Mathematical modelling of an airborne technical vision system development

A.I. Novikov^{1,*}, E.R. Muratov¹, M.B. Nikiforov¹, and D.A. Kolchaev¹

¹Ryazan state radio engineering university named after V.F. Utkin, 59/1, Gagarina str, Ryazan, 390005, Russia

Abstract. An airborne technical vision system of a modern aircraft is, on the one hand, a complex of multispectral sensors and, on the other, airborne computer equipment. The onboard computer solves a wide range of complex digital image processing tasks. The strict requirements are imposed on the computer mathematical software on the image processing time and on the quality of the result displayed on the pilot screen. The software development for such a complex system involves mathematical modelling and use modern information technologies in process of computational algorithms development for task solving. The proposed work discusses ways of solving subtasks of one of the most important subsystems of the airborne complex - the image fusion subsystem and solving the inverse navigation problem. The subsystem is represented by two blocks of tasks: tasks of primary image processing and high-level tasks. The primary processing unit solves the problems of image noise suppression, image enhancement, image fusion and edge detection. In the high-level block of tasks, the tasks of key points detection on real and virtual images, the tasks of a geometric transformation estimation of one image to the plane of another, and the task of a 3D model construction of the underlying surface are solved. The paper presents the results of mathematical modelling in the process of algorithms development for above problems solving. The new results of mathematical modelling of the noise estimation problem are presented in detail. The description of methods and results of other problems solving is less detailed due to article limit but provided with links to the corresponding publications of the authors.

1 Introduction

Depending on the aircraft class and the list of tasks to be solved the aircraft technical vision system is represented by a different set of multispectral sensors. It may consist of color or monochrome television cameras, thermal imaging cameras, radar, lidar, ultrasonic sensors [1]. The specific set of onboard aircraft sensors depends on the type of aircraft and the tasks it solves. The images formed by the sensors have a different physical nature, a different mathematical relationship with the different environment. The image from television cameras is obtained in a familiar to the human eye form. However, images from other

* Corresponding author: novikovanatoly@yandex.ru

sensors require special preprocessing before they can be displayed to the pilot. They should be preprocessed, which may include image enhancement, multi-spectral image fusion. At a higher level, the tasks of fusion of a real television or thermal image (RI) with a virtual image (VI) synthesized from a digital terrain map is solved, the tasks of moving objects detection and tracking, etc. Another of the most important tasks solved as part of the onboard technical vision system is a reconstruction of a 3D model of the underlying surface in the Earth's plane using a stereo images pair sequence. The whole complex of these tasks and special-purpose tasks should be solved in real-time - in 0.04 seconds. It is high strictly requirement and it inevitably affects the choice of a method for solving each subproblem of the complex.

The mathematical concept of one of the subsystems of the aircraft technical vision system, designed to solve one of the most important problems - the problem of combining the radar image and visual image, as well as solving the inverse navigation problem on this basis, was first published in [2, 3]. The mathematical software for the above subsystem of the aircraft technical vision system includes two blocks of tasks to be solved: a block of primary image processing tasks and a block of high-level tasks.

The primary processing unit solves the problems of image noise suppression, image enhancement, image fusion and edge detection. Algorithms for this group of problems should have the lowest possible computational complexity and, at the same time, should provide a high-quality problem solution. To suppress discrete Gaussian noise in images, a modified version of the sigma filter is proposed [4]. Sigma filter for noise suppression is chosen due to its simplicity, a good character of noise reduction among other nonlinear filters and the ability to preserve the boundaries of the brightness difference with the correct choice of the cutoff threshold [5-7]. But there is a problem of correctly determining the cutoff threshold on a sigma filter applying. According to threshold computation by formula $\Delta = m\sigma$, which includes σ - the value of the noise level root-mean-square estimation (RMS), m - the specified parameter (recommended value $m = 2$). The correct threshold value allows to effectively suppress noise while maintaining the boundaries of the brightness difference. Below we consider a fast algorithm of the image noise estimation which gives the estimation of the noise RMS close to the true values.

The Canny detector [8] is rightly considered as one of the best edge detectors. However, it has a rather large computational complexity and creates a large number of short lines in the resulting contour image. In [9], an original algorithm is proposed, which has a computational complexity 2-3 times lower than Canny's algorithm, and at the same time creates a contour image with a minimum number of short lines.

