PAPER

Specific and Class Object Recognition for Service Robots through Autonomous and Interactive Methods

Al MANSUR[†], ***Nonmember and*** **Yoshinori KUNO**[†], ***Member***

SUMMARY Service robots need to be able to recognize and identify objects located within complex backgrounds. Since no single method may work in every situation, several methods need to be combined and robots have to select the appropriate one automatically. In this paper we propose a scheme to classify situations depending on the characteristics of the object of interest and user demand. We classify situations into four groups and employ different techniques for each. We use Scale-invariant feature transform (SIFT), Kernel Principal Components Analysis (KPCA) in conjunction with Support Vector Machine (SVM) using intensity, color, and Gabor features for five object categories. We show that the use of appropriate features is important for the use of KPCA and SVM based techniques on different kinds of objects. Through experiments we show that by using our categorization scheme a service robot can select an appropriate feature and method, and considerably improve its recognition performance. Yet, recognition is not perfect. Thus, we propose to combine the autonomous method with an interactive method that allows the robot to recognize the user request for a specific object and class when the robot fails to recognize the object. We also propose an interactive way to update the object model that is used to recognize an object upon failure in conjunction with the user's feedback.

key words: *service robot, object recognition, SIFT, KPCA, SVM, human-robot interaction*

1. Introduction

Helper or service robots have attracted the attention of researchers for their potential use with the handicapped and elderly. We are developing a service robot that can identify a specific object or a general class of objects requested by the user. The robot receives instructions through the user's speech and carries out two tasks: 1) detects a specific object, and 2) detects a class of objects. For instance, if a user asks a robot to locate a 'coke can', his/her request is for a specific object. If the user asks the robot to find a 'can', his/her request is for a class of objects. The robot needs to have a vision system that can locate various objects in complex backgrounds in order to carry out these two tasks.

There is no single object recognition method that can work equally effectively on various types of objects and backgrounds. Rather, the robot must rely on multiple methods and should be able to select the appropriate one depending on the characteristics of the object.

SIFT [1], is capable of detecting an object that the system had previously seen with an incomparable performance. Unfortunately, this method generates very few or no keypoints if the object is plain and does not have much detail. As a result, SIFT is not well suited to recognize such objects. SIFT is also not appropriate for recognizing object class.

In a recent work [2], Serre, Wolf and Poggio proposed the standard model that is suitable for class recognition. Although the results are impressive for some object categories, there are some objects for which the detection rate is not good enough.

In [3], Kernel PCA is used in conjunction with SVM (KPCA+SVM) to learn the view subspaces for multi-view face detection and recognition. In [4], Gabor-based KPCA is used for face recognition. These methods can be applied to class recognition. When KPCA and SVM are used for object recognition, feature selection is crucial. Feature selection is important in order to achieve a good recognition performance on particular classes of objects. It is also possible to construct a feature vector using multiple features, but in this case processing time for an input image is long and requires a large number of training images. Our policy in this research is to use only effective features to realize efficient and reliable object recognition method. In this paper we report on our study of intensity, color, and Gabor features and propose a way of selecting a feature depending on the characteristics of the object.

To develop an integrated object recognition platform for service robots, we split the object recognition problem into several cases depending on the task and object category. In this paper we present scenarios that have been encountered by a service robot to carry out an object recognition task and propose solutions to these challenges. There are some cases when object recognition fails. In these cases the robot communicates with a human user and the user guides it to recognize the object through short, 'user-friendly' conversation. In our application, the user is conceived of as a physically handicapped person who can speak clearly. Thus it should not be difficult for this person to interact with the robot to help it to locate the requested object. The service robot learns through failure and continuously improves its model of an object whenever it makes a mistake. The user helps it in this learning process.

Our proposed categorization scheme enables the robot to choose an appropriate detection method. Through experiments we show that a technique se-

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

lected by the categorization scheme performs better than other techniques. We introduce the categorization scheme in section 2. In section 3 we discuss the recognition framework and feature extraction. Experimental results are shown in section 4 and interactive object recognition is discussed in section 5. Finally we draw conclusions of our work in section 6.

2. Object Categorization

Most of the objects encountered by service robots can be described by their color, shape, and texture. By 'texture' we mean the pattern (not necessarily regular and periodic) within the object contour. For example, in our notation, the label on a bottle is its texture. We used three features for recognition: intensity, Gabor feature, and color. We split the objects into five categories using the object characteristics described below. Examples of each category are provided in section 4.

Category 1 (plain, simple shape objects): These objects are plain, do not contain texture, and their colors are different. They have similar shapes and this shape information is a clue to detect them. Class recognition or specific object recognition of this category is not possible using SIFT. KPCA+SVM can be used although it uses the same strategy for both class and specific object detection.

Category 2 (differently textured objects): In this category, some objects have textures although these textures do not characterize them and the texture contents of different members of the class are not the same. Some members of those classes have a texture-free body. As a result we need to use information regarding their shapes in order to describe them. Using SIFT, any specific textured object of this category can be recognized. To recognize a texture-free specific object or a class of this category we use KPCA+SVM. Since these objects are shape-based, we should use Gabor feature because it works well on objects with different textures.

Categories 3-1 and 3-2 (similarly textured objects): These objects have similar texture and texture is required for their recognition. An example of this class includes fruit (e.g. pineapple) and computer keyboard. KPCA+SVM based method works on this type of object. However, in our experiments, we found that intensity feature works better than or the same as Gabor feature for some objects of this type. For other objects of this type, Gabor feature obtains a better recognition rate. Many of the texture classification methods [12], [13], [14] use Gabor filters for feature extraction. Robust feature extraction using Gabor filters requires a large set of Gabor filters of various scales and orientation. This makes the computation huge. In this respect, intensity feature is desirable due to its simplicity and speed. Objects with similar textures in which grayscale or intensity feature has satisfactory performance are named category 3-1 objects. Gabor feature works better on other types of objects with similar texture. These are designated category 3-2 objects. Characteristics of category 3-1 and 3-2 objects and how to separate them is discussed in section 2.1.

Category 4 (similar color objects): These objects have similar color histograms. We use a combination of color and intensity features for their recognition. An example of this category is fruit, such as orange and banana.

