AAS 17-263

INITIAL POSE ESTIMATION USING PMD SENSOR DURING THE RENDEZVOUS PHASE IN ON-ORBIT SERVICING MISSIONS

Ksenia Klionovska,* and Heike Benninghoff†

This paper describes a designed visual model-based algorithm using the PMD (Photonic Mixer Device) sensor for the initial pose estimation of the target in future On-Orbit Servicing missions. The initial relative pose (position and orientation) in a close range has to be estimated, starting less than 7 meters between target and a camera. The verification of the algorithm is conducted by comparing the estimated pose with a ground truth. The ground truth is derived from the high-accuracy hardware-in-the-loop European Proximity Operations Simulator offered for the simulations of On-Orbit Servicing scenarios on the ground.

The results of the simulations have shown the feasibility of the algorithm to estimate the pose with sufficient accuracy as required for a pose initialization algorithm. Consequently, the designed algorithm is applicable for the initial pose estimation using PMD sensor with definite working parameters and conditions.

INTRODUCTION

On-orbit failures, space debris and orbital crowding are the prerequisites for the high demand in the field of On-Orbit Servicing (OOS) missions. Nowadays, there are more than 100 million‡ of artificial objects orbiting Earth, e.g. active/inactive satellites, used rocket stages, or other fragmentation debris. Partly damaged and uncontrolled satellites represent a hazardous risk for the "healthy" ones and have to be removed from operational orbits or repaired to the operating status. [1,2]

The orbital real estate at Geostationary Earth Orbit (GEO) is a limited resource that is in high demand for communication satellites. Removing the inoperative satellites at GEO and reuse their orbital slots or refueling to extend the satellite's lifetime are also the issues of the servicing activities.[3] OOS reduces on-orbit losses of the satellites by maintaining and improving space-based capabilities without launching new spacecrafts[4].

An On-Orbit servicing mission consists of the following main phases: inspection; rendezvous and docking (RvD); and manipulation and maneuvering missions. Nowadays, there is a

* DLR/GSOC, Münchener Str. 20, 82234, Germany, Tel. +49 8153 28 3465, E-mail: ksenia.klionovska@dlr.de.

† Dr., DLR/GSOC, Münchener Str. 20, 82234, Germany, Tel. +49 8153 28 1465, E-mail: heike.benninghoff@dlr.de.

‡ Garcia, M., Space Debris and Human Spacecraft, Last updated: 27.07.2016, https://www.nasa.gov/mission_pages/station/news/orbital_debris.html

tremendous need to provide OOS duties autonomously, regarding the target satellite as a non-cooperative and passive target.[5]

In this work we focus mainly on the rendezvous part, especially, in the close range rendezvous phase (less than 7 meters). In order to maneuver and approach the target the servicer spacecraft has to determine the relative 6 degrees of freedom (DOF) (position and orientation) between the approaching spacecraft (chaser) and the target. At the moment two groups of optical technologies are taken in operation for rendezvous missions.[6, 7] The first one contains 2D cameras (monocular- or stereo- cameras) and the second includes some types of LIght Detection And Ranging (LIDAR) sensors (e.g. scanning LIDAR or Flash LIDAR).

In 2000, the new type of 3D Time of Flight (ToF) sensor appeared on the terrestrial market: the Photonic Mixer Device (PMD) camera.[8, 9] The PMD camera is a ranging system, where the distance to the object can be measured for each pixel of the sensor chip. Modulated light from the LEDs of the camera illuminates the scene, phase shift between the emitted and reflected signal is measured and the distance to every individual pixel can be worked out.[8]

For simulation and verification of the OOS scenarios before the real mission takes place, German Aerospace Center operates the European Proximity operations simulator (EPOS 2.0).[10] It is a hardware-in-the-loop RvD simulator, where a client satellite mockup is mounted on one robot of the EPOS facility and optical sensors, such as the so-called DLR-Argos3D - P320 with an included PMD chip, is mounted on the second robot.

In this paper, we propose and prove the designed initial relative model-based pose estimation algorithm using the data provided from the PMD sensor. It is supposed that the CAD model of the target is known in the form of the vertices with predefined normal vectors to the surfaces in every vertex. We tested the algorithm for several distances between two robots and for two shapes of the known 3D model. Here the term shape means consideration only of some parts of the CAD model and not the whole 3D mockup model.

Comparing the estimated poses with a ground truth from the EPOS facility, we measured divergences in the translation and rotation components. The maximum acceptable limits for the divergences are predefined in accordance with that fact that we process the raw PMD data from the sensor for the initial pose estimation. It means we do not filter the outliers in the depth images and do not provide the distance calibration. Moreover, we take in account some inabilities of the camera by working in different environmental conditions in terms of occasionally partial loss of the taken object in the image. Consequently, we determine the follow bounders for the performance of the initial pose estimation: for the translation variables along X, Y, and Z axes is 20 *cm*, for the rotation angels around X, Y, and Z axes is10 *degrees*. A subsequent pose tracking and refinement can use the result of the initial pose acquisition, presented in this paper, as start value.

POSE ACQUISITION ALGORITHM

Overview of existing pose estimation algorithms

We suppose in our case that the model-based pose estimation is a process of estimating the rigid transformation (translation and rotation) between model point cloud and source point cloud. Some simple deterministic methods as Principal Component Analyses (PCA) and Singular Value Decomposition (SVD) are used quite often for these purposes.[11] Both algorithms are pairwise registration based on the covariance matrix and the cross correlation matrix between two point sets. The other common approach is a spin image.[12] For this algorithm, sets of the spin images are constructed for both considered point clouds. After the comparison of these spin images

among each other and finding similarities, the surfaces are declared as matched and the transformation matrix is found. Along with the above referred algorithms, it is likewise quite usual to use different variants of the Iterative Closest Point (ICP)[13] algorithm for pose estimation. The key idea of the ICP algorithm is iteratively to minimize the average squared distance between the nearest points from the model and scene cloud sets. The algorithm continues until the error converges or the maximum number of iterations is reached.

The other known possible methods are based on the calculation of so-called features (or descriptors) for 3D point clouds.[14] In the work of Luis A. Alexandre14[14] the author discusses different possible descriptors, which can be used and exist nowadays in a PCL library.[15] Throughout the variety of all the suggested descriptors, here we focus on the point pair feature descriptor.

