PHOTOGRAMMETRIC MODEL BASED METHOD OF AUTOMATIC ORIENTATION OF SPACE CARGO SHIP RELATIVE TO THE INTERNATIONAL SPACE STATION

Yuri B. Blokhinov *, Alexey S. Chernyavskiy, Sergei Y. Zheltov

State Research Institute of Aviation Systems (FGUP GosNIIAS), Moscow, Russia
(yury.blokhinov, achern, zhl)@gosniias.ru

Commission III, WG III/1

KEY WORDS: Digital, Photogrammetry, Modelling, Orientation, Automation, International, Space, Real-time

ABSTRACT:

The technical problem of creating the new Russian version of an automatic Space Cargo Ship (SCS) for the International Space Station (ISS) is inseparably connected to the development of a digital video system for automatically measuring the SCS position relative to ISS in the process of spacecraft docking. This paper presents a method for estimating the orientation elements based on the use of a highly detailed digital model of the ISS. The input data are digital frames from a calibrated video system and the initial values of orientation elements, these can be estimated from navigation devices or by fast-and-rough viewpoint-dependent algorithm. Then orientation elements should be defined precisely by means of algorithmic processing. The main idea is to solve the exterior orientation problem mainly on the basis of contour information of the frame image of ISS instead of ground control points. A detailed digital model is used for generating raster templates of ISS nodes; the templates are used to detect and locate the nodes on the target image with the required accuracy. The process is performed for every frame, the resulting parameters are considered to be the orientation elements. The Kalman filter is used for statistical support of the estimation process and real time pose tracking. Finally, the modeling results presented show that the proposed method can be regarded as one means to ensure the algorithmic support of automatic space ships docking.

1. INTRODUCTION

The primary goal of the projected television measuring system of the planned Space Cargo Ship (SCS) for the International Space Station (ISS) is to carry out the monitoring of mutual position of SCS and ISS in the process of spacecraft docking approach. Continuous video tracking should be done in automatic mode and should perform the following measurements: angles of deviation SCS from the line of sight, distance and radial speed of SCS, and angles of rotation between the active and passive ships.

In order to perform these calculations at each stage of tracking on the basis of the analysis of consecutive images of ISS, it is necessary to carry out a preliminary processing and analysis of these images for the purpose of detection and recognition of ISS or its nodes and their subsequent binding to the coordinate system of the ISS. Starting at a distance of 500 m the task of exterior orientation is solved for every single video frame. The detection and recognition of details of the ISS and the use of their characteristic features is made using computer vision methods. The coordinates of reference points the ISS coordinate system are measured by using a highly detailed three-dimensional model of the ISS. The 3D coordinates are then registered to the corresponding positions of these points on the image.

The main property of the problem under consideration is that the process of video-assisted docking is performed on a wide range of distances starting at 500 m and ending at 0 m. This leads to high object appearance changes in the course of docking (Figure 1). Because of this property, the detection and tracking task must be solved by a set of different algorithms depending on the current distance between SCS and ISS and this is the reason to use very detailed digital model of the ISS including 16000 vertices. Formulas of exterior orientation which form the basis for calculation of orientation elements and errors of their estimations remain valid in all cases.

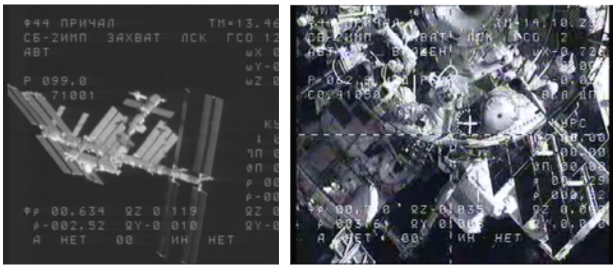

Figure 1. The ISS viewed from a distance of 600 m and 30 m

Another important requirement that defines and restricts the choice of algorithms and the sequence in which various algorithms must be used is the real-time constraint. Measurement should also be highly accurate. In order to meet all these prerequisites, the contour-based algorithm was used. Its essential characteristics are investigated in this work. In order to increase the reliability of orientation parameters estimation and ensure pose tracking in real time, the statistical treatment of the resulting time series were also used; the ones allowing to obtain statistically validated forecasts.

