A COMPRESSION DOMAIN IMAGE RETRIEVAL METHOD USING EDGE INFORMATION AND PROJECTION HISTOGRAM

Wen-Jia Kuo, Ruey-Feng Chang, and So-Wei Pong

Department of Computer Science and Information Engineering
National Chung Cheng University, Chiayi, Taiwan, R.O.C.
Email:rfchang@cs.ccu.edu.tw

ABSTRACT

With the development of multimedia technologies, the problem of how to retrieve a specified image from a large image database becomes an important issue. In this paper, we extract the features directly from the DCT coefficients in the compressed domain as the keys for the retrieval instead of fully reconstructing the compressed images. The advantage is that the complexity and speed of building indices can be efficiently improved. The edge types are used as an image feature for our image retrieval. In the proposed method, users can use the partial images and sketch images to retrieve the desired images. Also, we propose an efficient image retrieval scheme using the projection histogram of edge information to speed up the image retrieving. According to the experiment results, we know that our method is useful for the image retrieval.

1. INTRODUCTION

In the recent years, the number of digital images has enormously increased in the computer storage or network transmission. The newest technologies enable us to effectively transmit and store a large number of images. People can also make use of images to represent their ideas more clearly. As the number of images gets larger and larger, we need a good and efficient way to manage and find the image that we are interested in. However, retrieving images from a large and varied collection using the image content as a key is a challenge. Hence, it is necessary to develop efficient methods to retrieve images from a large image database.

The main purpose of image retrieval is to find the same or the most similar image from the database. Thus, the images that satisfy the perceptual similarity with the users' query could be retrieved from the database. In addition, we should also consider the time and complexity when retrieving images. In the past few years, the content-based visual query (CBVQ) techniques are being developed. The CBVQ techniques index the images [1]-[4] and videos [5] by the visual features for the purpose of providing a system with search capabilities. The process of content-based image retrieval has been investigated in several recent efforts such as Virage's commercial image retrieval system [2], the Chabot image retrieval system [3], IBM's QBIC project [6], and MIT's Photobook system [7]. The QBIC system has investigated features such as color, texture, and shape. The Photobook system has focused on the texture, shape, and facial features. The QBIC system has also placed a research focus on designing strategies for indexing the features [8]. On the other hand, the Photobook system has investigated new methods for representing image features not only to provide the discrimination of image, but also to preserve their semantic information. Some systems allow user to sketch the outlines of objects to find out the desired images. For example, the QVE (Query by Visual Example) system [9] first normalizes the images in the database as regular-sized images and then processes them to get abstract images. The abstract images are used to be pictorial index. Users sketch rough outlines of objects for the desired image and the rough drawing is matched with the abstract images to find out the most similar image. Currently, finding effective methods to incorporate content-based image retrieval approaches into an image database system is an attractive research [3]-[12].

In this paper, we propose a method to retrieve images compressed in the DCT frequency domain. The edge type of a block is detected in the DCT frequency domain directly. The edge information is then used as the key to match the candidate images with the query image. Although the frequently used histogram technique [13]-[15] for the edge information can speed up the image retrieving, this histogram technique will lose the helpful location relationship. Hence, we propose the technique of projection histogram not only to speed up the image retrieving but also to use the location relationship. There are two kinds of query methods in our proposed issue. Users can use a partial image to find the desired image or just sketch the rough outlines of objects to find the image with similar objects.

The rest of this paper is organized as follows. In Section 2, we describe the edge detection in DCT frequency domain. In Section 3, we introduce the edge-based image retrieval method. The proposed content-based image retrieval scheme using projection histogram of edge information is introduced in Section 4. The experimental results are shown in Section 5 and some conclusion is drawn in Section 6.

2. EDGE DECTION IN DCT FREQUENCY DOMAIN

The edges usually represent the major characteristic of the image such as shapes, textures, and so on. Hence, it can be considered as an important information that we can use for image retrieval. For the reason that most of the general compressed images are encoded in the frequency domain, we need a method to extract the edge feature in the compression domain. In this section, we briefly describe the method of finding edges directly in the DCT frequency domain.

