THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

TECHNICAL REPORT OF IEICE.

Combining Three Different Types of Local Features for Generic Object Recognition

Yasunori KAMIYA†, Yoshikazu YANO††, Shigeru OKUMA†††,

Tomokazu TAKAHASHI†, Ichiro IDE†, and Hiroshi MURASE†

††† Graduate School of Engineering, Nagoya University Furo-cho, Chikusa-ku, Nagoya, 464-8603, Japan
†† Electronics Course, Aichi Institute of Technology 1247 Yachigusa, Yakusa-cho, Toyota, 470-0392, Japan
† Graduate School of Information Science, Nagoya University Furo-cho, Chikusa-ku, Nagoya, 464-8603, Japan
E-mail: †{kamiya,ttakahashi}@murase.m.is.nagoya-u.ac.jp, ††yoshiyano@aitech.ac.jp,
†††okuma@okuma.nuee.nagoya-u.ac.jp, ††††{ide,murase}@is.nagoya-u.ac.jp

Abstract Many types of local features have been proposed in various researches. The local features are grouped by: (1) distinguishing texture pattern; (2) area uniform in color; (3) and boundary between different colors or textures. However in generic object recogntion, previous reseaches use mainly only type (1). For improving recognition performance, we propose recognition method combining all types local feature with considering the effectivity of each type for the object. In the experiment, we show the method's effectiveness using all types of local features and compare its performance with previous works by Caltech database and Graz-02 dataset.

Key words generic object recognition, object category recognition, different types of local feature

1. Introduction

One of the big difficulty in the object recognition is the various appearances of object. Various appearances can be divided into two types. First, objects included in one category have various appearances (fig.1). For example, motorbikes vary in color, shape, and in small details such as sheet, muffler, engine, etc. Second the images are usually taken under various photo conditions such as view point(size and position of objects in images) changes, brightness differences, under shadow, and occlusions. The difference of photo conditions is another difficulty. Generic object recognition is a object recogniton field which attacks these difficulty.

Figure 1 Various appearances. (top row: difference of individual objects, bottom row: difference of photo conditions.)

Recently in many generic object recognition researches, object categories have been represented by focusing on the local areas [1]〜[3]. Several methods using local areas are extracted as small images and described by the feature values calculated from the small images. Combinations of these areas describe targeted categories. The method that extracts local areas with distinguishing texture patterns is called "detector", Many types of methods are proposed, for instance, [4], [5]. The method that describes these areas as feature values is called "descriptor," which is widely proposed, including SIFT [6],PCA-SIFT [7]. Comparisons of performance [8], [9] can be done for both methods because they are divided into two processes: detector and descriptor. Now, ways that focus on local areas with salient texture patterns are common and widely used in generic object recognition.

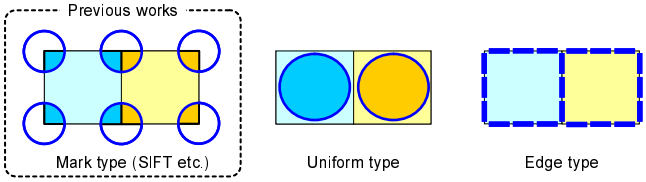

Figure 2 Imagery of each type of local feature

However, if we simply treat this way as the way that focuses on "local areas," we can find many researches that propose

other methods that focus on local areas. Let the feature that represents local areas be called "local features." The ways of detecting and describing local features are different for each method. But if we consider these methods based on the essential differences of features, we can group local features as the following three types:

- A type that deals with the areas with distinguishing texture patterns.
- A type that deals with the areas uniform in color.
- A type that deals with partial edge lines.

In this paper, we call these types "Mark type local feature," "Uniform type local feature," and "Edge type local feature." Fig2 shows imagery of each type of local feature. The circles at Mark type and Uniform type show the areas that each type focuses. The short and bold lines at Edge type show partial edge lines which this type focuses. Based on this thinking, most local features used in generic object recognition untill now (e.g., SIFT) are grouped into Mark type local feature.