In the high-level tasks block, one of the most important and most difficult is the task of the images fusion - real television or thermal image, on the one hand, and virtual, on the other [1]. The correct fusion of real and virtual images makes it possible to create an image, which significantly increases the crew situational awareness in poor visibility conditions. Combining the real and virtual images allows to simultaneously solve the inverse navigation problem - to correct the erroneous values of the navigation parameters. The aircraft position in space is determined by the six-dimensional vector of navigation parameters $\mathbf{v}_r = (x_r, y_r, h_r, \theta_r, \psi_r, \gamma_r)$. The first three components of the vector are the coordinates of the aircraft, and the last three are the orientation of the aircraft in space (Euler angles). According to these data and a digital map of the area stored in the on-board computer memory, a virtual image is formed. Aircraft sensors give values of vector parameters $\mathbf{v}_r$ with some unknown errors generally. The estimation $\Delta \mathbf{v} = (\Delta x, \Delta y, \Delta h, \Delta \theta, \Delta \psi, \Delta \gamma)$ can be obtained by fusion virtual image with real. It is necessary to determine such a correction to the values of the navigation parameters vector $\hat{\Delta \mathbf{v}}_{opt}$ so that the virtual image generated from the corrected vector $\hat{\mathbf{v}}_{opt} = \mathbf{v}_r + \hat{\Delta \mathbf{v}}_{opt}$ matches as closely as possible to the real image.

The virtual image contains only the contours of objects that constantly present on the Earth's surface. Therefore, before solving the actual problem a real image and virtual image fusion, it is necessary to convert the real to the form of a contour image. Actually images fusion is to determine a correspondence between the points of the real and virtual contour images. There are many ways to solve this problem. Correlation algorithms are among the most efficient in terms of the geometric alignment accuracy of the real and virtual images and the accuracy of solving the inverse problem [10-11]. However, these methods require the construction of the 10^6 digital map views and for each view require the criterion function computation. As a result, the computational complexity of the method turn out to be unacceptably large, excluding its application in a real onboard technical vision system. An alternative way is to use affine or projective transformations for geometric alignment [12-13]. The computational complexity of this group methods is rather low, allowing real-time operation. The only drawback is the complexity of the inverse navigation problem solving within this approach. In [14], a combined method for image fusion and the inverse navigation problem solving was proposed. At the algorithm launching stage and when the alignment fails, a modified version of the correlation algorithm, which has low computational complexity, executes. After that, the tracking mode is turned on, in which the navigation parameters values are estimated in a sliding mode as predicted values from the previous real values.

2 Noise reduction

We will consider the additive image model

$$I_{ij} = U_{ij} + \xi_{ij} \quad i = \overline{1, M}, \quad j = \overline{1, N}, \quad (1)$$

which includes U_{ij} - useful signal, the smooth image component, ξ_{ij} , $\xi \in N(0, \sigma_\xi^2)$ - random component, noise with zero mathematical expectation and unknown variance $D[\xi] = \sigma_\xi^2$. The sigma filter applies to discrete Gaussian noise suppression **A**

$$\mathbf{A}: \hat{I}_i^{smh} = \frac{1}{K} \sum_{s=-K}^K \sum_{t=-K}^K \delta_{st} I_{i+s, j+t}, \quad (2)$$

where

$$K = \sum_{s=-K}^K \sum_{t=-K}^K \delta_{st}, \quad \text{and} \quad \delta_{st} = \begin{cases} 1, & \text{if } |I_{i+s, j+t} - I_{ij}| \leq \Delta, \\ 0, & \text{if } |I_{i+s, j+t} - I_{ij}| > \Delta. \end{cases} \quad (3)$$

The filter is needed to preserve the boundaries of the brightness difference. It is important because the next system task is border detection. In (3) $\Delta = m\sigma$ is the cut-off threshold, σ is a RMS noise estimation, m - the specified parameter. The correct sigma filter operation depends on parameter σ accuracy used to calculate the threshold value.

