OXFORD BROOKES UNIVERSITY

Department of Computing and Communication Technologies

MASTER OF SCIENCE DISSERTATION

Dissertation Title:	Vision Algorithms for Noncooperative Rendezvous of Objects in Space
Surname:	Marti-Rocafull
First Name:	Paula
Student Number:	13043032
Supervisor:	Fabio Cuzzolin
Dissertation Module:	P00998
Course Title:	Computer Vision
Date Submitted:	25/09/2015

Statement of originality

Except for those parts in which it is explicitly stated to the contrary, this project is my own work. It has not been submitted for any degree at this or any other academic or professional institution.

....
Signature of Author

.........25/09/2015..........
Date

Regulations Governing the Deposit and Use of Master of Science Dissertations in the Department of Computing and Communication Technologies, Oxford Brookes University.

1. The "top" copies of dissertation final reports submitted in fulfilment of Master of Science course requirements shall normally be kept by the Department.

2. The author shall sign a declaration agreeing that, at the supervisor's discretion, the dissertation may be submitted in electronic form to any plagiarism checking service or tool.

3. The author shall sign a declaration agreeing that the dissertation be available for reading and copying in any form at the discretion of either the dissertation supervisor or in their absence the Head of Postgraduate Programmes, in accordance with 5 below.

4. The project supervisor shall safeguard the interests of the author by requiring persons who consult the dissertation to sign a declaration acknowledging the author's copyright.

5. Permission for anyone other than the author to reproduce in any form or photocopy any part of the dissertation must be obtained from the project supervisor, or in their absence the head of Postgraduate Programmes, who will give his/her permission for such reproduction only to the extent which he/she considers to be fair and reasonable.

I agree that this dissertation may be submitted in electronic form to any plagiarism checking service or tool at the discretion of my project supervisor in accordance with regulation 2 above.

I agree that this dissertation may be available for reading and photocopying at the discretion of my project supervisor or the Head of Postgraduate Programmes in accordance with regulation 5 above.

....
Signature of Author

.........25/09/2015..........
Date

Vision Algorithms for Noncooperative Rendezvous of Objects in Space

Paula Marti
Deimos Space UK Ltd.
Harwell, Oxfordshire, UK
paula.marti@gmail.com

***Abstract*— The term "rendezvous of objects in space" refers to the action of a spacecraft approaching another object travelling in space. As part of this maneuver, it is often important that the chaser is able to detect its relative position with respect to the target body. This dissertation explores vision algorithms for un-cooperative rendezvous in which the chaser uses a single camera and a known 3D computer model of the target body to recognize the distance and the pose of the target that is approaching. A database of reference poses or views of the 3D computer model is created and feature descriptors are used to characterize them. A mock-up physical model of the same target is used to produce sequences of images simulating a chaser approaching the target. The system detects the target, selects a matching pose from the database of poses and uses previous knowledge from the sequence of images to correct classification errors. Using a reference database with views of the 3D computer model taken every 10º about all axes, the system is able to return the expected pose with up to 78% accuracy. In the best case, the mean error, calculated as the distance between the expected pose and the classified pose in degrees, is of only 1.5º. Poses within an error of 10º or less from the expected pose are selected up to 96% of the cases. The results returned by the developed algorithms are a good approximation of the real pose and the algorithms themselves are simple and fast. If higher accuracy is needed then the selected pose could be used as an initial input to feed more complex algorithms that could refine it.**

Keywords—HOG; template matching; pose detection; rendezvous

I. INTRODUCTION

A. Background

The term "rendezvous of objects in space" refers to the action of a spacecraft approaching another object travelling in space. An example could be an automated pod carrying supplies to the International Space Station (ISS), which needs to maneuver towards the ISS and perform docking. This kind of docking is considered as a "cooperative" rendezvous, since the ISS has some markers to signal the target and aid docking while, due to the high level of accuracy required, some manual operations are performed as well. There are, however, some applications for which cooperation is not possible. These include: removing a piece of debris from an orbit, capturing a decommissioned satellite or approaching an asteroid. A spacecraft chasing these objects will need to be able to measure distance and pose of the uncooperative target so that action can be taken when appropriate. Even though sensors and techniques such as LiDAR (Light Detection and Ranging) might seem more accurate for measuring shape and distance of target objects, these radar techniques use up much power, which is a scarce resource in space. For this reason, this dissertation will look into the use of a single camera to estimate the relative pose and distance to an object travelling in space.

