Specular 3D Object Tracking by View Generative Learning

Yukiko Shinozuka, Francois de Sorbier and Hideo Saito
Keio University
3-14-1 Hiyoshi, Kohoku-ku
223-8522 Yokohama, Japan
shinozuka@hvrl.ics.keio.ac.jp

Abstract

This paper proposes a novel specular 3D object tracking method. Our method works with texture-less specular objects and objects with background reflections on the surface. It is a keypoint-based tracking using a view generative learning. Conventional local features are robust to scale and rotation, but keypoint matching fails when the viewpoint significantly changes. We apply a view generative learning to improve the robustness to viewpoint changes. To be robust to large appearance changes, our method does view-dependent rendering for generating views and stores all the descriptors of the keypoints on the generated images and its 3D-position in the reference database called "feature table". We conducted quantitative evaluation on the object pose and showed our method outperforms compared with the other view generative learning methods in terms of tracking accuracy and learning process.

Keywords: View Generative Learning, 3D Object Tracking, Local Feature, Feature Table, Specular Object

1 Introduction

Object pose estimation is necessary to augment a virtual object on a real environment for augmented reality. To estimate the object pose, it is required to find correspondences between a reference dataset and an input image for vision-based methods. There are two types of methods based on the target objects; planar model-based methods [Lepetit and Fua, 2006] and 3D model-based methods [Drummond et al., 2002]. Our proposed algorithm takes the latter solution to track a 3D object.

Keypoint matching is one of the solutions to find correspondences. Local features such as scale-invariant feature transform (SIFT) [Lowe, 2004] is well-known as a keypoint extractor and descriptor. It is robust to rotation, translation and illumination changes, but there is a limit for affine transformation. There are plenty of studies of local features to improve this limitation. Harris-affine [Baumberg, 2000] and Maximally-stable extremal region detector (MSER) [Matas et al., 2002] are known for the invariance to affine transformation. However there is no descriptor for each of them. That means even if the keypoints are extracted, the feature description will be different from the ones extracted on the image before transformation. Affine-SIFT (ASIFT) [Morel and Yu, 2009] is also known as affine invariant. This method applies several possible transformation before matching.

A generative learning method is proposed as another solution for keypoint matching. It is a learning method which uses local features which is not invariant to affine transformation. It virtually generates the possible views and extracts keypoints from them. If the same keypoint is extracted from different views, this point is considered as "stable keypoint". Stable keypoints are robust under strong perspective view changes. For creating a reference database, machine learning process is often conducted. Lepetit *et al.*'s method [Lepetit and Fua, 2006] uses randomized trees with huge amount of training dataset, whereas Thachasongtham *et al.*'s method [Thachasongtham et al., 2013] uses k-means clustering with much less dataset. However, in

both methods, they consider a target object is covered with Lambartian surface, so they do not take highlight and specular areas into consideration.
Our motivation is to estimate 3D object pose by vision-based 3D object tracking using view generative learning. Our contributions are to propose a novel view generative learning method "feature table" to track a specular object. In the experiments, we compare with other generative learning methods (randomized trees [Lepetit and Fua, 2006] and k-means [Thachasongtham et al., 2013]). We evaluate the rotation matrix and translation vector of the target object and computational time. Our experimental results show that our proposed method outperforms in tracking specular 3D objects with less training datasets.

2 Related Works

This section refers to other generative learning tracking methods and the recent trials on specular object tracking.

As already mentioned in section 1, local features such as SIFT [Lowe, 2004] is not invariant to perspective transformation. A generative learning is proposed to improve this weakness for keypoint matching. It is the learning method which generates the possible views by affine or perspective transformation and selects robust stable keypoint.

Randomized trees method [Lepetit and Fua, 2006] is a generative learning method which considers a keypoint matching problem as a patch classification problem. It applies affine transformations to the image patches around the extracted keypoints and trains with them by randomized trees. It requires large amount of training dataset for learning. Learning process computational costs time due to the size of the dataset and the recursive algorithm of randomized trees, but tracking runs quite fast.