In this paper we propose the recognition method including Uniform type and Edge type local features which are almost not used in generic object recognition so far. The method combines these types with considering the effectiveness of each type local feature. Generic object recognition deals with objects having many types of appearances. Therefore, for better discribing these objects, various types of describing method is needed.

The structure of this paper is as follows. An overview of the proposed method is given in Section 2. A method to calculate each type of local feature from images is described in Section 3. A learning model for each type of local feature and a combining way are described in Section 4. Section 5 describes the experiments, and we conclude in Section 6.

2. Overview of proposed method

An overview of the proposed method is described. Fig. 3 shows its process flow. First, the local features of each type are calculated from input images. The learning model for each type of local feature is learned by the calculated local features. The targeted category is described as the three learning models for each type of local feature.

This research targets two class classification. The notification that the object in input image is same with the learned object or not is the classification result. The recognizing process flow resembles the learning process flow. The local features of each type are calculated from recognition images. Each type of learning model recognizes these local features and gets three recognition results. Finally, these results are combined to get a final recognition result.

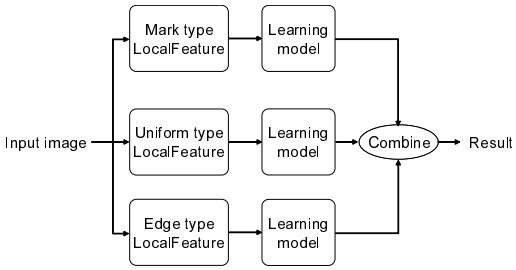

Figure 3 Overview of proposed method

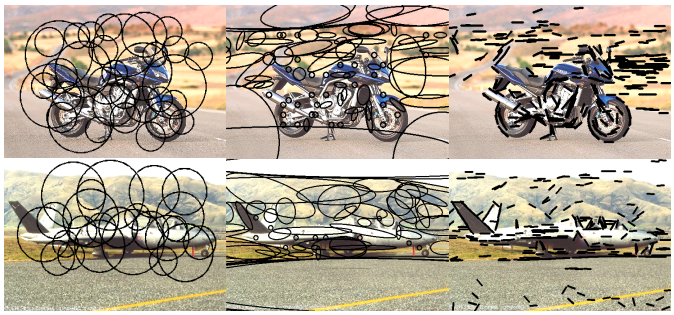

Figure 4 Examples of each extracted local feature. (left column: Mark type local feature, center column: Uniform type local feature, right column: Edge type local feature.)

3. Calculation method for each local feature

In this section we describe the way of extracting each local feature from input images. Fig.4 shows examples of each extracted local feature by the each method described in this section.

3.1 Mark type local feature

In this research, we use the detector method and the descriptor method that are often used in previous works. We use KB detector [4] for the detector which has better repeatability performance than DoG (detector of SIFT) and Discrete Cosine Transform (DCT) for the descriptor. First, the position and size of the areas in the image are detected by KB detector. Small images at each position and size are extracted to calculate feature values. After all small images are normalized to identical sizes (e.g., 10 pixels $\times$ 10 pixels), these images are represented by the first 20 coefficients calculated by DCT without DC. 23 is the number of feature values that represent local features, constructed by 20 coefficients by DCT, two values represent the position of local features (x,y), and one value represents the size of the local features.

3.2 Uniform type local feature

A model that represents color uniformity is proposed for detecting color uniform areas. Some segmentation methods based on color are commonly used. However, if they are applied on a region that includes many smaller areas with

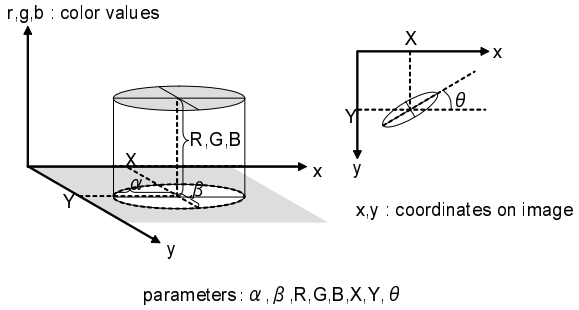

Figure 5 Model which represents color uniform area

color differences, the region is divided into many smaller areas that act like noise. Thus, we choose and propose a model-based method for color uniform area detection because such a calculation using model is not generally influenced easily by smaller value changes.

