3D Object Pose Estimation Using Viewpoint Generative Learning

Dissaphong Thachasongtham, Takumi Yoshida,
François de Sorbier, and Hideo Saito

Graduate School of Science and Technology,
Keio University Yokohama, Japan
{dth,tkm,fdesorbi,saito}@hvrl.ics.keio.ac.jp
http://www.hvrl.ics.keio.ac.jp

Abstract. Conventional local features such as SIFT or SURF are robust to scale and rotation changes but sensitive to large perspective change. Because perspective change always occurs when 3D object moves, using these features to estimate the pose of a 3D object is a challenging task. In this paper, we extend one of our previous works on viewpoint generative learning to 3D objects. Given a model of a textured object, we virtually generate several patterns of the model from different viewpoints and select stable keypoints from those patterns. Then our system learns a collection of feature descriptors from the stable keypoints. Finally, we are able to estimate the pose of a 3D object by using these robust features. In our experimental results, we demonstrate that our system is robust against large viewpoint change and even under partial occlusion.

Keywords: pose estimation, generative learning, stable keypoint.

1 Introduction

The problem of 3D pose estimation of rigid objects has been studied for several decades because estimating the pose (position and orientation) of a known object in an unknown scene is a significant issue in the fields of Computer Vision and Augmented Reality. So far, a number of model-based methods have conventionally been proposed in order to solve this problem. Most of them can be classified into two categories based on reference object as follows: planar model-based methods [1,2], and 3D model-based methods [3,4]. Both have different pose estimation processes. Planar model-based methods are to estimate a homography which defines perspective transformation between a reference object and an input image. On the contrary, 3D object-based methods are to recover all six degrees of freedom of a reference object with respect to the scene.

Keypoint matching is one of the most important technologies to achieve good performance of pose estimation for both categories of the above methods. Typically, local features, such as SIFT [5] or SURF [6], are widely used to estimate a homography for a planar surface because of their own scale and rotation invariance. However, they often fail when a reference object is moved because these

J.-K. Kämäräinen and M. Koskela (Eds.): SCIA 2013, LNCS 7944, pp. 512–521, 2013.

features are sensitive to large perspective change. Therefore, for a 3D object where perspective changes always occur when moving it, the keypoint matching using the original processes of SIFT or SURF is not enough to estimate its pose. An additional learning process is a compulsory task to handle large perspective change problems.

Improving these features to estimate the pose of a 3D object is our main objective. In our previous work, Yoshida et al. [7] presented a stable keypoint matching method which is robust even under strong perspective changes. We used viewpoint generative learning to train our system before the beginning of the keypoint matching. After the learning, our system is able to effectively estimate the pose of an object based on keypoint matching. As a result, our learning method can handle the large perspective change problem; however, it was limited to only planar surfaces. In this paper, we propose a 3D pose estimation method which extends this previous work on viewpoint generative learning to 3D object. Given a model of a textured 3D object, we virtually generate several patterns from different viewpoints of the model and collect feature descriptors from those patterns. Only the feature descriptors of keypoints, which can be detected repeatedly in different poses of generated patterns, are collected. After the learning, our system is able to estimate the pose of a 3D object based on 3D-to-2D keypoint matching. To sum up, these conventional local features are able to estimate the pose of a 3D object by using our viewpoint generative learning method.

2 Related Works

Apart from using keypoints as local features, 3D object pose estimation can be done by several approaches. Some approaches use more than one local feature. For example, in [8], both edges and vertices of the 3D model were utilized to estimate the pose of a 3D object. Moreover, by combining with information of the external system such as magnetic sensor or vision marker, the accuracy and the robustness under rapid camera movement can be improved. However, edge-based methods are not suitable for round objects or objects with hard-to-detect edges. Nowadays, depth camera is an alternative option to use in 3D pose estimation and tracking [9]. Moreover, due to an existence of depth cameras, point cloud-based methods also became popular in 3D recognition and pose estimation [10]. Although there are many advantages of depth information, depth camera is not a convenient tool. If depth information is not available, keypoint matching is still a powerful technique to estimate the pose of a 3D object by using only images from an RGB camera.

