Accurate contactless measurement of triangular objects composed of 3D point clouds of low quality and unclear rotation

Radek Matoušek
Faculty of Electrical Engineering and Informatics
University of Pardubice
Pardubice, Czechia
radek.matousek@upce.cz

***Abstract*—Image processing has become an important tool in various fields of human activity. One of its important applications lies in the determination of the size and dimensions of an object. Thus, in the present paper, we deal with the contactless measurement of triangular components, which was found to have certain limitations once performed on a production line in real time. These are (i) low quality output with unclear rotation in the space of a 3D scanner (missing accurate Cartesian coordinate system rotation data) and (ii) fully autonomous measurement for the selected shape of the component (no manual data tweaking as possible in the laboratory). Besides measurement, the application program contains preprocessing, such as segmentation or removing outliers. But we focus on just the key problem of measurement, i.e., finding the best-fitting hyperplane through three preprocessed point clouds and measuring the sides of the triangle projected onto the hyperplane. The problem is solved by orthogonal linear regression utilizing the Principal Component Analysis approach.**

***Index Terms*—regression, point cloud, vision system, PCA contactless measurement, best-fit, hyperplane**

I. Introduction

Image processing is a method used to perform operations on an image to enhance it or to extract useful information. It is a type of signal processing where the input is an image, and the output can be an image or characteristics/features associated with that image. Image processing has a wide range of applications across various fields. The prominent applications include medical imaging, remote sensing, computer vision, industrial automation, communication, security and surveillance, document processing, and scientific research. Image processing continues to evolve, with advancements in machine learning and artificial intelligence driving new capabilities and applications across diverse fields.

Image processing plays a crucial role in the detection of objects, which is a key component of computer vision. Object detection involves identifying and locating objects within an image or video and describing them, among others, in terms of size and dimensions. Image processing has gained a prominent position in analyzing images from 3D scanners, which became popular in various fields, including reverse engineering, quality control, and healthcare, to name a few. To analyze 3D scans, several methods exist. Here, we discuss utilization of orthogonal linear regression in combination with the Principal Component Analysis approach.

A. Description of the problem

To demonstrate the significance of the proposed approach, we scanned an object in the shape of a triangle (see Fig. 1). For our project, we decided to use linear regression to find a normal vector orthogonal to the hyperplane over three independent data sets. The used data sets are three *point clouds* captured by a 3D scanner and further extracted and filtered by the fast segmentation and outliers suppression techniques based on the various iterative methods and Random sample consensus (RANSAC). Most of the methods used were taken from the open source PCL library.

The regression analysis is a statistical methodology that utilizes the relation between two or more quantitative variables so that a response or outcome variable can be predicted from the other or others [1]. This methodology is widely used in many scientific disciplines.

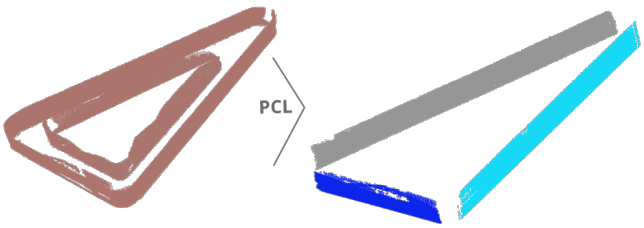

Fig. 1. Three data sets extracted from the scan of the triangular component.

The extension, linear regression with more quantitative (dependent) variables, is still a simple regression but with more explanatory variables [2]. Geometrically, such a model is represented in multidimensional space, where the regression function creates a hyperplane, as opposed to the regression line in the case of simple linear regression with only one explanatory (dependent) variable.

For example, the two explanatory variables in the model may have coordinates $\boldsymbol{x}$ and $\boldsymbol{y}$. From the statistical point of view, we can look at them as two *expected* values (parameters)

with the *observed* (measured) $\boldsymbol{z}$ values. What we are looking for is the best unbiased estimates. Then, the estimators $\widehat{\boldsymbol{\beta}}$ may be used for predicting any value on the hyperplane $\widehat{\boldsymbol{Y}} = \mathbf{X}\,\widehat{\boldsymbol{\beta}}$.

The often applied approach for solving the statistical (regression) model is *Ordinary Least Square* (OLS) method, but it is not suitable for our relatively extensive data set. In addition, we expect our model to do something different from what is common in statistical analysis.

