pre-print

00 (2017) 1–21

Appearance Learning for 3D Pose Detection of a Satellite at Close-range

Nassir W. Oumer[a], Simon Kriegel[a], Haider Ali[a], Peter Reinartz[b]

[a]*Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Oberpfaffenhofen, Germany*
[b]*Remote Sensing Technology Institute, German Aerospace Center (DLR), Oberpfaffenhofen, Germany*

Abstract

In this paper we present a learning-based 3D detection of a highly challenging specular object exposed to a direct sunlight at very close-range. An object detection is one of the most important areas of image processing, and can also be used for initialization of local visual tracking methods. While the object detection in 3D space is generally a difficult problem, it poses more difficulties when the object is specular and exposed to the direct sunlight as in a space environment. Our solution to a such problem relies on an appearance learning of a real satellite mock-up based on a vector quantization and the vocabulary tree. Our method, implemented on a standard computer (CPU), exploits a full perspective projection model and provides near real-time 3D pose detection of a satellite for close-range approach and manipulation. The time consuming part of the training (feature description, building the vocabulary tree and indexing, depth buffering and back-projection) are performed offline, while a fast image retrieval and 3D-2D registration are performed on-line. In contrast, the state of the art image-based 3D pose detection methods are slower on CPU or assume a weak perspective camera projection model. In our case the dimension of the satellite is larger than the distance to the camera, hence the assumption of the weak perspective model does not hold. To evaluate the proposed method, the appearance of a full scale mock-up of the rear part of the TerraSAR-X satellite is trained under various illumination and camera views. The training images are captured with a camera mounted on six degrees of freedom robot, which enables to position the camera in a desired view, sampled over a sphere. The views that are not within the workspace of the robot are interpolated using image-based rendering. Moreover, we generate ground truth poses to verify the accuracy of the detection algorithm. The achieved results are robust and accurate even under noise due to specular reflection, and able to initialize a local tracking method.

Keywords: Satellite pose detection, pose estimation, pose initialization, appearance learning, Feature clustering

1. Introduction

Visual localization of a space object such as a malfunctioned satellite is currently an interesting research topic. One of the applications foreseen is an on-orbit servicing. There exist plenty of defective satellites in space which occupy precious orbits such as geostationary orbit (GEO). The ultimate goal is either to service or deorbit them. The on-orbit servicing may be performed by launching a robot mounted on a servicer satellite. The relative pose of the client satellite with respect to a servicer needs be estimated and predicted over time in order to approach and perform a vision-based control (visual servoing) of the malfunctioned satellite.

The visual environment of a space and the optical characteristics of the target satellite which predominately create saturated pixels (due to specular surface) pose difficulties in vision-based localization. The localization of a satellite generally comprises of tracking and detection. Vision-based tracking has been recently the main focus of the on-orbit servicing, and a number of literature has been presented [1, 2, 3]. In contrast to tracking, less attention has been paid to the detection of a satellite. Therefore, in this paper we focus on the detection of a satellite for an orbit-servicing.

Detection of a satellite in our context refers to determining a region of interest (a satellite) in an image. Here we assume the image contains a single satellite, and we estimate the position and orientation (pose) of the satellite in 3D space based on images and a CAD model. Further, we assume that the specific location (orbit) of the desired satellite in space is pre-determined from ground or space based observation, i.e. no more object recognition is required. Pose detection is generally used for the initialization of any tracking method, which is based on a local optimization. A global detection of an object is difficult due to the large search space in six degrees of freedom (DOF), background clutter, and change of illumination. The detection is more difficult in case of a satellite pose estimation, as the highly intensive direct sunlight creates strong specular reflection on the surface of a multilayer insulation (MLI) of the satellite. The MLI is a thermal protective material wrapped around the satellite surface to protect the on-board electronics from radiation. The MLI is highly reflective and poses challenges for pose detection of the satellite.

In this paper we provide a pose detection and estimation method based on training the most likely lighting conditions and appearances. The proposed pose detection method integrates the state of the art methods of feature extraction, the hierarchical clustering and vocabulary tree for pose estimation. Moreover, we address the problem of the missing appearances during collecting train images in a constrained robot workspace. To evaluate the pose detection method, we conduct an experiment in a robotic testbed similar to the space light conditions and optical characteristics of the client satellite. The evaluation consists of two data sets; the first data set is used to evaluate the accuracy of the position and orientation estimates when a servicer approaches the client satellite. With the the second data set, the camera line of sight targets a specific part of the satellite, called launcher interface bracket (LIF) during approach trajectory, and is used to evaluate position estimation of an attitude controlled satellite in three degrees of freedom.

In the remainder of this paper, we review related work to our pose detection in Section 2. In Section 3 we present our methodology of satellite pose detection, while describing the offline training in Subsection 3.1 and Subsection 3.2, and on-line testing in Subsection 3.3. We present experimental results in Section 4 and finally we summarize and conclude the paper in Section 5. -

2. Related Work

In literature, a number of object detection methods exist to address various problems. View-based approaches [4, 5, 6, 7] compare a search image with the 2D views of the object precomputed from its 3D model by clustering views. Related to these methods, [8] uses no training templates but instead adapts the template through on-line learning. The view-based methods sample views, starting on the lowest level by applying an over-sampling of the views. Then a similarity measure is employed to hierarchically partition all the neighboring camera views. The views with the highest similarity are selected and merged into one cluster (view), and the similarities between the new cluster and its neighboring views are computed. The view re-sampling process is repeated until the highest similarity is below a certain threshold. This is known as a hierarchical view clustering. In addition to the hierarchical view clustering, [5] employs image pyramids for efficient recognition. The recognition speed up is achieved with exhaustive search on top of the pyramid. The highest level of the pyramid consists of fewer views, therefore exhaustive searching is fast and robust. Most recently, [7] employed a foreground and background segmentation of the image as a preprocessing step, assuming static background and a few image sequences. This pose detection method relies on hierarchical view clustering and pose estimation as [5], but first segments the object in the image before matching with reference views. For the segmentation of the foreground object, a sequence of images are assumed to be available so that the segmentation based on feature motion can be performed. The experimental results demonstrated in [5, 7] appear relevant for 3D detection of a space object. However, such methods require prominent edges of the object to succeed in accurate pose detection.

On the other hand, in order to tackle the problem of the large search space, feature-based approaches [9, 10, 11, 12] exploit image features such as corners, lines, intersection of lines and complex features from grouping of various image features. The corresponding 3D features are assumed to be on the 3D model of the object in consideration. Thus, the extracted image features (edges, corners and lines) are matched to the 3D features, and the 3D pose is computed. The main drawback of such methods lies in dealing with the possible large search space related to establishing the correspondence between the image and object features. Moreover, background clutter in the image makes the feature extraction difficult.

Descriptor-based methods [13, 14] learn local or semi-local features extracted from training images. Particularly, [13] considers a wide-baseline point matching as a classification problem. They employed the randomized-tree

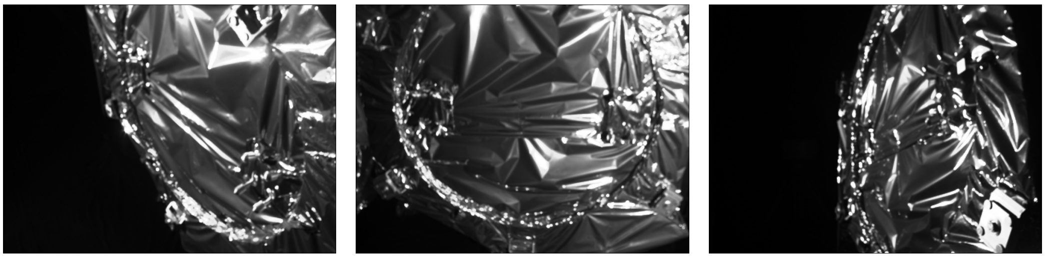

Figure 1. Images of the rear part of TerraSAR-X generated in our robotic testbed: the satellite surface is fully covered with a multilayer insulation (MLI) and consists of weak edges, hence difficult for pose detection.

for the classification of features from views. The advantage of the descriptor-based approaches is that their run-time is independent of the size of the geometric search space [5]. They show good performance in recognition of textured objects under slight illumination change. The illumination change is handled by normalizing view intensity. The authors [13] use at least one training image, while many others are generated through computer-graphics techniques such as a texture mapping and rendering of the 3D model. However, the texture mapping from a given image and the 3D rendering of specular object under a direct sunlight is inaccurate. Moreover, intensity normalization could not make the detection robust to strong illumination change, in particular for space lighting and reflective surface of the satellite.

In contrast, in this paper, we present an appearance-based pose detection method which learns both lighting and appearance from various view points, creating a database of feature points to cluster and learn in a vocabulary tree based on [15]. The observed image is then queried in real-time and the corresponding pose is computed based on descriptor matching. The training images are obtained from a camera system mounted on six degree of freedom robot. The operational workspace of the robot is limited, which poses difficulty to reach all the required view points. Therefore, we employ a 3D image warping to complement those missing views due to workspace constraint during acquisition of training data. Therefore, the main contribution lies in integrating a scalable image recognition method for pose detection in a difficult lighting condition as well as a reflective surface.

3. Pose Detection by an Appearance Learning

The appearance of a satellite changes significantly with the change of view and direction of the sun because of the reflective property of the surface. In the absence of strong edges, detection of the satellite with edge templates under space lighting is difficult. Therefore, it is demanding to achieve the desired accuracy for an initialization of a local tracking. The detection of the satellite with distinctive edges can be achieved as demonstrated in [7]. However, when the surface of the satellite is fully covered with a multilayer insulation (e.g. TerraSAR-X in Fig. 1), it is very difficult to detect the pose using edge-based method because of strong clutters created by the MLI.

