PAPER *Special Issue on Machine Vision and its Applications*

Recognition of Plain Objects Using Local Region Matching

Al MANSUR , **Katsutoshi SAKATA** , **Dipankar DAS** , ***Nonmembers***, ***and*** **Yoshinori KUNO** , ***Member***

SUMMARY Conventional interest point based matching requires computationally expensive patch preprocessing and is not appropriate for plain objects with negligible detail. This paper presents a method for extracting distinctive interest regions from images that can be used to perform reliable matching between different views of plain objects or scene. We formulate the correspondence problem in a Naive Bayesian classification framework and a simple correlation based matching which makes our system fast, simple, efficient, and robust. To facilitate the matching using a very small number of interest regions, we also propose a method to reduce the search area inside a test scene. Using this method, it is possible to robustly identify objects among clutter and occlusion while achieving near real-time performance. Our system performs remarkably well on different plain objects where some state-of-the art methods fail. Since our system is particularly suitable for the recognition of plain object, we refer to it as Simple Plane Object Recognizer (SPOR).

key words: *object recognition, interest point, interest region, region matching.*

1. Introduction

Interest point detection across images is essential in many computer vision problems. The potential similarities between two images can be achieved using the local region matching technique, where the regions are centered on the detected interest points. In the context of object and scene recognition, local region matching has many advantages over global matching because it is more efficient to occlusions and viewpoint changes. Local region matching has been used for broad category of applications. Typical applications include image registration, and object detection [1, 4, 5].

Recognition of a particular object is one of the important applications of local region matching. SIFT [1] is one of the most successful methods being used for such applications. However, due to high computational overhead, SIFT is not suitable for real-time applications. Moreover, performance of SIFT greatly depends on the object type. For example, SIFT found only a single match between the images shown in Figure 1 although lot of keypoints have been found in both images. The cause of failure of SIFT on this object is that the approach used for SIFT is not robust enough to extract stable keypoints form a plain object with negligible texture/pattern content as shown here. In this paper, these types of objects are called 'plain objects'. Plain objects usually do not have much detail. Matching based on many popular interest point detectors such as [6–8] perform poorly on plain objects.

In this paper we propose a method to extract interest points from plain objects. Our approach to localize the keypoints is similar to [3]. However, the approach used in [3] for assigning orientation to the keypoints is not suitable for plain objects. We propose a different way of assigning orientations to keypoints that is applicable to plain objects. In addition to the orientations, we assign also a region lengths to the interest regions which is very effective in region matching. We call our system Simple Plain Object Recognizer (SPOR).

For region matching, our approach relies on an offline training phase. In [3] multiple views of the keypoints to be matched are used to train randomized trees [2] to recognize them based on a few pairwise intensity comparisons. However, in SPOR we used simple intensity features to train a simple Naive Bayesian classifier. As in [3], we train our classifier by synthesizing many views of the keypoints extracted from a training image to deal with scale and affine change. As the regions found in a plain object is quite similar, we also use a correlation based region matching technique in parallel to the Naive Bayesian classifier.

As the interest regions found in a plain object are very plain and texture-free, many false matching happens with the interest regions found in the background. If we could roughly estimate the location of the object, then the number of false matches would significantly reduce. We have developed a technique for representing a multicolor object using only a single color which is included in SPOR to roughly segment a plain object from the background and narrow down the search area.

It has been shown that SPOR is fast and works well on plain objects. SPOR yields both real-time performance and robustness to viewpoint and lighting changes. This makes SPOR effective for real-time object detection.

We might think that a plain object recognition is easy compared to that for complex objects. However, this may not be necessarily true, since the number of available features is small as mentioned before. Any single object recognition method might not work for all objects. Therefore, we have proposed an object recog-

The author is with the Department of Information and Computer Sciences, Saitama University, Saitama 338-8570 Japan.

nition method in which we classify object recognition cases depending on the object complexity and other attributes and recognition task, and use an appropriate object recognition method for each case [10]. The method proposed in this paper can be used for a case in this framework to recognize plain objects in the task to detect specific objects seen before.

We describe the interest region detection technique in section 2. In section 3, we discuss the region matching. Training procedure of the Naive Bayesian classifier is discussed in section 4. In section 5 we present the ways to to improve the recognition. We show the results in section 6 and finally we draw conclusions of our work in section 7.