2.1 Edge Detection in DCT Frequency Domain

An edge is the boundary between two regions with relatively distinct gray-level properties. The basic idea for most of the edge-detection techniques is the computation of a local derivation operation. For example, the Sobel operator uses two masks to evaluate the magnitude of the gradient G_x and G_y at pixel (x,y).

However, many compressed images are block-based encoded instead of the pixel-based encoding in the spatial domain. Therefore, the Sobel operator can not be directly applied to the compressed images. Fortunately, some edge classifying methods using the DCT coefficients in the frequency domain [16], [17] have already been proposed in the recent year.

Shen and Sethi [18] introduce the method of extracting the edge features from the DCT coefficients of the compressed image. The equations of DCT and inverse DCT are shown as follows

$$F(u,v)=\left(\frac{1}{4}\right)C(u)C(v)\sum_{x=0}^{7}\sum_{y=0}^{7}f(x,y)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16}\,,\quad(1)$$

$$f(x,y)=\left(\frac{1}{4}\right)\sum_{u=0}^{7}\sum_{v=0}^{7}C(u)C(v)F(u,v)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16}\,,\quad(2)$$

where $C(u),C(v)=\begin{cases}1/\sqrt{2} & x=0\\ 1 & otherwise\end{cases}$.

Next, we briefly discuss the way of finding the edge information using the DCT coefficients. According to the equation (1), we consider the coefficient $F(0,1)$.

$$\begin{aligned}F(0,1)&=\left(\frac{1}{4}\right)C(0)C(1)\sum_{x=0}^{7}\sum_{y=0}^{7}f(x,y)\cos\frac{(2y+1)\pi}{16}\\&=\left(\frac{1}{4}\right)C(0)C(1)\sum_{x=0}^{7}f(x,y)\sum_{y=0}^{7}\cos\frac{(2y+1)\pi}{16}\end{aligned}\,.\quad(3)$$

Because of $\cos(\pi-\theta)=-\cos\theta$, the equation (3) can be further rewritten as

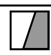
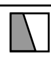
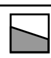
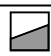
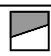
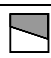
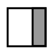
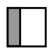
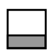
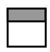
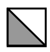
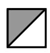
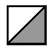
$$\begin{aligned}F(0,1)=\left(\frac{1}{4}\right)C(0)C(1)\Bigg[&\cos\frac{\pi}{16}\left(\sum_{x=0}^{7}f(x,0)-\sum_{x=0}^{7}f(x,7)\right)\\&+\cos\frac{3\pi}{16}\left(\sum_{x=0}^{7}f(x,1)-\sum_{x=0}^{7}f(x,6)\right)+\cos\frac{5\pi}{16}\left(\sum_{x=0}^{7}f(x,2)-\sum_{x=0}^{7}f(x,5)\right)\\&+\cos\frac{7\pi}{16}\left(\sum_{x=0}^{7}f(x,3)-\sum_{x=0}^{7}f(x,4)\right)\Bigg]\end{aligned}\quad(4)$$

Similarly,

$$\begin{aligned}F(1,0)=\left(\frac{1}{4}\right)C(1)C(0)\Bigg[&\cos\frac{\pi}{16}\left(\sum_{y=0}^{7}f(0,y)-\sum_{y=0}^{7}f(7,y)\right)\\&+\cos\frac{3\pi}{16}\left(\sum_{y=0}^{7}f(1,y)-\sum_{y=0}^{7}f(6,y)\right)+\cos\frac{5\pi}{16}\left(\sum_{y=0}^{7}f(2,y)-\sum_{y=0}^{7}f(5,y)\right)\\&+\cos\frac{7\pi}{16}\left(\sum_{y=0}^{7}f(3,y)-\sum_{y=0}^{7}f(4,y)\right)\Bigg]\end{aligned}\quad(5)$$

