GENERIC OBJECT RECOGNITION BY GRAPH STRUCTURAL EXPRESSION

Takahiro Hori, Tetsuya Takiguchi, Yasuo Ariki

Graduate School of System Informatics, Kobe University
1-1 Rokkodai, Nada-ku, Kobe, Hyogo Pref., Japan 657-8501

ABSTRACT

This paper describes a method for generic object recognition using graph structural expression. In recent years, generic object recognition by computer is finding extensive use in a variety of fields, including robotic vision and image retrieval. Conventional methods use a bag-of-features (BoF) approach, which expresses the image as an appearance frequency histogram of visual words by quantizing SIFT (Scale-Invariant Feature Transform) features. However, there is a problem associated with this approach, namely that the location information and the relationship between keypoints (both of which are important as structural information) are lost. To deal with this problem, in the proposed method, the graph is constructed by connecting SIFT keypoints with lines. As a result, the keypoints maintain their relationship, and then structural representation with location information is achieved. Since graph representation is not suitable for statistical work, the graph is embedded into a vector space according to the graph edit distance. The experiment results on an image dataset of 10 classes showed that, the proposed method improved the recognition rate by 14.08%.

Index Terms— generic object recognition, graph, SIFT, graph edit distance, vector-space embedding

1. INTRODUCTION

Generic object recognition means that the computer recognizes objects real world images by their general name (see Fig. 1). It is one of most challenging tasks in the field of computer vision. Regarding the achieving of near-human vision by a computer, it is expected that any such technology will be applied to robotic vision. Moreover, due to the spread of digital cameras and the development of high-capacity hard disk drives in recent years, it is getting difficult to classify and to retrieve large-volume videos and images manually. Therefore, computers are being looked at to assist in automatically classifying and retrieving videos and images. In particular, generic object recognition is becoming more and more important.

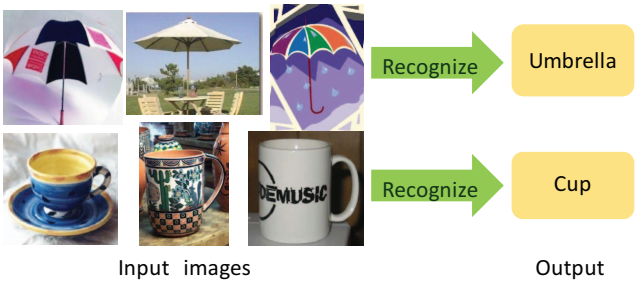


Fig. 1. Generic Object Recognition

There have been two typical approaches in the past concerning general object recognition. One is a method based on image segmentation. This is a technique for automatic annotation to the segmented image area, word-image-translation model by Barnard [1, 2, 3]. However, when the image has occlusion and the image segmentation fails, it becomes difficult for this technique to work correctly.

On the other hand, to solve this problem, a method based on the local pattern is proposed. This is a technique for collating the image by combining local features of the image. The technique for characterizing the entire image is often used for the appearance frequency histogram of the localizing features (known as Bag of Features, as shown in Fig. 2 [4]). However, there is a problem with this approach because the location information and the relationships between keypoints are lost.

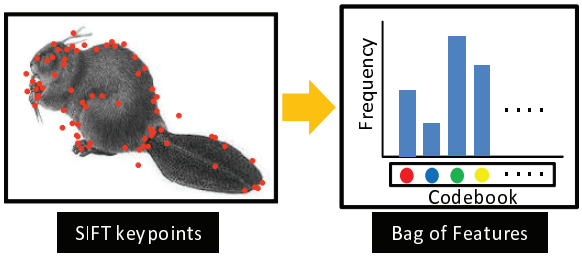


Fig. 2. Bag of Features

To deal with this problem, we propose a method in this paper to connect keypoints with lines, as shown in Fig. 3, and to express the sets of the local features as a graph. Moreover, we propose a technique with high recognition performance that integrates the object structure and the local features by embedding the graph into a vector space using the graph edit distance (GED). Thus, the objects are expressed by a simple vector of the statistical work, and trained and classified by Support Vector Machine (SVM). The results of our object recognition experiments show the effectiveness of our method.