[*] Corresponding author.

A detailed description of the developed algorithms is presented in Section 2 of this paper. Results obtained on synthetic and real dataset are shown in Section 3.

2. ITERATIVE POSE ESTIMATION ALGORITHM BASED ON CONTOUR TEMPLATES

2.1 Distance between image and projected contours

Assuming approximate initial pose estimate is available, the model is projected onto the image plane, and a contour representation of the projection if obtained. This contour, which is one pixel thick, is called contour template. The pose estimation problem consists in finding pose parameters (camera position specified by three rotation angles and three coordinates) which minimize the difference between the contour template and the image. In this work we present an iterative algorithm for pose estimation based on distances from the model's projected contours and image edges.

The first step is to obtain the projected contour of the model. Since the 3D model of the object is known, this task is performed using *z*-buffering and rasterization algorithms. Next, in each point belonging to the projected contour we compute the distance from the point to the nearest edge on the image. The search for the nearest edge is performed along the line parallel to the vector normal at the current contour point. Only edges which are parallel to the projected contour are considered. To achieve this, the image is convolved with a bank of filters which accentuate edges of particular direction. In this work we quantize the search directions into four bins corresponding to 0°, 45°, 90° and 135°. The video frame is thus convolved with four 7x7 filters, with kernels equal to the Gaussian derivative along the *x*-coordinate rotated in the direction θ (Geusebroek, J.-M., 2003). Various parts of ISS are accentuated by different directional filters.

The contour-based pose estimation algorithm originates from the works of Lowe (Lowe, D. G., 1991) and (Vacchetti, L., et al., 2004). For each point (x_0, y_0) located on the contour template we inspect a set of image points, $X_j(x_0, y_0; \theta) = (x_0 + j\cos\theta, y_0 - j\sin\theta)$, $j \in [-R, R]$. The points are located on both sides of the contour, on the straight line L_θ which is perpendicular to the contour in the current point. The radius of the search neighborhood along the line is denoted by *R*. In every point $X_j(x_0, y_0; \theta)$ the absolute value of the previously computed directional derivative along direction θ is analyzed. Points where these values exceed a threshold are stored into a list. For every point (x_j, y_j) from the list, we compute the signed distance from (x_j, y_j) to the line specified by the normal vector with coordinates $(\cos\theta, -\sin\theta)$ passing through the point (x_0, y_0) (the line is tangential to the projected contour). We use the screen coordinate system in which the y-axis points downwards and (0, 0) corresponds to the upper left corner of the image. The search for the nearest edge is illustrated in Figure 2; the projected contour is drawn by a dashed line. The signed distance is computed as

$$d_j = (x_j - x_0)\cos\theta - (y_j - y_0)\sin\theta \,. \tag{1}$$

The distance is positive if the points (x_0, y_0) and (0,0) are located in the same half-plane with respect to the straight line; otherwise it is negative.

The misfit function is the sum of all squared reprojection errors, i.e. distances between all projected contour points to the closest corresponding image edges:

$$F(\mathbf{m}) = \|\mathbf{d}(\mathbf{m})\|^2 \,. \tag{2}$$

The vector $\mathbf{m} = \{\omega, \varphi, \kappa, T_x, T_y, T_z\}$ contains the pose parameters, three rotation angles and three camera coordinates that are to be found.