In contrast, the wrinkles of the MLI and satellite structures such as blobs and junctions provide useful cues for the detection of the satellite. Therefore, we employ a feature based pose detection of a satellite by computing a descriptor vector for each region of interest. We adapt an appearance learning, in which we learn appearances of the satellite under various view points and lighting directions. To efficiently retrieve the observed image in a large number of views stored in a database, the appearance (feature) learning builds up on the popular technique of the vocabulary tree [15]. A large scale feature vector quantization based on the vocabulary tree is shown to be effective also in 3D reconstruction [16]. Therefore, the representative keypoints from several views and different direction of the sun can be incorporated into the learning scheme. In the context of satellite pose detection, the strength of such approach is its robustness against occlusion, background clutter, and illumination.

The pose detection process follows two main procedures: offline learning and on-line pose computation as illustrated in Fig. 2. For the offline training (mint part) , we collect images by sampling view points at different scales and sun directions. Features are extracted from each image and quantized in a vocabulary tree. The 3D points corresponding to the keypoints are extracted from the depth map. During the on-line phase (blue part of Fig. 2), features are extracted from the observed image and the corresponding image is searched in the vocabulary tree based on distance

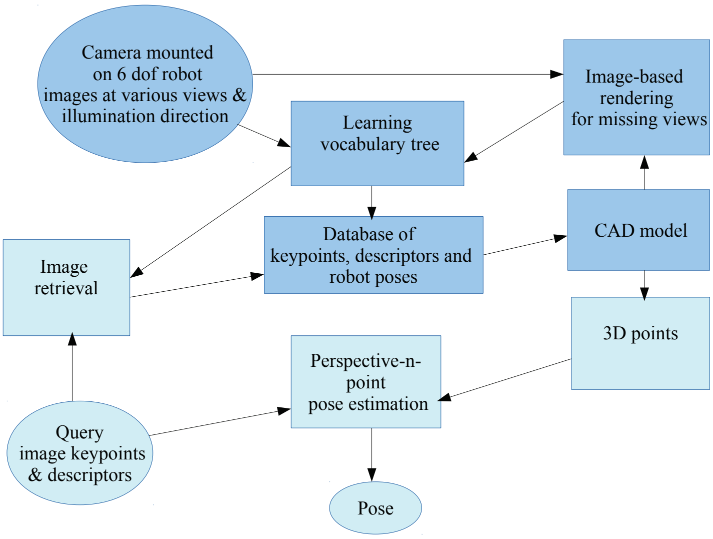

Figure 2. Illustration of the pose estimation based on an appearance learning. The pose detection is partly based on offline processing (shown in dark gray or blue) and on-line processing (in light gray or mint). The training images are obtained from cameras mounted on a robot. A few train images at the view points where the robot cannot reach them, are obtained from an image-based rendering. The extracted features are hierarchically clustered and vocabulary tree is built for fast retrieval of the query image.

metric between each feature descriptors of both, query and database images. Finally, we find the correspondence between features of retrieved and query images and compute the pose.

Photo-realistic training images can be obtained in two different ways depending on the existing information about the client satellite. In either case, we assume a CAD model and a number of image archives of the satellite can be obtained. In many cases, the CAD model of the satellite consists of geometric parts in a mesh form, and no material properties such as multilayer insulation (MLI). In the perspective of image processing, the MLI is highly determinant of the performance of the tracking, therefore it has to be modeled accurately. Reproducing the specular property of the MLI is very difficult and by itself a research area in computer graphics. One way to obtain training images is a photo-realistic rendering based on a ray-tracing [17]. The distant parallel sun light can be simulated with the computer graphics accurately, and the MLI can be modeled by texture mapping of the image of the satellite and its 3D model. However, the texture mapping of a reflective specular surface may not accurately reproduce the original material and structure property of the satellite, hence a less reliable training data.

Therefore, we reproduce the MLI more accurately with a hardware (a satellite mock-up) for training images. The MLI of the satellite can be reproduced by taking an account of the geometry and material property. The advantage of the satellite mock-up is that the true MLI can be wrapped around the surface of the satellite. We take the advantage of this method to get training images with the mock-up of the TerraSAR-X satellite. View points are first generated by placing the object in a center of virtual sphere and moving the camera around the surface of the sphere at an interval of 8° in latitude and longitude. The lower the interval is the better is the accuracy, however the size of the training images in the database explodes. Therefore, the interval of 8° in our case provides better trade off between size of the database (speed and memory) and accuracy. The trajectories generated in this manner, but in a hemisphere were implemented using a real stereo camera mounted on six DOF robot. The robot moves in pre-planned trajectories, repeating those views at different sun directions, where the actual sunlight was simulated with a high power flood light. The image recording setup, satellite mock-up, and utilized hardware are described in detail in [18]. In contrast to their image data set, we obtain images on different spherical caps instead of linear trajectories and also use different sun positions for training and testing.

The disadvantage of the proposed method is that, the workspace of the robot arm is constrained, and thus it is

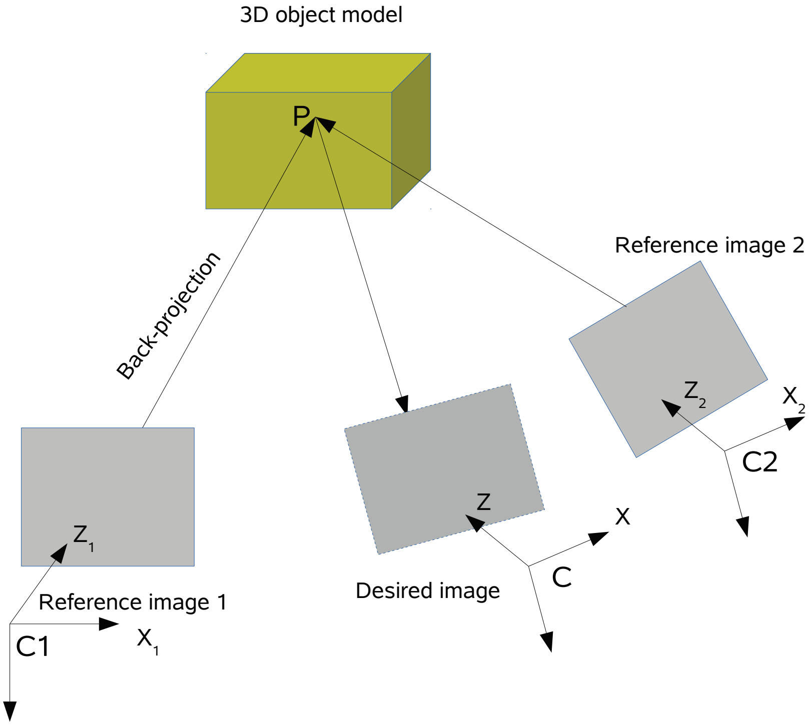

Figure 3. Image-based rendering using the 3D warping. The reference image pixels (image 1 and image 2) are back-projected to the 3D space using depth map computed from the 3D model of the object. The 3D points P are then re-projected onto the desired view to synthesize desired image.

not possible to reach all the desired view points. Although the camera can be moved to an arbitrary pose in the work space, there exist configurations of the robot which could not allow to reach some angles. In order to address this problem, we employ an image-based rendering technique based on the 3D warping. A virtual camera image can be synthesized at a desired view point from at least one reference image, by back-projecting each pixel of the reference views using the depth map and re-projecting to the new view, as shown in Fig. 3 according to the Equation (1). Given the reference image pixel p_1, corresponding depth X with respect to the desired target camera C of focal length f and a calibration matrix K, we can compute the target pixel p by forward mapping

$$\lambda p = K^{-1}(C_1 - C) + \lambda_1 K^{-1} K_1 p_1 \tag{1}$$

where C_1 and C are reference and desired (target) camera centers respectively, $\lambda = [0 \quad 0 \quad 1/f]^t K^{-1}(X - C)$. Thus, we can compute the position of target pixel p on the target screen and transfer the intensity from p_1 to p. Similarly, if a second reference camera view C_2 exists from reference image 2 shown in Fig. 3, the position of the pixel p on the target screen can be transferred from p_2 and the final intensity image at p can be computed by interpolating intensities at pixel p_1 and p_2. A rendered image using a reference image is shown in Fig. 4. The reference image is rotated 16° about the camera optical axis.

The image rendered by the 3D warping has known problems of gaps due to round off error resulting from the decimal to integer conversion of pixels, magnification when the camera moves closer and depth discontinuity causing dis-occlusion [19]. Post-processing the newly synthesized image has been proposed to handle the problems of gaps or hole due to sub-pixel location of the desired image. However, the post-processing blurs the image, which is undesirable property for feature-based matching. We employ a simple but an effective morphological dilation, which

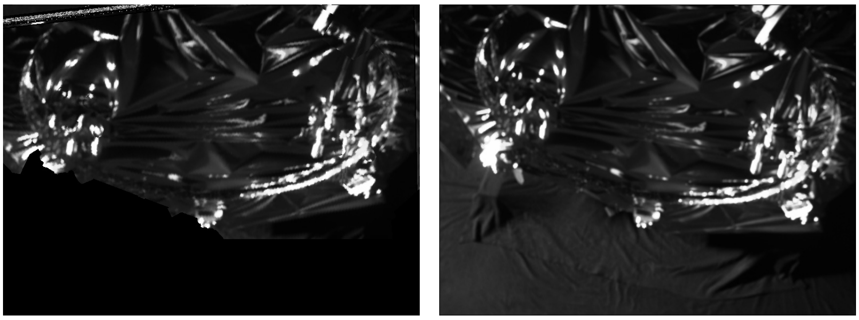

Figure 4. Image-based rendering for missing views during the training. The rendered image (left) is used for unreachable view points with the robot base based on a reference image (right).

does not blur the image to fill the small gaps. Moreover, the second view (Fig. 3) helps to reduce depth discontinuities caused by occlusion.

