

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Rozpoznávání registračních značek motorových vozidel
Matěj Černý

Diplomová práce
2010

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Matěj ČERNÝ**
Osobní číslo: **I08333**
Studijní program: **N2612 Elektrotechnika a informatika**
Studijní obor: **Komunikační a řídicí technologie**
Název tématu: **Rozpoznávání registračních značek motorových vozidel**
Zadávací katedra: **Katedra elektrotechniky**

Z á s a d y p r o v y p r a c o v á n í :

Cílem práce je vytvoření algoritmu obrazové detekce a rozpoznání znaků registračních značek motorových vozidel a jejich následná implementace do vestavěných zařízení s digitálními signálovými procesory a to za použití nižších programovacích jazyků. Součástí práce bude návrh algoritmu řešící samostatně problematiku snímání obrazových dat, vyhodnocení pozice vozidla, lokalizaci pozice registrační značky vozidla, segmentaci znaků a jejich rozpoznání. Pro tyto účely je možné použít jak deterministických postupů, tak postupů založených na aplikaci např. neuronových sítí. Pro vlastní návrh algoritmu je možné využít vhodné vývojové prostředí vyšší úrovně (např. programové prostředí Matlab s Image processing toolboxem), implementace algoritmu do cílové platformy by však měla být realizována např. v jazyce C, či C++.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování diplomové práce: tištěná/elektronická

Seznam odborné literatury:

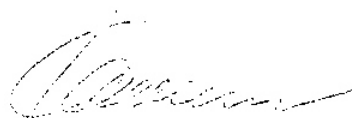
- [1] Dobrovolný M.: Rychlý algoritmus rozpoznání registračních značek vozidel, Perner's Contacts, Pardubice 2009, ISSN: 1801-674X
- [2] Dobrovolný M.: Aplikace Houghovy transformace pro kompenzaci rotace obrazu registračních značek vozidel, Seminář: Elektrotechnika a elektronika v dopravě, Univerzita Pardubice 2009, ISBN: 978-80-7395-194-8
- [3] Forsyth D., Ponce J.: Computer Vision: A Modern Approach, Prentice Hall, August 24, 2002, ISBN: 0-130851-98-1
- [4] Castleman K.: Digital Image Processing, Prentice Hall, September 2, 1995, ISBN: 0 132114 6754
- [5] Russ J.: The Image Processing Handbook, Fourth Edition, CRC Press LLC, London, 2002, ISBN:0-8493-1142-X
- [6] Klíma M., Bernas M., Hozman J., Dvořák P.: Zpracování obrazové informace, ČVUT Praha 1999, ISBN: 80-01-01436-3
- [7] Žák L.: Shluková analýza I.-III., Automatizace 3/47 5/47., Praha 2004

Vedoucí diplomové práce:

Ing. Martin Dobrovolný, Ph.D.
Katedra elektrotechniky

Datum zadání diplomové práce: 15. ledna 2010

Termín odevzdání diplomové práce: 21. května 2010



prof. Ing. Simeon Karamazov, Dr.
děkan

L.S.



Ing. Zdeněk Němec, Ph.D.
vedoucí katedry

V Pardubicích dne 31. března 2010

Prohlášení autora

Prohlašuji, že jsem tuto práci 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., autorský zákon, 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. Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 13. 9. 2010

Matěj Černý

Poděkování

Chtěl bych poděkovat vedoucímu mé diplomové práce Ing. Martinu Dobrovolnému, PhD., za cenné rady a připomínky při tvorbě této práce.

Anotace

Tato práce se zabývá problematikou rozpoznávání registračních značek motorových vozidel. Dále se zabývá implementací algoritmů obrazové detekce do vestavných zařízení s operačním systémem *uClinux* na platformě *Blackfin*.

Klíčová slova

Rozpoznávání registračních značek, openCV, Blackfin, uClinux

Title

Motor vehicle registration plates recognition

Annotation

This work deals with problems around motor vehicle registration plate recognition. Further is dealing with image detection algorithms implementation into embedded systems with operation system *uClinux* on *Blackfin* platform.

Keywords

Registration plates recognition, openCV, Blackfin, uClinux

Obsah

1 Úvod.....	11
2 Úvod do řešené problematiky.....	12
3 Analýza současných řešení z hlediska použitých algoritmů.....	14
3.1 Detekce vozidla.....	14
3.1.1 Hardwarová detekce.....	15
3.1.2 Softwarová detekce.....	16
3.2 Lokalizace registrační značky.....	17
3.2.1 Prahování jasu.....	17
3.2.2 Detekce hran.....	17
3.3 Segmentace znaků.....	18
3.3.1 Použití pevných hranic z rozměrů RZ.....	18
3.3.2 Použití vertikálního a horizontálního součtu.....	19
3.3.3 Použití spojených oblastí.....	20
3.4 Rozpoznání znaků	21
3.4.1 Porovnávání se šablonami.....	21
3.4.2 Použití neuronových sítí.....	21
4 Registrační značky v ČR.....	22
5 Rozbor vlastností navrhovaného systému.....	24
6 Návrh systému.....	26
6.1 Určení požadavků na rychlost systému.....	26
6.2 Pořízení dat.....	26
7 Popis navržených algoritmů obrazové detekce.....	32
7.1 Určení polohy vozidla v obraze.....	32
7.2 Lokalizace registrační značky.....	34
7.3 Segmentace jednotlivých znaků.....	37
7.4 Rozpoznání znaků.....	40
8 Návrh konkrétních řešení.....	42
8.1 Specifika návrhu.....	44
8.2 Postup nastavení PC pro cross- kompilaci.....	45
8.2.1 Nastavení a kompilace OpenCV.....	46
8.2.2 Nastavení IDE Eclipse.....	47
9 Alternativní postup detekce vozidla.....	51
10 Implementace.....	54
10.1 Určení polohy vozidla v obraze.....	55
10.2 Lokalizace RZ.....	56
10.3 Segmentace znaků.....	57
10.4 Rozpoznání znaků.....	58
11 Optimalizace.....	60
12 Zhodnocení.....	62
13 Závěr.....	64

Seznam zkratek

AVI	Audio Video Interleaved	- Komprimovaný video-formát
CAN	Controller Area Network	- Typ standardizované sběrnice
CCD	Charge Coupled Device	- Technologie snímacích čipů
DAC	Digital to Analog Converter	- Číslicově analogový převodník
DSP	Digitální Signálový Procesor	
HW	Hardware	
I/O	Input/Output	- Vstupně/výstupní
I2C	Inter-Integrated Circuit	- Typ standardizované sběrnice
IDE	Integrated Development Environment	- Integrované vývojové prostředí
IEEE	Institute of Electrical and Electronics Engineers	- Institut pro elektrotechnické a elektronické inženýrství
LAN	Local Area Network	- Místní síť
MB	Mega Byte	Megabajt
MMC	Memory Management Unit	- Jednotka ochrany paměti
OCR	Optical Character Recognition	- Optické rozpoznávání znaků
SAD	Sum of Absolute Differences	- Součet absolutních rozdílů
SW	Software	
UART	Universal Asynchronous Receiver/Transmitter	- Universální asynchronní přijímač/vysílač
USB	Universal Serial Bus	- Univerzální sériová sběrnice
VGA	Video Graphics Array	- Standard grafického rozhraní
WDM	Windows Driver Model	
RISC	Reduced Instruction Set Computer	Počítač s omezenou instrukční sadou
MD	Ministerstvo dopravy	

Seznam obrázků

Obrázek 1- Systémy úsekového měření rychlosti.....	12
Obrázek 2- Příklad Natočené RZ.....	19
Obrázek 3- Příklad zkorigované RZ.....	20
Obrázek 4- Použití spojených oblastí.....	20
Obrázek 5- RZ typ101, zdroj: MD ČR.....	22
Obrázek 6- RZ typ 151, zdroj: MD ČR.....	22
Obrázek 7- RZ typ 201, zdroj: MD ČR.....	22
Obrázek 8- RZ typ 301, zdroj: MD ČR.....	23
Obrázek 9- RZ vydávaná 2001 - 2004, zdroj: MD ČR.....	23
Obrázek 10- Nizozemská znaková sada.....	23
Obrázek 11- Starší RZ.....	23
Obrázek 12- Schéma celého systému.....	24
Obrázek 13- Basler Scout, zdroj: Basler.....	27
Obrázek 14- Nákres umístění snímací sestavy – boční pohled.....	28
Obrázek 15- Nákres umístění snímací sestavy – pohled shora.....	29
Obrázek 16- Objektiv MP1214-MP, zdroj: Computar.....	30
Obrázek 17- Fotografie snímacího stanoviště.....	31
Obrázek 18- Součet rádků.....	33
Obrázek 19- Oblast výřezu.....	34
Obrázek 20- Matice součtů oblastí.....	35
Obrázek 21- Výřez RZ.....	36
Obrázek 22- Součet sloupců (Modrá-součet sloupců, oranžová-střední hodnota, zelená- nalezené hranice RZ).....	36
Obrázek 23- RZ po vertikálním oříznutí.....	37
Obrázek 24- Postup segmentace znaku.....	37
Obrázek 25- RZ před ekvalizací (vlevo), po ekvalizaci (vpravo).....	38
Obrázek 26- Prahovaný obraz.....	38
Obrázek 27- Výřez bez horizontálních linií.....	38
Obrázek 28- Výřez připravený pro rozpoznání.....	39
Obrázek 29- Znak a) před úpravou , b) po úpravě	40
Obrázek 30- Šablony znaků.....	40
Obrázek 31- Průběh koeficientu SAD.....	41
Obrázek 32- Vývojový kit IPT-Shark537, a) pohled zdola , b) pohled shora.....	43
Obrázek 33- HW uspořádání kitu, zdroj: IP Thinking.....	43
Obrázek 34- Vývojové prostředí Eclipse.....	47
Obrázek 35- Instalace toolchainu Eclipse	48
Obrázek 36- Založení projektu Eclipse.....	48
Obrázek 37- Nastavení kompilátoru Eclipse.....	49
Obrázek 38- Nastavení linkeru Eclipse.....	50
Obrázek 39- Vývojový diagram SW detekce.....	51
Obrázek 40- Výchozí snímek s vyznačenou referenční oblastí.....	53
Obrázek 41- Vývojový diagram kompletního systému.....	55
Obrázek 42- Vývojový diagram určení polohy vozidla.....	56
Obrázek 43- Vývojový diagram lokalizace RZ.....	57
Obrázek 44- Vývojový diagram segmentace znaků.....	58
Obrázek 45- Vývojový diagram rozpoznání znaků.....	59
Obrázek 46- Vyjádření integrálního obrazu.....	61
Obrázek 47- Výstupní obraz.....	62

Seznam tabulek

Tabulka 1- Ujetá vzdálenost mezi snímky.....	27
Tabulka 2- Parametry videokamery.....	28
Tabulka 3- Parametry objektivu.....	29
Tabulka 4- Parametry vývojového kitu.....	44
Tabulka 5 - Parametry sériové komunikace.....	46
Tabulka 6 - Parametry detekce vozidla.....	52
Tabulka 7- Úspěšnost navrženého systému.....	62

1 Úvod

Cílem této práce je vytvoření systému automatického rozpoznávání registračních značek vozidel, vhodného například pro systémy úsekového měření rychlosti. Pro tyto účely je v práci analyzována problematika určení polohy vozidla, určení polohy registrační značky, segmentace jednotlivých znaků a jejich rozpoznání.

Rešerše zpracovaná v analytické části se stala východiskem pro návrh vlastních algoritmů obrazové detekce. Při návrhu algoritmů byl kladen důraz na snadnou implementaci v nižším programovacím jazyce a na nízkou výpočetní náročnost.

Druhá část práce se věnuje implementaci navržených algoritmů do vestavných zařízení s výkonným signálovým procesorem *Blackfin* firmy *Analog Devices*. Je zde také popsána problematika *cross-kompilace*¹ zdrojových kódů a knihoven funkcí pro *uClinux* na platformě PC a problematika spojená s vývojem aplikací pro operační systém *uClinux* na platformě *Blackfin*.

Druhotným cílem práce je ověřit možnosti využití knihoven *openCV* pro ostatní aplikace zpracování obrazu ve vestavných zařízeních používajících operační systém *uClinux*.

Cílem práce bylo také vytvořit referenční popis problematiky zpracování obrazu na zařízeních s vestavnými operačními systémy a problematiky s tím související.

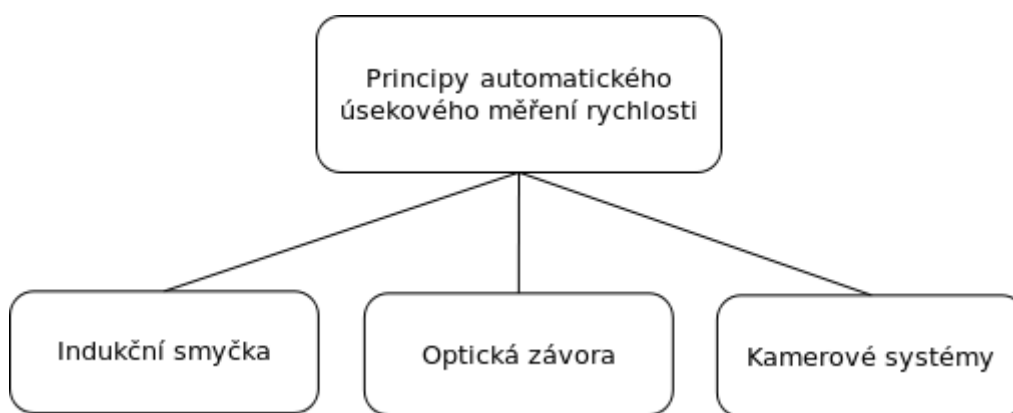
Nedílnou součástí práce je také popis procesu shromáždění vstupních dat, výběr vhodného snímacího zařízení a příprava dat pro vlastní postupy zpracování obrazu.

1 Křížová kompilace

2 Úvod do řešené problematiky

Systemy úsekového měření rychlosti zaznamenávají v současnosti veliký rozvoj, který je částečně spojen s pokračující výstavbou mýtných bran. Kromě těchto aplikací na rychlostních komunikacích se úsekové měření rychlosti rozšiřuje i ve městech a obcích, kde pomáhá snižovat rychlost vozidel a tím zvyšovat bezpečnost silničního provozu.

Princip úsekového měření rychlosti spočívá v měření času za který, vozidlo urazí pevně daný úsek. V současné době se pro úsekové měření rychlosti vozidel na pozemních komunikacích používají tři základní přístupy znázorněné na obrázku 1.



Obrázek 1- Systémy úsekového měření rychlosti

Prvním způsobem je použití indukčních smyček, které jsou umístěny pod vrchní vrstvou vozovky v přesně dané vzdálenosti. Vozidlo je detekováno na principu změny elektromagnetické indukce. Tento způsob je vhodný jen pro krátké vzdálenosti, čímž se blíží principům bodového měření rychlosti. Další nevýhodou je nutnost zásahu do vozovky což narušuje její integritu.

Druhým způsobem je použití minimálně dvou optických, resp. laserových závor, umístěných v přesné vzdálenosti. Vozidlo je detekováno při přerušení závory. Nevýhodou je možnost použití pouze na krátkých úsecích v řádech metrů a možnost použití pouze na jednoproudých úsecích. Tím se výrazně omezuje možnost použití tohoto systému pro úsekové měření rychlosti.

Třetí možností je využití kamerového systému. Stěžejní částí této metody je systém rozpoznávání vozidel. Vozidla je možné rozpoznávat množstvím způsobů, od obsluhy, která provádí spárování vozidel ručně až po využití plně automatických systémů

rozpoznávání vozidel. Vzhledem k technickým možnostem jsou v dnešní době využívány převážně automatické systémy.

Na začátku a na konci měřeného úseku je zaznamenáno vozidlo společně s přesným časem průjezdu a po spárování odpovídajících vozidel je z času vypočtena průměrná rychlost v úseku. Pro spárování vozidel se v současnosti nejčastěji využívá rozpoznání registrační značky vozidla. Tuto metodu je možné použít na dlouhém úseku i několika kilometrů, protože vozidla mohou libovolně přejíždět mezi jízdními pruhy. Tím se efektivně zpomaluje provoz po celé délce úseku a ne jen v oblasti měření. Tato metoda umožňuje zároveň předávat registrační značky vozidel vyšším systémům, které mohou např. pátrat po odcizených vozidlech. Tato metoda je v současné době nejrozšířenější, předchozí dvě možnosti jsou pro své nedostatky používány velmi málo a spíše jako doplňkové[16].

3 Analýza současných řešení z hlediska použitých algoritmů

Cílem práce je návrh obrazového systému rozpoznávání registračních značek vozidel. Tyto systémy je možné rozdělit do několika, na sebe navazujících částí. Tyto části jako celek vedou k úspěšnému rozpoznání registrační značky vozidla. Prakticky všechny současné systémy pro rozpoznávání registračních značek obsahují čtyři základní části:

1. Detekce vozidla.
2. Lokalizace registrační značky.
3. Segmentace znaků.
4. Rozpoznání znaků (OCR²).

Prvním problémem je určení, zda se ve snímaném záběru vyskytuje vozidlo. Čím přesnější podává tato fáze výsledky, tím jsou menší rozdíly pozice vozidel v obraze. Při menších rozdílech je snazší lokalizovat registrační značku.

Jako další je nutné určení pozice registrační značky. Přesnost lokalizace výrazně ovlivňuje nutnost provádění dalších úprav a úspěšnost správné segmentace znaků. Při lokalizaci registrační značky se využívá znalosti jejího rozměru. Úspěšnost lokalizace registrační značky také ovlivňuje počet pokusů nutných k její správné lokalizaci a tím i celkovou rychlost systému.