Recently, improving conventional local features such as SIFT or SURF in order to handle large perspective change for automatic 3D object pose estimation has become a considerable interest. Because our method focuses on using these features to estimate the pose of a 3D object, we discuss in this section about other methods which focus on the same goal as follows: Randomized Trees [11] and Gravity-Aware [12]. Both of them are keypoint-based learning methods which

can recognize keypoints and estimate the pose of 3D object by using generative learning.

Randomized Trees is a generative learning method that considered keypoint matching problem as a patch classification problem. It applies affine transformations to image patches around detected stable keypoints to train the randomized trees. It recognizes the corresponding patches based on intensity comparison.

Gravity-Aware is a camera localization method. This method creates synthetic views from a reference image and creates descriptors for them. The system selects the most representative subset of the descriptors, which is used in online feature matching. After finish the learning, it recognizes the corresponding keypoints by comparing descriptors from an input image with all representative descriptors. Furthermore, a gravity orientation from inertial sensors can be used to increase the accuracy of camera localization.

3 Overview

In this section, we discuss the overview and the structure of our pose estimation system. It can be divided into two phases: learning phase and detecting phase. A 3D object is used as a reference. First of all, we assume that a 3D model with texture of the object has been reconstructed and the intrinsic parameters of the camera are obtained. In the learning phase, we modify the viewpoint generative learning for planar surface [7] to train our system. The feature descriptors from stable keypoints obtained during the learning phase are stored in a database. In the detecting phase, we utilize this database to match stable keypoints from an input image, and then estimate the pose of the reference object. The structure of our system is illustrated in Fig. 1.

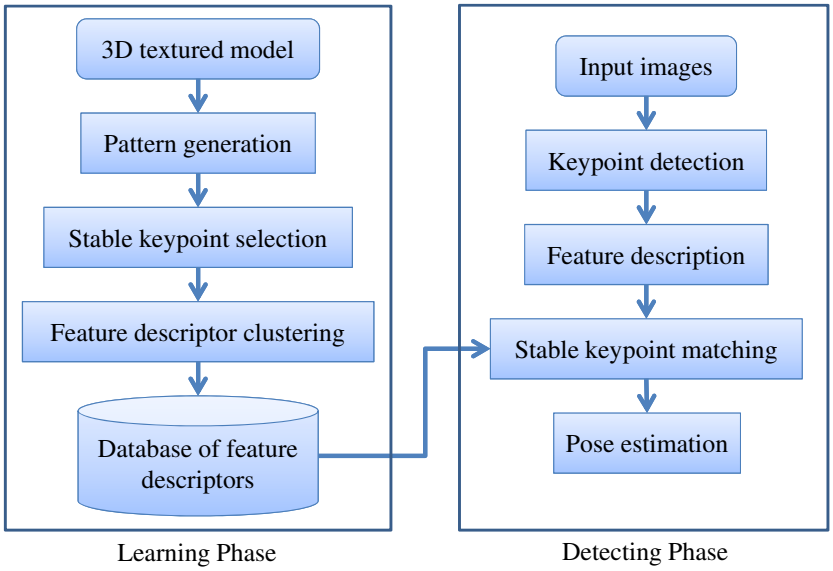


Fig. 1. Overview of our proposed method which is divided into learning phase and detecting phase

4 Viewpoint Generative Learning

Viewpoint generative learning is a key method during the learning phase. By using this method, our system is able to learn various features from patterns which are virtually generated from several viewpoints. First, we generate patterns of the model from various viewpoints. Then we detect stable keypoints from the generated patterns and create a database of feature descriptors. After the learning stage, the model and an input image can be matched by comparing their features using this database.