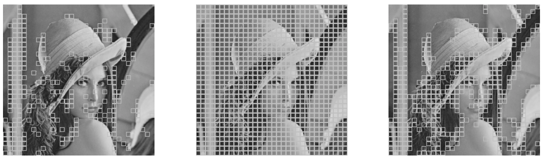
The result shows that the value of $F(1,0)$ depends on the intensity differences in the vertical direction between the upper and lower parts of the input block. We can find that the coefficients $F(1,0)$ and $F(0,1)$ can be used to roughly classify the edge blocks into 16 types. Table 1 shows these 16 block types and the relationship between coefficients $F(0,1)$ and $F(1,0)$. The coefficients $F(2,0)$, $F(3,0)$, $F(0,2)$, and $F(0,3)$ should also be considered if we want to further achieve accurate classification.

Table 1. Edge types and relationship between $F(0,1)$ and $F(1,0)$ [12].

Vertical-dominant	$\lvert F(0,1)\rvert > \lvert F(1,0)\rvert$	$F(0,1)\le 0$	$F(1,0)>0$	
			$F(1,0)<0$	
		$F(0,1)\ge 0$	$F(1,0)>0$	
			$F(1,0)<0$	
Horizontal-dominant	$\lvert F(0,1)\rvert < \lvert F(1,0)\rvert$	$F(1,0)\le 0$	$F(0,1)>0$	
			$F(0,1)<0$	
		$F(1,0)\ge 0$	$F(0,1)>0$	
			$F(0,1)<0$	
Vertical	$F(1,0)=0$	$F(0,1)<0$		
		$F(0,1)>0$		
Horizontal	$F(0,1)=0$	$F(1,0)<0$		
		$F(1,0)>0$		
Diagonal	$\lvert F(0,1)\rvert = \lvert F(1,0)\rvert$	$F(0,1)<0$, $F(1,0)>0$		
		$F(0,1)>0$, $F(1,0)<0$		
		$F(0,1)>0$, $F(1,0)>0$		
		$F(0,1)<0$, $F(1,0)<0$		

3. EDGE-BASED IMAGE RETRIEVAL METHOD

In this section, we introduce our proposed edge-based image retrieval method. The method contains two kinds of queries. First, a partial image can be used as the query image to find the desired image in the database. Second, users can draw rough outlines of objects to search images with the desired objects.



(a) (b) (c)

Fig. 1. Edge blocks of the 256 × 256 Lena image. (a) 6 coefficients, (b) 2 coefficients, (c) 2 coefficients with threshold 64 are used to classify edge blocks.

3.1 Edge Block Detection

In section 2, we explain the way of detecting an edge block using the AC coefficients in the DCT frequency domain. In fact, it is not enough to detect edge blocks using only 2 coefficients in an 8 × 8 block. On the contrast, it will get much accurate classification result using more coefficients. However, our major goal is to see whether the block contains an obvious edge or not. Hence, it is enough for us to use only two AC coefficients to detect the edge block. The major concern of detecting vertical dominant and horizontal dominant edge block is the relationship between $F(0,1)$ and $F(1,0)$. In order to avoid the confused result of weak edge block, we use a threshold technique to achieve this purpose. A threshold TH is defined to eliminate the weak edge blocks. If the absolute value of $F(0,1)$ or $F(1,0)$ in an edge block is greater than TH, we consider it as a strong edge block. Fig. 1 shows the edge blocks of Lena image detected using different number of coefficients. Using the threshold technique, the weak edge blocks are considered as non-edge blocks.