Proposed Method

Initial pose estimation algorithm based on the point pair feature descriptors using ToF cameras is outlined in the work of Drost et al.[16] For this method it is presumed that the model and the scene are presented in the form of the oriented points (e.g. vertices and normal vectors in every vertex): $m_i \in M$ points belong to the model and $s_i \in S$ points belong to the scene, respectively. Firstly, the global description model is calculated, which includes point pair feature vectors for different combinations of the model's vertices. Secondly, scene descriptors are calculated when the data arrives from the sensor. This global model is used further for finding corresponding matches between a set of the scene's and model's descriptors. When the matches have been found, the voting scheme and clusterization processes for the best pose of the object are involved. By varying some initial parameters of the algorithm, it was noticed, that the pose refinement process is still necessary, because discrepancies of the rigid transformation between the model and the scene point clouds exist. The ICP algorithm is employed and refines the pose retrieved after the clustering phase.

Point Pair Feature Vector

Let us consider m_i as a reference and m_j a referred point throughout all model points M, where the point normal vectors are n_i and n_j, respectively. In the Equation (1),

$$F_m(m_i,m_j) = (f_1,f_2,f_3,f_4) = (\|d\|,\angle(n_i,d),\angle(n_j,d),\angle(n_i,n_j)) \quad (1)$$

the point pair feature vector for the model F_m is defined as a four component vector and composed by the Euclidean distance $\|d\|$ between two points, as well as an angle between d vector and the normal n_i, an angle between d vector and the normal n_j, and an angle between two normal vectors n_i and n_j.

Vector F_m and the components from the Equation (1) are depicted on the Figure 1. The distances and angles are sampled in steps by setting parameters d_{angle} and d_{dist} as follow: $d_{angle} = 2\pi/n_{angle}$ with an angle sampling value n_{angle}, which refers to the number of angles bins to be quantized; and d_{dist} dependent on the sampling rate τ_d and model diameter $d_{dist} = \tau_d diam(Model)$.The model diameter is defined as Euclidean norm of a vector constituted by the differences between maximum and minimum values of the vertices coordinates.

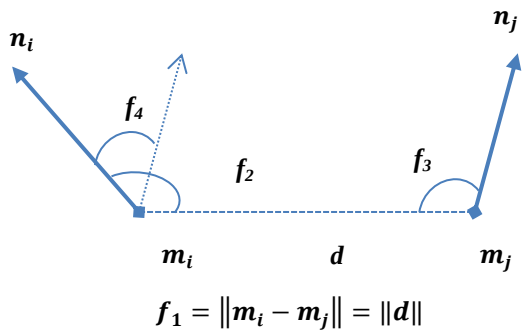

Figure 1. Illustration of the point pair feature vector.

Description of the Pose Estimation Algorithm

As it was already mentioned, we have to construct the global description model, by calculating the point pair descriptors. Here the global description model is presented in the form of a hash table, where the feature vector *F* is used as a key.

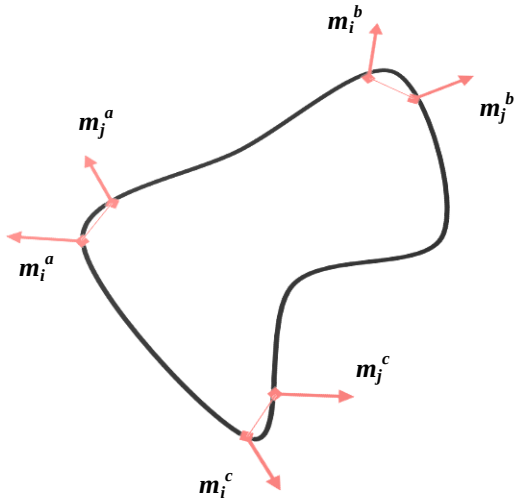

Figure 2. An example of similar point pairs of the model.

The model can contain some similar point pair feature vectors, e.g. depicted in Figure 2. Reasonably, we put together equal feature vectors in the same cell reducing the dimension of the hash table. Usually, in one cell of the hash table there is more than one pair of points of the model.

Having created the hash table, the global model is created. Since the point cloud *S* has been obtained with a PMD camera, the point pair feature vectors $F_s(s_i, s_j)$ are calculated as well for the pairs of points(s_i, s_j). As we opted previously, the feature vectors F_s are the keys for our hash table.

We extract out all pairs (m_i, m_j) from the hash table, if the feature vector of the model F_m is equal to the feature vector of scene F_s. This chain of successive action is illustrated in Figure 3.

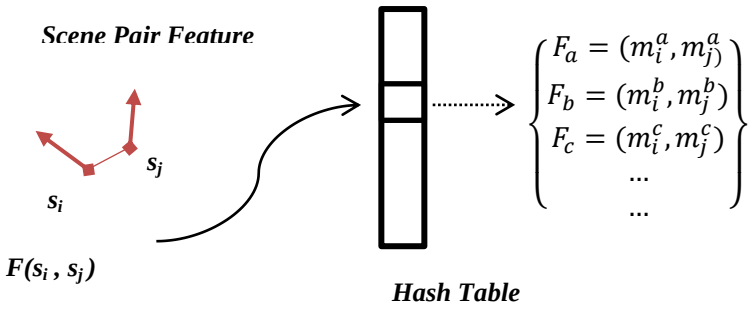

Figure 3. An extraction of the similar pairs of points for the scene and model.

Once the match between two pairs of points exists, one is able to compute the rigid transformation that aligns the model to the scene. In the work of Drost et al.[16], the author introduces the meaning of the *local coordinates*. Local coordinates are pairs of the form (m_i, α) with respect to reference point s_i, where α denotes the rotating angle and m_i is a reference point on the model. For local coordinates, one can derive the rigid transformation. The transformation that aligns the model with a scene is computed via follow expression is given as follows:

$$T_{m\to s} = T_{s\to g}^{-1} R(\alpha) T_{m\to g} \tag{2}$$

In the Equation 2, the transformation $T_{s\to g}$ moves the scene point s_i into the origin and aligns its normal n_i^s with the x-axis, see the Figure 4.

