USING VIDEO IMAGES FOR DETERMINING RELATIVE DISPOSITION OF TWO SPACECRAFTS

V.A. Simakov, V.F. Gubarev, N.N. Salnikov, S.V. Melnichuk

Space Research Institute of the National Academy of Science of Ukraine and State Space Agency of Ukraine, Kyiv, Ukraine

В работе рассматривается математическая проблема определения взаимного положения и ориентации двух орбитальных аппаратов по данным видеосъемки. Обсуждаются недостатки простейших процедур оценивания, основанных на прямом (попиксельном) сравнении изображений, и предлагается более строгий метод, обеспечивающий быстрое и точное оценивание относительных параметров. Метод составлен из трех основных частей (инициализация, обновление и наблюдение), каждая из которых допускает независимую программную реализацию. Предложенный алгоритм тестировался на упрощенной задаче и обнаружил высокую точность получаемого результата.

Ключевые слова: орбитальная стыковка, оценивание положения и ориентации, орбитальная видеосъемка, техническое зрение.

В роботі розглянуто математичну проблему визначення взаємного положення та орієнтації двох орбітальних апаратів за даними відео зйомки. Обговорюються недоліки найпростіших процедур оцінювання, що ґрунтуються на прямому (попіксельному) порівнянні зображень, і пропонується більш строгий метод, що забезпечує швидке та точне оцінювання відносних параметрів. Метод складається з трьох основних частин (ініціалізація, оновлення та спостереження), кожна з яких може бути реалізована незалежно від інших. Запропонований алгоритм випробувався на спрощеній задачі та показав високу точність одержуваного результату.

Ключові слова: орбітальна стиковка, оцінювання положення та орієнтації, орбітальна відео зйомка, технічний зір.

INTRODUCTION

The problem of mutual pose estimation of two spacecraft (a *target* with respect to a *chaser*) relates to the area of space rendezvous. Its solution strongly relies on technical facilities of each participant of the process. Natural docking conditions assume that the shell of the target is equipped with special reflectors which simplify its tracking by the chaser by means of built-in location tools. This case allows the easiest possible solution which consists in selection of attitude and position of the target for which computed configuration of reflectors coincides with the actually observed.

In this article we considered situations which do not assume friendly docking interfaces such as reflectors. Target objects corresponding for such situations would be further referred to as non-cooperative space objects (NSO). They include broken-down satellites and other parts of space debris. Thus, the *docking* problem should be reformulated as the *berthing* problem which consists in capturing NSO by the onboard manipulator of the chaser. Noted conditions require specific methods for estimation of position and attitude of the target object with respect to the chaser. These methods are generally based on computer vision techniques and machine learning theory.

Each object is being associated with its reference frame (RF) whose directions are fixed with respect to the shell. By *relative attitude* (*relative position*) we will understand attitude (origin position) of the target RF within the chaser RF. For the sake of brevity we will further skip the word *relative* when referring to these notions. Attitude is specified in terms of traditional parameters such as Euler angles or quaternion which characterize angular position of the axes of the target frame with respect to the same axes of the chaser frame. Hence, the problem is reduced to determination of mutual position and orientation of reference frames. Parameters to be determined we will refer to as configuration variables and denote as a 6-DOF vector $x = (r, \Omega)^T$, where r is a radius-vector connecting target RF with chaser RF, Ω is a vector of Euler angles.

Traditionally, there are two major approaches for solving specified problem.

- Feature-based approaches assume preliminary selection of visual features of observed object and loading them into the memory of the onboard computer. Basically, visual features represent a set of texture spots belonging to the object shell surface. Information to be obtained must be enough to recognize correlation between geometric appearance of features and configuration variables of the object. In this area, one of the most powerful methods is based on scale-invariant feature transform (SIFT) [1], which is notable for its relatively high performance, but is useless for objects with uniform texture.

- Model-based approaches require the visual 3D model of an object under observation to be available during execution. It should be noted that for most artificial orbital objects such model is really available (or, at least, reproducible) along with other design information.

The purpose of the article is to construct a model-based algorithm for online solving aforementioned problem.

First, it is essential to define the concept of model. To wide extent, this term covers any data structure whose content uniquely describes space configuration of the object. This interpretation assumes to consider as a model any mathematical construction which is enough to restore position and attitude of the object. In particular, model-based approach formally involves feature-based and reflector-based approaches. We will deal with the visual models in their proper sense, i.e. with wire-frame models which characterize geometry of the object.