3.1. Feature Extraction and Clustering

Once the training images are acquired, we apply a key point detector on each image to extract and describe robust regions. In order to detect the satellite pose at a range of scales, a scale-invariant features are learned during the training. For this purpose, we employ the SIFT feature detector and descriptor [20] and represent each training image in a database of features. There exist several other rotation and scale invariant feature descriptors [21, 22, 23, 24] for recognition and image matching. The SIFT descriptor is found to perform best in recognition and image matching [25], but at the price of relatively higher computation. The SIFT descriptor is invariant to a scale, rotation and to some extent to illumination. The scale-invariance of the SIFT is essential to handle the change of a scale during matching with the query image, which may be far or close to the object. The extracted features are represented with 128-dimensional feature vector, resulting in $N \times 128$ feature descriptors for N keypoints and stored as a text file containing the feature descriptor for each point and the location, orientation and scale of feature points in database.

The feature descriptors of each training image are clustered in 128-dimensional space using k-means clustering [26]. The k-means clustering is a vector quantization method, which divides M feature vectors into k partitions (clusters); each feature vector belongs to a cluster with nearest mean. The k-means clustering is a non-convex problem, but an efficient heuristic algorithm and converges quickly to a local optimum although it is not necessarily the minimum of the sum of squares.

The algorithm aims to minimize the objective function

$$f = \sum_{i=1}^{M} \sum_{j=1}^{k} ||x_i^{(j)} - c_j||^2 \tag{2}$$

where $||x_i^{(j)} - c_j||^2$ is a distance measure between feature point x_i^j and a cluster center c_j. The algorithm is simple and follows the steps:

1. select k points into the 128-dimensional space. These points represent initial cluster centroids,
2. assign each feature descriptor to the cluster that has the closest centroid,
3. when all descriptors are assigned, re-compute the positions of the k centroids,
4. repeat steps 2 and 3 until the centroids no longer move.

3.2. The Vocabulary Tree

The vocabulary tree is a hierarchical set of cluster centers, and is used in many cases for recognition in very large database using indexing approach. The object recognition with vocabulary method follows three procedures as described in [15]. Firstly, we organize the descriptors of images in a tree and store inverted files at each node with scores. The inverted files are references to the images containing the instance of the node. Secondly, we generate a score for a given query image based on Term Frequency-Inverse Document Frequency (TF-IDF) and finally we find the images in the database that best matches the score.

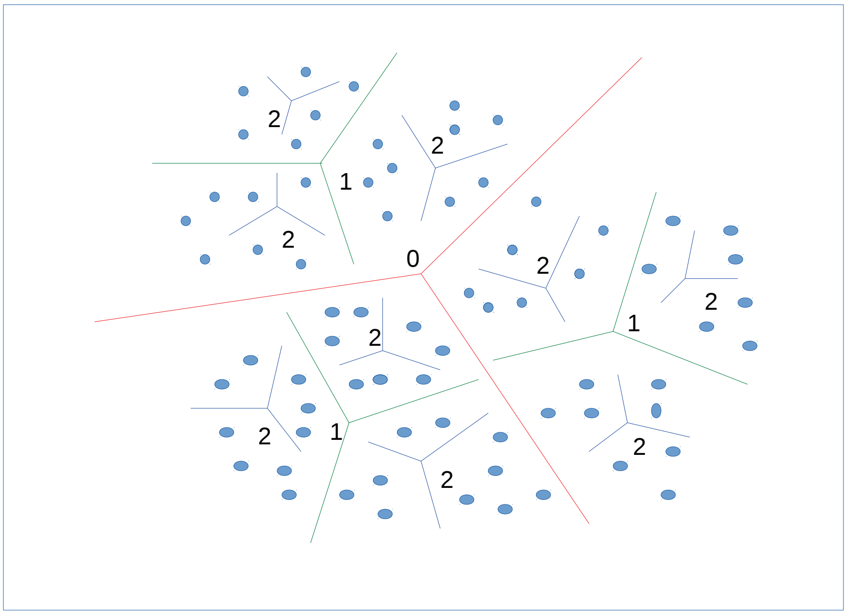

Figure 5. Illustration of feature clustering in a vocabulary tree. The features are hierarchically clustered into three groups (branches). The numbers indicate depth level of the hierarchical clustering, in this example we have a maximum depth level of three and each cluster has three branches.

Training with the Vocabulary Tree

The feature descriptors of all the training images are hierarchically quantized and built into the vocabulary tree of branch k and depth L according to [15]. At the root level, each bag-of-features are initially quantized to k branches. Each branch is then sub-quantized sequentially and repeated until reaching the maximum depth level, as illustrated in Fig. 5 and Fig. 6 . Each node within the tree is linked to its associated features as well as the images. Notice that the k-means clustering is continuously applied at each depth level of the tree to split the parent features and distribute them to k children nodes, for example in Fig. 6 the depth levels indicated by numbers (0, 1, 2), the features at each depth level are split into three branches. We adapt the square root kernel distance metric (Bhattacharyya distance) to measure the similarity between the feature descriptors (as rootSIFT in [27]). The branch factor and depth of the tree influences the retrieval result, however increasing these values does not necessarily improve the retrieval results.

Scoring and Retrieval

The scheme of the vocabulary tree assigns weights to the tree nodes and defines relevance scores associated to images. We assign weights for the nodes of the tree to determine how each feature descriptor (word) from the quantized descriptors (code book) votes for each view. The weighting of the node depends on the number of features assigned to the node and its depth level, i.e. at each node i weight w_i is assigned with entropy weighting scheme:

$$w_i = log(\frac{N}{N_i}) \tag{3}$$

where N is the number of database images and N_i is the number of images with at least one descriptor vector path through node i. Notice that the Equation (3) is the inverse document frequency. This weighting enables a depth-bounded search and is useful to reduce retrieval time. Related to the hierarchical weighting is a flat weighting, where only the leaf nodes are weighted with normalized histogram of features. Further, we define a query q_i and database d_i

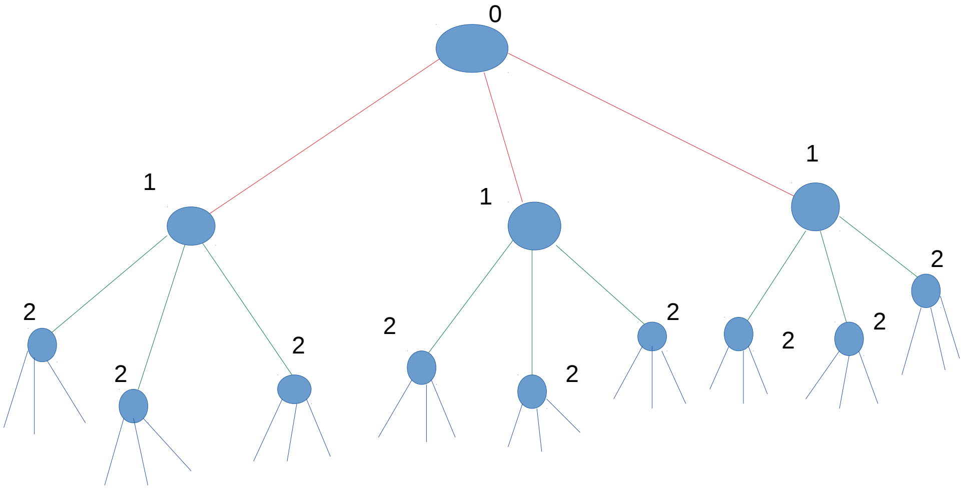

Figure 6. Illustration of a vocabulary tree corresponding to Fig. 5. In this example, the vocabulary tree has two levels and three branches. The clusters of the same level are indicated with edges originating from the same number or of same color. The nodes (cluster centers) of the tree are indicated by numbers, which represent the depth level of the vocabulary tree.

vectors according to assigned weights as

$$q_i = n_i w_i \tag{4}$$
$$d_i = m_i w_i \tag{5}$$

where n_i and m_i are the number of descriptors of the query and database images, respectively with a path through node i. The relevance score s of the database image is given by

$$s(\mathbf{q}, \mathbf{d}) = || \frac{\mathbf{q}}{||\mathbf{q}||} - \frac{\mathbf{d}}{||\mathbf{d}||} || \tag{6}$$

Each database image is given a relevance score according to the Equation (6) and the scores for the images in the database are accumulated.

For the retrieval of the corresponding image, each feature of the query image searches for the nearest node on each depth level, applying the same distance metric used for the training. Thus, for each feature of the query image a path of selected nodes traverses the whole tree from the root node to the leaf node. The weights of the selected nodes and training image provide a similarity measure, which is used for retrieval of the best views for the query image. The best match is retrieved based on the accumulated weights of each training image. We create a histogram with the list of weights of the related nodes to those selected by the bag of features from the training images. Finally images with the highest accumulated weight in the histogram are returned as the best match. In our experiment, we take the histogram of weights as a confidence measure for retrieving the corresponding image.

3.3. Feature Matching and and Pose Estimation

For the on-line testing phase, we extract features of a query image unlike the offline training phase with the Harris detector, and describe with the SIFT descriptor. For each feature descriptor of the query image, we search the corresponding feature descriptor in the the tree. After a successful retrieval of the corresponding image or its descriptors, we match features between query and retrieved image (see Fig. 7). Notice that we employed the Harris feature

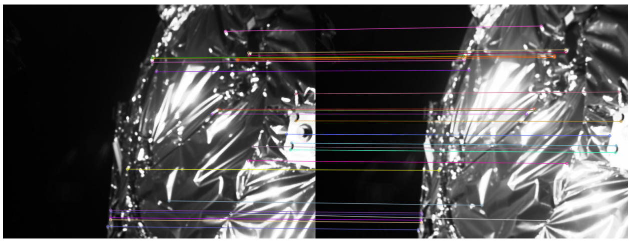

Figure 7. The feature matching for pose estimation. The features of the query image (right) are matched with the retrieved image (left) in the database.

detector for the on-line phase of the image retrieval because of its better computational efficiency and localization accuracy. The scale invariance is achieved by using training images at various radii of the view sphere (scale), thus we expect the scale of the test image is approximately within the scale range of the train images. Therefore, we can avoid the time consuming computation of the scale-invariance of SIFT by pre-computing features at various scales in the training image, slightly similar to [28]. Consequently, we employ the Harris detector for query image while SIFT is used for the description of the region around the features. The Harris detector is faster than SIFT detector, which is essential for a real-time pose estimation. The computational efficiency of the pose estimation can be improved by employing Harris features for both retrieved and query images. However, this speed up is at the expense of memory which requires to store the Harris features of the training images. In contrast, the computational efficiency and storage could be further improved by implementing [28], which is not currently considered in this paper.