There are many methods of an image noise level estimation, in particular, median methods [14], block methods [15-18] and methods based on the wavelet transform and Fourier transform [19, 20]. In computational complexity and accuracy of estimation, the method briefly described in [18] is most suitable for use in an onboard computer. The method is quite simple. The image splits into the same size blocks. The block with the minimum variance is selected from the set of blocks. The image in a block is smoothed by the simplest linear filters with kernels (masks) of certain sizes. In mentioned work, it is recommended to use masks of sizes 9x9 and 11x11. Then one smoothed image is

subtracted from another and the sample variance of the remainder is calculated. Based on a sample variance, the noise variance estimation of the original image is calculated.

In [18], the translation rule from a sample variance to the noise variance estimation of the original image is not given. So it had to be restored and tested in a series of model experiments. The algorithm is based on the following reasoning. If $\mathbf{A}$ is the linear smoothing filter acting according to the rule

$$\mathbf{A}I_{ij} = \sum_{s=-k}^k \sum_{t=-k}^k \alpha_{st} I_{i+s, j+t}, \quad (4)$$

and $\mathbf{E}$ is uncorrelated Gaussian noise with zero mathematical expectation, then the variance of the smoothed random component $\eta_{ij} = \mathbf{A}\{\xi\}$ will be found by the formula

$$D[\eta] = \left(\sum_{s=-k}^k \sum_{t=-k}^k \alpha_{st}^2 \right) D[\xi] = \gamma^2 \sigma_\xi^2. \quad (5)$$

From (5) follows that the noise standard deviation estimation $\hat{\sigma}_\xi$ can be found by the formula

$$\hat{\sigma}_\xi = \left(\frac{D[\eta]}{\sum_{s=-k}^k \sum_{t=-k}^k \alpha_{st}^2} \right)^{\frac{1}{2}}. \quad (6)$$

Subtracting the results of one smoothing from another is equivalent to the operator $B = A_{(2k-1) \times (2k-1)} - A_{(2k+1) \times (2k+1)}$. This operator should suppress the smooth low-frequency component of the image. The mask of this operator will have the size $(2k+1) \times (2k+1)$. The mask weights in the first two and last two rows and columns are equal to $\left(-\frac{1}{(2k+1)^2}\right)$. There are $8k$ such coefficients. The remaining $(2k-1)^2$ coefficients of the inner submatrix are the size of $\frac{1}{(2k-1)^2} - \frac{1}{(2k+1)^2} = \frac{8k}{(2k-1)^2(2k+1)^2}$. As a result, in accordance with the formula (5)

$$\begin{aligned} D[B(\xi)] &= \left(\sum_{s=-k}^k \sum_{t=-k}^k \alpha_{st}^2 \right) \cdot D[\xi] \\ &= \left(8k \left(\frac{1}{(2k+1)^2} \right)^2 + (2k-1)^2 \left(\frac{8k}{(2k-1)^2(2k+1)^2} \right)^2 \right) D[\xi], \end{aligned}$$

or

$$D[B(\xi)] = \frac{8k}{(4k^2-1)^2} \cdot D[\xi]. \quad (7)$$

In accordance with formula (6), the estimation of the noise RMS in the original image will be found by the formula

$$\hat{\sigma}_\xi = \frac{(4k^2-1)}{\sqrt{8k}} (\tilde{D}[B(\xi)])^{\frac{1}{2}}. \quad (8)$$

For example, for 5x5 and 7x7 masks formula will have a form $\hat{\sigma}_\xi = \frac{35}{2\sqrt{6}} (\tilde{D}[B(\xi)])^{\frac{1}{2}}$.

Table 1 shows the results of the imposed noise estimation using the described algorithm.

Table 1. The noise RMS estimation based on the method [18].

RMS σ_{ξ} of the imposed noise	A found estimation of noise RMS $\hat{\sigma}_{\xi}$			
	25 × 25	50 × 50	75 × 75	100 × 100
1	1.2	1.7	3.6	7.2
5	5.0	5.1	6.1	8.8
10	9.5	10.0	10.6	12.3
20	18.1	19.9	20.5	21.2
30	28.6	29.7	30.2	30.5

In this experiment, a typical image of the underlying surface was taken as the initial image (Figure 1) and was imposed with the noise of a given intensity $\sigma \in \{1.5, 10, 20, 30\}$. Smoothing was done by masks of 5×5 и $7 \times 7 (k = 3)$ size. The sizes of an original image divided blocks were varied from 25×25 to 100×100 with a step of 25.