Figure 1 Failure of SIFT to rcognize a plain object.

2. Definition of Plain Object

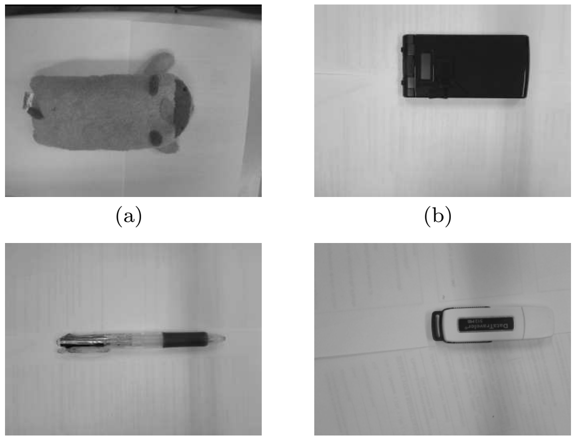
An object with following characteristics may be denoted as a plain object: (1) negligible texture content (2) location of keypoints (SIFT, Harris and similar) changes with the change of viewpoint and lighting conditions (3) Negligible sharp corners and therefore popular corner detectors do not work. Some examples of plain objects have been shown in Figure 2. To show the weakness of SIFT for plain object matching, we perform an experiment and show the result in Figure 3. In both of the images we found many keypoints. However, not a single match have been found.

3. Interest Region Detection

In our approach, we need to use a method to detect interesting regions in a plain object. Conventional interest point extraction methods such as Harris corner detection, SIFT perform poorly on this type of objects. We choose the method described in [3] to extract such regions for its speed, simplicity and stability. Such regions are identified by one or more so called keypoints.

The basic idea of [3] is to consider the intensities along a circle centered on each candidate keypoint on an interest region. Here intensities of two diametrically opposed pixels on this circle are compared with that of the candidate keypoint at the center to test whether the point is a keypoint or not.

Keypoints found at lower scales are useful for non-plain objects because there location are stable in these



(a) (b) (c) (d)

Figure 2 (a) toy with cloth surface (b) mobile telephone (c) ball pen (d) portable drive

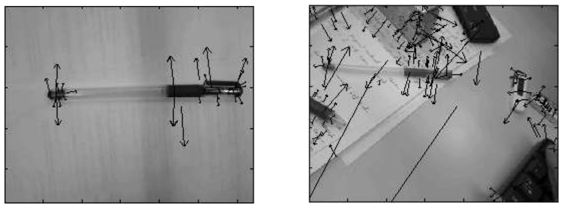


Figure 3 keypoints found on the different images of a pen

objects. However, in aplain object, locations of keypoints at lower scales are quite unstable and usually they are not placed on the interest regions found in a plain object. As we are interested in extracting the interest regions from the plain objects, we discard the lower scale keypoints and retain only the higher scale keypoints. To reduce the processing time, we extract keypoints only at two different scales. As a result we can extract interest regions at real-time. Interest regions found on a soft toy at different scales are shown Figure 4(a) and Figure 4(b).

However, the framework used in [3] to attribute the orientation to the keypoints is not suitable in our application. A plain object does not have corner like points and the locations of keypoints resulting from a plain object are not very stable. We need to assign orientation to the keypoints such that the orientations are invariant to slight changes in location. In [3] a keypoint is assigned the orientation α_m (see Figure 5)such that:

$$\alpha_m = \underset{\alpha \in [0;2\pi]}{\operatorname{argmax}} \; |I(m) - I(m + dR_\alpha)|$$

If the location of the keypoint slightly changes (which is very common in a plain object), the orientation also changes. This is shown in Figure 5.

To compute a more stable orientation, eight lines are drawn passing through a keypoint at angles from $0°$ to $360°$ degree at intervals of $45°$ degree (see Figure 4(c)). Then we calculate the length of the portion of a line $l_{\theta i}$ containing pixels of approximately the same intensities. We take the orientation θ such that:

$$\theta = \underset{\theta_i \in [0;2\pi]}{\text{argmax}} \; l_{\theta i}$$

The length of the line l_θ is also assigned to a keypoint as the region length. This helps to classify interest regions correctly.

