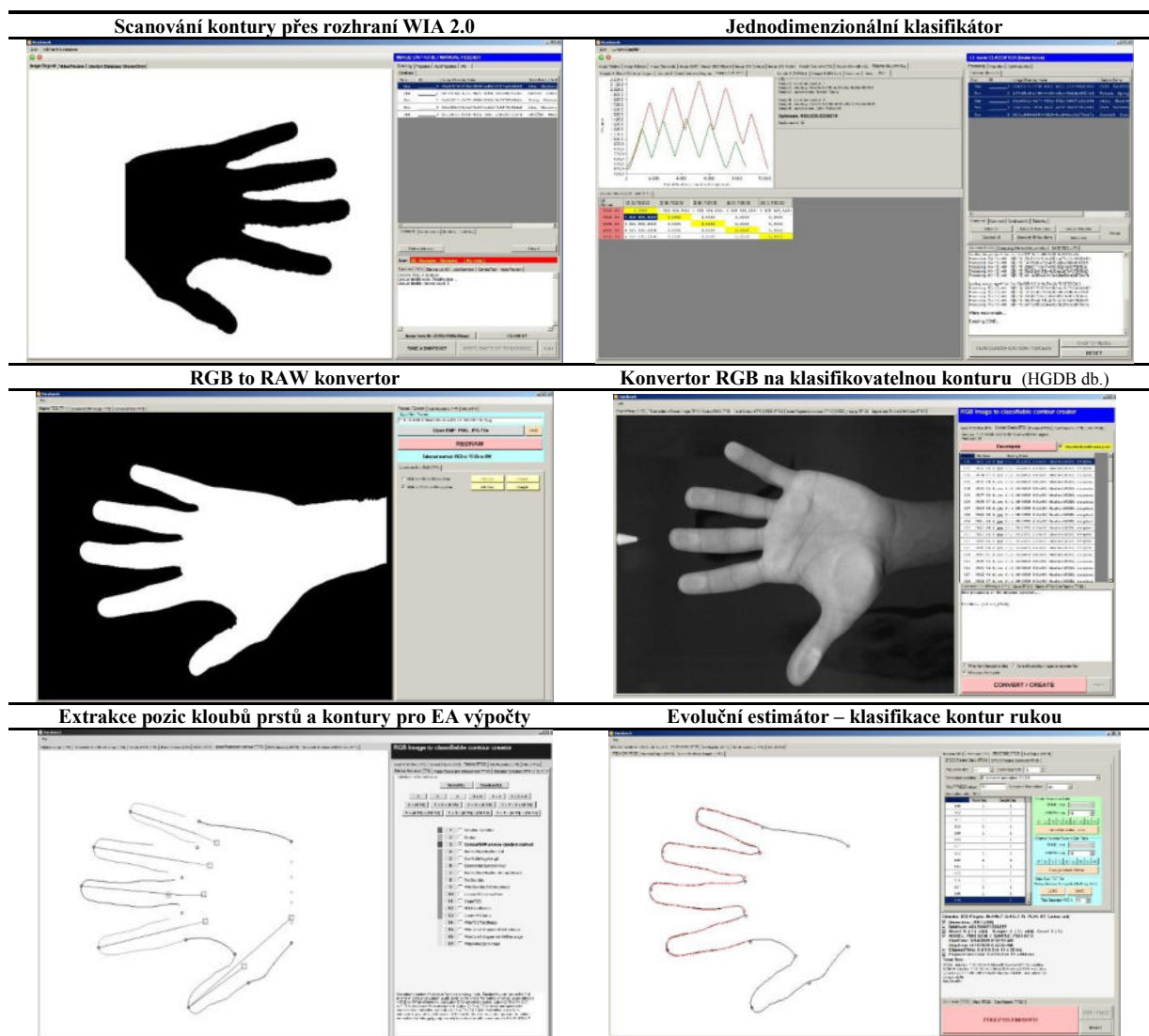
V rámci předkládané práce byla také navržena a pouze rámcově otestována metoda určená k identifikaci osob s využitím kontury ruky označovaná jako jednoduchý jednodimenzionální klasifikátor. Celá metoda je zkoumána spíše z teoretického pohledu; základní výzkum. Na rozdíl od metod využívajících poměrně sofistikované kombinace evolučních optimalizátorů a různých jiných gradientních iterativních metod je navržený jednodimenzionální klasifikátor založen na modifikované konvoluční funkci a Radial Distance Diagramu, který je vypočten přímo z 2D kontury ruky, která je reprezentovaná množinou bodů. Klasifikovaná kontura ruky je výhradně pětiprstá. Aby bylo možné zajistit kvalitní výsledky i při použití tohoto typu jednoduchého klasifikátoru, je nezbytné, aby ruka byla v biometrickém scanneru vložena do stavícího přípravku s kuličky (tzv. pegs). Takovýto přípravek je samozřejmě součástí navrženého biometrického scanneru.