4.1 Generation of Various Patterns

We now present a method for virtually generating the patterns as illustrated in Fig. 2. It consists of pre-captured textured 3D model of the reference object located at the center of the sphere, and a virtual camera at a point on the surface of the sphere. We simply apply perspective transformations in order to generate all the patterns. For each generated viewpoint, we collect not only the patterns but also the depth and viewpoint information. Note that our method does not apply any transformation to any single image patch but apply perspective transformation with the purpose of moving a virtual camera around the model. As a result, the generated patterns are images of the reference model captured by the virtual camera from multiple viewpoints.

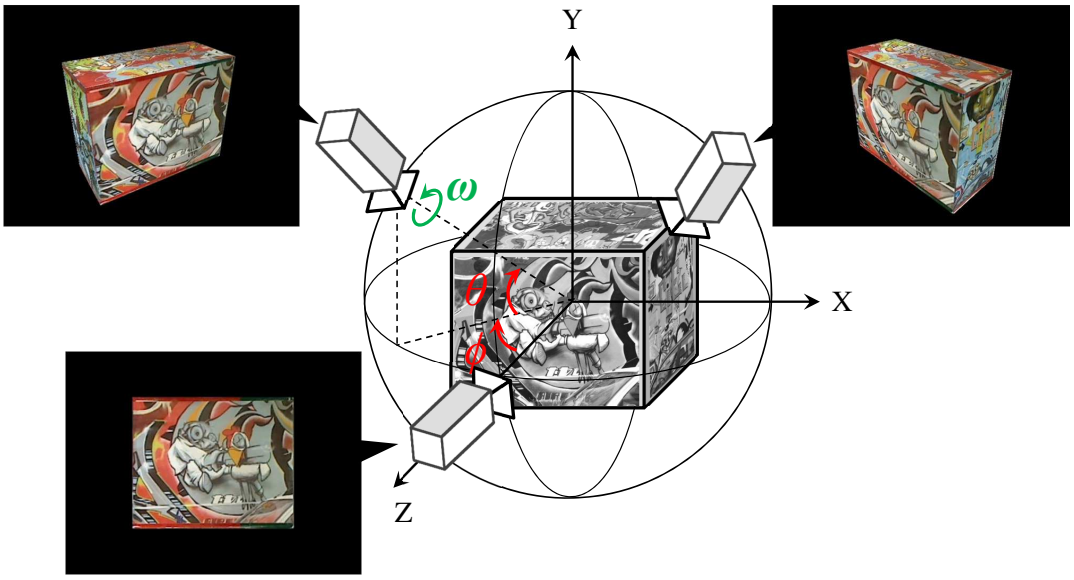


Fig. 2. Our method virtually captures the 3D model for generating several patterns from different viewpoints

The parameters θ and ϕ respectively describe the rotation around the X-axis and the rotation around the Y-axis of the virtual camera. The parameter ω defines the spin of the virtual camera. The rotation matrix R and the translation

vector t are shown in (1), where d is the distance between the virtual camera and the model.

$$R = \begin{bmatrix} \cos\omega & -\sin\omega & 0 \\ \sin\omega & \cos\omega & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}, t = \begin{bmatrix} 0 \\ 0 \\ d \end{bmatrix} \quad (1)$$

In order to collect data from every single viewpoints of the model, we set the rotation ranges as, $\theta \in [-90°; 90°]$ and $\phi \in [0°; 360°]$. Thanks to the rotation invariance, the parameter ω that describes the spin of the virtual camera is constant. Besides, because of the scale invariance, the distance d between the virtual camera and the model is also constant.