3.2 Query by Partial Image

In the image retrieval application, people often focus on the result whether some objects they specified are in the images or not. Hence, it is important if we can provide the technique of querying by a partial image. The partial image may be composed of a major part of the original image such as a face, a building, or a car. The partial image is also compressed in the DCT based format with smaller size than the size of original image. Moreover, the partition of 8×8 DCT blocks may be different from the one of 8×8 DCT blocks in the original image. In the querying procedure, the edge type of each block in the partial image is first detected and then forms a 2D map of edge block types. For each image in the database, we also detect edge type of each block and get its corresponding 2D map. The images in the database are then matched with the 2D map of the partial image from left to right and from top to down as shown in Fig. 2. A match score can be evaluated by counting the number with the same block edge type in the match procedure. The maximum match score of different locations in the image is then considered as the final match score of this image. After comparing the partial image with all the images in the database, the images with higher match scores are the best candidates.

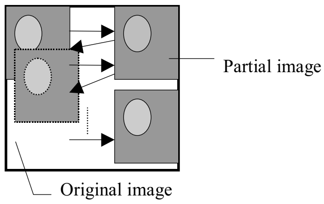
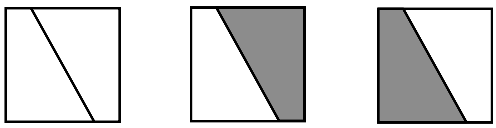


Fig. 2. The partial image matches the candidate image.

3.3 Query by the User's Sketch Image

The other method introduced in this section is the query by user's sketch image. In the sketch querying, users directly draw the outlines of the desired objects. The user's sketch image is also divided into blocks of size 8×8 and compressed into the DCT frequency domain. After the partition and transformation, the edge types of blocks for each object are detected in the compression domain. Note that the user's sketch image does not show the edge strength because of the simple line drawing. Hence, some of the edge types can be merged as the same edge type. For example, the edge block sketched by a user in Fig. 3(a) can be considered as the edge blocks shown in Fig. 3(b) and (c). They have the same edge direction while user's sketch image shows edge directions only. The merged edge types are listed in Table 2. The edge types are stored as keys for the future image retrieval.



(a) (b) (c)

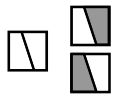
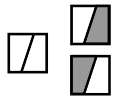
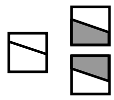
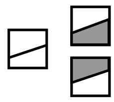
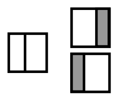
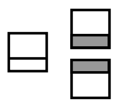
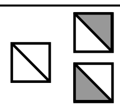
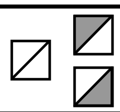
Fig. 3. The edge block types (b) and (c) that can be merged into type (a).

3.4 Complexity of Query by Partial Image

In our proposed method, we provide a convenient way for users to query by partial image. In this section, we will discuss the complexity needed in the query steps. Assume that the image size in the database is $M \times N$, and the size of partial image for querying is $M_p \times N_p$. In the query procedure, the partial image is matched from left to right and from top to bottom. Thus, the number of block comparison needed in the match step of each possible position is $\left\lceil \frac{M_p}{d} \right\rceil \times \left\lceil \frac{N_p}{d} \right\rceil$ if the DCT block size is $d \times d$. Meanwhile, the total number of possible matching positions is $\left\lceil \frac{M - M_p}{d} \right\rceil \times \left\lceil \frac{N - N_p}{d} \right\rceil$. Hence, the complexity for comparing blocks of an image in the image database is

$O\left(\frac{MNM_pN_p}{d^4}\right)$. We find that it is time-consuming for comparing blocks in the procedure of querying by partial image. Hence, we propose another image retrieval scheme using projection histogram of edge information to reduce the complexity needed for comparing blocks in the query procedure.

Table 2. Merged Edge types from Table 3.2.