Figure 5 shows the model used. Axes x and y in Fig. 5 represent the horizontal axis and the vertical axis of the image. The "r,g,b" axis in Fig. 5 represents the color values at the pixels in the image. In Fig. 5, "r,g,b" axis represents three values by one axis for simplification of figure. The ellipse at the center of the left figure is the model representing the color uniform ellipse area. The number of model parameters is eight. R, G, and B are the color values of the uniform area that the model represents. X,Y are the positions of the ellipse area in the image. α, β represent the large and small radii of the ellipse. The right figure is the figure viewed from the left figure from overhead. θ in the right figure represents the angle of the ellipse area. Uniform areas are detected by estimating the model parameters. To detect the entire field of the image, the initial position that estimation starts is slided by small steps on the image.

We describe parameter estimation as follows. First, the initial R, G, and B values of the model are determined by the R, G, and B values at the initial positions when estimation starts. α, β are set at values that are small enough. θ is set at 0°. Parameter estimation proceeds by iteration that the parameters are modified to get the largest evaluation value that represents the matching degree between the uniform area represented by the model and the pixels of the input image inside the ellipse of the model. When increasing evaluation value is converged, estimation results. Evaluation value is shown at (1). Let I be the input image, M the model, and p the pixel position in the ellipse area of the model. This formula means counting pixels that the color is similar with the color of the model in the ellipse area. Difference of color is shown at (3). $\overrightarrow{\mathrm{RGB}}_{I,p}$ in (3) is the vector representation of the R, G, and B values at position p on input image I, and $\overrightarrow{\mathrm{RGB}}_{M}$ is the vector representation of the R, G, and B values of model M. If the difference of color is lower than threshold ϵ, the pixel is counted, and otherwise the count is decreased as penalty (2). Threshold ϵ is set up experientially. As for computation time, our implementation of this iteration process takes 3-20 sec per image on 2GHz PC.

$$f(I,M) = \sum_{p \in ellipse(M)} v(I,M,p) \tag{1}$$

$$v(I,M,p) = \begin{cases} 1 & (d(I,M,p) < \epsilon) \\ -1 & (d(I,M,p) \geqq \epsilon) \end{cases} \tag{2}$$

$$d(I,M,p) = \left| \overrightarrow{\mathrm{RGB}}_{I,p} - \overrightarrow{\mathrm{RGB}}_{M} \right| \tag{3}$$

Local features are described by model parameters. However, the angle of the ellipse is described by two values, a_1, a_2, which are calculated by our proposed converting method for keeping angle continuity. Then the number of feature values that describe local features is nine.

The formula that converts the angle to two values is shown at (4).

$$\begin{cases} a_1 = cos(2\theta) \\ a_2 = sin(2\theta) \end{cases} \tag{4}$$

3.3 Edge type local feature

The partial edge lines that resemble straight lines are extracted from edge lines obtained by the edge filter and are represented by values of angle and position. A representation way exists that uses patch images for partial edge lines, but these edge lines are mostly straight or slow curve lines. We use angle value to represent partial edge lines because finding similarities between partial edge lines is easy.

First, an input image is applied to the edge filter. We used [10] for the edge filter. Next, The position on the edge line extracted by edge filter is choosen randomly. A partial edge line candidate is determined to be the partial edge line around the chosen position. The small image (e.g., 11 pixels $\times$ 11 pixels) around the chosen position that includes the edge line of the chosen position is extracted. The angle of the edge line is calculated from the small image and judged for degree of similarity with the straight line by Principal Component Analysis (PCA). We set each pixel on the edge line in the small image as data of PCA and position of pixel (X,Y) as parameters of PCA. However there is a problem: the candidate sometimes does not have a suitable shape for angle value representation. Therefore, all edge line candidates are judged to have a degree of similarity with straight lines. There is another problem: the edge line, which is not the edge line of the chosen position, is included in the small image. To resolve this, PCA is done after removing these edge lines. The degree of similarity with straight lines is decided from two eigenvalues calculated by PCA. If $\frac{first\ eigenvalue}{second\ eigenvalue}$