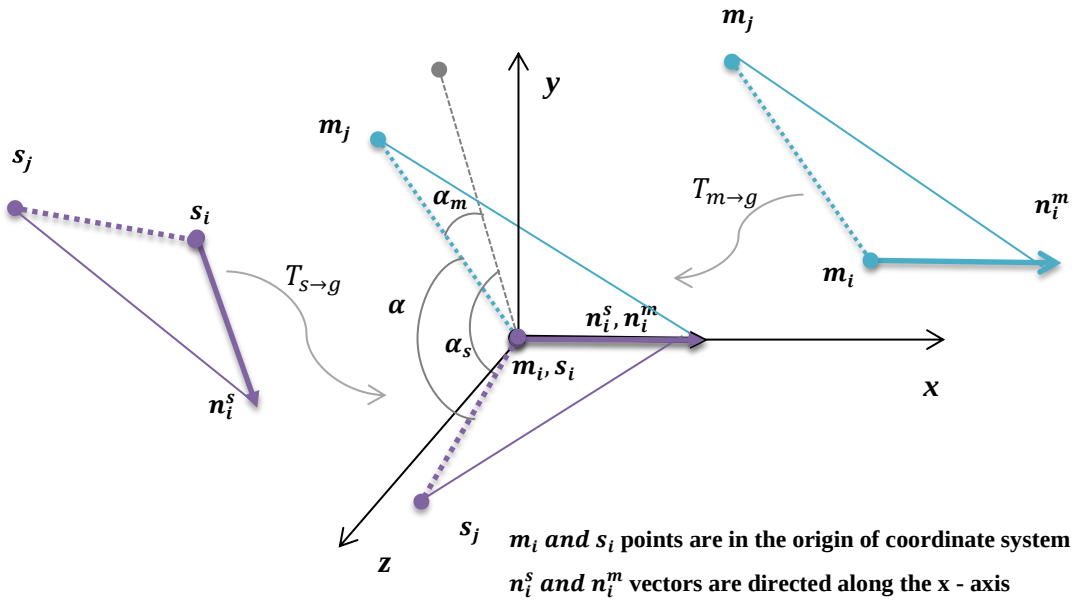

Figure 4. A transformation that aligns the model with the scene.

The same procedure $T_{m\to g}$ is implemented for the model reference point m_i and the corresponded normal vector n_i^m allocable to that point. In order to complete the final transformation $T_{m\to s}$, one more transformation $R(\alpha)$ is needed to align two left points, m_j and s_j. The rotating angle α is determined as follows:

$$\alpha = \alpha_m - \alpha_s, \tag{3}$$

where α_m is the angle between the vector m_i- m_j and an upper xy half-plane of the intermediate coordinate system and α_s is the angle between the vector s_i- s_j and the upper xy half-plane of the intermediate coordinate system (see Figure 4).

To find the best *local coordinates* at a specific point s_i, the number of points in the scene lying on the model has to be maximized. For that purpose Drost et al.[16] suggested to use the 2D accumulator massive, where the hypotheses vote for the local coordinates. The rows of the accumulator massive correspond to the reference points of the model m_i and the columns correspond to the sampled rotation angles α. The size of the accumulator massive is $N_m \times N_{angle}$, where N_m is equal to the number of the model points m_i and N_{angle} is the number of sample steps of the rotation angle α.

When the voting process takes place, all point pairs corresponding to the model (m_i, m_j) are retrieved out from the cell for each calculated point feature vector F_s. Using the Equation 3, the rotation angle α is computed. After completing these steps, it is supposed that we have local coordinates (m_i, α), which can move (m_i, m_j) to (s_i, s_j). The vote for this hypothesis is thrown in the related cell of the local coordinate (m_i, α) in the accumulator massive. It should be noted, that every voting result has a certain number of votes. When all point feature vectors of the scene F_s are processed, the set of local coordinates with the highest number of votes is retrieved from the accumulator massive. The transformations from the model to the scene coordinate frame are calculated for each returned local coordinates.

Consequently, knowing the series of object poses from each reference point, the final pose must be extracted. For that purposes, the object poses are grouped in one cluster if they do not differ in translation and rotation components more than predefined thresholds. For the translation component the threshold is taken as 1/10th of the model diameter,$t_{thresh} = 0.1 * diam(M)$, and for the rotation component as $r_{thresh} = 2 * \pi/30$. Poses collected in each cluster are averaged and the votes are summed up. The clusters with the highest scores are considered as the applicants for the best estimated relative pose.

Pose Refinement

In some cases, the issues of the proposed algorithm vary quite significantly from the correct pose of the object. In this situation, we would suggest to include the pose refinement algorithm as an extra correction step. Since the pose had been obtained by the presented method above, the point cloud of the model has been transformed to the point cloud of the scene. In order to revise the transformation, we applied the Iterative Closest Point algorithm[13], similarly with the works of Hinterstoisser et al.[17] and Birdal et al.[18] By that, the difference between two point sets of the model and the scene is minimized:

$$E(R,t) = \sum_{i=q}^{N} \|Rs_i + t - m_i\|^2 \tag{4}$$

When the error converges or maximum number of iterations is reached in the Equation 4,

one gets the refined pose.

3D TIME-OF-FLIGHT SENSOR

PMD camera

The vision camera, a DLR-Argos3D - P320 camera, has been used to measure the relative position and attitude (pose).

Figure 5. DLR-Argos3D - P320 camera.

The DLR-Argos3D - P320 camera, see Figure 5, combines two types of sensors inside of the housing: a 2D CMOS sensor and a ToF depth sensor, namely PMD sensor with resolution of 352x287 pixels. In the current work we use only the depth images for the initialization of the pose. On this basis, detailed information about the measurement principle and features of the PMD sensor is given below.

Measurement principle

The basic time-of-flight principle is to measure the distance from the absolute time delay between the transmitted wave fronts from the sender illumination unit and the wave fronts reflected by the object's surface[9,19]. Alternatively, the phase shift between transmitted and reflected waves can be measured and is used for computing the distance. The DLR-Argos3D - P320 camera includes 12 IR-flash LEDs, which illuminate the entire scene with a modulated light. The distance to the target can be calculated pixelwise easily as follows:

$$d = \frac{c\varphi}{4\pi f_{mod}} \tag{5}$$

In the Equation 5, c is a speed of light c=3x10^8 *m/s*, ϕ is the measured phase shift and f_{mod} is the modulation frequency of the emitted signal. The modulation frequency of the signal can be set in the proposed camera depending on the purposes of the user. In our test scenario, it was always specified as $f_{mod} = 20\ MHz$.