4.2 Selecting Stable Keypoints

Stable keypoints are defined as the keypoints that can be detected repeatedly in different poses of generated patterns at the corresponding locations. First, in order to select the stable keypoints, we detect keypoints for every generated pattern with a local feature detector. Then, all detected keypoints are projected from pixel coordinate system to 3D coordinate system by using the collected depth information and viewpoint information of each generated pattern. We search for the closest keypoint in the 3D coordinate system by measuring Euclidean distance. If the Euclidean distance between the keypoint and the closet keypoint is lower than a threshold, we consider those keypoints as the same keypoint and then increase its repeatability. On the other hand, if no keypoint is found around the projected position of the keypoint, we conveniently create a new keypoint at that position in the 3D coordinate system. Figure 3 shows examples of keypoint projection to 3D coordinate system and stable keypoint selection. When all generated patterns are processed, the keypoints with the highest repeatability are selected as stable keypoints. In summary, we select a set of stable keypoints based on repeatability of detection among the generated patterns.

Our method is able to cover all keypoints which can be detected on generated patterns because it detects keypoints not on a single reference image but on the entire set of generated patterns.

4.3 Creating Database of Feature Descriptors

When keypoints are detected on the generated patterns, we also collect the descriptors of each keypoint. As a result of selecting stable keypoints, the number of descriptors for each keypoint is equal to the number of patterns in which the keypoint was detected. This would result in a huge database which slows down the matching process, therefore the number of descriptors has to be decreased. A clustering algorithm, k-means++ [13], is used to cluster the set of collected descriptors for each stable keypoint. Then, we collect the barycenter of the set of descriptors of each stable keypoint into a database. As a result, the database contains $N \times K$ feature descriptors, where N is the number of stable keypoints and K is the number of clusters.

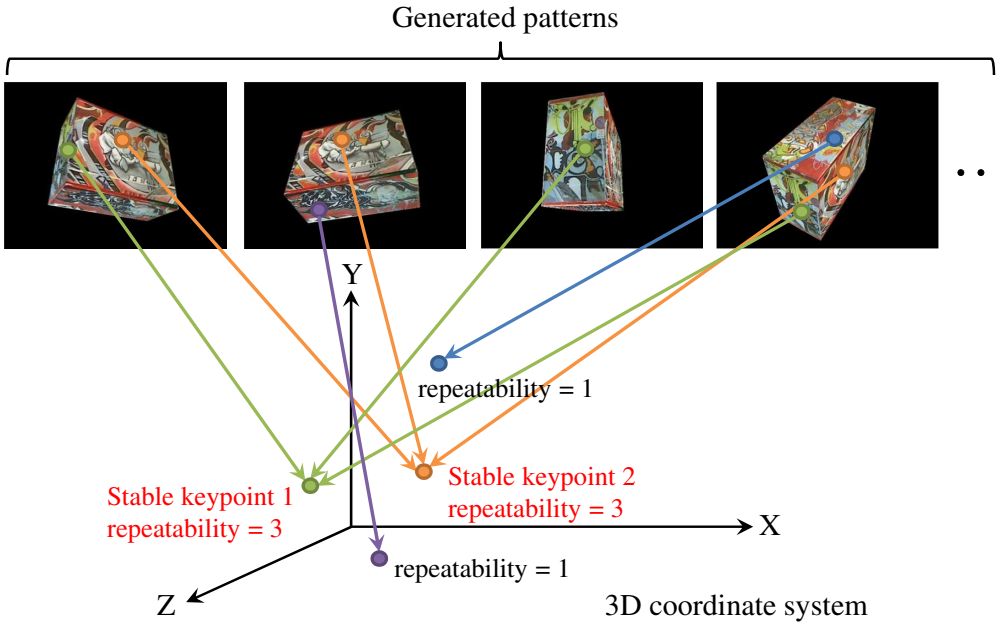


Fig. 3. We select the keypoints with the highest repeatability among the patterns generated from our pre-computed textured 3D model