Následná segmentace znaků je klíčová pro všechny metody jejich rozpoznání. Základem přesné segmentace je přesné určení polohy registrační značky. K zvýšení přesnosti segmentace je možné využít znalosti pozice a rozměrů znaků na registrační značce. Přesnost segmentace výrazně ovlivňuje úspěšnost následného rozpoznání znaků.

V poslední fázi probíhá vlastní rozpoznání znaků. Vstupem do této fáze jsou obrazy segmentovaných znaků a výstupem je textové vyjádření těchto znaků. Pro úspěšnou funkci celého systému je potřebná velmi vysoká procentní úspěšnost rozpoznání.

Různé systémy se však odlišují v realizaci jednotlivých částí, s tím se také liší jejich rychlost, úspěšnost a v neposlední řadě cena.

3.1 Detekce vozidla

V této části můžeme postupy rozdělit na dva základní přístupy, prvním je použití hardwarové detekce a druhým je softwarová detekce. Oba přístupy mají své klady i zápory.

2 Optical Character Recognition

Při rozhodování, který z přístupů použit je nutné zvážit hlavně prostředky, které jsou k dispozici. Každý z přístupů klade rozdílné požadavky na hardwarové vybavení systému.

3.1.1 Hardwarová detekce

Při hardwarové detekci vozidla je oblast snímána jen v okamžiku, kdy systém dostane od detektoru signál, že je v záběru přítomné vozidlo. To umožňuje použití kamery s nižší snímkovací frekvencí a použití zařízení s nižším výpočetním výkonem. Na druhou stranu je ale požadováno další zařízení, což zvyšuje cenu systémů. Nejobvyklejší hardwarové způsoby detekce vozidla jsou:

- Detekce elektromagnetickými indukčními cívkami,
- detekce piezoelektrickými tlakovými senzory,
- detekce optickou závorou (infračervená, laserová),
- detekce dálkoměry (laserové, ultrazvukové).

Detekce elektromagnetickými cívkami využívá principu změny elektromagnetické indukce při přejezdu vozidla nad smyčkou, která je umístěna ve svrchní části vozovky. Hlavní nevýhodou této metody je nutnost stavebního zásahu do vozovky a tím její narušení. Výhodou tohoto systému je jeho jednoduchost a spolehlivost.

Při detekci piezoelektrickými tlakovými snímači je využíváno jejich schopnosti při deformaci měnit hodnotu napětí. Piezoelektrické senzory se umísťují do svrchní vrstvy vozovky napříč jízdním pruhem a reagují na změnu zatížení při přejezdu vozidla. Nevýhoda tohoto systému, stejně jako u elektromagnetických cívek spočívá v nutnosti zásahu do vozovky.

Při detekci optickou závorou je napříč jízdním pruhem vysílán laserový nebo infračervený paprsek, který je na druhé straně vozovky odrážen zpět k vysílači, kde je umístěn i detektor. Při přerušení tohoto paprsku je snímacímu systému předána informace, že je v záběru přítomno vozidlo. Nevýhodou tohoto řešení je, že je možné ho instalovat jen na jednoproudé úseky.

Detekce pomocí dálkoměrů je v současné době nejrozšířenější metodou. Dálkoměry se umísťují nad vozovku, nejčastěji na silniční brány nebo na stožáry. Dálkoměry jsou namířeny směrem dolů k vozovce, kde má být vozidlo detekováno. Při nepřítomnosti vozidla je měřená vzdálenost konstantní, pokud do záběru vjede vozidlo vzdálenost se sníží. Tento způsob je díky své přesnosti a jednoduchosti instalace nejvhodnější pro použití

v navrhovaném systému. Jelikož ale dálkoměr při tvorbě této práce nebyl k dispozici, byla detekce pro účely této práce řešena softwarově.

3.1.2 Softwarová detekce

Při softwarové detekci vozidla se využívá zpracování obrazu, proto je třeba snímat scénu kontinuálně. Díky tomu jsou kladeny vysoké nároky také na snímkovací frekvenci kamery, tím velmi narůstá množství zpracovávaných dat a nároky na výpočetní výkon. Výhodou je, že není nutné systém doplňovat o další zařízení. Je zde ale větší náročnost na samotnou aplikaci, protože je nutné řešit detekci vozidla softwarově. To lze nejjednodušeji řešit třemi způsoby:

- Porovnávání snímků s referenční oblastí.
- Separace statického pozadí[18].
- Optický tok [20].

Při porovnávání snímků s referenční oblastí jsou parametry referenční oblasti porovnávány s parametry stejných oblastí v následujících snímcích. Nejčastěji je porovnávána střední hodnota jasu v oblasti. Nevýhodou této metody je nutnost uložení referenční oblasti bez vozidla a náchylnost na změny světelných poměrů v průběhu dne. Náchylnost na změny světelných poměrů je možné odstranit periodickým snímáním referenční oblasti. Největší výhodou této metody je její jednoduchost. Proto byla v této práci tato metoda zvolena jako náhrada hardwarové detekce.

Při separaci pozadí je sečteno několik po sobě jdoucích snímků a je vypočten medián každého obrazového bodu. Takto získaný obraz obsahuje pouze pozadí bez ohledu na to, zda je na snímku vozidlo nebo není. Detekce se provádí porovnáním aktuálního snímku se separovaným pozadím. Tato metoda je velmi efektivní, ale zpracovává větší množství dat než předchozí metoda. Vzhledem k tomu, že softwarová detekce je zde použita pouze jako náhrada hardwarové detekce, byl zvolen předchozí postup pro svou jednoduchost.

Při použití optického toku se algoritmus snaží vyhledat v po sobě jdoucích snímcích směr pohybu jednotlivých obrazových bodů nebo oblastí. Pokud je vozidlo v obraze bude výrazný optický tok v oblasti vozidla. Nevýhodou tohoto postupu je vysoká výpočetní náročnost, tu je možné snížit např. pomocí metody Lucas-Kanade [2].

3.2 Lokalizace registrační značky

Vzhledem k tomu, že pro lokalizaci registrační značky je možné použít velké množství přístupů, v této části se většina systémů liší. Dvěma hlavními směry jsou:

- Prahování jasu a
- detekce hran.

3.2.1 Prahování jasu

Předpokladem pro použití této metody je, že rozdíl jasu mezi registrační značkou a jejím okolím musí být dostatečně veliký, což je podmínka, kterou není možné spolehlivě splnit. Uvažujeme-li české registrační značky, problém by se vyskytl u vozidla bílé barvy, kde by rozdíl jasu mezi registrační značkou a okolím (karoserií vozu) byl tvořen pouze černým ohraničením registrační značky, které je poměrně tenké a zvláště při horší kvalitě záznamu nemusí být zřetelné. Tato metoda se při tvorbě práce neosvědčila, proto nebyla použita.

Tato metoda je popsána např. v [11], nebo [15], kde je po prahování provedena morfologická dilatace vhodným strukturním elementem, která vyplní oblasti znaků registrační značky a takto získaná oblast je označena jako oblast registrační značky. Následně jsou oblasti testovány podle velikosti a nevyhovující oblasti jsou odstraněny. V [7] se pro detekci využívá blízkosti jednotlivých znaků a morfologické uzavírání v horizontálním směru, čímž je docíleno jejich slití. Následujícím morfologickým otevíráním vhodným strukturním elementem o velikosti poloviny výšky registrační značky jsou nevyhovující oblasti odstraněny a zůstává pouze oblast registrační značky.

3.2.2 Detekce hran

Segmentace pomocí detekce hran má univerzálnější využití, které je ale vykoupeno vyšší výpočetní náročností. V hranovém obraze je oblast registrační značky hledána několika způsoby. V [5] jsou nejdříve v hranovém obraze extrahovány uzavřené hranice a následně jsou pomocí Houghovy transformace vyhledávány dvě paralelní hrany, které odpovídají rozměru registrační značky.

Další možností je v hranovém obraze vyhledávat maximální shodu s předem definovanou šablonou [4]. Pro tento postup je nutné, aby byly záběry čisté, a aby byla velikost RZ^3 v obraze konstantní.

3 Registrační značka

Další přístupy většinou využívají toho, že registrační značka obsahuje velké množství vertikálních hran. V [6] je k určení polohy použit součet jednotlivých řádků, ve kterém je vyhledáván největší rozdíl mezi řádky, tím jsou určeny horizontální hranice registrační značky. Následně je v této oblasti proveden součet sloupců a vypočten aritmetický průměr. Místa, kde součet překračuje aritmetický průměr, jsou označena jako možné oblasti registrační značky a na základě poměru výšky a šířky oblasti je určena nejpravděpodobnější oblast.

V [12] je po hranové detekci použita série morfologických uzavření a otevření strukturním elementem o velikosti $1/2$ největší mezery mezi znaky, čímž jsou získány slité oblasti. Následně jsou dvěma morfologickými otevíráními eliminovány nevyhovující oblasti. Použité strukturní elementy musí být užší a nižší než registrační značka. Otevíráními jsou eliminovány oblasti, které jsou menší než strukturní element.

Další možností je vyhledávání oblasti o rozměrech registrační značky, která má největší součet jasů [4]. Tato oblast je prohlášena za registrační značku. Výhodou tohoto řešení je, že není nutné provádět žádnou morfologickou operaci. Proto byla pro účely této práce zvolena tato metoda.

3.3 Segmentace znaků

Správná segmentace znaků je klíčová pro následné správné rozpoznání znaku. Deterministické i stochastické metody rozpoznávání znaků jsou velmi náchylné na přesnost určení hranic znaků. Pro správné určení hranic znaků je nutná znalost parametrů rozpoznávané registrační značky. Použité metody se mohou také lišit v závislosti na provedení registrační značky (jednořádkové, víceřádkové). K určení pozice se nejčastěji používají následující přístupy:

- Použití pevných hranic z rozměru RZ,
- použití vertikálního a horizontálního součtu,
- použití souvislých oblastí,

nebo například shlukování, které je velmi výpočetně náročné a používá se pouze ojediněle.

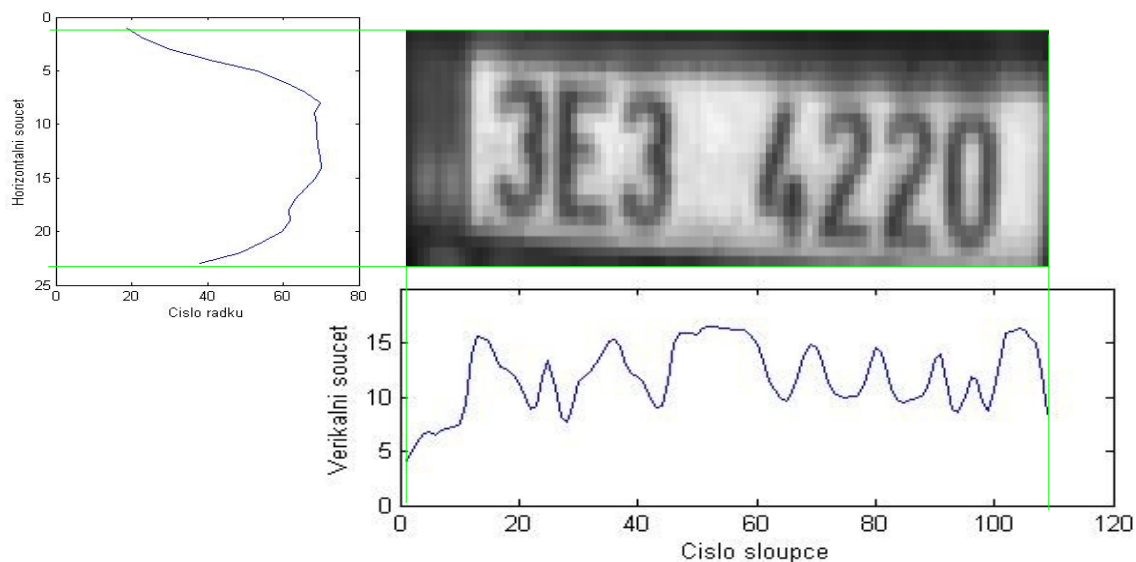
3.3.1 Použití pevných hranic z rozměrů RZ

Při použití této metody se vychází z toho, že znaky jsou na značce vždy na stejné pozici. To umožňuje použití pouze pro jeden typ registračních značek. Navíc je tato metoda

velmi náchylná na přesnost určení polohy registrační značky a její natočení. Pro tyto nedostatky tato metoda není příliš vhodná a nebyla v této práci použita.

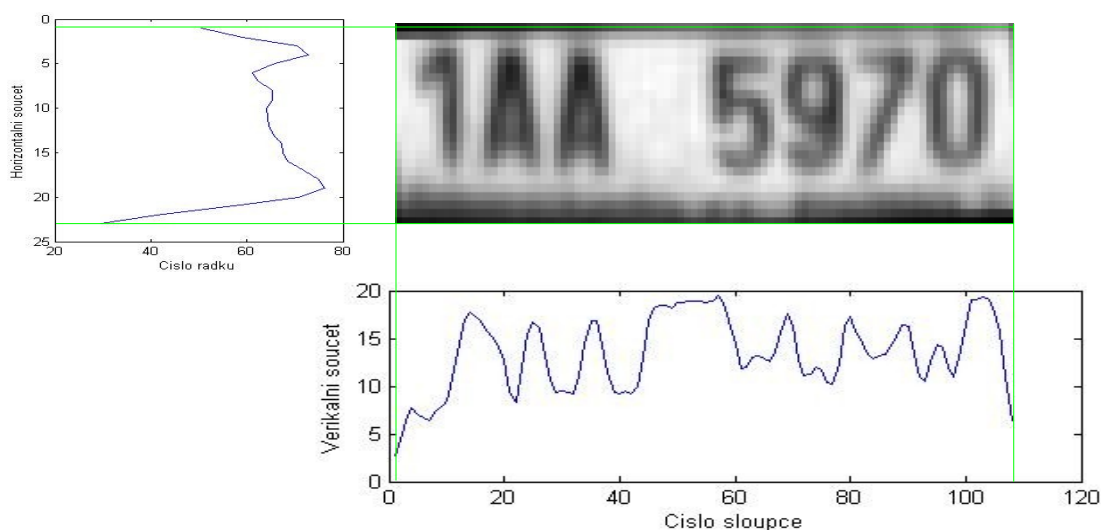
3.3.2 Použití vertikálního a horizontálního součtu

Při této metodě se využívá toho, že je mezi horní hranicí znaku a horní hranou registrační značky, stejně tak mezi dolní hranicí znaku a dolní hranou registrační značky výrazné maximum horizontálního součtu jasu. Horizontální součet je proveden sečtením hodnot jasu obrázku přes řádky. Při vertikálním součtu se objeví maxima průběhu mezi znaky, velikost maxim závisí na velikosti mezery mezi znaky. Vertikální součet je vypočten sečtením hodnot jasu obrázku přes sloupce. Tato metoda je však náchylná na natočení registrační značky. Natočená registrační značka s průběhy horizontálních a vertikálních součtů je na obrázku 3.



Obrázek 2- Příklad Natočené RZ

Natočení je možné kompenzovat například na základě Houghovy transformace [14], to ale zvyšuje náročnost systému na výpočetní výkon. Ukázka zkorigované registrační značky s průběhy horizontálních a vertikálních součtů je na obrázku 3. Z obrázků 2 a 3 je zřejmé, že i při poměrně malém natočení je určení hranic téměř nemožné.



Obrázek 3- Příklad zkorigované RZ

V [21] je nejprve vypočten vertikální součet a každé minimum v součtu je označeno jako možná hranice znaku. Pro každou takto nalezenou oblast je vypočten horizontální součet a podle jeho maxim vykreslena horní a spodní hranice. Na všechny takto nalezené linie je aplikována Houghova transformace, pomocí které se vyloučí hranice, které neleží na jedné přímce. Tím odpadá nutnost rotace obrázku, alespoň v této fázi. Vertikální hranice jsou určovány také z minim a navíc jsou porovnávány se známým rozložením znaku na registrační značce.

3.3.3 Použití spojených oblastí

V této metodě jsou v binarizovaném obraze vyhledávány spojitě oblasti, které svými parametry (výškou, šířkou, pozicí a počtem obrazových bodů) odpovídají předpokládaným parametrům znaku [8] [10]. Převod na binarizovaný obraz je realizován prahováním. Prahování je nejdůležitější částí této metody. Při nesprávném určení prahu hrozí buď ignorování slabých znaků nebo naopak slití znaků dohromady s okolím, což velmi ztěžuje jejich následné rozpoznání. Příklad výsledku segmentace znaků pomocí spojených oblastí je na obrázku 4.



Obrázek 4- Použití spojených oblastí

Hlavní výhodou tohoto řešení je odolnost proti natočení. Tím že není nutné obraz natáčet se výrazně snižuje výpočetní čas. Proto bylo toto řešení zvoleno pro použití v této práci.

3.4 Rozpoznání znaků

V rozpoznávání znaků jsou zpracovávány nalezené oblasti znaků. Pro rozpoznávání znaků je možné zvolit dva základní přístupy: deterministické a stochastické. Z deterministických metod je nejpoužívanější porovnávání se šablonami (template matching) a ze stochastických použití neuronových sítí. Porovnávání se šablonami je vhodné pouze pro omezenou množinu šablon, při větších množinách šablon narůstá doba potřebná pro rozpoznání znaku.