Depth accuracy is defined by the amount of active light arriving at each pixel. It depends on the illumination modules and the optics of the camera, as well as spectral sensitivity and active area of the pixel.[9] Investigation of the error sources in depth measurements generated inside of the camera or dependent on the environmental conditions is out of scope in this work.

RENDEZVOUS SIMULATION

In order to test and verify the visual sensors and navigation algorithms to ensure the reliable OOS missions, ground-based simulations have to be performed.

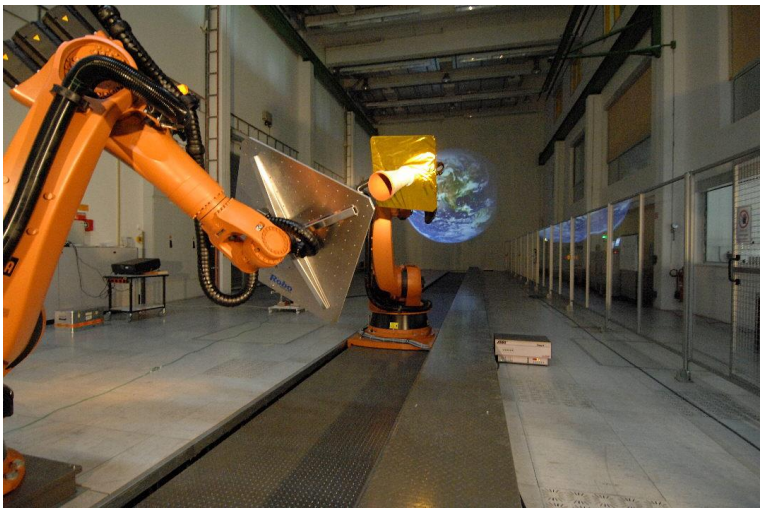

Figure 6. The robotics-based test bed EPOS 2.0.

For this reason German Aerospace Center in Oberpfaffenhofen has established a test facility, called European Proximity Operations Simulator (EPOS) 2.0, see Figure 6, with the aim to provide test and verification capabilities for complete RvD action during OOS missions.[10] It is a hardware-in-the loop RvD simulator consisting of two 6 degrees of freedom industrial robots for physical real-time simulations of RvD maneuvers. This test bed allows users to simulate the critical last rendezvous phase from 25 down to 0 meters.

For the test scenarios handled in this paper the client satellite mockup is mounted on one robot of the EPOS facility and DLR-Argos3D - P320 camera is mounted on the second robot. The relative ground truth poses of both robots are known in a Global Laboratory Coordinate frame. The PMD sensor and the target mockup are mounted fixed on the robots during the simulations. The positions of both them related to the robots are known as well. Due to the transformations between the coordinate frames, we can gain the relative pose between camera and mockup, which is admitted as the ground truth for the evaluation of the estimated pose extracted by the proposed algorithm.

CAD Model of Satellite Mockup

The initial CAD model of the satellite mockup is a high structural data set, which includes the vertices and faces. Observing the model from the frontal side, one can detect a hexagon back part and a cone front part (“nose”) with an octagon at the end.

The target mockup mounted on the robot in our laboratory is depicted in Figure 7 and the original full CAD model of that mockup is presented in Figure 8.

The full CAD model (Figure 8) contains 70002 vertices. To speed up computations, we reduce the data set and keep only the most significant geometric shapes (hexagon and cone front part). We consider further two shapes of the given 3D model. Let us name the first one as Shape 1 (see Figure 9, left) and the other one as Shape 2 (see Figure 9, right).

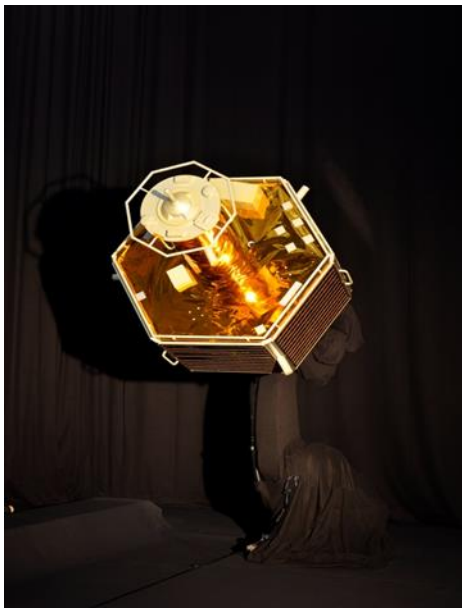

Figure 7. The target mockup in laboratory.

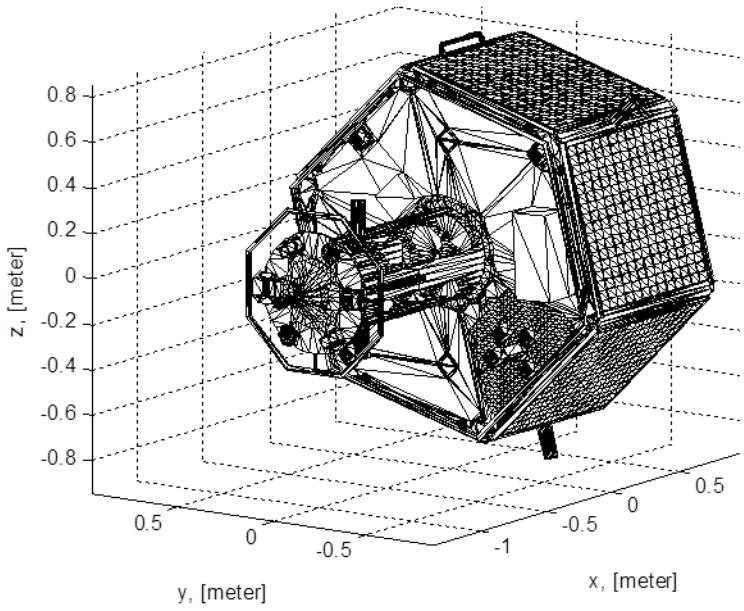

Figure 8. The original CAD model of the target mockup.