In our model, we do not have $\boldsymbol{x}$ and $\boldsymbol{y}$ as independent parameters but expect that all three values interact and have the approximately same unknown error $\boldsymbol{\varepsilon}$. Therefore, unlike the ordinary regression, our model must have been different from the beginning. We know that in two-dimensional space, with two variables only $\boldsymbol{x}$ and $\boldsymbol{y}$, the problem could be solved by orthogonal regression, also known as the Demming regression. Its regression function is the line to which all perpendicular distances from all points are minimal. Unfortunately, the simple Demming solution is constrained on two variables only, which dit not solve our problem.

However, the orthogonal regression in multidimensional space, as for the three variables $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ in our case, can be solved by another technique. One of the most commonly used method is *Principal Components Analysis* (PCA). PCA also minimizes the perpendicular distances from the data to the fitted model, just like a Demming regression, but in any n-dimensional space; it can be easily realized with the function `princomp()` in, e.g., MATLAB.

Although it might seem that PCA could solve our problem, this is not the case. The plane of the regression function in our model has to be built by three data sets, not just one. From the outset, our idea was based on the assumption that the approximation of three planes creates a compromise that partly compensates for the lack of points on any side. This idea promised some robustness despite the missing points, which PCA cannot solve with a one common data set. As it turns out, this problem is not common, and we must have had to find special techniques. Finally, we found PCA-based solution over multiple data sets by using techniques [3] based on QR decomposition and *Singular Value Decomposition* (SVD).

B. Used methodology

The QR decomposition is often called QR factorization. Many algorithms exist for decomposition into QR matrices. The first was developed in the early twentieth century; Jørgen Pedersen Gram used the already known idea of successive or-thogonalization based projection. This procedure, including enhanced technique, is known as the Gram-Schmidt process (or procedure).

Although the newer Housholder method is the preferred method in many QR algorithms, such as the MATLAB `qr()`, in this paper we will use the original Schmidt method to explain it because of its simplicity.

After the orthogonalization to the QR matrices the matrix $\mathbf{R}$ has to be processed with *symmetrical orthogonalization* using the singular value decomposition SVD. The implementation of this function `svd()` is probably even more common than the previous one.

The SVD was introduced independently by Beltrami in 1873 and Jordan in 1874. Almost a hundred years later, Gene Golub popularized the SVD as an essential computational tool and developed the first reliable algorithms for computing it. In 1965, Gene Golub and W. Kahan published the first effective algorithm [4]. Later, another algorithms followed.

In 1967, Åke Björck published a paper called *Solving linear least squares problems by Gram-Schmidt orthogonalization* [5]. Computers greatly accelerated the development of techniques for analyzing large data sets containing a high number of dimensions. Used modified Gram-Schmidt algorithm was the first attempt at stabilization of Schmidt's algorithm. Although the computed matrix R was remarkably accurate, Q need not have to be orthogonal at all. Nevertheless, as was pointed out by Björck [5], *modified Gram Schmidt may be used to solve least squares problems*. Few years later, a review was published providing enhancement by using reorthogonalization [6]. And in the final section, Gander remarks: *We are aware that reorthogonalization doubles the computing effort and that for this reason, the method of Householder* [7] *is usually preferred. The situation is not quite as bad for Gram Schmidt if we do need the matrix Q explicitly.*

So much in a nutshell about the history of the methods we used for solving our problems. They have a rich history and are included in many programming languages or libraries. And, of course, various enhancement exist [8], improving the performance on modern computers.

For us, it was sufficient to improve performance by using so-called *reduced* QR decomposition at the cost of a slight inaccuracy. This option is often available in algorithms and is easy to enable. Reduced QR decomposition allows larger data sets to be used and has lower memory requirements.

