# General Algorithm for Gross Error Filtering Utilizing Weighted Arithmetic Mean Value Preceding Least Square Adjustment

**Tamás Jancsó**

Alba Regia Faculty, Obuda University, Budai út 45, 8000 Székesfehérvár, Hungary, jancso.tamas@amk.uni-obuda.hu

*Abstract: In the context of error detection in measurements, various methods are available depending on the magnitude of the errors. Very large gross errors, often coming from mistakes, typically need to be removed before the adjustment process. In fields such as photogrammetry and geodesy, these errors, among others, can include swapping of coordinates and misidentification of measurement points. The next two categories consist of moderate and small gross errors, which are more challenging to identify, as they result inaccuracies rather than mistakes. Assuming we have redundancy in measurements for the computation of unknown parameters, we solve the task through adjustment using the least squares method. Faulty measurements are characterized by multiples of the mean error of unit weight. We consider measurements burdened with small gross errors as those where the error magnitude exceeds three times the mean error of unit weight, but does not reach twenty times. Most established methods aim to identify and filter out these errors during the adjustment process, either in a single step or through iterative refinement by modifying the weight functions. This paper introduces an error-filtering algorithm capable of identifying small, moderate, and large gross errors before executing the adjustment. The sole requirement is that the given task can be solved with the minimum necessary number of measurements, i.e., without redundancy in measurements. After presenting the general algorithm, the effectiveness of the method is demonstrated through one example.*

*Keywords: gross error filtering; weighted arithmetic mean value; photogrammetry*

## 1 Introduction

Gross errors, also known as outliers, are measurements that significantly deviate from the expected values due to errors or anomalies in the data. Detecting and removing these gross errors is a critical step in data analysis to ensure the reliability and accuracy of results.

Specifically in photogrammetry, several gross error filtering methods and techniques are employed to detect and correct errors in measurements [2], [3], [4], [6]. Some common methods include:

• Residual Analysis: This method involves the examination of the residuals, which are the differences between observed and computed values. Large residuals can indicate the presence of gross errors.

• Checkpoints: Checkpoints are additional control points whose coordinates are known with high accuracy. They are used to assess the quality of the adjustment and can help identify gross errors when there is a significant deviation between the measured and known coordinates.

• Cross-validation: Cross-validation is a technique where different subsets of measurements are used to adjust the parameters. Discrepancies between the results from different subsets can reveal the presence of gross errors [5].

• Statistical tests: Various statistical tests, such as the Grubb's test or the T-test, can be applied to identify outliers or gross errors in the data.

• Geometric consistency checks: These checks involve assessing the geometric relationships between different points or features in the photogrammetric model. If these relationships do not conform to geometric constraints, it may indicate the presence of gross errors.

• Image Matching Algorithms: In the case of stereo photogrammetry, image matching algorithms can be used to identify inconsistencies between corresponding points in stereo image pairs. These inconsistencies can be indicative of gross errors.

• Quality control procedures: Implementing quality control procedures in data collection and processing can help prevent and identify gross errors. This includes ensuring proper sensor calibration and data collection techniques.

• Manual inspection: Visual inspection and expert judgment play a role in detecting gross errors, especially in cases where anomalies are visually apparent in the data.

• Bundle block adjustment: This is a photogrammetric technique that involves adjusting the orientation and position of all images simultaneously. Gross errors can be identified during the adjustment process [7].

Among the above methods, the algorithm proposed in this article is based on combining cross-validation and statistical test. In the following, during the detailed discussion, we will take the field of photogrammetry as a basis, but when we come to the summary of the article, we will see that the proposed method can also be used in other areas.

# 2 Algorithm

## 2.1 Cross-Validation

The specific method or combination of methods used in photogrammetry will depend on the nature of the data, the accuracy requirements, and the available tools and software. Additionally, advancements in computer vision and machine learning have led to the development of automated methods for gross error detection in photogrammetry. The cross-validation is indeed a useful technique for identifying gross errors and ensuring the reliability of measurements and parameter adjustments. When different subsets of measurements are used to adjust parameters, discrepancies in the results can indicate the presence of gross errors. Here are some methods and approaches for gross error filtering in photogrammetry using cross-validation:

- Residual analysis: When cross-validation is employed, residual analysis is a fundamental technique. It involves comparing the differences between the observed and calculated values for each measurement. Large or systematic discrepancies in residuals among different subsets can be indicative of gross errors.
- Checkpoint validation: Cross-validation often involves the use of checkpoints (additional control points with known accurate coordinates). The discrepancy between the coordinates derived from checkpoints and the results obtained from different subsets of measurements can help identify gross errors.
- K-Fold cross-validation: This method divides the dataset into 'k' subsets or folds. The parameters are adjusted 'k' times, each time using a different fold for validation and the remaining folds for adjustment. Discrepancies in the results between these iterations can reveal gross errors.
- Leave-one-out cross-validation: In this method, each measurement point is left out one at a time, and the parameters are adjusted without that particular measurement. The resulting discrepancies when the point is omitted can highlight errors associated with that specific measurement.
- Comparison of subsets: Subsets of measurements can be randomly or systematically chosen for adjustment. By comparing the parameters and residuals obtained from these subsets, inconsistencies can indicate the presence of gross errors.
- Robust estimation techniques: Cross-validation can be used in combination with robust estimation techniques, such as RANSAC (Random Sample Consensus), which iteratively selects subsets of measurements while identifying and filtering out outliers.

### 2.1.1 RANSAC Method

By applying cross-validation in photogrammetry, these methods help to systematically identify and filter out gross errors, ultimately leading to more accurate and reliable photogrammetric results. The choice of method may depend on the nature of the data, the available tools, and the specific requirements of the project.

Before I describe the combined error filtering method mentioned in the introduction in more detail, let's briefly examine one of the most well-known methods, the Random Sample Consensus (RANSAC) method [8]. By reviewing this approach, we also understand the strengths and weaknesses of this procedure.