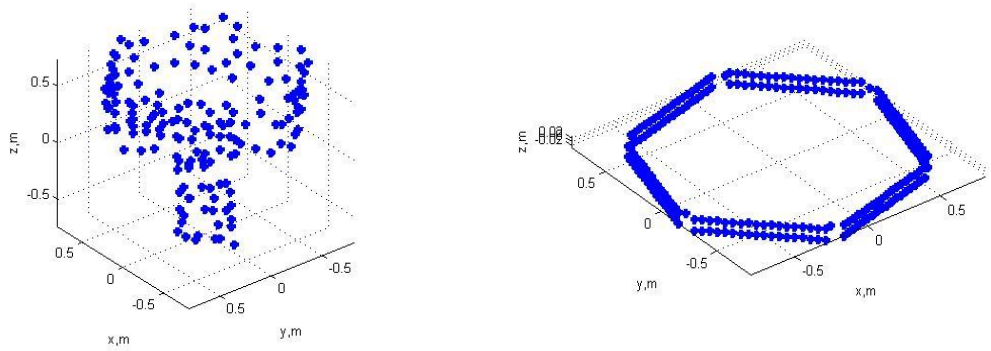

Figure 9. Left image is the Shape 1 and the right one is the Shape 2.

The mockup's surface consists of the materials, which have close optical properties as real satellites. But ToF sensors in general are not reliable and applicable for all surface materials, e.g. not suitable for specular or high reflective materials. It can lead to the completely wrong depth measurements of the target. We have to keep this fact in mind for the further evaluation of the results, because the estimated pose does not depend only on the algorithm performance, but also on the accurate measured data.

The amount of the points considered for Shape 1 is chosen in the way that the contour of the hexagon and the nose part are well-defined. For the sub-sampling implementation the Fast Poisson Disc Sampling[20] was applied. It should be noted, the computing time of the algorithm increases significantly with an increasing number of points for the model and for the scene. Due to the mockup's geometric form, there are some regions with a high amount of the parallel surface normals. These parts result in formation of the equal point pair feature vectors, which sometimes can lead to an inaccurate calculation of the pose. Therefore, the number of points for Shape 1, Shape 2 and scene point clouds must be adjustable for every single task.

Segmentation

Corresponding to the resolution of the current PMD sensor, the point cloud of the scene has more than one hundred thousand points. Therefore, the segmentation of the mockup from the surrounding objects is necessary in order to accelerate computation time of the algorithm and eliminate the background in the laboratory. Due to the close space conditions in the EPOS laboratory, we can very fast isolate the mockup point cloud from the background just by eliminating unreliable distance values. For the DLR-Argos3D – P320 camera this default value is a maximal 16-bit unsigned value. This type of segmentation is necessary provide only in the laboratory, because in the space the situation is much simpler. Since there is no background, there is no need to segment the model from it.

SIMULATION RESULTS AND ALGORITHM PERFORMANCE ANALYSIS

Test Performances

The algorithm was tested for the three different distances between the camera and the satellite mockup. For every test position the ground truth between the camera and the mockup was counted.

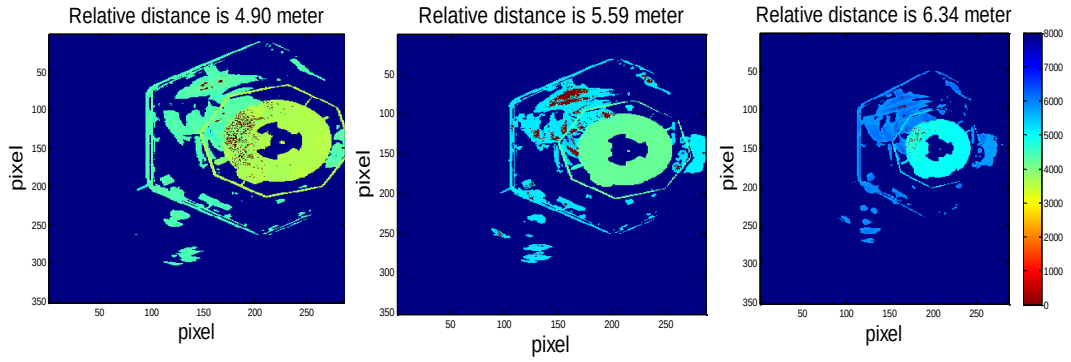

Figure 10. Depth images of the satellite mockup with 3 different distances.

In Figure 10 the depth images of the satellite mockup in the EPOS laboratory at the distances 4.90 *m*, 5.59 *m* and 6.34 *m* are illustrated. The color bar indicates the distance to the target in millimeters.

The depth images on Figure 10 do not fully reflect the contours of the mockup and some parts are missing, where the sensor could not receive reflected light. The 3D sub-sampled point cloud of the scene at the distance 6.34 *m* is shown in Figure 11 from different views. It should be noted, the partial point clouds with some outliers are of greatest interest. We are not able to predict precisely spacecraft materials and light conditions in the future OOS missions, therefore the quality of the point clouds is able to be completely differ. Accordingly, in this work we consider mainly the robustness of the applied algorithm with data sets received from PMD sensor without special tune-up of the camera.

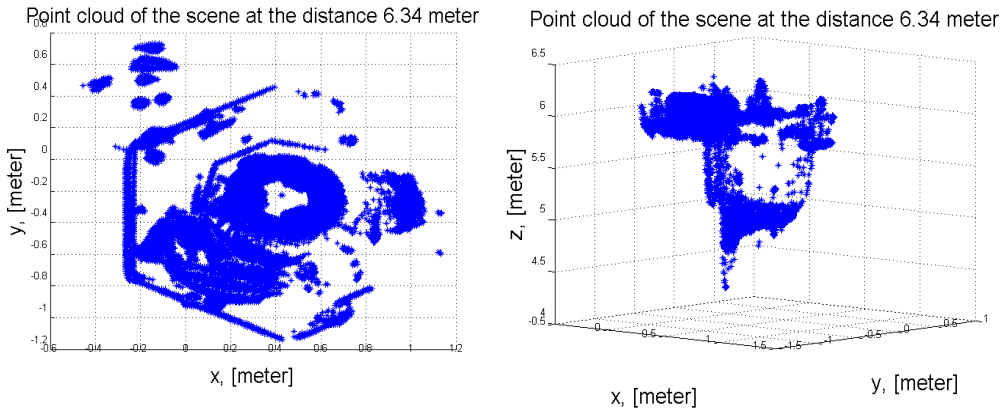

Figure 11. The sub-sampled point cloud of the satellite mockup from different views at the distance 6.34 meter.