Kromě výše uvedených výsledků byl také sestaven poměrně rozsáhlý softwarový balík, který umožnil provádět snímkování, předzpracování obrazů, jejich filtraci, katalogizaci, klasifikaci a samozřejmě i vyhodnocení výsledků podle nejnovější metodologie používané v oblasti strojového učení. Výsledkem jsou pak standardní statistické ukazatele označované jako FAR, FRR, EER, TPR, FPR, ROC a další, díky kterým bylo možné poměrně jednoduše rozhodnout u efektivnosti a úspěšnosti navržených evolučních estimátorů a porovnat výsledky s ostatními autory. Používané statistické ukazatele spadají do relativně nové metodologie označované jako „matice záměn“ neboli „confusion matrix“, která si v posledních cca 10-15ti letech teprve razí cestu na svět a do širšího podvědomí se dostává díky publikacím v oblasti strojového učení.

DODATKY

SOFTWAREVÝ BALÍK

Pro experimentální účely byl sestaven speciální software označený jako Handwork (HW), který umožnil provádět snímkování rukou v biometrickém scanneru přes připojený digitální fotoaparát, který podporuje knihovnu MS WIA, katalogizaci snímků, kalibraci optické soustavy digitálního fotoaparátu a odstranění aberací/vad distorze obrazu, předzpracování získaných digitálních snímků a filtraci šumu a samozřejmě i klasifikaci kontur rukou několika různými druhy evolučních estimátorů.



Celý komplex (HW) byl vytvořen s využitím programovacích jazyků MS-VS-C++/CLI 2008/2012 x32 .NET 3.5/4.0 WinForms + SlimDX + CenterSpace NMath .NET + Math .NET Iridium + Math .NET Numerics + MS WIA2.0 a MS-VS-C# 2019 x32 .NET 4.7.2 pro operační systémy Windows 7, 8.x, 10 x32/x64, primárně pro Windows 7 SP1 Profi x32/x64 ClassicTheme, pod kterým byl (HW) vyvinut. (HW) je kompletně 32bitový systém a převážná většina nástrojů v něm jsou jedno-vláknové aplikace. (HW) je vlastně takový chytrý nástroj určený pro zpracování dat v oblasti antropometrie se specializací na identifikaci osob využívající konturu ruky. Jedná se však o velmi specifický nástroj.

SEZNAM VLASTNÍCH PUBLIKACÍ

Seznam publikací, které se vztahují k této disertační práci.

Časopisecké publikace

- Moravec J., Hub M. 2016. Automatic Correction of Barrel Distorted Images Using Cascaded Evolutionary Estimator. *Journal Information Sciences*. vol. 366, pp. 70-98, DOI: <http://dx.doi.org/10.1016/j.ins.2016.03.055>
- Moravec J., Hub M. 2015. Camera Calibration Using Direct Mapping and Adaptive Metaheuristic. *International Journal of Bio-Science and Bio-Technology*. vol. 7, no. 3, pp. 111-120, DOI: [10.14257/ijbsbt.2015.7.3.11](https://doi.org/10.14257/ijbsbt.2015.7.3.11)
- Moravec J., 2020. Hand Contour Classification Using Differential Evolution Algorithm with Ensemble of Parameters and Mutation and Crossover. *Information Technology and Control*. vol. 49, no. 1, pp. 55-79, open access, DOI: <https://doi.org/10.5755/j01.itc.49.1.24140>
- Moravec J., 2020. A Polynomial-Division Based Correction Model for Camera Calibration: A Large Comparative Study. *Sadhana-Academy Proceedings in Engineering Science*. vol. 45, no. 92, p. 17, DOI: <https://doi.org/10.1007/s12046-020-1304-9>

Konference

- Moravec J. (2020) 7-Dimensional Optimization Task: PBO-Nature-Inspired Optimizer Versus 10-Years-Old Differential Evolution Based Optimizer 3rd Generation EPSDE. In: Silhavy R. (eds) Artificial Intelligence and Bioinspired Computational Methods. CSOC 2020. Advances in Intelligent Systems and Computing, vol 1225, pp. 11-25. Springer, Cham. https://doi.org/10.1007/978-3-030-51971-1_2
- Moravec J. (2020) Using Simple Genetic Algorithm for a Hand Contour Classification: An Experimental Study. In: Silhavy R. (eds) Artificial Intelligence and Bioinspired Computational Methods. CSOC 2020. Advances in Intelligent Systems and Computing, vol. 1225, pp. 93-109. Springer, Cham. https://doi.org/10.1007/978-3-030-51971-1_8
- Moravec J. (2020) A Hand Contour Classification Using Ensemble of Natural Features: A Large Comparative Study. In: Silhavy R. (eds) Artificial Intelligence and Bioinspired Computational Methods. CSOC 2020. Advances in Intelligent Systems and Computing, vol 1225, pp. 26-45. Springer, Cham. https://doi.org/10.1007/978-3-030-51971-1_3

Technické zprávy

- Moravec J. 2020. Biometrické identifikační systémy, výkresová dokumentace biometrického scanneru, Technická zpráva, UPCE FEI Pardubice 2020, p. 20.