2.1 Classification of Situations

In Table 1, we summarize the object categorization as discussed in the previous section. The object recognition problem has been classified into several cases based on task and object category.

Table 1 Categorization of an Object Recognition Scenario.

Object category	Specific or class	Case	Applicability			
			SIFT	KPCA+SVM		
				intensity	Gabor	color + intensity
1	Specific	1			●	
	class					
2	Specific (textured)	2	●			
	Specific (texture-free)	3			●	
	class	4			●	
3-1	Specific	5	●			
	class	6		●		
3-2	Specific	7	●			
	class	8			●	
4	Specific	9				●
	class					

To categorize the scenarios into one of the nine cases, we need two kinds of information: object category and object specificity. We apply the algorithm shown in Figure 1 to classify an object class into category 1, category 2, category 3-1, category 3-2 or category 4. The robot is programmed on all the objects (on which the robot works) using the algorithm prior to recognition. Finally, object specificity will be known from the robot user. Now we deploy appropriate strategies for four groups of cases as follows:

Method 1 (SIFT based): cases 2, 5 and 7
Method 2 (Gabor based KPCA+SVM): cases 1, 3, 4 and 8
Method 3 (Intensity based KPCA+SVM): case 6
Method 4 (Color and intensity based KPCA+SVM): case 9

To categorize a particular object class into one of the five categories using the given algorithm, images of different objects of the same class are required. The objects should appear in plain background. This ensures that no keypoint or feature is generated from the background. Note that this is not a recognition step and is done offline. As a result we can use images of objects with a plain background. At the first stage of the algorithm we classify the objects into two types using a threshold of SIFT keypoint count. To find the threshold we collect a sufficient number of images of plain objects. Then we extract SIFT keypoints from each of these images and take a record of these keypoint counts (label 1). We also count the SIFT keypoints for non-plain objects (label 2). Then we estimate the parameters of Gaussian mixture model for given labeled data samples, and finally we construct the decision boundary of a Bayesian classifier. This classifier has the quadratic discriminant function:

$$f(\mathbf{x}) = \langle \mathbf{x} \cdot \mathbf{A}\mathbf{x} \rangle + \langle \mathbf{B} \cdot \mathbf{x} \rangle + c$$

where $\mathbf{A}$, $\mathbf{B}$ and c are the parameters of quadratic term, linear term and bias of the model respectively. The classification strategy is

$$q(\mathbf{x}) = \begin{cases} plain & f(\mathbf{x}) \geq 0 \\ nonplain & f(\mathbf{x}) < 0 \end{cases}$$

The number of SIFT keypoints may depend on some parameters. To investigate the influence and bias of different SIFT parameters on decision procedure, we conducted experiments on one plain object (apple) and one textured object (pineapple). We changed the following SIFT parameters: contrast threshold, principle curvature ratio threshold, number of octaves, and number of scales per octave. The ranges over which these parameters were varied are as follows:

Contrast threshold: 0.01 to 0.05
Principle curvature ratio threshold: 5 to 13
Number of octaves: 1 to 5
Number of scales per octave: 1 to 5

The image size of both objects are kept same when the parameters are varied. When the contrast threshold was greater than 0.03, it is found that the number of keypoints in both objects are almost similar. However, for the other three parameters, large differences between the numbers of keypoints on two objects are found for all settings. The differences are also observed when the contrast threshold is smaller than 0.03.

Based on this preliminary experiment, we have adopted the following values for the for the four parameters:

Contrast threshold: 0.02
Principle curvature ratio threshold: 10
Number of octaves: 4
Number of scales per octave: 3

We use these parameters in all experiments described later.

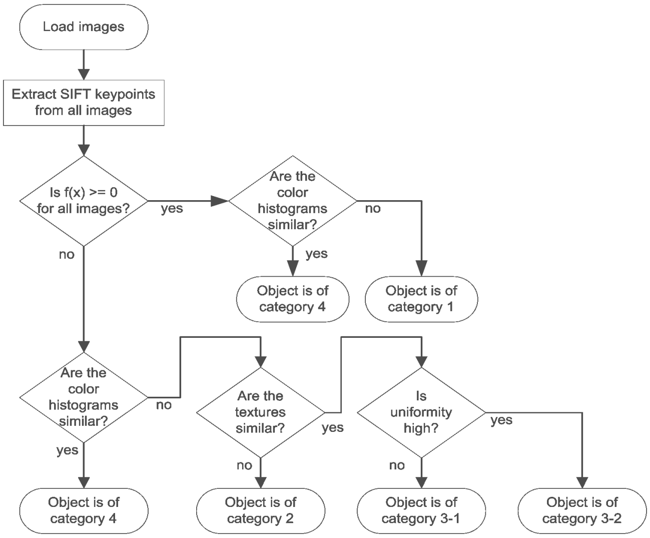

Load images
Extract SIFT keypoints from all images
Is f(x) >= 0 for all images?
yes
no
Are the color histograms similar?
no
yes
Object is of category 4
Object is of category 1
Are the color histograms similar?
no
yes
Object is of category 4
Are the textures similar?
yes
no
Object is of category 2
Is uniformity high?
yes
no
Object is of category 3-1
Object is of category 3-2

Figure 1 Object categorization algorithm.

When all the objects of a class are plain and have non-similar color histograms, they are marked as category 1 objects. On the other hand, plain objects with similar color histograms are marked as category 4 objects. Similarity in color histograms is checked by Euclidean distance between HSV color histograms.

If the objects have texture, they produce a large number of SIFT keypoints, and we need further investigation to categorize them. If the HSV color histograms are similar, they will be marked as category 4. Non-plain objects with dissimilar color histograms are further divided into two categories according to the similarity of their texture contents.

To check the similarity in texture we compute the local binary pattern (LBP) histograms [15] and calculate the log likelihood statistic [15]. Object with dissimilar textures, variance of the log likelihood statistic is higher. To get the threshold, a Bayesian classifier is trained using well known Brodatz [11] texture. This database consists of 116 different texture classes. Each of the 512×512 images is divided into 36 overlapping subimages of size 120×120. Log likelihood statistic of LBP histograms are computed from each pair of subimages. These pairs contain both similar and dissimilar textures. The threshold is learnt using the variances of the log likelihood statistics of all pairs.