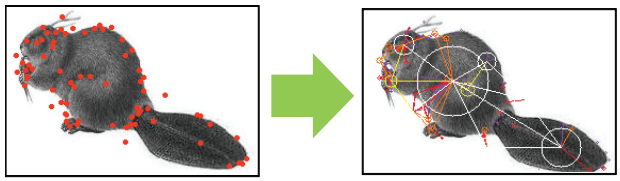


Fig. 3. Graph construction

This paper is organized as follows. In Sections 2, 3, 4, and 5, the proposed method is described. In Section 6, the performance of the

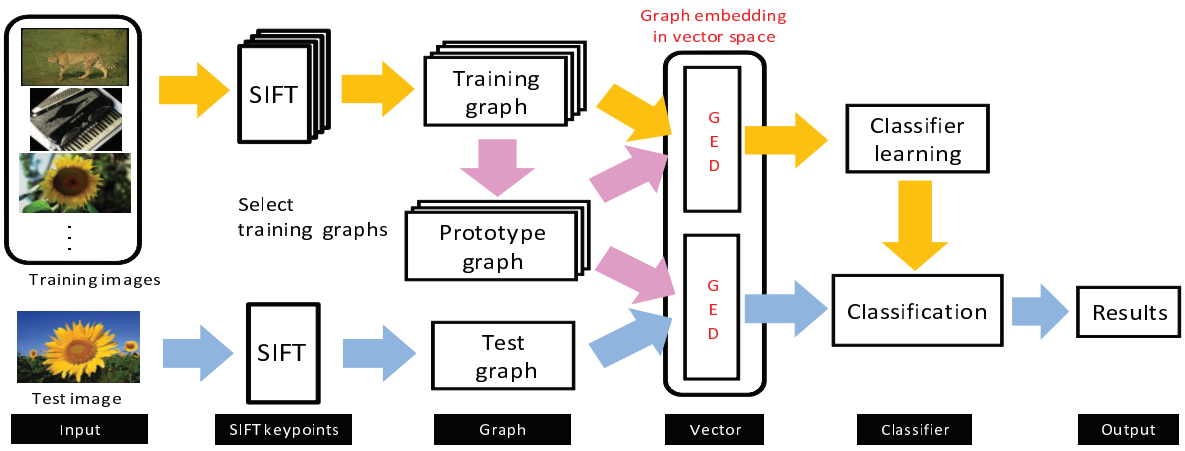


Fig. 4. System overview

proposed method is evaluated for a 10-class image dataset. Section 7 provides a summary and discusses future work.

2. OVERVIEW OF THE PROPOSED METHOD

Fig. 4 shows the system overview. First, the SIFT keypoints and features [5] of all images are extracted. The extracted keypoints are connected and a graph of each image is constructed. The graphs constructed from the training images are called training graphs and those of the test images test graphs. Next, n prototype graphs are selected from the training graphs, and GED is calculated n times between the prototype graphs and each graph (training graphs and test graphs). Thus, the graphs are embedded into an n-dimensional vector space. The classifier is trained by this n-dimensional vectors of the training images. Finally, the test data is classified by the trained classifier and the recognition result is output. In the following sections, each process in the proposed method is described in detail.

3. GRAPH STRUCTURAL EXPRESSION

In this paper, we use the notation and structural representation of the graph proposed in [6]. In the formalization, the graph is noted as $G = (V, E, X)$ where E represents the set of edges, V is the set of vertices and X the set of their associated unary measurements (in our case, a SIFT descriptor). The node is a keypoint detected by SIFT, and the associated unary measurements represent the 128-dimension SIFT descriptor of the corresponding keypoints. The edge $e_{\alpha\beta} \in E$ connects two nodes $u_\alpha \in V$ and $u_\beta \in V$. Hence, prototype graphs G^p and other graphs (called scene graph G^s) are distinguished.