3.4.1 Porovnávání se šablonami

Při tomto postupu se obraz znaku porovnává postupně se šablonami všech znaků. První z možností porovnávání je použití korelačního koeficientu. Čím je korelační koeficient vyšší, tím menší je rozdíl mezi obrazem a šablonou. Při absolutní shodě je korelační koeficient 1. Tento postup je použit např. v [11]. Tento postup je však poměrně výpočetně náročný lze ho však nahradit metodou SAD⁴, která poskytuje téměř stejně přesné výsledky a je méně výpočetně náročná [4]. Na rozdíl od korelačního koeficientu je při absolutní shodě koeficient SAD 0. Pro svou nízkou výpočetní náročnost byl pro tvorbu práce zvolen koeficient SAD. Porovnávání s šablonami je náchylné na šum v obraze, který výrazně snižuje pravděpodobnost správného rozpoznání znaků. Tyto nevýhody byly v této práci kompenzovány použitím kvalitní videokamery. Deterministický způsob podával uspokojivé výsledky, proto byl pro tuto práci zvolen.

3.4.2 Použití neuronových sítí

Využití neuronových sítí k rozpoznávání znaků je díky lineární separabilitě obrazové interpretace znaků nejvíce používanou metodou. Nejčastěji je pro rozpoznávání znaků využíván McCullochův-Pittsův model neuronu a to hlavně pro svou snadnou interpretaci rovnic modelu neuronu. Tato metoda je použita například v [8] [9] [17]. Neuronová síť je třívrstvá. V první vrstvě je rozhodováno zda obrazový bod náleží znaku nebo pozadí. V druhé vrstvě je rozhodováno, zda obrazový bod přísluší k některé skupině bodů znaku nebo pozadí. V třetí vrstvě jsou spojovány skupiny náležící k pozadí nebo k znaku. Tento postup nebyl pro práci zvolen, protože návrh neuronové sítě je časově náročný. Následně je potřebné provést trénování, což je také časově náročné.

4 Sum of Absolute Differences

4 Registrační značky v ČR

V současné době jsou v České republice nejrozšířenější registrační značky vydávané od roku 2004. Na obrázku 5 je zobrazena registrační značka Typ 101, která je nejrozšířenější.



Obrázek 5- RZ typ101, zdroj: MD ČR

Tyto značky obsahují modrý pruh s logem evropské unie a označením státu původu. Tento pruh by mohl tvořit problémy při segmentaci registrační značky, protože snižuje rozdíl jasu v okolí okraje registrační značky. Značka má rozměry 520mm x 110mm a obsahuje 7 černých znaků na bílém pozadí znaků. Přičemž značka musí obsahovat minimálně jednu číslici a jedno písmeno.

V této sérii se mohou vyskytovat i typy značek s jiným barevným provedením, například typ 151, zobrazený na obrázku 6, určený pro traktory a samojízdné pracovní stroje,



Obrázek 6- RZ typ 151, zdroj: MD ČR

nebo typ 201, zobrazený na obrázku 7, určený pro vozidla členů diplomatických misí,



Obrázek 7- RZ typ 201, zdroj: MD ČR

nebo typ 301, zobrazený na obrázku 8, určený pro historická vozidla, který má zároveň minimální možný počet znaků.



Obrázek 8- RZ typ 301, zdroj: MD ČR

Dalším typem značek, který je na českých silnicích hojně zastoupen, je typ vydávaný v letech 2001 – 2004, který svými rozměry, variantami a uspořádáním znaků odpovídají registračním značkám vydávaným po roce 2004, ale neobsahuje modrý pruh s logem evropské unie. Vzor registrační značky tohoto typu je na obrázku 9



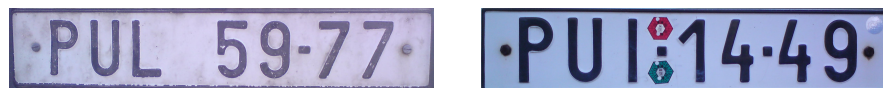
Obrázek 9- RZ vydávaná 2001 - 2004, zdroj: MD ČR

Na registračních značkách vydávaných po roce 2001 je použita znaková sada založená na DIN 1451 Mittelschrift. Pro snadnější rozpoznávání se na registračních značkách nepoužívají znaky „G“ a „Q“ které mohou být snadno zaměněny za znaky „C“ a „O“. Například v Nizozemí je od roku 2002 na registračních značkách používán speciální font, který je upravený pro automatické rozpoznávání. Princip spočívá v tom že linie znaků, které jsou si podobné, jsou na různých místech přerušeny. Znaková sada nizozemských registračních značek je na obrázku 10.



Obrázek 10- Nizozemská znaková sada

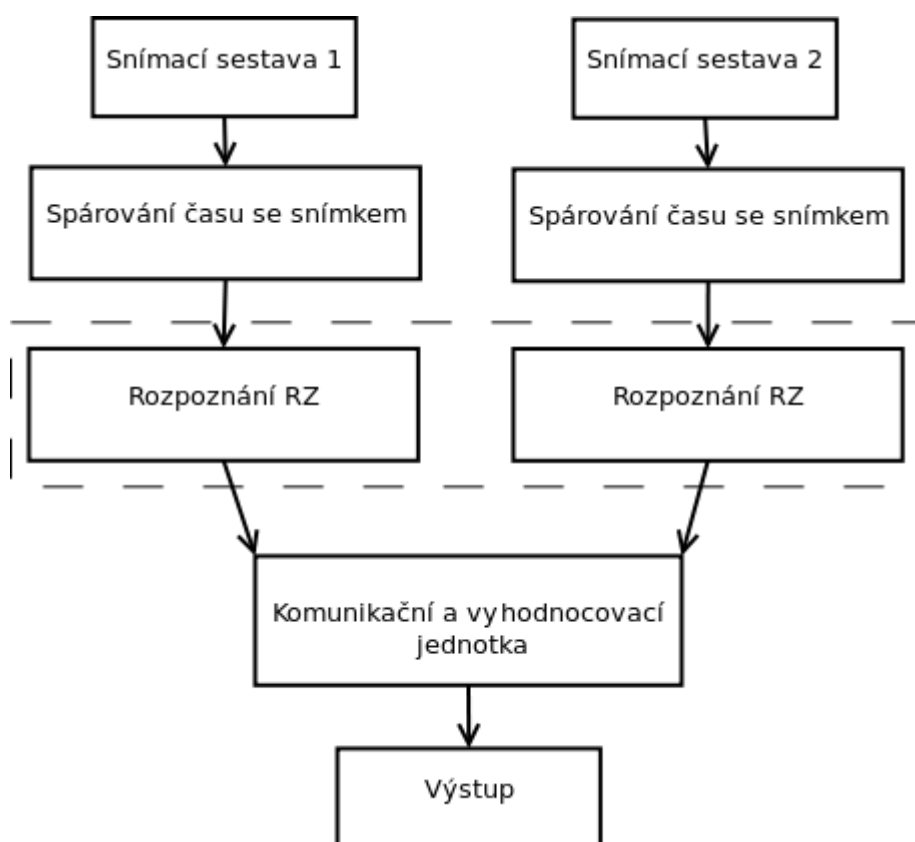
V menší míře jsou zastoupeny starší registrační značky vydávané před rokem 2001, které běžně obsahují na prvních třech pozicích písmena a zbývajících čtyřech číslice. Příklady starších registračních značek jsou na obrázku 11.



Obrázek 11- Starší RZ

5 Rozbor vlastností navrhovaného systému

Navržený algoritmus rozpoznávání registračních značek bude možné používat jako součást systému úsekového měření rychlosti. Jako nejvhodnější se jeví systém složený ze dvou samostatných jednotek, které se budou skládat z ethernetové kamery a vestavného zařízení. V každé z jednotek bude implementován navržený algoritmus pro rozpoznávání značek a jedna z jednotek bude navíc zastávat funkci vyhodnocovací jednotky a komunikační jednotky. Na obrázku 12 je schéma celého systému s vyznačenou oblastí, která je řešena v této práci.



Obrázek 12- Schéma celého systému

Systém bude navržen tak, že jej bude možné umístit na silniční bránu nebo nadjezd. Navržený systém bude pro detekci vozidla v obraze využívat hardwarovou detekci vozidla dálkoměrem, tato metoda poskytuje dostatečně přesnou detekci a nevyžaduje žádný stavební zásah do vozovky.

Navržený systém by měl být dostatečně rychlý, aby ho bylo možné používat i v hustém provozu na silnicích s více jízdními pruhy.

Systém by měl být schopen rozpoznávat přední registrační značky používané v České republice o standardním evropském rozměru 520x110mm.

V zadání práce nebyla specifikována požadovaná pravděpodobnost správného rozpoznání registrační značky. Běžně používané systémy mají udávanou pravděpodobnost správného rozpoznání registrační značky přes 0,95. Tyto systémy ale většinou používají stochastické metody rozpoznávání jednotlivých znaků, které dosahují vyšší úspěšnosti. Za úspěch by bylo považováno, kdyby pravděpodobnost správného rozpoznání jednotlivých znaků byla min. 0,80 a pravděpodobnost správného rozpoznání registrační značky vyšší než 0,60. Tyto hodnoty pravděpodobnosti jsou uvažovány za běžných světelných podmínek a za předpokladu nepoškozených a neznečištěných registračních značek. Při znečištění registrační značky rapidně klesá pravděpodobnost jejího úspěšného rozpoznání.

6 Návrh systému

Vzhledem k tomu, že systém bude provozován na vestavných zařízeních, které disponují omezeným výpočetním výkonem, bylo při návrhu systému nutné zvážit efektivitu a potřebný výkon jednotlivých metod rozpoznávání registračních značek optickými metodami. Pro vyhledávání oblasti registrační značky byl zvolen postup pomocí detekce vertikálních hran pro své poměrně přesné výsledky. Při tomto způsobu lokalizace bylo nutné provádět pouze minimum dodatečných úprav. Segmentace znaků byla zvolena metodou spojených oblastí pro svou odolnost vůči natočení registrační značky. Samotné rozpoznání znaků bylo řešeno deterministickým způsobem pro svou snadnou implementaci a návrh.

6.1 Určení požadavků na rychlost systému

Jako podklad pro stanovení nároků na rychlost systému byly použity intenzity dopravy v roce 2005 [1] na silnicích 1. třídy, která se pohybuje pod 30-ti tisíci vozidel za 24 hodin na silnicích s dvěma jízdními pruhy. Uvážíme-li, že dálnice mají v nejvytíženějších úsecích (v okolí Prahy) 6 jízdních pruhů a intenzitu dopravy pod 100 tisíc vozidel za 24 hodin, vychází i tam intenzita na jeden jízdní pruh přibližně stejná. Proto byla, s určitou rezervou, určena jako minimální intenzita vozidel na 20 tisíc na jeden pruh a 24 hodin. Z toho vyplývá, že za předpokladu možnosti odloženého výpočtu, musí být zpracování jednoho vozidla kratší než 4,3 vteřin.

6.2 Pořízení dat

Prvotním problémem při realizaci bylo pořízení kvalitních obrazových dat. Požadavky na pořízení snímků byly následující: vozidla budou zabírána v jednom jízdním pruhu zepředu a shora tak, aby bylo simulováno snímání ze silniční brány. Při požadavcích na rozlišení se vychází z minimálního počtu horizontálních obrazových bodů připadajících na jeden znak potřebných pro správnou funkci rozpoznávání znaků deterministickými metodami. Toto minimum bylo po předchozích experimentech určeno na 5 obrazových bodů. Z důvodu zvýšení robustnosti systému bylo minimum zvýšeno na dvojnásobek. Jelikož šířka znaku je přibližně 1/10 šířky registrační značky, minimální počet horizontálních obrazových bodů je určen na 100.

Snímky použité v této práci byly snímány na silnici I/37 poblíž obce Hrobice ze silničního nadjezdu z výšky cca. 5 metrů. Toto místo bylo zvoleno pro svou dobrou

dostupnost, dostatečnou intenzitu dopravy a podobnost pozice jako při snímání ze silničních bran, na kterých jsou většinou zařízení pro úsekové měření umístěny.

Dalším krokem je výběr vhodné videokamery. Při výběru videokamery je nutné zvážit požadavky na rozlišení a na rychlost snímání. Jelikož bude pro účely této práce využita detekce vozidla zpracováním obrazu je důležitý prvek i snímkovací frekvence. Nároky na snímkovací frekvenci závisí na předpokládané rychlosti vozidel. Pokud bude frekvence velmi nízká, může se stát, že vozidlo projede snímanou oblastí mezi jednotlivými snímky. Vzdálenost, kterou vozidlo urazí mezi dvěma snímky, je rovna maximálnímu rozdílu polohy vozidla na snímku, ve kterém bylo vozidlo detekováno. Ukázka ujetých vzdáleností mezi snímky při vybraných rychlostech a snímkovací frekvenci 10 snímků za vteřinu je v tabulce 1.

Tabulka 1- Ujetá vzdálenost mezi snímky

Rychlost [km/h]	50	90	130	200
Ujetá vzdálenost [m]	1,39	2,5	3,61	5,56

Pro snímání byla použita videokamera scA1300-32fc z řady Scout od firmy Basler zobrazené na obrázku 13.



*Obrázek 13- Basler Scout,
zdroj: Basler*

Tato kamera byla zvolena hlavně pro své vysoké rozlišení, kvalitu snímače a možnost přesné konfigurace snímacích parametrů. Parametry videokamery jsou uvedeny v tabulce 2. Jedinou nevýhodou této kamery byla absence ethernetového rozhraní.

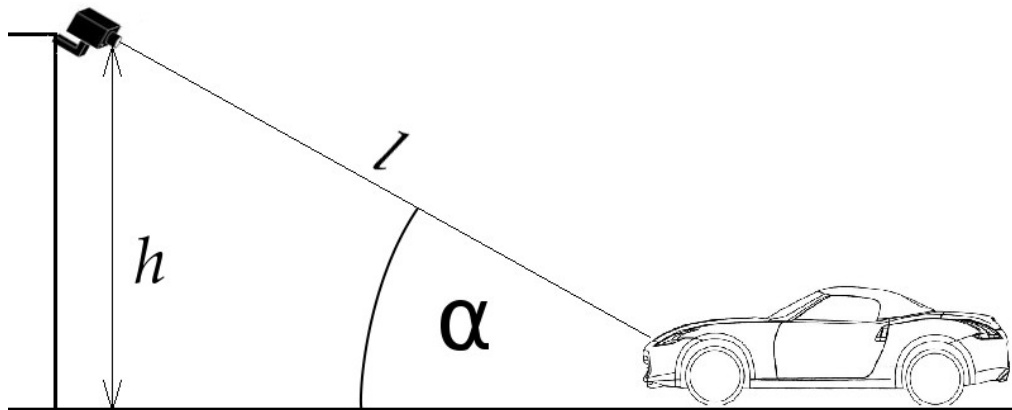
Tabulka 2- Parametry videokamery

Rozlišení (px)	Snímač	Technologie	Velikost snímače	Snímací frekvence	Rozhraní	Objektiv
1294x964	Sony ICX445	CCD	1/3"	32 snímků/s	IEEE 1934b	C - mount

Další klíčovou částí při pořizování dat je výběr vhodného objektivu. Pro výběr objektivu je nutné zvolit správnou ohniskovou vzdálenost. Ohnisková vzdálenost objektivu byla vypočtena na základě požadované elevace a minimálního počtu obrazových bodů na šířku registrační značky. Elevace byla stanovena na 30 stupňů. Při snížení elevace dochází k skrývání blízko jedoucích vozidel za sebe, naopak při zvětšování elevace dochází k větší deformaci znaků, což způsobuje problémy při rozpoznávání. Snímací vzdálenost byla vypočítána pomocí elevace a vzdáleností kamery nad vozovkou ze vzorce:

$$l = \frac{h}{\sin(\alpha)} \quad (1)$$

kde l je snímací vzdálenost, h je výška nad vozovkou a α je elevace. Pro tento případ dostáváme snímací vzdálenost cca 10 metrů. Nákres pozice snímací sestavy z boku je na obrázku 14.

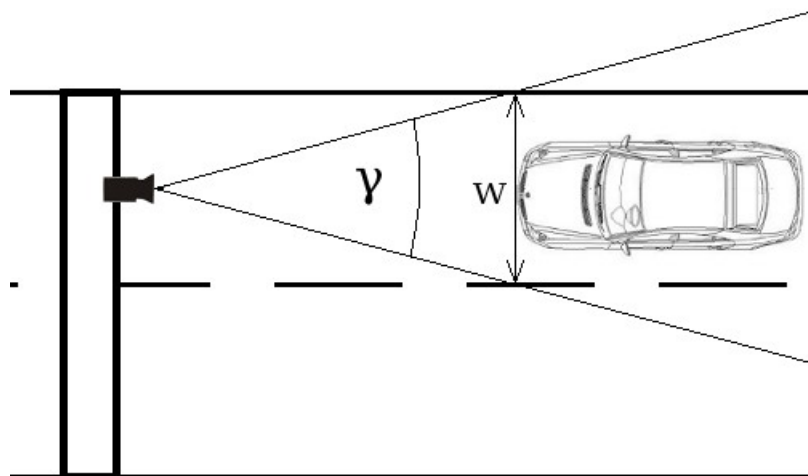


Obrázek 14-Nákres umístění snímací sestavy – boční pohled

Pro výpočet úhlu záběru byla nejdříve vypočtena maximální šířka záběru. Ze znalosti minimálního počtu obrazových bodů na registrační značku, šířky registrační značky a rozlišení kamery byla vypočtena pomocí následujícího vzorce:

$$w = \frac{pw \cdot H_{res}}{P_{min}} \quad (2)$$

kde w je šířka záběru, pw je šířka registrační značky, $Hres$ horizontální rozlišení kamery a $Pmin$ je minimální počet obrazových bodů na šířku registrační značky. Nákres snímací sestavy shora je na obrázku 15.



