Graph/Image Legend Retrieval

Ying-Wen Huang[1], Kuo-Hao Lu[1],
Shu-Yuan Chen[1@] and Chen-Chung Liu[2]
1:Department of Computer Science and Engineering, Yuan Ze University,
135 Yuan-Tung Road, Chung-Li, Taoyuan 320, Taiwan, R.O.C.
2: Graduate Institute of Network Learning Technology, National Central University,
300 Jhongda Rd., Chung-Li, Taoyuan, 320, Taiwan, R.O.C

Abstract

There are many content-based retrieval methods for image databases, however, none of them coped with graph and image simultaneously. Moreover, existing graph retrieval methods handled for a silhouette or a graph component rather than the whole graph. Hence, it is the goal of this paper to propose a graph/image legend retrieval method. The proposed method consists of two phases: legend extract and legend retrieval. The features used in our method include aspect ratio, number of legend components and spatial histograms of pixel number, border length and gray level. Since the processed legends can be properly divided into two categories: graph and image, type-based matching is adopted to evaluate the similarity between the query legend and each database legend using different similarity measures according to the type of the query legend which can be automatically determined. In this way, the correct legend can be retrieved. The effectiveness and practicability of the proposed method have been demonstrated by various experiments.

Keywords: legend retrieval, graph retrieval, image retrieval, content-based image retrieval, legend extraction, type-based matching

1. Introduction

With the advent of computing technology, media acquisition/storage devices, and multimedia compression standard, more and more digital data are generated and available to users all over the world. Nowadays, it is easy to access electronic books, electronic journals, web portals, and video streams. Hence, it will be convenient for users to provide related information retrieval for the query legend. Unfortunately, although there are many content-based retrieval methods for image databases, none of them is specifically designed for coping with graph and image simultaneously and handling the whole legend rather than for a silhouette or a graph component.

There are many content-based retrieval approaches for image databases [1,2,5,7,8]. In general, they can be divided into shape-based [1,2], image-based [7,8] and region-based approaches [5]. The shape-based method is concerning with a silhouette rather than our method for legend comprised by a lot of curves or lines. The image-based method usually adopts similarity measures based on color histogram. However, the legend may be colored or monochrome so gray-level histogram should be used but it is not as powerful as the color histogram. The region-based approach should segment the query and database images into semantic regions that are nonsense for graph.

Some approaches [3,6] focus on managing graphs existing in document such as company logos, engineering diagrams, maps, business charts, fingerprints, musical scores, and so on. Those approaches are concerned with the topics related to the analysis and recognition of graphic component rather than the whole graph retrieval as proposed by our method.

It is the goal of this paper to propose legend retrieval for handling graph and image retrieval simultaneously. Hence, the main difference between the proposed legend retrieval and other existing image retrieval methods are as follows. First, our approach separates the foreground from the background and identifies the legend part for matching so as to reduce processing time as well as increase retrieval accuracy. Second, type-based matching is proposed in this study to handle graph and image retrieval simultaneously rather than other retrieval systems cope with only graph [3,6] or image [1,2,5,7,8] individually. Third, the proposed method can handle whole graph including many curves and lines rather than other methods only for a silhouette [1,2] or for a graph component [3,6].

2. Legend Database

The processing unit of the proposed method is a

* This work was partially supported by the National Science Council of Taiwan, R.O.C., under Grants NSC-95-2745-E-155-008-URD.
@To whom all correspondence should be sent.
Tel: 886-3-463-8800 ext. 2357
E_mail: cschen@saturn.yzu.edu.tw

legend which may exist in electronic books, video mediums, or web portals. The processed legends can be properly divided into two categories: graph and image. In general, graph-legends consist of simple lines and curve, while the image-legends may be diversiform.

In this study, there are no limitations for the size of legend image and it can be variable. Moreover, legend images were collected from power point files or electronic books to construct database for experiment. The number of images in our database is 450 including 250 graph and 200 image legends. The legend existing in each image of database is also scanned from books as the query image.

3. Legend Extraction

3.1. Border detection

Since the legend may be colored or monochrome, the legend images should be transformed into the standard form so that the comparison between the two kinds of legends is possible. In this study, colored images are transformed into gray-level images by transferring RGB color space into YIQ color space [4]. Henceforth, we can get gray-level legend image, $g(x,y)$, with Y value as the gray value.