Thachasongtham *et al.* propose a generative learning method with k-means clustering for 3D object tracking [Thachasongtham et al., 2013]. His method is similar to randomized trees, but there are three main differences. In randomized trees method, keypoints are extracted once before affine transformation whereas Thachasongtham *et al.*'s method extracta the keypoints from every generated patterns. For learning, they apply k-means to determine the centroid of the stable keypoint. For the datasize of the learning data, k-means method requires much less than randomized trees method.

However, both methods assume that a surface of a target object is covered with Lambartian surface. Therefore when the specular reflection occurs, it is hard to extract the keypoints from the same area from different views.

Torki *et al.* propose that a regression was a key to estimate a 3D object pose with specular highlight [Torki and Elgammal, 2011]. The regression is calculated from the 2D-position of each keypoints and its descriptors from the video sequences. They succeed in estimating rotation of cars. Netz *et al.* consider high light is one of the features of the image, then use the specular as features [Netz and Osadchy, 2011]. Our method uses the same concept that the highlight area can be characteristics in the images.

3 View Generative Learning – Feature Table

This section refers to the algorithm of a generative learning method. A generative learning is proposed to be robust to viewpoint changes. It is a keypoint-based method which virtually generates the possible images of different viewpoints and extracts keypoints from them for the creation of the reference database. If the keypoint is extracted at the same position in 3D world coordinate from different views, this point is considered as a "stable keypoint". Dataset consists of the descriptors of these stable keypoints and their position in 3D world coordinate.

The algorithm is shown in Figure 1. The method can be divided into learning and tracking phase. The learning phase has to be done off-line phase before tracking.

There are main two differences between our method and other view generative learning methods [Lepetit and Fua, 2006] [Thachasongtham et al., 2013]. Both points contribute to im-

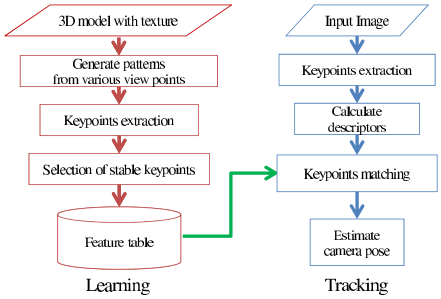


Figure 1: Overview

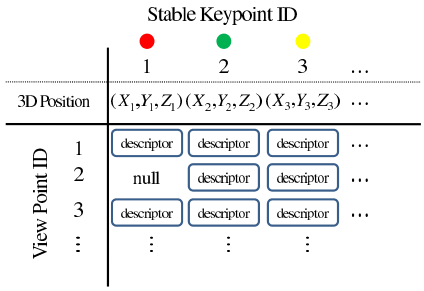


Figure 2: Feature Table

prove the robustness to appearance changes such as highlight. First of all, we conduct view-dependent rendering to create the possible views whereas the conventional methods only do affine-transformation. This process enables to include the highlight area in the database. Second point is to create "feature table". Feature table is a table with descriptors of the stable keypoints. Its vertical axis is for viewpoint ID, and the horizontal for stable keypoint ID as shown in Figure 2. We store all the descriptors extracted from the possible views in the table. This process enables to absorb the difference of the descriptors on the same stable keypoint.

4 Learning

4.1 Generate Views

We require a 3D model as an input data for learning. The images from various viewpoints are generated virtually from it. There are two important concerns in this phase. First point is the background of the learning phase. If the object is static in a scene and the background texture is available, the 3D model of the background is reconstructed and we also learn the background. If the texture of the background is not available, we generate the virtual views with a random colored background to be robust to the noisy background. Second point is lighting condition. We generate the views by view-dependent rendering to extract the keypoints around the highlight areas.