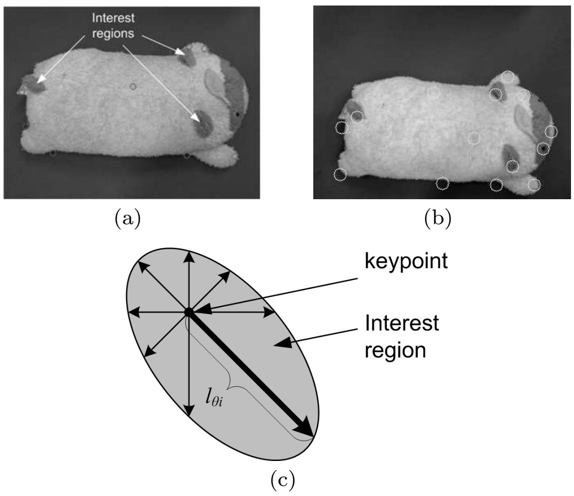


Figure 4 (a) Interest points found at lower scales are not located on the interest regions (b) Interest points found at higher scales are usually located on the interest regions (c) Computation of orientation and region length

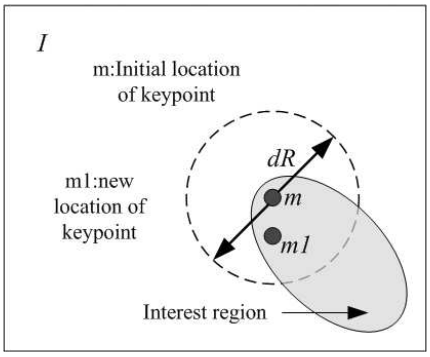


Figure 5 orientation of a keypoint changes with the slight change in location

4. Region Matching

After the feature points have been extracted from the images, two main classes of approaches can be used to achieve a matching.

In the first, computation of local descriptors invariant to changes such as perspective and lighting [1, 9] is done.

A second class uses statistical learning based techniques to model the set of possible appearances of a patch. The approach used in [4] uses PCA and Gaussian Mixture Models but does not account for perspective distortion. This has been considered in [3] using Randomized Trees.

In [3] the set of possible patches around an image feature under changing perspective and lightning conditions has been considered as a class. This approach is fast and effective to achieve a real-time performance. In region matching, a true matching between all patches is not required; it is enough to recognize some patches successfully. A robust estimator such as RANSAC can be used to detect the object.

We follow the statistical learning based technique for region matching. However, in a plain object, number of interest regions is very small and variation within these regions is small. Sometimes, number of correct matchings found from a single classifier is not enough to detect the object. To overcome this difficulty we apply a correlation based method in parallel to increase the number of correct matches.

4.1 Local Region Matching using Naive Bayesian Classifier

A class represents the set of all possible appearances of an interest region surrounding a keypoint. Our aim is to classify the interest regions found in a test image into the most likely class. Let $\mathbf{C} = \{c_1, c_2, ..., c_k\}$ be the set of K possible classes and $\mathbf{x} = \{x_1, x_2, ..., x_d\}$ is the set of continuous features extracted from a patch. Given a feature vector $\{x_1, x_2, ..., x_d\}$ our task to estimate the most probable class such that

$$\hat{c_i} = \underset{c_i}{\text{argmax}} \; P\left(C = c_i | x_1, x_2, ..., x_d\right)$$

Using Bayes' theorem, we write

$$P\left(C = c_i | x_1, .., x_d\right) = \frac{p\left(x_1, .., x_d | C = c_i\right) P\left(C = c_i\right)}{p\left(x_1, ..., x_d\right)}$$

If the prior P(C) is uniform, our problem is to find

$$\hat{c_i} = \underset{c_i}{\text{argmax}} \; p\left(x_1, x_2, ..., x_d | C = c_i\right) \qquad (1)$$

For a patch of size 20×20 the length of a feature vector d is 400. Therefore, evaluation of joint probability in Eq.(1) is not feasible. Under the "naive" conditional independence assumption, the conditional distribution over the class variable C can be expressed as:

$$p\left(x_1, x_2, .., x_d | C = c_i\right) = \prod_{j=1}^{d} p\left(x_i | C = c_i\right)$$

However, in the real world, the independence assumption may not be true. In order to meet the independence assumption, we do PCA before applying the data to the Naive Bayes classifier. By decorrelating the features, PCA makes them statistically independent. PCA also reduces the dimension of the feature vectors by removing the irrelevant features.