5 Keypoint Recognition and Pose Estimation

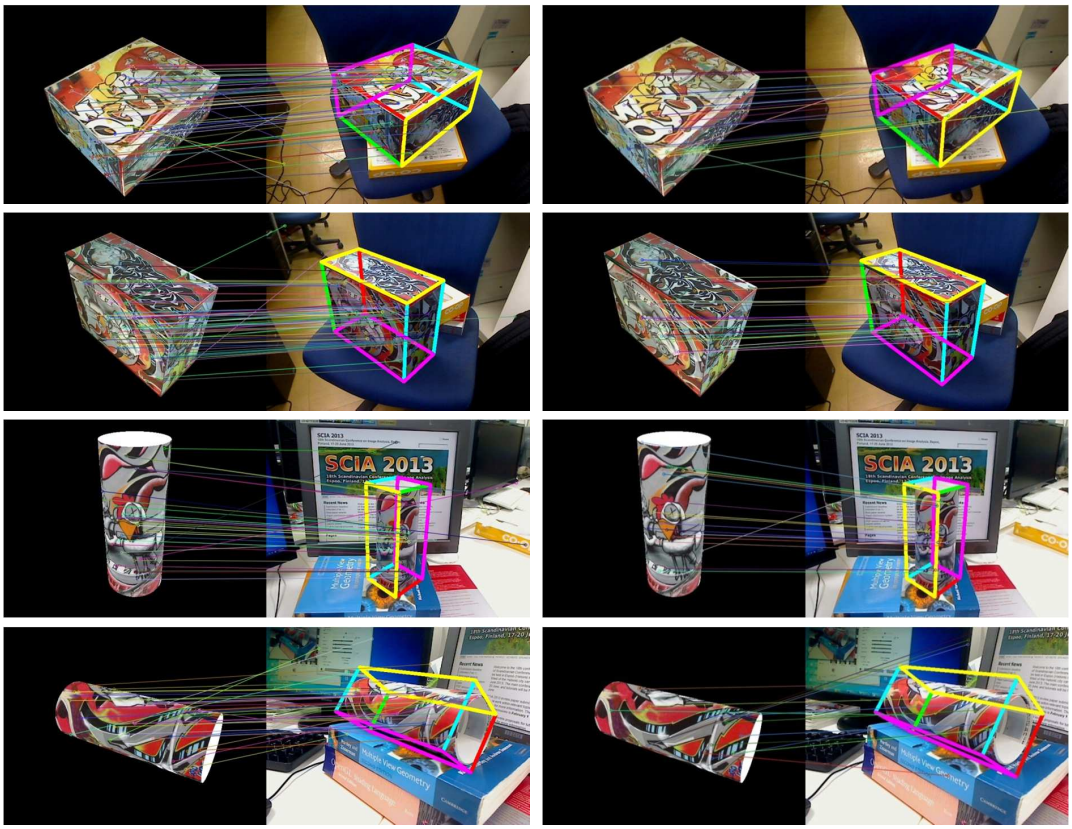
After finishing the learning, our system can detect a reference object and estimate its pose by keypoint matching. In the detecting phase, we first acquire an input image from a camera with known intrinsic parameters. We detect keypoints and compute feature descriptors from the input image. Then we compare the obtained descriptors with descriptors in the generated database by computing Euclidean distance of their high dimensional value. We validate the matching result by applying nearest-neighbor distance ratio [14] to the distance ratio between the nearest-neighbor feature descriptors and the second nearest-neighbor feature descriptors. As a result, we can identify the correspondences between the stable keypoints on the model in 3D and the keypoints on the input image in 2D. These correspondences, together with the intrinsic parameters of the camera, are used to estimate the pose of the reference object by solving the related Perspective-n-Point (PnP) problem [15] with a robust estimator RANSAC.

6 Experimental Results

In order to evaluate the performance of our method, we implemented a 3D pose estimation system based on our proposed method. We used a Logitech Webcam Pro 9000 to capture input images with a resolution of 640×480 pixels. Before starting any experiment, we estimated the intrinsic parameters of the camera and reconstructed the reference object in advance. We created textures of the reference object by pasting Graffiti images [16] on the reference object and applied the texture information of the reference object to the 3D model.