The RANSAC method is a robust statistical technique used in various fields, including photogrammetry, to identify and filter out gross errors and outliers from datasets. RANSAC is particularly effective when dealing with data that may contain a significant number of erroneous measurements or when the distribution of errors is not known. Here's an overview of how RANSAC works when applied for gross error filtering:

1) Initialization: RANSAC begins with an initialization process by setting a few key parameters: the maximum number of iterations to perform (usually a predetermined value), a threshold value that defines the acceptance criterion for inliers, and a minimum number of data points required to fit the model (this number depends on the model being estimated).
2) Random Sample Selection: The process continues by randomly selecting a subset of data points from the entire dataset. This subset is often referred to as the "sample" or "minimal sample".
3) Model estimation: Using the randomly selected data points, RANSAC estimates a mathematical model that best fits this subset of the data. The choice of model depends on the specific problem at hand. For instance, in photogrammetry, this model could represent the transformation between image points and real-world coordinates, or the model could represent the relationships between image features, camera parameters, or other relevant parameters. The main idea is, that the model is estimated based on the randomly selected subset of data points. The number of these data points contains only the minimum number of measurements required for the solution of the given task, i.e. only as much as is necessary for the solution without the application of adjustment procedure.
4) Inlier classification: RANSAC then evaluates how well the estimated model fits the remaining data points. Data points that fall within a certain tolerance or threshold of the model are considered "inliers." These inliers are presumed to represent the "good" data, while data points that deviate significantly from the model are considered "outliers" or gross errors.

5) Model quality assessment: RANSAC performs an assessment of the quality of the model by counting the number of inliers, which is a measure of how well the model describes the "correct" data.

Steps 2 to 5 are repeated for a predetermined number of iterations. In each iteration, a new random sample is selected, a model is estimated, and the number of inliers is counted. After multiple iterations, RANSAC selects the model that has the largest number of inliers. This model is considered the best estimate for the "correct" data. RANSAC identifies the data points that are inliers of the selected model. The remaining data points, which are outliers, are considered potential gross errors or erroneous measurements.

The procedure can be refined. Depending on the application, further refinement or filtering of the gross errors may be performed. For instance, additional statistical tests or techniques may be applied to verify whether the identified outliers should be removed or retained.

RANSAC is especially useful in situations where the data may contain a significant proportion of gross errors, as it focuses on estimating models based on the consensus of the "good" data while filtering out the erroneous measurements. It is a versatile technique that can be applied in various domains, including image analysis, computer vision, and photogrammetry, to enhance the accuracy and reliability of data analysis and parameter estimation.

On the other hand, RANSAC method does not guarantee the filtering of all outliers in a dataset. The purpose of RANSAC is to identify and filter a subset of outliers, specifically those that are inconsistent with a particular model, while retaining the data points that are considered inliers. The key reasons for this are:

- Random Sampling: RANSAC selects random subsets of data points during its iterations. If there are a majority of measurements with gross errors, the random sampling process makes it possible with some probability that there will never be at least one subset that consists of only measurements without gross errors.
- Threshold setting: RANSAC employs a distance or error threshold to determine which data points are inliers. Data points that fall within this threshold distance of the estimated model are considered inliers, while those beyond the threshold are considered outliers. If the threshold is too strict, some legitimate data points may be incorrectly labeled as outliers. Conversely, if the threshold is too lenient, some true outliers may be treated as inliers.
- Minimum inlier requirement: To establish that a model is a good fit for the data, RANSAC typically requires a minimum number of inliers. If this requirement is not met during the iterations, the model may be discarded, even if it is a reasonable fit for the data. Conversely, a model with the minimum number of inliers may still leave some outliers unfiltered.

- Stochastic nature: RANSAC is a stochastic algorithm, meaning that its results can vary depending on the random subsets selected and the order of iterations. This stochastic character can result in differences in which outliers are identified and filtered in different runs of the algorithm.

While RANSAC is effective in filtering a subset of outliers that are inconsistent with the voting scheme and its main goal is not to filter all outliers with hundred percent probability, rather, the method is designed to use a specific voting scheme to find the optimal fitting result with a certain probability. If we want to increase the probability, more iterations are needed. In this respect, the method presented in this article is based on clearly identifying all gross errors based on the entire variation scheme. Thus, this method can be an alternative to the RANSAC method when analyzing a small number of data, but it is not claimed that the presented method is more accurate or less complex. The advantage of the presented method lies in the fact that the error filtering takes place before the calculation of the adjusted values. The method also includes the calculation of the adjusted values as a final step as well.

Its primary goal is to find a model that best fits the majority of the data (inliers) while isolating the most significant outliers. For the filtering of all outliers, other techniques, such as more advanced outlier detection methods, may be required.

### 2.1.2 Combination of Cross-Validation and Statistical Test

Let's review the proposed combined method. Our main goal is to detect all gross errors before the adjustment procedure is performed. Another goal is to calculate the adjusted unknown parameters the value of which should be equivalent to the result obtained by the method of least squares. The general process of error filtering consists of the following steps:

1) Derivation of a solution for the given task using only the minimally necessary $k$ measurements. The solution should be done directly, without the use of iterations.
2) Forming $g = C_k^n$ groups from $i = 1, \dots, n$ measurements in all possible combinations forming $\boldsymbol{l}_j$ vectors based on the minimally required $k$ measurements.
3) Find the solution for each $j = 1, \dots, g$ group with minimal number of measurements $k$. The solutions are stored in $\boldsymbol{x}_j$ vectors.
4) Calculate $\boldsymbol{P}_j$ weight matrix for each $j = 1, \dots, g$ group. The weight matrix is calculated by the implicit error propagation using Jacobians [9] as it is indicated in Eq. (1).

$$\boldsymbol{C}_{xj} = \boldsymbol{J}_{xj}{}^{-1}\boldsymbol{J}_{yj}\boldsymbol{C}_y\boldsymbol{J}_{yj}{}^{t}\boldsymbol{J}_{xj}{}^{-t} \rightarrow \boldsymbol{P}_j = c^2\boldsymbol{C}_{xj}{}^{-1} \quad (1)$$

In Eq. (1) $\boldsymbol{C}_y$ indicates the covariance matrix of measurements, usually it is a diagonal matrix, where the squared value of the expected $\mu_0$ standard error of measurements are placed in the diagonal. $\boldsymbol{C}_{xj}$ is the covariance matrix of unknowns. $\boldsymbol{J}_{xj}$ and $\boldsymbol{J}_{yj}$ are Jacobians of unknowns and measurements respectively. $\boldsymbol{P}_j$ is calculated with the help of inverse of $\boldsymbol{C}_{xj}$, $c$ is a proportionality factor and its value can be chosen freely, and usually its value is equal to the expected (a-priori) $\mu_0$ standard error of measurements.

5) Formation of $g' = C_{k+1}^{n}$ groups in every combination by increasing $k$ number of minimally required measurements by one. This means that the system of equations will be overdetermined. After this. we form $g'' = C_k^{k+1}$ groups from each $m' = 1, \dots, g'$ group.