4.2 Local Region Matching Using Correlation

Correlation is a simple way to find the putative matching between interest regions. This can be done done by looking for regions that are maximally correlated with each other within windows surrounding each keypoint. Only points that correlate most strongly with each other are kept.

At first, from both the train and test images, images smoothed with an averaging filter are subtracted. This compensates for brightness differences in each image. Then a correlation matrix is constructed which holds the correlation strength of every point relative to every other point. Let p1 and p2 are the arrays of keypoints in the training and test image respectively. Max operation is done along rows to get strongest match in p2 for each p1 and along columns to get strongest match in p1 for each p2. Final matches are those that are consistent in both directions.

5. Training of Naive Bayes Classifier

In our application, the number of classes K is small. As a result we can easily estimate the class prior $p\left(C=c_i\right)$ by treating C as a multinomial random variable:

$$p\left(C=c_i\right)=\pi_c$$

where π is a vector containing class probabilities. The Maximum Likelihood Estimation (MLE) is done as:

$$\pi_c^{MLE}=\frac{N_c}{N}$$

where N_c is the number of training examples with class label c and N is the total number of training examples. As there is no zero counts in any class, Dirichlet prior is not required. To evaluate the class conditional densities $p\left(x_1,x_2,...,x_d|C=c_i\right)$, we assume that the parameters of each distribution is independent. Moreover, we assume that features are normally distributed. Now due to Naive Bayes assumption, we evaluate the class conditional densities as

$$p\left(x_1,x_2,..,x_d|C=c_i,\theta_c\right)=\prod_{j=1}^{d}\mathcal{N}\left(x_i|\mu_{jc},\sigma_{jc}\right)$$

$K\times d$ separate Gaussian parameters μ_{jc}, σ_{jc} have been estimated from the training data. To generate the feature vectors, we croped regions ranging from 10×10 to 20×20. Then we resized these patches to 10×10. This results in a feature vector of length 100. Using PCA feature vector dimension is reduced to 40. We use a single image and generate many new views of the object using affine deformations, and crop training patches for each class.

6. Improving Correspondence

Recognition of a plain object in a cluttered scene is highly challenging using region correspondence alone as only very few interest regions are available in such an object. As an example, we get around 5 interest regions on the experimental object shown in Figure 4(a). During recognition, many false matching occurs with the interest regions found from the background. To solve this problem, we propose a novel way to reduce the candidate area of a test scene. We also use affine solution to eliminate the outliers and to estimate the object pose.

6.1 Segmentation

We use color information of the object of interest to search on the scene for a putative match. However, for a multicolor object, it is difficult to do so. If we could represent a multicolor object using only a single color it becomes easier. To find the base color of a multicolor object we calculate the convex hulls of color regions and select the color with largest area inside the convex hull as the base color. We illustrate the process in Figure 6. Here a glass shaped container of 'ramen snack' is considered. There are two dominant colors in this object: red and yellow. We extracted both colors and computed the number of pixels in each color. In the 'Red' area there are 6712 pixels and in the 'Yellow' area there are 6661 pixels; both colors cover almost equal areas. Then, the convex hull is computed and filled for 'Red' as shown in Figure 6(f)and for 'Yellow' as shown in Figure 6(i). In this case, as the 'Red' area is larger, 'Red' is considered as the base color of the object and will be used to describe it. We use a non-plain object in this example to show the effectiveness of our method. For a plain object it is much easier.

In Figure 7 we demonstrate the segmentation process. Our task is to roughly locate the object of interest and to eliminate the interest regions coming from the background.

At first, we convert the test image from RGB color space to L*a*b* color space. Then we classify the colors in a*b* space using K-means clustering. After that we label every pixel in the image using the results from K-means. Using pixel labels, we separate the color regions and retain that region which is the most similar to the base color of the model object. Now the segmented regions are filled by computing convex hull. Now we have to search only these areas for a putative match.