The camera pose setting is important to generate the virtual views. Since local feature such as SIFT is scale invariant, there is no need to take the distance from the object into consideration. It means the distance between the virtual viewpoint and the object scene does not have to be changed for learning different views. The different rotation angles of the camera also do not have to be learned, because SIFT is rotation invariant. Thus, we change only two angles, the longitude ϕ and the latitude θ for generating different viewpoint images for generative viewpoint learning.

4.2 Keypoints Extraction and Stable Keypoints

We extract the keypoints from the generated patterns by local feature. Each keypoint p on the image is reprojected to p' in the 3D world coordinate by perspective matrix P. The perspective matrix is already given in section 4.1. The equation of the reprojection is shown in equation (1).

$$p'_i \sim P p_i \tag{1}$$

We compare the Euclidean distance of the reprojected points from different views. If their Euclidean distance is under the threshold, these points are considered as the same point in 3D world coordinate. The keypoints with high repeatability are called "stable keypoints" because they can be extracted from other viewpoint images. We store the stable keypoints with high repeatability. We sort the stable keypoints in order of repeatability and store the top N stable keypoints. If the target object has less-texture, the number of the stable kepoints can be lower than threshold N. If it happens, we store all the stable keypoints in the database.

4.3 Creating Feature Table

Feature table is a table with descriptors of the stable keypoints. Its vertical axis is for viewpoint ID, and the horizontal for stable keypoint ID as Figure 2 shows. The descriptor at the same keypoint can be described differently depending on highlights and viewpoint changes. Therefore, all the descriptors from the generated images are stored in our method, whereas the conventional methods did not consider the differences.

Each stable keypoint has multiple descriptors and one position in the 3D world coordinate. The position is calculated by getting the centorid of the keypoints in each stable keypoint group.

5 Tracking

To estimate the object pose for tracking, the projection matrix is calculated by referring to the feature table. After extracting local feature on an input image, we find the nearest descriptor by fast approximate nearest neighbor matching in the feature table. To decrease false matching, we apply nearest-neighbor distance ratio between the first (D_A) and second closest (D_B) as Mikolajczyk *et al.* mentioned in [Mikolajczyk et al., 2005]. If the keypoint fulfills equation (2), it is considered as a correct correspondence. We set $\tau = 0.6$ in our experiments. After finding the correspondences, we use a robust estimator RANSAC and calculate projection matrix with the 2D and 3D positions of the keypoints.

$$\frac{|D_A|}{|D_B|} < \tau \tag{2}$$

6 Experimental Results

6.1 Parameters and System Configuration

We conducted two experiments and compared our method with randomized trees [Lepetit and Fua, 2006] and k-means method [Thachasongtham et al., 2013].

In the first experiments, we set a texture-less specular object **Box** as a target object. We conducted the learning in a random color background. We rotated the object from -15 degrees to 85 degrees in longitude ϕ and 0 to 360 degrees in latitude θ for every 10 degrees.The distance between the camera and the object was set as 40 cm.In the second experiment, we tracked a object with background reflection **Teapot**. We used the texture and 3D structure of the background in the learning. We rotated the object every one degree in latitude where its longitude equals to zero degree. The distance between the camera and the object was set as 30 cm.

In both experiments, the video sequences are created by computer graphic to get the ground truth. We used the same video for learning and testing. We set the number of the stable keypoint N equals to 2000. All the experiments were implemented on Windows 7, 64 bits, Intel Core i7-3930K 3.20GHz CPU, 16.00GB RAM and GeForce 310 589MHz GPU. We chose SIFT-GPU [of North Carolina,] for local feature.

6.2 Texture-less Specular Object

This section shows the tracking result of object **Box**. Figure 3 shows our proposed "feature table" worked the best of all. Randomized trees method did not track the object in any frame because randomized trees is designed for large amount of database, but the number of the stable keypoints was only a few due to less-texture for object **Box**. (The number of the stable keypoint was 807.) This result shows our method works with much less training data.