During the pattern generation process, the angular distances (θ, ϕ) between any two consecutive patterns were set to $(20°, 0°)$ or $(0°, 20°)$. This resulted in 162 different generated patterns under the condition of scale and rotation invariance.

To verify the effectiveness of our proposed method, we demonstrate the results of stable keypoint matching by drawing lines that show stable keypoint correspondences, without outlier removal, between input images and their most similar generated patterns as shown in Fig. 4. We also demonstrate the results of 3D pose estimation after using RANSAC by superimposing the bounding boxes of the reference object onto the input images with geometrical consistency. A box-shaped object was used as a reference object due to ease of modeling and a cylinder-shaped object was also used as a reference object to demonstrate that our method can relax planar constraint. As shown in the result images in Fig. 5, our system can estimate the pose accurately even under partial occlusion.



(a) (b)

Fig. 4. Result images of stable keypoint matching without outlier removal and pose estimation after using RANSAC. (a) Using SIFT as feature. (b) Using SURF as feature.

6.1 Evaluation of Robustness

In this experiment, the robustness of our system was defined as a pose estimation success rate on sample images. Since the robustness of our system depends on the number of stable keypoints N and the number of clusters K, we conducted

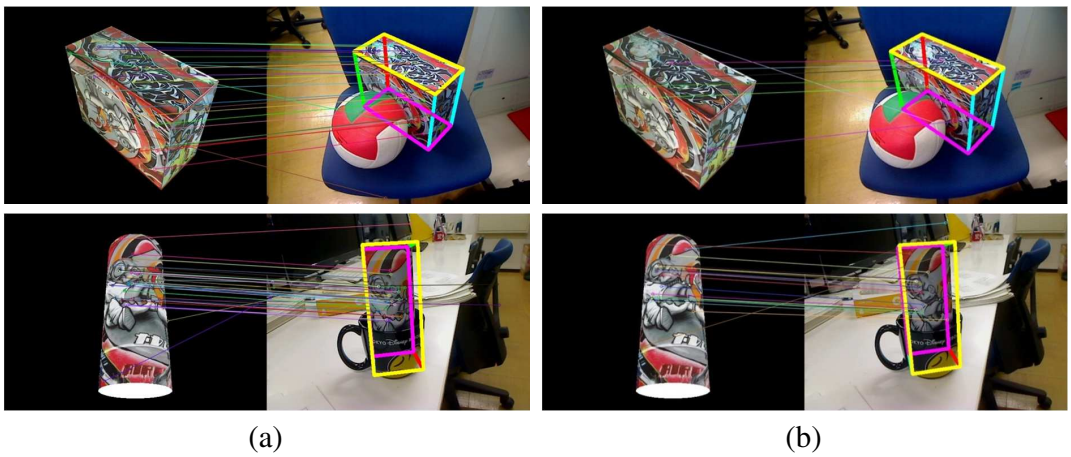


Fig. 5. Result images with partial occlusion. (a) Using SIFT as feature. (b) Using SURF as feature.