Obrázek 15- Nákres umístění snímací sestavy – pohled shora

Maximální šířka záběru pro náš případ vychází 6,72 metrů. Z toho vyplývá, že jeden obrazový bod bude široký cca. 5,2 mm. Maximální horizontální úhel záběru byl vypočten podle následujícího vzorce:

$$\gamma = 2 \cdot \arctan\left(\frac{w}{2 \cdot l}\right) \quad (3)$$

kde γ je maximální horizontální úhel záběru, w šířka záběru a l snímací vzdálenost. Odtud dostáváme maximální horizontální úhel záběru 37,1°. Proto byl zvolen objektiv M1214-MP od firmy Computar, zobrazený na obrázku 16, parametry objektivu jsou v tabulce 3.

Tabulka 3- Parametry objektivu

Ohnisková vzdálenost	Clona	Vertikální úhel záběru	Horizontální úhel záběru	Ostřicí vzdálenost
12 mm	F1,4 - 16C	22,7°	30°	0,3m - inf.



Obrázek 16- Objektiv MP1214-MP,
zdroj: Computar

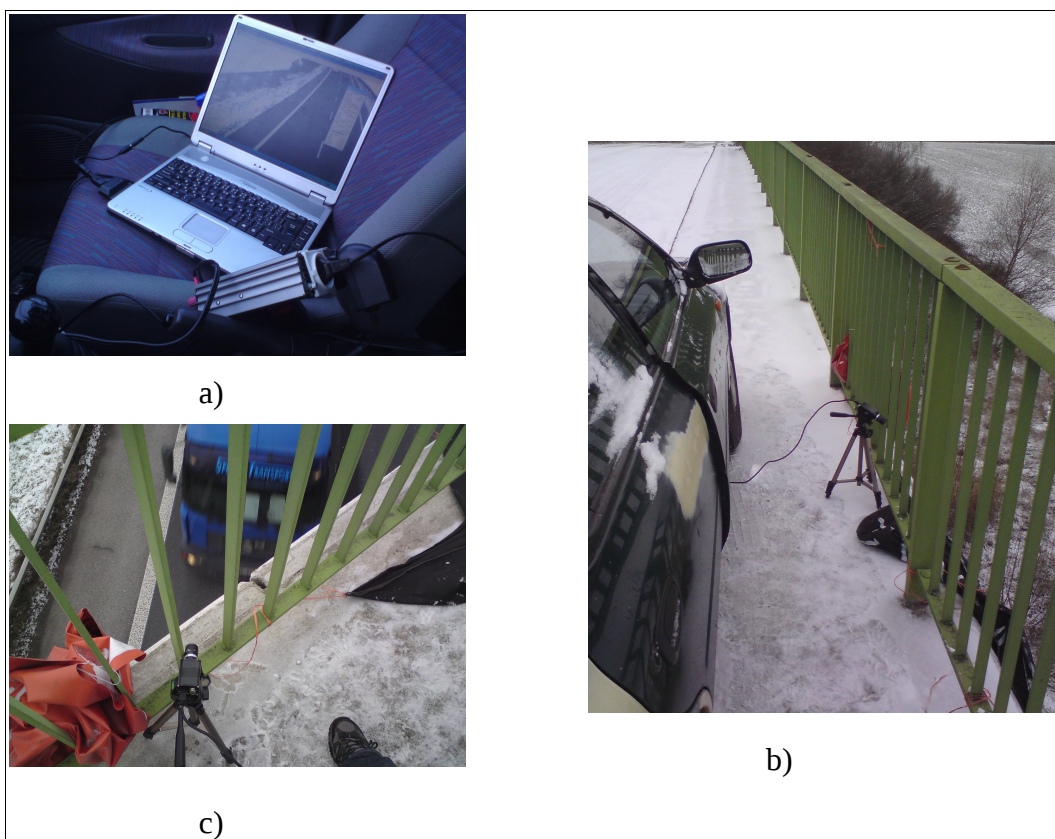
Při použití tohoto objektivu bude záběr přibližně 5,78 metrů široký a na registrační značku bude v horizontálním směru připadat 116 obrazových bodů. Takto vypočtené parametry objektivu se ukázaly jako vyhovující pro účely pořízení záběrů pro tuto práci.

Jelikož při tvorbě této práce byla k dispozici pouze kamera s rozhraním Firewire, ale pro budoucí použití je počítáno s kamerou s ethernetovým rozhraním je ethernetová kamera simulována pomocí počítače. Pro tyto účely byl vytvořen program pro PC, který realizuje předávání snímků z počítače do vývojového kitu. Zdrojový kód a zkompilovaný program je na příloženém CD. Pro ukládání záběrů do počítače byl použit program *VirtualVCR*. Program *VirtualVCR* umožňuje ukládat video ve formátu AVI⁵ ze zařízení používajících ovladače WDM⁶.

Při snímání byla kamera umístěna na stativu a objektiv byl zaostřen přibližně na střed záběru. Fotografie snímacího stanoviště jsou umístěny na obrázku 17.

5 *Audio Video Interleave*

6 *Windows Driver Model*



Obrázek 17- Fotografie snímacího stanoviště

Snímky jsou snímány černobíle s maximálním rozlišením. Pro získání co nejostřejších snímků je nutné nastavit co nejmenší dobu expozice, jelikož v době snímání byly dobré světelné podmínky a použitý objektiv měl dostatečnou světelnost, byla doba expozice nastavena na 22 μ s. Při zhoršených světelných podmínkách by bylo nutné systém doplnit o automatickou korekci doby expozice a zisku snímače podle parametrů snímané scény. Pro správnou funkci snímacího zařízení i za zhoršených světelných poměrů je nutné použít infračervené přisvětlení/blesk, použití viditelného přisvětlení/blesku není možné z důvodu možného oslnění řidičů.

7 Popis navržených algoritmů obrazové detekce

Stěžejní částí práce je návrh algoritmů obrazové detekce. Důsledný návrh jednotlivých kroků zvyšuje úspěšnost rozpoznání registrační značky. Při návrhu algoritmů byl zohledněn předpokládaný výkon a možnost přenosu do nižšího programovacího jazyka. Navržené algoritmy jsou rozděleny do několika základních částí:

- Určení polohy vozidla,
- lokalizace registrační značky,
- segmentace znaků,
- rozpoznání znaků.

7.1 Určení polohy vozidla v obraze

Prvním problémem byla lokalizace vozidla v obraze. Proto byla zvolena metoda, kdy jsou vypočteny parametry jasu vozovky a poloha vozidla je určena na základě odchylky od těchto parametrů. Nejprve jsou v obraze spočteny součty hodnot jasu přes jednotlivé řádky. Výpočet je proveden podle vzorce:

$$SumH(y) = \sum_{x=1}^X i(x, y) \quad (4)$$

kde $SumH(y)$ je vektor se součty jednotlivých řádků, X je počet sloupců v obraze a $i(x, y)$ je obraz.

Jelikož bylo zjištěno, že snímky předané z předchozí fáze mají na spodní hraně snímku minimálně 30 řádků vozovky bez vozidla je z této oblasti možné vypočítat statistické parametry jasu vozovky. Pro statistický popis jasu je použito střední hodnoty a směrodatné odchylky tyto parametry byly počítány podle vzorců 5 a 6.

$$\bar{x} = \frac{1}{n} \cdot \sum_{i=1}^n x_i \quad (5)$$

$$s = \sqrt{\frac{1}{n-1} \cdot \sum_{i=1}^n (x_i - \bar{x})^2} \quad (6)$$

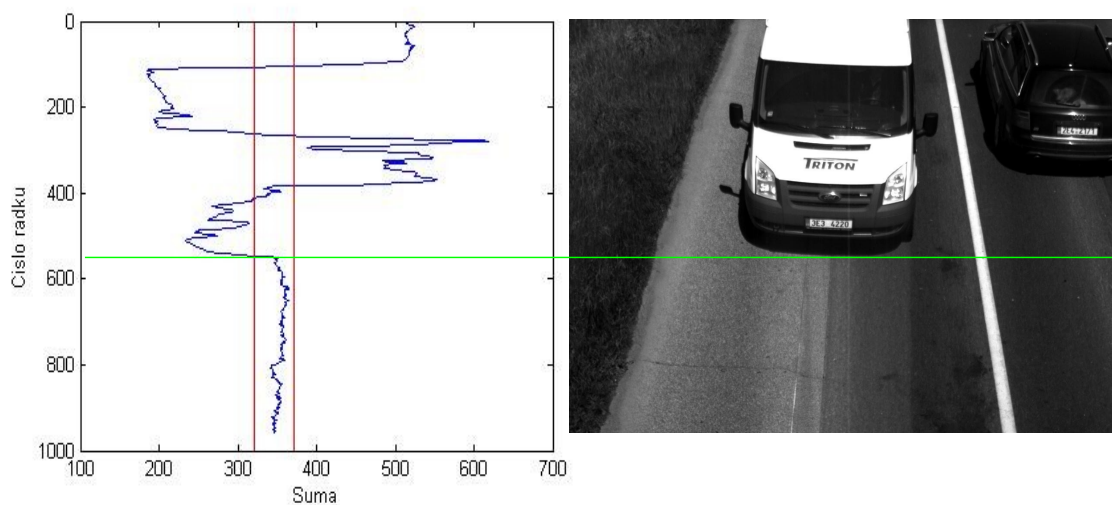
Z předpokladu, že při dostatečně velkém výběru obrazových bodů mají hodnoty jasů vozovky Gaussovo rozdělení (Centrální limitní věta), vyplývá, že v intervalu:

$$\langle \bar{x} - 3 \cdot s; \bar{x} + 3 \cdot s \rangle \quad (7)$$

leží více než 99,7% obrazových bodů vozovky. Vzhledem k tomu, že 30 řádků je malá oblast, výběr neodpovídá přesně Gaussovu rozdělení pravděpodobnosti. Proto byl zvolen interval:

$$\langle \bar{x} - 5 \cdot s; \bar{x} + 5 \cdot s \rangle \quad (8)$$

který lépe reprezentuje oblast vozovky. Následně je od konce zjišťováno, kde se součet řádku dostane pod tento interval a tento řádek je označen za spodní hranici vozidla. Vychází se z toho, že je vozovka pod vozidlem tmavší (vozidlo vrhá stín). Pro úplnost by bylo možné detekovat i překročení horní hranice. Ukázka součtu jednotlivých řádků s vyznačenými hranicemi intervalu a detekovanou hranou vozidla je na obrázku 18.



Obrázek 18- Součet řádků

Poté je uložen výřez 150 obrazových bodů nad spodní hranicí vozidla a šířkou jen pro jízdní pruh, ve kterém je třeba rozpoznávat registrační značky, naznačení tohoto výběru je na obrázku 19. Z důvodu snížení výpočetní náročnosti je dále zpracováván pouze tento výřez.



Obrázek 19- Oblast výřezu

7.2 Lokalizace registrační značky

Jelikož pozice vozidla v obraze a tím i velikost registrační značky se může významně lišit, je nutné odhadnout rozměry registrační značky v obrazových bodech. Díky axonometrickému zobrazení lze předpokládat, že vozidlo, které je více vzdáleno od snímací soustavy je reprezentováno v horizontálním směru menším počtem obrazových bodů. Tato závislost je modelována následujícím vztahem:

$$PlateW = ((Pos - 910) \cdot 0,08) + 137 \quad (9)$$

kde $PlateH$ se šířka značky v obrazových bodech a Pos je číslo řádku, který byl označen jako hranice vozidla. Výška registrační značky je poté dopočítána jako poměr šířky, podle vztahu:

$$PlateH = PlateW \cdot 0,2 \quad (10)$$

kde $PlateH$ je výška značky v obrazových bodech.

Hlavním předpokladem je že oblast RZ značky obsahuje velké množství vertikálních hran po aplikaci vhodného hranového detektoru. Tento přístup je jeden z nejčastěji aplikovaných a přináší velmi přesné výsledky. Při určení polohy registrační značky je využito toho, že registrační značka obsahuje velké množství svislých hran. Pro zvýraznění hran je využito konvoluce:

$$f(x, y) * g(w, h) = \sum_{i=1}^w \sum_{j=1}^h f(x+i, y+j) \cdot g(i, j) \quad (11)$$

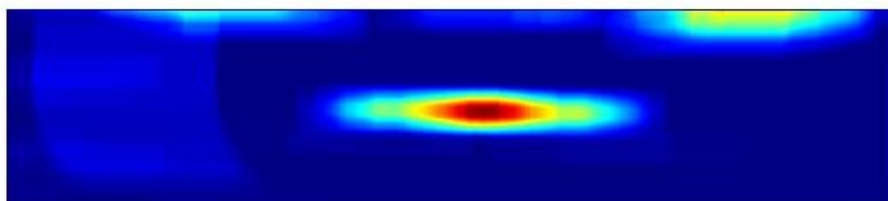
kde $f(x, y)$ je obraz a $g(w, h)$ je konvoluční jádro. Jako konvoluční jádro je použit vertikální Sobelův operátor:

$$s_x = \begin{vmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{vmatrix} \quad (12)$$

Ve výsledném obraze je hledáno maximum součtu jasu oblasti odpovídající rozměrům registrační značky. Součet je realizován následujícím vztahem:

$$\text{Sum}(x, y) = \sum_{i=1}^{\text{PlateW}} \sum_{j=1}^{\text{PlateH}} E(x+i, y+j) \quad (13)$$

kde $\text{Sum}(x, y)$ je matice součtů, PlateW a PlateH je šířka a výška registrační značky a $E(x, y)$ je hranový obraz. Součty jsou uloženy do matice pro každou pozici, aby bylo možné v případě neúspěšného určení pozice registrační značky opakovat postup s další pozicí. Příklad matice součtů je zobrazena na obrázku 20, pro zvýraznění diferencí je matice zobrazena v pseudobarvách. Na obrázku je patrné výrazné maximum, které odpovídá pozici registrační značky



Obrázek 20- Matice součtů oblastí

Následně je oblast o rozměrech registrační značky vyříznuta a další zpracování probíhá na tomto výřezu. Na obrázku 21 je zobrazen výsledný výřez.

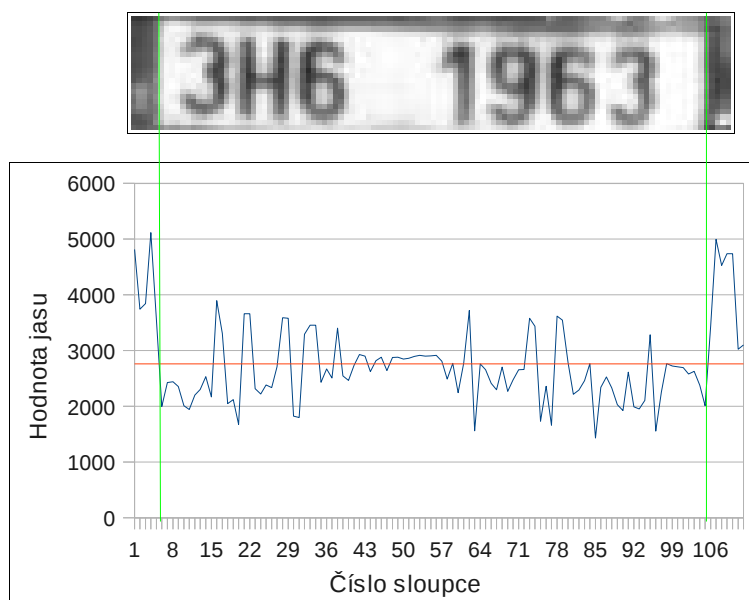


Obrázek 21- Výřez RZ

Dále je pozice zpřesněna dodatečným vertikálním ořezem. Ořez je realizován ze součtu sloupců, realizovaném podle (14), ve kterém jsou zleva a zprava vyhledávány okrajové body, které klesnou pod střední hodnotu všech součtů.

$$SumV(x) = \sum_{y=1}^Y i(x, y) \quad (14)$$

kde $SumV(x)$ je vektor se součty jednotlivých sloupců, Y je počet řádků v obraze a $i(x,y)$ je obraz. Toto místo značí přechod z tmavšího okolí na bílou část registrační značky. Na obrázku 22 je registrační značka před oříznutím a součet sloupců s vyznačenými hranicemi a střední hodnotou, na obrázku 23 jsou ukázky výsledných výřezů.



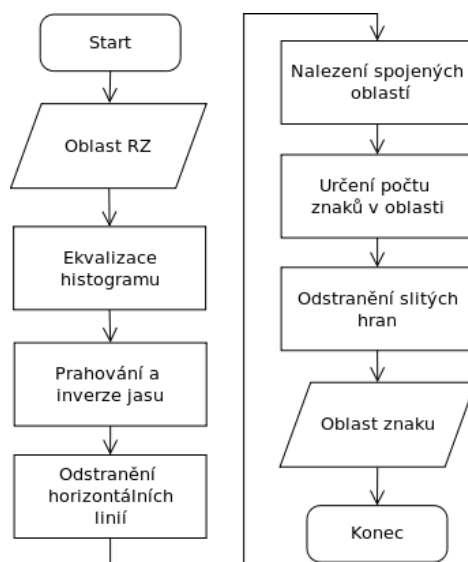
Obrázek 22- Součet sloupců (Modrá-součet sloupců, oranžová-střední hodnota, zelená- nalezené hranice RZ)



Obrázek 23- RZ po vertikálním oříznutí

7.3 Segmentace jednotlivých znaků

Další fází je segmentace jednotlivých znaků, ve které se ve výřezu z předchozí fáze separují černé znaky od pozadí. Postup segmentace znaků je znázorněn na obrázku 24.