Objects with similar textures are further subdivided into two categories depending on their uniformity measure, U which is given by:

$$U = \sum_{i=0}^{L-1} p^2(z_i)$$

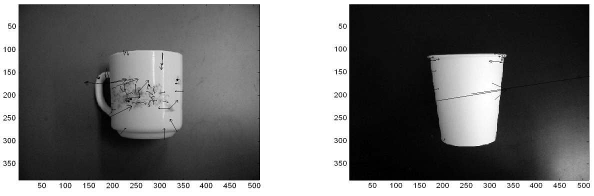
Where, L is the number of intensity levels and $p(z_i)$ is the probability that intensity level of a pixel is equal to z_i). This measure is maximum when all gray levels are equal (maximally uniform) and decrease from there.

On the Brodatz subimages, we apply both Gabor-based and intensity-based KPCA + SVM for texture recognition. We labeled the textures on which intensity feature does better than or the same as the Gabor feature as category 3-1 and the remaining textures as category 3-2. Then we built a Bayesian classifier using the uniformity as the feature. This gave us a classifier that can provide information about the robustness of intensity or Gabor feature on a particular texture type. Uniformity of category 3-2 objects are usually larger than those of category 3-1 objects.

We would like to summarize the basic points in our categorization and recognition methods. Our object recognition is based on appearance. We consider color, texture, and shape as descriptors of appearance. SIFT-based method shows a good performance for specific object recognition. Therefore, our categorization method first checks if SIFT can be used. Then, it examines if color and texture can be useful. If they are useful, appropriate feature is chosen for the object class. However, note that all recognition methods except SIFT-based one are based on KPCA and SVM. These methods implicitly use shape information. Shape changes among objects in the same class and/or those caused by viewpoints are dealt in the KPCA and SVM framework although the capability cannot be sufficient. This problem is left for future work.

2.2 Categorization Example

When the categorization algorithm is applied to 'cup' (Figure 2) it has been found that some of the sample images of this object have very few keypoints, while the others have a large number of keypoints. As a result, the discriminant function (based on SIFT keypoint count) returns positive as well as negative values. The color and orientation histograms of different 'cup' images are not similar. Consequently 'cup' has been classified as a category 2 object. The application of the categorization algorithm to 'keyboard' resulted in the finding that it is not a category 1 object and their color histograms are not similar. However, textures of different 'keyboard' examples (Figure 3) are similar (variance of log likelihood statistic of LBP histograms is low) and its uniformity is also high. As a result, this object has been categorized as category 3-2.



(a) (b)

Figure 2 (a) A textured cup has 67 SIFT keypoints (b) A plain cup has 17 SIFT keypoints.

Figure 3 Examples of 'keyboard'.

3. Recognition Framework

3.1 Method 1

We follow [1] in this method. First, the original image is progressively filtered using Difference of Gaussian filters with σ in a band from 1 to 2 resulting in a series of Gaussian blurred images. This processing produces a scale space representation. Then these images are subtracted from their direct neighbors (by σ) to produce a new series of images. Each pixel in the image is compared to its eight neighbors and the nine pixels in each of the other pictures in the series. Keypoints are then chosen from the extrema in scale space. To derive the SIFT keypoint descriptors for each keypoint, histograms of gradient directions are computed in a 16×16 window using bilinear interpolation.

Sometimes different objects have a common brand name or logos. In this case, SIFT produces an incorrect matching (see Figure 4). To solve this problem we omit those keypoints that are common in different objects.

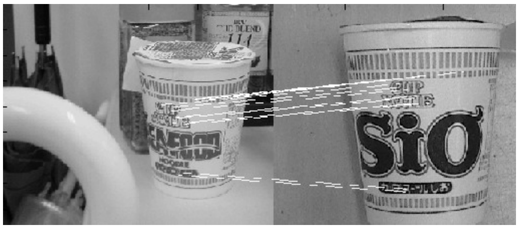


Figure 4 Incorrect matching due to common logo.

3.2 Method 2

We apply a battery of Gabor filters to each of the training and test images (grayscale) to extract the edges oriented in different directions. These filters come in four orientations with eight scales in each orientation. Let $(p_1, p_2, ..., p_m)$ be the positive images and $(n_1, n_2, ..., n_m)$ be the negative images provided for training. These images are resized to 120×120 pixels. 4×8 Gabor filters are applied to each of the positive and negative training images. We take max over the scales to provide scale invariance. Now we have four Gabor response maps. Each map contains edges in a particular direction determined by the orientation of the Gabor filter. These maps are normalized and augmented into a single column vector. We obtain KPCA-based [5] feature vectors by computing principal component projections of each orientation map of the training sample onto the nonlinear subspaces of positive and negative samples. These features are used to train a SVM classifier.

3.3 Method 3

All the test and training images are converted to grayscale and resized to 120×120 pixels. These resized images are then normalized to compensate for the effect of varying illumination. Finally they are converted into column vectors and KPCA features are derived. Then a support vector classifier is trained using the intensity values of the pixels to build the classifier.

3.4 Method 4

At first all the test and training images are resized and normalized. Then a 48-bin HSV histogram is computed from each image. Each H, S and V channel is represented by a 16-bin histogram and combined. Then a histogram normalization is done. We train an SVM classifier with these feature vectors. Another intensity-based SVM classifier is trained (as in method 3) and used to reduce the false positive results. In a test scene, the first classifier gives positive results when it finds similar colors in the background. Those regions of the test scene are further checked by the second classifier (intensity based). The regions where both classifiers agree are marked as detected objects.

4. Experimental Results

We carried out experiments in order to (1) evaluate the effectiveness of our categorization algorithm, (2) verify the hypothesis of categorization and prove the effectiveness of feature selection, and (3) evaluate the object-recognition performance in our application domain.

4.1 Evaluation of Categorization Algorithm

To validate the automatic object categorization algorithm we perform experiments on nine object classes. For each class, we use ten different objects. These objects are taken from Caltech and our own dataset (described in section 4.2) and each image is manually segmented from the background. To implement the categorization algorithm, at first SIFT keypoints are extracted from 100 images (10 images for each class). Results of these experiments are shown in Table 2. The forth column, LBP, is the pairwise log likelihood statistics of the LBP histograms.