DEVELOPMENT OF A MODEL-BASED ALGORITHM

The simplest solution of the stated problem assumes considering a camera snapshot as 2-dimensional array of fixed size whose items encode the color of pixels of the image. Given this array it is possible by means of exploring local variation of color intensity to build a bitmap corresponding to a contour of observed object. The bitmap representation is equivalent to specifying discrete indicator function for those pixels of the image where the largest color difference is localized. Next, it is being compared with virtual (synthetic) plane projections of 3D model corresponding to the plane of CCD-sensor. As the comparison criterion

it is usually taken the quadratic functional defined on the set of all matrices of given order, wherein the comparison test is implemented as the least squares method for this functional. We stress the fact that this solution is ineligible at real conditions since it is non-scalable depending on object position and assumes that image of the object is strictly fixed within the bounds of the snapshot. Furthermore, its implementation requires significant computational resources.

Thereupon, it is reasonable to divide the solution into several generic stages listed below.

- *Initialization* is intended to discovering rough initial approximation of configuration variables.

- *Pose refinement* provides adjustment of previously estimated values as updated video information arrives. This stage is implemented as a recurrent procedure where the first step uses initial approximation and each subsequent step is based on the previous one.

- *Pose tracking*, i.e. further adjustment of the result provided that equations of motion are available. If these equations are given the stage is being merged with the previous, if equations are unknown the stage is being skipped.

The content of the further subsections is devoted to the detailed discussion of each of the listed stages.

Initialization. As stated above, initial approximation is obligible for subsequent stages due to their recurrent nature. It is required both for preliminary tuning of estimation algorithm and for the case when the target is being lost to view or computational fail occurs. Note that this stage is redundant if there are other sources of initial approximation.

Initialization stage should be implemented as a self-suficient procedure which in its turn does not require any initial approximations. Thus, it may be much more computationally intensive than further stages. However, since initialization is nonrecurring (i.e. one-time-only), this claim is mainly defensible.

Implementation of the stage may be based on the well-known techniques from the theory of pattern recognition [2, 3].

Let's consider a procedure of initial estimation which is based on the methods of linear algebra [3]. To be exact, a video frame from CCD-sensor is being matched with the certain collection of previously prepared images which make up a training set and correspond to the known values of configuration variables. The result of the matching is an image from the training set which is closest to the actual image from CCD-matrix. Its configuration values approximately characterize position and attitude of the target. In order to make it possible to match and compare images, we build a certain linear space whose basis is composed from the items of the trainig set. The structure of this space makes it possible to approximate arbitrary image of given resolution as a linear combination of basis images, wherein proximity is understood in the metric sense.

Implementation of any image processing method assumes specific convention about the image storage format. Traditional representation of a raster graphical object involves specification of a bitmap, i.e. a matrix of the fixed size $n \times m$ whose entries characterize color of each pixel. For the aims of the present method, this matrix should be transformed into nm-vector by means of arrangement of

neighbouring columns into a single column. As the result of this operation we will get a set of p vectors that correspond to the training set of p images.

Note that the color is usually defined as an object of a certain color space. Such a space has a fixed set of primary colors (color channels), all other colors are generated as combinations of primary colors with the use of color coordinates. Each color coordinate characterizes the value of color intensity regarding appropriate color channel. In practice, the RGB color space (with *Red*, *Green* and *Blue* primary colors) is widespread. If the image is monochrome, one confines himself with a single color channel. But in the case of color images, the set of bitmaps corresponding to each channel should be specified separately. To keep things simple, we will consider only monochrome images.

Further calculations utilize the linear span of the training set of p vectors. In order to construct an orthonormal basis for this span, we make up rectangular matrix A of size $nm \times p$ and apply singluar-value decomposition (SVD):

$$A = U\Sigma V^T , \qquad\qquad (1)$$

where U, V — orthogonal matrices of order nm and p respectively, Σ — $nm \times p$ -matrix with singular values on the diagonal and nulls on other places. Singular values generalize the concept of eigenvalues for the case of rectangular matrices. By definition, they are arithmetic square roots of common part of the set of eigenvalues for the matrices $A^T A$ and AA^T. Representation (1) gives an opportunity to select the orthonormal basis corresponding to the original training set. Indeed, according to geometric meaning of SVD [4], matrix U is comprised of principal axes of an ellipsoid which is obtained from a unit sphere under transformation A with singular values corresponding to principal components of the ellipsoid. If some of principal components are insignificant compared to others they and their training images may be regarded as negligible as considered ellipsoid degenerates to the ellipsoid of lower dimension. As far as diagonal entries of Σ are filled with the singular values in non-increasing order, all images with small singular values are localized at the end of the training set with indices $\widetilde{p}+1, \ldots, p$. Then the columns of the matrix U with numbers $1, \ldots, \widetilde{p}$ constitute orthonormal basis for the linear span of the training set.