Obrázek 24- Postup segmentace znaku

Segmentace je prováděna na binárním obraze. Před převodem na binární obraz je zvýrazněn kontrast pomocí ekvalizace histogramu, která zvýrazní kontrast hlavně u příliš světlých nebo příliš tmavých snímků. Příklady registračních značek před a po ekvalizaci histogramu jsou na obrázku 25.



Obrázek 25- RZ před ekvalizací (vlevo), po ekvalizaci (vpravo)

Převod na binární obraz je proveden pomocí prahování, hodnoty jasu obrazových bodů, které jsou menší než práh jsou nastaveny na maximální jas a hodnoty, které jsou větší jsou nastaveny na minimální jas. Tím je zároveň provedena inverze jasu. Práh je hledán pomocí následující iterační metody:

1. Je zvolen práh ve středu rozsahu hodnot.
2. Obraz je segmentován prahem na dvě oblasti.
3. Jsou vypočteny střední hodnoty jasu obrazových bodů v každé oblasti.
4. Nový práh je zvolen podle vzorce:

$$TH = \frac{\bar{x}_1 + \bar{x}_2}{2} \quad (15)$$

kde TH je hodnota prahu, \bar{x}_1 a \bar{x}_2 jsou střední hodnoty jasu v oblastech. Jelikož je algoritmus iterační, body 2 - 4 jsou opakovány, dokud práh použitý v bodě 2 neodpovídá prahu vypočtenému v bodě 4. Obraz po aplikaci prahu je na obrázku 26.



Obrázek 26- Prahovaný obraz

Pro minimalizování slévání znaku v horní a dolní části registrační značky jsou v obraze vyhledávány souvislé bílé řádky delší než 1/5 šířky registrační značky a ty jsou vyplněny černě. Výsledný obraz zobrazený na obrázku 27 obsahuje pouze znaky a malé oblasti, které budou vyloučeny dále.



Obrázek 27- Výřez bez horizontálních linií

V dalším kroku jsou z obrazu vybrány souvislé oblasti, které splňují následující podmínku:

$$S_A > \frac{1}{4} \cdot PlateW \cap W_A > 2 \cap H_A > 2 \quad (16)$$

Kde S_A je plocha souvislé oblasti, $PlateW$ je šířka registrační značky, W_A je šířka oblasti a H_A je výška oblasti. Všechny rozměry jsou uvažovány v obrazových bodech. Tím získáme výřez zobrazený na obrázku 28, který obsahuje pouze oblasti znaků.



Obrázek 28- Výřez připravený pro rozpoznání

Následně je nutné určit polohu jednotlivých znaků. Jelikož v některých případech bývá několik znaků slitých do sebe, jako například první tři znaky na obrázku 28, je nutné rozhodnout zda v oblasti není více znaků. Z šířky registrační značky je vypočtena předpokládaná šířka jednoho znaku, ze které je odhadnut, počet znaků v souvislé oblasti. Výpočet přibližné šířky znaku je proveden podle vzorce:

$$CharW = \frac{PlateW}{14,5} \quad (17)$$

kde $CharW$ je šířka znaku a $PlateW$ je šířka registrační značky. Pokud je v oblasti předpokládáno více znaků, je oblast rovnoměrně vertikálně rozdělena. Před samotným rozpoznáním znaku je ještě provedena poslední úprava výřezu.

Experimenty bylo zjištěno, že znaky často bývají slité s horní nebo spodní hranou registrační značky a tato hrana většinou bývá spojena se znakem pouze jedním obrazovým bodem, příklad takového znaku je zobrazen na obrázku 29 a. Proto jsou kontrolovány tři nejvyšší a tři nejnižší řádky, zda je na nich méně než dva obrazové body. Pokud je takový řádek nalezen, je horní, resp. dolní hranice posunuta od tohoto řádku o jeden obrazový bod směrem ke středu. Při tomto ořezu zůstane nalevo příp. napravo několik prázdných sloupců. Po oříznutí těchto prázdných sloupců je výřez předán dál k rozpoznání znaku. Ukázka znaku po odstranění slité hrany je na obrázku 29 b.



Obrázek 29- Znak a) před úpravou , b) po úpravě

7.4 Rozpoznání znaků

Pro rozpoznání znaků byl zvolen deterministický přístup z důvodu rychlosti a snadné implementace. Základem je porovnávání obrazů znaků se šablonami vytvořenými ze znakové sady použité na českých registračních značkách. Šablona znaků je zobrazena na obrázku 30.



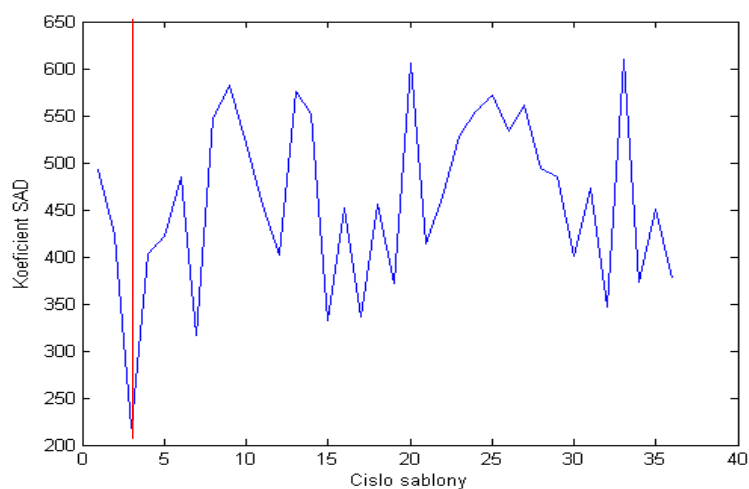
Obrázek 30- Šablony znaků

Znaky „O“ a „0“ se považují za totožné a rozdíl mezi nimi se v rozpoznávání neřeší. V šabloně nejsou obsaženy znaky „G“ a „Q“, které se na českých registračních značkách nevyskytují, což usnadňuje rozpoznávání. Při použití systému pro jiné znakové sady by bylo nutné šablonu upravit. Před porovnáním je ještě upravena velikost výřezu tak, aby byla stejná jako velikost šablony. Při změně velikosti se využívá bilineární interpolace.

Porovnání je provedeno metodou SAD, která vypočítává absolutní hodnoty rozdílů v každém bodě. Koeficient SAD je vypočítán následovně:

$$SAD(i, j) = \sum_{x=0}^w \sum_{y=0}^h |I_{x+i, y+j} - T_{x, y}| \quad (18)$$

kde $SAD(i, j)$ je matice koeficientů, w je šířka šablony, h je výška šablony, $I(i, j)$ je obraz, ve kterém je šablona $T(x, y)$ vyhledávána. Za předpokladu, že obraz $I(i, j)$ má stejné rozměry jako šablona $T(x, y)$ má matice $SAD(i, j)$ pouze jeden prvek. Tam, kde je hodnota SAD nejnižší, je rozdíl mezi šablonou a výřezem nejmenší. Tento znak je vrácen jako nejpravděpodobnější. Ukázka průběhu koeficientu SAD pro jednotlivé šablony s vyznačením nalezeného minima je na obrázku 31.



Obrázek 31- Průběh koeficientu SAD

Na obrázku 31 je zřetelné výrazné minimum u šablony číslo 3, se kterou má porovnávaný znak nejlepší shodu. V případě, že by se vyskytla nutnost určovat automaticky pravděpodobnost správného určení registrační značky, bylo by nutné ji vypočítat z hodnot koeficientů SAD pro jednotlivé znaky tvořící registrační značku.

8 Návrh konkrétních řešení

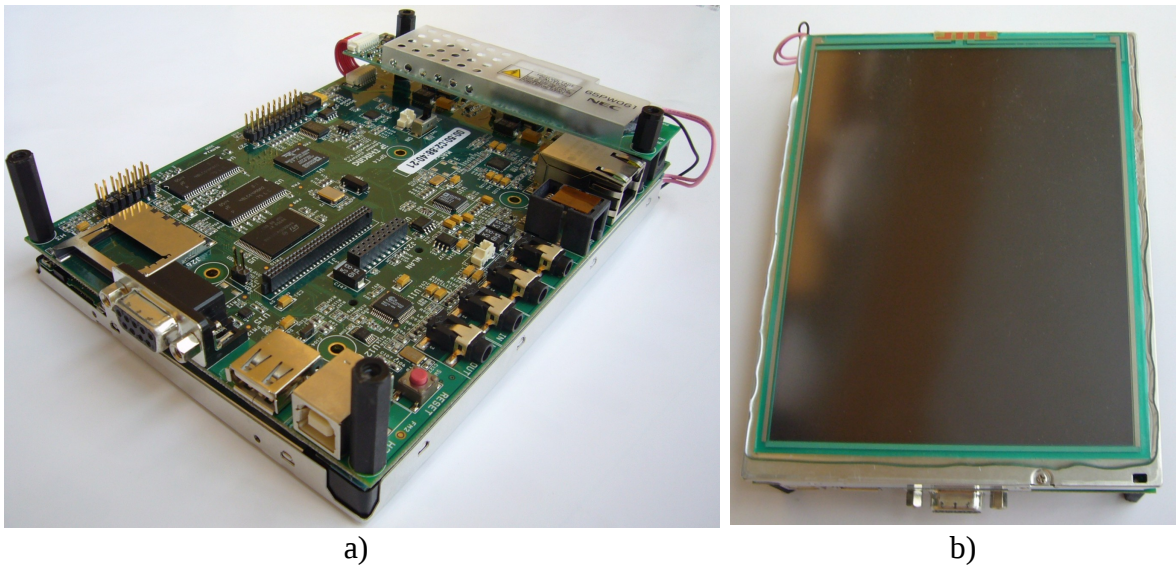
Vzhledem k vysokým požadavkům na rozlišení záběrů, ze kterého vyplývá velký objem zpracovávaných dat, bylo nutné zvolit hardwarovou platformu s dostatečným výpočetním výkonem. Proto bylo rozhodnuto, že pro realizaci bude zvolena platforma s výkonným signálovým procesorem (DSP⁷). Byl zvolen procesor ADSP-BF537 řady *Blackfin* do firmy Analog devices. ADSP-BF537 je jeden z nejvýkonnějších procesorů z této řady. Jedná se o 32-bitový procesor s frekvencí 600MHz a architekturou *RISC*⁸. Tento procesor byl zvolen především pro svůj vysoký výpočetní výkon.

Jelikož celý systém je poměrně komplexní bylo rozhodnuto o využití některého z vestavných operačních systémů, to sice částečně ubírá na efektivitě procesoru, ale výrazně zjednodušuje vývoj aplikací a zejména přístup k perifériím. Jako operační systém byl zvolen *uLinux*, hlavně pro svou rozšířenost na procesorech řady *Blackfin* a pro podobnost návrhu aplikací s operačním systémem *Linux* na platformě PC. Použití operačního systému umožňuje vyvíjet aplikace v podstatě nezávislé na platformě a bez detailní znalosti její hardwarové architektury. Nicméně *uLinux* má jistá omezení, která mírně komplikují vývoj aplikací. Ale výhody použití tohoto operačního systému pro navrhovaný systém převažují.

Při výběru vývojového kitu byly zohledněny předchozí požadavky a nakonec byl zvolen kit IPT-Shark537 od dánské firmy IP Thinking. Tento kit obsahuje procesor ADSP-BF537 a zároveň je dodáván s funkčním systémem *uLinux*. Kit také disponuje dostatečným množstvím paměti a velkým množstvím různých rozhraní. Vývojový kit je zobrazen na obrázku 32.

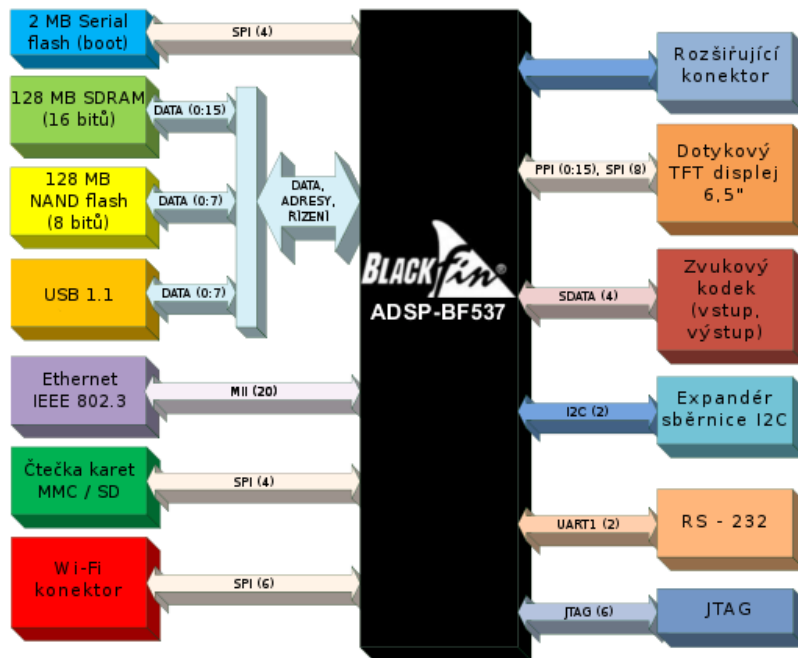
7 Digitální signálový procesor

8 Reduced Instruction Set Computer



Obrázek 32- Vývojový kit IPT-Shark537, a) pohled zdola , b) pohled shora

Kit je dodáván již předinstalovaný s ukázkovými aplikacemi, na DVD dodávaném společně s kitem jsou zdrojové kódy zavaděče a operačního systému. Vzhledem k tomu je možné si zavaděč i operační systém překonfigurovat, popř. přidat další ovladače. Hardwarové uspořádání kitu s jednotlivými rozhraními je na obrázku 33.



Obrázek 33- HW uspořádání kitu, zdroj: IP Thinking

Pro využití v této práci bylo nejdůležitější rozhraní *Ethernet*, které je využíváno pro připojení ethernetové kamery a zároveň může sloužit pro komunikaci s nadřazenými systémy. Vývojový kit může být dovybaven i zařízením pro bezdrátovou komunikaci Wi-Fi, tato možnost nebyla při tvorbě této práce využita, protože rychlost přenosu prostřednictvím sítě Wi-Fi by byla pro tento účel nedostatečná. Klíčové parametry kitu jsou na tabulce 4, kompletní specifikace je v příloze A

Tabulka 4- Parametry vývojového kitu

Procesor	Frekvence CPU	Frekvence sběrnice	Paměť SDRAM	Paměť Flash
ADSP-BF537	525 MHz	131 MHz	128 MB	128 MB

Jako zavaděč operačního systému je použit U-Boot, který je uložen v sériové paměti flash. Operačním systémem je uClinux 2008r1, který využívá jádro Linux 2.6.33.18. Je možné použít i jinou verzi operačního systému i jádra, ale je nutné aplikovat záplatu pro tento typ kitu, která nemusí být pro všechny verze k dispozici. Operační systém je uložen v paměti NAND flash. Při zpracování této práce bylo kromě Ethernetu použito i rozhraní RS-232, na které je přeměrována konzole operačního systému, plánováno bylo i využití displeje, pro zobrazení výsledků, ale z časových důvodů se nepodařilo displej zprovoznit.

Použitím operačního systému uClinux je možné pro metody zpracování obrazu využít knihovny funkcí pro zpracování obrazu OpenCV. OpenCV je opensource knihovna funkcí, která obsahuje více než 500 optimalizovaných algoritmů pro zpracování obrazu. Funkce jsou k dispozici v jazycích C, C++ a Python. Pro vlastní návrh byl zvolen jazyk C. Použitím knihoven funkcí OpenCV se výrazně zrychluje vývoj aplikací pro zpracování obrazu. Většina funkcí je také optimalizovaných, takže jsou velice rychlé. Jediným problémem při použití knihoven funkcí OpenCV na platformě Blackfin je rozdílná alokace paměti oproti platformě PC, což způsobuje, že některé funkce nefungují správně.

8.1 Specifika návrhu

Jelikož cílem práce nebyl jen návrh a implementace algoritmů rozpoznávání registračních značek, ale také vytvoření referenčního popisu problematiky zpracování obrazu na vestavných zařízeních s operačním systémem uClinux, jsou zde popsána specifika návrhu aplikací pro operační systém *uClinux*.

Aplikace je možné psát stejně jako pro operační systém Linux na platformě PC v jazyce C/C++, jen namísto standardních kompilátorů `gcc/g++` je nutné použít následující kompilátory:

```
bfin-linux-uclibc-gcc
```

```
bfin-linux-uclibc-g++
```

které jsou dostupné na DVD přiloženém k vývojovému kitu.

Hlavní omezení plynou z toho, že systém *uClinux* je určen pro procesory které nepodporují MMU⁹, z toho plyne pouze omezená ochrana paměti. Operační systém *uClinux* poskytuje rozdělení paměti na uživatelský prostor a na prostor jádra, čímž ochraňuje registry používané jádrem operačního systému. Další ochranou paměti je vyhrazený prostor o velikosti 4 kb na začátku paměti, kam ukazují neinicializované ukazatele (NULL pointer). Při pokusu o přístup do této oblasti se aplikace ukončí. Ovšem při posunu neinicializovaného ukazatele o více než 4 kb dojde k posunu mimo vyhrazenou oblast a dochází většinou k pádu celého operačního systému. Při vývoji aplikace je nutné důsledně kontrolovat použití ukazatelů a neinicializovaných proměnných. Přístup ukazatelem mimo alokovaný prostor znamená většinou pád nejen aplikace, ale i celého operačního systému. Stejně tak pokus o čtení neinicializované proměnné znamená pád celého systému.