Once we have the Harris features and SIFT descriptors of query and retrieved image, we apply descriptor matching using a fast KD tree. Moreover, we compute the 3D points corresponding to the feature points of the training images, by back-projection using the depth map computed with the known pose and the CAD model. Then, given matched 2D features and the 3D points of the retrieved image, we can estimate the pose using a 2D-3D registration method commonly known as a perspective-n-point problem (pnp). We apply an iterative pnp with the RANdom SAmple Consensus (RANSAC), because of the fact that after the matching not all the correspondences are correct. RANSAC is an iterative and non-deterministic method to estimate the parameters of a mathematical model (in our case, translation and rotation parameters) from a set of data points which contains outliers. Therefore, the RANSAC is used to reject the false correspondences (outliers) while estimating the pose.

4. Experiments

Here, the appearance based pose detection method described in Section 3 is evaluated with real image sequences. we evaluate the accuracy of the detection, through the ground truth poses obtained from the measurements of robot kinematics, robot-camera (hand-eye) and camera calibration with the DLR calibration toolbox-CalLab [29]. The experimental setup (Fig. 9) consists of a mock-up of the rear-part of the TerraSAR-X satellite (Fig. 10) in full scale, stereo cameras mounted on six DOF robot and the sun simulator. The rear part of the TerraSAR-X satellite is fully covered with an MLI and consists of launcher interface brackets, which are suitable for grasping. The TerraSAR-X satellite was placed in a fixed position and orientation, and the motion trajectories of the satellite were simulated by the six DOF robot motion. In this experiment, the sun simulator was placed in three main directions for the appearance training phase from the satellite mock-up. For the testing, the sun direction was shifted by approximately 15° from the corresponding sun directions of the training phase shown in Table 1.

Furthermore, we present results of detection on a purely translation motion. In this case, the training and testing images as well as ground truth poses are taken from Croos-Vis data set [18]. The data consists of images taken from various views, lighting directions and camera shutter speed. Here we use this data set to estimate initial position of a launcher interface bracket (LIF) of the TerraSAR satellite (see Fig. 8). We remark that the training images are possible approach trajectories, when a servicer robot approaches attitude controlled client satellite. In an attitude controlled satellite, the orientation of the client is aligned with that of the servicer with help of on-board attitude measurement sensors such as star tracker. Therefore, the camera based method is used for the localization of the position.

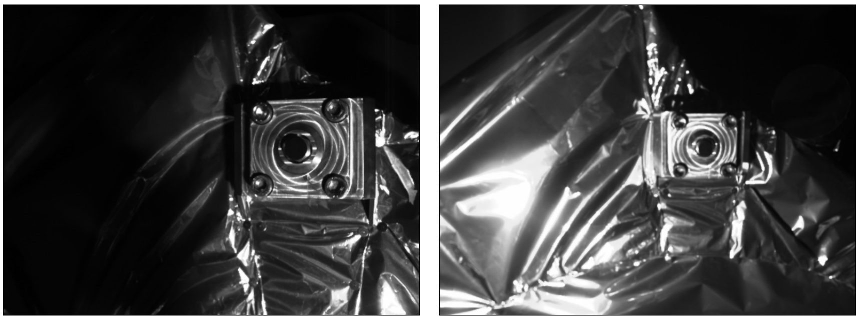

Figure 8. One of the launcher interface brackets (LIF) of the TerraSAR satellite as viewed from different ranges.

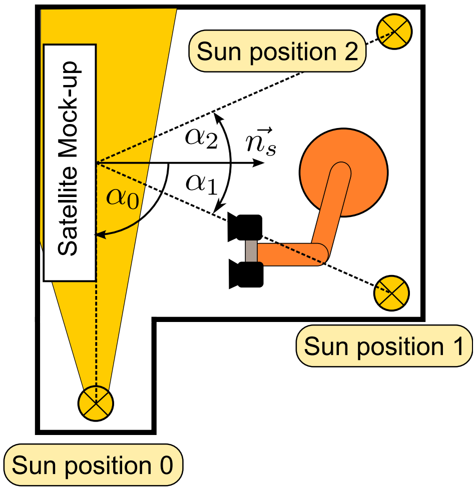

Figure 9. The illustration of the experimental setup: the cameras, the robot, the satellite mock-up and the position of the sunlight simulator relative to the satellite mock-up. The sun incidence angles (α_1, α_2 and α_3) are measured relative to the normal $\vec{n}_s$ of the mock-up as in [18].

We implemented the proposed pose detection algorithm using C++/OpenCV with desktop linux computer 2.8 GHz. The average processing time of the unoptimized on-line pose estimation is $0.8s$. We used the OpenCV implementation of SIFT for the training as well as testing. We have also tested the original David Lowe's implementation of SIFT on our data set; the OpenCV implementation provides as good features as the original implementation by adjusting parameters of the SIFT detector.

A calibrated camera of resolution 780 × 582 pixel and a lens of focal length 6 mm was used to capture images. This choice of camera system has similar characteristics as real space certified cameras and ensures the required field of view within the close range of 1.5 m. During vector quantization through the vocabulary tree, we selected a branch factor (width of the tree) of 6 and a level of 10 empirically. The centroids of the k-means clustering are initialized with three random starting attempts to make the clustering more robust. An alternative to random starting, the cluster centroids could be deterministically initialized according to the ordered views of the images during sampling in a sphere, because of the fact that the training images are sequentially obtained. This preserves order of views and could be used to classify the descriptors roughly to desired clusters (in our experiment $k = 6$). However, the random starting performed equally, thanks to well separation of clusters in a large dimensional (128-D) space.

Table 1. Sunlight directions during training and testing

Training	−90°	−30°	30°
Testing	−75°	−45°	15°

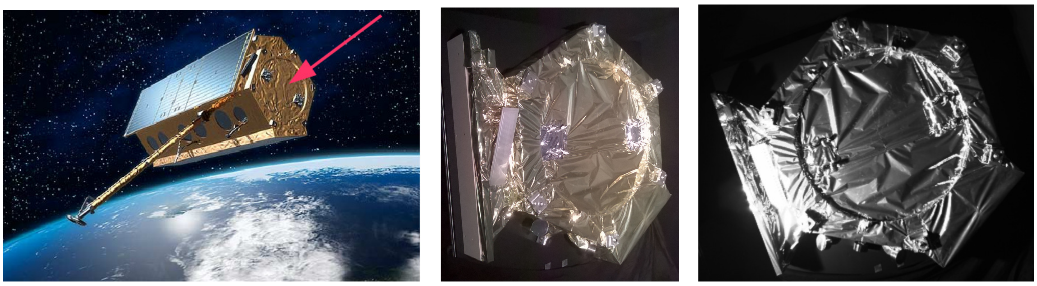

Figure 10. The German Radar satellite TerraSAR-X (left), the full scale mock-up of the rear part of the TerraSAR-X under indoor lighting (middle) and lighting similar to space (right). The arrow indicates the part of the TerraSAR-X satellite used to build the mock-up shown in the middle.

4.1. Evaluation Criteria

The goal of the experiment is to evaluate the pose detection in sequence of images under various lighting conditions. Based on the correct retrieval of the corresponding query image and accurate pose estimation, the detection rate is about 90%, however we are interested to assess the pose error and emphasize on the capability of the detection to initialize a local tracking. Hence, the main evaluation criteria is, the pose detection is said to be successful when the estimated pose is able to initialize the local tracking, i.e the ground truth error of the pose in all axes (optical, lateral and vertical) is less than 100 mm and 2.5°. The detection rate is here defined as the percentage of the ratio of the number of detections that correctly initializes the local tracker to the total number of attempts or images. The detection failure is reported when the retrieval score is low (<45) or the number of matched features are below the minimum (>4). We have determined the threshold of the retrieval score (<45) empirically, while considering false positives and negatives. The database consists of several images which appear to be similar to query image and provide apparently higher score during retrieval. When we use a higher value than the determined threshold, many of actual corresponding images are rejected (false negative). On the other hand, the retrieval score below the threshold (<45) results in several false positives during retrieval. According to this evaluation criterion, the detection rate of various tests is presented in Table 3. In the following sections, we present pose detection results on roto-translation motion (Section 4.2) and pure translation (Section 4.3).

4.2. Roto-translation motion

In this section, we present the detection result for motion comprising of both rotation and translation. Also we present the effects of scale change and illumination on detection of the client satellite undergoing roto-translation motion. In order to show the correct alignment of the model with the image at estimated pose, we provide the qualitative results of pose estimation in Fig. 11; the overlay of query image as well as keypoints (shown as circles) corresponding to the feature descriptors of the query image (left) and the re-projection of the 3D model points on to the image plane at the estimated pose indicate the accuracy of the pose estimation. The alignment of the query image and re-projected model are shown on the right.