For the further construction we need to modify original decomposition (1) in the following way. Let $E = E_{nm \times p}$ be "rectangular unity", i.e. identity matrix of order $\min(nm, p)$ augmented with null rows or columns to the size $nm \times p$. Since $EE^T = I$, we have:

$$A = U\Sigma V^T = UEE^T\Sigma V^T = (UE)_{nm \times p}(E^T\Sigma)_{p \times p}(V^T)_{p \times p} .$$

We now redenote $U := UE$, $\Sigma := E^T\Sigma$ and consider U as an $nm \times p$ -matrix everywhere in what follows.

Given the orthogonal basis $u^1, \ldots, u^{\widetilde{p}}$ as the columns of the orthogonal matrix U, one can easily take projection of an arbitrary vector $e \in R^p$ on the span $\langle u^1, \ldots, u^{\widetilde{p}} \rangle$:

$$e \approx \sum_{i=1}^{\widetilde{p}} c_i u^i \qquad (2)$$

with coefficients $c_i = (e, u^i)$ taken as dot-products of the vector e with each of the basis vectors. Vector $c = (c_1, \ldots, c_{\widetilde{p}})$ is approximation of e in the subspace generated by the span $\langle u^1, \ldots, u^{\widetilde{p}} \rangle$. If $w^1, \ldots, w^p$ are corresponding expansions of the vectors of the training set in the same basis then the closest to e will be an image whose vector w^s has minimal euclidean distance $\| w^s - c \|$.

Regarding to the procedure of image fitting it should be noted the following. At first, in order for the method to give correct result, it is required the large training set to cover as many different geometric configurations of the target as possible. At second, the distance between the object and camera can be estimated by introducing additional parameters specifying the scale. This implies the procedure of fitting to be modified in the way that the sought-for coefficients include these scale parameters. In the similar manner, it is possible to take into account the layout of the target image inside the video frame as well as planar affine transformations (such as rotations and distortions) within the frame. Finally, in the case of noisy images, expansion (1) with the use of usual dot-product can cause severe divergence. The thing is in the fact that approximating vector c, whose elements were found as the dot-products on u^i, corresponds to the minimum of quadratic error:

$$E(c) = \sum_{j=1}^{nm} \left(e_j - \sum_{i=1}^{\widetilde{p}} c_i u_j^i \right)^2 . \qquad (3)$$

Actually, this means that coefficients c_i are being obtained by the least squares method (LSM). However, it is commonly known that the LSM is sensible to outliers in input data which include digital noises and occlusions. Hence, instead of implicit minimization of (3) by LSM it is proposed to implement minimization in robust norm:

$$E(c) = \sum_{j=1}^{nm} \rho \left(e_j - \sum_{i=1}^{\widetilde{p}} c_i u_j^i \right), \qquad (4)$$

where, e.g.,

$$\rho(x) = \frac{x^2}{\sigma^2 + x^2} = \frac{(x/\sigma)^2}{1 + (x/\sigma)^2} \qquad (5)$$

39

(σ is a scale parameter). The idea of representation (5) is similar to one for penalty functions. It allows to cut off exceeding deviations at each configuration variable (see Fig. 1). The image fitting problem is reformulated as nonlinear minimization which can be solved explicitly with the use of numerical methods.
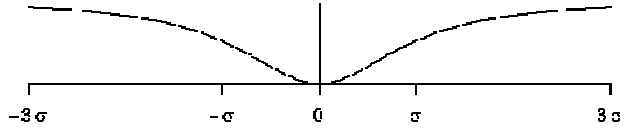

Fig. 1: Graphical view of robust norm (5)

Pose refinement. When dealing with video acquisition, it seems natural to use the fact of closeness between neighboring frames. This will presumably allow to simplify computations by means of fitting previous result with observable configuration. Obvious drawback of this approach is in requirement of rather good initial approximation. But invoking initialization stage allows to satisfy this restriction.