Figure 4 shows the L-2 norm error of rotation matrix and the error of translation vector in each axis. They show the huge translation error occurred often in k-means method compared with feature table.

Table 1: Computational Time

Learning method	**Box**	**Teapot**	Tracking method	**Box**	**Teapot**
Randomized Trees [sec]	11304	3115	Randomized Trees [msec]	541	108
K-means [sec]	340	106	K-means [msec]	415	762
Feature Table [sec]	379	114	Feature Table [msec]	26612	22050

6.3 Object with Background Reflection

This section shows the result of object **Teapot**. Figure 3 shows k-means and feature table method tracked 3D object whereas randomized trees did not did not track the object in any frame. It shows our method outperformed in tracking. It is because the descriptors on the same stable keypoints are too different from each other so that the other methods did not absorb these differences.

We evaluated the results on 3D object pose (rotation matrix and translation vector) in Figure 5. The translation errors are better in our proposed method and there is not much difference in rotation.

6.4 Computational Time

Table 1 shows the computational time for learning and tracking. The computational time of randomized trees method for learning cost more than that of the others. It is because the algorithm of randomized tees is recursive. The tracking time of feature table was the slowest of all because the database is not compressed and the size is the largest.

7 Conclusion

This paper proposed a novel specular 3D object tracking method "feature table". It worked with the texture-less specular objects and objects with background reflections. Our contributions are following two points. We used the idea that highlight or non-feature area should be included in database to be robust to large appearance changes. Second point is that our method applied a generative learning with less training dataset to improve the robustness to viewpoint changes.

Our experimental results showed our method outperformed in terms of tracking accuracy compared with other methods [Lepetit and Fua, 2006] [Thachasongtham et al., 2013]. Speed in tracking should be improved, but our method required less training computational time with less training data.

References

[Baumberg, 2000] Baumberg, A. (2000). Reliable feature matching across widely separated views. In *CVPR*, pages 1774–1781.

[Drummond et al., 2002] Drummond, T., Society, I. C., and Cipolla, R. (2002). Real-time visual tracking of complex structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:932–946.

[Lepetit and Fua, 2006] Lepetit, V. and Fua, P. (2006). Keypoint recognition using randomized trees. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(9):1465–1479.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110.

[Matas et al., 2002] Matas, J., Chum, O., Urban, M., and Pajdla, T. (2002). Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, pages 36.1–36.10. BMVA Press.

[Mikolajczyk et al., 2005] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Gool, L. V. (2005). A comparison of affine region detectors. *Int. J. Comput. Vision*, 65(1-2):43–72.

[Morel and Yu, 2009] Morel, J.-M. and Yu, G. (2009). Asift: A new framework for fully affine invariant image comparison. *SIAM J. Img. Sci.*, 2(2):438–469.

[Netz and Osadchy, 2011] Netz, A. and Osadchy, M. (2011). Using specular highlights as pose invariant features for 2d-3d pose estimation. In *CVPR*, pages 721–728. IEEE.

[of North Carolina,] of North Carolina, U. Siftgpu.

[Thachasongtham et al., 2013] Thachasongtham, D., Yoshida, T., Sorbier, F., and Saito, H. (2013). 3d object pose estimation using viewpoint generative learning. volume 7944, pages 512–521. Springer Berlin Heidelberg.

[Torki and Elgammal, 2011] Torki, M. and Elgammal, A. (2011). Regression from local features for viewpoint and pose estimation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2603–2610.