Effect of scale: The training images are taken in hemisphere of radii 0.5 m, 0.75 m and 1.2 m. This training ensures that the pose of the satellite can be accurately estimated in spite of change of scale. Here we present the pose estimation evaluated at two scales; the client satellite placed 1.3 m and 0.7 m along the optical axis of the camera with the same (-45°) sun direction. This performance evaluation enables to assess the effect of a scale on the performance of the pose detection algorithm. As we can observe from Fig. 13 and Fig. 14, most errors are around the means (0.8° and 24 mm) at both distances. Therefore, the estimated pose can be obviously used to initialize a model-based tracking irrespective of the scale change. The distribution of the error does not degenerate gracefully because of the significant illumination change at various view points (see Fig. 12). This is particularly eminent for highly specular object such as a satellite. The larger errors (spikes) in the plot indicate that pose detection in certain camera frames tend to fail. The pose estimation in this case results in a large error due to an incorrectly retrieved image features in the database and insufficient valid feature correspondences. The number of features detected and the matching features found are shown in Table 2.

Effect of the sun direction: The accuracy of pose estimation of a highly specular satellite depends on the view points (Fig. 12) as shown in Fig. 13, and the direction of illumination. We conducted an experiment to study the

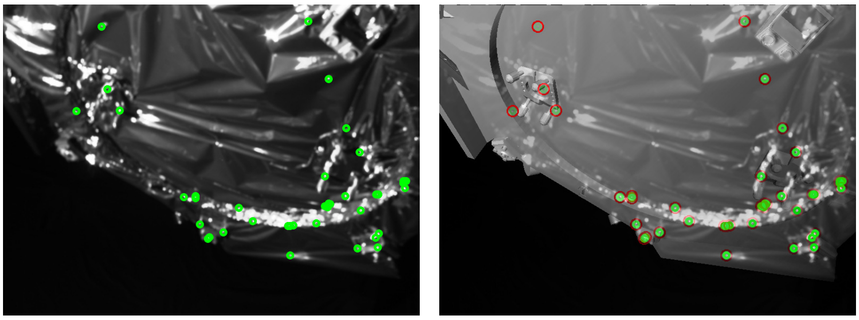

Figure 11. Quantitative results of pose estimation: the overlay of a query image and the corresponding view re-projected onto the image plane using the 3D model at the estimated pose (right). The keypoints used in pose estimation are shown by circles on both query (left) and overlay images (right).

Table 2. Number of features extracted from some test images and the number of features matches. In spite of hundreds of detected features, the matched features are quite a few because of the specular reflection.

frame	1	7	20	38	59	64	81	1 37	145
detected features	433	573	560	545	345	499	690	497	392
matched features	22	2	5	62	60	2	97	29	9

sensitivity of the pose detection to the direction of the sun; we present the pose estimation error associated to three illumination directions -75°, -45° and 15° (see Table 1). Here we have the same trajectory, only the sun position with respect to the camera line of sight is changed. The effect of the illumination on rotation and translation estimation at a sun angle of -45° can be observed respectively in Fig. 13 and Fig. 14. The error at some frames are larger because of the variation of the lighting among the frames and the effect of specular reflection. At both scales, the pose estimation error follows similar distribution (outliers due to specular reflection). Notice that the pose (rotation and translation) estimation errors are separately plotted for the sake of readability as well as comparison with respect to illumination direction. Accordingly, the rotation error plots on the left column of Fig. 13 correspond to the translation on the left column of Fig. 14, the rotation on the left column of Fig. 15 corresponds to the left column of the translation in Fig. 16, and similarly the corresponding rotation to the translation are shown on the right columns of the figures.

When the sun direction is about -75° with respect to the line of sight of the camera, the distribution of the pose detection error (Fig. 15 and Fig. 16) is similar to the above (Fig. 13 and Fig. 14) in spite of change of illumination. The noticeable difference among the pose error due to illumination direction is the the location of outliers. The outliers at a sun angle of 15° could be inliers at a sun angle of -75°, for example see frame 45 in left and right columns of Fig. 15. This is due to the fact that the specular reflection depends on the direction of light and camera view point. In the case of the same view point, the direction of the sun significantly changes the location of the reflection. In spite of the change in location of the outliers because of the sun angle, the error distribution is very similar as shown in Fig. 15 and Fig. 16. This is because of the training of the appearance at the neighborhood sun direction (-90° and -30°), so that the influence of illumination change is reduced. However, it is necessary to train the illumination at a

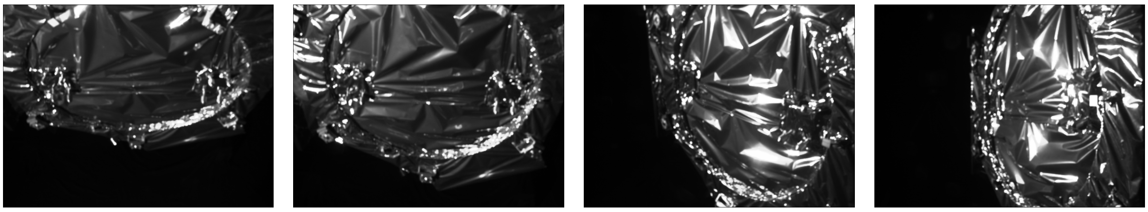

Figure 12. Some test images with the sun direction of $-45°$. The significant illumination change occurs due to the view change in camera pose, and results in non-graceful error distribution as in Fig. 13.

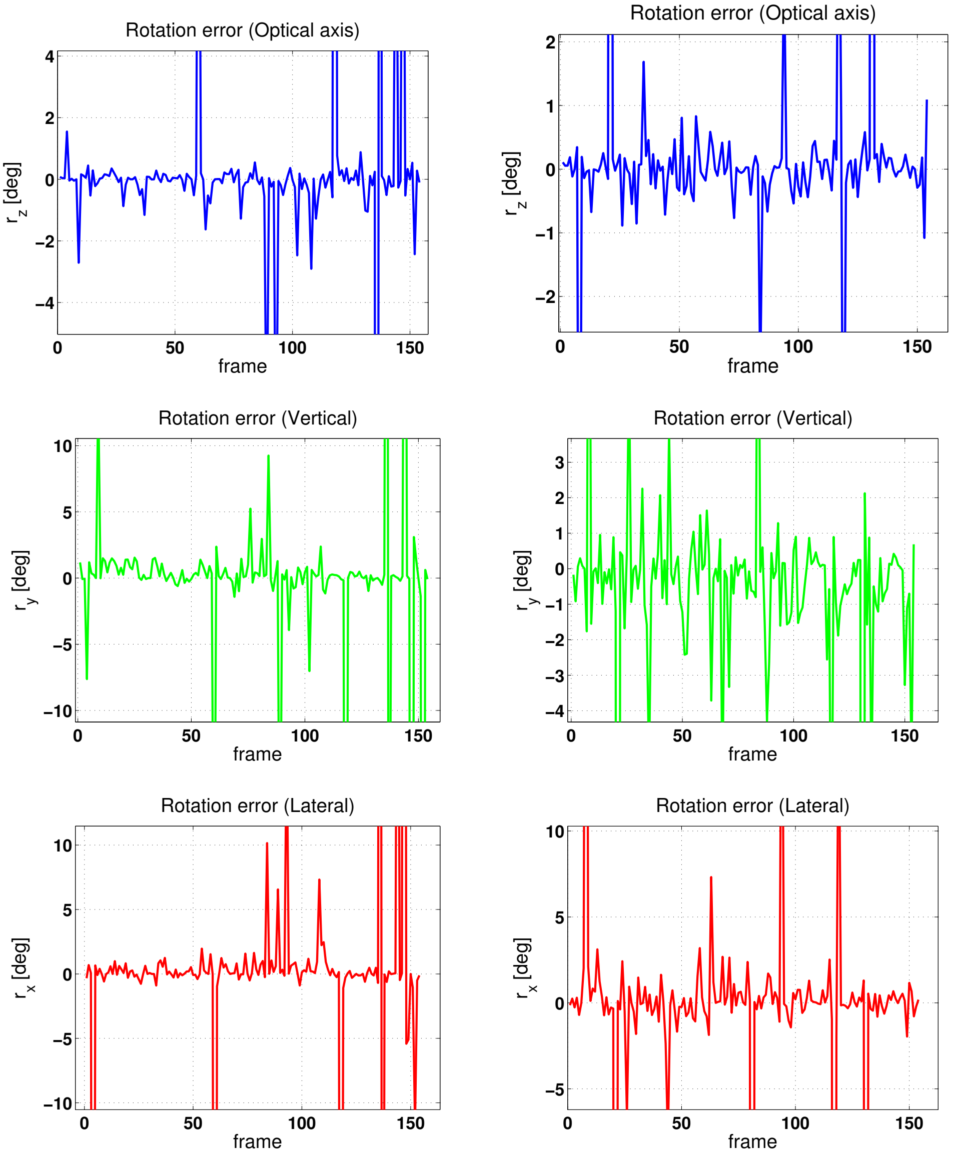

Figure 13. The pose estimation error: the ground truth rotation error by X,Y and Z angles as evaluated from a distance of 1.3 m (left column) and 0.7 m (right column). The pose detection algorithm is able to detect the correct views and estimate rotation at different scales. This is because, the training data consists of images at various distances.