6) Calculation of the unknown parameters in $\boldsymbol{x}_{m'}$ vectors for each overdetermined $m' = 1, \dots, g'$ group by calculating Jacobian weighted arithmetic mean as it is shown in Eq. (2). During the calculation of the weighted mean value, we use the previously determined $\boldsymbol{P}_j$ weight matrices and $\boldsymbol{x}_j$ solutions as $\boldsymbol{P}_{m''}$ and $\boldsymbol{x}_{m''}$ corresponding to the appropriate $m'' = 1, \dots, g''$ group.

$$\boldsymbol{x}_{m'} = \left(\sum_{m''=1}^{g''} \boldsymbol{P}_{m''}\right)^{-1} \times \sum_{m''=1}^{g''} (\boldsymbol{P}_{m''} \boldsymbol{x}_{m''}) \tag{2}$$

7) Calculation of $\boldsymbol{v}_{m'}$ vectors of residuals for each $m' = 1, \dots, g'$ group containing overdetermined measurements. Determination of $\sigma_0$ standard error of unit weight and the chi-square test value [10] for each $m'$ combination.

8) The adjustment clearly not burdened by a gross error, if $\sigma_0$ standard error of unit weight obtained after adjustment is smaller than the $c$ value taken in Eq. (1), where $c$ is equal to the expected (a-priori) $\mu_0$ standard error of measurements. If the $\sigma_0$ standard error of unit weight obtained during adjustment is greater than $\mu_0$, then we proceed as follows. Let's set up the null hypothesis to compare two standard deviations as it is shown in Eq. (3).

$$H_0 \colon c^2 = \mu_0^2 \tag{3}$$

In the case of the null hypothesis in Eq. (3), the chi-square test value is used as follows in Eq. (4).

$$\chi_f^2 = f \frac{\sigma_0^2}{\mu_0^2} \tag{4}$$

At a certain probability level of $p = 0.95\%$ and a certain degree of freedom of $f$ , the statistics is given for the comparison of the standard errors of unit weight. Let's consider this value as a theoretical value and denote it by $\chi_t^2$. At the same time, its value can be calculated based on Eq. (5).

$$\chi_c^2 = f\frac{\sigma_0^2}{c^2} \quad (5)$$

The calculated chi-square test value of $\chi_c^2$ cannot exceed the pre-set value $\chi_t^2$ according to the null hypothesis formed in Eq. (3), otherwise the corresponding combination is regarded as a solution containing gross errors. In summary, it means, the adjustment is not burdened with a gross error if the condition in Eq. (6) is true.

$$\chi_c^2 \leq \chi_t^2 \quad (6)$$

9) The measurements together with $\sigma_0$ standard error of unit weight, the chi-square test value and the decision (accepted or denied) are stored in a matrix for each $m'$ combination.

10) By examining the decision matrix using a deductive method, we can select the measurements responsible for gross errors. One possible algorithm for selecting measurements with gross error is as follows:

   - Selection of the smallest $\sigma_0$ standard error of unit weight among all combinations. The smallest $\sigma_0$ error of unit weight cannot exceed the pre-set apriori value according to the chi-square statistical test value by null hypothesis as it is explained in Eq. (6). The corresponding combination is regarded as the best estimated solution. which is free from gross errors.
   - We add the remaining measurements to the previously selected and deemed the best solution one by one. With this, we increase the number of measurements by two compared to the minimally required measurements.
   - We form all the possible combinations from this subset forming groups in every combination by increasing the number of minimally required measurements by one.
   - We check whether the given combination is incorrect based on the previously compiled decision matrix. If the answer is yes, then it is clear that the extra measurement added to the best and error-free combination causes the error and that measurement is marked as a measurement having a gross-error.

11) After excluding the measurements having gross errors, we repeat the calculation of the Jacobian mean value for the unknown parameters using all the remained groups with minimally necessary measurements and weight matrices. The obtained values will be equivalent to the values calculated by the method of least squares.

With this algorithm, we can filter out all measurements with gross errors. One of the important conditions for this is that we have at least enough error-free measurements that provide enough equations for the adjustment in at least one group among all combinations.

This algorithm can be modified by the following as an alternative error filtering method:

The steps are the same until step 8.

9) Selection of the smallest $\sigma_0$ standard error of unit weight among all combinations. The smallest $\sigma_0$ error of unit weight cannot exceed the pre-set apriori value according to the chi-square statistical test by null hypothesis as it is explained in Eq. (6). The corresponding combination is regarded as the best estimated solution. which is free from gross errors.

10) Using the best estimated solution, we calculate $v_i$ residuals for all measurements.

11) Where $v_i$ residual exceeds the preset $\varepsilon$ allowable error, the measurement is considered to have a gross error. Normally $\varepsilon$ allowable error is equal to $3\mu_0$, where $\mu_0$ denotes the estimated standard error of measurements.

12) After excluding the measurements having gross errors, we repeat the calculation of the Jacobian mean value for the unknown parameters using all the remained groups with minimally necessary measurements and weight matrices. The obtained values will be equivalent to the values calculated by the method of least squares.

# 3 Numerical Example

## 3.1 Sample Data Explanation

In photogrammetry, on a digital image with fiducial marks, the origin of the pixel coordinate system is in the upper left corner of the image, the $x$ coordinate axis points to the right, and the $y$ axis points down. The image coordinate system having $\xi$ and $\eta$ is taken at the geometric center M of the image, as shown in Figure 1.

The affine transformation can be used to convert $x, y$ pixel coordinates into $\xi, \eta$ image coordinates as it is shown in Eq. (7).

$$\begin{aligned} \xi &= a_0 + a_1 x + a_2 y \\ \eta &= b_0 + b_1 x + b_2 y \end{aligned} \tag{7}$$

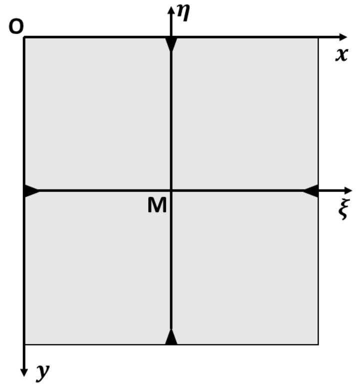


Figure 1
Pixel and image coordinate systems

The parameters of $a_0, a_1, a_2, b_0, b_1, b_2$ required for the transformation can be calculated during the interior orientation based on the fiducial marks as common points [1]. $\xi, \eta$ image coordinates of the fiducial marks are assumed to be known and error-free, which were determined during the camera calibration. $x, y$ pixel coordinates of the fiducial marks are determined manually or by automatic measurement using image matching with cross correlation technique. Our task is to filter out measurements with gross-error before finalizing the transformation parameters calculated by least-square adjustment procedure.

## 3.2 Error Filtering Process

Table 1 shows the image and pixel coordinates of the fiducial marks without gross errors. The expected $\mu_0$ measurement accuracy is 0.5 pixel, which corresponds to 0.007 mm since the pixel size is 0.014 mm.

