Improvement of Feature-based Object Recognition Using Affine Transformation for Mobile Robot Navigation

La Tuan Anh and Jae-Bok Song
Department of Mechanical Engineering, Korea University, Seoul, Korea
(Tel : +82-2-3290-3363; E-mail: latuananh, jbsong@korea.ac.kr)

Abstract - Object recognition is very important in indoor navigation because objects are easily found in indoor environments. Feature-based object recognition methods such as SIFT and SURF are currently used because of their computational efficiency and robustness to various transformations, such as rotation, scaling, or small affinity. However, when a robot moves in the environment, a change in viewpoint of an object becomes large in most cases, which lowers the object recognition rate. To deal with this problem, this paper proposes a method for creating additional images seen from different viewpoints for the same object using affine transformation. These additional models can enhance the object recognition rate even for the case of a large change in viewpoint. Various experiments were conducted on locally planar objects such as pictures, posters, and so on. The experimental results showed that the proposed method could improve the object recognition rate significantly.

Keywords − Feature-based object recognition, SIFT, SURF, affine transformation.

1. Introduction

For indoor navigation, perception of the environment is the most fundamental task because most navigation techniques such as mapping, localization, and obstacle avoidace require sensing measurements. Among various perception data, objects are useful information because they are frequently found in the environment. Although some commercial solutions are available for object recognition, their performance depends heavily on the environmental conditions such as illumination change, occlusions, and viewpoint change, and so on. Therefore, object recognition is still a challenging subject.

Much research has been done on object recognition for the past decades. Most object recognition methods are based on detection of local features (e.g., corners, high entropy regions, scale space maxima, etc.) and construction of local descriptors for these features [1-5]. Most feature detectors involve the computation of derivatives or more complex measurements such as the second-order moment matrix for the Harris detector [1] or entropy for the salient region detector [2]. Since this process needs to be repeated in the feature coordinate space where the positions, scales, and shapes of the features are represented, local feature extraction takes a long time, which makes the object recognition slow and not suitable for real-time applications.

For mobile robot navigation which processes a huge amount of data in real time, the two scale-invariant methods, SIFT (Scale Invariant Feature Transform) [3] and SURF (Speeded Up Robust Features) [4] are widely used for object recognition because their extraction processes are efficient. Moreover, their local features are invariant to translation, scale change, and rotation of images and they are partially robust to illumination and affine transformation.

However, if the robot moves around the environment, the viewpoints of objects from the robot change rapidly and substantially. This has a great effect on the scales and shapes of local feature characteristics. A comparison of affine region detectors [5] points out that the repeatability of detected features is not good with the significant changes of viewpoint. The repeatability of all detectors varies between 40% and 80% for a viewpoint change of 20°, decreases for large viewpoint angles to 10%-46%, and it is not enough to recognize the object. Therefore, the scale invariant detectors such as SIFT and SURF always fail in the case of a significant change in viewpoint. One solution to this problem is to register some images of an object with different viewpoints in the database [3]. However, in the applications such as vision-based SLAM in which the database should be constructed on-line [6], this approach cannot be used.

This paper proposes a method which considers significant affine transformation to increase the recognition rate. The basic idea of the proposed method is to make additional images seen from various viewpoints for an identical object. Of course, these additional images are not taken by the camera, but created from the original image by some transformations. It is assumed that most objects in the indoor environment are locally planar so that the additional images are made by warping the image patches around the image points under affine or homographic transformations.

The remainder of this paper is organized as follows. Section 2 briefly overviews both SIFT and SURF algorithms which are the local feature-based object recognition methods. Section 3 deals with the geometric modeling of the associated image pairs and the construction of additional images of an identical object. Section 4 describes some experimental results. Finally, section 5 presents conclusions.

2. Local Feature-based Object Recognition

Local features have become increasingly popular over the last few years. Today, they are widely used for solving a variety of problems from wide baseline matching to the recognition of object classes. This section explains the general local feature-based object recognition scheme and the efficient object recognition methods such as SIFT and SURF.

2.1 General Approach

For object recognition, local invariant features are extracted from the object image. The descriptors of features are stored in the database, together with the pointers indicating the corresponding coordinates in the image. Up to this point, computation can be done off-line because all processes are independent of the input image before a specific input has been made.

After extracting local feature points, a matching mechanism is conducted as follows. All the descriptors of feature points in the camera image are compared with all the descriptors in the database. This matching can be conducted in an efficient way using the indexing technique which makes the object recognition less dependent on the number of objects in the database. If the camera image contains the object stored in the database, local features of the camera image will be matched with the corresponding features in the database. As a result, the matched object will probably get higher similarity than the other objects in the database, leading to the correct recognition.