3.1. Proximity Graph

It is a complete graph if all the keypoints extracted from the image are connected mutually by the edges. However, it is not usually suitable for the calculation. Additionally, because the relationships between keypoints over a long distance are weak, it is preferable to connect them only within their "neighborhood." Thus, we simply define the proximity graph as a graph in which distant keypoints are not connected. Formally, we restrain the set of edges to:

$$E = \left\{ e_{ij} \;\middle|\; \forall i, j \;\; \frac{\|\mathbf{p}_i - \mathbf{p}_j\|}{\sqrt{\sigma_i \sigma_j}} < \chi \right\} \tag{1}$$

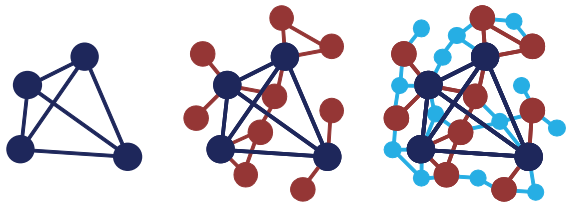
where $\mathbf{p} = (p_x, p_y)$ denotes a keypoint position, σ its scale, and χ is a constant. By this definition, the larger scale keypoints connect to the more distant keypoints. The edge is not drawn in case where the value is longer than constant χ. Because an extra edge is not drawn by this constraint, the constructed proximity graph reduces the computation load considerably, and improves the detection performance at the same time. Both the prototype graphs and the scene graphs are constructed as proximity graphs.

3.2. Pseudo-Hierarchical Graph

In general, when the scale is large, the SIFT features show high reliability. Therefore, the proximity graph is divided into the hierarchy by the size of the scale of the keypoints. This is defined as a pseudo-hierarchical graph. The improvement in recognition and computation performance is achieved by starting the graph matching from a hierarchical level that has high reliability, and going down the hierarchy gradually. We decompose the graph into a set of subgraphs $\{G_l\}_{l=1}^{L}$ based on the scale of the keypoints. For each level l, only the features whose scale is superior to a threshold s_l are retained.

$$s_l = \sigma_{min} \left(\frac{\sigma_{max} - \sigma_{min}}{\sigma_{min}} \right)^{\frac{L-l}{L-1}} \tag{2}$$

where σ_{max} and σ_{min} are the maximum and minimum scale of the keypoints in each graph. Fig. 5 shows an example of subgraph $\{G_l\}_{l=1}^{3}$ divided into three hierarchical levels.



(a) G_1 (b) G_2 (c) $G_3 = G$

Fig. 5. Pseudo-Hierarchical Graph

4. GRAPH EDIT DISTANCE

The process of evaluating the structural similarity of two graphs is generally referred to as graph matching. This issue has been addressed by a large number of studies [7]. We use the graph edit distance [8, 9], one of the most widely used methods, to compute the difference between two graphs [10, 11, 12]. A pseudo-hierarchical graph is employed in order to improve the computational complexity of graph edit distance.

The basic idea of the graph edit distance is to define the difference of two graphs as the minimum amount of edit operations required to transform one graph into the other. Namely, it is computed using the number of edit operations composed of insertion, deletion, and substitution of nodes and edges. Two Graphs G_1 and G_2 have the edit path $h(G_1, G_2) = (e_{d1}, ..., e_{dk})$ (each e_{di} indicates the edit operation) to convert G_1 into G_2 using specific editing. Fig. 6 shows the example of an edit path between two graphs G_1 and G_2. Each edit cost c is defined as the amount of the distortion in the transformation. The graph edit distance between graphs G_1 and G_2 is computed as

$$d(G_1, G_2) = \min_{(e_{d1},...,e_{dk}) \in \mathbf{h}(G_1,G_2)} \sum_{i=1}^{k} c(e_{di}) \tag{3}$$

where, $c(e_d)$ denotes the penalty cost of the edit operation e_d.