Vertical-dominant	$\lvert F(0,1)\rvert > \lvert F(1,0)\rvert$ $\lvert F(0,1)\rvert \geq TH$	$F(1,0) > 0$	
		$F(1,0) < 0$	
Horizontal-dominant	$\lvert F(0,1)\rvert < \lvert F(1,0)\rvert$ $\lvert F(1,0)\rvert \geq TH$	$F(0,1) > 0$	
		$F(0,1) < 0$	
Vertical	$F(1,0) = 0$	$\lvert F(0,1)\rvert \geq TH$	
Horizontal	$F(0,1) = 0$	$\lvert F(1,0)\rvert \geq TH$	
Diagonal	$\lvert F(0,1)\rvert = \lvert F(1,0)\rvert$ $\lvert F(0,1)\rvert \geq TH$	$F(1,0)\times F(0,1) < 0$	
		$F(1,0)\times F(0,1) > 0$	

4. CONTENT-BASED IMAGE RETRIEVAL SCHEME USING PROJECTION HISTOGRAM OF EDGE INFORMATION

In order to speed up the image retrieving, we adopt the traditional histogram method. In general, the histogram method usually focuses on the amount of data shown in an image. Although the histogram method can provide the fast retrieving, it ignores the spatial relationship among image data. Thus, we propose a new projection histogram method that not only provides fast retrieving but also makes use of the spatial relationship.

4.1 Projection Histogram of Edge Information

Assume that the blocks of the image we are interested have been classified into edge types according to the edge detection method introduced in the previous section. The basic idea of the projection histogram is to count the number of blocks with each edge type in the vertical or horizontal direction. For the image with $M \times N$ DCT blocks, the number of elements for the projection histogram P_V in the vertical direction is N. That is, $P_V = \{x_1, x_2, \ldots, x_N\}$, where x_i = the number of blocks with the specified edge type block in the vertical position i. Fig. 4 shows an example of edge type blocks information and its corresponding projection histogram for edge type 3.

4.2 Retrieving Images using Projection Histogram of Edge Information

In this section, we will introduce the way of retrieving images using the projection histogram of edge information. When retrieving an image, the user's query image is first evaluated to obtain the corresponding projection histogram in the vertical and horizontal directions. The edge type with the largest number of blocks in the image is considered as the major edge type to find the desired images. For each image in the database, the projection histograms for the major edge type of the query image are used to compute the distance score. The distance score *total_dist* is defined as follows

$$total\_dis = dis\,(V) + dis\,(H), \quad (6)$$

$$dis\,(V) = \lVert DP_v - P_v \rVert,$$

$$dis\,(H) = \lVert DP_H - P_H \rVert,$$

where DP_V and P_V are the vertical projection histograms of the candidate image in the database and the query image and DP_H and P_H are the horizontal projection histograms of the candidate image in the database and the query image.

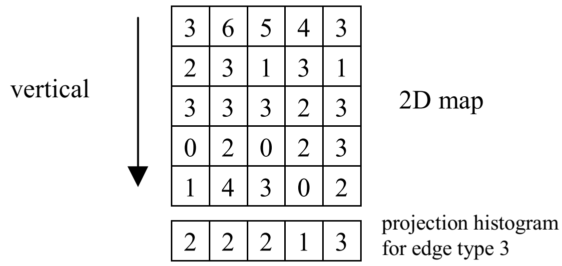


Fig. 4. The 2D map of edge information and its corresponding projection histogram of edge block type 3 in the vertical direction.

For the query using partial images, the evaluation of the distance score is performed for each possible location in the original images of database from left to right and from top to bottom. The smallest distance score is then considered as the final distance score of the image. After the distance score is evaluated for each images in the database, the images with smaller distance scores are then returned to the user as the candidate images. In the sketch querying, the processing steps are similar to the query for partial images except that the query image is changed into the user sketch image.

4.3 Complexity of Query by Partial Image with Projection Histogram

In the previous section, we know that the complexity needed for comparing blocks of an image in the query steps is $O\left(\frac{MNM_pN_p}{d^4}\right)$. By using the projection histogram proposed in this section, the complexity can be effectively reduced. In this proposed method, the number of block comparison needed in the match step of each possible position is $\left\lceil \frac{M_p}{d} \right\rceil \times \left\lceil \frac{N_p}{d} \right\rceil$. The total number of possible matching positions is $\left\lceil \frac{M - M_p}{d} \right\rceil \times \left\lceil \frac{N - N_p}{d} \right\rceil$. Hence, the complexity for comparing blocks of an image in the image database is $O\left(\frac{MN(M_p + N_p)}{d^3} \right)$. The result is apparently better than the one without using projection histogram.