For 'apple', 'red apple' and 'orange', number of SIFT keypoints are very small and the condition '$f(x) \geq 0$' (see Figure 1) is satisfied for all images of these three objects. The average pairwise distance between color histograms is large for 'apple' (due to several colors) and small for 'red apple' and 'orange'. Consequently, 'apple' is categorized into category 1 whereas 'red apple' and 'orange' are categorized into category 4. Although large numbers of keypoints are found on 'litchi' and 'sunflower', average pairwise distances between color histograms are found small. These two objects also have been categorized into category 4.

For the remaining five objects, we compute the LBP histograms and calculate the log likelihood statistic to check the similarity in texture. For 'cup' and 'cup noodles', variance of the log likelihood statistic is higher. This means that they do not have similar textures and consequently these two objects have been classified into category 2.

To determine the category of leopard, pineapple and keyboard we finally compute the average uniformity for these three objects. As the average uniformity for leopard is lower, it means that its gray levels are far from equal and this object is classified into category 3-1. On the other hand, pineapple and keyboards are classified into category 3-2.

4.2 Use of Appropriate Feature

This section presents recognition performance of the proposed methods obtained on two multi-class object datasets. The first one is a subset of the widely used Caltech dataset (available at www.vision.caltech.edu). Some of the images of this subset are shown in Figure 5. We have taken images of twelve different classes from this dataset. As we are interested in the recognition of household and daily life objects those are encountered by a service robot, we consider a limited object classes from Caltech dataset. Although 'leopard' is not a such type of object, we chose it to increase the number of category 3-1 objects. The negative training and test sets are collected from the internet images those are totally unrelated to the keyword category. The second dataset, which we collected from the images of fruits and home objects taken at our households and also from internet, consists of eight object classes: orange, cup noodles, coffee jar, pineapple, litchi, keyboard, can, and apple. Figure 6 shows some of the sample images of this dataset. The number of images per class is not more than forty and we randomly split each object class into two sets: training set and testing set. The first set is used for training and the second one for testing. Negative images for this dataset were taken from random background images from home and laboratory environment. The numbers of images for training and testing of each object class are shown in the last two columns of Table 3. In these columns, the left numbers indicate the numbers of positive images and the right numbers those of negative images respectively.

From the results shown in Figure 7, we can conclude that for category 2 objects (1) when intensity feature is used, detection rate is poor if the number of the KPCA components is kept below 30. It rises with an increase in the number of components. However, a false-positive rate also increases simultaneously. Therefore, intensity feature is not suitable for the detection of 'car'. (2) When the Gabor feature is used, both detection rate and false positive rate are excellent. The detection rate is almost flat, and a false-positive rate degrades if we increase the number of KPCA components beyond 30. The use of a small number of KPCA components is desirable because it minimizes training and recognition time.

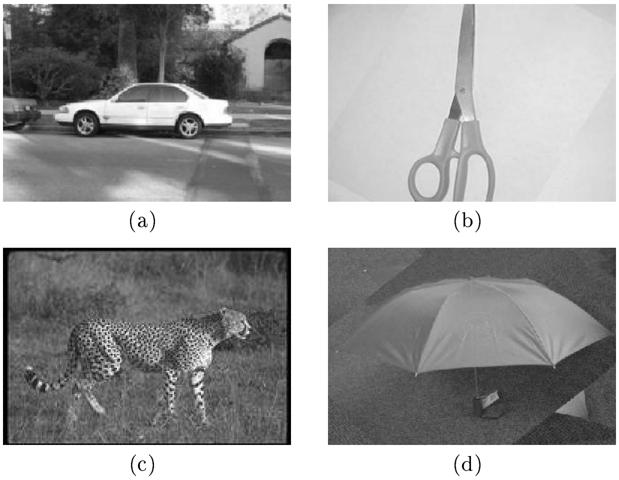

(a) (b) (c) (d)

Figure 5 Some test images from Caltech database: (a) car (b) scissors (c) leopard (d) umbrella.

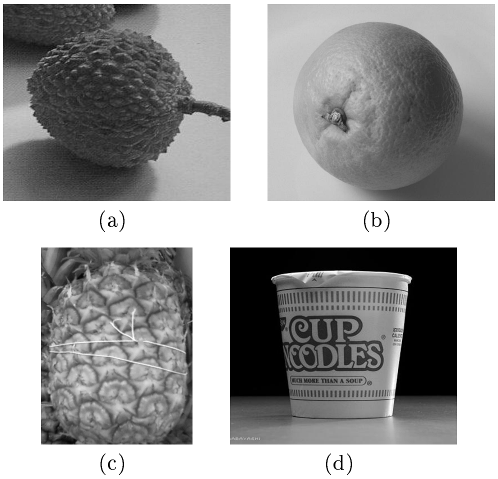

(a) (b) (c) (d)

Figure 6 Some test images from our own database: (a) litchi (b) orange (c) pineapple (d) cup noodles.

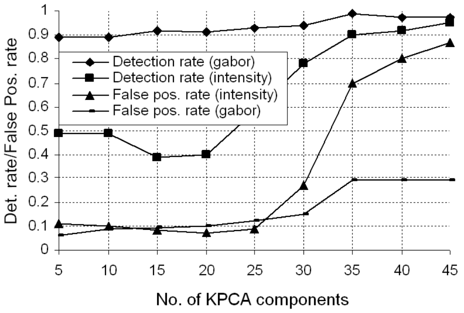


Figure 7 Recognition performance on 'car' object.

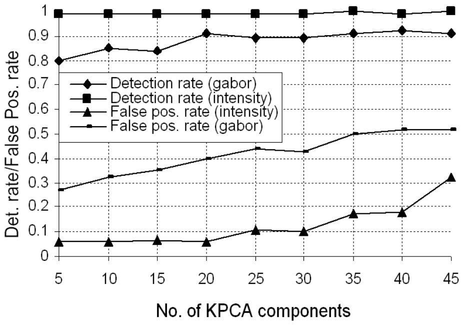


Figure 8 Recognition performance on 'leopard' object.