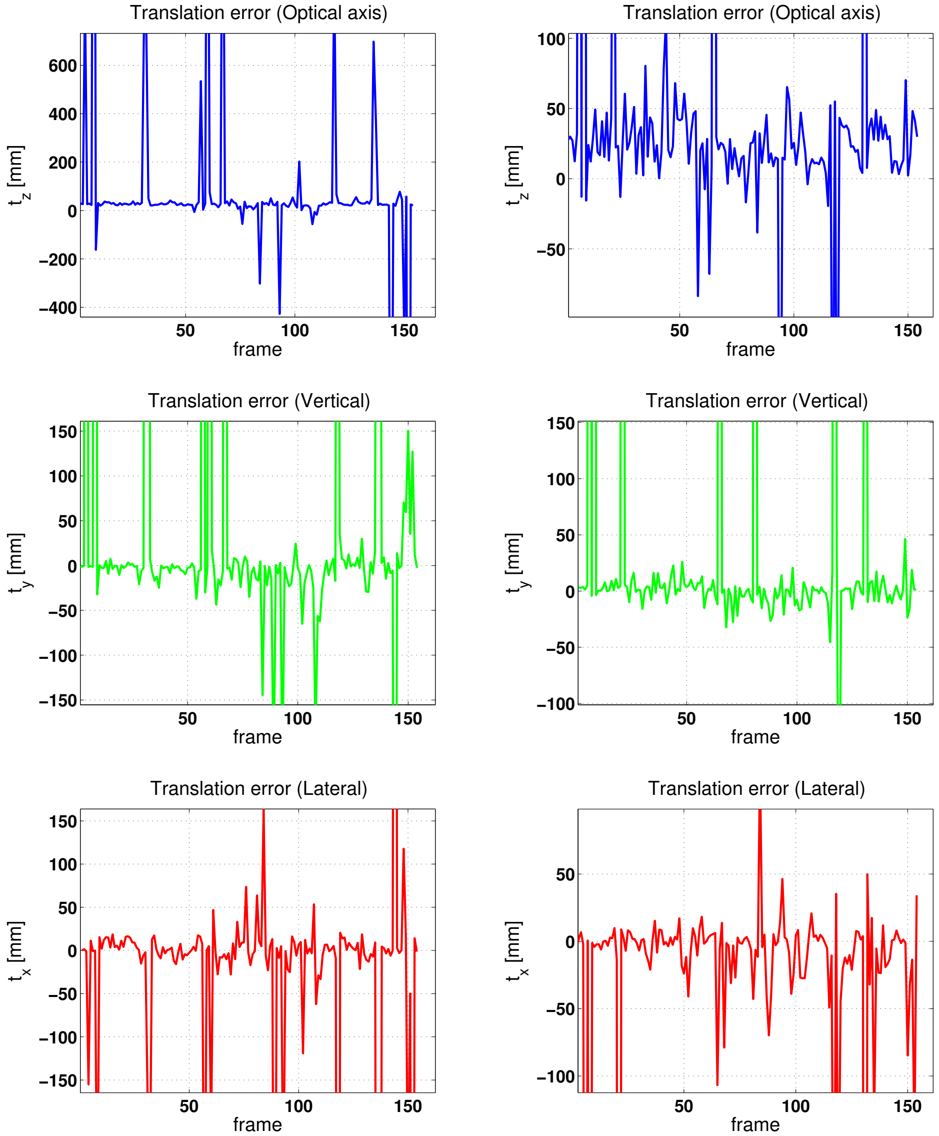

Figure 14. The pose estimation error: the ground truth translation error in X, Y and Z coordinates as evaluated from a distance of 1.3 m (left column) and 0.7 m (right column). The algorithm is able to estimate position in spite of changes of scale due to camera motion along the optical axis (depth). The larger errors (outliers) are due to incorrectly retrieved image features and insufficient valid feature correspondences.

sufficient interval. For example, in our experiment the pose detection fails completely when the test sequence was at the direction of -60°, i.e. the nearest training direction to the test sequence was -90° and -30° (for position of sun directions, see Fig. 9).

In general, sampling the sun direction in 3D space at an interval of 15° results in the desired accuracy in pose detection. However, such fine sampling creates hundreds of training sun directions, which may be unacceptable in terms of memory requirement to store respective features. Moreover, it increases the query time during retrieval. In practice however, the fine sampling the full sun directions is not necessary since the range of the sun direction will be known during a particular on-orbit servicing mission. Therefore, we restrict the possible directions of sun in a certain range provided by the on-orbit servicing mission. This in turn reduces a memory requirement for storing database features and query time.

Table 3. Detection rate at different sunlight directions. A pose error (r and t) smaller than 2.5° and 100 mm in all axes (x,y and z) indicate a successful detection.

Pose parameters	r_x	r_y	r_z	t_x	t_y	t_z
Detection rate [%] at sun angle $-75°$ and from a distance of 1.30 m	95.45	90.90	95.45	95.45	94.15	94.15
Detection rate [%] at sun angle $-45°$ and from a distance of 1.30 m	90.90	89.61	93.50	88.96	88.96	88.96
Detection rate [%] at sun angle 15° and from a distance of 0.70 m	90.26	88.96	95.45	93.50	94.80	94.15

4.3. Pure translation motion

We evaluate the pose detection method described in Section 3 with image sequences of purely translation motion. This experiment is more difficult for pose estimation, because of more parameter (camera shutter speed) introduced when collecting train and test images in addition to the change of sun angle. In fact, we have less parameters (3 DOF) to estimate which accelerates the speed of computation although the lighting condition is challenging. Moreover, we do not have scale problem since the train trajectories span the translation along the camera axis from 2 m to 0.5 m during the approach, therefore all the possible pre-planned trajectories can be used for the training. In this experiment, the camera view is centered at 5 locations on the launcher interface bracket with three lighting directions and camera shutter speed as described in [18]. The test images are different from the training images due to the camera shutter speed which varies the amount of light entering to the camera. The database consists of image features and descriptors from 2590 images recorded with shutter speeds 3 ms, 10 ms and 70ms. This experimental setting is used to evaluate the pose detection method in the presence of strong variation of intensity of light.

Accordingly, we conducted tests on three approach trajectories with shutter speed settings 5 ms, 30 ms and 40 ms. The camera shutter speed influences the pose detection accuracy significantly, and it is important parameter of camera setting to control amount of light entering to the camera sensor. We discuss the effect of lighting variation due to higher camera shutter time (30 ms and 40 ms, i.e. over-illuminated target) and lower shutter time (5 ms, i.e. under-illuminated target). Notice that such change of lighting can be also achieved approximately by varying the direction of the light source (the Sun) with respect to the camera line of sight. We observe the effect of change of intensity because of shutter time 40 ms in Fig. 17 where the position error along each-axis and corresponding sample images are shown. In this case the outliers in Fig. 17 are because of the variation of light as the result of the camera shutter change. Thus, some of the keypoints in the database image are no more keypoints in the test, resulting feature matching error. Similarly, we evaluate the pose detection with a shutter speed of 30 ms. In this shutter speed setting, some images are shown in Fig. 18. The number of outliers in pose detection reduced significantly in Fi.g 18 because of the reduced shutter speed, consequently less saturated images. On the other hand, when the intensity of light is significantly reduced with a shutter speed of 5 ms, the images become under-saturated (see Fig. 19). However, the pose estimation error and outliers did not increase. This is because the training images contain saturated images taken with a shutter speed of 3 ms.

We have demonstrated that the pose detection in Section 4.2 depends highly on the sun direction. Similarly, the amount of light hitting the surface of the target satellite which is controlled by camera shutter, determines the accuracy of pose detection as shown in Fig. 17, 18 and 19. The experiment is used to identify the suitable shutter speed for a given sun direction. The experimental results indicate that the higher shutter time of the camera benefits the pose detection algorithm when the sun angle approaches 90° to the line of the sight of camera. On the other hand, when the

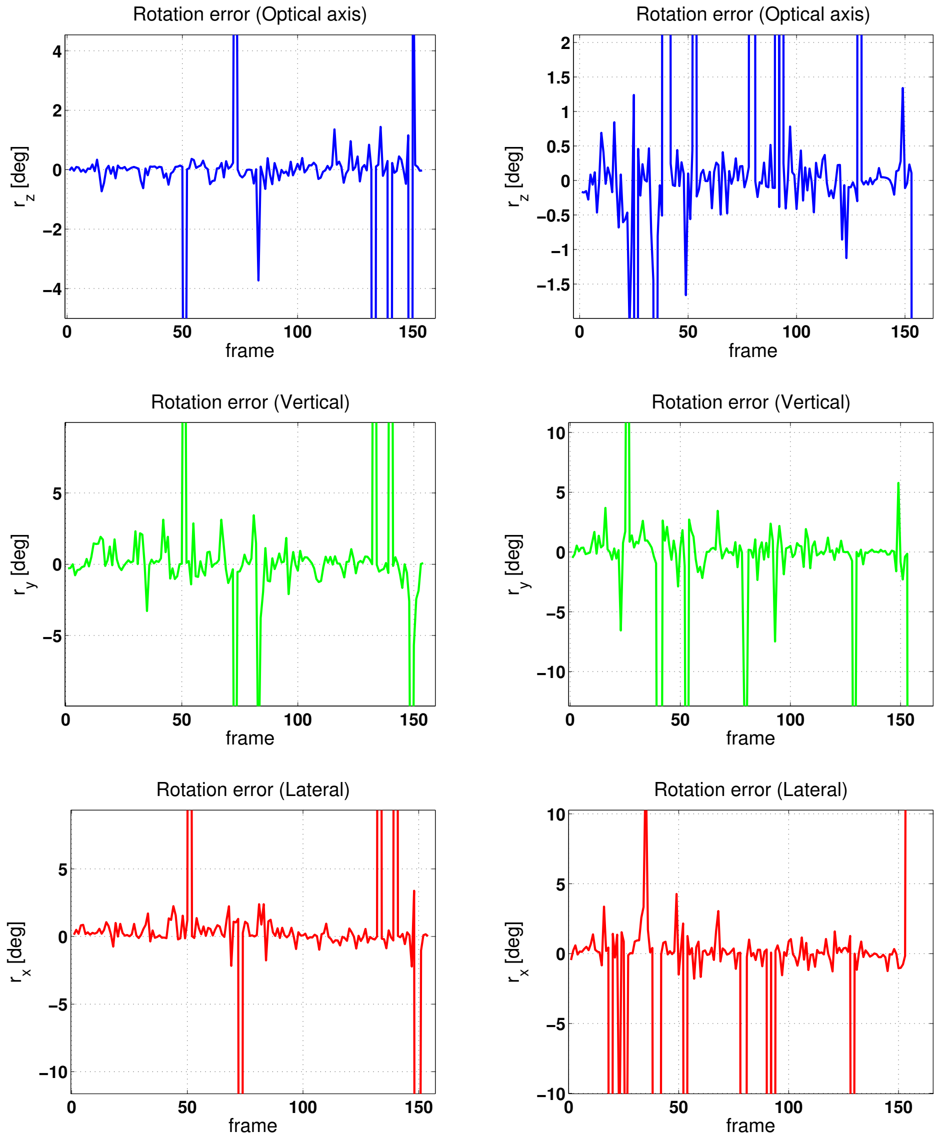

Figure 15. The accuracy of the rotation estimate compared to the ground truth rotation obtained from the robot measurements. The Figure indicates the ground truth rotation error at the **sun direction of -**75° **(left column) and** 15° **(right column)** with respect to camera line of sight. The outliers (high peaks in the plot) are due to the inaccurate feature matching and insufficient number of features.