Operační systém *uClinux* podporuje pouze použití statických knihoven. Proto je nutné ke každé aplikaci přikompilovat knihovny které využívá. Tento přístup značně zvětšuje velikost aplikací využívajících rozsáhlé knihovny jako např. *openCV*. Odpadá zde ale problém s kompatibilitou různých verzí knihoven. Jelikož použitý vývojový kit disponuje dostatečně velikou pamětí, není velikost aplikací problémem.

8.2 Postup nastavení PC pro cross- kompilaci

Zde uvedený postup je pro nastavení počítače, na kterém budou kompilovány aplikace pro cílovou platformu *Blackfin*. Ideální je použití operačního systému *Linux*, je však možné kompilovat aplikace i pod operačním systémem *Windows*, ale nastavení je komplikovanější. Proto byl zvolen operační systém *Ubuntu 10.04 LTS - Lucid Lynx*, následující postup je pro tuto distribuci, ale postup na ostatních distribucích *Linuxu* bude obdobný. V systému by měly být nainstalovány následující balíčky:

- `binutils-dev`,
- `gcc`,
- `libc6`,

9 Memory Management Unit

- libncurses-dev,
- libzlib-dev,
- texinfo.

Pro instalaci balíčků je možné použít buď příkaz `apt-get` nebo správce balíčků *Synaptic*. Pro prvotní přístup k zařízení je nutné mít nainstalovaný sériový terminál např. *minicom*. Parametry sériové komunikace jsou uvedeny v tabulce 5. Pro další přístupy je vhodnější použít přístup pomocí protokolu *telnet*.

Tabulka 5 - Parametry sériové komunikace

Parametr	Hodnota
Rychlost	57600
Parita	Žádná
Datové bity	8
Stop bity	1
HW řízení toku	Žádné
SW řízení toku	Žádné

Dále je třeba nainstalovat soubory vývojových nástrojů (Toolchains), které jsou dodávány na DVD spolu s vývojovým kitem. Soubory vývojových nástrojů obsahují gcc a g++ kompilátory a linkery, které jsou upravené pro *uClinix* na platformě *Blackfin*. Soubor vývojových nástrojů je dodáván ve formě balíčku *rpm*, v distribuci *Ubuntu* je nutné stáhnout aplikaci *Alien*, která převede balíček *rpm* na balíček *deb*. Po instalaci soubory vývojových nástrojů je nutné přidat cestu ke kompilátorům a linkerům pomocí příkazů:

```
export PATH=$PATH:/opt/uClinix/bfin-uclinux/bin
export PATH=$PATH:/opt/uClinix/bfin-linux-uclibc/bin
```

8.2.1 Nastavení a kompilace OpenCV

Při kompilaci *OpenCV* pro *uClinix* je nutné v konfiguraci nastavit pouze statické knihovny, definovat cílovou platformu a nastavit kompilátory a linkery C/C++ pro *uClinix* na platformě *Blackfin*. Dále je vhodné vypnout podporu všech nepoužívaných doplňků, v opačném případě by bylo nutné všechny knihovny pro tyto doplňky zkompilovat pro *Blackfin uClinix*. Konfiguraci provedeme příkazem:

```
./configure --host=bfin-uclinux --without-gtk --without-carbon --without-quicktime --without-1394libs --without-ffmpeg --without-python --without-swig --without-v4l --enable-static --disable-shared --disable-apps CC=bfin-linux-uclibc-gcc CXX=bfin-linux-uclibc-g++ CPPFLAGS=-
```

```
I/usr/include
```

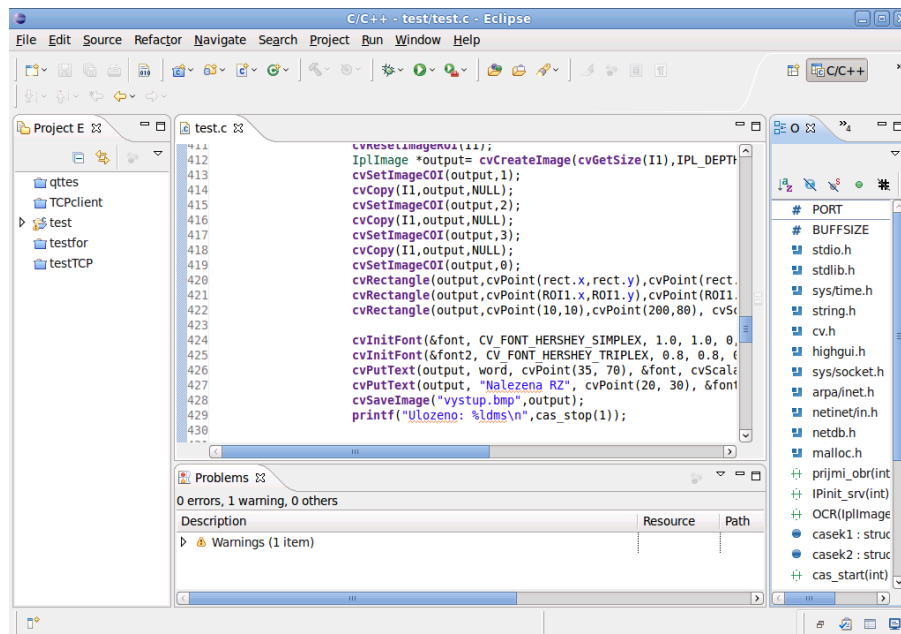
kompilaci provedeme standardně příkazem `make`. Příkaz `make install` zkopíruje knihovny a hlavičky funkcí do složek:

```
/usr/local/lib/
```

```
/usr/local/include/opencv/.
```

8.2.2 Nastavení IDE Eclipse

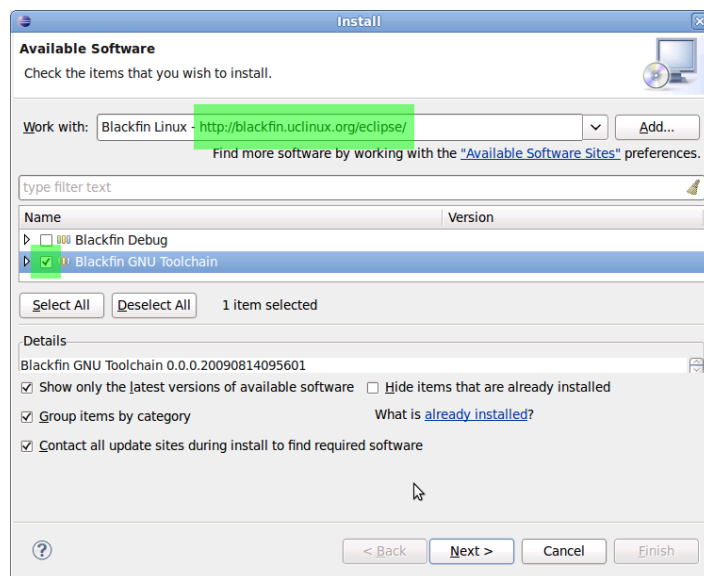
Pro snadnější vývoj aplikací je vhodné použít vývojové prostředí (IDE¹⁰). Před použitím vývojového prostředí je nutné ho správně nastavit. Při řešení této práce bylo použito vývojové prostředí *Eclipse*. Proto budou následující postupy určené pro toto vývojové prostředí. Ukázka vývojového prostředí *Eclipse* je na obrázku 34.



Obrázek 34- Vývojové prostředí Eclipse

Pro tuto aplikaci je třeba *Eclipse* s podporou C/C++, program není nutné instalovat, stačí jej jen rozbalit do složky a spustit. Pro podporu vývoje aplikací pro *uLinux* na platformě *Blackfin* je nutné nainstalovat *Blackfin plugin*. Nejprve je třeba přidat adresu „<http://blackfin.uclinux.org/eclipse/>“ v okně „Install New Software“ v nabídce „Help“. Po vybrání zadané adresy v nabídce stačí označit „Blackfin GNU Toolchain“ a tlačítkem „Next“ spustit instalaci. Ukázka vzhledu okna při instalaci *toolchainu* je na obrázku 35.

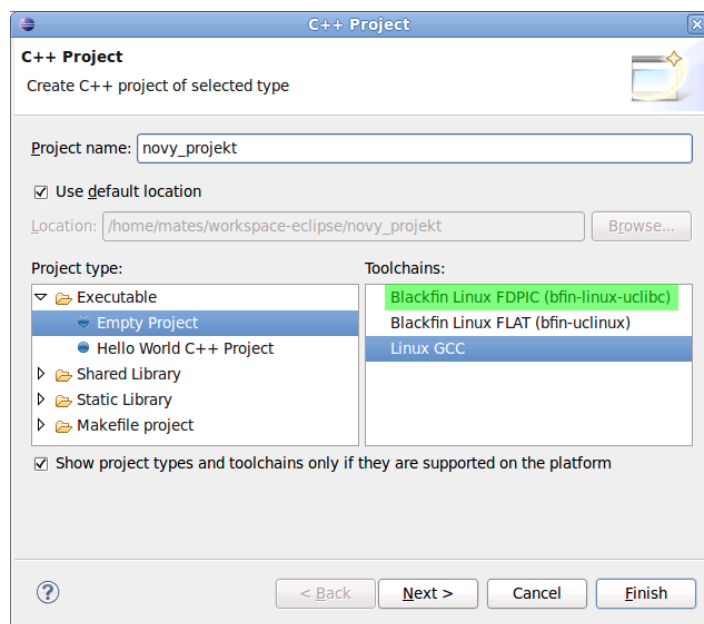
10 Integrated Development Enviroment



Obrázek 35- Instalace toolchainu Eclipse

Po úspěšné instalaci je při zakládání projektu možné zvolit kromě standardního „Linux GCC“ i „Blackfin Linux FDPIC“ nebo „Blackfin Linux FLAT“ *toolchain*. *Toolchain* „Blackfin Linux FDPIC“ je určen pro kompilaci aplikací běžících pod *uClinuxem* a „Blackfin Linux FLAT“ je určen pro kompilaci jádra *uClinuxu* a zaváděče *U-boot*.

Pro vývoj aplikací běžících pod *uClinuxem* je tedy třeba založit nový C++ projekt a jako *toolchain* zvolit „Blackfin Linux FDPIC“. Ukázka založení projektu je na obrázku 36.



Obrázek 36- Založení projektu Eclipse

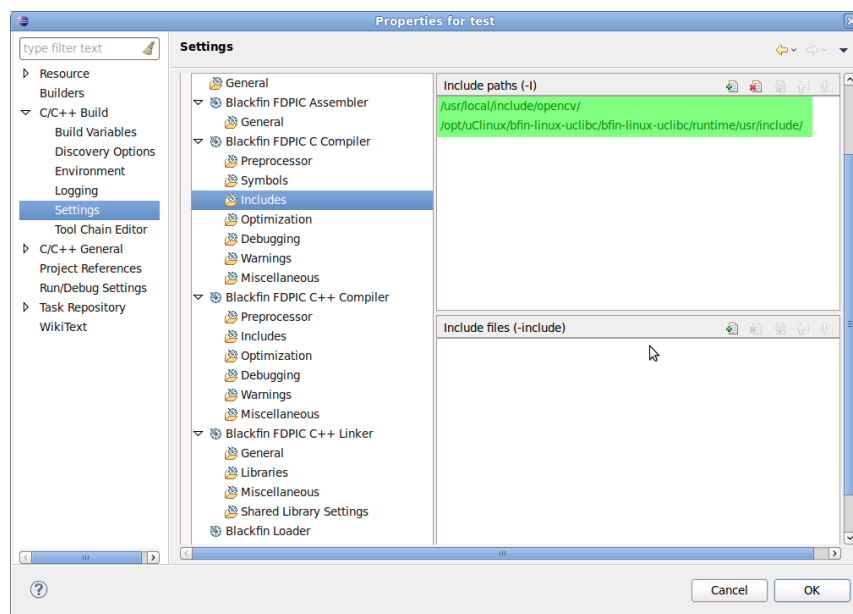
Pro použití funkcí *OpenCV* je třeba provést několik změn v nastavení projektu (Project->Properties). Nastavení se provádí v položce „Settings“ v menu „C/C++ Build“. Zde je nutné nastavit v nastavení kompilátoru cesty k hlavičkovým souborům. Při standardní instalaci jsou hlavičkové soubory *OpenCV* umístěny v:

```
/usr/local/include/opencv
```

a ostatní pro *Blackfin uClinux* v:

```
/opt/uClinux/bfin-linux-uclibc/bfin-linux-uclibc/runtime/usr/include/
```

Ukázka nastavení kompilátoru je na obrázku 37.



Obrázek 37- Nastavení kompilátoru Eclipse

V nastavení linkeru je potřeba zadat cesty ke knihovnám a ručně připojit knihovny. Standardně jsou knihovny umístěny ve složkách:

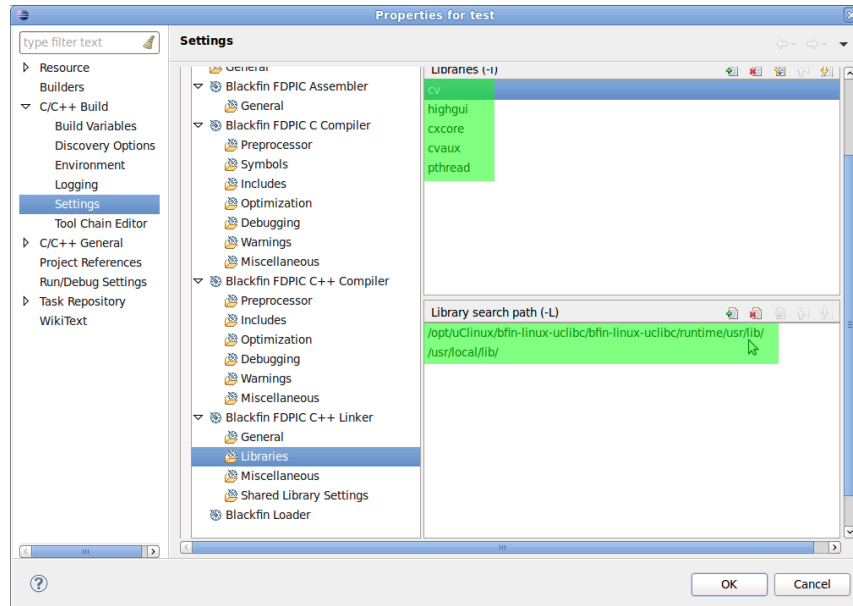
```
/usr/local/lib/
```

```
/opt/uClinux/bfin-linux-uclibc/bfin-linux-uclibc/runtime/usr/lib/
```

Následně je třeba připojit knihovny:

- cv,
- highgui,
- cxcore,
- cvaux,

- pthread,
- příčemž knihovna pthread musí být v seznamu jako poslední. Ukázka nastavení linkeru je na obrázku 38.

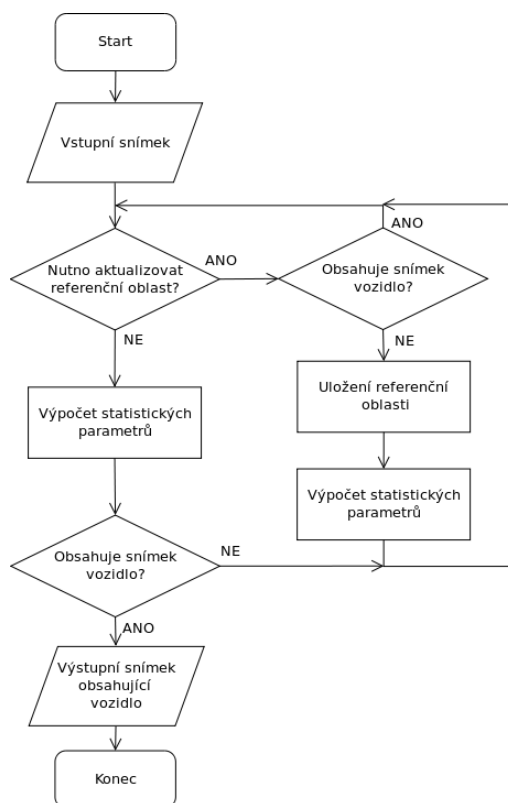


Obrázek 38- Nastavení linkeru Eclipse

Pro použití funkcí *OpenCV* stačí už jen ve zdrojovém souboru, pomocí direktivy preprocesoru „#include“, připojit hlavičkové soubory „cv.h“, „highgui.h“, „cvaux.h“ a „cxcore.h“.

9 Alternativní postup detekce vozidla

Jelikož pro účely tvorby této práce nebylo z technických důvodů možné použít žádnou z hardwarových metod detekce vozidla, byl vyvinut vlastní postup softwarové detekce pomocí referenční oblasti. Vývojový diagram algoritmu je na obrázku 39.