The detection of an object pose and distance is a common problem in applications involving robots performing a task such as grabbing an item or avoiding an obstacle. This is a typical example of an application of computer vision. Although there are many papers on the topic of recognition of objects, not many of them are applied to space objects and therefore this dissertation might be considered to be of certain research significance.

The final goal is to estimate the relative position and orientation (pose) of the body (target) in relation to the spacecraft (chaser) using images from an optical on-board camera. The approach proposed here is for the chaser to use a known 3D model of the target body, so that shape or template matching techniques can be applied and to explore.

B. Previous work

The algorithms for these maneuvers are often proprietary to a company and therefore they are not publicly available. Even so, some papers have been published regarding vision systems for rendezvous such as the series of articles from Petit, Marchand and Kanani [1] [2] [3]. They also use a 3D model of the target spacecraft for the initialization phase in which the closest view from the 3D model is selected to match the current pose of the target, although the techniques used are not described. Then a tracking phase follows in which the consecutive differences between the previous image pose and the current image pose are calculated and corrected. For this, 3D computer model is needed and stored on-board. The approach followed in this paper does not require storing the 3D model but only feature descriptors, which save computation power. Some research was also commenced by Elecnor Deimos for Guidance Navigation and Control (GNC) to approach a spherical target orbiting Mars [4].

Outside the space applications domain, the study of techniques for pose recognition using 3D template matching is quite active. There are many analogous applications such as localisation of vehicles using a monocular camera [5] or tracking vehicles for autonomous convoy driving [6]. The review of literature of objects based on their shape and boundaries shows that the Histogram of Gradients (HOG) is one of the better performant algorithms. Used for the first time in 2005 [7] to detect pedestrians, the use of this descriptor has been extended to many different applications, from matching simple sketches to photographs [8] to autonomous driving systems for cars [9]. HOG is used for hand gesture recognition in [10]. Although what they want is to be able to detect a gesture even when looking at the hand from different angles, instead of determine the angle and pose, the article highlights the fact that HOG is sensitive to object rotation which is beneficial to our application. Also, their algorithm is run on a mobile device, which shows that very little processing power is needed, which is a key requirement for satellite-based applications.

II. Development approach

A. Methodology

Fig.1 shows a block diagram of the functionality implemented. The inputs are the picture of the target and information from the 3D shape of the target which is extracted from a 3D computer model. The main output should be the distance from the chaser to the target and the pose of the target.

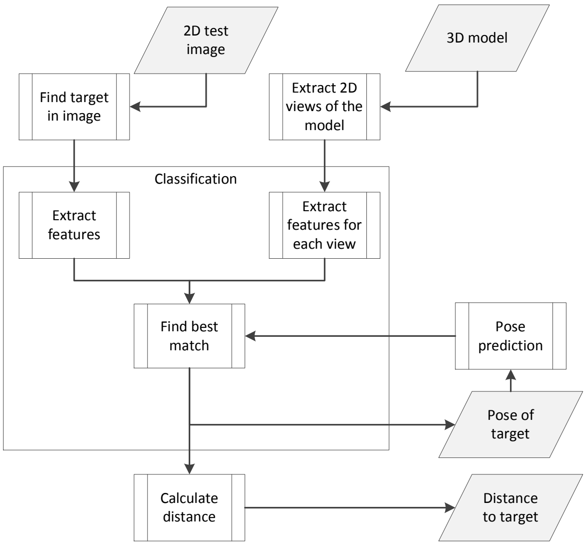

Fig. 1. Block diagram of the system

The problem can be described as matching the pose of the target in the picture with its corresponding view of the 3D computer model. This will be done by generating a database of reference views from the 3D computer model and by trying to find the reference view that best matches the target's pose.

B. Input data generation

We used a 3D computer model of the Deimos-2 satellite and a matching physical scale model of about 20cm, both kindly provided by Elecnor Deimos, to generate the input data.