3.2. Foreground extraction

For legend retrieval, the foreground of legend image should be extracted first. The foreground is specified by foreground map and foreground border images in this study. The simple bi-level thresholding and border detector are adopted to get them, respectively. More specifically, the foreground map and border images, $f(x,y)$ and $b(x,y)$ can be defined by the following equations, respectively.

$$f(x,y)=\begin{cases}1, & g(x,y)\le TH_b\\ 0, & \text{otherwise}\end{cases} \tag{1}$$

$$b(x,y)=\begin{cases}1, & g(x,y)\ \text{satisfys border criterion}\ B(x,y)\\ 0, & \text{otherwise}\end{cases} \tag{2}$$

where TH_b is determined automatically by the proposed method and $B(x,y)$ is the criterion of border points as described later. Examples of foreground map and border images are shown in Fig. 1.

The essential factor of bi-level thresholding is threshold determination. The threshold value is crucial to separate foreground from background which in turn dominates the retrieval accuracy. The difficulty to determine threshold are due to the following factors. First, the images of legends may be captured under different illumination and environment. Second, different resources may have different signal magnitude. In this study, moment-preserving thresholding [9] is adopted and modified to handle these problems so as to obtain good foreground map.

Given an image $g(x,y)$ with n pixels, the ith moment m_i of $g(x,y)$ is defined as

$$m_i=\left(\frac{1}{n}\right)\sum_x\sum_y\left(g(x,y)\right)^i,\ i=1,2,3,\cdots \tag{3}$$

Let $g(x,y)$ with values less than and greater or equal to TH_b be replaced by z_0 and z_1, respectively in the bi-level image. Let the fractions of them be p_0 and p_1, respectively. The threshold value TH_b can then be chosen as $p_0-\text{tile}$ of the histogram of $g(x,y)$, where p_0 can be calculated by

$$\begin{aligned}
&p_0=\left(\frac{1}{p_d}\right)\begin{vmatrix}1 & 1\\ m_1 & z_1\end{vmatrix},\ p_d=\begin{vmatrix}1 & 1\\ z_0 & z_1\end{vmatrix}\\
&z_1=\left(\frac{1}{2}\right)\left[-c_1+\sqrt{\left(c_1^2-4c_0\right)}\right]\\
&c_0=\left(\frac{1}{c_d}\right)\begin{vmatrix}-m_2 & m_1\\ -m_3 & m_2\end{vmatrix},\ c_1=\left(\frac{1}{c_d}\right)\begin{vmatrix}m_0 & -m_2\\ m_1 & -m_3\end{vmatrix}\\
&c_d=\begin{vmatrix}m_0 & m_1\\ m_1 & m_2\end{vmatrix}
\end{aligned} \tag{4}$$

The details of derivation can be referred to Ref. [9].

In our method, we further utilize Sobel detector [4] to find the proper threshold values for legend retrieval. Assume that resulting image of applying the Sobel detector to the legend image $g(x,y)$ is $s(x,y)$. We can then get edge image $e(x,y)$ as described by

$$e(x,y)=\begin{cases}1, & s(x,y)\ge TH_e\\ 0, & \text{otherwise}\end{cases} \tag{5}$$

In this study, TH_e is set to 127. An example of edge image is shown in Fig. 2.

We then re-execute moment preserving thresholding on the image $h(x,y)$

$$h(x,y)=\begin{cases}g(x,y), & e(x,y)=1\\ 0, & \text{otherwise}\end{cases} \tag{6}$$

Note that the difference between $g(x,y)$ and $h(x,y)$ is that $h(x,y)$ include only those pixels of $g(x,y)$ as edge pixels. In other words, only the gray levels of edge pixels are concerned for threshold determination. Actually, we found that the thresholding results on $h(x,y)$ are better than those on $g(x,y)$.

3.2.2. Foreground border image

In this study, border detector [4] is applied to foreground map image $f(x,y)$ to get border image $b(x,y)$. The border detector is performed by scanning $f(x,y)$ from top to bottom and from left to right. More specifically, for each pixel $f(x,y)$, if one of the upper neighbor $f(x,y-1)$ and left neighbor $f(x-1,y)$ has different binary labels from $f(x,y)$, the point (x,y) is denoted as a border point, i.e., $B(x,y)=1$. An example of border image is shown in Fig. 1(c).