For real-time applications, we focus on two feature detectors with computational efficiency. The difference-of-Gaussian detector used in the SIFT approximates the Laplacian using multiple scale space pyramids and the SURF makes use of integral images to efficiently compute a coarse approximation of the Hessian matrix. In the next section, we briefly describe these two methods.

2.2 SIFT and SURF

The scale invariant feature transform (SIFT) algorithm [3] was developed for generation of image features for object recognition. The features are invariant to translation, scaling, rotation of the image and partially invariant to illumination and affine change. As illustrated in Fig. 1, the image is smoothed several times using the Gaussian convolution mask. These smoothed images are combined in pairs to compute a DoG (Difference of Gaussian) function which is the approximation of a LoG (Laplacian of Gaussian) function. Local maxima of the DoG images are found over both space and scales. They are then refined by the quadratic interpolation to remove the outliers of local maxima and the refined local maxima are referred to as keypoints.

The SIFT descriptor uses the estimated position of keypoints and the distribution of their gradients to avoid the effects of estimation errors of the keypoints in scale or space. SIFT also uses relative strengths and orientations of gradients to reduce the effect of illumination change.

The best-bin-first (BBF) algorithm, which is based on the k-d tree search, was used for matching due to a large number of keypoints and high dimensionality of their descriptors. The BBF finds the closest matched keypoints with a high probability and enables feature matching to run in real time.

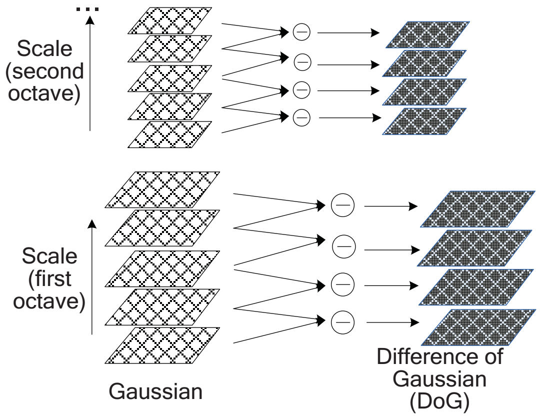


Fig. 1. Example of difference of gaussian in [3].

The speeded up robust feature (SURF) algorithm was proposed by Bay et al.[4]. The SURF detector is based on the Hessian matrix. The Hessian matrix is roughly approximated, using a set of box-type filters, and no smoothing is applied while going from one scale to the next. Figure 2 shows the Gaussian second-order partial derivative in the *y* direction and the *xy* direction and their approximation using a box filter. The box filter is calculated very fast by using the integral images and the calculation time is independent of its size. In site of rough approximations, the performance of the feature detector is compared to the results obtained with the discretized Gaussians. The properties of a SURF descriptor are similar to those of a SIFT descriptor, but the complexity of SURF is lower than that of SIFT with the help of integral images to estimate the Haar-wavelet responses.

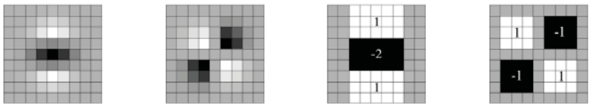


Fig. 2. Using the box filter to approximately estimate the second-order Gaussian derivatives in [4].

With the approximate approaches in both the detector and descriptor, both SIFT and SURF can be processed in real time. Hence, both methods are widely used to recognize objects for real time applications (e.g., SLAM [7]).

Although features of both methods are invariant to orientation and scale change, they usually fail when the viewpoint to the object is significantly different. In the next section, we discuss how to build an additional image from one object image to increase the recognition rate.

3. Homography between Two Images

This section deals with homography between two images of the same object with different viewpoints. Figure 3 illustrates the geometrical description of homography. Two cameras from two different poses observe a planar object. The position and orientation between two cameras are defined by the rotation matrix R and the translation vector t, respectively.

3.1 Perspective Projection

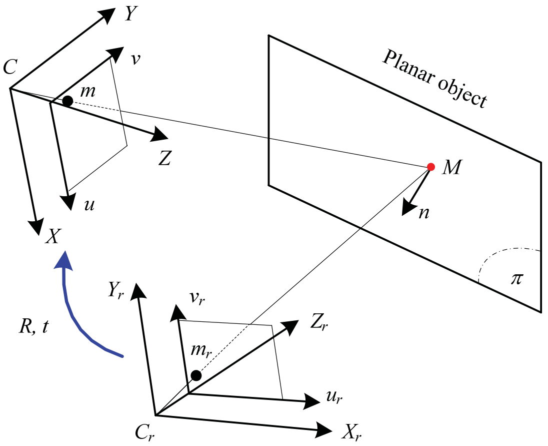


Fig. 3. Homography between two images of a plane.