An application called SatelliteProfiler was developed using the Unity framework. This application allows the generation of a database of reference views of the 3D computer model from different angles by turning the satellite around the three main axes using the specified angle increment and saving the views as 2D images. Data was produced and stored by the SatelliteProfiler with increments of 10º, which meant a database of reference images of more than 23,500 views.

Photos of the scale model were taken from different angles in a simulated black background environment and used as test data for the algorithms. They were divided into two datasets: (1) A collection of 40 images from different angles in order to capture a wide variety of poses and (2) 3 sequences of images approaching the model, where distance varies from far to close and pose varies progressively.

C. System design

The implementation of the full processing chain as shown in Fig.1 requires the implementation and integration of several building blocks, or components. The main components are the preparation of the input data, finding the target in the image, the classification algorithm to find the pose (which includes feature extraction and classification) and the calculation of the distance. There is also a block to be used when the images are part of a sequence of images (i.e. the chaser takes images continuously whilst approaching the target) in which case a prediction using information from previous classified images is used to correct errors.

1) Feature descriptors

Scale-Invariant Feature Transform (SIFT), Principal Components (PC) and HOG were selected as feature descriptors candidates because they have already proven successful in other areas and they are able to identify objects from their shape. The SIFT implementation used was based on the original paper from Lowe from 2004 [11] and was implemented by Xing Di. HOG implementation by Ludwig was written for his paper on pedestrian detection [12]. And the PC descriptor was obtained from adapting the implementation for face recognition by K.S Dash. From these, HOG was the one that returned better accuracies and therefore it was used to generate the results on this paper.

2) Edge detection

The feature descriptors researched use some sort of measure of direction in a global or local area. These algorithms perform much better when the input are edges instead of a grayscale image, since it is easier for the algorithm to measure the direction of simple lines. Two methods were tried, Sobel-Feldman [13] and Canny [14], Canny was chosen eventually because it returned consistently better results.

3) Blob detection

The object travelling in space that will be a target for the chaser will be a geometrical object (e.g. a spacecraft) or a bigger than average shape (e.g. as a comet) and therefore it should be easy to spot on a mainly dark background scattered with small stars. Using blob detection techniques, the bounding box of the object is obtained and the image can be cut snug around the target object, which will be the input for the pose detection algorithms.

4) Pose prediction

Assuming that we have a sequence of images of a target and that the changing position, orientation and size of the target evolves in a smooth, continuous way, it is possible to predict the pose of the satellite for each progressive image. Two approaches have been tried for prediction:

(1) Use the previous pose as prediction.

(2) Use the trend of the previous poses to calculate the next pose.

For (2) a simple algorithm based on linear regression is applied independently for each of the three rotation angles. We assume that subsequent poses are close enough that the rotation angles (*a*, *b* and *c* in the equation below) can be approximated by a linear function dependent on the distance from the chaser to the target (*d*):

$$a = \alpha_0 + \alpha_1 d$$
$$b = \beta_0 + \beta_1 d$$
$$c = \gamma_0 + \gamma_1 d$$

For each direction of rotation, using the angles from previously classified poses in the sequence, the parameters α_0, α_1, β_0, β_1, γ_0, γ_1 are calculated and used to predict the next angles *a, b, c*. For instance, if we take n previous instances of angle *b* we can build the following system of equations in matrix form:

$$\begin{bmatrix} b1 \\ \vdots \\ bn \end{bmatrix} = \begin{bmatrix} 1 & d1 \\ \vdots & \vdots \\ 1 & dn \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$

Simplifying the notation to $B = D\,\beta$, the coefficients can be solved by $\beta = B \backslash D$.

5) Distance to target calculation

To calculate the distance to the target we can rely on the relationship between the image and the camera parameters:

$$Distance = \frac{f\ * h_{physical\_target} * h_{full\_image}}{h_{target\_in\_image} * h_{sensor}}$$

Where:

- *f* is the focal length of the camera, often in mm.
- h_{full_image} is the height of the whole image, in pixels.
- $h_{target\_in\_imag}$ is the height of the target in the image, in pixels
- h_{sensor} is the height of the sensor of the camera used, in mm.
- $h_{physical\_target}$ is the real height that the target would have in that pose, which we put in mm to be consistent with the rest of the measurements

The dimensions of the real satellite are known as per the satellite specifications. The body has a diameter of 1500 mm and a height of 1940 mm. The height used in the equation is not the specifications' height, but the height of the target as seen from the camera. This means that the real target height has to be obtained for each of the possible poses. The real height is calculated off-line for each of the reference views and stored.