3.3. Region of interest

Sine each legend may not occupy the whole legend image, the region enclosing the legend only should be detected first to reduce processing time as well as increase retrieval accuracy. In other words, it is necessary to find the region of interest (ROI) according to the boundary of legend. The projection technology is adopted in this study to solve this problem. The ROI detection consists of two stages: content and legend detection to find region of content (ROC) and region of legend (ROL), respectively.

3.3.1. Content detection

The projection in horizontal and vertical direction can be used to find the horizontal and vertical boundaries, respectively. Since the methods for both directions are similar, we describe the method for the horizontal case only. The horizontal and vertical projections $p_v(x)$ and $p_h(y)$ are obtained.

Let the legend image have width of *W* and height of *H*. The most top and bottom rows having horizontal projections $p_h(y)$ greater than a threshold value TH_p are regarded as the horizontal boundary of ROC. In this study, TH_p is set to 3. The same method can be used in the vertical detection. In this way, we can find the boundary of ROC.

3.3.2. Legend detection

Since ROC may or may not include caption, the caption part should be detected and deleted so that the matching evaluation can be performed on only the legend part (ROL) to increase retrieval accuracy. The caption part can also be detected using the projection technology.

Let ROC superimpose on the foreground map image $f(x,y)$. The horizontal projections $p_h^r(y)$ or vertical projections $p_v^r(x)$ during the ROC are computed. The first and the second rows having $p_h^r(y)$ greater than a threshold value TH_p are regarded as the horizontal border of the first candidate, and the third and the fourth as the second candidate and so on. The vertical borders of candidates can be obtained in the similar way. Each candidate is then verified by the following constraints in sequence to determine whether it is a caption or legend.

(1)Legend constraint

The constraint is derived from the observation that caption is comprised by text and text has shot vertical lines rather the graph has long vertical lines. Those candidates with long vertical lines will not be caption. The details of this constraint are described as follows.

Let the candidate have width of W_c and height H_c. Let each candidate superimpose on the foreground map image $f(x,y)$. The vertical projections $p_v^c(x)$ during the candidate are computed. If there is a column having $p_v^c(x)$ greater than a threshold value $W_c/16$ or a sum of $p_v^c(x)$ belonging to three consecutive columns greater than a threshold $W_c/14$, the candidate is considered as legend.

(2)Caption constraint

The constraint is derived from the observation that caption is composed of text. Those candidates have too few or too many pixels in foreground map image will not be caption. The details of this constraint are described as follows.

Let each candidate superimpose on the foreground map image $f(x,y)$. The number of pixels, $N_c^{(f)}$, covered by the candidate is then counted. The pixel number ratio related to the candidate, $R_c^{(f)}$, can be defined as

$$R_c^{(f)} = \frac{N_c^{(f)}}{W_c \times H_c} \qquad (7)$$

Those candidates with pixel number ratio, $R_c^{(f)}$, less than a threshold value TH_l or larger than a threshold value TH_u are regarded as captions and deleted. In this study, TH_l and TH_u are set to 0.058 and 0.3, respectively. After captions are deleted, the ROC detection as described in Section 3.3.1 is applied again on those remaining parts of ROC to obtain the graph/image part, i.e., ROL.

4. Legend Retrieval

The type of query legend is first determined as one of the three types: graph, image and hybrid of graph and image according to the number of foreground pixels in the query legend. Then, different retrieval strategies will be adopted according to the type of the query legend.

4.1. Feature extraction

The features used in this study for legend retrieval include aspect ratio, number of legend components and spatial histograms derived from foreground map, border and quantized images, namely histograms of pixel number, border length and gray level, respectively. They are defined as follows.

Let the ROL have width of W_l and height of H_l. The aspect ratio of ROL, Asp, can be defined as

$$Asp = \frac{W_l}{H_l} \tag{8}$$

Let ROL superimpose on the foreground map image $f(x, y)$. The horizontal projections $p_h^l(y)$ or vertical projections $p_v^l(x)$ during the ROL are computed. The first and the second rows having $p_h^l(y)$ greater than a threshold value TH_p are regarded as the horizontal border of legend component, and the third and the fourth as the second components, and so on. For each horizontal legend component, the vertical borders of each component can be obtained in the similar way. The number of legend components is then counted as a feature Noc.