From the same series of experiments (Figure 8) for category 3-1 object ('leopard') we notice that: (1) Use of intensity feature results in flat and very high (99%) recognition rate. It also results in a low false-positive rate when the number of KPCA components is kept below 40. Therefore we can safely use a small number of KPCA features. (2) When Gabor feature is used, the detection rate lies above 80% but the false-positive rate becomes high. Moreover, the false-positive rate becomes worse with an increase in the number of KPCA features. In Table 3, class recognition performances of three methods are compared. From the table it is clear that (i) category 1, category 2 and category 3-2 objects are best recognized by method 2, (ii) category 3-1 objects are best recognized by method 3, and (iii) category 4 objects are best recognized by method 4. The object category shown in the second column is determined by the categorization algorithm shown in Figure 1. For categorization, we use ten representative objects from a class and segment them manually from the background. We follow the same procedure explained in section 4.1.

For all objects except category 4, color histograms are not similar within the same object class. Cropped 'leopard', produces similar histograms. However, in the Caltech dataset 'leopard' images contain much of the background. For this reason, we applied method 4 only for category 4 objects. Images of oranges are used as negative examples in the experiment with 'apple' to make the recognition challenging. For dimensionality

Table 2 Object Categorization Using the Categorization Algorithm.

object	no. of SIFT keypoints			avg. pairwise distance between color histograms	LBP			uniformity	derived category
	min	max	avg.		min	max	variance		
apple	11	46	29	407	-	-	-	-	1
red apple	14	23	18	116	-	-	-	-	4
orange	3	12	7	127	-	-	-	-	4
litchi	74	409	202	157	-	-	-	-	4
sunflower	85	146	108	152	-	-	-	-	4
cup	13	518	172	379	2.4	5.6	0.0012	-	2
cup noodles	75	166	124	423	2.83	4.75	1.2070e-004	-	2
leopard	464	770	640	352	2.64	2.73	1.5458e-010	0.00554	3-1
pineapple	297	564	450	305	4.5	4.7	1.5469e-009	0.00705	3-2
keyboard	752	934	827	348	3.57	3.7	2.7559e-009	0.00919	3-2

Table 3 Class Recognition Performance.

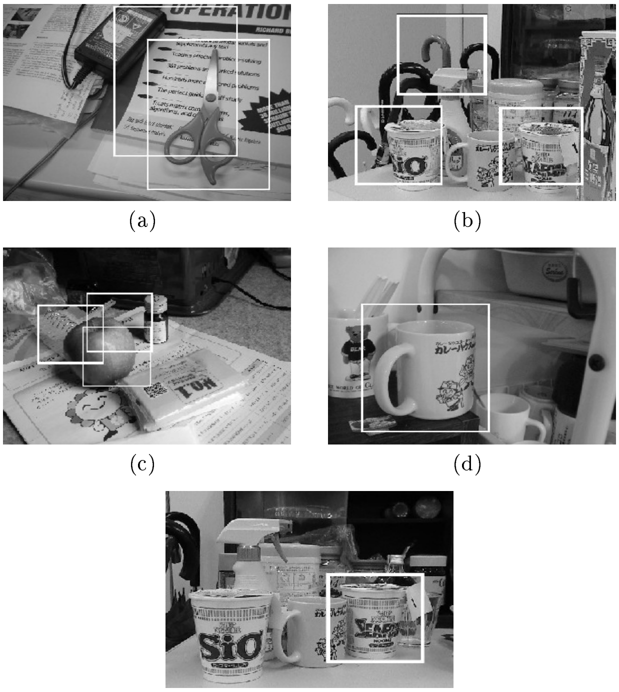
Object	Category	Method 3 (intensity based)		Method 2 (Gabor based)		Method 4 (color+intensity)		No. of training images	No. of test images
		Detection Rate	False-Positive Rate	Detection Rate	False-Positive Rate	Detection Rate	False-Positive Rate		
car	2	0.41	0.06	0.91	0.1	-	-	50+50	50+50
leopard	3-1	0.99	0.06	0.91	0.4	-	-	50+50	50+50
umbrella	2	0.7	0.3	0.86	0.21	-	-	37+50	37+50
scissors	2	0.83	0.51	0.79	0.3	-	-	25+50	25+50
pizza	2	0.7	0.59	0.93	0.4	-	-	25+50	25+50
keyboard	3-2	0.73	0.52	0.79	0.15	-	-	22+30	22+30
can	2	0.77	0.34	0.71	0.14	-	-	30+30	30+30
red apple	4	0.53	0	0.80	0.7	0.87	0.2	40+40	40+40
apple	1	0.49	0.03	0.81	0.2	-	-	20+30	20+30
cup	2	0.44	0.33	0.72	0.05	-	-	20+30	18+30
cup noodles	2	0.33	0.06	0.6	0	-	-	20+30	15+30
pineapple	3-2	0.75	0.31	1.0	0.12	-	-	20+30	16+30
watch	2	0.68	0.02	0.86	0.06	-	-	50+50	50+50
stapler	2	0.83	0.04	0.97	0.05	-	-	22+30	22+30
camera	2	0.64	0.08	0.86	0.24	-	-	25+30	25+30
coffee jar	2	0.89	0.41	0.90	0.05	-	-	20+80	20+80
dollar bill	3-1	0.97	0.01	0.88	0.05	-	-	17+40	17+40
soccer ball	2	0.65	0.14	0.99	0	-	-	20+30	20+30
orange	4	0.2	0.03	0.7	0.04	1.0	0	10+30	10+30
litchi	4	0.17	0.03	0.41	0.1	0.76	0.1	10+30	17+30
sunflower	4	0.82	0.03	0.65	0.01	1	0.1	11+30	20+30

reduction of intensity and Gabor feature vectors we retained 15 to 20 KPCA components in all experiments.

Finally in Table 4, we compare method 2 with Serre's work [2]. These four objects (all from category 2) are included in the ten worst case categories in [2]. In Serre's method 800 features were used and the training and recognition times are 1200 sec/25 images and 6 sec/image respectively. In method 2 we used only 20 features and the training and recognition times are 20 sec/25 images and 0.1 sec/image respectively. As to recognition rate, our method is comparable to Serre's method.