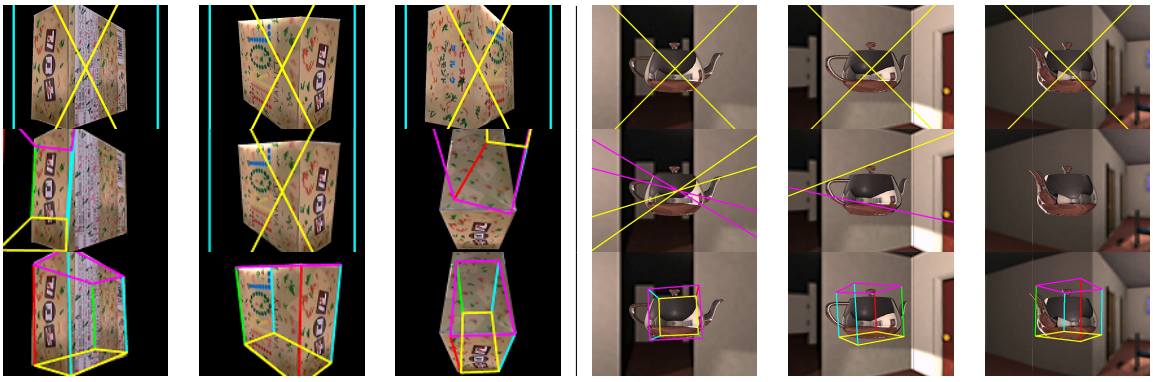
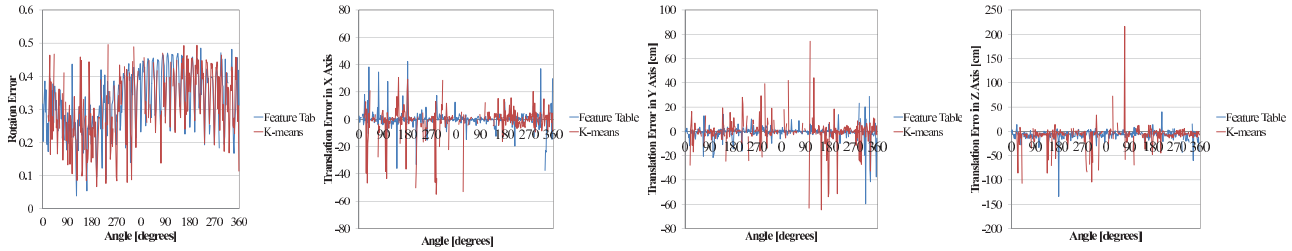
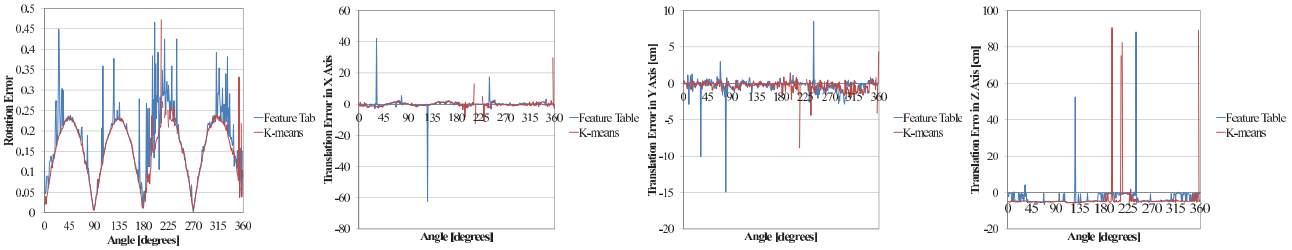


Figure 3: Tracking Results (Top : Randomized Trees, Middle : K-Means, Bottom : Feature Table) (Left : **Box** where $(\theta, \phi) = (-5, 140), (15, 220), (55, 180)$ in degrees, Right : **Teapot** 0,28,239 degrees



(a) Rotation (b) Translation in X Axis (c) Translation in Y Axis (d) Translation in Z Axis

Figure 4: Average Error of Object Pose (Texture-less Specular Object)



(a) Rotation (b) Translation in X Axis (c) Translation in Y Axis (d) Translation in Z Axis

Figure 5: Average Error of Object Pose (Background Reflection)