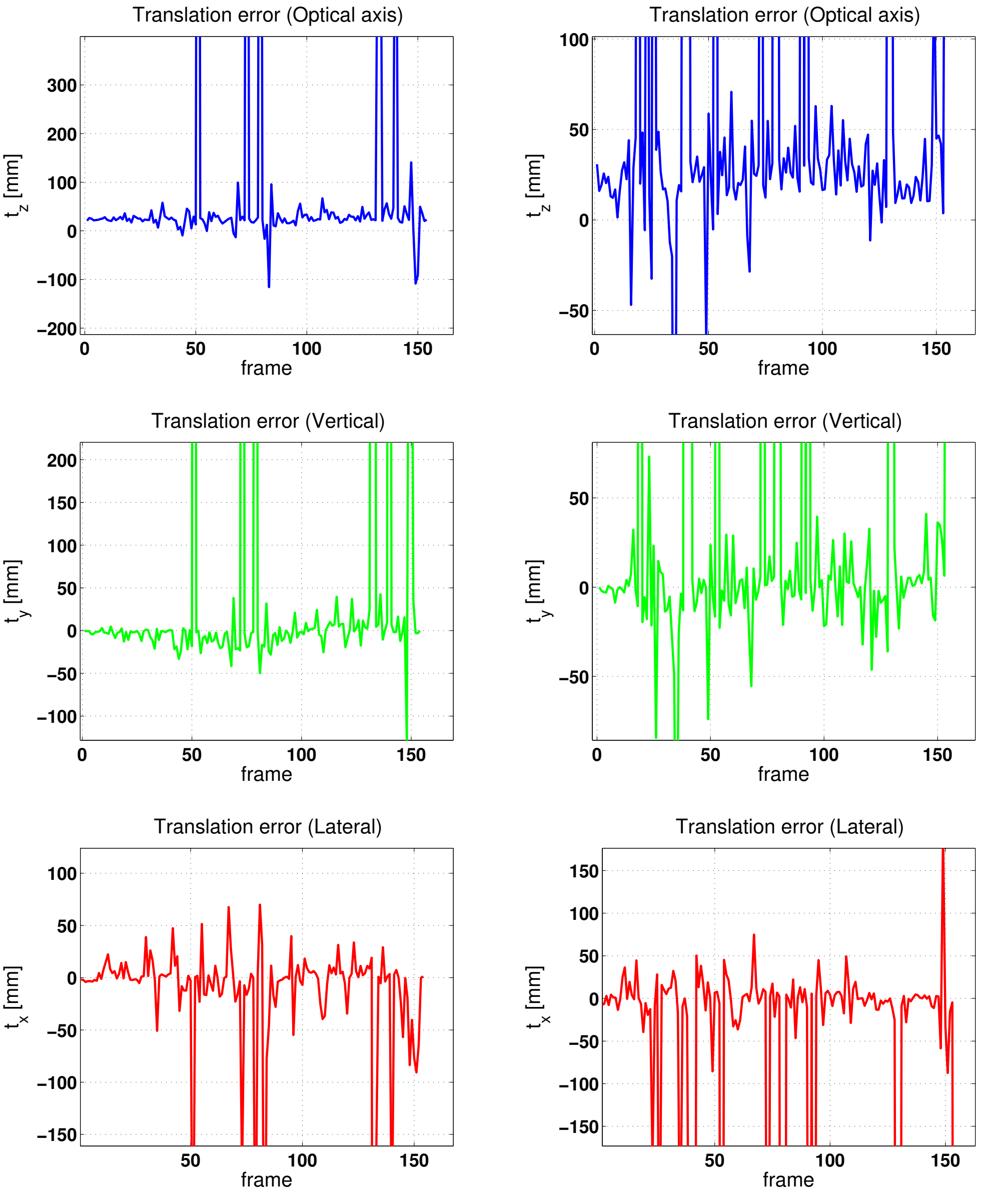

Figure 16. The accuracy of the position estimate compared to the ground truth translation obtained from the robot measurements. The Figure indicates the ground truth translation error at the **sun direction of** -75° **(left column) and** 15° **(right column)** with respect to the camera line of sight. Similar to rotation in Fig. 15, the outliers (high peaks in the plot) are due to the inaccurate feature matching and insufficient number of features.

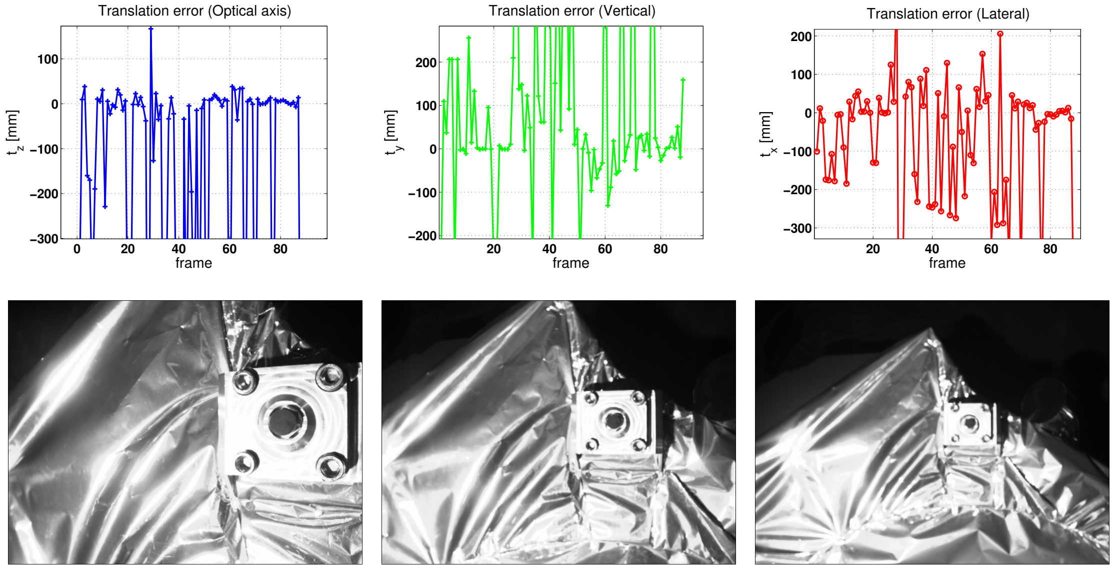

Figure 17. The position error of the launcher interface bracket with a camera shutter time of 40ms. The translation errors in X, Y and Z coordinates (first row) are computed based on a ground truth obtained from robot measurement, hand-eye and camera calibration. Exemplary images (second row) taken from the closest, middle and furthest camera view of the test images correspond to the above plots.

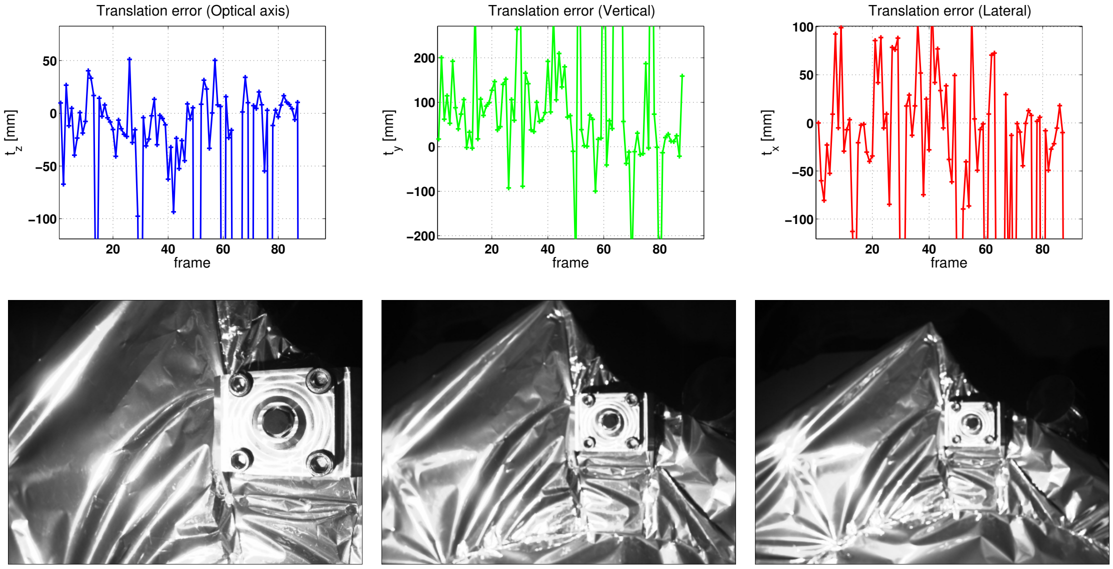

Figure 18. The position error in X, Y and Z coordinates of the launcher interface bracket with a camera shutter time of 30ms (first row). The exemplary images (second row) are less saturated than in Fig. 17, therefore resulting in fewer outliers.