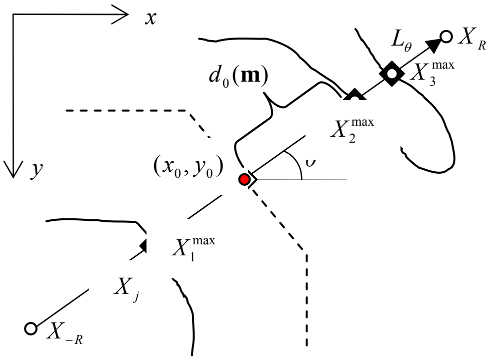

Figure 2. Search for the closest edge

2.2 Expressions for contour-based distance and its derivatives

Before starting the process of iterative pose estimation using the projected contour template, it is necessary to find the three-dimensional coordinates of points whose projections belong to the contour. The "inverse projections" of the points lying on the contour are found by tracing the imaginary rays coming from the camera through the image plane and finding their intersection with the 3D model, assuming that pose parameters are known. In this way, every contour point is represented by a set of its 2D pixel coordinates (x_i, y_i) and 3D coordinates (X_i, Y_i, Z_i). Let us assume that the camera (or observer) coordinates are given by $\mathbf{T} = (T_x, T_y, T_z)^T$. The angles of rotation around the coordinate axes are ω, φ and κ. The pixel coordinates (x, y) of a point with homogeneous coordinates $\mathbf{x} = (X, Y, Z, 1)^T$ can be found according to

$$\begin{aligned} x &= u / t \\ y &= v / t \end{aligned} \tag{3}$$

$$[u, v, w, t]^T = \mathbf{PVx} \,, \tag{4}$$

$$\mathbf{V} = \begin{bmatrix} \mathbf{R}^T & & & -\mathbf{R}^T\mathbf{T} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5}$$

$$\mathbf{P} = \begin{bmatrix} f & 0 & -W/2 & 0 \\ 0 & -f & -H/2 & 0 \\ 0 & 0 & z_f/(z_f - z_n) & -z_f z_n/(z_f - z_n) \\ 0 & 0 & -1 & 0 \end{bmatrix} \tag{6}$$

where
V = view matrix
P = projection matrix
f = focal distance in pixels
W, *H* = width and height of the image
z_n, z_f = coordinates of the near and far clipping planes.

The 3x3 rotation matrix **R** in (5) is fully specified by three angles, and depends on the system of angles used. Since the screen coordinates of any 3D point can be computed analytically using formulas (3)-(6), it becomes possible to obtain the expressions for partial derivatives of pixel coordinates with respect to pose parameters:

$$\begin{aligned} \frac{\partial x}{\partial m_i} &= \frac{1}{t}\left(\frac{\partial u}{\partial m_i} - \frac{u}{t}\frac{\partial t}{\partial m_i} \right) \\ \frac{\partial y}{\partial m_i} &= \frac{1}{t}\left(\frac{\partial v}{\partial m_i} - \frac{v}{t}\frac{\partial t}{\partial m_i} \right) \end{aligned} \tag{7}$$

where m_i is the *i*-th component of **m**. Partial derivatives of the signed distance are given by

$$\frac{\partial d}{\partial m_i} = -\frac{\partial x}{\partial m_i}\cos\theta + \frac{\partial y}{\partial m_i}\sin\theta \, . \tag{8}$$

2.3 Iterative minimization of contour-based distance

In order to find the six unknown parameters that minimize the function (2) we use Levenberg-Marquardt method. Assuming that an initial guess $\mathbf{m}^0$ exists, on every iteration the residual $\mathbf{r} = \mathbf{d}(\mathbf{m}^t)$ and the Jacobian matrix $\mathbf{J}(\mathbf{m}^t)$ are computed and used to update the estimate according to

$$\delta_{\mathbf{m}} = -\left(\mathbf{J}^T\mathbf{J} + \lambda \mathrm{diag}(\mathbf{J}^T\mathbf{J})\right)^{-1}\mathbf{J}^T\mathbf{r} \, , \tag{9}$$