Let M be a point in 3D space. Its coordinates in the frame $\{C_r\}$ is $M_r = [X_r\ Y_r\ Z_r\ ]^T$. Using the perspective projection model, the point is projected onto a camera image plane perpendicular to the optical axis and one meter away from the projection center in the point $p_r = [x_r\ y_r\ 1]^T$ determined by the following relation:

$$p_r = \frac{1}{Z_r} M_r \tag{1}$$

The real image coordinate $m_r = [u_r\ v_r\ 1]^T$ in pixels can be obtained from the following equation:

$$m_r = K\,p_r \tag{2}$$

where K is the camera intrinsic parameter matrix.

Let $R \in \mathrm{SO}(3)$ and $t \in \mathrm{R}^3$ be the rotation matrix and the translation vector between the two frames $\{C_r\}$ and $\{C\}$, respectively. In the frame $\{C\}$, the point M has the following coordinate $M = [X\ Y\ Z]^T$ and we have:

$$M = RM_r + t \tag{3}$$

In the similar way, the point M is projected onto the normalized image in $p = [x\ y\ 1]^T$ where

$$p = \frac{1}{Z} M \tag{4}$$

and is projected onto the current image in $m = [u\ v\ 1]^{\mathrm{T}}$ where:

$$m = K\,p \tag{5}$$

Let us suppose that the point M belongs to a plane π. Let n_r be the coordinate of the normal vector n in the frame $\{C_r\}$ and d_r be the distance between the plane π and the center of projection C_r ($n_r^T M_r = d_r$). If we choose the magnitude of n_r as

$$\|n_r\| = \sqrt{n_r^T n_r} = \frac{1}{d_r} \tag{6}$$

to normalize the depth component, then we can consider only the translation and rotation. From Eq. (6), we get

$$n_r^T M_r = 1 \tag{7}$$

By using Eq. (1), (3), (4), and (7), we obtain the following relationship between p and p_r

$$\frac{Z}{Z_r} p = Hp_r \tag{8}$$

where H is the homography matrix and it is given by

$$H = R + t\,n_r^T \tag{9}$$

By using equations (2), (5), and (8), we obtain the following relationship between m and m_r

$$\frac{Z}{Z_r} m = G\,m_r \tag{10}$$

where the matrix G can be given by

$$G = KHK^{-1} \tag{11}$$

Therefore, the correlation between the images of a plane is represented as a homography matrix G, which depends on the camera position, the intrinsic parameters of the camera, and the plane that contains objects [8].

If the camera and robot are well calibrated, the matrix G can be estimated. We can calculate the point m in different image coordinates (i.e., different camera pose) from point m_r by Eq. (10). In this way, other images of an object with different viewpoints can be created accurately. However, estimation errors occur due to errors in the normal vector of an object plane in the real environment and the camera calibration parameters. These errors make the proposed method difficult to apply. In the next section, an alternative method is proposed to create different viewpoint images from the reference image easier than the previous one.

3.2 Alternative synthetic image construction

The method in section 3.1 concentrated on accurate estimation and matching the feature points extracted in the images. However, the features do not have to be calculated accurately, since the goal of object recognition is not to accurately match each individual feature point, but to analyze their similarity for matching. If the object can be assumed to be locally planar, an image with a different viewpoint can be created by warping the original object image using an affine transformation which approximates the actual homography [9]. It means that G can be estimated by solving the linear equations using at least 4 points of the plane. The algorithm is as follows:

(i) Extract 4 points of the original image m_r. For simplicity, 4 corner points of the image are chosen.
(ii) Set the 3D rotation and calculate the 4 points with the rotated viewpoint. Because a robot usually moves parallel to planar objects, two important viewpoints are the ones seen from both the left and the right.
(iii) Compute the homography matrix G which maps from the 4 points in the original object image to the 4 points in the rotated image.
(iv) Apply G to all the points of the original image m_r to make the new image m with different viewpoints (Eq. 13).

$$m = Gm_r = \begin{pmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{pmatrix} m_r \qquad (12)$$

Figure 4 illustrates the images with different viewpoints which were constructed from the original image. Although the accuracy of the images is not as good as that with the method in section 3.1, the calculated images are good enough for object recognition.