4.3 Object Recognition for Service Robot

We also experimented with daily objects placed in home scenes. These results are shown in Figure 9. In the first scene two bounding boxes detected a scissors. Part of the scissors is contained in both of them. Since there is some overlapping areas between these two boxes, the location of the scissors is assumed to be on the overlap. Gabor feature based KPCA+SVM is used since 'scissors' is a category 2 object and the user request was to find any available scissors (class). In Figure 9(b) the user made a request to find any 'cup noodle' without mentioning a particular choice. Since 'cup noodle' is a category 2 object, the robot used the Gabor feature based KPCA+SVM. Here the robot detected three 'cup noodles'. One of them is false positive and the other two are true positives. In Figure 9(c), an attempt to detect an apple produced three bounding boxes. Since all of them are overlapping, the robot can estimate the position of the apple. Here intensity and color-based KPCA+SVM was used since an apple is a category 4 object. In another session, the user instructed the robot to find a specific mug and a ‘seafood cup noodle’. The robot detected these objects using SIFT since both of these are category 2 objects. Figures 9(d) and (e) illustrate the results. For cup noodle, scissors, and apple, the number of training images of each class was 40.



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

Figure 9 (a)-(c) Class recognition results: (a) scissors (b) cup noodles (c) apple (d)-(e) Specific object recognition results: (d) cup (e) cup noodle.

Table 4 Comparison With [2].

Object	Serre’s method		Method 2	
	Detection rate	False-positive rate	Detection rate	False-positive rate
Watch	0.85	0.13	0.88	0.12
Ewer	0.79	0.20	0.81	0.22
Lamp	0.80	0.18	0.77	0.28
Chair	0.57	0.25	0.68	0.24

5. Incorporating User Interaction

5.1 Interactive Object Recognition

We are implementing our algorithms on our experimental robot Robovie-R Ver.2 (Figure 10) [6]. This 57 kg robot is equipped with three cameras (2 pan-tilt and one omnidirectional), wireless LAN, various sensors, and two 2.8 GHz Pentium 4 processors.

Our service robot has access to a few variants of a certain class of objects and its training set is usually small. In spite of a small training set we achieved a reasonable recognition rate. However, the recognition methods are not 100% accurate. It is desirable to improve the robot vision in any feasible way. In our application the robot user is assumed to be a physically disabled person with speaking capability. The robot is designed to help him or her bring an object upon request. When the robot fails to find the object it may ask the user to assist it to find the object using some short, ‘user-friendly’ conversation. We have already developed some interactive object-recognition methods [7–9]. In these works, we handled only single color objects in single color backgrounds where the users mention objects by their colors and shapes, not by the object names. However, in this paper, we consider real world objects in complex backgrounds where the user can mention an object by its name in a natural way.

Figure 10 Robovie: our experimental robot.

In order to implement interactive object recognition, robots have to understand the user’s instruction. We have developed the following method at present. Instructions are grouped into eight categories. In order to build a sentence pattern, words or phrases must be selected from the vocabulary list. Some words are marked as optional. We limit the vocabulary list to eliminate ambiguity during speech recognition. The user must follow the sentence structure (Table 5) and choose the words from the registered word list (Table 6) for the corresponding vocabulary type to successfully initiate the command. Optional words, though not required, provide more natural speech. For example, the user can say, “Get me a noodles.” This satisfies the grammar of ‘Object ordering: class’ and it uses the vocabulary from Phrase 1 and Object Name. Likewise, the user could also say, “May I have the Nescafe (brand name) Coffee jar?” Words not appearing in the vocabulary list may not be used. The vocabulary list are shown in Table 5. Language processing presented here is not the state of the art. We developed it for checking the effectiveness of the interactive object-recognition technique. At present, user instruction is given through a keyboard and the robot response is generated by text to speech. We will use the results developed by researchers on natural language understanding in the future.

Results of autonomous object recognition can be

Table 5 Grammar.

Purpose	Sentence structure	Example
Feedback	Feedback	Yes/No
Object Ordering: class	Phrase 1+a/an+ Object Name	Get an apple.
Object Ordering: specific	Phrase 1 + (the) + (Specifier) +Object Name (at least one 'the' or 'specifier' is required	Get my cup.
Positional information 1	Verb 1 + Adjective/ Preposition 1 + (Article) + Specifier + Object Name	Look at the left of Seafood noodle.
Positional information 2	Verb 1 + Adjective/ Preposition 1 + that + (Object Name)	Look behind that.
Positional information 3	Verb 1 + Preposition 2 + (Article) + (Specifier) + Object Name + (and) + (Article) + (Specifier) + Object Name	Look between Pepsi can and tea bottle.
Instruction to point	(Phrase 2) + Verb 2	Please show me.
Instruction to find	(Phrase 3) + Verb 3 + (Article) + Specifier + Object Name	Can you find the wooron tea bottle?

classified as shown in Figure 11. Shaded nodes represent the cases which are not handled in this paper.

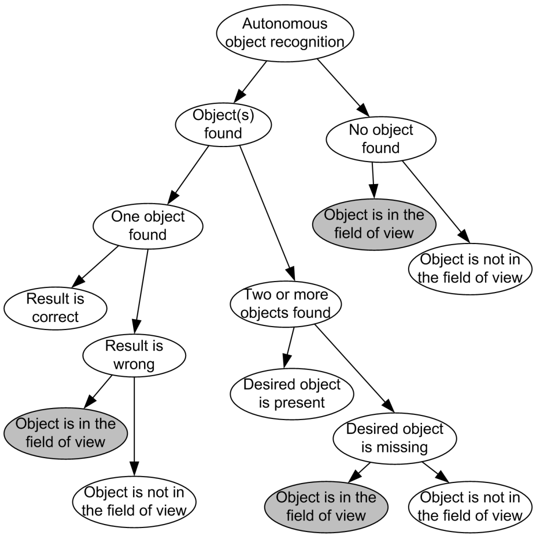

Autonomous object recognition
Object(s) found
No object found
One object found
Object is in the field of view
Object is not in the field of view
Result is correct
Two or more objects found
Result is wrong
Desired object is present
Desired object is missing
Object is in the field of view
Object is not in the field of view
Object is in the field of view
Object is not in the field of view

Figure 11 Outcomes of autonomous object recognition.