$$\mathbf{m}^{t+1} = \mathbf{m}^t + \delta_{\mathbf{m}} \tag{10}$$

The scalar parameter $\lambda > 0$ is changed dynamically over the course of the iterations. The second term in (9) allows taking into account the non-uniform curvature of the misfit function. At the first iteration $F(\mathbf{m}^0)$, $\mathbf{r}(\mathbf{m}^0)$ and $\mathbf{J}(\mathbf{m}^0)$ are computed. Using the initial value of $\lambda = 10^{-3}$ a trial solution $\tilde{\mathbf{m}}$ is found according to (9) and (10). If at iteration *i* the condition $F(\tilde{\mathbf{m}}) < F(\mathbf{m}^{i-1})$ is true, then $\mathbf{m}^i = \tilde{\mathbf{m}}$, $\lambda := \lambda/10$. In this case the Jacobian is recomputed, along with the new value of the residual $\mathbf{r}(\mathbf{m}^i)$, and a new iteration starts. Otherwise, $\lambda := 10\lambda$; the computation of the trial solution $\tilde{\mathbf{m}}$ is performed until either the misfit function decreases, or any other termination condition is satisfied.

False edges are frequent in images. Such edges may have a higher absolute value of the directional derivative than true image edges. It is therefore necessary for each inspected contour point (x_i, y_i) to take into account *all* the local maxima of the image derivative along direction θ_i that are within the *R*-neighborhood of it. This can be done by using a modified misfit function:

$$F(\mathbf{m}) = \sum_{i=1}^{N_C} \rho(d_i) \tag{11}$$

where N_C is the number of pixels lying on the projected contour, and the function $\rho(x)$ penalizes outliers, i.e. points located at a distance that exceeds some threshold (Vacchetti, L., 2004). In our work we used the Tukey function

$$\rho_{\mathrm{Tuk}}(x) = \begin{cases} \dfrac{c^2}{6}\left[1 - \left(1 - \left(\dfrac{x}{c}\right)^2\right)^3\right], & \text{if } |x| \le c \\ c^2/6 & \text{otherwise.} \end{cases} \tag{12}$$

The minimization of (11) may be reinterpreted as a least squares minimization with reweighting (Lepetit, V., Fua, P., 2005). Weights are chosen according to the formula

$$w_i = \frac{\sqrt{\rho(d_i)}}{d_i} \, . \tag{13}$$

The misfit function then becomes

$$F(\mathbf{m}) = \|\mathbf{Wd}\|^2 \tag{14}$$

where $\mathbf{W} = \mathrm{diag}(w)$ is a matrix of weights, $w = \{w_1, \ldots, w_{N_C}\}$. The Levenberg-Marquardt step (9) is rewritten as

$$\delta_{\mathbf{m}} = -\left(\mathbf{J}^T\mathbf{W}^2\mathbf{J} + \lambda \mathrm{diag}(\mathbf{J}^T\mathbf{W}^2\mathbf{J})\right)^{-1}\mathbf{J}^T\mathbf{W}^T\mathbf{r} \, . \tag{15}$$

2.4 Contour-based algorithm and Kalman filter

Contour-based algorithm for estimation of exterior orientation elements of ISS based on appropriate contour templates is very time-consuming. It doesn't manage to process each video frame in real time. Therefore parallel algorithm of processing and tracking on the basis of widely known Kalman filter was realized. The algorithm under consideration uses the results of contour based one as input data.

1. At receipt of the frame with number n on the basis of the previous data of estimation algorithm Kalman filter is initialized.
2. Kalman filter predicts data at the moment of n+1.
3. These data move to contour algorithm as new position estimation.
4. As contour algorithm works the values for n+1, n+2, …, n+k are defined by Kalman filter as a forecast in the mode of absence new input data.
5. At the moment of time of n+k new data are arrived from contour algorithm from the moment number n.
6. According to these data Kalman filter corrects values of parameters for time n, and then data for n+k in are recalculated in the mode of absence new input data.
7. Step 2 is repeated.