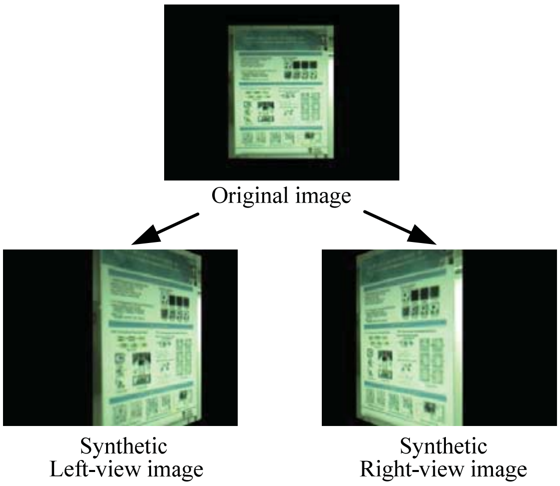


Fig. 4. Synthetic views of an object in database.

Figure 5 illustrates the overall scheme using the proposed method to help the robot detect the near objects and approach the objects. To recognize the specific objects, an object model is generated. Next, two other images with different viewpoints are made from the original image. Then local invariant features are extracted from the original and additional images, and their descriptors are stored in the database. Up to this point, computations can be done off-line. Then, the robot moves and recognizes the objects which have models in the database. When detecting the objects with the left-view or right-view models, the robot knows that it is near the objects and adjusts its position to see the object with the original model (database center). From the observed object, the robot can extract the distance and angle values for subsequent navigation steps.

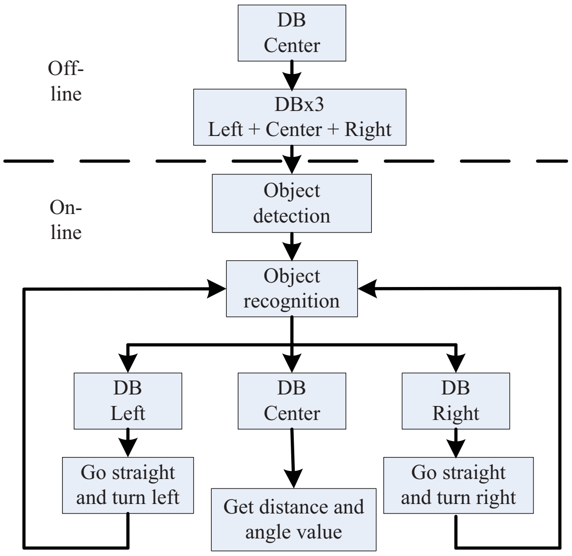
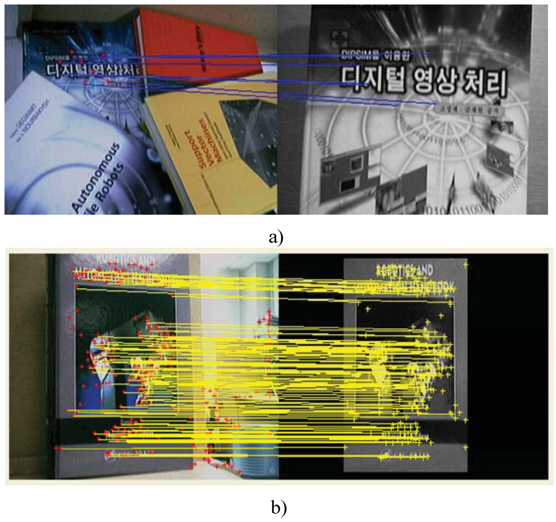


Fig. 5. Scheme for using additional database in robot navigation.

4. Experiments

We first tried both SIFT and SURF methods on planar objects. Figure 6 shows the matching results between the camera images and the images in the database using SIFT and SURF. If the viewpoint change of the objects is not great, SIFT and SURF showed good performance.



a)

b)

Fig. 6. Detection of books by (a) SIFT and (b) SURF: inlier matches established in real time.

In the next experiment, a stereo camera (Videre design STH-MDI-C) was adopted as a main sensor for object recognition. A series of experiments were conducted in the environment of a living room and an office. Figure 7 illustrates some usual objects in the indoor environment used in the experiment.

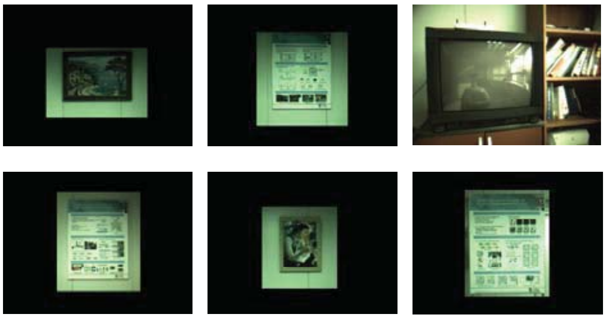


Fig. 7. Objects used as visual landmarks.