SUMMARY

The scientific area of the biometric identification of persons became very attractive in last two decades. A lot of scientific teams around the world attending a big focus to solve contemporary problems in this branch. This phenomenon is given by increasing social demands and increasing power of commonly attainable computers which can be used in the area of biometric research. There is uncountable number of ways how to identify a person which use of the biometric markers. Unfortunately, the methods of the person identification with use of the biometric markers are from general point of view highly computation-demanding.

In presented thesis the summary of commonly used sensorial systems in the area of biometric identification was described since ancient history to present time. The biggest boom in this scientific area was recorded at very beginning of 80. of last century thanks to cheap computers and cheap sensorial systems. This trend is still visible and more and more scientist try to contribute to this very interesting area.

In frame of the thesis was proposed, assembled and practically tested a biometric scanner. The purpose of the biometric scanner is to obtain high quality images of a human hand contour. The biometric scanner contains a dark chamber which enables to obtain precious rgb images under conditions of homogenous light. The hand can lie on a projection desk or can be hold freely above the projection desk. The hand can be arranged with stretched fingers or the hand can be clenched in fist. There is a lot of different configurations. The primary reason of the biometric scanner creation was to obtain the high-quality images with use digital camera. Such images can be processed which use of the simple digital filtration methods such us thresholding. The thresholding of a digital image is very simple and fast method enabling to convert the rgb image to the black and white image, which can be used for next processing and classification.

A significant part of this thesis makes the methods of an image aberration correction namely distortion defect correction no matter if the defect is type of pillow or barrel. Selected correction methods were designed primarily to correct a radial centric defect of an optical system. The proposed methods use modern, advanced and effective evolutionary algorithms which enable to find the correct coefficients of the correctional polynomial equations. Used evolutionary algorithms utilize a set of working parameters. Correct setting of the working parameters is very important and effect the stability and final result accuracy of all used methods. Proposed methods fall into an area of semi-automatic correction methods. A user has to define very precisely the positions of the calibration marks and then the proposed correction method calculates the selected set of coefficients of the correctional polynomial. There is also the way of fully automatic correction method which, indeed, depends on very expensive and sophisticated technical equipment.

A second part of this thesis makes a biometric identification of persons with use of a human hand contour. Unlike many other biometric markers such as e.g. finger print, palmprint or iris structure the classification based on human hand contour is not such accurate. However, it is identically computational demanding like other methods. Especially if it is necessary to compare the scrutinized sample with a set of samples in a large database. Based on an extensive research it was ascertained that as the suitable technical means for person identification based on hand contour classification can be used a lot of different methods such as factor analysis, neural nets, simple geometric approach, generally L1-norm/L2-norm based methods etc. All these methods provide almost identical accuracy in classification. A method presented in here which utilizes the Euclidean metric provides very accurate results which can be compared with other state-of-the-art methods. A disadvantage of proposed method is that the individual fingers of a hand have to be fully moveable in computation and thank to that the positions of the individual knuckles have to be estimated precisely. A significant advantage of all methods is that an averaged model of a hand can be created with use of the set of images of the hand from the identical person. Such model represents “a better” hand from statistical point of view and is suitable for all methods mentioned above. In experiments with a comparative database it was ascertained that the key role plays the original image resolution. The tested comparative database THID provides original images in resolution 640x480 pixels. If such small resolution is used, then it is not possible to reach very good results which are represented by limits 1.0-0.5 percent EER or smaller. It was absolutely necessary to resample all images to higher resolution 1280x960 pixels with use the bilinear approximation. Next important factor is the quality of the input image. An input image with many shadows and random light sources provides very bad information for next processing and classification. It is very important to have a very good apparatus enabling to receive the images without unwilling defects. The classification process of the hand with dynamically variable reconfigurable contours makes standalone section of this thesis. The identification and classification of the dynamically reconfigurable structures belong to very heated areas of the machine learning, but computational demands still fall beyond the limits of standard technical equipment.

A big helper in all scientific calculations was a classic office computer with affordably price and multi-core processor. Such equipment represents more or less serial data processing system although the task is divided to more processor cores. There are, indeed, relatively very powerful computational systems e.g. multiprocessor systems Cerebras or computational cards Xilinx Alveo which are specialized to operate in the area of machine learning and bulk data processing. Such systems are very expensive for “everyday” use in very central Europe. It is possible to use a super-computer but the energy demands are very high. There is also an aspect which is not visible at the first glance and it is that the time necessary to understand how the specialized hardware is working is indispensable due to high complexity of such computational systems.