II. Linear regression model

Multiple regression is a simple regression with multiple explanatory variables. Another extension is if qualitative variables are used instead of quantitative ones [2]. The qualitative variables are referred to as *factors*. The levels of factor create regressors called *contrasts*. In our model, data sets represent one factor with three levels. The model with factor levels is called the model with interactions and our problem can be described by this model exactly by the equation

$$\begin{aligned} Y_i = \beta_1\, I_{[I.\ dataset\,p]}(i) + \beta_2\, I_{[II.\ dataset\,q]}(i) \\ + \beta_3\, I_{[III.\ dataset\,r]}(i) + \beta_4\, x_{xi} + \beta_5\, x_{yi} + \varepsilon_i, \\ i = 1, 2, \ldots, n. \end{aligned} \quad (1)$$

The first three regressors β are factor levels. I_i are so-called dummy variables that only take the values one or zero. If the point with the given index belongs to the given data set, the value is one, otherwise the value is zero. The index i is the index of coordinates x_i, y_i, and z_i in all data sets. The first two coordinates are parameters, denoted x_{xi} and x_{yi}, and z_i

is the observed value, denoted Y_i. The matrix notation of the above equation can be written as follows

$$\underbrace{\begin{pmatrix} z_{p1} \\ z_{p2} \\ \vdots \\ z_{q1} \\ z_{q2} \\ \vdots \\ z_{r1} \\ z_{r2} \\ \vdots \\ z_{rn} \end{pmatrix}}_{\boldsymbol{Y}} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & x_{p1} & y_{p1} \\ 1 & 0 & 0 & x_{p2} & y_{p2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & x_{q1} & y_{q1} \\ 0 & 1 & 0 & x_{q2} & y_{q2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & x_{r1} & y_{r1} \\ 0 & 0 & 1 & x_{r2} & y_{r2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & x_{rn} & y_{rn} \end{pmatrix}}_{\mathbf{X}} \underbrace{\begin{pmatrix} b_p \\ b_q \\ b_r \\ m_1 \\ m_2 \end{pmatrix}}_{\widehat{\boldsymbol{\beta}}} + \underbrace{\begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}}_{\boldsymbol{\varepsilon}}. \quad (2)$$

The beta hat values are the solution of a linear system solved by OLS. The first three beta estimators b are the intercepts, and the remaining two m are the slopes of the two directional vectors of the hyperplane. It is clear that the model reduces to one hyperplane as a compromise between thhhree hyperplanes from three data sets. It is analogous to the common problem solved only in two-dimensional space, which is a school example of finding parallel lines through two or three data sets with coordinates x and y [9].

The hyperplane computed by OLS is not an orthogonal regression. The method based on QR and SVD provides an effective solution, which we describe in the next part; using only SVD on large amounts of data would be ineffective.

III. QR and SVD based solution

Our QR decomposition is based on the described stochastic model (1) with ε on the left side of the equation

$$\begin{aligned} \varepsilon_i = \beta_1\, I_{[I.\ dataset\, p]}(i) + \beta_2\, I_{[II.\ dataset\, q]}(i) \\ + \beta_3\, I_{[III.\ dataset\, r]}(i) + \beta_4\, x_{xi} + \beta_5\, x_{yi} + \beta_6\, x_{zi}, \\ i = 1, 2, \ldots, n. \end{aligned} \quad (3)$$

We write the right side to the matrix notation and denote as the matrix $\mathbf{A}$

$$\mathbf{A} = \begin{pmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{P}_p \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{P}_q \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{P}_r \end{pmatrix}. \quad (4)$$

The notation is here simplified by using block matrices in matrix $\mathbf{A}$. Three blocks inside are zero matrices with unit columns. Each of these matrices is in relation with the particular matrix of three coordinates of points belonging to one data set $\mathbf{P}_p$, $\mathbf{P}_q$ and $\mathbf{P}_r$. The matrix $\mathbf{A} \in \mathbb{R}^{n \times 2d}$ is the input matrix of the QR decomposition process of size $n \times 2d$, $n = n_p + n_q + n_r$ is the number of all points and $d = 3$ is dimension. Such a matrix has typically six columns and thousands of rows.

The QR decomposition algorithm creates two matrices $\mathbf{Q}$ and $\mathbf{R}$, which, when multiplied, are equal to the matrix $\mathbf{A}$

$$\mathbf{A} = \mathbf{QR}. \quad (5)$$

QR decomposition transforms the set of linearly independent vectors into orthonormal vectors spanning the same space as the original set. The matrix $\mathbf{Q} \in \mathbb{R}^{2d \times n}$ is an orthonormal matrix and $\mathbf{R} \in \mathbb{R}^{2d \times 2d}$ is an upper triangular matrix that we only use. The matrix is successively constructed using the Gram-Schmidt method