Table 1
Dataset of fiducial marks without gross errors

| Point No. | Pixel Coordinates [pixel] | | Image Coordinates [mm] | |
|---|---|---|---|---|
| | $x$ | $y$ | $\xi$ | $\eta$ |
| 1 | 773.236 | 15751.796 | 106.006 | -106.006 |
| 2 | 15913.793 | 15708.444 | -106.003 | -106.002 |
| 3 | 15870.307 | 566.669 | -106.003 | 106.003 |
| 4 | 729.299 | 610.475 | 106.007 | 106.006 |
| 5 | 8344.258 | 16016.505 | 0.000 | -110.008 |
| 6 | 16176.831 | 8137.517 | -110.002 | -0.001 |
| 7 | 8298.946 | 303.384 | -0.001 | 110.004 |
| 8 | 466.456 | 8181.997 | 110.005 | 0.000 |

All measurements are considered to have the same weight. Adjustment according to the method of least squares gives the results indicated in Table 2.

Table 2
Affine transformation parameters derived from an error-free dataset

| $a_0$ | $a_1$ | $a_2$ | $b_0$ | $b_1$ | $b_2$ |
|---|---|---|---|---|---|
| 116.19862 | -0.01400 | 0.00004 | 114.58093 | -0.00004 | -0.01400 |

The calculated residuals are listed in Table 3. The result of $\sigma_0$ standard error of unit weight calculated on the basis of residuals was 0.006, which also corresponds to the root mean square error calculated from the residual errors as 0.006 mm. It indicates that we obtained a result better than the expected error of 0.007 mm, and our measurements are not burdened by gross errors.

Table 3
Calculated residuals without gross errors

| Point No. | Residuals [mm] | |
|---|---|---|
| | $v_x$ | $v_y$ |
| 1 | 0.0016 | 0.0046 |
| 2 | -0.0031 | 0.0009 |
| 3 | -0.0060 | 0.0077 |
| 4 | 0.0040 | -0.0020 |
| 5 | 0.0019 | -0.0032 |
| 6 | 0.0067 | -0.0051 |
| 7 | 0.0024 | -0.0035 |
| 8 | -0.0075 | 0.0007 |

After this, we change intentionally the measurements and there are gross errors in measurements of 3 points (point 2, 4 and 6) as it is seen in Table 4. The values of intentionally committed errors are also indicated in the last three columns in mm. In this case, the result of $\sigma_0$ standard error of unit weight calculated on the basis of residuals was 0.1387, which also corresponds to the root mean square error calculated from the residual errors. It indicates that there should be gross errors among the measurements. Table 5 shows the residuals. We can see that large residuals exist in every measurement compared to Table 3. Table 5 is a good example, as we can see that gross errors are distributed among all the points, and therefore it is difficult to judge which points are responsible for these errors.

After this, let's review the process of error filtering according to the proposed methodology.

Table 4
Dataset of fiducial marks with gross errors

| Point No. | Pixel Coordinates [pixel] | | Image Coordinates [mm] | | Gross errors [mm] | |
|---|---|---|---|---|---|---|
| | $x$ | $y$ | $\xi$ | $\eta$ | $d_u$ | $d_v$ |
| 1 | 773.236 | 15751.796 | 106.006 | -106.006 | | |
| 2 | 15923.793 | 15708.444 | -106.003 | -106.002 | +0.140 | |
| 3 | 15870.307 | 566.669 | -106.003 | 106.003 | | |
| 4 | 729.299 | 640.475 | 106.007 | 106.006 | | +0.420 |
| 5 | 8344.258 | 16016.505 | 0.000 | -110.008 | | |
| 6 | 16156.831 | 8137.517 | -110.002 | -0.001 | -0.280 | |
| 7 | 8298.946 | 303.384 | -0.001 | 110.004 | | |
| 8 | 466.456 | 8181.997 | 110.005 | 0.000 | | |

Table 5
Calculated residuals in case of gross errors

| Point No. | Residuals [mm] | |
|---|---|---|
| | $v_x$ | $v_y$ |
| 1 | 0.0308 | 0.0576 |
| 2 | -0.1619 | -0.0838 |
| 3 | -0.0703 | 0.0596 |
| 4 | -0.0111 | -0.2333 |
| 5 | 0.0080 | -0.0213 |
| 6 | 0.2444 | -0.0228 |
| 7 | -0.0388 | 0.1195 |
| 8 | -0.0002 | 0.1242 |

We need minimally the coordinates of three points to calculate the parameters of the affine transformation. It means, we can create 6 equations with 6 unknowns based on Eq. 7. Forming groups of 3 points in each combination, the number of combinations will be 56. We form the weight matrices for each combination based on the concepts of implicit error propagation based on Eq. (1). To do this, we first form the functions $f(\xi)$, $f(\eta)$ for every combination with three points. Eq. (8) shows an example of system of equations for points 1, 2 and 3. We create the other systems of equations for each combination group $g$ in analogous way.

$$\begin{aligned}
f(\xi)_1 &:= a_0 + a_1 x_1 + a_2 y_1 - \xi_1 = 0 \\
f(\eta)_1 &:= b_0 + b_1 x_1 + b_2 y_1 - \eta_1 = 0 \\
f(\xi)_2 &:= a_0 + a_1 x_2 + a_2 y_2 - \xi_2 = 0 \\
f(\eta)_2 &:= b_0 + b_1 x_2 + b_2 y_2 - \eta_2 = 0 \\
f(\xi)_3 &:= a_0 + a_1 x_3 + a_2 y_3 - \xi_3 = 0 \\
f(\eta)_3 &:= b_0 + b_1 x_3 + b_2 y_3 - \eta_3 = 0
\end{aligned} \tag{8}$$

Based on Eq. (8), we can form Jacobians $\boldsymbol{J}_x$ and $\boldsymbol{J}_y$ for unknowns and measurements respectively. Eq. (9)-(10) and (11)-(12) show Jacobians $\boldsymbol{J}_x$ and $\boldsymbol{J}_y$ for group of points 1, 2 and 3 as an example.