6) Full processing chain for sequences of images

The whole processing chain is put together, using HOG as feature descriptor for the classification.

First, the database of views from the 3D computer model is pre-calculated off-line using the developed Unity application and a vector of features of each view is kept. This means that neither the original 3D computer model nor the views are required for the system to run.

The operations on the captured image from the target will be processed on-line, the processing being invoked every time a new image from the target is received. The input for the system is a set of images in sequence. The full system processing chain includes the target detection in the image (blob detection), the edge detection and the features calculation, the selection of the best matching pose and the calculation of the distance to the target.

In order to integrate the pose prediction component to correct potential classification errors, the following two conditions are implemented:

1. The system avoids using the predicted pose if the previously classified poses do not follow a consistent pattern. This is because if a wrongly classified pose is used to calculate the next one, then the prediction will be erroneous too. We measure consistency in the sequence by calculating the variation between one pose and the next and assuming that a pose will not vary more than 10 degrees between images
2. If a prediction can be made and the difference between predicted pose and classified pose is more than 20 degrees, the assumption is that the classifier has failed and the prediction should be used instead.

These assumptions work if we assume that the changes in the target's pose happen very slowly and there are no sudden unexpected movements, which should be the case.

III. Results

The results presented here are for a database of reference images in 10 degrees increments about every angle. This means that the problem becomes the selection of the best matching view from over 23,500 views.

A. Classification results

The target in the image can be seen from different distances from the chaser, which means the target size in pixels varies in each image and it is also different to the size of the views in reference dataset. The size of the cell used for HOG needs to take into account the size of the target object, otherwise the feature vectors will not be comparable. To account for this, the targets are always analyzed using the same number of cells, which means the number of pixels per cell varies depending on the size of the object but the features are comparable. Also the number of cells used is important, so that the shape information is captured properly.

For the tests, the pose selected by the classifier is compared with the expected pose. The expected pose has been manually chosen by superimposing the test images on different reference views and selecting by eye the one in which the body of the satellite aligns better.

TABLE I. includes the accuracy of the classifier (percentage of views selected matching the expected pose) and also the mean pose error (difference between the selected pose by the classifier and the expected pose in degrees).

TABLE I. ACCURACY OF CLASSIFIER USING HOG WITH DIFFERENT NUMBER OF CELLS

Number of cells	Accuracy	Mean pose error	Views one step away from expected (+/- 10^o)
20x20	55%	22^o	85%
28x28	36%	31^o	82%
29x29	36%	17^o	85%
30x30	67%	24^o	85%
31x31	27%	48^o	64%
32x32	42%	31^o	79%
35x35	45%	22^o	88%

Fig. 2 shows two examples of correctly identified pose and two examples of failed classification. When the classification fails, in many cases the selected poses are not too different from the expected.

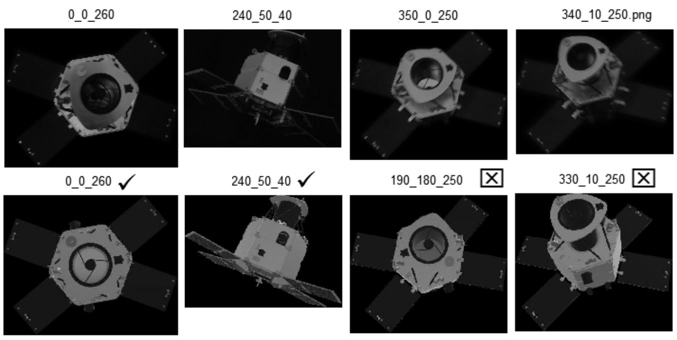

Fig. 2. Test images (top row) and classified pose (bottom row). Two examples of successful classification and two examples of failed classification are shown.