$$\mathbf{R} = \begin{pmatrix} \sqrt{n}_p & 0 & 0 & \bar{x}_p & \bar{y}_q & \bar{z}_r \\ 0 & \sqrt{n}_q & 0 & \bar{x}_p & \bar{y}_q & \bar{z}_r \\ 0 & 0 & \sqrt{n}_r & \bar{x}_p & \bar{y}_q & \bar{z}_r \\ 0 & 0 & 0 & ||\boldsymbol{\delta}_{\bar{\mathbf{x}}}|| & \langle \mathbf{y}, \mathbf{n}_{\boldsymbol{\delta}\bar{\mathbf{x}}} \rangle & \langle \mathbf{z}, \mathbf{n}_{\boldsymbol{\delta}\bar{\mathbf{x}}} \rangle \\ 0 & 0 & 0 & 0 & ||\boldsymbol{\delta}_{\bar{\mathbf{y}}}|| & \langle \mathbf{z}, \mathbf{n}_{\boldsymbol{\delta}\bar{\mathbf{y}}} \rangle \\ 0 & 0 & 0 & 0 & 0 & ||\boldsymbol{\delta}_{\bar{\mathbf{z}}}|| \end{pmatrix}. \quad (6)$$

By understanding the Gram-Schmidt process, it is possible to derive the individual values in the R matrix, so that have allowed us to work with individual elements. The matrix $\mathbf{R}$ consists of three blocks:

The first is an upper left (diagonal) submatrix $\mathbf{R_n} \in \mathbb{R}^{d \times d}$, producing a diagonal vector of length $N = 3$ with the size of data sets below the square root.

The next upper-right submatrix $\mathbf{R_{centers}} \in \mathbb{R}^{N \times d}$ contains the coordinates of center points; we denote this matrix as $\mathbf{C}$. For obtaining the three center points, the values must be divided first by the relevant normalizing values from the previous matrix $\mathbf{R_n}$, i.e.,

$$\mathbf{P_c} = \mathbf{C} \odot \mathrm{diag}(\mathbf{R_n}^{-1}). \quad (7)$$

The local center points have coordinates on matrix $\mathbf{P_c}$ rows. The average values of center points of three data sets can be used for the calculation of global center point $\mathbf{P_C}$ by multiplying matrix $\mathbf{C}^T$ with diagonal vector $\mathrm{diag}(\mathbf{R_n}^{-1})$ and divided by 3, i.e.,

$$\mathbf{P_C} = \frac{\mathbf{C}^T \cdot \mathrm{diag}(\mathbf{R_n}^{-1})}{3}. \quad (8)$$

The remaining lower right submatrix $\mathbf{R_H} \in \mathbb{R}^{d \times d}$ is the upper triangular matrix, which we call $\mathbf{H}$ matrix. The triangular matrix is crucial for linear regression. Therefore, we focus more on its elements, which contain the hat values and errors. These values are hidden behind what is going on in the Gram-Schmidth process. To understand them, digging deeper into the Gram-Schmidth process is necessary.

The first three steps of the Gram-Schmidth process only produce the vector of coefficients $\mathrm{diag}(\mathbf{R_n})$, which is the number of points of each data set under the square roots. The next step of the process produces the matrix $\mathbf{H}$ first value. It is the sum of the squared deviations from the means under the square roots:

$$SRSS_x = \sqrt{\sum_{j}^{N=3} \sum_{i}^{n_j} (x_{ji} - \overline{x}_j)^2}. \quad (9)$$

It is generated from three types of deviation from three data set means. Thus, the standard deviation $s_x = SRSS_x/\sqrt{n}$, which is an indicator of the amount of total variance, is approximation of three deviations belonging to different means. The common $SRSS_x$, which is the first element of the matrix $\mathbf{H}$ denoted $||\boldsymbol{\delta}_{\bar{\mathbf{x}}}||$ (6) (13), is so the numerator. The denominator is the number of total points below the square root, which is the *norm* value of $\mathrm{diag}(\mathbf{R_n})$, thus,

$$s_x = \frac{SRSS_x}{\sqrt{n}} = \frac{||\boldsymbol{\delta}_{\bar{\mathbf{x}}}||}{||\mathrm{diag}(\mathbf{R_n})||}. \tag{10}$$

We can see that our model with interaction makes it possible to balance (approximate) the variances of the three data sets, but not only them, as we will see next.