This technique including fusion of contour-based algorithm and Kalman filter was tested on modelling and real data/

3. NUMERICAL EXPERIMENTS

In our situation the complexity of testing of the developed method was connected with that for real videos of spacecraft docking there are no exact measurements of trajectory coordinates and angles. For testing of algorithm the modeling data were used. These data include the trajectory, simulating real trajectory of spacecraft docking and synthesized by specialized software modeling high realistic video film corresponding to given trajectory. Known values of coordinates and angles allow to make an assessment of accuracy of algorithm at this stage. Then the algorithm was tested on a real video with only visual quality assessment. The 3D model of the ISS used for the experiments presented in this work consisted of about 16000 vertices.

The algorithm for iterative model-based pose estimation has been used to track the International Space Station by first analyzing a synthetic video sequence depicting the approach procedure. Although, in principle, the ISS contains moving parts such as solar panels, we assume that its exact configuration is known for each particular docking mission. The initial estimate of SCS position (three angles and three coordinates) relative to the ISS is known also. The algorithm was initialized by the approximate pose parameters. In Figure 3 we demonstrate the initial estimate superimposed over the video image, and in Figure 4 the final solution obtained by running the iterative contour-based algorithm is shown. Visual quality assessment from Figure 5 shows that contour visually matches the real image edges. The tracking of the model was performed over a period of several minutes during which the SCS travelled along a spiral-like trajectory for a total displacement of about 600 meters. Figures 6–8 demonstrate the orientation angles corresponding to the modeled trajectory (solid lines) as well as the orientation angles recovered by our contour-based tracking algorithm (dashed lines). The mean squared errors (MSE) for angles under estimation are:

$$\mathrm{MSE}(\omega) = 0.1627,$$
$$\mathrm{MSE}(\varphi) = 0.6571,$$
$$\mathrm{MSE}(\kappa) = 0.4397,$$

all values are satisfactory.

The distance threshold for the search of closest edges was fixed at R = 50, the Tukey function parameter was set to c = 10. These parameters can be changed depending on the translation value between two consecutive video frames and the standard deviation of distances from the projected contour to the nearest edges. Regions of the image were treated as potential edges if the absolute value of the directional derivative exceeded a threshold of 5. The threshold value depends on the image contrast, and in principle it should be updated dynamically depending on photometric conditions.

Five hundred random uniformly distributed contour points were used for computing the misfit function on each iteration. Most computation time was spent on computing the projected contour and calculation of the Jacobian matrix. These operations involve simple 3x3 or 4x4 matrix-vector multiplications and are easily parallelizable. It was observed that the Levenberg-Marquardt method usually converges in about eight iterations, and the misfit function is computed, on average, 15-17 times per video frame.