If the viewpoint change is small as shown in Fig. 8(a), both SIFT and SURF can detect the object. The center point of the object is represented as a red dot. However, when the robot wanders in the environment in most time, the viewpoint change between the robot and the object becomes large as shown in Fig. 8(b). The robot cannot recognize the object with only one reference image in the database. However, the object with large viewpoint change (over 60°) can be detected if the additional database is used as shown in Fig. 8(c).

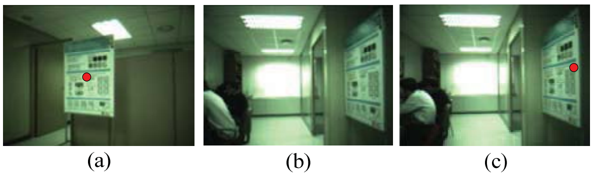


Fig. 8. (a) Small change in viewpoint from the robot to the object, (b) object detection fails due to a large change in viewpoint, and (c) object detection is successful with additional database in SURF (or SIFT).

Figure 9 shows that the processing time for object recognition does not increase significantly although the number of feature points in the database increases. That means the number of images in the database does not have a great effect on the processing time because the SIFT and SURF methods use the k-d tree-based search.

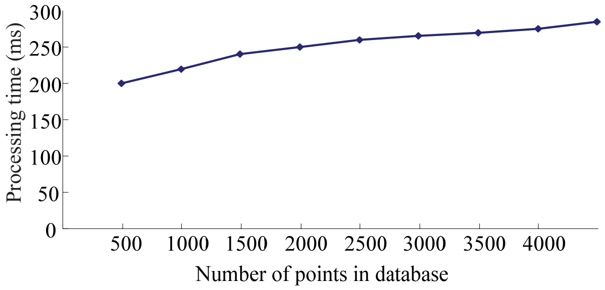


Fig. 9. Processing time as size of the database.

5. Conclusions

Local feature-based object recognition is useful for navigation of a mobile robot because objects can be used as landmarks or goals. The ability of autonomous navigation can be enhanced if the robot can recognize the objects from different viewpoints in the environment. This paper presented a method for construction of additional images seen from different viewpoints for an identical object.

Although a considerable change in viewpoint occurs, the proposed method enables a robot to recognize the objects in real-time in combination with the object recognition algorithms such as SIFT or SURF. With the same object information, the proposed method increases the efficiency of object recognition in real time navigation in indoor environments.

Acknowledgements

This paper was performed for the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Commerce, Industry and Energy of Korea.

References

[1] K. Mikolajczyk and C. Schmid. “Scale & affine invariant interest point detectors,” *International Journal of Computer Vision*, Vol. 60, No. 1, pp. 63–86, 2004.

[2] T. Kadir and M. Brady, “Scale, Saliency and Image Description,” *International Journal of Computer Vision*, Vol. 45, No. 2, pp.83-105, 2001.

[3] D.G. Lowe, “Distinctive image features from scale invariant keypoints,” *International Journal of Computer Vision*, Vol. 60, No 2, pp. 91-110, 2004.

[4] H. Bay, T. Tuytelaars, and L. van Gool, “SURF: Speeded up robust features,” *Proceeding of European Conference on Computer Vision,* pp. 404-417*,* 2006.

[5] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool, “A comparison of affine region detectors,” *International Journal of Computer Vision,* Vol. 65, No. 1-2, pp. 43–72, 2005.

[6] Y.-J. Lee and J.-B. Song, “Autonomous Selection, Registration, and Recognition of Objects for Visual SLAM in Indoor Environments,” *Proceeding of International Conference on Control, Automation, and Systems,* pp. 668-673, 2007.

[7] S. Se, D.G. Lowe, and J. Little, “Vision-based mobile robot localization and mapping using scale-invariant features,” *International Conference on Robotics and Automation*, pp. 2051–2058, 2001.

[8] O. Faugeras, and F. Lustman, “Motion and structure from motion in a piecewise planar environment,” *International Journal of Pattern Recognition and Artificial Intelligence*, Vo. 2, No. 3, pp. 485–508, 1988.

[9] R.I. Hartley, and A. Zisserman, *Multiple View Geometry*, Cambridge University Press, 2000.