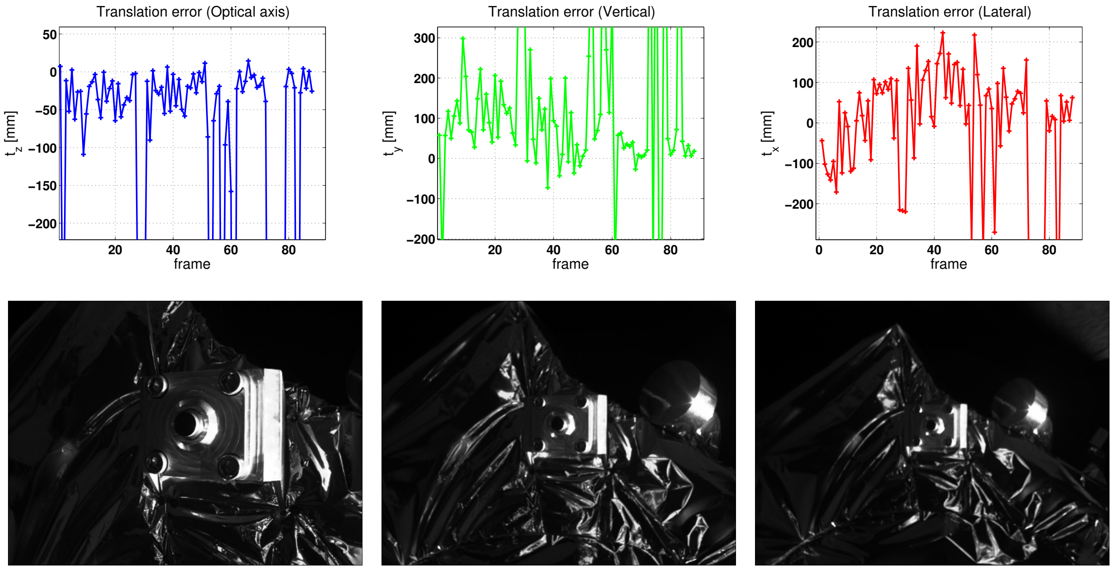

Figure 19. The position error of the launcher interface bracket with a camera shutter time of $5ms$. The ground truth translation error (first row) in X, Y and Z coordinates and corresponding exemplary images (second row). A short shutter time results in under-saturated images, however the position estimation accuracy could not deteriorate because of similar training images with a shutter time of $3ms$ in the database.

sun angle approaches 0° to the line of the sight of the camera, the lower shutter time is the most suitable for accurate pose detection.

5. Conclusion

An object detection in 3D space is generally a challenging problem in computer vision and image processing. The problem of pose detection is more difficult under space lighting and with reflective surface. The geometry of the object also determines the performance of the detection method. On one hand, the pose of an object with several distinctive edges can be estimated with edge-based methods. On the other hand, feature-based methods are used to estimate the pose of textured objects. However, in certain cases the region of interest of the object such as a satellite may consist of a highly reflective multilayer insulation (MLI) with several wrinkles which dominate useful features such as edges. The MLI poses difficulty in edge-based detection because of reflection and virtual unmodeled edges. In this paper, we address the pose detection of a highly specular object such as the TerraSAR-X satellite in close-range.

The pose detection is based on an appearance learning of a full scale mock-up of the rear part of the TerraSAR-X satellite. We use several images under various view points and sun direction for the training with a vocabulary tree. The vocabulary tree is effectively used to represent the training images by quantizing respective feature descriptors with the hierarchical K-means clustering. The training images are taken from a camera mounted on 6 DOF robot. We employ a 3D image warping to synthesize missing images because of the limited workspace of the robot. After successful retrieval of the corresponding image to the query image, we match feature descriptors with the fast KD tree. The 3D points corresponding to the training features are computed using the depth map, which is in turn obtained by rendering the 3D model with the Z-buffer. The pose is estimated from the correspondence based on iterative 2D-3D registration and RANSAC.

We validate the pose detection method with challenging space lighting conditions. The evaluation criteria is based on the accuracy of the estimated poses with respect to the ground truth poses and a capability to initialize a local tracking. The pose detection, evaluated with several lighting condition, is accurate and able to initialize local tracking. However, we observe that as the disparity between the training and testing sun directions deviates significantly from 15°, the accuracy and robustness of pose detection decreases. The drawback of this method is the necessity of the mock-up of the satellite for training appearances, which could be replaced with synthetic photo-realistic images from the model of the satellite. However, rendering a photo-realistic images of a multilayer insulation is difficult and requires further research. In spite of the difficulties of the modeling of MLI, we are interested to use synthetic photo-realistic images rendered from the model of the satellite in the future.

Acknowledgement

The Author would like to thank Prof. Daniel Cremers and Dr. Giorgio Panin who provided insights, and Martin Lingenauber and Michael Kassecker for their support during conducting the experiment.

References

[1] N. W. Oumer, G. Panin, Q. Mühlbauer, A. Tseneklidou, Vision-based localization for on-orbit servicing of a partially cooperative satellite, Journal of Acta Astronautica 117 (2015) 679–698.

[2] A. Petit, E. Marchand, K. Kanani, A robust model-based tracker combining geometrical and color edge information, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp. 3719–3724.

[3] N. W. Oumer, G. Panin, 3d point tracking and pose estimation of a space object using stereo images, in: Proceedings of 21st International Conference on Pattern Recognition, 2012, pp. 796–800.

[4] C. M. Cy, B. B. Kimia, 3d object recognition using shape similiarity-based aspect graph, in: Proceedings of 8th IEEE International Conference on Computer Vision, Vol. 1, 2001, pp. 254–261.

[5] M. Ulrich, C. Wiedemann, C. Steger, Cad-based recognition of 3d objects in monocular images, in: Proceedings of IEEE International Conference on Robotics and Automation, 2009, pp. 1191–1198.

[6] C. Reinbacher, M. Rüther, H. Bischof, Pose estimation of known objects by efficient silhouette matching, in: Proceedings of International Conference on Pattern Recognition, 2009, pp. 1080–1083.

[7] A. Petit, E. Marchand, R. Sekkal, K. Kanani, 3d object pose detection using foreground/background segmentation, in: Proceedings of IEEE International Conference on Robotics and Automation, Seattle, WA, 2015, pp. 1858 –1865.

[8] S. Hinterstoisser, S. Benhimane, N. Navab, N3m: Natural 3d markers for real-time object detection and pose estimation, in: Proceedings of 11th International Conference on Computer Vision, 2007, pp. 1–7.

[9] D. G. Lowe, Three-dimensional object recognition from single two-dimensional images, Journal of Artificial Intelligence 21 (3) (1987) 335–395.

[10] M. S. Costa, L. G. Shapiro, 3d object recognition and pose with relational indexing, Journal of Computer Vision and Image Understanding 79 (3) (2000) 364–407.

[11] P. David, D. DeMenthon, Simultaneous pose and correspondence determination using line features, in: Proceedings of IEEE Computer Vision and Pattern Recognition, 2003, pp. 424–431.

[12] P. David, D. DeMenthon, Object recognition in high clutter images using line features, in: Proceedings of 10th IEEE International Conference on Computer Vision, 2005, pp. 1581–1588.

[13] V. Lepetit, P. Fua, Keypoint recognition using randomized trees, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (9) (2006) 1465–1479.

[14] V. Lepetit, J. Pilet, P. Fua, Keypoint recognition using randomized trees, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, Vol. 2, 2004, pp. II–244 – II–250.

[15] D. Nister, H. Stewenius, Scalable recognition with a vocabulary tree, in: Proceedings of IEEE Computer Vision and Pattern Recognition, 2006, pp. 2161–2168.

[16] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, R. Szeliski, Building rome in a day, in: Proceedings of IEEE 12th International Conference on Computer Vision, 2009, pp. 72–79.

[17] A. Appel, Some techniques for shading machine renderings of solids, in: Proceedings of American Federation of Information Processing Societies (AFIPS) Conference 32, 1968, pp. 37–45.

[18] M. Lingenauber, S. Kriegel, M. Kaßecker, G. Panin, A dataset to support and benchmark cvision development for close range on-orbit servicing, in: 13th Symposium on Advanced Space Technologies in Robotics and Automation ASTRA, 2015.

[19] H.-Y. Shum, S.-C. Chan, S. B. Kang, Image-based rendering, Springer, 2007.

[20] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 2 (60) (2004) 91–110.

[21] H. Bay, A. Ess, T. Tuytelaars, L. V. Gool, Speeded-up robust features (SURF), Journal of Computer Vision and Image Understanding 110 (3) (2008) 346–359.

[22] M. Calonder, V. Lepetit, C. Strecha, P. Fua, Brief: Binary robust independent elementary features, in: 11th European Conference on Computer Vision, 2010, pp. 778–792.

[23] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, Orb: An efficient alternative to sift or surf, in: Proceedings of IEEE International Conference on Computer Vision, 2011, pp. 2564–2571.

[24] S. Leutenegger, M. Chli, R. Y. Siegwart, BRISK: binary robust invariant scalable keypoints, in: Proceedings of IEEE International Conference on Computer Vision, 2011, pp. 2548–2555.

[25] N. Khan, B. McCane, S. Mills, Better than sift?, Machine Vision and Applications 26 (6) (2015) 819–836.

[26] J. B. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1, 1967, pp. 281–297.

[27] R. Arandjelovic, A. Zisserman, Three things everyone should know to improve object retrieval, in: Proceedings of IEEE Computer Vision and Pattern Recognition, 2012, pp. 2911–2918.

[28] P. Azad, T. ASfour, R. Dillmann, Combining harris interest points and the sift descriptor for fast scale-invariant object recognition, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009, pp. 4275–4280.

[29] K. H. Strobl, G. Hirzinger, Optimal hand-eye calibration, in: Proceedings of IEEE International Conference on Intelligent Robots and Systems, 2006, pp. 4647–4653.