Fig. 1. The underlying surface initial image.

We can note good results of noise RMS estimation when dividing the original image into 25x25 and 50x50 blocks for all noise intensity values except $\sigma = 1$.

The computational complexity of the algorithm can be reduced by applying vector masks for smoothing with the weight coefficients calculated by the formula

$$\alpha_j = \frac{3(3k^2 + 3k - 1 - 5j^2)}{(2k - 1)(2k + 1)(2k + 3)}, \quad j = 0, \pm 1, \dots, \pm k.$$

(9)

Table 2 shows the results of the imposed noise estimation for the same image and for the same block sizes and imposed noise intensities set. To images smoothing, a pair of filters with vector masks of lengths 5 and 7 were used

$$\frac{1}{35}(-3, 12, 17, 12, -3), \frac{1}{21}(-2, 3, 6, 7, 6, 3, -2).$$

Table 2. The noise RMS estimation based on the proposed method.

RMS of the imposed noise σ_{ξ}	A found estimation of noise RMS $\hat{\sigma}_{\xi}$			
	25 × 25	50 × 50	75 × 75	100 × 100
1	1.0	1.1	1.1	1.2
5	5.1	5.0	5.0	5.1
10	9.4	9.8	10.1	10.1
20	18.6	20.1	20.0	20.1
30	29.1	29.7	30.6	30.6

Table 1 and Table 2 comparison allows to make two main conclusions. First, the proposed method, in contrast to the method [18], gives stable RMS estimation regardless of the image block sizes and for all noise intensities. Secondly, an image fragment with a $2k + 1$ vector mask convolution is performed faster than with a matrix mask. Also, the estimation does not undergo significant changes when only a part of the block rows is used. For Table 2 case smoothing was implemented only in every 4th row of the block.

3 Edges detection

Although the brightness difference boundaries detection is an auxiliary task, the accuracy of the real image and virtual image alignment depends on the contour image quality obtained as a result of brightness edges detection. As was mentioned in the introduction, Canny's method is one of the best edge detectors. However, taking into account the requirements for the onboard systems, it has a rather high computational complexity and at the same time create a contour image with a large number of short lines. Such Canny's method feature was noted in the book [22].

The gradient-based edge detector proposed in [9] differs from the Canny method

- images pre-smoothing is absent;
- a vector mask applying instead of Sobel or Prewitt matrix as a mask to estimate partial derivatives;
- applying a new strategy of a contour image constructing.

Vectorial mask

$$\mathbf{g} = (-k, -k + 1, \dots, -1, 0, 1, \dots, k - 1, k)$$

is applied to an image row or column of $2k + 1$ length. For example, the estimation of the partial derivative in the $i - th$ line along the j coordinate is found as a convolution

$$\hat{I}_{ij} = \mathbf{I} * \mathbf{g}, \quad \text{where } \mathbf{I} = (I_{i,j-k}, I_{i,j-k+1}, \dots, I_{i,j-1}, I_{ij}, I_{i,j+1}, \dots, I_{i,j+k-1}, I_{i,j+k}).$$

Convolution of image fragments in a row or in a column with a $\mathbf{g}$ mask provides a linear model $\hat{I} = a + bj$, $j = \overline{-k, k}$ coefficient b estimation that is optimal in the sense of the least-squares method, provides an optimal partial derivative estimation with respect to the corresponding coordinate. And it is smoothed estimation. Therefore, there is no need for additional image smoothing in low-intensity Gaussian noise conditions ($\sigma < 8$). The mask length $2k + 1$ affects on a partial derivative estimation under noise conditions accuracy. The higher the image random component intensity, the higher the parameter value should be. However, a large mask length can lead to the duplication of the selected line. Therefore, $l = 2k + 1$ it is advisable to choose the length of the sliding window equal to 5 for images with a low noise level and equal to 7 ... 9 - for noisy images.

The MATLAB edge() function has a trech parameter for the edge detector with the default value 0.7. It provides a "strong" line guaranteed detection. Increased value of the trech parameter leads to an exceeding joining a "weak" lines to the "strong" lines. As a result, a large number of short lines are formed in the Canny detector, it makes difficult the further contour image usage. In the proposed method, the automatic threshold selection provides the minimum number of short uninformative contour lines. It's achieved by choosing the lower threshold Δ_1 from the histogram, according to which the value of the upper threshold Δ_2 is then calculated according to certain rules [9].