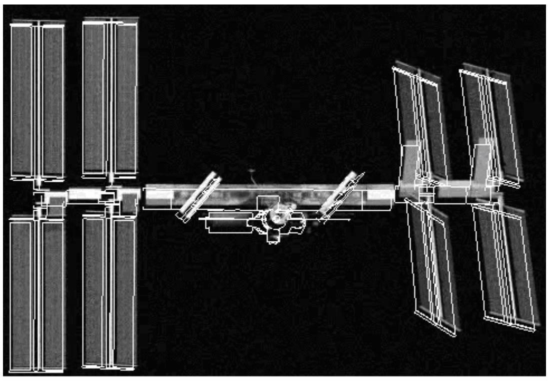

Figure 3. Contour generated by the initial orientation estimate

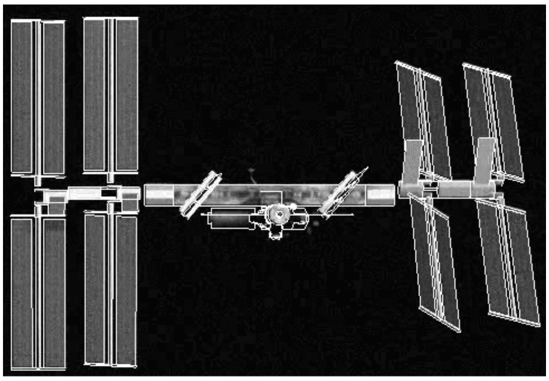

Figure 4. Model contour fitted to image using found pose parameters

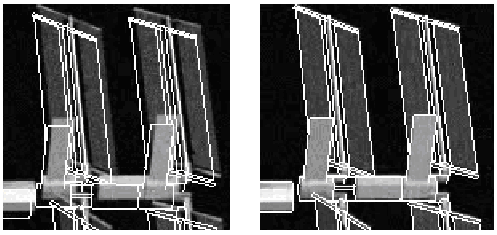

Figure 5. Zoomed-in fragments of Figure 4 and Figure 5

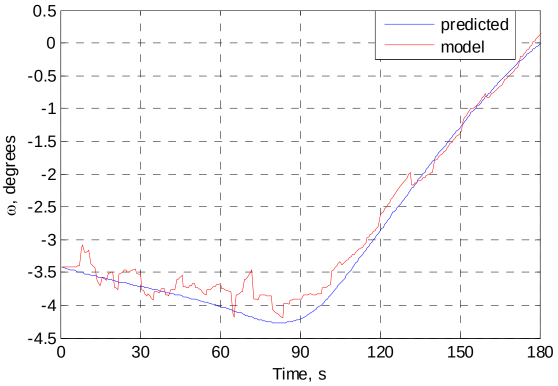

Figure 6. Recovered orientation angle ω

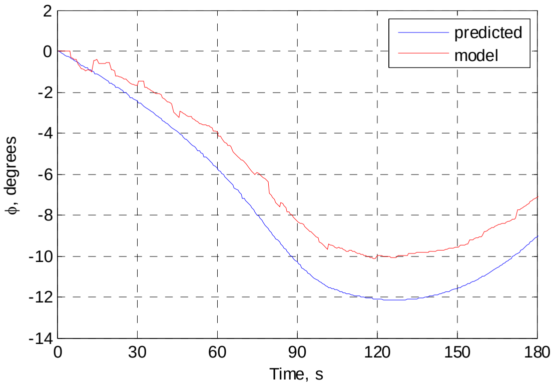

Figure 7. Recovered orientation angle φ

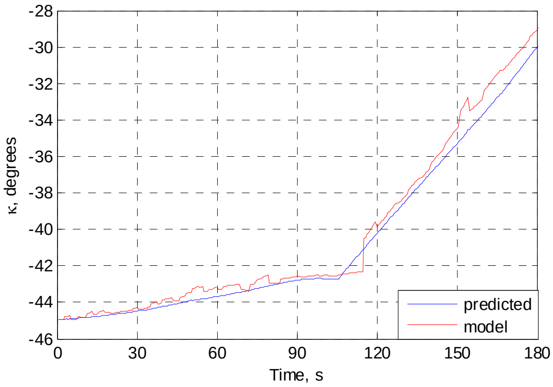

Figure 8. Recovered orientation angle κ

The algorithm allows tracking objects given their 3D models. The convergence of the iterative algorithm depends on the availability of a good initial estimate. The procedure of recovering orientation parameters is ambiguous since there are many configurations that give rise to approximately the same projected contours. It may happen that the algorithm does not converge to the right optimum; this is due to the presence of false edges such as structures that are parallel to the true edges. Contours of the docking module should allow resolving the ambiguities in the determination of angles – however, since the docking module is small relative to the ISS a whole, the contribution of its contours in the whole misfit function is relatively modes. False edges often arise from shadows, especially long and contrasting shadows are cast by solar panels. In some flybys, additional distracting contours appear due to the texture of the Earth's atmosphere.

Then the algorithm was tested on a real video with only visual quality assessment. The real dataset contained telemetry information which was overprinted on the video sequence. Pixels belonging to textual information have been ignored by means of a binary mask which was created offline. Along with high noise level, flare and unfavorable lighting conditions it makes the problem of ISS image matching very difficult (Figure 9). The total fraction of successful image matching for real video frames is 20-30% less then in case of modelling video. An example of successful fitting of the contour template to the real video frame is shown in Figures 10 and 11.