Following [5], we represent this idea in the form of the stepwise algorithm:

1.   Given the projection matrix P [6] of the camera, flat image corresponding to the previous step is being reconstructed. Matrix P is defined as the product

$$P = KE \tag{6}$$

of calibrating matrix

$$K = \begin{pmatrix} f_u & s & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{pmatrix},$$

where f_u and f_v are the focal lengths in horizontal and vertical directions, s specify the value of the pixel skew, u_0 and v_0 correspond to the center of the photo and euclidean projection matrix

$$E = (R, \xi),$$

where R, ξ are an orthogonal matrix and a vector which characterize angular orientation and spatial position of the camera with respect to the target. According to this, projection matrix is representable in the form:

$$P = (KR, K\xi),$$

where KR and $K\xi$ specify actually observable attitude and position of the camera which are being utilized in rendering of the flat image from the visual 3D model. Note that this step does not require reconstruction of entire graphic including

texture and lighting. The only requirement consists in building the wireframe of the properly oriented and localized object comprised by its vertices and edges.

2. Along with the visible contours generated on the previous step test points are being arranged. Further these points are used for the aims of image fitting. The number m of points depends on the size of the object as well as expected accuracy and computational facilities. The typical choice for it belongs to the range between 100 and 1000. Operations of the current and the previous steps, provided that they are supported, are being carried out by means of internal subroutines of the graphic library used. If it is impossible, this procedure is implemented manually with the use of the binary partitioning of the space [7].

3. Marked in the previously described manner synthetic image is put over the snapshot of the current video frame. Next, from each test point outer normal is being lined to the closest contour. Along with the normal, the distance (in pixels) to the contour is being calculated. Edge recognition is implemented by means of contrast analysis within the neighborhood of the test point. Previously, original image may be passed through the involutional Sobel transform [8] in order to emphasis visible contrasts and smooth the noise. As the result, we will obtain an m-dimensional vector d of signed distances between expected and observed contours in each test point.

4. Given the vector of distances d, we construct the matrix M defining the change of attitude and position of the object with respect to the previous frame. This matrix is of the following structure:

$$M = \begin{pmatrix} R & \xi \\ 0_{1\times3} & 1 \end{pmatrix},$$

where symbols R and ξ have similar meaning as for the matrix E. One can easily see that the set of all such matrices, denoted as $\mathrm{SE}(3)$, forms the Lie group with respect to the matrix product operation. Generators of this group are the elementary matrices $G_i, i = 1,\dots,6$ corresponding to unitary translations and rotations by $90°$ about each axis [5]. Each matrix $M \in \mathrm{SE}(3)$ may be expressed in terms of generators G_i as follows:

$$M = \exp\left(\sum_i \alpha_i G_i\right), \tag{7}$$

where α_i is a real coefficient. If changes on the adjacent frames are small (which is exactly the case of our interest) it should be enough to take only linear terms of the matrix exponent series:

$$M \approx I + \sum_i \alpha_i G_i. \tag{8}$$

Hence, the major difficulty of the present step consists in discovering coefficients α_i. Given the vector d of discrepancy and structure constants f_i^k of

acting i-th generator over k-th test point, this problem can be solved in terms of LSM:

$$\alpha = \operatorname{argmin} S(\beta), \qquad (9)$$

where

$$S(\beta) = \sum_k (d^k - \beta_i f_i^k)^2. \qquad (10)$$

It can be proved [5] that exact solution of (9) is given by the formula:

$$\alpha_i = \sum_j C_{ij}^{-1} v_j, \qquad (11)$$

where

$$
\begin{aligned}
v_i &= \sum_\xi d^\xi f_i^\xi, \\
C_{ij} &= \sum_\xi f_i^\xi f_j^\xi.
\end{aligned}
$$

Substituting computed coefficients into (8) we obtain desired matrix M.

5.   Matrix M allows to update euclidean projection matrix: $E \leftarrow EM$.

6.   Putting updated matrix E into (6), we actualize matrix P. After that, we wait for the next camera frame and jump to the beginning of the algorithm.