$$\boldsymbol{J}_{x(1,2,3)} = \begin{bmatrix} \frac{\partial f(\xi)_1}{\partial a_0} & \frac{\partial f(\xi)_1}{\partial a_1} & \frac{\partial f(\xi)_1}{\partial a_2} & \frac{\partial f(\xi)_1}{\partial b_0} & \frac{\partial f(\xi)_1}{\partial b_1} & \frac{\partial f(\xi)_1}{\partial b_2} \\ \frac{\partial f(\eta)_1}{\partial a_0} & \frac{\partial f(\eta)_1}{\partial a_1} & \frac{\partial f(\eta)_1}{\partial a_2} & \frac{\partial f(\eta)_1}{\partial b_0} & \frac{\partial f(\eta)_1}{\partial b_1} & \frac{\partial f(\eta)_1}{\partial b_2} \\ \frac{\partial f(\xi)_2}{a_0} & \frac{\partial f(\xi)_2}{a_1} & \frac{\partial f(\xi)_2}{a_2} & \frac{\partial f(\xi)_2}{b_0} & \frac{\partial f(\xi)_2}{b_1} & \frac{\partial f(\xi)_2}{b_2} \\ \frac{\partial f(\eta)_2}{\partial a_0} & \frac{\partial f(\eta)_2}{\partial a_1} & \frac{\partial f(\eta)_2}{\partial a_2} & \frac{\partial f(\eta)_2}{\partial b_0} & \frac{\partial f(\eta)_2}{\partial b_1} & \frac{\partial f(\eta)_2}{\partial b_2} \\ \frac{\partial f(\xi)_3}{\partial a_0} & \frac{\partial f(\xi)_3}{\partial a_1} & \frac{\partial f(\xi)_3}{\partial a_2} & \frac{\partial f(\xi)_3}{\partial b_0} & \frac{\partial f(\xi)_3}{\partial b_1} & \frac{\partial f(\xi)_3}{\partial b_2} \\ \frac{\partial f(\eta)_3}{\partial a_0} & \frac{\partial f(\eta)_3}{\partial a_1} & \frac{\partial f(\eta)_3}{\partial a_2} & \frac{\partial f(\eta)_3}{\partial b_0} & \frac{\partial f(\eta)_3}{\partial b_1} & \frac{\partial f(\eta)_3}{\partial b_2} \end{bmatrix} \tag{9}$$

$$\boldsymbol{J}_{x(1,2,3)} = \begin{bmatrix} 1 & x_1 & y_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_1 & y_1 \\ 1 & x_2 & y_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_2 & y_2 \\ 1 & x_3 & y_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_3 & y_3 \end{bmatrix} \tag{10}$$

$$\boldsymbol{J}_{y(1,2,3)} = \begin{bmatrix} \frac{\partial f(\xi)_1}{\partial x_1} & \frac{\partial f(\xi)_1}{\partial y_1} & \frac{\partial f(\xi)_1}{\partial x_2} & \frac{\partial f(\xi)_1}{\partial y_2} & \frac{\partial f(\xi)_1}{\partial x_3} & \frac{\partial f(\xi)_1}{\partial y_3} \\ \frac{\partial f(\eta)_1}{\partial x_1} & \frac{\partial f(\eta)_1}{\partial y_1} & \frac{\partial f(\eta)_1}{\partial x_2} & \frac{\partial f(\eta)_1}{\partial y_2} & \frac{\partial f(\eta)_1}{\partial x_3} & \frac{\partial f(\eta)_1}{\partial y_3} \\ \frac{\partial f(\xi)_2}{\partial x_1} & \frac{\partial f(\xi)_2}{\partial y_1} & \frac{\partial f(\xi)_2}{\partial x_2} & \frac{\partial f(\xi)_2}{\partial y_2} & \frac{\partial f(\xi)_2}{\partial x_3} & \frac{\partial f(\xi)_2}{\partial y_3} \\ \frac{\partial f(\eta)_2}{\partial x_1} & \frac{\partial f(\eta)_2}{\partial y_1} & \frac{\partial f(\eta)_2}{\partial x_2} & \frac{\partial f(\eta)_2}{\partial y_2} & \frac{\partial f(\eta)_2}{\partial x_3} & \frac{\partial f(\eta)_2}{\partial y_3} \\ \frac{\partial f(\xi)_3}{\partial x_1} & \frac{\partial f(\xi)_3}{\partial y_1} & \frac{\partial f(\xi)_3}{\partial x_2} & \frac{\partial f(\xi)_3}{\partial y_2} & \frac{\partial f(\xi)_3}{\partial x_3} & \frac{\partial f(\xi)_3}{\partial y_3} \\ \frac{\partial f(\eta)_3}{\partial x_1} & \frac{\partial f(\eta)_3}{\partial y_1} & \frac{\partial f(\eta)_3}{\partial x_2} & \frac{\partial f(\eta)_3}{\partial y_2} & \frac{\partial f(\eta)_3}{\partial x_3} & \frac{\partial f(\eta)_3}{\partial y_3} \end{bmatrix} \tag{11}$$

$$\boldsymbol{J}_{y(1,2,3)} = \begin{bmatrix} a_1 & a_2 & 0 & 0 & 0 & 0 \\ b_1 & b_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_1 & a_2 & 0 & 0 \\ 0 & 0 & b_1 & b_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_1 & a_2 \\ 0 & 0 & 0 & 0 & b_1 & b_2 \end{bmatrix} \tag{12}$$

After this, we calculate the transformation parameters as weighted mean values for each combination with 3 points. Then we form groups of 4 points in each combination, where the number of combinations will be 70. We calculate the weighted mean values of parameters for each combination with 4 points. Based on these parameters, we calculate the residuals in each combination of 4 points. Using the residuals, we calculate the standard errors of unit weight for each case.

We calculate the chi-square test values for each combination, and comparing them with the pre-set the chi-square test value, we can decide the result is free from gross errors or not by null-hypothesis. The results of the accepted combinations are summarized in Table 6 out of all 70 possible combinations formed from 4 points, where the chi-square test values are under the previously set value of $\chi_t^2$ =5.990. In any other combination of 4 points, there is a chance for a measurement with gross error at some point or points.

Table 6
Accepted combinations of 4 points

| Combinations of 4 Points | Standard Error of Unit Weight ($\sigma_0$) | Chi-square test value ($\chi_c^2$) |
|---|---|---|
| 1 3 5 7 | 0.0074 | 2.223 |
| 1 3 5 8 | 0.0047 | 0.895 |
| 1 3 7 8 | 0.0067 | 1.819 |
| 1 5 7 8 | 0.0045 | 0.819 |
| 3 5 7 8 | 0.0079 | 2.561 |

The smallest value of standard error of unit weight is 0.0045 and it belongs to the combination having points 1, 5, 7 and 8. The chi-square test value is 0.819, which is less than the previously set value of 5.990. It means we can accept this combination and we can regard it as the best estimated solution, which is free from gross errors. After that, by assigning the remaining points to the selected 4 points one by one, we can form 4 combinations of 5 points, where only one point differs from the other combinations. We calculate the chi-square test values for these combinations four times, since we can form four combinations from 5 points where the added point is always included. Comparing the chi-square values with the pre-set value of 5.990, we conclude that point 3 is free form gross-errors, but points 2, 4, 6 are having gross errors as it is seen in Table 7.