To obtain the histogram feature, the ROL is first partitioned into $4 \times 4 = 16$ blocks. Let the 16 blocks be indexed in ascendant order from left to right and from top to bottom. The number of pixels in $f(x, y)$ enclosed by the *i*-th block is counted as $N_i^{(p)}$. The histogram of pixel number, $\mathbf{h}^{(p)}$, can be defined as

$$\mathbf{h}^{(p)} = \left(h_i^{(p)}\right), h_i^{(p)} = \frac{N_i^{(p)}}{16 \times W_l \times H_l}, i = 1, \cdots, 16 \tag{9}$$

Similarly, the histogram of border length, $\mathbf{h}^{(b)}$, can be calculated by the following equation except that the foreground map image $f(x, y)$ is replaced by the foreground border image $b(x, y)$.

$$\mathbf{h}^{(b)} = \left(h_i^{(b)}\right), h_i^{(b)} = \frac{N_i^{(b)}}{16 \times W_l \times H_l}, i = 1, \cdots, 16 \tag{10}$$

where $N_i^{(b)}$ is the number of border points in $b(x, y)$ enclosed by the *i*-th block.

Finally, the histogram of gray level is obtained as follows. First, the quantized image, $q(x,y)$ of the input legend image $g(x,y)$ should be obtained by the following quantization method. The maximum and minimum of gray values, $g_{\max}$ and $g_{\min}$, in $g(x,y)$ are computed. The gray values are then uniformly quantized into four levels by dividing the range of $(g_{\min}, g_{\max})$ into four equal intervals. More specifically, $q(x,y)$ can be defined by

$$\begin{aligned} &q(x, y) = j, \\ &\text{if } g(x, y) \in \left(g_{\min} + (j-1) \times \delta, g_{\min} + j \times \delta\right), \\ &j = 1,2,3,4, \ \delta = \frac{g_{\max} - g_{\min}}{4} \end{aligned} \tag{11}$$

The histogram of gray level, $\mathbf{h}^{(g)}$, is then easily obtained as follows. The number of pixels with quantized label *j* in $q(x,y)$ enclosed by the *i*-th block is counted as $N_{i,j}^{(g)}$. The histogram of $\mathbf{h}^{(g)}$, can then be defined by

$$\begin{aligned} &\mathbf{h}^{(g)} = \left(h_{i,j}^{(g)}\right), h_{i,j}^{(g)} = \frac{N_{i,j}^{(g)}}{16 \times W_l \times H_l}, \\ &i = 1, \cdots, 16, \ j = 1, \cdots, 4 \end{aligned} \tag{12}$$

4.2. Type-based matching

In the retrieval stage, the feature of aspect ratio, *Asp*, is first used to prune irrelevant database legends. The type-based matching can then be performed between the query and only the database legends in the small plausible set. More specifically, the similarity measure between query legend *Q* and database legend *S* on the basis of *Asp* is defined respectively by

$$d_{Asp}(Q, S) = \left|Asp(Q) - Asp(S)\right| \tag{13}$$

where (Q) refers to features of Q and (S) refers to features of S, respectively. Thus, we can get the plausible set $P(Q)$ for Q as

$$P(Q) = \left\{S \middle| d_{Asp}(Q, S) \le 0.2\right\} \tag{14}$$

The type of Q, $T(Q)$, can be determined according to the number of pixels by the following equation

$$T(Q) = \begin{cases} \text{Graph}, & f(Q) < 0.25 \\ \text{Hybrid}, & 0.25 \le f(Q) \le 0.5, \\ \text{Image}, & 0.5 < f(Q) \end{cases} \quad f(Q) = \sum_{i=1}^{16} \frac{N_i^{(p)}(Q)}{W_l \times H_l} \tag{15}$$

The final similarity measures, $d(Q, S)$, between Q and S is defined by

$$\begin{aligned} &d(Q, S) = \\ &\begin{cases} d^{(l)}(Q, S) + d^{(b)}(Q, S), & T(Q) = \text{Graph} \\ d^{(l)}(Q, S) + d^{(b)}(Q, S) + d^{(g)}(Q, S), & T(Q) = \text{Hybrid} \\ d^{(g)}(Q, S) + d_{Noc}(Q, S), & T(Q) = \text{Image} \end{cases} \\ &d^{(l)}(Q, S) = \sum_{i=1}^{16} \left|h_i^{(l)}(Q) - h_i^{(l)}(S)\right| \\ &d^{(b)}(Q, S) = \sum_{i=1}^{16} \left|h_i^{(b)}(Q) - h_i^{(b)}(S)\right| \\ &d^{(g)}(Q, S) = \sum_{i=1}^{16} \sum_{j=1}^{4} \left|h_{i,j}^{(g)}(Q) - h_{i,j}^{(g)}(S)\right| \\ &d_{Noc}(Q, S) = \left|Noc(Q) - Noc(S)\right| \end{aligned} \tag{16}$$