Figure 2 shows the contour images obtained by the proposed method in comparison with the Canny method.

Fig. 2. Contour image of edges obtained by the proposed method – (a), contour image obtained by Canny method – (b).

The contour images are derived from the image shown in Figure 1. The advantages of the contour image obtained by the proposed method are obvious. The water bodies, roads and forest clearings contours on it are much easier to detect than on a contour image obtained by the Canny method. It is important for the contour image automatic processing.

4 2D images fusion

The search of the most suitable mathematical methods for solving the problem a real image and the virtual image fusion was carried out in two main directions. Firstly, it is methods based on affine and projective transformations of one image to the plane of another. And, secondly, correlation-based methods of image align. To align contour images by geometric methods, it is necessary to select the residual number of corresponding pairs of points (key points) on them. This subtask was solved as follows. Closed and open contours of sufficiently large length, which can be the boundaries of objects of constant presence on the underlying surface of the Earth (roads, water bodies, runways, large buildings, etc.) were distinguished on the contour image. Using the simplest but effective algorithm [2] the contours were approximated by polygons with a minimum number of vertices and edges. The vertices of polygons on the real image and virtual image were assigned as key points.

Here is a brief description of the algorithm for approximating the contours by polygons. A movable coordinate system Oxy is placed sequentially on each contour pixel. For each m coordinate of the contour of the pixel, from one and the other sides of the origin, the values of the coefficients k_1, k_2 of the straight lines $y = k_1x$ and $y = k_2x$ are calculated by the formula $k = \frac{\sum_{i=1}^m x_i y_i}{\sum_{i=1}^m x_i^2}$. Vectors $q_1 = (\pm 1, k_1)$ and $q_2 = (\pm 1, k_2)$ are formed. The signs "+" and "-" of vectors are selected depending on the quarter in which the end of the vector is located. Then the cosines of the angles between the vectors q_1 and q_2 are calculated by the well-known formula. For each contour, local maximums are determined on the set of cosines of the corners. Local maximum points are accepted as the vertices of the approximating polygon.

The parameter m is on the task input. It can be used to control the approximation order. The contour can contain several thousand pixels. For small parameter value m ($m = 3 \dots 7$), the number of vertices of the approximating polygon will be extremely large. For the correct alignment of two corresponding contours of the real image and virtual image 20 ... 30 such points are enough. It can be achieved by increasing the parameter. Figure 3a shows runway contour, and Figure 3b shows the approximation by the polygon. The resulting polygon has 4 vertices, which are enough for the contour description.

Fig. 3. Runway contour: a - binary image; b - vectorial description.

It was several mathematical models processed to find the best method of geometric transformation for images fusion and the actual images alignment based on considered models. The possibility of complex contour analysis for the alignment was also tested [3, 23]. It was proved that the best alignment results are achieved by the virtual image projective transformation to the plane of real image based on the homography matrix.

For the homogeneous coordinates $X^s = (x^s, y^s, 1)^T$ of a virtual image points, the homography matrix for coordinates transformation to the plain of a real image has the following form

$$\begin{pmatrix} \lambda x_i' \\ \lambda y_i' \\ \lambda \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix} \cdot \begin{pmatrix} x_i^s \\ y_i^s \\ 1 \end{pmatrix}, \quad (10)$$

or $X' = HX^s$ in matrix form. Usually, the homography matrix is unknown. But in the presence of a sufficient number of key points pairs $\{(X_i^s, X_i')\}$, relation (10) can be considered as a system of linear algebraic equations. Each key points pair $M_i^s(x_i^s; y_i^s)$ and $M_i'(x_i'; y_i')$ generates two linear equations

$$\begin{cases} x_i' y_i^s h_{31} + x_i' y_i^s h_{32} + x_i' = x_i^s h_{11} + y_i^s h_{12} + h_{13}, \\ y_i' x_i^s h_{31} + y_i' y_i^s h_{32} + y_i' = x_i^s h_{21} + y_i^s h_{22} + h_{23}. \end{cases} \quad (11)$$

There are 8 elements of the arithmetic vector $h = (h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32})$ are unknown in the system of equations (11), which means that to find these elements, at least 4 pairs of key points are needed.