Figure 6 Examples of edge lines (left: judged to be straight lines, right: judged to be not straight lines)

(these eigenvalues denote the distributions of each direction along the eigenvector) is large enough, the edge line is judged to be similar with the straight line and is used for the local feature. Fig. 6 shows examples of edge lines judged to be straight lines, and judged to be not straight lines. The angle of the partial edge line is calculated as the angle of the first eigenvector that ignores direction. The above process is iterated until the quantity of local features calculated reaches the desired quantity. Feature values that represent local features are the position (X,Y) and angle values. Angle is represented by two values that are identical to the uniform type local feature. The number of feature values is four.

4. Learning model used for each type of local feature and Combining

4.1 Learning model

In all local feature types for the combining process, we use the same learning model that is based on the method of Hillel [11]. This method is based on AdaBoost. Gaussian distribution and threshold construct weak learner.

The recognition result of learning model is calculated by (5). Let I be the input image, $h_k(I)$ the output value of the kth weak learner to input image I, α_k the weight of the kth weak learner, N the number of weak learners, ν the threshold. Also $h_k(I)$ is caluclated by (6) in this paper. Let $G(\cdot|\mu, \Sigma)$ be the gaussian distribution with average vector μ and covariance matrix Σ, $F(I)$ the set of local feature extracted from image I. The number of weak learners, N, is set before learning. α, ν, and parameters of weak learners(μ, Σ) are decided by learning.

$$H(I) = sign\left(\sum_{k=1}^{N} \alpha_k h_k(I) - \nu\right) \quad (5)$$

$$h_k(I) = \max_{x \in F(I)} G(x|\mu_k, \Sigma_k) \quad (6)$$

4.2 Combining

The final combined recognition result is calculated by summing the output values of each learning model (7). Let $f(I)$ be the value of $H(I)$ without $sign(\cdot)$, and $f_m(I)$, $f_u(I)$, $f_e(I)$ be this value for Mark type local feature, Uniform type local feature, and Edge type local feature, respectively. The learning model giving each $f(I)$ is same method, and when the type of local feature is effective for the object, value size of $f(I)$ become big, when not effecitive, become small. This combining way considers effectivity of each type.

$$H_c(I) = sign\big(\, f_m(I) + f_u(I) + f_e(I) \,\big) \quad (7)$$

Table 1 Comparison of error recognition rate to majority decision (%)

	Motorbikes	Car Rear	Airplanes	Faces	Average
majority decision	1.16	2.30	0.30	0.00	0.94
combining by (7)	0.58	1.82	0.20	0.00	0.65

In experiment section, for confirming effectiveness of this combinig way, it is compared to majority decision (8).

$$H_c^{'}(I) = sign\big(\, H_m(I) + H_u(I) + H_e(I) \,\big) \quad (8)$$

5. Experiment

First, we compare the combining way. Next, we compare the recognition results that only use one type, two types, and three types of local features. Finally, we compare the recognition results using three types of local features and the recognition results of previous works. We use the Caltech database [1] and Graz-02 [12] as datasets.

The Caltech database has been used by many previous works [1], [11], [13]. This image set contains four sorts of object images: Airplanes (1074), Car Rears (1155), Motorbikes (826), and Faces (450). Also two background images are contained: general background images (900) and background images for Car Rears (1370). Example images are shown in fig.7. The objects in these images have similar direction and position but their appearances widely differ. Also the size of objects in the images of Car Rears and Motorbikes widely differs.

We use Graz-02 for the more difficult dataset than the Caltech database. This dataset includes three sorts of object images: Bikes (365), Persons (311), and Cars (420). Also one sort of background image is included. Example images are shown in fig.8. The direction and position of objects in the images are different, and the background appearances in object images are not simple.