Obrázek 39- Vývojový diagram SW detekce

V každém snímku je pro referenční oblast vypočítán aritmetický průměr a směrodatná odchylka jasu podle vzorců 5 a 6. Rozhodovací úroveň, zda se na snímku vyskytuje vozidlo, je určena následujícím vzorcem:

$$|\text{mean}R - \text{mean}T| < 0,5 \cdot \text{stdev}R \quad (19)$$

kde $\text{mean}R$ je aritmetický průměr referenční oblasti, $\text{mean}T$ je aritmetický průměr testované oblasti a $\text{stdev}R$ je směrodatná odchylka referenční oblasti. Referenční oblast je periodicky obnovována a před uložením jako referenční, je kontrolováno, zda na snímku není vozidlo. Také je nastavena minimální vzdálenost mezi vozidly, tím se vyloučí, že jedno vozidlo bude detekováno vícekrát. Minimální vzdálenost ve snímcích je spočtena

z předpokládané rychlosti projíždějících vozidel, minimální vzdálenosti a snímkovací frekvence podle vzorce:

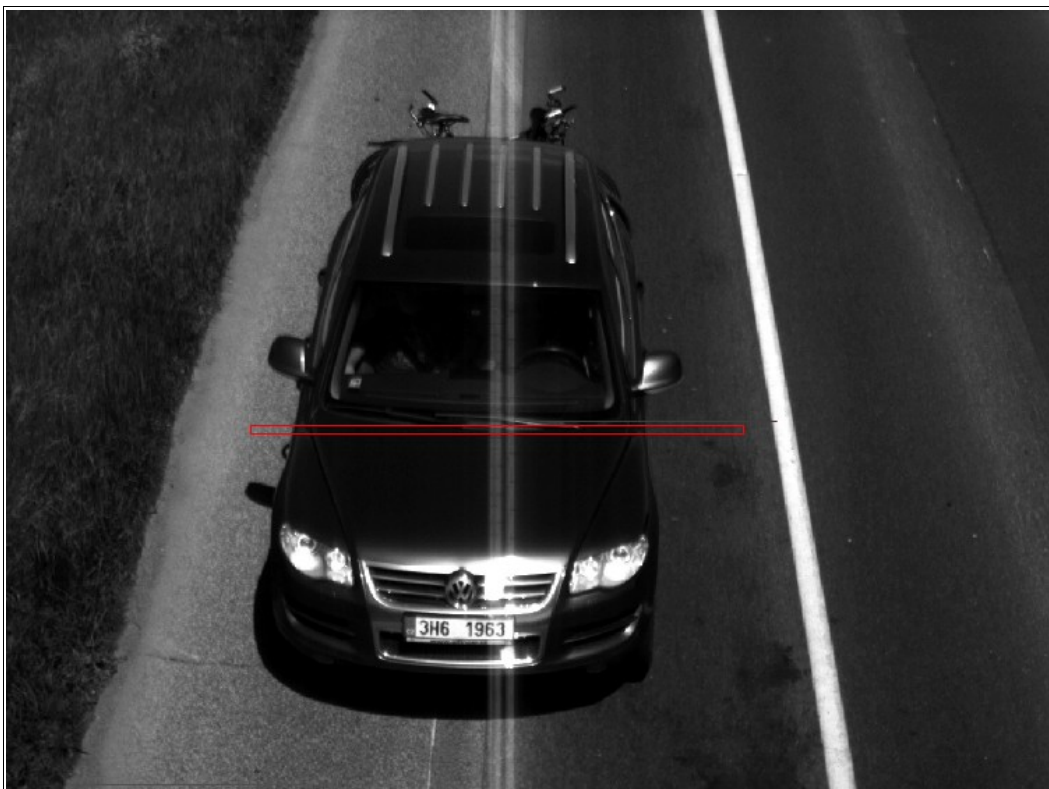
$$FrDist = \frac{MinDist \cdot 3,6}{AvSpeed} \cdot FR \quad (20)$$

kde *FrDist* je minimální vzdálenost ve snímcích, *MinDist* je minimální vzdálenost v metrech, *AvSpeed* průměrná rychlost v km/h a *FR* je snímkovací frekvence. Dlouhá vozidla, jako nákladní vozidla a autobusy jsou detekována vícekrát, ale tyto snímky budou vyloučeny v dalších fázích. Při prodlužování minimální vzdálenosti mezi vozidly hrozí, že blízko jedoucí vozidla budou chybně detekována jako jedno.

V této práci byla zvolena referenční oblast o rozměrech 600x10 obrazových bodů ve spodní části obrazu tak, aby oblast obsahovala jen jízdní pruh, ve kterém chceme vozidla detekovat. Ukázka snímku s vyznačenou referenční oblastí je na obrázku 40. Pro získání snímků pro tuto práci byly experimentálně nastaveny parametry viz. Tabulka 6. Realizace detekce vozidla v prostředí Matlab je přiložena na CD. Nevýhodou tohoto řešení je nutnost spuštění algoritmu bez přítomnosti vozidla. Výhodou naopak je jednoduchost a výpočetní nenáročnost.

Tabulka 6 - Parametry detekce vozidla

Čas do obnovení referenční oblasti	Minimální vzdálenost vozidel	Očekávaná rychlost
15 s	25 m	90 km/h



Obrázek 40- Výchozí snímek s vyznačenou referenční oblastí

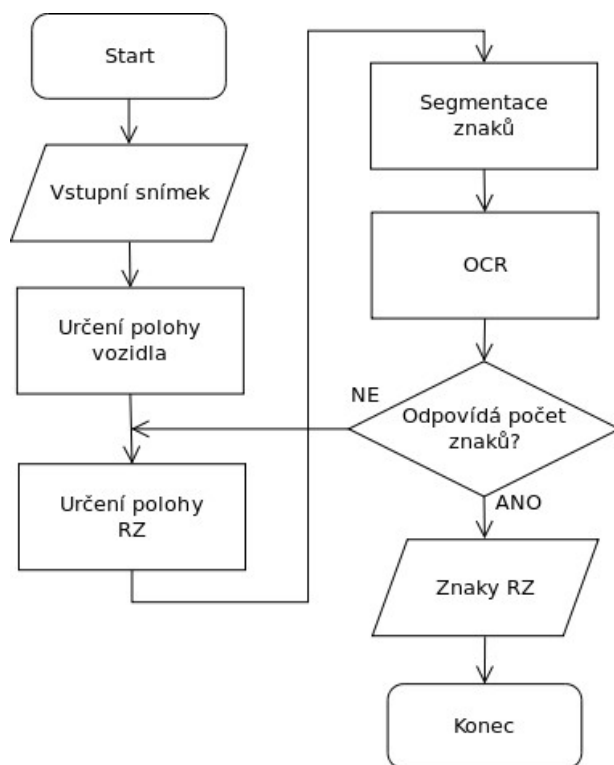
10 Implementace

Při implementaci algoritmů bylo nutné řešit několik problémů. Vzhledem k tomu, že prvotní návrh a testování algoritmů probíhaly v prostředí Matlab a implementace algoritmů pro vývojový kit je realizována v jazyce C, bylo nutné řešit implementační rozdíly mezi těmito programovacími jazyky. Hlavním rozdílem je reprezentace obrazových dat. V prostředí Matlab jsou obrazová data reprezentována maticemi, oproti tomu v jazyce C je možné využívat pouze jednorozměrná pole. Další rozdíly spočívají v rozdílné interpretaci funkcí použitých pro zpracování obrazu.

Implementaci algoritmů můžeme rozdělit do několika samostatných, po sobě následujících částí, kterými jsou:

1. Určení polohy vozidla v obraze,
2. lokalizace registrační značky,
3. segmentace jednotlivých znaků,
4. rozpoznání znaků.

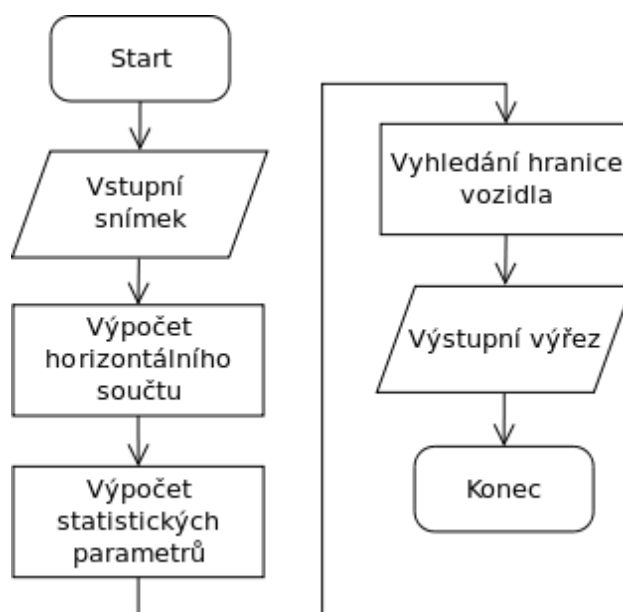
V dalších částech bude podrobně popsána implementace jednotlivých částí. Na obrázku 41 je vývojový diagram kompletního systému dekomponovaný do jednotlivých částí.



Obrázek 41- Vývojový diagram kompletního systému

10.1 Určení polohy vozidla v obraze

Poloha vozidla v obraze je určena na základě výpočtu parametrů jasu obrazových bodů vozovky a jejich rozdílu s parametry jasu obrazových bodů náležícím vozidlu. Vývojový diagram určení polohy vozidla je na obrázku 42.

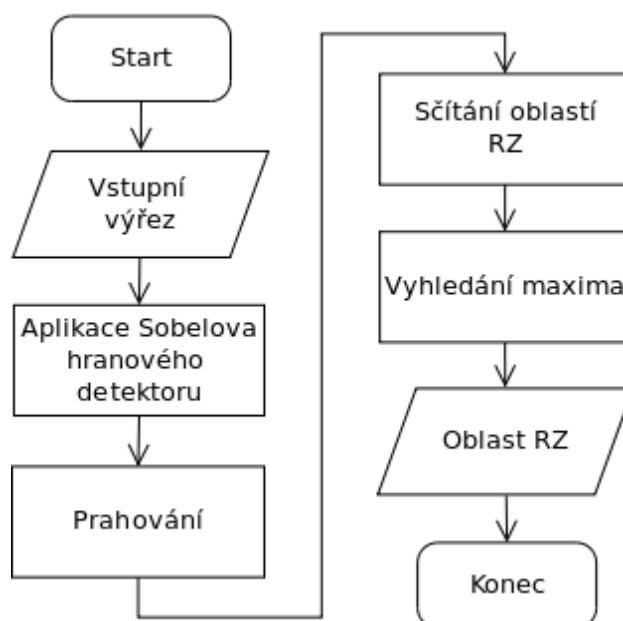


Obrázek 42- Vývojový diagram určení polohy vozidla

Jelikož knihovna openCV, na rozdíl od prostředí Matlab, neobsahuje žádnou funkci pro sečtení řádků nebo sloupců obrázku, je nutné realizovat součet řádků pomocí cyklů „for“. V cyklu jsou postupně procházeny všechny prvky pole, které jsou sčítány do pomocné proměnné a při každém násobku šířky obrázku je obsah pomocné proměnné přesunut do pole, ve kterém se součet řádků ukládá. Z tohoto pole je v cyklech „for“ vypočtena střední hodnota (\bar{x}) a směrodatná odchylka (s). V dalším cyklu „for“ je od konce hledáno kde hodnota součtu klesne pod hodnotu $\bar{x} - 5 \cdot s$. Výřez je nastaven pomocí funkce „cvSetImageROI()“, která nastaví oblast zájmu a další funkce pracují jen s touto oblastí místo celého obrazu

10.2 Lokalizace RZ

Při lokalizaci registrační značky je využito toho, že registrační značka po aplikaci hranového detektoru obsahuje velké množství vertikálních hran. Následně je v obraze vyhledávána oblast s vysokým výskytem vertikálních hran o rozměrech registrační značky. Vývojový diagram lokalizace registrační značky je na obrázků 43.

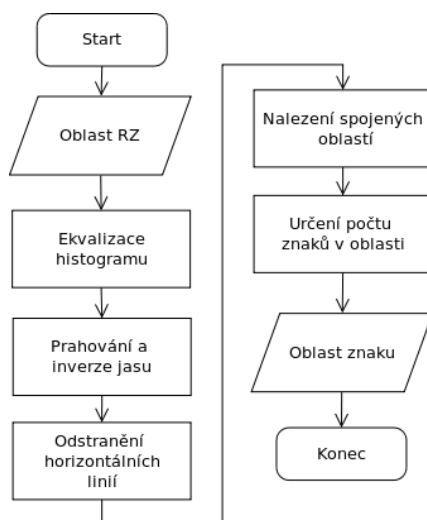


Obrázek 43- Vývojový diagram lokalizace RZ

K zvýraznění vertikálních hran je použita funkce „cvSobel()“, jelikož na rozdíl od funkce „edge()“ v prostředí Matlab není výstupem binární, ale 16-ti bitový obraz, je nutné prahováním odstranit nevýrazné hrany. Prahování je provedeno funkcí „cvThreshold()“. Při postupném procházení celého obrázku a ukládání součtu oblasti registrační značky je třeba použít tři vnořené cykly „for“ a podmínku, která zaručuje že se ukazatel, pomocí kterého se přistupuje k datům, ani na okrajích obrázku nedostane mimo alokovanou oblast. Součty jsou ukládány do pole, které má stejnou délku jako je počet obrazových bodů obrázku. Maximum a poloha maxima je hledána vzájemným porovnáváním prvků v cyklu „for“. Maximum přibližně odpovídá levému hornímu rohu registrační značky, od tohoto bodu je funkcí „cvSetImageROI()“ nastaven výřez.

10.3 Segmentace znaků

Pro segmentaci znaků je využito vyhledání a analýzy spojených oblastí. Před samotným vyhledáním spojených oblastí jsou z oblastí odstraněny dlouhé horizontální linie, což zpřesňuje segmentaci znaků. Vývojový diagram segmentace znaků je na obrázku 44.



Obrázek 44- Vývojový diagram segmentace znaků

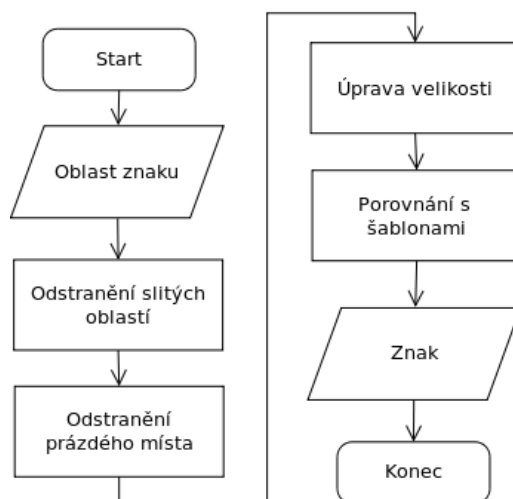
Prvním krokem segmentace znaků je ekvalizace histogramu, která je provedena funkcí „cvEqualizeHist()“. Pomocí cyklu „for“ je realizována iterační metoda určení prahu, která je popsána v kapitole 7.3. Aplikace prahu na obraz, zároveň s jeho inverzí je realizována dalším cyklem „for“. Následně vyhledávání horizontálních linií probíhá po jednotlivých řádcích pokud je nalezená linie dostatečně dlouhá je vnořeným cyklem „for“ nulována. Po odstranění horizontálních linií následuje vlastní vyhledání spojených oblastí.

Vyhledání spojených oblastí je realizováno funkcí „cvFindContours()“, která vrací ukazatele na jednotlivé spojené oblasti. Tyto oblasti jsou v následném cyklu „for“ podmínkami testovány a pokud splňují podmínky určující plochu znaku je nalezena obdélníková hranice oblasti pomocí funkce „cvBoundingRect()“. Takto nalezené hranice jsou dále testovány podmínkou, zda splňují minimální šířku a výšku znaku. Pokud oblasti podmínku splňují jsou pomocí funkce „cvDrawContours()“ vykresleny do výstupního obrazu na stejnou pozici jako v původním obraze. Následně je vypočtena předpokládaná šířka znaku a oblasti jsou testovány, zda obsahují jeden nebo více znaků. Pokud oblast obsahuje jeden znak je oblast uložena a předána funkci „ocr()“ k rozpoznání. Pokud oblast obsahuje více znaků, je oblast rovnoměrně vertikálně rozdělena na oblasti jednotlivých znaků a tyto jsou předávány funkci „ocr()“ k rozpoznání. Odstranění slitých znaků, i když logicky patří do segmentace znaků, je realizováno ve funkci „ocr()“ a popsáno v následující části.

10.4 Rozpoznání znaků

Rozpoznání znaků je realizováno deterministickou metodou „template matching“. Porovnávání znaků s šablonami je realizováno pomocí koeficientu SAD. Do této části bylo

vloženo odstranění slitých částí z předchozího kroku, protože se nachází společně se samotným rozpoznáním ve funkci „ocr()“. Vývojový diagram rozpoznání znaků je na obrázku 45.



Obrázek 45- Vývojový diagram rozpoznání znaků

Před samotným rozpoznáním znaků je ještě provedena dodatečná úprava segmentace znaku. Odstranění slitých oblastí je prováděno v horních a spodních třech řádcích, ve kterých je vyhledáváno zda je znak spojen méně než dvěma obrazovými body s okolím. Pokud je takový řádek nalezen je na něj přesunuta hranice znaku. Následně jsou nulové sloupce na okrajích oříznuty. Každá z těchto operací je realizována dvěma cykly „for“. Pro úpravu velikosti oblasti znaku na velikost šablony je provedeno funkcí „cvResize()“, bez použití interpolace. Následně je v cyklu „for“ porovnávána oblast znaku se všemi šablonami. Výběr šablony je určen regionem zájmu, který je realizován funkcí „cvSetImageROI()“. Výpočet koeficientu SAD je realizován funkcí „cvAbsDiff()“ následovanou funkcí „cvSum()“. Koeficienty SAD jsou ukládány do pole, ve kterém je následně vyhledáno minimum a poloze tohoto minima je přiřazen odpovídající znak, který je předán jako návratová hodnota funkce „ocr()“.

11 Optimalizace

Jelikož doba vykonávání algoritmu v této podobě trvala více než 10 vteřin, bylo pomocí měření času postupně zjišťováno, která část se vykonává nejdéle. Tím bylo zjištěno, že většinu času algoritmus stráví výpočtem součtu hodnot jasu oblasti o rozměrech registrační značky pro každý obrazový bod výřezu. Počet potřebných součtů před optimalizací je vypočten podle následujícího vzorce:

$$SumN = PlateW \cdot PlateH \cdot (CutW - PlateW) \cdot (CutH - PlateH) \quad (21)$$

kde $SumN$ je počet sčítání, $PlateW$ a $PlateH$ je šířka a výška registrační značky, $CutW$ a $CutH$ je šířka a výška výřezu. Pro rozměr výřezu 630x150 obrazových bodů a rozměr registrační značky 100x20 obrazových bodů je počet sčítání $1,378 \cdot 10^8$.