Table 7
Results of combinations consisting 5 points

| Point No. to be investigated | Combinations of 5 Points | Combinations of 4 Points | Standard Error of Unit Weight ($\sigma_0$) | Chi-square test value ($\chi_c^2$) |
|---|---|---|---|---|
| 2 | 1 2 5 7 8 | 1 2 5 7 | 0.0426 | 73.993 |
| | | 1 2 5 8 | 0.0412 | 69.464 |
| | | 1 2 7 8 | 0.0244 | 24.349 |
| | | 2 5 7 8 | 0.0525 | 112.686 |
| 3 | 1 3 5 7 8 | 1 3 5 7 | 0.0074 | 2.223 |
| | | 1 3 5 8 | 0.0047 | 0.895 |
| | | 1 3 7 8 | 0.0067 | 1.819 |
| | | 3 5 7 8 | 0.0079 | 2.561 |
| 4 | 1 4 5 7 8 | 1 4 5 7 | 0.1536 | 963.263 |
| | | 1 4 5 8 | 0.1194 | 581.387 |

| | | | | |
|---|---|---|---|---|
| | | 1 4 7 8 | 0.1288 | 677.456 |
| | | 4 5 7 8 | 0.1933 | 1524.465 |
| 6 | 1 5 6 7 8 | 1 5 6 7 | 0.0933 | 355.553 |
| | | 1 5 6 8 | 0.0625 | 159.286 |
| | | 1 6 7 8 | 0.0438 | 78.193 |
| | | 5 6 7 8 | 0.0973 | 386.603 |

After excluding these points, we repeat the calculation of the weighted mean value and using the adjusted parameters, we calculate the final residuals. There are residuals marked with asterisks which exceed three times the expected error in Table 8. We can conclude that the gross-error detection was successful, the residual values calculated at wrong points are close to the real errors (see also Table 4).

Table 8

Calculated residuals in case of gross error detection

| Point No. | Residuals [mm] | | Gross errors [mm] | |
|---|---|---|---|---|
| | $v_x$ | $v_y$ | $d_u$ | $d_v$ |
| 1 | 0.0017 | 0.0044 | | |
| 2 | -0.1451* | -0.0022 | +0.140 | |
| 3 | -0.0036 | 0.0049 | | |
| 4 | 0.0097 | -0.4223* | | +0.420 |
| 5 | 0.0009 | -0.0046 | | |
| 6 | 0.2870* | -0.0071 | -0.280 | |
| 7 | 0.0060 | -0.0051 | | |
| 8 | -0.0051 | 0.0004 | | |

Table 9 proves that the calculated parameters after excluding the points with gross-errors show a good fit with the parameters calculated from the error-free data set, and the standard error of unit weight is the same as the pre-estimated error.

Table 9

Affine transformation parameters derived from an error-free dataset

| Parameters | $a_0$ | $a_1$ | $a_2$ | $b_0$ | $b_1$ | $b_2$ |
|---|---|---|---|---|---|---|
| Gross-error free solution | 116.19862 | -0.01400 | 0.00004 | 114.58093 | -0.00004 | -0.01400 |
| Solution after detection | 116.20344 | -0.01400 | 0.00004 | 114.58073 | -0.00004 | -0.01400 |
| Difference | -0.00482 | 0.00000 | 0.00000 | 0.00020 | 0.00000 | 0.00000 |

## 3.2 Error Filtering Process with RANSAC

As a comparison, let's examine the example presented in the previous chapter using the RANSAC method. The first step is the initialization. We need to set the following parameters:

$s$ – the smallest number of points required to solve affine transformation

$d$ – the threshold used to identify a point that fits well to the solution

$T$ – the number of nearby points required to assert a model fits well

$N$ – the number of iterations required

In our case, the value of $s$ is 3, since at least this number of points is required to solve the affine transformation without adjustment. The value of $d$ can be chosen in many ways, for example based on the gross-error filtering theory, the value of $d$ should be five times the mean error of the expected image coordinate measurement, since exceeding this value the measurements are burdened by gross error. Accordingly, $d = 5 * 0.007\ mm$, i.e. $d = 0.035\ mm$. The value of $T$ depends on how many points are assumed to have gross error. If $n$ is the number of points, among which $m$ point are assumed to be faulty points, then $T = n - m$. Or to put it another way: let $e$ denote the probability that a point is an outlier, then $T = (1 - e) \cdot n$. The probability $e$ can be calculated as $e = o/n$ where $o$ denotes the number of faulty points as assumed. We decide the value of $o$ and setting this value has a big impact on the reliability of the RANSAC method. Since the number of outlier points in the previous chapter was 3, let's set this value here as well. Then. the value $e$ will be 3/8, i.e. 0.375. Accordingly, $T$ is 5. Value $T$ is used for early termination, which means we terminate the error detection process when the number of inliers reaches the value of $T$. We choose $N$ so that, with probability $p$, at least one random sample is free from outliers. e.g. $p = 0.95$. $N$ can be calculated by Eq. (7) [11].

$$N = \frac{\log(1-p)}{\log(1-(1-e)^s)} \tag{7}$$

The result must be rounded up to an integer number. According to our example $N = \frac{\log(1-0.95)}{\log(1-(1-0.375)^3)} = 11$.

After running the RANSAC algorithm, the result was the same as Table 8. Between multiple runs, there was only a difference in which of the 11 random samples the early termination occurred. As an example, let's look at one of the report received after the run in Fig. 2. We can see from the report that the 11 randomly selected samples did not have to be carried through since $T$=5 condition was already fulfilled at the 4th sample, so we were able to close the process with an early termination. Compare the obtained results with the values obtained in Tables 8 and 9. We can conclude that we got exactly the same solution.