The second element on the diagonal of the matrix $\mathbf{H}$ refers not only to the simple interaction of the same coordinate types in the three datasets, but also to the interaction with the previously calculated standard deviation s_x. In general, it shows how the mean of the deviations $\overline{\mathbf{x}}$ differs from the mean of the deviations $\overline{\mathbf{y}}$. Of course the comparision is only possible with $\overline{\mathbf{y}}$ normalised to the $\overline{\mathbf{x}}$. The first normalised value $\overline{\mathbf{y}}$ is the second element of the first row $\langle \mathbf{y}, \mathbf{n}_{\boldsymbol{\delta}\bar{\mathbf{x}}} \rangle$ in matrix $\mathbf{H}$ divided by the number of elements below the square root of a standard deviation. The value represents $\overline{\mathbf{y}}$ modified by a correlation coefficient, that is

$$r_{xy} = \frac{SRSS_{xy}^2}{SRSS_x\, SRSS_y}. \tag{11}$$

So the interaction of the second element of the first row in the matrix $\mathbf{H}$ is $s_{xy} = r_{xy}s_y$. From the projection point of view, it is the projected value $\overline{\mathbf{y}}$ to the base x. The error between s_x and s_{xy} is the minimum error ϵ_{xy} (13). Similarly, the value $||\boldsymbol{\delta}_{\bar{\mathbf{y}}}||$ in matrix $\mathbf{R}$ must be divided by $\sqrt{n}$ to get meaningful error,

$$\epsilon_{xy} = \frac{||\boldsymbol{\delta}_{\bar{\mathbf{y}}}||}{||\mathrm{diag}(\mathbf{R_n})||}. \tag{12}$$

Similarly, the third column in the matrix $\mathbf{H}$. The first element of the column contains the projection of the value $\overline{z}$ into the base x. The second element contains the projection $\overline{z}$ to the base $\mathbf{y}$, but in order to make projection orthogonal to $\mathbf{x}$ the some addition error is produced. Therefore, the projected value $r_{xz}r_{\epsilon zy}s_z$ (13) is not only based on successive correlations, but also on the orthogonalization with error. Then, the total minimum error is the ϵ_{xyz}.

It become evident that by division of matrix $\mathbf{H}$ by the total number of points below the square root $\sqrt{n}$ gives the elements of the matrix more sense and makes matrix $\mathbf{R}$ more transparent. ($\epsilon_x = ||\boldsymbol{\delta}_{\bar{\mathbf{x}}}||/\sqrt{n}$, $\epsilon_{xy} = ||\boldsymbol{\delta}_{\bar{\mathbf{y}}}||/\sqrt{n}$, $\epsilon_{xyz} = ||\boldsymbol{\delta}_{\bar{\mathbf{z}}}||/\sqrt{n}$). Therefore, we will do so

$$\frac{\mathbf{H}}{\sqrt{n}} = \begin{pmatrix} \epsilon_x = s_x & s_{xy} = r_{xy}s_y & s_{xz} = r_{xz}s_z \\ 0 & \epsilon_{xy} & r_{xz}r_{\epsilon zy}s_z \\ 0 & 0 & \epsilon_{xyz} \end{pmatrix}. \tag{13}$$

The first three columns of $\mathbf{Q}$ form an orthonormal basis for the column space of $\mathbf{A}$. But we do not need the matrix $\mathbf{Q}$ matrix. Instead, we apply a symmetric orthogonalisation from the matrix $\mathbf{R}$ using SVD. This is an easy and *cheap* way to utilise the QR decomposition instead of performing the SVD on the data sets directly. The previous decomposition produces tiny triangular matrix $\mathbf{R}$ which can now be easily processed by SVD. It's worth noting that we could also use Eigen decomposition, but SVD is known to be more stable and is often preferred. The well-known SVD equation is given by the relation

$$\mathbf{H} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}. \tag{14}$$

Logically we would use the matrix $\mathbf{H}/\sqrt{n}$, but it is more accurate to use $\mathbf{H}$, which contains larger numbers. Only we must not forget to divide the matrix $\boldsymbol{\Sigma}$ by $\sqrt{n}$.