Thus, the final similarity measure, $d(Q,S)$, between query legend Q and each database legend S in the set PQ) can be calculated by Eq. (16). Note that the weights of $d^{(l)}(Q,T), d^{(b)}(Q,T), d^{(g)}(Q,T)$, and $d_{Noc}(Q,T)$ are implicitly involved in the scales of those terns themselves. Those database legends in the set PQ) are then sorted in the ascending order of $d(Q,S)$. The database legend in the first rank is the retrieval result for the query legend Q.

5. Experimental Results

The proposed method has been implemented on a personal computer with a single AMD K-8 3200+ CPU and 512 Megabytes DDRAM. The operating system is Microsoft Windows XP Server Chinese version Service Pack 2. The program was developed in the C++ language and compiled under Borland C++ Builder version 6. The experiment database is described in Section 2.

The performance of legend retrieval can be measured by the retrieval accuracy. The retrieval accuracy is computed as the ratio of the number of legends correctly retrieved to the number of total query legends. Moreover, not only the retrieval accuracy with respect to the first rank but also the second and third ranks are concerned in our experiments.

Some options need to be chosen in the proposed method, for example the partition size and the number of features. These will be determined by experimentation. In this study, the partitions of 3×3 and 4×4 will be concerned. The combinations of features include only pixel number, $\mathbf{h}^{(p)}$ (denoted P), only border length, $\mathbf{h}^{(b)}$ (denoted B), only gray level $\mathbf{h}^{(g)}$ (denoted G), and the different combinations of $\mathbf{h}^{(p)}, \mathbf{h}^{(b)}$, and $\mathbf{h}^{(g)}$ (denoted P+B, P+G, B+G, P+B+G). Moreover, the necessary of type-based matching is also discussed. On the other hand, query and database legend images may include caption (denoted in) or may not include caption (denoted out). Thus, the retrieval status will be four: in/in, in/out, out/out, out/in and each u/v represents the query in the status of u and the database in the status of v.

First, the retrieval accuracies of 1st rank in the cases of in/in, out/out, in/out and out/in for the partition of 4×4 using different combinations of histograms (P, B, G, P+B, P+G, B+G, P+G+B) and our method are depicted in Fig. 3. From this figure, we found that the combination of pixel size, border length, and gray levels, i.e., P+G+B outperforms the others except our type-based matching method. In addition, we can conclude that the strategy of type-based matching is necessary since our method adopts this strategy and has the highest accuracy. Moreover, the accuracy of in/in is higher than those of out/out, in/out, out/in. It is not surprising since the query and database images in the case of in/in include more information than others.

Finally, the retrieval accuracies of our method with respect to 1st, 2nd, and 3th ranks for the partition of 4×4 are listed in Tables 1. From this table, we can conclude that the retrieval accuracy increase when more ranks are included. Some retrieval results are shown in Fig. 4.

On the other hand, the traditional image retrieval based on histogram matching (Swain method) [8] was implemented with resolutions of 64, 16, and 4 bins in this study for comparison. Provided that caption is excluded, our method has accuracy rate 89%, while the Swain method has only 0.87%, 1.45% and 2.91% accuracies with resolutions of 64, 16, and 4 bins. It is obvious that our method is the best one.

6. Conclusions

A legend retrieval method is proposed in this paper to provide related information retrieval for the legend. The proposed method consists of two phases: legend extract and legend retrieval. The former includes gray-level transformation, foreground extraction and legend extraction. The latter includes feature extraction and type-based matching. In fact, the main difference between the proposed legend retrieval and other existing image retrieval methods are as follows. First, our approach separates the foreground from the background and identifies the legend part for matching so as to reduce processing time as well as increase retrieval accuracy. Second, type-based matching is proposed in this study to handle graph and image retrieval simultaneously rather than other systems that cope with only graph or image individually. Third, the proposed method can handle whole graph including many curves and lines rather than other methods only for a silhouette or for a graph component. Experimental results demonstrate that the effectiveness of the proposed method.