We set the threshold of the uniform local feature type to 40. On the edge local feature type we set the size of small images to 11 pixels x 11 pixels and the number of local features to 500. The edge line where value $\frac{first\ eigenvalue}{second\ eigenvalue}$ is larger than 20 was determined to be a straight line. In the combining way when calculating recognition results using one type of local feature and any two types of local features, the output value of the learning model for the unused local feature type was assumed to be 0. In addition we determined the number of weak learners to be 50, which is identical to [11]. Half of the object and background images are for learning and the rest are for tests. Also we don't use the information which represents locations of objects in images such as bounding boxes for learning.

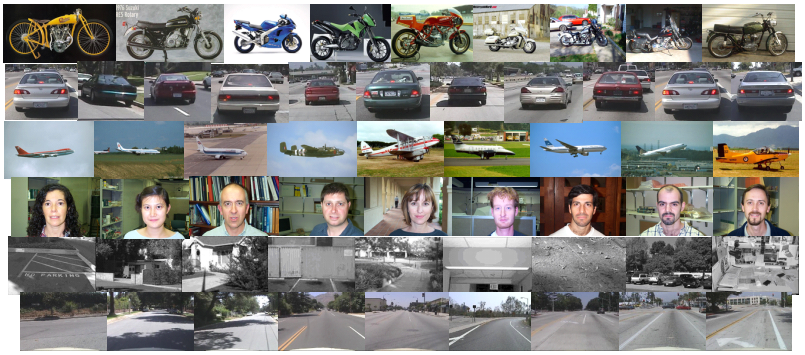

Figure 7 Example images of Caltech database. (Motorbikes, Car Rear, Airplanes, Faces, General background images, Background images for Car Rear)

Figure 8 Example images of Graz-02 dataset. (Bikes, Persons, Cars, Background images)

Table 2 Error recognition rate for Caltech database (%)

Type of local feature	Motorbikes	Car Rear	Airplanes	Faces	Average
Mark	5.80	4.51	1.42	0.00	2.93
Uniform	1.05	4.51	0.81	0.00	1.59
Edge	2.09	10.77	0.81	0.15	3.46
Mark & Uniform	1.28	2.53	0.30	0.00	1.03
Mark & Edge	1.39	2.77	1.02	0.00	1.30
Uniform & Edge	0.70	3.48	0.20	0.00	1.10
Mark & Uniform & Edge	0.58	1.82	0.20	0.00	0.65

Table 3 Comparison of error recognition rate with previous works (%): mark (M), uniform (U), edge (E)

	Our method	Hillel [11]	Fergus [1]	Opelt [13]	Opelt [14]
Type of local feature	M&U&E	M	M	E	M&E
Motorbikes	0.58	4.9	6.7	3.2	0.0
Car Rears	1.82	0.6	9.7	0.5	0.5
Airplanes	0.20	6.7	7.0	2.6	2.9
Faces	0.00	6.3	3.6	1.9	0.3
Average	0.65	4.62	6.75	2.05	0.93

5.1 Experimental results

First we compare combining way,(7) and majority decision by the Caltech database. Table 1 shows this comparison. For all objects without faces in which error recognition rate is already 0% under conditions of majority decision, the error recognition rates by (7) are lower than majority decision.

Next we considered the recognition results for the Caltech database and confirmed the effectiveness of combining Mark type, Uniform type and Edge type to improve recognition performance. Table 2 shows each error recognition rate under the following conditions: one type of local feature, any two types of local features, and three types of local features. The error recognition rate for each object and the average error recognition rate for all are also shown. We compare mainly the results by using the average error recognition rates because in generic object recognition, not the recognition performances to individual objects but a general performance to various objects is important. For comparing each average error recognition rate, the result when using Mark, Uniform, and Edge types of local features is lower than using Mark type only.