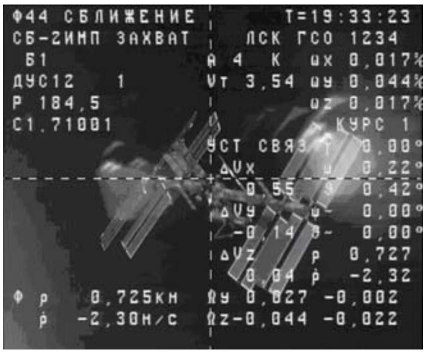

Figure 9. Frame with frare

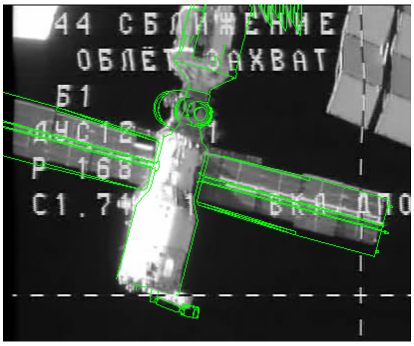

Figure 10. Initial pose estimate

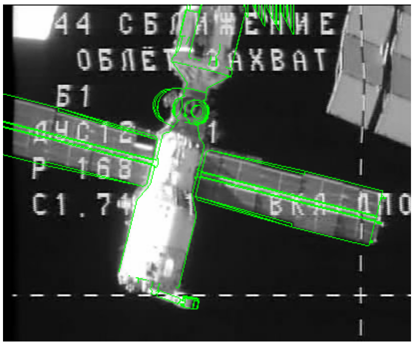

Figure 11. Contour obtained from found orientation

4. CONCLUSION

In this work we presented a method for estimating the orientation parameters based on the use of a highly detailed digital model of the ISS. One of specific features of the problem that we studied is that the ISS appearance on video frames substantially changes in the course of docking approach. Analysis shows that solving such a recognition and tracking problem requires different sets of algorithms depending on the range of distances between the SCS and ISS.

In order to solve the exterior orientation problem for frame image of ISS the contour-based algorithms are used. Statistical tools such as Kalman filter are used to support the estimation process and real time object tracking. The substantial feature of the method presented in this work is the usage of a very detailed digital 3D model of ISS for subsequent generation of raster wireframe templates, used for detection in each frame and tracking of the ISS in video sequence. Algorithms based on the usage of wireframe templates show high robustness, but unfortunately they are slower than real-time. A method combining contour algorithms and Kalman filter shows a good balance of robustness and speed.

Before applying the proposed methods to real data, the algorithms were tested on several synthetic video records that were generated for various lighting condition and dynamic trajectories. The results presented show that the method can be used for algorithmic support of automatic space ship docking. A less detailed 3D representation and/or the usage of several LODs (levels of detail) depending on the distance to the ISS should improve the speed and accuracy of the tracking. In order to estimate the parameters of movement of the space cargo ship relative to the ISS in a more reliable way, it is highly desirable to use values provided by the ship navigation system. The issue of the best combination of various algorithms in the course of object tracking for real on-board video recording requires additional research.

References from Journals:
Geusebroek, J.-M., Smeulders, A.W.M., van de Weijer, J., 2003. Fast anisotropic Gauss filtering. *IEEE Transactions on Image Processing*, 12, pp. 938-943.

Lepetit, V., Fua, P., 2005. Monocular model-based 3D tracking of rigid objects: a survey. *Foundations and Trends in Computer Graphics and Vision*, 1, pp. 1-89.

Lowe, D.G., 1991. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13, pp. 441-450.

References from Other Literature:
Calonder, M., Lepetit, V., Strecha, C., Fua, P., 2010. BRIEF: Binary Robust Independent Elementary Features, *Proceedings of the 11th European Conference on Computer Vision* (ECCV), pp. 778-792.

Vacchetti, L., Lepetit, V., Fua P., 2004. Combining edge and texture information for real-time accurate 3D camera tracking. *Proceedings of the International Symposium on Mixed and Augmented Reality* (*ISMAR*), pp. 48–57.