The principal components are contained in the matrix $\mathbf{V}$. The first two principal column vectors of the orthonormal matrix $\mathbf{V}$ are the vectors of the hyperplane. The third vector $\boldsymbol{v}_3$ is orthogonal to the plane. The centre of the new Euclidean space is the global centre $\mathbf{P_c}$. With these components and the origin of the new Euclidean space, we can define the equation of plane because

$$\begin{aligned} v_1 x + v_2 y + v_3 z + d &= 0, \\ v_1 + v_2 + v_3 &= 1, \\ v_1\, x_0 + v_2\, y_0 + v_3\, z_0 &= d. \end{aligned} \tag{15}$$

For getting the standard deviations from matrix $\boldsymbol{\Sigma}$, the values must be divided by $||\mathrm{diag}(\mathbf{R_n})||$, the same approach as on diagonal of matrix $\mathbf{R}$; for getting $\boldsymbol{\Sigma}$, variance matrix needs to be further squared.

The algorithm is universal and can be utilized also for getting the principal components of two sides (data sets), i.e.,

$$\mathbf{A} = \begin{pmatrix} \mathbf{1} & \mathbf{0} & \mathbf{P}_p \\ \mathbf{0} & \mathbf{1} & \mathbf{P}_q \end{pmatrix}, \tag{16}$$

or only one which is the same as traditional PCA algorithm,

$$\mathbf{A} = \begin{pmatrix} \mathbf{1} & \mathbf{P}_p \end{pmatrix}. \tag{17}$$

We used it for approximate measurement of three separate sides to get information about the dimension of the clouds of points used for the calculation.

As the normal distribution of our data sets is very flat, they can be thought of as uniform distribution that is calculated according to formula

$$\begin{aligned} \widehat{A} &= \overline{x} - \sqrt{3}\sigma, \\ \widehat{B} &= \overline{x} + \sqrt{3}\sigma. \end{aligned} \tag{18}$$

and the range

$$\widehat{B} - \widehat{A} = \Delta = 2\sqrt{3}\sigma. \tag{19}$$

As mentioned before, the standard deviations can be obtained by dividing the elements on the diagonal of the matrix $\mathbf{\Sigma}$ by $\sqrt{n}$ (or $||\mathrm{diag}(\mathbf{R_n})||$). Once we know the sigma, the values can be used to approximate the dimensions of certain sides:

$$\Delta x \approx \frac{2\sqrt{3}\,\sigma_1}{||\mathrm{diag}(\mathbf{R_n})||}, \qquad \Delta z \approx \frac{2\sqrt{3}\,\sigma_2}{||\mathrm{diag}(\mathbf{R_n})||}, \qquad \Delta y \approx \frac{2\sqrt{3}\,\sigma_3}{||\mathrm{diag}(\mathbf{R_n})||}. \tag{20}$$

The simple PCA 17 can be used to test the data set before measurement. For example, too many missing points in a corner will make the approximate height too low, which could invalidate the measurement result.

In the same spirit, the approximation can be made on the three sites at once, for example, for height and length as we used

$$\Delta z \approx \frac{2\sqrt{3}\,|||[\sigma_{3p}, \sigma_{3q}, \sigma_{3r}]||}{|||[R_{1p}, R_{1q}, R_{1r}]||}, \qquad \Delta y \approx \frac{2\sqrt{3}\,|||[\sigma_{2p}, \sigma_{2q}, \sigma_{2r}]||}{|||[R_{1p}, R_{1q}, R_{1r}]||}. \tag{21}$$

Thus, the approximated dimension of the point clouds 2 and 3 is the length $\widetilde{l} = \Delta x$ and height $\widetilde{h} = \Delta z$ and width (thickness) $\widetilde{w} = \Delta y$, the last two are made as approximation over all clouds too.

The results of our program are presented in Fig. 2, 3, and 4 and I and II. All values in tables are in milimeters if not specified.