During the execution of the program, we used the assumption that the correct number of errors was $o$=3, and accordingly e=0.375 and $N$=11. Now let's see what happens when the number of outlier points is $o$=2. Then $e$=0.25, $N$=6, $T$=6. In this case the result is varied.

```
******* RANSAC FOR AFFINE TRANSFORMATION v1.0 *******

Initialization:
s=3
p=0.95
e=0.375
d=0.35 mm
N=11
T=5

MXY(image)=0.007 mm

No.  x[pixel]    y[pixel]      xi[mm]        eta[mm]
----------------------------------------------------
1      773.236   15751.796     106.006      -106.006
2    15923.793   15708.444    -106.003      -106.002
3    15870.307     566.669    -106.003       106.003
4      729.299     640.475     106.007       106.006
5     8344.258   16016.505       0.000      -110.008
6    16156.831    8137.517    -110.002        -0.001
7     8298.946     303.384      -0.001       110.004
8      466.456    8181.997     110.005         0.000

Variations (0: outlier, 1: inlier):

No./Point No. 1 2 3 4 5 6 7 8
-----------------------------
1.            0 0 1 0 1 1 0 0
2.            0 1 1 0 0 1 0 0
3.            0 1 0 1 0 0 0 1
4.            1 0 1 0 1 0 1 1
MAXIN= 5  NVARIIND= 4  MINM0= 0.16862

Outlier points:
2 4 6

Parameters:
A0= 116.20344381149
A1=  -1.40030531636249E-2
A2=   4.01191306309966E-5
B0= 114.580733434457
B1=  -4.02371509058173E-5
B2=  -1.40016565376819E-2

Errors:

No.           Ex[mm]              Ey[mm]
-----------------------------------------
1             0.00173             0.00438
2*           -0.14507*           -0.00223
3            -0.00357             0.00485
4*            0.00973            -0.42232*
5             0.00092            -0.00462
6*            0.28695*           -0.00709
7             0.00603            -0.00507
8            -0.00511             0.00045

M0=0.00650 mm
```

Figure 2

RANSAC report with early termination of random sample number 4

```
******* RANSAC FOR AFFINE TRANSFORMATION v1.0 *******

Initialization:

s=3
p=0.95
e=0.25
d=0.35 mm
N=6
T=6

MXY(image)=0.007 mm

No.  x[pixel]     y[pixel]      xi[mm]        eta[mm]
-----------------------------------------------------
1      773.236    15751.796     106.006      -106.006
2    15923.793    15708.444    -106.003      -106.002
3    15870.307      566.669    -106.003       106.003
4      729.299      640.475     106.007       106.006
5     8344.258    16016.505       0.000      -110.008
6    16156.831     8137.517    -110.002        -0.001
7     8298.946      303.384      -0.001       110.004
8      466.456     8181.997     110.005         0.000

Variations (0: outlier, 1: inlier):

No./Point No. 1 2 3 4 5 6 7 8
-----------------------------
1.            0 1 0 0 0 1 1 0
2.            0 1 1 0 0 0 1 1
3.            0 0 0 1 1 1 0 0
4.            1 0 0 0 0 1 0 1
5.            0 0 0 1 1 0 1 0
6.            0 0 1 0 0 1 0 1
MAXIN= 4  NVARIIND= 2  MINM0= .18340

Outlier points:
1 4 5 6

Parameters:

A0= 116.145095591538
A1=  -1.39987601231366E-2
A2=   4.8478711520329E-5
B0= 114.581817664936
B1=  -4.03695521576296E-5
B2=  -1.40015341264301E-2

Errors:

No.            Ex[mm]              Ey[mm]
-------------------------------------------
1*             0.07838*            0.00729
2             -0.00374            -0.00133
3              0.01095             0.00391
4*            -0.04014*           -0.42126*
5*             0.11229*           -0.00268
6*             0.36599*           -0.00715
7             -0.01415            -0.00505
8              0.00694             0.00248

M0= 0.01468
```

Figure 3

RANSAC report with the assumed outliers of o=2

```
******* RANSAC FOR AFFINE TRANSFORMATION v1.0 *******

Initialization:

s=3
p=0.95
e=0.25
d=0.35 mm
N=6
T=6

MXY(image)=0.007 mm

No.  x[pixel]    y[pixel]      xi[mm]        eta[mm]
-----------------------------------------------------
1      773.236   15751.796     106.006      -106.006
2    15923.793   15708.444    -106.003      -106.002
3    15870.307     566.669    -106.003       106.003
4      729.299     640.475     106.007       106.006
5     8344.258   16016.505       0.000      -110.008
6    16156.831    8137.517    -110.002        -0.001
7     8298.946     303.384      -0.001       110.004
8      466.456    8181.997     110.005         0.000

Variations (0: outlier, 1: inlier):

No./Point No. 1 2 3 4 5 6 7 8
-----------------------------
1. 0 1 1 1 0 0 0 0
2. 0 0 0 0 0 1 1 1
3. 0 1 1 0 1 0 0 0
4. 0 1 0 0 1 1 0 0
5. 0 0 1 1 0 1 0 0
6. 0 0 1 0 0 1 1 0
MAXIN= 3  NVARIIND= 3   MINM0= .22148

Outlier oints:
1 4 6 7 8

Too many outliers!
```

Figure 4

RANSAC report with too many outliers

```
******* RANSAC FOR AFFINE TRANSFORMATION v1.0 *******

Initialization:

s=3
p=0.95
e=0.5
d=0.35 mm
N=23
T=4

MXY(image)=0.007 mm

No.  x[pixel]    y[pixel]      xi[mm]        eta[mm]
----------------------------------------------------
1      773.236   15751.796    106.006      -106.006
2    15923.793   15708.444   -106.003      -106.002
3    15870.307     566.669   -106.003       106.003
4      729.299     640.475    106.007       106.006
5     8344.258   16016.505      0.000      -110.008
6    16156.831    8137.517   -110.002        -0.001
7     8298.946     303.384     -0.001       110.004
8      466.456    8181.997    110.005         0.000

Variations (0: outlier, 1: inlier):

No./Point No. 1 2 3 4 5 6 7 8
-----------------------------
1.            1 0 1 0 0 0 1 1
MAXIN= 4   NVARIIND= 1   MINM0= .16696

Outlier points:
2 4 5 6

Parameters:

A0= 116.201461820112
A1= -1.40028886945493E-2
A2=  4.03075152328438E-5
B0=  114.5906418116
B1= -4.10593651934827E-5
B2= -1.40025983105388E-2

Errors:

No.            Ex[mm]              Ey[mm]
-----------------------------------------
1              0.00284            -0.00118
2*            -0.14147*           -0.02021
3             -0.00284             0.00118
4*             0.00799            -0.41362*
5*             0.00333            -0.01665
6*             0.28916*           -0.01813
7              0.00547            -0.00227
8             -0.00547             0.00227

M0=  0.00668
```

Figure 5
RANSAC report with four outliers

Sometimes we get the same result as in Fig. 2, sometimes we get false results as it is seen in Figs. 3 and 4. In Fig. 3 the result shows 4 outliers, which is not true. In Fig. 4 we got even 5 outliers, which means there are only 3 points remained for the affine transformation and it is not enough number of points for an adjustment procedure. Finally, in Fig. 5 we see the result when the half of points (four points) are assumed to be wrong. Even in some case we get the same result as in Fig. 3.