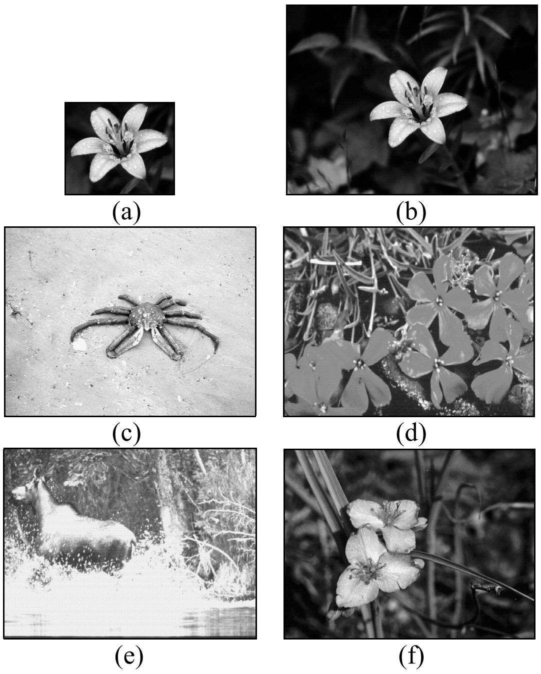


Fig. 5. The query results by the partial image. (a) The query image with size 280×240. (b)-(f) The top five candidate images.

5.EXPERIMENTAL RESULTS

The experimental results of our proposed method are shown in this section. There are 2,000 JPEG still images with size 640×480 in our image database. Both the query by partial images and user sketch images query are performed in our experiments. The number of query images used in our experiment is 20. The threshold used to avoid classifying a block into weak edge block type is 64. For each query image, the top five candidates are shown in the figures.

In our experiments, the DCT block partition of query images is different from those in the image database. If the partitions are the same, the accurate image will be correctly retrieved from the image database. Hence, the partition of 8×8 blocks for the query images is different from the original partition in the image database in our experiment in order to show the robustness of our method. The query results of edge-based retrieval method are shown in Figs. 5-6. Figs. 7-8 show the query results of our proposed projection histogram method. Let n_r and n_c denote the number of images retrieved and the number of correct retrieval result, respectively. The precision is defined as precision = $\frac{n_c}{n_r} \times 100\%$. In our experiments, the retrieved image is considered as a correct retrieval result if the desired image is in the top twenty candidate images. According to the experimental results, the precision of our proposed method is 90% when the projection histogram technique is not used and 85% when the projection histogram technique is used. For the user sketch retrieving method, the precision is 85% when the projection histogram technique is not used and 75% when the projection histogram technique is used. We can find that our proposed methods are useful for the partial image query, even though the query images are with the different DCT partition from the one of the original image. Meanwhile, not only the result of query by partial image is good, but also the result of query by user's sketch image is good in our experiments for both proposed methods.