The position estimation errors are retrieved in meters along each axis as the difference between the ground truth and estimated pose with the algorithm. The orientation estimation errors are presented in form of Euler angles. The angles have been calculated by the difference

quaternion $\Delta q = q_m \times \overline{q_{ref}}$ between the reference q_{ref} (ground truth) and the estimated q_m quaternion, and then have been converted to the pitch, yaw and roll angles.

Results with Shape 1

For the Shape 1, the algorithm was tested by $\tau_d = 0.01, \tau_d = 0.1, \tau_d = 0.15, \tau_d = 0.2$, thereby the model sampling step d_{dist} was varied. In order to sub-sample the scene point cloud, the quantity of the reference points s_i by calculating the feature vectors was determined as follow: every 2nd, 5th, 7th and 10th reference point was selected.

In the Table 1 the Root Mean Square Errors (RMSEs) are presented for tested scenarios of the proposed algorithm for the Shape 1.

Table 1. Root Mean Square Error for the Shape 1

	Position Error (m)			Angular Error (deg)		
	X	Y	Z	Roll	Pitch	Yaw
Root Mean Square Error, $\tau_d = 0.01$						
Distance 1	0.055	0.056	0.136	2.772	1.512	56.404
Distance 2	0.130	0.044	0.105	5.150	2.314	78.550
Distance 3	0.174	0.058	0.160	4.643	2.731	75.887
Root Mean Square Error, $\tau_d = 0.1$						
Distance 1	0.031	0.070	0.143	1.979	1.504	77.150
Distance 2	0.200	0.080	0.101	4.761	0.095	38.190
Distance 3	0.199	0.061	0.161	2.983	1.590	37.545
Root Mean Square Error, $\tau_d = 0.15$						
Distance 1	0.072	0.070	0.115	1.581	2.815	65.269
Distance 2	0.173	0.015	0.099	4.992	1.229	22.306
Distance 3	0.197	0.076	0.162	4.193	1.891	74.159
Root Mean Square Error, $\tau_d = 0.2$						
Distance 1	0.091	0.103	0.133	1.459	2.125	45.419
Distance 2	0.196	0.100	0.101	3.130	1.092	57.125
Distance 3	0.202	0.039	0.138	3.863	1.507	82.033

Some visual result interpretations of the initial pose estimation algorithm for the Shape1 are depicted in Figure 12 and Figure 13. The left image is a 2D view, the right one is 3D view.

In general, position errors along every axis do not exceed the predefined threshold, namely 20 *cm* for the translation components and 10 *degrees* for the rotation components. Theoretically, the matches between the Shape 1 and the point clouds are occurring, but the divergences of the estimated pose from the ground truth depend on the camera measurement errors. The depth deviations from the *Z* axis are identified by the measurement errors, which were caused by mockup's geometry and surface properties. We were expecting to get measurements of the mockup's front hexagon contour (see Figure 7), which is thin and not made from high reflected material. But in fact, we have slightly wide contour of points (see Figure 11, third image), inappropriate to the reality. It means the reflection from the surround material around the hexagon contour biases the depth measurement. The deviations from the *X* and *Y* axes are caused by the outliers presented in the depth image, as well as by the uncalibrated distance measurements.

Considering the angular errors, the most deviations appear in the rotation angle around *Z* axis. These deviations mainly are characterized by the symmetry of the hexagon contour. For the more accurate estimation of the roll angle, additional features, parts and contours of the mockup are

needed. The maximum error throughout other rotation angles around X and Y axes lies within an acceptable maximum limit and reaches 5.15 *degrees*.

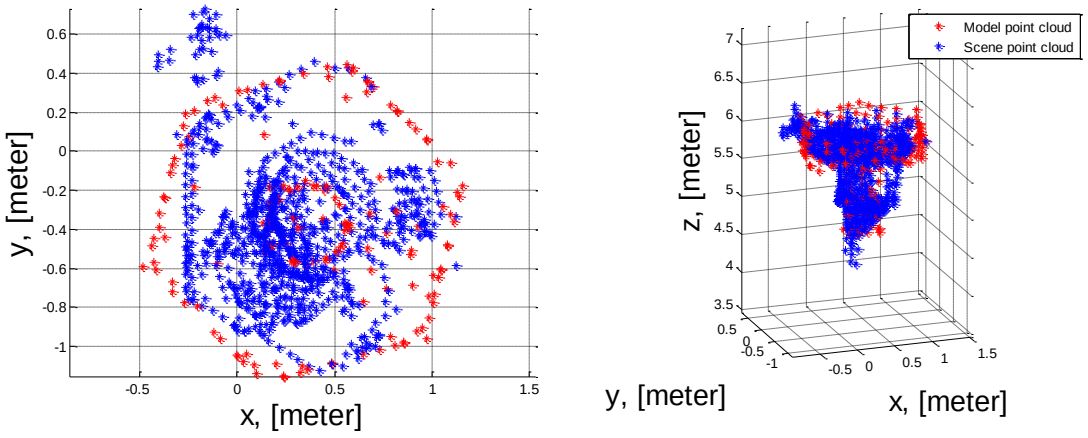

Figure 12. Output of the initial pose estimation algorithm for $\tau_d = 0.15$ and for every 2nd reference point of the scene.