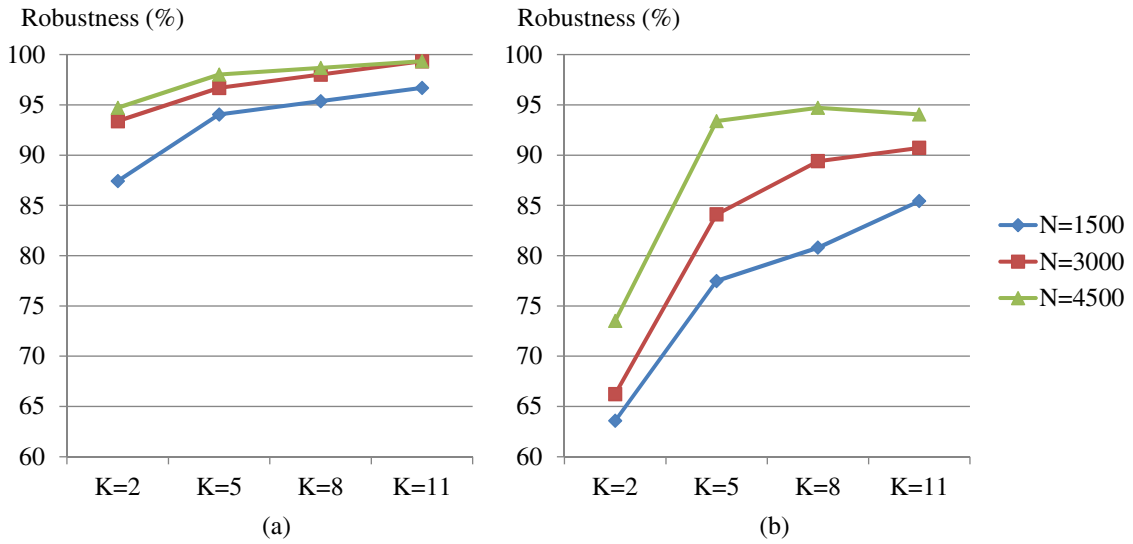


Fig. 6. The graphs represent the relation among the robustness, the number of stable keypoints, and the number of clusters. (a) Using SIFT as feature. (b) Using SURF as feature.

an experiment to find the relation between them. We estimated the pose of the object on 151 images as sample data by using our method. Both SIFT and SURF were used as local features in this experiment. The results of the sample data were classified into successful cases and failure cases. Based on the results in Fig. 6, we can summarize that the success rate varies directly with both the number of stable keypoints and the number of clusters. Besides, by having the same number of stable keypoints and the same number of clusters, it was shown that SIFT is more robust than SURF to estimate the pose of the object.

6.2 Evaluation of Accuracy

We also calculated the pose estimation errors in order to measure the accuracy of the system. To calculate the pose estimation errors, we computed the distance traveled by the reference object between two images captured by a stationary camera. Then, we compared the obtained distance with the groundtruth value measured by a motion capture system. In this experiment, 20 pairs of frames were used as input data. The parameters are set as follows: $N = 3000$ and $K = 8$. The result is shown in Fig. 7, and the mean error came out to be 1.85mm and 2.99mm using SIFT and SURF as local features respectively.

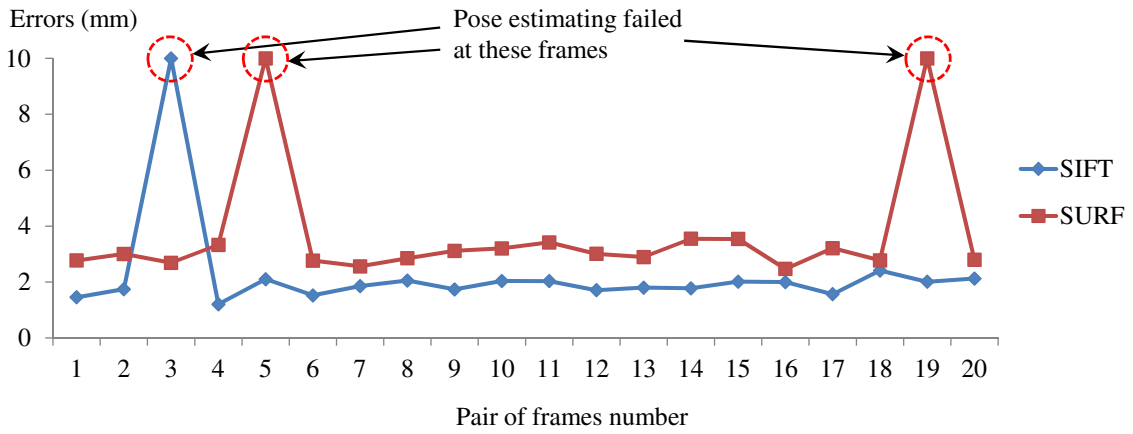


Fig. 7. The graph represents pose estimation errors between each pair of frames

7 Conclusion and Future Works

Viewpoint generative learning was successfully extended to estimate the pose of 3D object. Just giving only a textured model of an object is enough for our method to learn a set of feature descriptors from stable keypoints. After the learning, we can use conventional local feature such as SIFT or SURF to estimate the pose of a 3D object. The experimental results confirmed that our method is robust against large perspective change and even under partial occlusion.