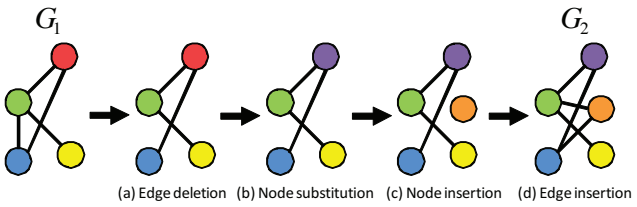


Fig. 6. An edit path between two graphs G_1 and G_2

5. GRAPH EMBEDDING IN VECTOR SPACES

The embedding method used in this paper follows the procedure proposed by [13]. This technique is used for the computation of the median graph etc., and the effectiveness is shown in [14, 15]. The approach is described as follows, and the outline is shown in Fig. 7. We prepare a set of the training graphs

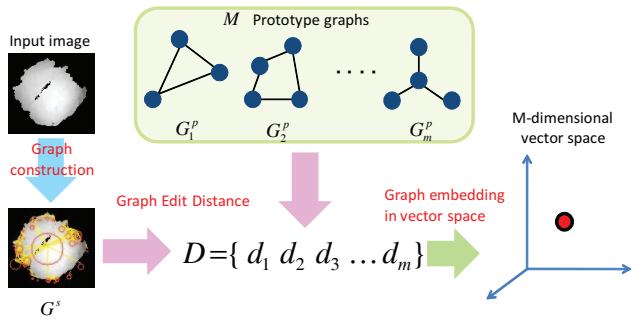


Fig. 7. Graph embedding in vector spaces

$T = \{G_1, G_2, ..., G_n\}$, and compute the graph edit distance $d(G_i, G_j)$ $(i, j = 1, ..., n; G_1, G_2 \in T)$. First, the set of the m prototype graphs $P = \{G_1^p, G_2^p, ..., G_m^p\}$ is selected from $T (m \leq n)$. Next, the graph edit distance between scene graph G^s and prototype graphs $G^p \in P$ is calculated. As a result, the m graph edit distances $d_1, ..., d_m (d_k = d(G^s, G_k^p))$ to the scene graph are obtained and assumed to form the m dimensional vector D. Thus, all scene graphs G^s can be embedded into the m dimensional vector space by using prototype graph set P. It can be described as follows:

$$\psi : G^s \longrightarrow \mathbf{R^m} \tag{4}$$

$$\psi \longrightarrow (d(G^s, G_1^p), d(G^s, G_2^p), ..., d(G^s, G_m^p)) \tag{5}$$

where, $d(G^s, G_i^p)$ is a graph edit distance. In this paper, as a set of the prototype graphs P, a set of the training graphs T is employed.

6. EXPERIMENTAL EVALUATION

6.1. Experimental Conditions

We used the Caltech-101 Database[1] for the experiment. It is composed of 101 classes, used for generic object recognition. We selected 10 object classes from among these 101 classes, and carried out comparative experiments between the proposed method and conventional methods (BoF). The examples of images used in the experiment are shown in Fig. 8. The training images were 30 images randomly selected from each class and the remaining images were used as the test images. In total, 300 training images and 541 test images were used for the 10 classes. The threshold χ, hierarchical level L of pseudo-hierarchical graphs, and edit cost c of the graph edit distance were used as the best value in the experiment. Because the number of training images was 300, all the images were embedded into the 300 dimension vector spaces. On the other hand, the Codebook size of the BoF method was 1000, the best value in the experiment. We used as the classifiers k-NearestNeighbor algorithm ($k = 10$) and multi-class SVM (linear and radial basis function (RBF)) to classify the vector formed by each method.