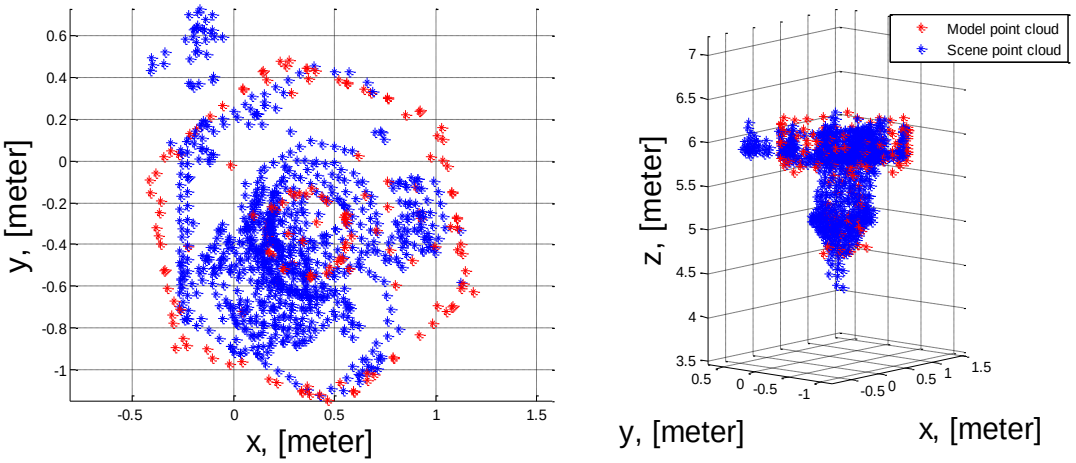

Figure 13. Output of the initial pose estimation algorithm for $\tau_d = 0.1$ and for every 7th reference point of the scene.

Results with Shape 2

For the Shape 2, the algorithm was tested by follow parameters: $\tau_d = 0.01, \tau_d = 0.02, \tau_d = 0.05, \tau_d = 0.1$. The same as in a previous scenario, the number of the reference points for the scene s_i was varied.

Let us consider the visual interpretation of some results of the initial pose estimation algorithm for the Shape 2. On Figure 14, one of the few better matches between Shape 2 and the scene point cloud is depicted. On Figure 15, the skewed point cloud of the Shape 2 states incorrected determined pose of the target. In both figures the right image is 2D view and the left one is 3D view.

In the Table 2 the RMSEs for the test scenarios for the Shape 2 are presented.

Table 2. Root Mean Square Error for the Shape 2

	Position Error (m)			Angular Error (deg)		
	X	Y	Z	Roll	Pitch	Yaw
Root Mean Square Error, $\tau_d = 0.01$						
Distance 1	0.370	0.159	0.212	52.487	23.535	24.924
Distance 2	0.349	0.069	0.117	30.803	43.904	79.997
Distance 3	0.318	0.063	0.078	13.623	16.736	127.325
Root Mean Square Error, $\tau_d = 0.02$						
Distance 1	0.526	0.060	0.119	15.711	35.705	118.011
Distance 2	0.350	0.076	0.138	31.246	41.817	110.621
Distance 3	0.328	0.021	0.039	3.492	1.589	74.637
Root Mean Square Error, $\tau_d = 0.05$						
Distance 1	0.497	0.084	0.025	2.031	1.018	38.384
Distance 2	0.342	0.082	0.133	35.411	56.318	17.278
Distance 3	0.312	0.063	0.100	28.366	63.603	119.511
Root Mean Square Error, $\tau_d = 0.1$						
Distance 1	0.438	0.115	0.120	15.878	35.931	112.67
Distance 2	0.304	0.281	0.102	36.237	39.935	68.33
Distance 3	0.269	0.310	0.366	39.62	12.628	101.26

The estimated results are much different from those derived for the Shape 1. The estimated poses have the larger divergences from the ground truth, which lie outside the defined boundaries. This is related to the shortage of the 3D points for the Shape 2. Here the maximum deviation of the estimated pose along the *X* axis reaches 52.*3 cm*, along the *Y* axis is 31 *cm*. The depth measurement errors are in the range up to 36.6 *cm*. The maximal angular errors along axes *X, Y, Z* are expressed in significant discrepancies from the ground truth: 52.48, 56.31 and 127.32 degrees respectively. Consequently, these results obtained within simulations with a Shape 2 indicate notable incorrect initial pose estimations of the target.

This experiment demonstrates that it is necessary to use all significant parts of the 3D model to obtain good pose estimation results. If we use the hexagon only (Shape 2), we will not be able to match the model with the scene points, resulting in a decreased performance.

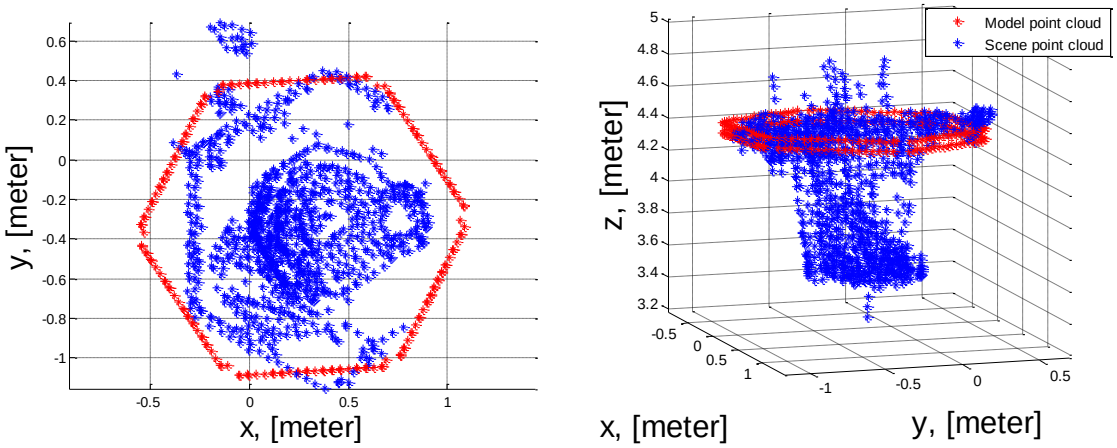

Figure 14. Output of the initial pose estimation algorithm for $\tau_d = 0.01$ and for every 5th reference point of the scene.