TABLE I
NORMAL VECTORS AND CENTROIDS.

	x	y	z
$\mathbf{n}$	$-5.479e-03$	-0.09898	-0.9951
$\mathbf{Pc}$	3.4638	387.1365	73.3321
$\mathbf{n}_1$	0.810044	-0.582680	0.065670
$\mathbf{Pc}_1$	-6.1802	407.4875	71.8538
$\mathbf{n}_2$	-0.987005	0.160154	-0.013137
$\mathbf{Pc}_2$	26.004	399.359	72.735
$\mathbf{n}_3$	0.1635	0.9810	-0.1042
$\mathbf{Pc}_3$	-9.4321	354.5626	75.4074

TABLE II
DISTANCES OF INTERSECTION POINTS AND ANGLES – LEFT.
APROXIMATED DIMENSIONS OF POINT CLOUDS – RIGHT.

l_{12}	132.750 mm	$\widetilde{l}_{l12}$	98.754 mm
l_{13}	118.814 mm	$\widetilde{l}_{l13}$	84.452 mm
l_{23}	59.551 mm	$\widetilde{l}_{l23}$	44.602 mm
Θ_1	26.489 °	$\widetilde{h}$	8.450 mm
Θ_2	90.165 °	$\widetilde{w}$	0.152 mm
Θ_3	63.346 °		

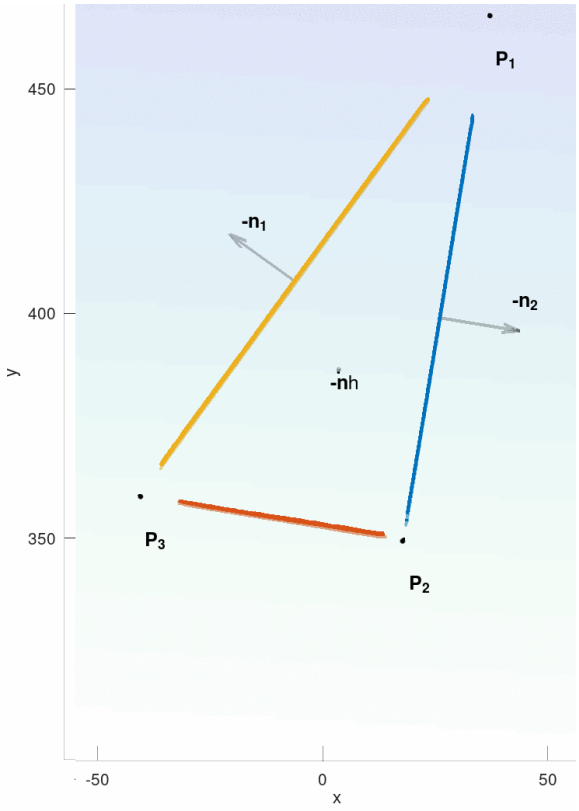

Fig. 2. The normal vectors and intersected points (top view).

IV. DISCUSSION

The use of orthogonal linear regression, as described, indicates that the absence of points on one side does not fundamentally affect the slope of the plane. This is illustrated in Figures 3 and 4, where the regression plane passes through the centroid of the side with missing points.

In addition, by computing the three planes of the specific sides according to 17, the intersection of all four regression planes can yield three measurement points, which form the vertices of the hypothetical triangle, as illustrated in Figures 2, 3, and 4. The measurement data, including central points and vectors for one measured component, are presented in Tables I and II. However, not all measured values were necessary.

Although the essential measurement of the dimension can be directly incorporated into the algorithm for finding the orthogonal planes, an additional measurement can be taken in the two-dimensional space after the projection is executed. For instance, the measurements of the three radii can be obtained from the ground plan at the origin of the common point cloud, in the direction of the normal vector of the main plane.

The application uses MATLAB-based free software Octave with built-in statistics functions `svd()` and `qr()` with enabled *reduced mode*. The script is written to be compatible with Matlab 2023.

The principal component analyses (PCA) conducted using `svd()` and `qr()` in reduced mode were performed on a dataset containing three point clouds with

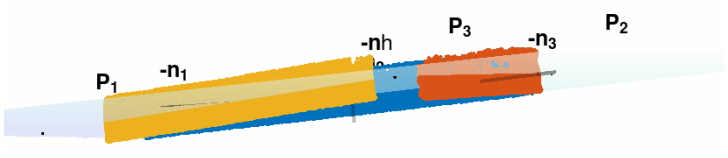

Fig. 3. The normal vectors and intersected points (side view).