In the general case, as a result of solving the problem key points matching, their number is significantly greater. There may be pairs with incorrect matches among all matches. The problem of 4-6 best pairs selecting is traditionally solved by RANSAC algorithm [24, 25]. It was shown in [13] that the RANSAC algorithm is very sensitive to correspondences mismatching errors between key points. Wrong correspondences can lead to the wrong homography transform estimation. At the same time, this work proposes another approach based on the entire set of key points direct involving. The system of linear algebraic equations (11), converted to the standard notation, has the following form $Ah = B$. With the number of key points pairs $k \gg 4$, the system of equations (11) consists of $2k$

equations. This means that this system of equations is overdetermined and, therefore, may turn out to be inconsistent in the classical sense. A pseudo-solution to such a system can be found from the condition $\|Ah - B\|^2 \rightarrow \min_h$. So we get a normal equation, as result: $(A^T A)h = A^T B$.

A detailed description of the virtual image and real image fusion by projective transformation can be found in [3, 13].

The RI and VI fusion based on the projective transformation of the VI to the plane of the RI has good quality and is realizable in real-time. But the inverse navigation problem remains unsolved. The problem solution was found within the joint approach. It is represented by two blocks providing search and tracking modes operation, respectively. In the search mode, the alignment is realised by a modified version of the correlation algorithm, which has low computational complexity. The search mode applied on a starting stage of the algorithm and in case of a wrong alignment when at least one of the six navigation parameters value leaves the predicted value. In the search mode, the alignment is implemented according to the predicted values of the navigation vector parameters $\mathbf{v}$ [14].

In the search mode, the optimal of the navigation vector parameters estimation $\hat{\mathbf{v}}_{opt}$ is found using a modified version of the correlation algorithm. To reduce the implemented algorithm computational costs, the virtual image view is being extended (Figure 4).

Fig. 4. The virtual image extended view.

The field of view expands vertically and horizontally by increasing the values of the pitch θ and yaw ψ angles obtained from the navigation system sensors by $2\Delta\theta$ and $2\Delta\psi$, respectively. Such an approach makes it possible to drop the brut search for the pitch and yaw angles while optimal alignment finding between the RI and VI. Due to this, the algorithm time execution is reduced. The next modification of the correlation algorithm is to introduce a two-stage scanning of image coordinates. On the first stage, coordinates are scanned in latitude, longitude and altitude. At the second stage, the coordinates are scanned with a halve step around the point founded at the first stage. Such an approach made it possible to reduce the algorithm time execution by more than 3 times in comparison with the scanning of all fine grid nodes.

In the tracking mode, which is the main mode, the synthesis of a virtual image from a digital terrain map is carried out according to the predicted values of each navigation parameter. The prediction is carried out according to linear models, which corresponds to the flight dynamics without distinct changes in at least one of the 6 navigation parameters.

When the values of at least one navigation parameter exceed limits, the algorithm switches to the search mode [14].

5 3D models of the underlying surface construction

The proposed method is designed to generate 3D images of the underlying surface from a sequence of this surface 2D images pairs, obtained from a stereo pair located on the aircraft board. The algorithm consists of several blocks, each of solves its own independent task:

- a depth maps construction block based on a sequence of the underlying surface 2D images pairs from a stereo pair;
- a point clouds construction block based on depth maps and camera parameters matrix;
- a key points detection block on $i - th$ and $i + 1 - th$ frame based on FAST algorithm [26];
- block of key points in a pair of point clouds matching based on a sparse optical flow Lucas-Kanade algorithm [27];
- block of $i - th$ and $i + 1 - th$ point clouds alignment and a rotation matrix R estimation.

On the first step of the algorithm, depth maps for the $i - th$ and $i + 1 - th$ frames are computed using a stereo pair. The depth map is built on the displacement basis between the corresponding pixels of the left and right images.

On the second step, point clouds P_i and Q_{i+1} are also formed for the $i - th$ and $i + 1 - th$ frames based on the corresponded depth maps in three-dimensional space.