In the future, it is possible to increase the speed of the system; most of the computation time results from comparing the descriptors, but since the descriptors are independent of each other, GPU-based parallel computing could be an interesting solution to speed up the process. Moreover, pose information of previous frame is one alternative method to decrease the computation time. Another topic that we are aware of is the problem that can be caused by the illumination changes; it is known that SIFT and SURF are only partially invariant to illumination variations. If the texture on the model and the input image are captured from different lighting environments, the keypoint matching might fail.

Acknowledgments. This work was partially supported by MEXT/JSPS Grant-in-Aid for Scientific Research(S) 24220004.

References

1. Pilet, J., Saito, H.: Virtually augmenting hundreds of real pictures: An approach based on learning, retrieval, and tracking. In: Proceedings of the 2010 IEEE Virtual Reality Conference, pp. 71–78 (2010)
2. Uchiyama, H., Marchand, E.: Toward augmenting everything: Detecting and tracking geometrical features on planar objects. In: Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality, pp. 17–25 (2011)
3. Vacchetti, L., Lepetit, V., Fua, P.: Stable Real-Time 3D Tracking Using Online and Offline Information. IEEE Trans. Pattern Anal. Mach. Intell. 26(10), 1385–1391 (2004)
4. Park, Y., Lepetit, V., Woontack, W.: Multiple 3D Object tracking for augmented reality. In: Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, pp. 117–120 (2008)
5. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision 60(2), 91–110 (2004)
6. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-Up Robust Features (SURF). Comput. Vis. Image Underst. 110(3), 346–359 (2008)
7. Yoshida, T., Saito, H., Shimizu, M., Taguchi, A.: Stable keypoint recognition using viewpoint generative learning. In: Proceedings of the 8th International Conference on Computer Vision Theory and Applications, vol. 2, pp. 310–315 (2013)
8. Hirose, R., Saito, H.: A vision-based AR registration method utilizing edges and vertices of 3D model. In: Proceedings of the 2005 International Conference on Augmented Tele-Existence, pp. 187–194 (2005)
9. Park, Y., Lepetit, V., Woo, W.: Texture-less object tracking with online training using an RGB-D camera. In: Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality, pp. 121–126 (2011)
10. Rusu, R.B., Bradski, G., Thibaux, R., Hsu, J.: Fast 3D recognition and pose using the Viewpoint Feature Histogram. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2155–2162 (2010)
11. Lepetit, V., Fua, P.: Keypoint Recognition Using Randomized Trees. IEEE Trans. Pattern Anal. Mach. Intell. 28(9), 1465–1479 (2006)
12. Daniel, K., Thomas, O., Selim, B.: Representative feature descriptor sets for robust handheld camera localization. In: Proceedings of the 2012 11th IEEE International Symposium on Mixed and Augmented Reality, pp. 65–70 (2012)
13. Arthur, D., Vassilvitskii, S.: k-means++: the advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1027–1035 (2007)
14. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A Comparison of Affine Region Detectors. Int. J. Comput. Vision 65(1-2), 43–72 (2005)
15. Moreno-Noguer, F., Lepetit, V., Fua, P.: Accurate Non-Iterative O(n) Solution to the PnP Problem. In: Proceedings of the International Conference on Computer Vision (2007)
16. INRIA image database, `http://lear.inrialpes.fr/people/mikolajczyk/Database/viewpoint.html`