The results show that using 30x30 cells with overlap and 9 bins returns the best accuracy. Different numbers of bins were also tried for HOG but the results did not change. Inspecting and comparing the test images with the expected pose and the selected pose for 30x30 cells, the following can be noted:

- Many of the misclassified images choose a reference view which is one step away from the expected pose (this is, +/-10 degrees away in one or more axes). This is explained by the fact that reference views used are separated by 10 degrees in each axis and the test image is in between two views. For instance, the test image shows the satellite turned at 23 degrees about x but the available reference views are from 20 and 30 degrees. So even though, only 67% of the images in the dataset are properly identified, the percentage goes up to 85% if we include the ones that have chosen a view only one step away from the expected. In fact the mean pose error is just over 20^o.
- A problem was encountered with the physical mock-up model, since the solar panels are not perfectly aligned in the same plane, as opposed to the views of the 3D computer model. This is a problem for the classifier because many times the panels will be more similar to those of one reference image whilst the body will be more aligned with a different reference image.

Looking at the results, these issues explain many of the misclassified examples in TABLE I.

B. Results for sequences of images

The first results that were obtained for the different sequences used are summarized in TABLE II. Taking the best sequence, number 3, the classifier has found the expected pose on 74% of the cases. The percentage is of 83% accuracy if we take the poses that are within 10^o of the expected one. The error between the correct pose and the selected pose is 29^o on average for this case.

TABLE II. PRELIMINARY ACCURACIES ACHIEVED FOR EACH SEQUENCE

Sequence	N. images	Accuracy	Mean pose error	Poses within 10^o of the expected
01	22	59%	37^o	73%
02	15	53%	59^o	60%
03	23	74%	29^o	78%

At this point, the system is trying to classify each image individually, without taking into account the fact that they are part of a sequence. Looking at the results of the classifier for each of the images in the sequences, it is clear that there are some spurious mistakes that could be fixed (or at least mitigated) by considering that the previous poses in the sequence are known.

TABLE III. shows the results obtained when applying the previous pose as predictor and TABLE IV. shows the results when using the 3 previous poses to calculate regression and predict the next pose.

TABLE III. ACCURACIES ACHIEVED USING THE PREVIOUS POSE AS PREDICTOR

Sequence	N. images	Accuracy	Mean pose error	Poses within 10^o of the expected
01	22	64%	21^o	82%
02	15	60%	46^o	67%
03	23	78%	1.5^o	96%

TABLE IV. ACCURACIES ACHIEVED USING REGRESSION AS PREDICTOR

Sequence	N. images	Accuracy	Mean pose error	Poses within 10º of the expected
01	22	64%	4º	86%
02	15	60%	46º	67%
03	23	78%	3º	91%

When comparing TABLE III. and TABLE IV. with TABLE II. the accuracies improve for all sequences but also the mean error is reduced, in some cases down to an average value of 1.5 degrees.

Regarding the two predictors, the previous pose proves to be a good predictor, especially in sequence 3 where the changes in pose are very smooth. However, the predictor based on regression gives better results overall and for longer sequences will probably give better results if more than three previous poses are used to solve the regression equations.

Fig. 3 shows the improvement on the error between the expected pose and the selected pose. Dotted lines show the error before the pose prediction is put in place. Solid lines show the error afterwards. The second plot in Fig. 3 explains why the sequence 2 results do not improve much with the predictors: misclassification happened at the beginning of the sequence when there are not enough samples to predict the next poses with some confidence.

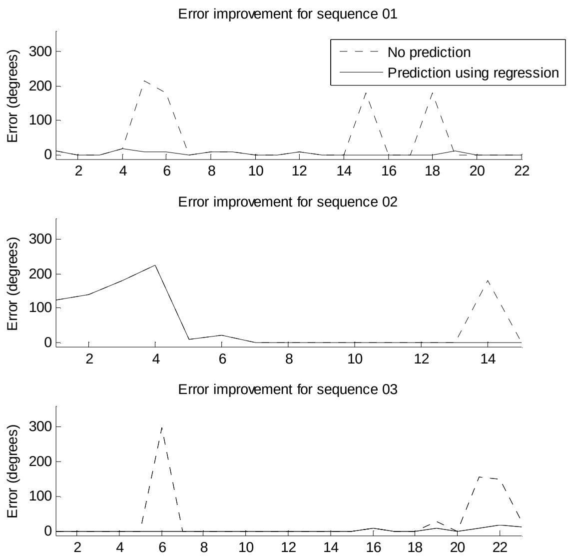

Fig. 3. Initial error (dotted line) and improved error through pose prediction (solid line) for each sequence.