Table 3 shows the error recognition rates of the our method and previous works that use the Caltech database for experiment datasets. Table 3 shows that the error recognition rate

Figure 9 Examples of correctly classified bike images (top row) and incorrectly classified bike images (bottom row) in Graz-02

Table 4 Error recognition rate for Graz-02 (%)

Type of local feature	Bikes	Persons	Cars	Average
Mark	25.74	23.70	32.25	27.23
Uniform	36.46	26.01	28.75	30.41
Edge	24.66	22.25	36.00	27.64
Mark & Uniform	27.35	22.54	29.25	26.38
Mark & Edge	23.06	22.54	28.25	24.62
Uniform & Edge	25.47	23.12	30.25	26.28
Mark & Uniform & Edge	22.79	20.81	28.50	24.03
Opelt [12]	22.2	18.8	29.5	23.5

of our proposed method is lower than these previous works. This result shows that recognition performance can be improved by combining Mark, Uniform, and Edge type local feature.

Table 4 shows the recognition results for Graz-02 which is more difficult dataset than the Caltech database. The error recognition rate is lower than result of Mark type only, which is identical to the Caltech database results. For reference results of [12] shows in table 4. [12] proposed the recognition method combining two types local feature, Mark type and Uniform type local feature. However best combinaiton of detector and discriptor is additionally considered. In addition, we consider that these recognition rates are near the improvement barrier. Because of dateaset composition, this dataset consists of normal difficulty images and few very high difficulty images. Fig. 9 shows examples of correctly classified bike images and incorrectly classified bike images.

6. Conclusions and future work

We grouped local features based on the essential differences of features, and proposed the recognition method that added other types of local features which are almost not used in generic object recognition until now. Experimental results show the effectiveness of improving recognition performance using three types of local feature and supplementing each other type. In addition, we compared the recognition performance of our proposed method and previous works.

For future work, we are considering the more advanced combining method which can deal difference of local feature adaptively.

References

[1] R. Fergus, P. Perona and A. Zisserman: "Object class recognition by unsupervised scale-invariant learning", Proc. CVPR (2003).

[2] L. Fei-Fei, R. Fergus and P. Perona: "Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories", CVPR Workshop on Generative-Model Based Vision (2004).

[3] A. Holub, M. Welling and P. Perona: "Combining generative models and Fisher kernels for object recognition", Proc. ICCV (2005).

[4] T. Kadir and M. Brady: "Saliency, scale and image description", International Journal of Computer Vision(IJCV), **45**, 2, pp. 83–105 (2001).

[5] K. Mikolajczyk and C. Schmid: "Scale and affine invariant interest point detectors", International Journal of Computer Vision(IJCV), **60**, 1, pp. 63–86 (2004).

[6] D. G. Lowe: "Object recognition from local scale-invariant features", Proc. ICCV (1999).

[7] Y. Ke and R. Sukthankar: "Pca-sift: A more distinctive representation for local image descriptors", Proc. CVPR (2004).

[8] 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(IJCV), **65**, 1/2, pp. 43–72 (2004).

[9] K. Mikolajczyk and C. Schmid: "A performance evaluation of local descriptors", IEEE Transactions on Pattern Analysis and Machine Intelligence(PAMI), **27**, 10, pp. 1615–1630 (2005).

[10] P. Meer and B. Georgescu: "Edge detection with embedded confidence", IEEE Transactions on Pattern Analysis and Machine Intelligence(PAMI), **23**, 12, pp. 1351–1365 (2001).

[11] A. B. Hillel, T. Hertz and D. Weinshall: "Efficient learning of relational object class models", Proc. ICCV (2005).

[12] A. Opelt, A. Pinz, M.Fussenegger and P.Auer: "Generic object recognition with boosting", IEEE Transactions on Pattern Analysis and Machine Intelligence(PAMI), **28**, 3, pp. 416–431 (2006).

[13] A. Opelt, A. Pinz and A. Zisserman: "A boundary-fragment-model for object detection", Proc. ECCV (2006).

[14] A. Opelt, A. Pinz and A. Zisserman: "Fusing shape and appearance information for object category detection", Proc. British Machine Vision Conference(BMVC) (2006).