Note that approach described here does not require high computatiuonal resources owing to the usage of the simple wireframe model. Note also that due to the same reasons as in previous section, using pure LSM for α_i estimation may cause ill-posed results (especially in the case of noisy images). In order to rise reliability of the LSM, it is proposed to use its weighed modification choosing:

$$s(d^\xi) = \frac{1}{c + |d^\xi|}, \qquad (12)$$

as the weighting factor (c is a numeric parameter).

Pose tracking. If the equations of relative motion of NSO are available, further improvement of the result is possible. For this, the pose tracking problem should be formulated. Equations of motion can be derived from the general principles of dynamics but their usage requires the knowledge of mechanical parameters of the target object. If this is not a problem, the system of equations can be processed by means of certain estimation procedure such as Kalman or Bayesian filtering [9, 10].

Here we will only describe the general scheme of this approach. First of all, equation of motion should be established. We will make use of the Newtonian equations of the motion in the gravitational field of the Earth [10] in order to describe the chaser motion in RF of the target object. Although the attitude dynamics is nonlinear, it is possible to linearize it and express the final system in the canonical form:

$$\dot{x} = Ax + Bu, \tag{13}$$

where $x = (x_1, x_2, x_3, \dot{x}_1, \dot{x}_2, \dot{x}_3, \theta_1, \theta_2, \theta_3, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3)$ is the state vector, $u = (F_1, F_2, F_3, N_1, N_2, N_3)$ is the vector of control action (which includes forces and torques along each axis respectively), A is the matrix of the system, B is control matrix. We recall that the state of the system is specified in terms of the phase space which in our case consists of translational and rotational parts. With this in mind, matrices from (13) are written in the block form as follows:

$$A = \begin{pmatrix} 0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ A_p^{(1)} & A_p^{(2)} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & A_a^{(1)} & I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & A_a^{(2)} \end{pmatrix}; B = \begin{pmatrix} 0_{3\times3} \\ B_p \\ 0_{3\times3} \\ B_a \end{pmatrix}. \tag{14}$$

Here submatrices $A_p^{(i)}, B_p$ contain internal parameters and control describing movement of the mass center, whilst $A_a^{(i)}, B_a$ correspond to attitude dynamics. They can be expressed in terms of dynamical parameters in the following way:

$$A_p^{(1)} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \omega^2 & 0 \\ 0 & 0 & 3\omega^2 \end{pmatrix}; \quad A_a^{(1)} = \begin{pmatrix} 0 & 0 & \omega_0 \\ 0 & 0 & 0 \\ -\omega_0 & 0 & 0 \end{pmatrix};$$

$$A_p^{(2)} = \begin{pmatrix} 0 & 0 & 2\omega \\ 0 & 0 & 0 \\ -2\omega & 0 & 0 \end{pmatrix}; \quad A_a^{(2)} = \omega_0 J^{-1} \begin{pmatrix} J_{31} & 2J_{32} & J_{33} - J_{22} \\ -J_{32} & 0 & J_{12} \\ J_{22} - J_{11} & -J_{12} & -J_{13} \end{pmatrix}; \tag{15}$$

$$B_p = \frac{1}{m_c} I_{3\times3}; \qquad B_a = J^{-1},$$

where $\omega = \sqrt{GM/r_t^3}$ is the angular rate of the chaser with respect to the target, G is universal gravitational constant, M is the mass of the Earth, r_t is the orbital altitude of the target above the Earth; ω_0 is the angular velocity of rotating RF associated with the chaser; m_c and J are respectively its total mass and inertia matrix.

Two principal questions arise here. First is what are the values of the system parameters and second is which quantities should be used as measurements for the stated model. The first question we assume to be answered according to the mission documentation. As for the second one, we will use actual values of configuration variables obtained during previous stage. Furthermore, we will assume that the additive white noise acts on our model. This allows implementation of classical Kalman filtering for estimation of configuration variables.

One-step time discretization for (13) gives the final computational model:

$$x_{k+1} = Fx_k + Gu_k + w_k;$$
$$y_k = Cx_k + v_k, \tag{16}$$

where $F = I + \tau A$, $G = \tau B$, $C = \mathrm{diag}(I_{3\times3}, 0_{3\times3}, I_{3\times3}, 0_{3\times3})$, w_k, v_k are independent normally distributed random values whose characteristics are supposed to be given. Alternatively, it is possible to rewrite (16) according to the exact solution of the continuous system (14) in terms of its fundamental matrix [10].