References

[1] I. Bartolini, "Warp: accurate retrieval of shapes using phase of Fourier descriptors and time warping distance," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 1, pp. 142-147, 2005.

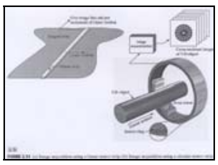
[2] A. Chalechale, G. Naghdy, and A. Mertins, "Sketch-based image matching using angular partitioning," *IEEE Trans. Systems, Man and Cybernetics*, vol. 35, no. 1, pp. 28-41, 2005.

[3] L.A. Fletcher and R. Kasturi, "A robust algorithm for text string separation from mixed text/graphics images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, no. 6, pp. 910-918, 1998.

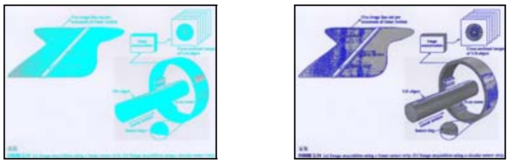
[4] R.G. Gonzalez and R.E. Woods, *Digital Image Processing*, Prentice Hall, NJ, 2002.
[5] F. Jing, M. Li, H.J. Zhang, and B. Zhang, "An efficient and effective region-based image retrieval framework," *IEEE Trans. Image processing*, vol. 13, no. 5, pp. 699-709, 2004.
[6] C.P. Lai and R. Kasturi, "Detection of dimension sets in engineering drawings," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.16, no. 8, pp. 848-855, 1994.
[7] H. Nezamabadi-Pour and E. Kabir, "Image retrieval using histograms of un-color and bi-color blocks and directional changes in intensity gradient," *Pattern Recognition Letters*, vol. 25, no. 14, pp. 1547-1557, 2004.
[8] M.J. Swain and D.J. Ballard, "Color indexing," *Int. J. of Computer Vision*, vol. 7, no.1, pp. 11-32, 1991.
[9] W.H. Tsai, "Moment-preserving thresholding: a new approach," *Computer Vision, Graphics, and Image Processing*, vol. 29, no.3, pp. 377-393, 1985.

Table 1. Retrieval accuracies of our method in the cases of in/in, out/out, in/out, and out/in for the partition of 4×4.

		1st correct %	2nd correct %	3rd correct %
in/in	Image	83.15	91.05	94.73
	Graph	98.46	98.84	99.61
	Total	92.00	95.55	97.55
out/out	Image	82.10	91.05	94.21
	Graph	94.23	96.15	96.92
	Total	89.11	94.00	95.77
in/out	Image	40.52	49.47	53.68
	Graph	40.00	45.00	48.84
	Total	40.22	46.88	50.88
out/in	Image	44.21	53.15	56.84
	Graph	39.61	43.07	46.15
	Total	41.55	47.33	50.66

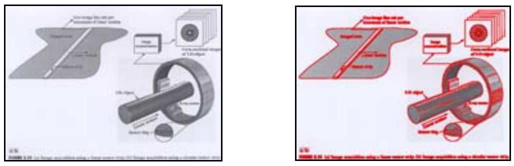


(a) Original legend image $g(x, y)$.



(b) Foreground map image $f(x, y)$.
(c) Foreground border image $b(x, y)$.

Fig. 1. An example of foreground and border images.



(a) Original legend image $g(x, y)$.
(b) The resulting edge image $e(x, y)$.

Fig. 2. An example of edge image.

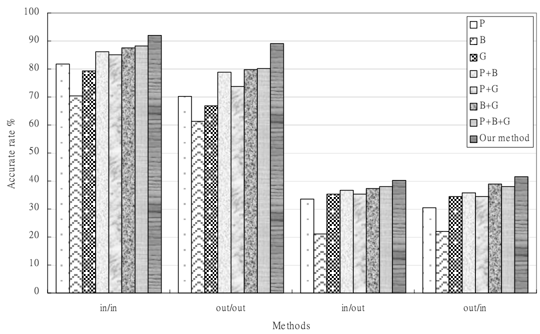


Fig. 3. Retrieval accuracies in the cases of in/in, out/out, in/out, and out/in for the partition of 4×4 using different combinations (P, B, G, P+B, P+G, B+G, P+B+G and our method).

(a) An example of graph retrieval.
(b) An example of image retrieval.

Fig. 4. Examples of retrieval results.