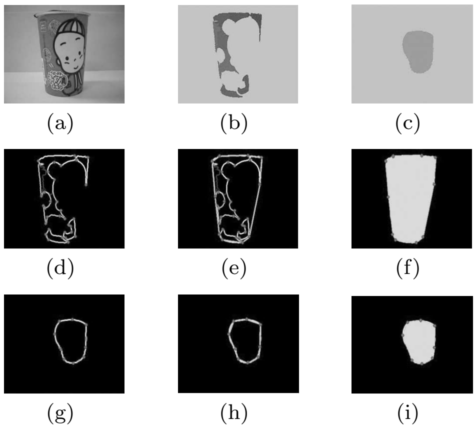


Figure 6 Base-part color detection (a) ramen snack (b) red area (c) yellow area (d) point set of red area (e) convex hull (f) area of red region (g) point set of yellow area (h) convex hull (i) area of yellow region.

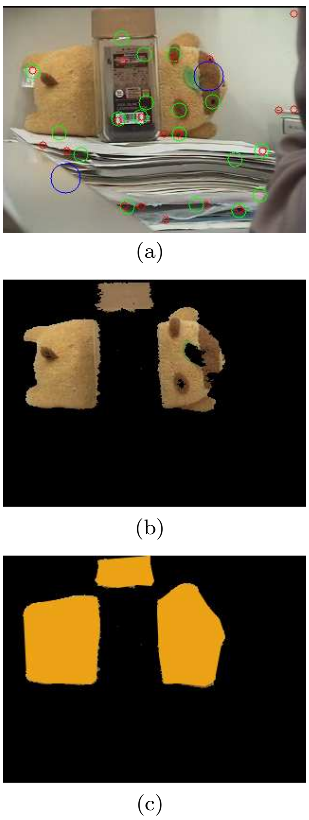


Figure 7 (a) A test scene with interest regions found both in the target and the background (b) Segmentation by base color to reduce the search area (c) Convex hull is filled to include the different color regions within the object contour.

6.2 Outlier Elimination and Pose Estimation

Sometime the interest regions found on a plain object are almost similar. As a result, false matching occurs frequently. Usually, an interest point on the model object may be matched with two or more regions on the test object. Moreover as the number of interest regions may be very few (e.g. 5), it is required to recognize the object as few matches as possible. We like to perform recognition with as few as 3 feature matches. The affine solution provides a satisfactory way to eliminate false matching and to estimate the object pose. For the examples of typical 3D objects used in this paper, an affine solution works well within a limited 2D and 3D rotation. Using a similar approach in [1], we write the affine transformation of a model point $[x\, y]^T$ to an image point $[u\, v]^T$ as:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{2}$$

where $[t_x\, t_y]^T$ is the model translation and the m_i parameters represent affine rotation, scale, and stretch. To solve for the transformation parameters, we rewrite Eq. (2) as:

$$\begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \\ & & \cdots & & & \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u \\ v \\ . \\ . \end{bmatrix}$$

This equation is written for a single match, and we need at least 3 matches to provide a solution. It can be written as:

$$\mathbf{Ax} = \mathbf{b}$$

The least-squares solution for x can be determined by solving:

$$\mathbf{x} = [\mathbf{A^T A}]^{-1}\mathbf{A^T b}$$

Outliers are eliminated by checking for agreement between each interest regions of the test object and the object model. If fewer than 3 points remain after discarding outliers, then the match is rejected. After outliers are removed, the least-squares solution is resolved with the remaining points and this process is repeated. The final decision of acceptance of a model hypothesis is found using the probabilistic model given in [2].

7. Results

We compare our system with SIFT [1], which is one of the state-of-the-art technique used for the recognition of a particular object. We show that SPOR yields much better performance than SIFT on plain objects inspite of its simplicity. Failure of SIFT is frequent on such objects. At first we compare recognition performance and then compare the processing times.

7.1 Comparison of Matching Performance

We compared the matching results of SIFT and SPOR on the test scenes shown in Figure 8. The experimental object was a soft toy with very few interest regions. The toy was presented with translation, scale change, affine transformation, and illumination changes. To produce results using SIFT, we used the code provided by David Lowe on his website, which computes the Laplacian at several levels for each octave. We did not tune any parameters of SIFT and default values were used. On the other hand, to test SPOR, we used the keypoint detector of [3] only at two scales. To compute the orientation and interest region length of the keypoints we used the method proposed in section 3. Although our method is much simpler, it performs surprisingly well as we see in the results. In Figure 8, first and second columns show the matching results found by SPOR and SIFT respectively. Matchings with occlusion, 2D rotation, change in lighting and 3D rotation is shown in the first, second, third and forth rows, respectively. SIFT fails in all cases whereas SPOR fails only in 3D rotation. In fifth row we tested both methods on another plain object.