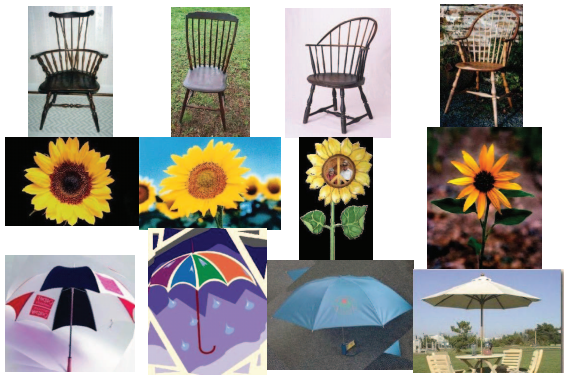


Fig. 8. Caltech-101 dataset

6.2. Experimental Results and Discussion

Fig. 9 and Fig. 10 show the recognition results for all classes and each class. From Fig. 9, it can be confirmed that the proposed method improved the accuracy. The recognition rate has improved

[1] http://www.vision.caltech.edu/Image_Datasets/Caltech101/

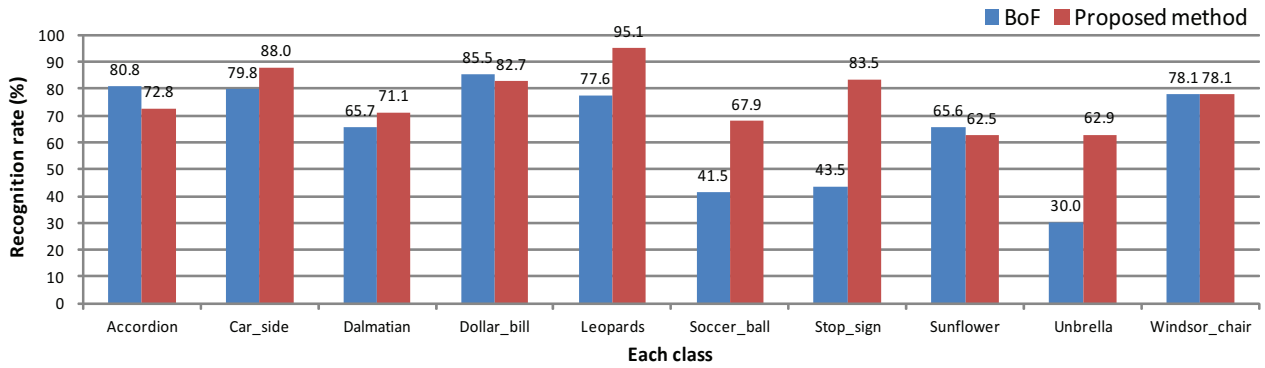


Fig. 10. Recognition result of each class (SVM(RBF))

with SVM (RBF) by 13.38% , SVM(linear) by 14.08%, and k-NN by 8.02% compared to the conventional method BoF.

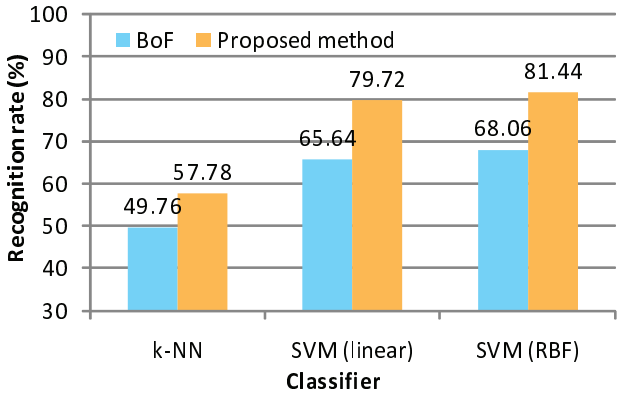


Fig. 9. Recognition result

In Fig. 10, the recognition result of each class shows that the proposed method provided stabler performance than the conventional method. This is because the conventional method uses only SIFT features, so it is strongly influenced by the accuracy of these features. In contrast, the proposed method can represents the shape and structure of the object using the graph.