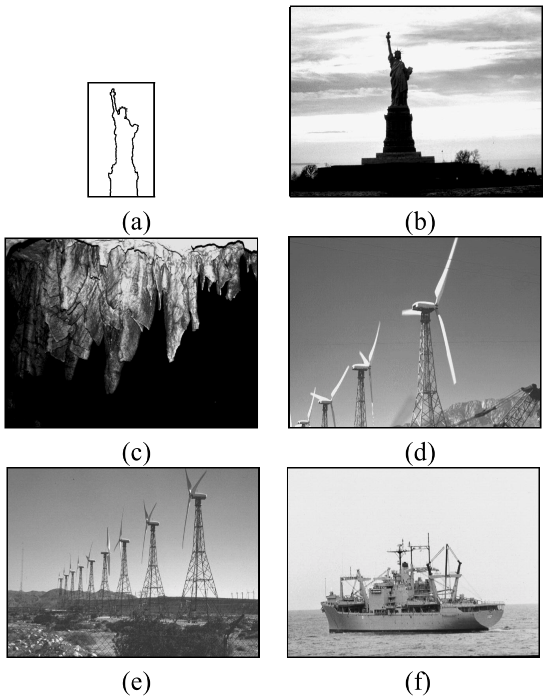


Fig. 6. The query results by user's sketch image. (a) The user's sketch image with size 160×280. (b)-(f) The top five candidate images.

6.CONCLUSION

In this paper, we propose an efficient and accurate image retrieval technique using the edge-based retrieval method and the projection histogram of edge information method. In our proposed method, the edge information of images is used as the main key when retrieving images. The edge detection for each block in the image is directly performed in the DCT frequency domain. In order to improve the retrieving speed, we propose the projection histogram method that also makes use of the spatial relationship. User sketch retrieval method makes the retrieving method with a higher flexibility. Experiment results show that our proposed methods can be efficiently used in the image databases and the performance is good. Hence, our proposed methods can be directly used in the image retrieval system in practice.

7.REFERENCES

[1] M. Fickner, H. Sawhney, W. Niblack, J. Ashkey, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: The QBIC system," *IEEE Computer*, vol. 28, no. 9, pp. 23-32, Sept. 1995.

[2] J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. C. Jain, and C. Shu. "Virage image search engine: an open framework for image management," in *Symposium on Electronic Imaging: Science and Technology – Storage and Retrieval for Image and Video Database IV*, vol. 2670, pp. 76-87, IS&T/SPIE, 1996.

[3] V. E. Ogle and M. Stonebraker, "Chabot: Retrieval from a relational database of images," *IEEE Computer*, vol. 28, no. 9, pp. 40-48, Sept. 1995.

[4] R. K. Srihari, "Automatic indexing and content-based retrieval of captioned images," *IEEE Computer*, vol. 28, no. 9, pp. 49-56, Sept. 1995.

[5] S. W. Smoliar and H. Zang, "Content-based video indexing and retrieval," *IEEE Multimedia Mag.*, vol. 12, pp. 62-72, Summer 1994.

[6] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, and C. Faloutsos, "The QBIC project: Queryimg images by content using color, texture, and shape," in *Storage and Retrieval for Image and Video Database*, vol. 1908, SPIE, Feb. 1993.

[7] A. Pentland, R. W. Picard, and S. Sclaroff, "Photobook: Tools for content-based manipulation of image database," in *Proceeding of the SPIE Storage and Retrieval Image and Video Database II*, Feb. 1994.

[8] J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack, "Efficient color histogram indexing for quadratic form distance functions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 7, pp. 729-736, 1995.

[9] T. Tako, T. Kurita, N. Otsu, and K. Hirata, "A Sketch retrieval method for full color image database: Query by visual example," in *Proceeding of 11th International Conference on Pattern Recognition*, pp. 530-533, 1992.

[10] G. L. Gimel'farb and A. K. Jain, "On retrieving textured images from an image database," *Pattern Recognition*, vol. 29, no. 9, pp. 1461-1483, 1996.

[11] V. N. Gudivada and V. V. Raghavan, "Design and evaluation of algorithms for image retrieval by spatial similarity," *ACM Trans. on Information Systems*, vol. 13, no. 2, 1995.

[12] D. H. Huang and C. L. Huang, "A content-based image retrieval system," *Proceedings of IPPR Conference on Computer Vision, Graphics, and Image Processing*, Taiwan, pp. 259-266, 1996.

[13] J. R. Smith and S. F. Chang, "Tools and techniques for color image retrieval," In *Symposium on Electronic Imaging: Science and Technology – Storage and Retrieval for Image and Video Database IV*, vol. 2670, pp. 426-437, IS&T/SPIE, 1996.