The error does not seem to have a correlation with time, since errors appear equally at the beginning or at the end of the sequences. However, by analysing the misclassified images, the reasons for misclassification do seem to have a correlation with the size of the satellite in the image.

Misclassification at the beginning of the sequence 2 seems to be related to the lack of detail when the satellite is small for certain poses and also due to some images being blurred. The fact that the classifier manages to correct early errors is good.

The errors at the end of the sequence are more critical, since the chaser is closer to the target. However, the error at the end of the sequence is often between 0º and 20º, which is not too bad for the reference database used.

The misclassification at the end of the sequence it is often related to blob detection issues, when the satellite is bigger in the image. Parts of the satellite might appear disconnected and further away, the blob detection algorithm fails sometimes to see them as part of the same object. These could be solved by using knowledge about the sequence: the subsequent targets in a sequence will be slightly bigger than the previous one and in a similar position in the image. This will be a future improvement.

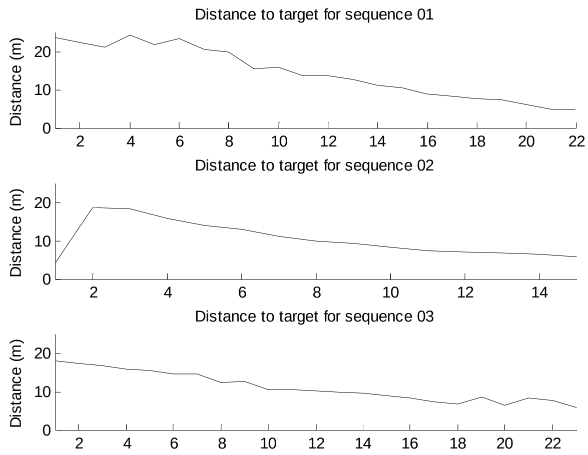

Fig. 4. Calculated distance to the target for each sequence.

The distance to the target is plotted in Fig. 4. It shows that misclassification has an impact on distance calculation so that when the selected pose is not correct, the distance calculation cannot be trusted as happens at the beginning of sequence 2.

IV. DISCUSSION AND CONCLUSION

The aim was to estimate the relative pose of a chaser in relation to a target body it is approaching using images from an on-board camera and having prior knowledge of the 3D shape of the target.

Different methods have been tried for each of the building blocks of the system, mainly target detection in an image, prediction techniques and different computer vision techniques for template matching to classify the pose. The result has been a working end-to-end system.

As descriptors for the classifier, HOG stood out by being able to describe the pose of the object with minimal calculations.

The best accuracies obtained by the system for pose detection were of 78% with a mean error of only 1.5º. This is calculated as the number of images for which the classifier returned the expected pose from a reference database with views of the target taken every 10º. These results include the implementation of a very simple predictor that was quite good at reducing the error, from an error of 29º to 1.5º.

Analyzing further the error obtained, we can say that the classified pose is often close to the expected pose. In fact, the accuracy increases to 96% if the selected poses that are only less than 10º away (one step away in the reference database) from the expected pose are included. A potential improvement would be to reduce the number of reference images to only the ones around the first classified poses. This would bring two benefits: first, the second pass could use a reference dataset with smaller increment between views (for instance, 5º instead of 10º) which should result in lower mean error. And second, the processing time also decreases as the reference dataset is smaller and there are fewer matches to check.

Another suggestion to reduce the error is to use a technique such as the Hough Transform to detect straight lines of the satellite and to adjust the pose detection by looking at the lines' angles in comparison to the angles of the reference dataset.