Pro optimalizaci bylo použito algoritmu, který poprvé zveřejnil Crow [3] v roce 1984, v širší míře se ale začal používat až po roce 2001, kdy byl představen „Viola-Jones object detection framework“ [19], který je využíván hlavně pro detekci tváří v obraze.

Nejprve je vypočítán integrální obraz podle vzorce:

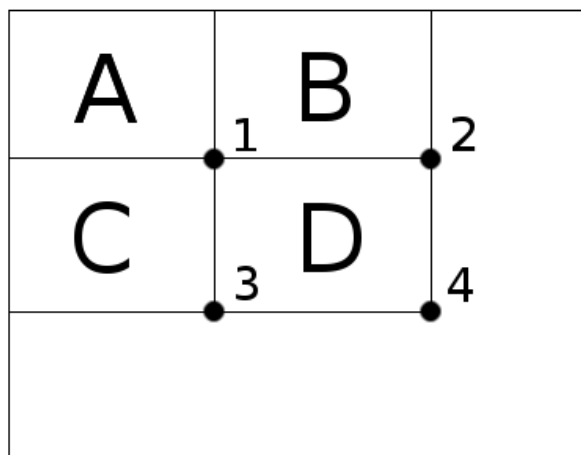
$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (22)$$

kde $ii(x, y)$ je integrální obraz a $i(x, y)$ je původní obraz. Podle následujícího výrazu je možné spočítat integrální obraz během jednoho průchodu obrazem:

$$ii(x, y) = i(x, y) + ii(x-1, y) + ii(x, y-1) - ii(x-1, y-1) \quad (23)$$

pro aplikaci tohoto algoritmu je nutné zvětšit původní obraz vlevo a nahoře o jeden obrazový bod a tuto oblast vyplnit nulami.

V takto vypočteném obraze je součet jasu obrazových bodů v obdélníkovém výběru vypočten jen dvěma součty a dvěma rozdíly bez ohledu na velikost oblasti.



Obrázek 46- Vyjádření integrálního obrazu

V integrálním obraze znázorněném na obrázku 46 bod 1 představuje součet oblasti A v původním obraze, bod 2 součet oblastí A a B, bod 3 součet oblastí A a C a bod 4 součet oblastí A, B, C a D. Součet oblasti D tedy vypočteme podle následujícího vztahu:

$$suma(D) = ii(4) - ii(3) - ii(2) + ii(1) \quad (24)$$

kde $suma(D)$ je součet oblasti D v původním obraze a ii jsou jednotlivé body v integrálním obraze.

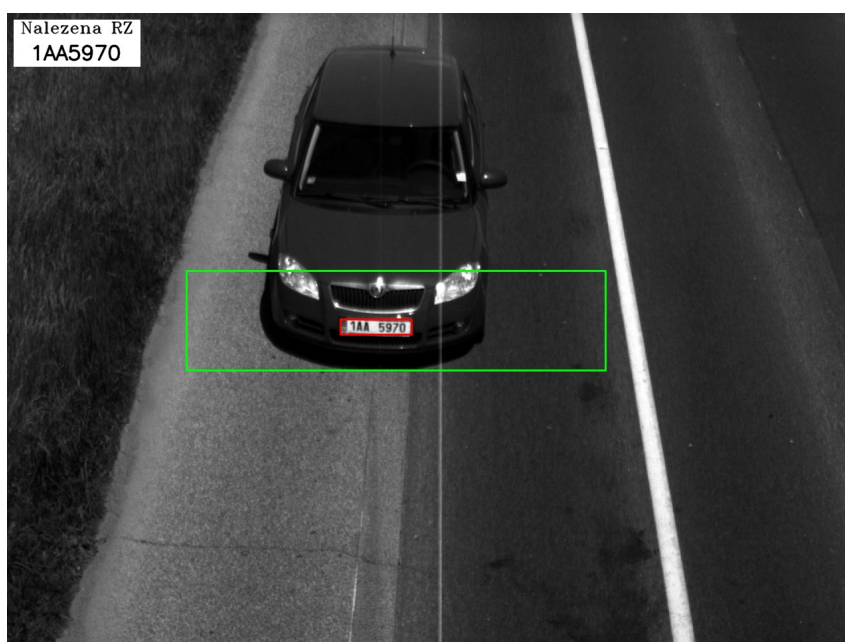
Použitím tohoto algoritmu je výrazně snížen počet sčítání. Počet potřebných součtů, resp. rozdílů po optimalizaci je vypočten podle následujícího vzorce:

$$SumN = CutW \cdot CutH \cdot 8 \quad (25)$$

kde $SumN$ je počet součtů a $CutW$ a $CutH$ je šířka a výška výřezu. Pro stejný případ, který byl uveden dříve dostáváme 16000 sčítání. Došlo tedy k snížení počtu sčítání z $1,378 \cdot 10^8$ na 16000 což představuje snížení o více než 99,98%.

12 Zhodnocení

Výstupem celého navrženého systému je série znaků představujících registrační značku, tyto znaky je možné vypisovat v konzoli, ukládat do souboru, nebo odesílat po síti. Pro účely prezentace v této práci byl ze systému také exportován snímek, na kterém byla vyznačena oblast registrační značky a vypsané rozpoznané znaky. Ukázka výstupního snímku je na obrázku 47.



Obrázek 47- Výstupní obraz

Navržený systém byl testován na souboru 100 různých vzorků vozidel. Z testování na těchto vzorcích byly vypočteny přibližné hodnoty pravděpodobnosti úspěšného rozeznání znaků a celé registrační značky a také průměrný čas zpracování. Tyto parametry společně s předpokládanými parametry jsou v tabulce 7.

Tabulka 7- Úspěšnost navrženého systému

	Úspěšnost rozeznání znaku	Úspěšnost rozeznání RZ	Čas výpočtu
Předpokládané parametry	80%	60%	4,3 s
Výsledné parametry	90%	63%	2,4 s

Úspěšnost rozpoznání znaku je vypočtena z následujícího vztahu:

$$P_{CH} = \frac{CH_{OK}}{CH_{tot}} * 100 \quad (26)$$

kde P_{CH} je úspěšnost rozpoznání znaku v procentech, CH_{OK} je počet správně rozpoznaných znaků a CH_{tot} je celkový počet znaků. Úspěšnost rozpoznání byla vypočtena následovně:

$$P_{PL} = \frac{PL_{OK}}{PL_{tot}} * 100 \quad (27)$$

kde P_{PL} je úspěšnost rozpoznání registrační značky v procentech, PL_{OK} je počet správně rozpoznaných registračních značek a PL_{tot} je celkový počet registračních značek. Za správně rozpoznanou značku je považována značka, která má všechny znaky rozpoznané správně.

Pokud by byla požadována vyšší pravděpodobnost rozpoznání registračních značek bylo by nutné výrazně zvýšit pravděpodobnost správného rozpoznání znaku. Toho by mohlo být dosaženo například použitím neuronových sítí pro rozpoznávání znaků.

Čas uvedený v tabulce je včetně přenosu snímku prostřednictvím protokolu TCP do vývojového kitu a při nalezení registrační značky na první pokus. Jelikož program pro přenos snímku byl realizován pouze pro testování a nebyl optimalizován, je zde ještě prostor pro další urychlení. Výpočet bez přenosu obrázku trvá přibližně 1,5 s.

13 Závěr

Předložená práce se zabývá problematikou rozpoznávání registračních značek motorových vozidel. V úvodní části práce je zpracována teoretická rešerše současných řešení používaných pro rozpoznávání registračních značek motorových vozidel. Analýza současných řešení se stala východiskem pro návrh vlastního algoritmu pro zpracování registračních značek.

V práci se podařilo vytvořit funkční algoritmy pro určení polohy registrační značky, segmentaci jednotlivých znaků a jejich správné rozpoznání. Vyvinuté algoritmy jsou založeny na deterministickém přístupu a pro rozpoznání jednotlivých znaků využívají porovnávání s šablonou. Tyto algoritmy byly úspěšně implementovány do vývojového kitu s operačním systémem *uCLinux* na platformě *Blackfin*.

Dále byla v práci zpracována problematika pořízení vstupních dat, včetně výběru, konfigurace a umístění snímací sestavy. Problematika pořízení obrazových dat se nakonec ukázala jako klíčová pro úspěšnost systémů rozpoznávání registračních značek.

Pro účely zpracování této práce byl také navržen program, který zajišťuje přenos obrazových dat mezi počítačem a vývojovým kitem pomocí protokolu *TCP*, který simuloval připojení ethernetové videokamery, která nebyla při tvorbě práce k dispozici.

V druhé části práce je ucelený popis postupů nutných pro vývoj aplikací využívajících knihovny funkcí *OpenCV* pro operační systém *uCLinux* na platformě *Blackfin*. Tento popis je možné využít jako příručku pro další vývoj aplikací zpracování obrazu na vestavných zařízeních.

Navržené algoritmy byly podrobeny testování na vzorku vstupních dat, přičemž rychlost výpočtu nebyla uspokojivá. Po optimalizaci kritické části algoritmu došlo k výraznému zrychlení výpočtu tak, že bez problému splňuje požadavky kladené na rychlost. Proto by bylo možné jeho použití v praxi, například pro úsekové měření rychlosti.

Všechny cíle vytyčené v úvodu práce se podařilo naplnit. Přesto je v navržených algoritmech stále značný potenciál pro optimalizace s ohledem na rychlost a robustnost detekce. Věřím, že tato práce poslouží všem případným zájemcům o tuto problematiku a pomůže v jejím dalším studiu.

Seznam použité literatury

- [1] České dálnice > Odborné informace > Intenzity dopravy. URL: <<http://www.ceskedalnice.cz/odborne-info/intenzity-dopravy>> [cit. 2010-8-6].
- [2] BOUGET, Jean-Yves. *Pyramidal Implementation of the Lucas Kanade* . URL: <http://trac.assembla.com/dilz_mgr/export/272/doc/ctl-tracking/algo_tracking.pdf> [cit. 2010-8-6].
- [3] CROW, Franklin. Summed-area tables for texture mapping. In *SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*. vol 18. no. 3. 1984.
- [4] DOBROVOLNÝ, Martin. Rychlý algoritmus rozpoznání registračních značek vozidel . *Perner's contacts*, 2009, roč. 4, č. 3. ISSN: 1801-674X.
- [5] DUAN, T, D. HONG Du, T, L. PHUOC, T, V. HOANG, N, V. Building an Automatic Vehicle License-Plate Recognition System . In *Intl. Conf. in Computer Science*. Can Tho: 2005.
- [6] ENYEDI, B. KONYHA, L. FAZEKAS, K. Real time number plate localization algorithms. *Journal of electrical engineering*, 2006, vol. 57, no. 2, 69 -77.ISSN 1335-3632 .
- [7] IWANOWSKI , Marcin. *Automatic car number plate detection using morphological image processing* . URL: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.140.5346&rep=rep1&type=pdf>> [cit. 2010-8-6].
- [8] KOVAL,V. TURCHENKO, V. KOCHAN, V. SACHENKO, A. MARKOWSKY, G. Smart License Plate Recognition System Based on Image Processing Using Neural Network. In *Proceedings of the Second IEEE International Workshop on Advanced Computing Systems* . Lviv: 2003. ISBN: 0-7803-8138-6.
- [9] MAMEDOV, F. ABU HASNA , J, F. *Character recognition using neural networks* . URL: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.86.2207&rep=rep1&type=pdf>> [cit. 2010-8-6].
- [10] ODONE, F. *Experiments on a License Plate Recognition System* . URL: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.144.972&rep=rep1&type=pdf>> [cit. 2010-8-6].
- [11] OZBAY, S. ERCELEBI E. Automatic Vehicle Identification by Plate Recognition . In *Proceedings of world academy of science, engineering and technology*. vol. 9. 2005. ISSN: 1307-6884.

- [12] QIU, Y. SUN, M. ZHOU, W. License Plate Extraction Based on Vertical Edge Detection and Mathematical Morphology . In *Computational Intelligence and Software Engineering* . Wuhan: 2009. s. 1-5. ISBN: 978-1-4244-4507-3.
- [13] RUSS, J, C. *The image processing handbook* . 5th edition. 2006. ISBN 0-8493-7254-2
- [14] SHAPIRO, V. DIMOV, D. BONCHEV, S. VELICHKOV, V. GLUHCHEV, G. Adaptive License Plate Image Extraction . In *Proceedings of the 5th international conference on Computer systems and technologies* . Rouse: 2004. s. 1-7. ISBN:954-9641-38-4.
- [15] SULEHRIA, H, K. ZHANG Ye. IRFAN, D. Mathematical Morphology Methodology for Extraction of Vehicle Number Plates . In *International journal of computers* . Issue 3. vol. 1. 2007 .
- [16] *Traffic Measurement Technology*, Bern: Federal Office of Metrology METAS, 2010, 16 str., ISBN 978-3-952-3289-6-5.
- [17] VALA, Tomáš. *Rozpoznávání SPZ z jednoho snímku*. Praha: Univerzita Karlova. Matematicko-fyzikální fakulta. Katedra softwarového inženýrství, 2006. 84 s. Vedoucí diplomové práce RNDr. Jana Štanclová .
- [18] VIBHA, L. VENKATESHA, M. PRASANTH, G, Rao. SUHAS, N. DEEPA, S. VENUGOPAL, K, R. PATNAIK, L, M. Moving Vehicle Identification using Background Registration Technique for Traffic Surveillance . In *Proceedings of the International MultiConference of Engineers and Computer Scientists 2008* . Hong Kong: 2008. ISBN: 978-988-98671-8-8 .
- [19] VIOLA, Paul. JONES, Michael. Robust Real-time Object Detection . In *Second international workshop on statistical and computational theories of vision – modeling, learning, computing, and sampling*. Vancouver: 2001.
- [20] YAGUCHI, M. OGAWA, K. Robust Vehicle Detection with Optical Flow. *Bulletin of Computational Science Research Center*, 2004, vol. 17, s. 91-96. ISSN:1347-6726.
- [21] ZHANG, Y. ZHANG , C. *A New Algorithm for Character Segmentation of License Plate*. URL:
<http://www.csai.tsinghua.edu.cn/Personal_Homepage/zhangchangshui/bigeye/paper/A%20New%20Algorithm%20for%20Character%20Segmentation%20of%20License%20Plate.pdf> [cit. 2010-8-6].

Obsah Příloženého CD

V souboru „DP_Cerny.pdf“ je text této práce.

V adresáři „detekce Matlab“ je vložen zdrojový kód (car_det.m) algoritmu alternativní detekce vozidla v obraze, realizovaný v prostředí Matlab viz. Kapitola 9.

V adresáři „program Bfin“ jsou uloženy zdrojový kód a zkompileovaný program pro vývojový kit. V souboru „test.c“ je vlastní realizace rozpoznávání značek, popsáné v kapitole 10. Soubor „test „ je zkompileovaný program pro platformu Blackfin.

V adresáři „program PC“ je zdrojový kód programu (TCPclient.c) pro přenos snímků mezi PC a vývojovým kitem, popsáný v kapitole 6.2 a zkompileovaný program pro PC.

Příloha A -Specifikace IPT-SHARK537

- Procesor
 - ADSP-BF537 rev. 0.3
 - Frekvence jádra až 600 MHz (s uClinux 525 Mhz)
 - Frekvence externí sběrnice až 133 MHz (s uClinux 131MHz)
 - 182 pinové mini-BGA pouzdro
 - 25 MHz oscilátor
- Paměť SDRAM
 - MT48LC64M8A2P-75, 2 x 64 MB paměti (celkem 128 MB)
- Paměť Serial Flash
 - M25P16, 2 MB (použita pro „u-boot“ a MAC adresu)
- Paměť NAND Flash
 - NAND01G, 128 MB (použita pro uClinux a softwarové aplikace)
- Analogové audio rozhraní
 - AD1981BL, AC '97 SoundMAX kodek
 - Plně duplexní stereo 20-ti bitový PCM DAC
- Ethernetové rozhraní (LAN)
 - LAN8700
 - Odpovídá IEEE 802.3 -2005
 - 10-BaseT a 100-BaseT
- UART
 - ADM3101EACPZ
 - RS232 prostřednictvím UART1
- USB rozhraní
 - SL811HS Cypress USB 1.1 Host/Slave kontrolér
- TFT Displej (16-ti bitové paralelní rozhraní)

- NEC NL6448BC20-18D
- Rozlišení 640 x 480 obrazových bodů (VGA)
- Dotyková obrazovka
 - Kontrolér AD7877
 - Standardní čtyř-vodičový dotykový panel
- Expandér pro sběrnici I2C
 - PCF8575 16-ti bitový I/O expandér
- MMC/SD - Card reader přes SPI
- Rozšiřující konektor
 - 16-ti bitová datová sběrnice (D0 - D15)
 - 20-ti bitová adresní sběrnice (A1 - A19)
 - Řídící signály (Reset, AMS2 – 3, AWE, ARE)
 - Sběrnice CAN
 - Sběrnice I2C
 - Přerušení pro I2C
 - Sběrnice SPI
 - UART0
- Spotřeba energie
 - Při bootování: 2,5 W
 - V nečinnosti: 2 W
 - Při maximálním jasu TFT displeje: 8,2 W
 - Při minimálním jasu TFT displeje: 4 W
 - Maximální spotřeba kitu: 9 W