Case 1. One instance of the required object is found

Here interaction is not required since the robot successfully recognizes the desired object. If the result is wrong, this turns into case 4.

Table 6 Vocabulary.

Type	Registered words
Feedback	Yes, No
Phrase 1	May I have, Can I have, Can I get, (Please) get (me), (Please) Bring, I'd like, I would like, Give (me)
Phrase 2	Please, Could you (please), Can you (please)
Phrase 3	Could you, Can you
Verb 1	(Please) look (at/to), (Please) check
Verb 2	Show (me), Point
Verb 3	Find, See
Specifier	Green, Red, My, Coke, [brand name], etc.
Adjective	Left, Right
Preposition 1	Front, Behind, Top, Bottom
Preposition 2	Between
Object Name	Noodles, Cup, Jar, Bottle, Coffee jar, etc.

Case 2. More than one object are found (See Figure 12)

Here the user wants 'a coffee jar'. The robot uses Gabor feature based KPCA+SVM and finds two objects, one of which is a creamer jar. The robot confirms the false positive result through user interaction and then rejects the object found in the lower bounding box, and only the correct object remains. The robot updates its model of 'coffee jar' by including the image of false positive in the negative training set. If the robot makes the same mistake again, it again adds one more instance of that object. This gives more weight to that particular image. If none of the found object is true positive, situation turns into case 4.

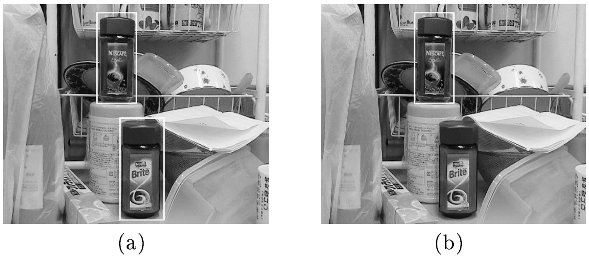


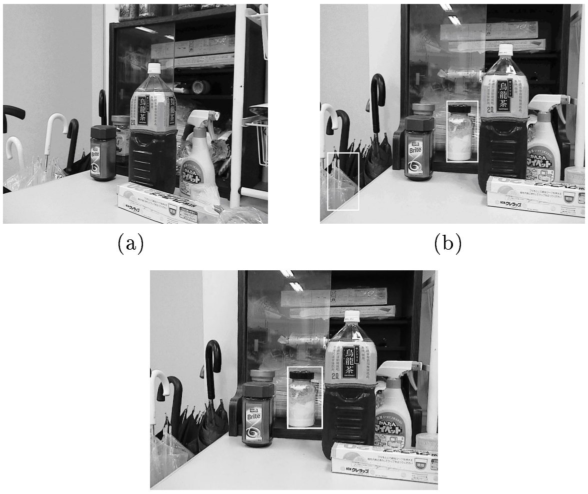
Figure 12 (a) Two objects have been found where the lower one is false positive (creamer) (b) False positive is removed through user interaction.

Case 3. No object found due to occlusion (See Figure 13)

Here the user wants the sugar jar and there is only one sugar jar in the house and the order is specific. The sugar jar is plain and only one example is available. As a result, the robot uses the color histogram for recognition. Since the object is in back of the wooron tea bottle, the robot could not find it and informs the user. The user helps the robot get it and uses some reference objects that are easy to find. In front of the robot there are two such objects: a wooron tea bottle and Brite coffee creamer. Both of these have good texture and many SIFT keypoints. The robot uses SIFT to locate them first and then follows the directions with respect to these reference objects to get the required sugar jar. When the user says 'creamer' and 'tea bottle', the robot understands 'Brite creamer jar' and 'wooron tea bottle' respectively since there is only one of each object type and those objects were mentioned before by the user in the same conversation.

Case 4. No object found although there is no occlusion

In this case, the robot cannot find the object even though the object is in the robot's field of view. The robot needs to obtain some information from the user to recognize the object. This case is not handled in this paper. We are now working on this problem. We have presented preliminary results in [10].



(a)

(b)

(c)

Figure 13 (a) The required object is not found due to occlusion (b) Two objects have been found after the robot moved to the left. The left object is false positive (c) False positive is removed through user interaction.

5.2 Learning through Failure

In interactive object recognition, we have to consider that the interaction took place earlier for a particular object should not be repeated by the robot. Therefore, the robot should learn from failures. We have developed a simple method of interactive learning. Figure 14 shows the flow. When the system cannot detect a requested object, the system uses interaction with the user to detect it. After successful detection, the system updates the model of the object by adding the image of the detected object. In our experiments, we noticed that the inclusion of even a single representative image in the training set can improve the recognition results significantly. In Figure 15 we demonstrate the effectiveness of the object model update on failure through user interaction. Here, the user requests the robot to get a coffee jar. However, the robot detected two objects, one of which was false positive (Figure 15(a)). The robot knew the correct one and wrong one through interaction with the user. The robot included the false positive image in the negative training set and ran the learning program again, thus updating the object model. Now the robot can detect the object without any false positive as shown in Figure 15(b). In this experiment, the number of training images is 30 for positive set and 80