7. CONCLUSION

In this paper, we proposed a new method to recognize generic objects by incorporating graph structural expression by embedding a graph into the vector spaces. By employing the graph structure of the object, the class recognition became robust to the SIFT features variance. As a result, the recognition accuracy was improved by 14.08% (SVM(linear)). In the future, we will study the selection method of more effective prototype graphs and the method of reducing calculation cost for graph edit distance. Moreover, we are planning to extend this proposed method to three-dimensional graphs and general object recognition using three-dimensional information.

Acknowledgment: This research was supported in part by MIC SCOPE.

8. REFERENCES

[1] K. Barnard and D. A. Forsyth, "Learning the semantics of words and pictures," in *IEEE International Conference on Computer Vision*, 2001, vol. 2, pp. 408–415.

[2] K. Barnard, P. Duygulu, N. de Freitas, and D. Forsyth, "Matching words and pictures," in *Journal of Machine Learning Research*, 2003, vol. 3, pp. 1107–1135.

[3] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth, "Visual categorization with bags of keypoints," in *European Conference on Computer Vision*, 2002, vol. IV, pp. 97–112.

[4] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, and C.Bray, "Object recognition as machine translation:learning alexicons for a fixed image vocabulary," in *ECCVWorkshop on Statistical Learning in Computer Vision*, 2004, pp. 1–22.

[5] D. G. Low, "Distinctive image features from scale invariant keypoints," in *Journal of Computer Vision*, 2004, vol. 60, pp. 91–110.

[6] J. Revaud, Y. Ariki, and A. Baskurt, "Scale-invariant proximity graph for fast probabilistic object recognition," in *Conference on Image and Video Retrieval (CIVR)*, 2010, pp. 414–421.

[7] D. Conte, P. Foggia, C. Sansone, and M. Vento, "Thirty years of graph matching in pattern recognition," in *International Journal of Pattern Recognition and Artificial Intelligence*, 2004, vol. 8, pp. 265–298.

[8] H. Bunke and G. Allerman, "Inexact graph matching for structural pattern recognition," in *Pattern Recognition Letters*, 1983, vol. 1, pp. 245–253.

[9] A. Sanfeliu and K. Fu, "A distance measure between attributed relational graphs for pattern recognition," in *IEEE Transactions on Systems, Man, and Cybernetics*, 1983, vol. 13, pp. 353–362.

[10] D. Justice and A. Hero, "A binary linear programming formulation of the graph edit distance," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006, vol. 28, pp. 1200–1214.

[11] M. Neuhaus, K. Riesen, and H. Bunke, "Fast suboptimal algorithms for the computation of graph edit distance," in *Joint IAPR International Workshops, SSPR and SPR 2006, Lecture Notes in Computer Science*, 2006, vol. 4109, pp. 163–172.

[12] K. Riesen and H. Bunke, "Approximate graph edit distance computation by means of bipartite graph matching," in *Image and Vision Computing*, 2009, vol. 27, pp. 950–959.

[13] K. Riesen, M. Neuhaus, and H. Bunke, "Graph embedding in vector spaces by means of prototype selection," in *Graph-based representations in pattern recognition (GbRPR)*, F. Escolano et al, Ed., Springer-Verlag Berlin, Heidelberg, 2007, pp. 383–393.

[14] E. Valveny and M. Ferrer, "Application of graph embedding to solve graph matching problems," in *CIFED*, 2008, pp. 13–18.

[15] M. Ferrer, E. Valveny, F. Serratosa, K. Riesen, and H.Bunke, "Generalized median graph computation by means of graph embedding in vector spaces," in *Pattern Recognition*, 2010, vol. 43, pp. 1642–1655.