7.2 Comparison of Speed

As comparison of speed between two algorithms depends on the codes, we tried to be fair as much as possible. To compare SPOR against SIFT we used the hybrid codes written in C++ and MATLAB as the SIFT code available to us is hybrid. The size of the input images are 320×240 in both cases. In Table 1 comparison results have been shown. In MATLAB, SPOR does not yield real-time performance. However, in C++ we achieved real-time performance from SPOR. In this comparison, we used a Intel Pentium 4, 2.8 GHz , 512 MB RAM machine with Windows XP.

Table 1 Comparison of speed between SPOR and SIFT

Platform		SIFT	SPOR
C++ and MATLAB	Feature extraction	3.55 sec	0.13
MATLAB	Feature matching	4.48 sec	0.24 sec
MATLAB	Training	N/A	1.22 sec

8. Conclusions

We proposed a method named SPOR which is very effective for the recognition of a particular plain object where many state-of-the art methods fail. The extraction of interest regions described in this paper is particularly useful in matching plain objects, which enables the correct match between a test image and the model image. This distinctiveness is achieved by assigning orientation and region length to a local region of the image. Computation of these interest regions is efficient, yielding a real-time performance on standard PC hardware. This representation is found effective in occlusion, affine distortion, scale change, and change in illumination. Usually the interest regions found in a plain object are very plain, texture-free, and vulnerable to false matching. We proposed a technique for representing a multicolor object using only a single color which is useful to segment a plain object from the background. This reduces false matching considerably. SPOR uses two parallel methods for the classification of interest regions. One is naive Bayesian classifier and the other one is correlation based matching. Future research directions include deriving invariant and distinctive image features. The feature we used in this paper is only the grayscale intensity. Further distinctiveness could be achieved using illumination-invariant color descriptors. Moreover, extensive testing is required using full 3D viewpoint and illumination changes using rich dataset of plain objects.

References

[1] D. Lowe, "Distinctive Image Features from Scale-invariant Keypoints," International Journal of Computer Vision, vol. 60, no.2, 2004, pp 91-110.
[2] D. Lowe, "Local feature view clustering for 3D object recognition," ICCV, 2001, pp 682-688.
[3] V. Lepetit and P. Fua, "Keypoint recognition using randomized trees," IEEE PAMI, vol.28,no.9, 2006, pp 14651479.
[4] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," IEEE PAMIe, vol. 28, no.4, 2006, pp 594 611.
[5] R. Fergus, P. Perona, and A. Zisserman, "A Sparse Object Category Model for Efficient Learning and Exhaustive Recognition," In Conference on Computer Vision and Pattern Recognition, July 2005.
[6] C. Harris and M. Stephens, "A combined corner and edge detector," In Alvey Vision Conference, 1988, pp 147151.
[7] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," In ICCV, 2001, pp 525531.
[8] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector.," In ECCV, 2002, pp 128142.
[9] C. Schmid and R. Mohr, "Local Grayvalue Invariants for Image Retrieval," IEEE PAMI, vol.19, no.15, 1997, pp 530-534.
[10] A. Mansur and Y. Kuno, "Selection of Object Recognition Methods According to the Task and Object Category," Proc. MVA 2007, Tokyo, Japan, May 2007.