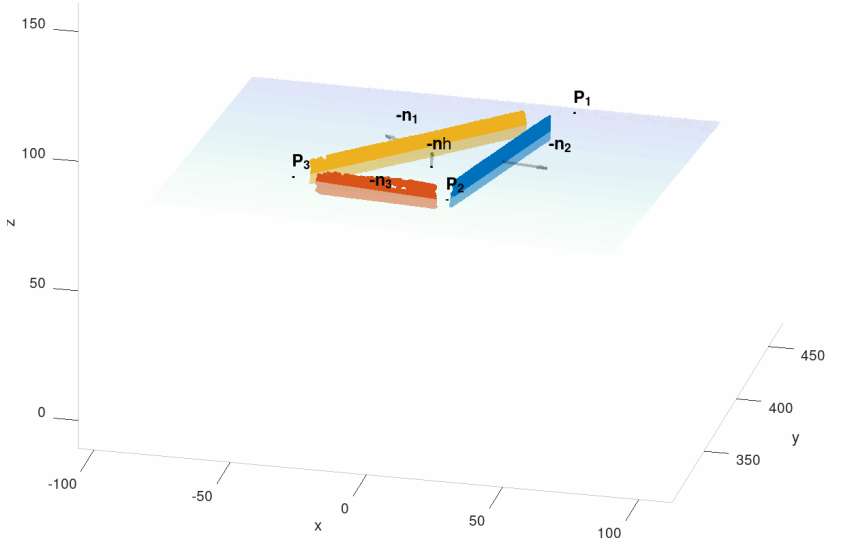

Fig. 4. The normal vectors and intersected points (side view).

approximately 150,000 points. The measurement indicates 1.8 GB (heap + bss + data) and requires 20 ms of CPU time for the computation of the main hyperplane, along with an additional 10 ms for the computation of three PCAs on the three sides. These measurement values are not limiting for us.

The MATLAB notebook (densely commented source code) is available in the repository[1] with two datasets, each containing approximately 150,000 points.

V. Conclusion

In this work, we developed an accurate contactless measurement of objects composed of 3D point clouds of low quality with unclear rotation. The method is based on orthogonal linear regression using QR factorization with symmetric orthogonalization by SVD. These methods are included in many programming languages or their libraries. The speed and memory consumption proved to be acceptable. Performance can be further improved by reducing the number of points during pre-processing using different filters or by applying the more efficient new modern parallel algorithms if necessary. The described method proved to be sufficiently robust and accurate even in the case of corrupted data sets, where some of the points are missing. We also demonstrated the essential measurement of the triangle, mainly for initial evaluation and rejection of very bad pieces. A more accurate measurement can be made in the ground plane of the projection according to the normal vector. For example, the precise measurement of the lengths of sides would be more accurate by using the linear regression with constraints in the ground plan. We experimented with this but have not included the results in this paper because the key problem in this project and paper was to find the normal vector of the ground plane. Once the ground plane is known, accurate measurements are not a problem.

[1] https://doi.org/10.5281/zenodo.13890804

References

[1] M. H. Kutner, C. J. Nachtsheim, J. Neter, and W. Li, *Applied linear statistical models*, 5th ed., Boston: McGraw-Hill, 2005, p. 2.

[2] E. Fišerová, *Lineární statistické modely*, 2nd ed., Olomouc: Univerzita Palackého v Olomouci, 2015, pp. 114–120.

[3] A. Sharma, K.K. Paliwal, S. Imoto, et al. *Principal component analysis using QR decomposition*, Int. J. Mach. Learn. & Cyber. 4, 2013, pp. 679–683.

[4] G. H. Golub, W. Kahan, *Calculating the singular values and the pseudo – inverse of a matrix*, SIAM J. Numer. Anal. Ser. B, 2, 1965, pp. 205-224.

[5] A. Bjiirck, *Solving linear least squares problems by Gram-Schmidt orthogonal – ization*, BIT, 7, 1967, pp. 1-21.

[6] W. Gander, *Algorithms for the QR-decomposition*, Technical Report 80-02, Angewandte Mathematik, ETH, 1980.

[7] A. S. Householder, *The Theory of Matrices in Numerical Analysis*, New York: Dover Publications, 1975.

[8] T. Sayadi and P. J. Schmid, "Parallel QR algorithm for data-driven decompositions," in *Pro-ceedings of the Summer Program 2014: Center for Turbulence Research*: Stanford University, 2014, pp. 335–343.

[9] W. Gander and J. Hřebíček, GANDER, Walter - HŘEBÍČEK, Jiří. *Solving problems in scientific computing using Maple and Matlab*, 4th ed., Berlin : Springer, 2004.