On the third step, the multiple pairs of key points are detected. Key points are detected on the $i - th$ image by the FAST detector. For the detected key points, using the optical flow, the corresponding points on the $i + 1 - th$ image are matched.

The next 8 steps of the algorithm coincide to the ICP algorithm logic [28]. At these steps, the centring of neighbouring point clouds is performed, the covariance matrix is calculated, its singular value decomposition is computed and the rotation matrix and the translation vector are calculated, so the clouds are aligned. A detailed description of the algorithm is given in [29].

Figure 5 shows the result of point clouds stitching obtained from 150 stereo images pairs processing according to the described algorithm.

Fig. 5. 3D model of the underlying surface.

6 Conclusion

The mathematical approach of image alignment for an aircraft onboard vision system is proposed. The results of mathematical modelling of an image noise estimation are presented. The mathematical models of other problems are observed in general and literature with its detailed implementation is given.

References

1. 2016 *Image processing in aircraft vision systems* (Moscow: Fizmatlit) p 240
2. Novikov A I, Sablina V A, Nikiforov M B and Loginov A A 2015 *Pattern Recognition and Image Analysis* **25**(1) 73
3. Novikov A I, Sablina V A, Efimov A I and Nikiforov M B 2016 *Journal Coupled Systems and Multiscale Dynamics* **4**(4) 251
4. Ustukov D I, Pronkin A V, Muratov E R and Novikov A I 2020 *9th Mediterranean Conference on Embedded Computing (MECO-2020)* 534
5. Rodionova N V and Sinilo V P 1993 *Avtometriya, Siberian Branch of the Russian Academy of Sciences* **5** 96
6. Lee J 1983 *Computer Vision, Graphics and Image Process* **24**(2) 255
7. Ponomarenko N, Lukin V, Egiazarian K, Astola J, Vozel B, and Chehdi K 2006 *Int. Conf. on Advanced Concepts for Intelligent Vision Systems ACIVS 2006* 46
8. Canny J 1986 *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**(6)
9. Novikov A I and Pronkin A V 2019 *8th Mediterranean Conference on Embedded Computing (MECO)* 364
10. Zlobin V K, Kolesnikov A N and Kostrov B V 2011 *Bulletin of the Russian State Technical University* **3**(37) 12
11. Chochia P A 2009 *Information processes* **9**(3) 117
12. Novikov A I, Loginov A A, Muratov E R and Nikiforov M B 2015 *Dynamics of complex systems* **9**(1) 33
13. Efimov A I and Novikov A I 2016 *Computer Optics* **40**(2) 258
14. Novikov A I, Loginov A A and Kolchaev D A 2017 *Digital signal processing* **2** 53
15. Donoho D L 1995 *IEEE Transactions Theory* **41**(3) 613
16. Olsen S I 1993 *8th Scandinavian Conference on Image Analysis*
17. Kalinkina D A 2005 *Int. Conference of Students and Postgraduates in Fundamental Sciences (Lomonosov 2005)*
18. Kovalevsky V *Effective filtering and boundary detection*
19. Ghasal M, Amer A and Ghayeb A 2006 *Proc. IEEE Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)* 845
20. Lapshenkov E M 2012 *Computer Optics* **36** 439
21. Novikov A I 2015 *Pattern Recognition and Image Analysis* **25**(2) 230
22. Gonzalez R and Woods R 2012 *Digital image processing* (Moscow: Technosphere) p 1104
23. Furman Y, Krevetskiy A, Peredreev A, Rozhentsov A, Khafazov R, Egoshina L and

- Leukhin A 2003 *Introduction to contour analysis; applications for image and signal processing* (Moscow: Fizmatlit) p 592
24. Chen C 1999 *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21(11)** 1229
 25. Hast A, Nysjö J and A. Marchett 2013 *Journal of WSCG* **21(1)** 21
 26. Rosten E, Drummond T 2006 *9th Conference on Computer Vision (ECCV 2006)* **1** 430
 27. Lucas B and Kanade T 1981 *Proc. of the 7th international joint conference on Artificial intelligence* **2** 674
 28. Besl P and Neil D 1992 *IEEE Transactions on Pattern Analysis and Machine intelligence* **14(2)** 239
 29. Novikov A I, Efimov A I and Kolchaev D A 2018 *CEUR Workshop Proceedings* **2210** 56