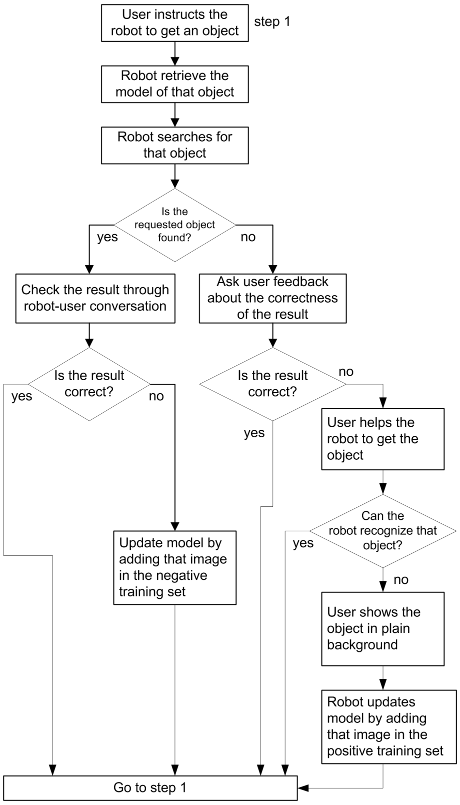
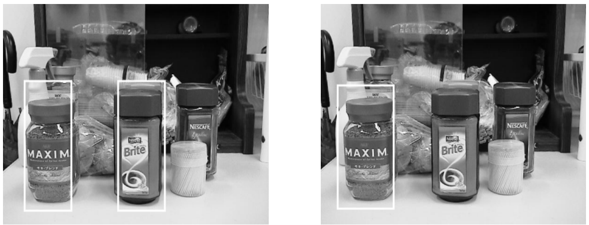

User instructs the robot to get an object
step 1
Robot retrieve the model of that object
Robot searches for that object
Is the requested object found?
yes
no
Check the result through robot-user conversation
Ask user feedback about the correctness of the result
Is the result correct?
yes
no
Is the result correct?
no
yes
User helps the robot to get the object
Can the robot recognize that object?
yes
no
Update model by adding that image in the negative training set
User shows the object in plain background
Robot updates model by adding that image in the positive training set
Go to step 1

Figure 14 Interactive learning.



(a)

(b)

Figure 15 (a) Detected object along with false positive (b) No result of false positive after inclusion of the previous false positive image in the negative training set.

for negative set.

The size of the training sets are usually small as the number of objects is limited in a house. As the robot fails more and more, the negative training set grows gradually and the influence of a single image on a large training set may have a negligible effect. Actually, different classifiers handle this issue in a different manner. We need further study on 'learning through failure' by different classifiers. At present we are investigating AdaBoost to tailor it for this purpose and to handle small positive and large negative training set.

6. Conclusions

To make a service robot's vision system work well in various situations, we have integrated several methods so that the robot can use the appropriate one. We have proposed a scheme to classify situations depending on the characteristics of the object of interest and the user's demand. It has been shown that it is possible to categorize the objects into five categories and to employ suitable techniques for each category. Our categorization scheme enables a service robot to automatically select the appropriate feature and detection method to use. SIFT and KPCA in conjunction with SVM have been employed for different categories of objects. The categorization scheme has been applied to select color, intensity, or Gabor feature to use in the KPCA based technique to achieve better recognition results. Our experimental results confirm the advantage of categorization. We have also proposed an interactive object-recognition system to recover from failure. This also makes the robot learn and update the object model from failure and improve the recognition performance continuously. Further study on interactive object recognition and learning from failures are left for future work.

References

[1] D. Lowe, "Distinctive Image Features from Scale-invariant Keypoints", International Journal of Computer Vision, vol.60, no.2, pp.91-110, 2004.

[2] T. Serre, L. Wolf, and T. Poggio, "A New Biologically Motivated Framework for Robust Object Recognition", Ai memo, 2004-026, cbcl memo 243, MIT, 2004.

[3] S. Z. Li et. al. , "Kernel Machine Based Learning for MultiView Face Detection and Pose Estimation", Proc. 8th International Conf. on Computer Vision, Vancouver, Canada, vol.2, pp.674 - 679, July 2001.

[4] C. Liu, "Gabor-Based Kernel PCA with Fractional Power Polynomial Models for Face Recognition", IEEE Trans. Pattern Anal. Mach. Intell., vol.26, no.5, pp.572-581, 2004.

[5] B. Schölkopf, A.J. Smola and K.-R. Muller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem", Neural Computation, vol.10, no.5, pp.1299-1319, 1998.

[6] Intelligent Robotics and Communication Laboratories, http://www.irc.atr.jp/index.html

[7] M.A. Hossain, R. Kurnia, A. Nakamura, and Y. Kuno, "Interactive Object Recognition through Hypothesis Generation and Confirmation", IEICE Trans. Inf.& Syst., vol.E89-D, no.7, pp.2197-2206, 2006.

[8] M.A. Hossain, R. Kurnia, A. Nakamura, and Y. Kuno, "Interactive Object Recognition System for a Helper Robot Using Photometric Invariance", IEICE Trans. Inf.& Syst., vol.E88-D, no.11, pp.2500-2508, 2005.

[9] R. Kurnia, M. A. Hossain, A. Nakamura, and Y. Kuno, "Generation of Efficient and User-friendly Queries for Helper Robots to Detect Target Objects", Advanced Robotics, vol.20, no.5, pp.499-517, 2006.

[10] K. Sakata and Y. Kuno, "Detection of Objects Based on Research of Human Expression for Objects", Proc. Symp. on Sensing Via Image Information, Yokohama, Japan, June 2007 (in Japanese).

[11] http://www.ux.uis.no/ tranden/brodatz.html

[12] D. Dunn and W.E. Higgins, "Optimal Gabor Filters for Texture Segmentation", IEEE Trans. Image Processing, vol.4, no.7, pp.947-964, 1995.

[13] A.K. Jain and F. Farrokhnia, " Unsupervised Texture Segmentation Using Gabor Filters", Pattern Recognition, vol.24, no.12, pp.1167-1186, 1991.

[14] B.S. Manjunath and W.Y. Ma, " Texture Features for Browsing and Retrieval of Image Data ", IEEE Trans. Pattern Anal. Mach. Intell., vol.18, no.8, pp.837-842, 1996.

[15] Ojala T., Pietikäinen M. and Mäenpää T., "Multiresolution gray-scale and rotation invariant texture classification with Local Binary Patterns", IEEE Trans. Pattern Anal. Mach. Intell., vol.24, no.7, pp.971-987, 2002.

Al Mansur received 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 present (now on study leave). He is currently a Ph.D. student in the graduate school of Science and Engineering, Saitama University, Japan. His research interests include Statistical Pattern Recognition. He is now researching robust object recognition for service robots.

Yoshinori Kuno received B.S., M.S., Ph.D. degrees in 1977, 1979, and 1982, respectively, 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.