MAIN RESULTS

In order to illustrate workability of proposed algorithms, we have constructed a simplified 3D model of an abstract geometric object without surface and texture, namely edge skeleton of a tetragonal prism. We suppose that position of the object is fixed and known, whereas attitude is mutable and shall be determined during estimation. Taking into account that, due to symmetry, for different attitudes of this object we may have identical images, we will assume that the set of all possible configurations is reduced in the way where there is one-to-one correspondence between images and configurations. Assume also that rotations of the object during its observation always remain within the set of these configurations. For initialization we take 100 training pictures, pose refinement is implemented with the use of 100 test points for image fitting.

For the aims of modeling, we have utilized results of the previous section. Note that this is the case where the inertia matrix is diagonal. For a fixed initial attitude we have obtained solution of problem (13) and formed associated input images with the step of 0.5 time units.

Subject to specified assumptions, we try to reconstruct configuration variables using the stages explained in previous sections. For this, proposed algorithm has been programmed and executed on a typical desktop PC. After substitution of initial data, the following results were obtained. Initialization stage for the objects of given structure gives acceptable recognition with the error within the range of 10° in the neighborhood of true attitude by each angle. Fig. 2 contains illustration of image fitting for initialization stage. This operation is being executed during about 100 ms. On the other hand, pose refinement restores exact attitude provided that the image flow arrives without interruption. Maximal time for each step was about 10 ms with maximal error of about 10^{-4}. Note that the pose tracking stage for our case did not lead to significant improvements (see Fig. 3). However, if the input flow has discontinuities, it provides validation at good level of accuracy (about 10^{-2}).
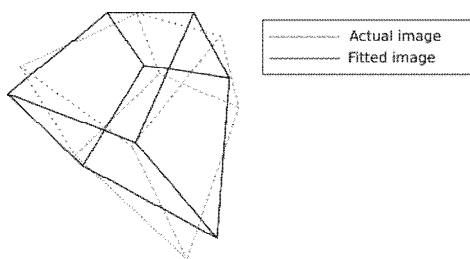
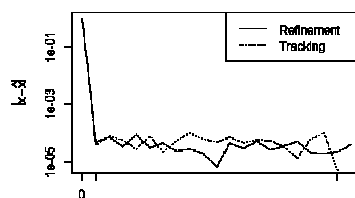Fig. 2: Initial image fitting

Fig. 3: Observed error of iterative estimation

CONCLUSION

Our researching has covered crucial stages of the problem of determining space configuration of the target spacecraft with respect to the chaser. Each stage is characterized by its requirements and implementation details. Technically, the most important is initial stage as it does not require a priori information about position and attitude of the moving target and is able to work with static images. Thereafter, initialization, although it gives rather rough estimation of configuration variables, is the most numerically expensive. That's why it is supplemented with two additional stages whose execution is much cheaper and that allow drastically improving the quality of the result. The latter is mainly related to the fact of usage dynamically updated video information. According to this, pose refinement operates with the video stream and is based on the mutual closeness of neighboring scenes. The stage of pose tracking is optional. It requires the equations of relative motion to be known and provides additional validity control in the case of signal loss.

Computational results have confirmed preliminary assumptions about accuracy and complexity of each stage. For the simple example considered in this work we could obtain very high accuracy at moderate performance.

It should also be noted that the content of considered stages may differ depending on mission requirements. Therefore, its program implementation should be replaceable. This, in turn, makes it critical to think over communications protocols and interfaces between the units of onboard computer and implement final subroutines in accordance with these conventions. This will provide independence of the hardware on the software. The present paper has not touched this question.

1. Lowe D.G. Distinctive Image Features from Scale-Invariant Keypoints // International Journal of Computer Vision, 2004. — 60 (2). — P. 91–110.
2. David P. SoftPOSIT: Simultaneous Pose and Correspondence Determination // International Journal of Computer Vision, 2004. — 59 (3). — P. 259–284.
3. Black M.J., Jepson A.D. EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation // International Journal of Computer Vision, 1998. — 26 (1). — P. 63–84.
4. Trefethen L.N., Bau D. Numerical Linear Algebra. — Philadelphia: SIAM, 1997. — 361 pp.
5. Drummond T., Cipolla R. Real-Time Visual Tracking of Complex Structures // IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002. — 24 (7). — P. 932–946.