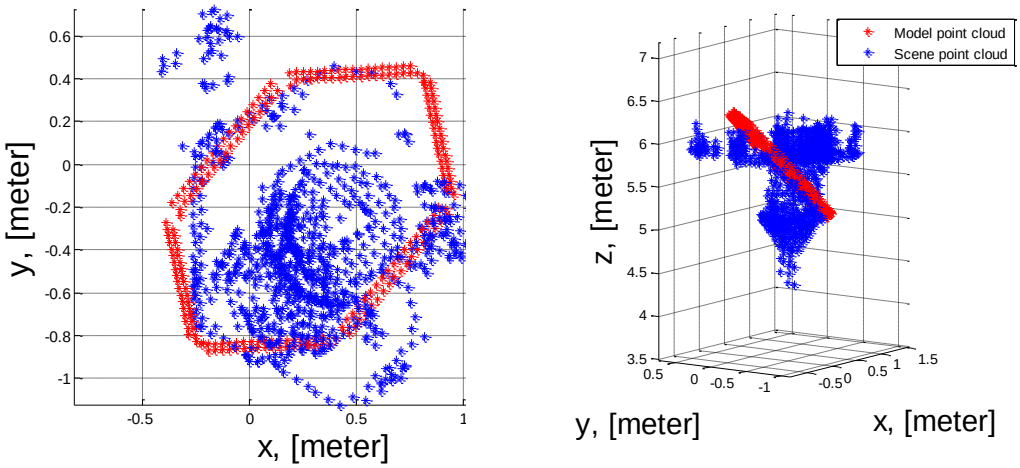

Figure 15. Output of the initial pose estimation algorithm for $\tau_d = 0.01$ and for every 10th reference point of the scene.

CONCLUSION

This paper presents the results of the initial model-based pose estimation algorithm, which could be applied in the OOS applications, namely by close range approach of the servicer satellite to the target. The key feature of present work is verification of the designed algorithm using the data sets from the PMD sensor by varying the distance between the camera and the satellite mockup. At the same time, the algorithm was tested under different adjustable parameters for the

model and the scene point clouds, as well as for two shapes of the known CAD model of the satellite mockup.

The observed results for the Shape 1 prove the ability to estimate the coarse initial pose with satisfying accuracy. It is associated largely with the completeness of the Shape1 relating to the known CAD model of the mockup, i.e. sparse model point cloud contains the hexagon contour and the nose cone.

The discrepancy of the estimated pose from the ground truth during simulations with the Shape 2 can be interpreted by the incompetence and partial shape of the model itself. Moreover, the inaccurate result is caused by the partial loss of the point cloud of the scene.

The problem in estimating the rotation angle around the *Z* axis with a Shape 1 still remains to be solved. Potentially, the solution could be addressed to the PMD sensor characteristics and possibilities in terms of acquisition of entire and high quality PMD data sets of the observed mockup. That would help us identifying some other target features and eliminating discrepancies in the estimated angle due to the hexagon symmetry.

REFERENCES

[1] Orbital Debris Quarterly News, National Aeronautics and Space Administration, Vol. 19, Issue 2, April 2015

[2] Orbital Debris Quarterly News, National Aeronautics and Space Administration, Vol. 20, Issue 1& 2, April 2016

[3] On-Orbit Satellite Servicing Study. Project Report. National Aeronautics and Space Administration, Goddard Space Flight Center, October, 2010

[4] Richards, Matthew G., Shah, Nirav B., Hastings, Daniel E.: Agent Model of On-Orbit Servicing Based on Orbital Transfers

[5] Benninghoff, H., Tzschichholz T., Boge T., Rupp T.: Hardware-in-the-Loop Simulation of Rendezvous and Docking Maneuvers in On-Orbit Servicing Missions, *28th International Symposium on Space Technology and Science*, 05.-12. Jun. 2011, Okinawa, Japan

[6] Christian, J.A, Cryan, S.: A Survey of LIDAR Technology and its Use in Spacecraft Relative Navigation, AIAA Guidance, Navigation, and Control (GNC) Conference

[7] Jasiobedzki, P., Stephen Se, S., Pan, T., Umasuthan, M., Greenspan,M. : Autonomous Satellite Rendezvous and Docking Using LIDAR and Model Based Vision, *Spaceborne Sensors II*, edited by Peter Tchoryk, Brian Holz Proc. of SPIE Vol. 5798, pp. 54-65.

[8] Schilling, K., Regoli, L.: The PMD 3D-camera for Rendezvous and Docking with Passive Objects. *In 8th International ESA Conference on Guidence, Navigation& Control Systems,* 2011

[9] Ringbeck,T., Hagebeuker, B.: A 3D Time of Flight Camera for object detection, *Oprtical 3-D Measurement Techniques*, 09-12.07.2007, ETH Zürich

[10] Boge, T., Benninghoff, H., Tzschichholz, T.: Visual Navigation for On-Orbit Servicing Missions. *5th International Conference on Spacecraft Formation Flying Missions and Technologies*, 2013

[11]Sorkine, O.: Least-Squares Rigid Motion Using SVD, Technical notes, vol.120, pp.3, 2009/2

[12] Johnson, A., Hebert, M.: Using Spin Images for Efficient Object Recognition in Cluttered 3D Scene, *IEEE Transactions on pattern analysis and machine intelligence, vol. 21 , no. 5,* 1999.

[13] Besl, P., McKay, N.: A Method for Registration of 3-D Shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.14,no.2, February 1992

[14] Alexandre L. A.: 3D Descriptors for Object and Category Recognition: a Comparative Evaluation

[15] Rusu, R., Cousins, S.: 3D is here: Point Cloud Library (PCL), *IEEE International Conference on Robotics and Automation (ICRA)*,Shanghai, China, May 2011

[16] Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model Globally, Match Locally: Efficient and Robust 3D Object Recognition. *IEEE Conference* ,2010

[17] Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N.: Model Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes, *Asian conference on computer vision*, 548-562

[18] Birdal, T., Ilic, S.: Point Pair Features Based Object Detection and Pose Estimation revisited. In: IEEE International Conference on 3D Vision ,2015

[19] Luan, X.: Experimental Investigation of Photonic Mixer Device and Development of TOF 3D Ranging Systems Based on PMD Technology, PhD thesis, *University Siegen*, 2011

[20] Bridson, R.: Fast Poisson Disk Sampling in Arbitrary Dimensions, *In CM SIGGRAPH Sketches*, p. 22, 2007