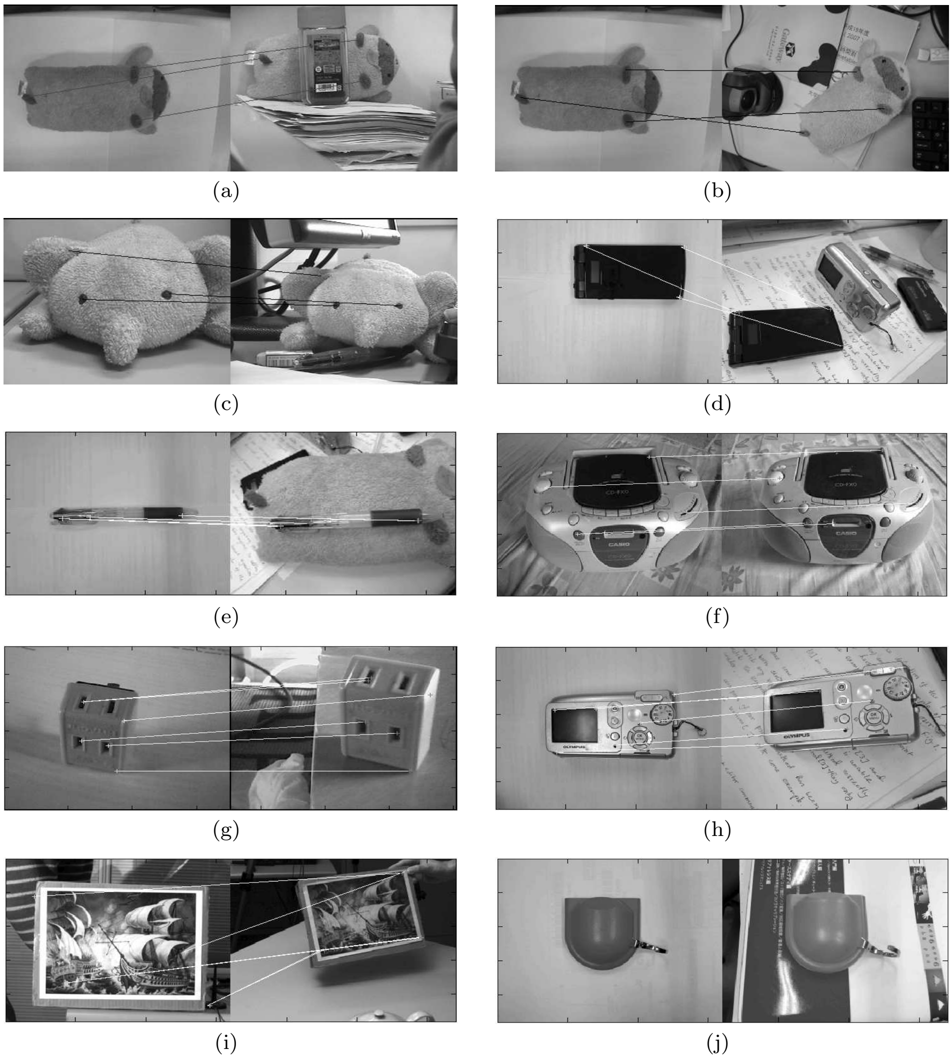


Figure 8 Comparison of recognition performance between SPOR and SIFT.

Al Mansur received the B.Sc. in Electrical Engineering from Chittagong University of Engineering and Technology, Bangladesh and M.Eng. in Telecommunications from Asian Institute of Technology, Thailand in 1999 and 2002 respectively. He is an Assistant Professor of the former university from October 2003 to continuing (now in study leave). He is currently a Ph.D. student in the graduate school of Science and Engineering, Saitama University, Japan. His research interest includes Statistical Pattern Recognition. He is now researching on robust object recognition for the service robots.

Katsutoshi Sakata received his B.Sc. degree in Information and Computer Sciences from the Saitama University, Saitama, Japan in 2007. Currently, he is a Masters student in same department of the Saitama University. His research interest is Object Recognition.

Dipankar Das received his B.Sc. and M.Sc. degrees in Computer Science and Technology from the University of Rajshahi, Rajshahi, Bangladesh in 1996 and 1997, respectively. He is an assistant professor of the department of Information and Communication Engineering of the same university. Currently, he is also a PhD student in the graduate school of Science and Engineering, Saitama University. His research interest includes Object

Recognition, Human Computer Interaction and Speech Processing.

Yoshinori Kuno received the B.S. degree, the M.S. degree, and the Ph.D. degree in 1977, 1979, and 1982, respectively, all in electrical and electronics engineering from the University of Tokyo. In 1982, he joined Toshiba Corporation. From 1987 to 1988, he was a Visiting Scientist at Carnegie Mellon University. In 1993, he moved to Osaka University as an associate professor in the Department of Computer-Controlled Mechanical Systems. Since 2000, he has been a professor in the Department of Information and Computer Sciences, Saitama University.