6. Hartley R., Zisserman A. Multiple View Geometry in Computer Vision // 2nd Edition. — Cambridge University Press, 2004. — 655 pp.
7. Paterson M.S., Yao, F.F. Efficient Binary Space Partitions for Hidden-Surface Removal and Solid Modeling // Discrete and Computational Geometry, 1990. — 5. — P. 485–503.
8. Kelsey J.M., et. al. Vision-Based Relative Pose Estimation for Autonomous Rendezvous and Docking // 2006 IEEE Aerospace Conference. — 20 pp.
9. Wenfu X., et. al. Autonomous Rendezvous and Robotic Capturing of Non-Cooperative Target in Space // Robotica, 2010. — 28. — P. 705–718.
10. Fehse W. Automated Rendezvous and Docking of Spacecraft. — Cambridge University Press, 2003. — 495 pp.

USING VIDEO IMAGES FOR DETERMINING RELATIVE DISPOSITION OF TWO SPACECRAFTS

V.A. Simakov, V.F. Gubarev, N.N. Salnikov, S.V. Melnichuk

Space Research Institute of the National Academy of Science of Ukraine and State Space Agency of Ukraine, Kyiv, Ukraine

Introduction. Automatic orbital berthing systems require permanent availability of relative position and attitude of a target spacecraft. In the most general case the only source of information is video filming. Extracting mutual disposition parameters from a video frame is based upon special techniques which can be divided into two large groups: feature-based and model-based. Major difference between them is defined by data structure used for the target description (individual points for feature-based approach vs. rigorous visual model for model-based one). This article is devoted to the research of mathematical problem that appears in considering pose estimation for two orbital spacecraft in the presence of wireframe model of the target when only video filming is available.

The purpose of the article is to construct a model-based method that provides fast and accurate estimation of relative position and attitude of the target spacecraft. We discuss possible drawbacks of direct procedures based on straightforward (pixel-wise) image fitting and propose a subtle algorithm which satisfies formulated conditions.

Results. The algorithm composed of three independent parts (initialization, pose refinement and pose tracking) has been developed and tested on simple initial datum. Initialization stage, responding for rough estimation in the absence of preliminary information, has given relatively poor but quite enough accuracy for the aims of initial approximation. Pose refinement stage which is implemented as iterative procedure based on closeness of neighboring frames demonstrated almost total matching with actual values. Pose tracking (state estimation based on equations of motion) was redundant for our simple example as it could not improve the result provided by pose refinement.

Conclusions. Constructed algorithm has been tested on simplified situation and demonstrated very high precision. More realistic conditions including noises and occlusions can bring to corrupted result that should be recovered. This requires introducing additional steps into the algorithm which are reflected in the text. The notable feature of the algorithm is its high modularity which allows each stage to

be implemented and configured independently according to available resources and mission requirements.

Keywords: orbital rendezvous, pose estimation, orbital video filming, computer vision.

1. Lowe D.G. Distinctive Image Features from Scale-Invariant Keypoints // International Journal of Computer Vision, 2004. — 60 (2). — P. 91–110.
2. David P. SoftPOSIT: Simultaneous Pose and Correspondence Determination // International Journal of Computer Vision, 2004. — 59 (3). — P. 259–284.
3. Black M.J., Jepson A.D. EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation // International Journal of Computer Vision, 1998. — 26 (1). — P. 63–84.
4. Trefethen L.N., Bau D. Numerical Linear Algebra. — Philadelphia: SIAM, 1997. — 361 pp.
5. Drummond T., Cipolla R. Real-Time Visual Tracking of Complex Structures // IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002. — 24 (7). — P. 932–946.
6. Hartley R., Zisserman, A. Multiple View Geometry in Computer Vision // 2nd Edition. — Cambridge University Press, 2004. — 655 pp.
7. Paterson M.S., Yao, F.F. Efficient Binary Space Partitions for Hidden-Surface Removal and Solid Modeling // Discrete and Computational Geometry, 1990. — 5. — P. 485–503.
8. Kelsey J.M., et. al. Vision-Based Relative Pose Estimation for Autonomous Rendezvous and Docking // 2006 IEEE Aerospace Conference. — 20 pp.
9. Wenfu X., et. al. Autonomous Rendezvous and Robotic Capturing of Non-Cooperative Target in Space // Robotica, 2010. — 28. — P. 705–718.
10. Fehse W. Automated Rendezvous and Docking of Spacecraft. — Cambridge University Press, 2003. — 495 pp.