In order to obtain significantly lower error (i.e. less than 1º between selected and expected pose), more sophisticated 3D template matching and tracking techniques are necessary, such as in [15], where the 3D model is used to match the contour of the object in the image by successively projecting the 3D model, comparing contours and turning the model in the direction of the gradient, which gives the most probable convergence. This technique, however, would be significantly more demanding in terms of power and probably memory, which are precious resources on a spacecraft. The solution developed during this project uses very low overhead computation algorithms and only stores HOG descriptors of the reference views. A recommendation would be to use HOG to find the initial pose for another algorithm so that convergence to the optimal solution is quicker.

When looking at the results, it is worth noting that the mock-up model used did not exactly match the 3D computer model, because some of the solar panels on the physical mock-up were slightly bent, which also limited the accuracy that could be achieved.

For further work, bad illumination conditions have to be taken into consideration such as partial occlusion due to lack of light in space, where half of the object can be illuminated by the sun and the other half can be completely in the dark, has not been assessed.

Overall, the research goals have been met by testing suitable techniques for extraction of features and investigation into ways of classifying the pose of the satellite using information from the 3D computer model. An integrated solution was implemented that detects the target and estimates the distance and the relative pose of the target object.

Acknowledgment

I would like to thank Elecnor Deimos for allowing me time to work on my MSc and for providing the input data for this project, the 3D model and model mock-up of the Deimos-2 satellite. I would also like to thank Javier Montaño for suggesting the topic of this dissertation. And thanks to Andy for his support and ideas.

References

[1] A. Petit, E. Marchand and K. Kanani, "Vision-based Detection and Tracking for Space Navigation in a Rendezvous Context," in Int Symp on Artificial Intelligence, Robotics and Automation in Space, i-SAIRAS, Turin, 2012.

[2] A. Petit, E. Marchand and K. Kanani, "Vision-based Space Autonomous Rendezvous: A Case Study," in 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, 2011.

[3] A. Petit and K. Kanani, "Vision Based Navigation for Debris Removal Missions," IAC, 2012.

[4] J. C. Bastante, J. Vasconcelos, L. F. Peñín, A. Caramagno, J. Dinis, J. Rebordao and S. Salehi, "Performance Assessment of a Vision Based GNC for Automated RV and Capture of a Passive Target in a Martian Elliptical Orbit".

[5] S. Hoermann and P. Borges, "Robot localization using 3D'moedls and an off-board monocular camera," in Proc. IEEE Int. Conf. Comput. Vis. Workshop, 2011.

[6] C. Fries, T. Luettel and H.-J. Wuensche, "Combining Model- and Template-based Vehicle Tracking for Autonomous Convoy Driving," in Proceedings of IEEE Intelligent Vehicles Symposium (IV), Australia, 2013.

[7] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2005, San Diego, CA, USA, 2005.

[8] R. Hu and J. Collomosse, "A Performance Evaluation of Gradient Field HOG Descriptor for Sketch Based Image Retrieval," in Computer Vision and Image Understanding, 2013.

[9] A. Chayeb, N. Ouadah, Z. Tobal, M. Lakrouf and O. Azouaoui, "HOG based multi-object detection for urban navigation," in IEEE 17th International Conference on Intelligent Transportation Systems (ITSC), Qingdao, 2014.

[10] L. Prasuhn, Y. Oyamada, Y. Mochizuki and H. Ishikawa, "A HOG-based hand gesture recognition system on a mobile device," in IEEE International Conference on Image Processing (ICIP), Paris, 2014.

[11] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Points," International Journal of Computer Vision, 2004.

[12] O. J. Ludwig, "Trainable classifier-fusion schemes: an application to pedestrian detection," Intelligent Transportation Systems, vol. 2, 2009.

[13] I. Sobel and G. Feldman, "A 3x3 Isotropic Gradient Operator for Image Processing," Stanford Artificial Intelligence Project (SAIL), 1968.

[14] J. Canny, "A Computational Approach to Edge Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vols. Vol. PAMI-8, no. 6, pp. 679-698, 1986.

[15] V. Prisacariu and I. D. Reid, "Pwp3d: Real-time segmentation and tracking of 3d objects," International Journal of Computer Vision, vol. 98, no. 3, pp. 335-354, 2012.