In summary, we can state that the RANSAC method is sensitive to the setting of the initial parameters. A reliable and repeatable good result is obtained if we can precisely set the number of points considered faulty. In other cases, if the value of $o$ is taken to be smaller or larger, the good result is obtained in the majority of cases, but not in all cases. It is important to note that in the examples presented, the ratio of outlier points to total points is higher than usual. If the RANSAC method provides the correct result, the values of the obtained parameters and standard errors are the same as the results of the method presented in this paper. Furthermore, it can be stated that the RANSAC method cannot be considered more complex from a programming point of view compared to the presented method. In my opinion, the two methods are similar in complexity. The complexity of the programs is well reflected if we compare the running times. Running the presented example on the computer with both methods resulted in similar running times. Using the RANSAC method, the running time was 336 msec without early termination. Due to its early termination, running times varied, with an average of 150 msec. The running time of the program based on weighted arithmetic mean value was 249 msec on average. This time also includes the calculation of adjusted values as well.

**Conclusions**

In conclusion, the detection and removal of gross errors or outliers are crucial steps in ensuring the accuracy and reliability of data analysis, especially in fields like photogrammetry. This article discussed various methods employed for gross error filtering, ranging from traditional techniques such as residual analysis and statistical tests to more advanced approaches like image matching algorithms and bundle block adjustment.

The focus then shifted to a proposed algorithm that combines cross-validation and statistical tests for gross error detection in photogrammetry. The algorithm aims to identify all gross errors before the adjustment procedure and calculate adjusted unknown parameters with values equivalent to those obtained through the least squares method. By systematically deriving solutions based on different subsets of measurements and employing statistical tests, this method offers a comprehensive approach to identify and filter out gross errors in photogrammetric data.

The detailed exploration of the Random Sample Consensus (RANSAC) method provided valuable insights into its strengths and limitations. While RANSAC is effective in filtering a subset of outliers consistent with a selected voting scheme, it does not guarantee the filtering of all outliers in a dataset due to its stochastic nature and reliance on random sampling.

The combined method utilizing weighted arithmetic mean values introduced in the article addresses the limitations of RANSAC by systematically considering different subsets of measurements, performing statistical tests, and iteratively refining the results. The algorithm's ability to filter out all measurements with gross errors is contingent on having a sufficient number of error-free measurements, ensuring there are enough equations for the adjustment in at least one group among all combinations.

Additionally, an alternative modification to the algorithm was presented, introducing an allowable error threshold to identify gross errors based on residuals. This modification provides flexibility in handling gross errors and offers an alternative approach to ensure the reliability of the photogrammetric results.

In summary, the proposed combined method of gross-error filtering using weighted arithmetic mean values demonstrates a comprehensive and systematic approach to gross error filtering in photogrammetry. Its adaptability to other domains suggests its potential applicability beyond the scope of photogrammetry, explaining the versatility of the algorithm in ensuring the accuracy and reliability of measurements in diverse fields.

The presented methodology for error filtering in photogrammetric measurements has demonstrated its effectiveness in identifying and eliminating gross errors in the dataset. In the numerical example, the process involved the application of an affine transformation to convert pixel coordinates into image coordinates, with parameters calculated through the least-squares adjustment procedure. The coordinates of fiducial marks served as common points for interior orientation, assuming known and error-free $\xi,\eta$ image coordinates obtained during camera calibration.

The error filtering process involved intentionally introducing gross errors to evaluate the robustness of the proposed methodology. The results showed that the standard error of unit weight ($\sigma_0$) and the root mean square error remained within acceptable limits for the error-free dataset. However, deliberate errors led to increased $\sigma_0$, indicating the presence of gross errors.

The proposed methodology for error filtering included forming combinations of three and four points, calculating weighted mean values, and performing a chi-square test to identify potential gross errors. The process successfully identified and excluded erroneous points, and the final residuals demonstrated close agreement with the intentionally introduced errors.

The effectiveness of the proposed methodology was further validated by comparing the calculated parameters before and after error detection. The standard error of unit weight for the gross-error-free solution and the solution after detection remained consistent, indicating the reliability of the error filtering process.

In summary, the presented approach provides a systematic and robust method for identifying and filtering out gross errors in photogrammetric measurements, enhancing the accuracy and reliability of the resulting affine transformation parameters. The successful application of this methodology contributes to the overall improvement of data quality in photogrammetric applications.

**Acknowledgement**

The paper was supported by the Bilateral Chinese-Hungarian Project No. 2019-2.1.11-TÉT-2020-00171, the project title is "Investigation of the characteristics of surface shapes in rural environment based on point clouds and remote sensing data".

**References**

[1] Albertz J., Wiggenhagen M., Taschenbuch zur Photogrammetrie und Fernerkundung: Guide for Photogrammetry and Remote Sensing, Publisher: Wichmann Herbert, ISBN-13: 978-3879073849, 2009, p. 48

[2] Kang Q., Guoman H., Yang S., A Gross Error Elimination Method for Point Cloud Data Based on Kd-tree, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 42 (2018): pp. 719-722

[3] OR K., A step-by-step strategy for gross-error detection, Photogrammetric Engineering and Remote Sensing 50.6 (1984): pp. 713-718

[4] Zaky K. M., Ashraf A. G., Application of Robust Estimation Methods for Detecting and Removing Gross Errors from Close-Range Photogrammetric Data, Am. Acad. Sci. Res. J. Eng. Technol. Sci 55 (2019): pp. 111-120

[5] Jancso T., Gross Error Detection of Control Points with Direct Analytical Method, International Archives of Photogrammetry and Remote Sensing 35 (2004)

[6] Youcai H., Application of Robust Estimation Close-Range Photogrammetry, Photogrammetric Engineering and Remote Sensing 53.2 (1987): pp. 171-175

[7] El-Hakim S. F., A practical study of gross-error detection in bundle adjustment, The Canadian Surveyor 35.4 (1981): pp. 373-386

[8] Derpanis K. G., Overview of the RANSAC Algorithm, Image Rochester NY 4.1 (2010): pp. 2-3

[9] McGlone J. C. (Editor), Mikhail E. M. (Editor), Bethel J. S. (Editor), Mullen R. (Contributor), Manual of Photogrammetry, ASPRS, 5th edition, ISBN 978-1570830716, (2004): p. 72

[10] Moore D. S., Notz W. I, Flinger M. A., The basic practice of statistics, 6th edition, New York, NY: W. H. Freeman and Company (2013)

[11] Gao J., Collins R. T., Hauptmann A. G., Wactlar H. D., Articulated Motion Modeling for Activity Analysis, Third International Conference on Image and Video Retrieval (CIVR'04), Workshop on Articulated and Nonrigid Motion (ANM'04), Dublin City, Ireland, July 21-23 (2004)