[14] B. M. Mehtre, *et al.*, "Color matching for image retrieval," *Pattern Recognition Letters*, vol. 16, pp. 325-331, 1995.

[15] M. Stricker and A. Dimai, "Color indexing with weak spatial constrains," *Symposium on Electronic Imaging: Science and Technology – Storage and Retrieval for Image and Video Database IV*, vol. 2670, pp. 29-41, IS&T/SPIE, 1996.

[16] J. W. Kim and S. U. Lee, "A transform domain classified vector quantizer for image coding," *IEEE Trans. Circuits and Syst. Video Technol.*, vol. 2, no. 1, pp. 3-14, 1992.

[17] M. H. Lee and G. Crebbin, "Classified vector quantization with variable block-size DCT models," in *IEEE Proceeding of Vis. Image Signal Process.*, vol. 141, no, 1, pp. 39-48, 1994.

[18] B. Shen and I. Sethi, "Direct feature extraction from compressed images," in *SPIE Proceedings Storage and Retrieval for Image and Video Databases IV*, vol. 2670, 1996.

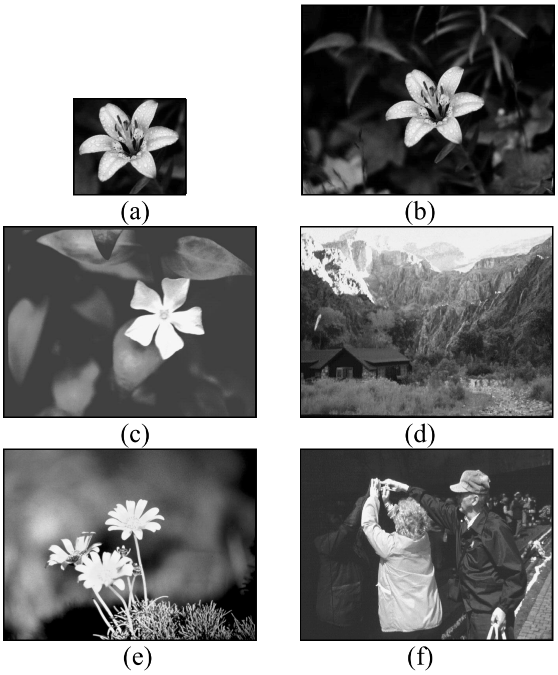


Fig. 7. The query results by the partial image using projection histogram method. (a) The query image with size 280×240. (b)-(f) The top five candidate images.

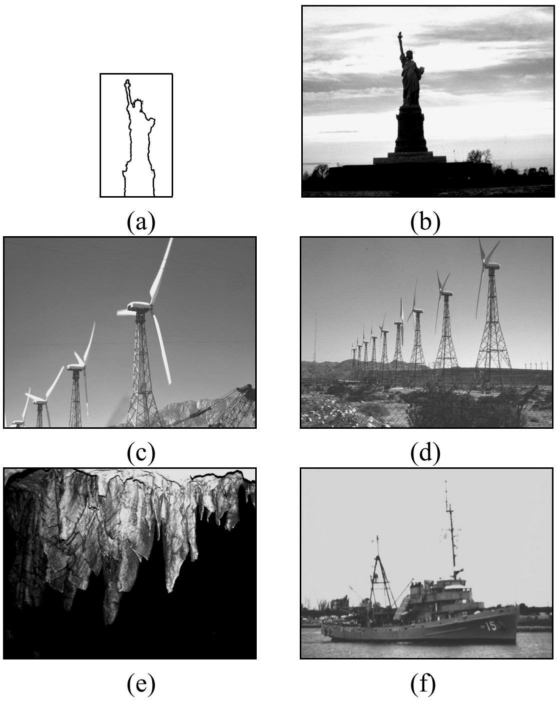


Fig. 8. The query results by user's sketch image using projection histogram method. (a) The user's sketch image with size 160×280. (b)-(f) The top five candidate images.