

A Vision-Based Navigation System for Tamsui Historical Buildings

Wen-Chuan Wu¹, Zi-Wei Lin¹, and Yi-Hui Chen²

¹Dept. of Computer Science & Information Engineering, Aletheia University, Taipei, Taiwan, R.O.C.

E-mail: au4387@au.edu.tw Tel: +886-2-26212121

²Dept. of Information Science & Applications, Asia University, Taichung, Taiwan, R.O.C.

E-mail: chenyh@asia.edu.tw Tel: +886-4-23323456

Abstract—This paper proposes a vision-based navigation system for Tamsui historical buildings. Most travelers in Tamsui take a picture of the historical buildings by using their mobile phone's camera. Our system is designed to automatically extract the color and edge features out of images, and then search similar images from an image database by comparing with these features. For the searched buildings, their contextual navigation and 3D model will be displayed or projected in a screen. This makes the navigation process easier and more interesting. Experimental results showed that the proposed scheme retrieves the target building images effectively and accurately.

I. INTRODUCTION

Mobile devices are becoming more and more popular in recent years. Such handheld devices [12] are usually pocket-size computing machines. Typically, it has a small display screen with touch input and a miniature keyboard, while its weight is almost less than six hundred grams [7]. For example, Smartphones, tablet computers, and personal digital assistants (PDA) are the products of this type of device. Reasons for the prevalence of its usage include it being equipped with Wi-Fi, Bluetooth and GPS capabilities that can allow connections to the Internet, enabling users to quickly obtain a variety of information. In addition, a camera with digital video is also provided in these devices such that people can create images and videos easily and then transmit them via the public and open network.

Generally speaking, multimedia data including text, audio, still image, animation, video, or the combination are huge in size. That is a real challenge in terms of data computation and storage. At the same time, digital data, images and videos especially, are becoming more commonly used everywhere while the mobile devices become more popular. In order to access them, retrieving and browsing large-scale digital data have become a very difficult and time-consuming task. Content-based image retrieval [11],[15],[16], CBIR for short, is one of the most efficient techniques for retrieving relevant images from a large unlabelled database. It is an automatic query system that is different from keyword indexing [3]. Each image in a database is preprocessed to extract its unique features. When a user gives a query image, the system will automatically extract the relevant images by comparing with those representative features. Hence, automated retrieval of image database becomes possible and hassle-free. Fig. 1 shows the architecture of a typical CBIR system.

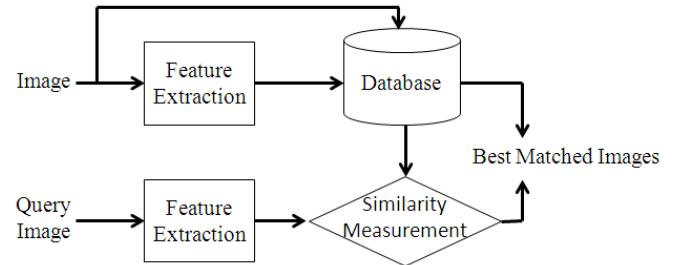


Fig. 1 The simple architecture of a typical CBIR system.

The earliest mention of content-based image retrieval can be found in the literature [9]. And, the well-known image retrieval system in commerce is the QBIC developed by IBM [5]. Another commercial system for image and video retrieval is VIRAGE [6]. Most of these available systems have a very similar architecture for browsing and archiving images by using some efficient and low-level features such as color, shape, texture, spatial features, their combinations, etc. Color feature is the most frequently referred to among them. The simplest color histogram scheme [14] is easy to implement by counting the appearance of pixel values. But, it includes only the global color distribution instead of considering the spatial relationship between pixels. The local spatial relationship in [3] is taken into account by using the color difference co-occurrence matrix. As regards texture, that is usually used to represent the detailed structure of the image content. Quadtree segmentation in [2],[4],[15],[16] is one of the famous texture descriptors. Recent approaches [1],[8],[11],[15] of image retrieval are almost focusing on the object-based features owing to the demand of digital video.

In this paper, we aim at emphasizing the application of the image retrieval technique. We intend to design a vision-based navigation system to help travelers to see and appreciate the historical buildings or spots. During a tour, travelers could use a mobile device to take photos for those historical buildings then upload these images to the system server through the network. Our system would automatically extract the color and edge features of images and find out the most similar images from a large image database. The system would display the contextual navigation and 3D model further for the purpose of vision guide. The remainder of this paper is organized as follows: First, we briefly review the quadtree-

based texture feature [15] and introduce the edge detection technique in Section II. In Section III, we describe our vision-based navigation system in detail as to how an image retrieval scheme can help travelers navigate Tamsui historical buildings. Then, the experimental results are presented and discussed in Section IV. Conclusions are finally drawn in Section V.

II. RELATED WORKS

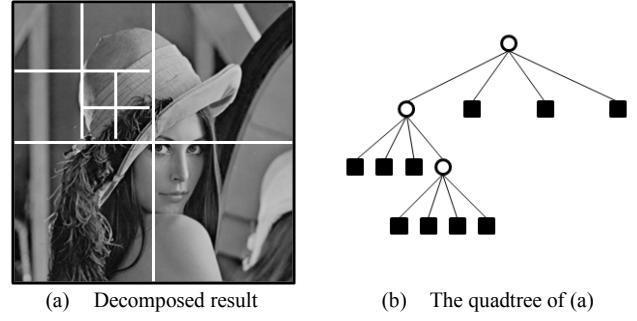
A. Quadtree-Based Texture Feature

Tseng et al. proposed a color and texture feature extraction scheme [15] that employs the quadtree decomposition. The technique of the quadtree decomposition employs a kind of hierarchical spatial segmentation. It is to successively divide an image block into four quadrants depending on the block complexity. If an image block is smooth, the pixels in it are homogeneous. Usually, that will stop splitting such a block. In [15], the gray-level values of each color pixels are acquired first to assist in the quadtree decomposition. Next, we treat the whole gray-level image as a block unit and then compute its average gray-level $\text{avg}(x, y)$, which is defined in (1), wherein $I(x+i, y+j)$ is the gray-level value at position $(x+i, y+j)$ that is oriented to the coordinate (x, y) at the top left corner of the block and n is the block size. The average gray-level is used in (2) to determine if the block is homogeneous or not. In other words, each pixel in that homogeneous block is varied among some specified threshold TH . In case it is not a homogeneous block, the segmentation routine will recursively repeat on it, and vice versa.

$$\text{avg}(x, y) = \frac{\sum_{i=1}^n \sum_{j=1}^n I(x+i, y+j)}{n \times n}. \quad (1)$$

$$|\text{avg}(x, y) - I(x+i, y+j)| \leq TH, \text{ where } 0 \leq i, j < n. \quad (2)$$

Fig. 2 is a partial result of the quadtree decomposition for the image “Lena”, where square nodes in Fig. 2(b) represent homogeneous blocks while circle nodes represent non-homogeneous blocks. It is quite obvious that a complex area has more leaf nodes than that in a smooth area. After the quadtree decomposition, 7×7 RGB pixels are sampled for each homogeneous block. These forty-nine pixels are transformed into HSV color space and then the average H, S, and V attributes are calculated as the color features. As for the texture features, the technique of the statistical co-occurrence matrix is used to generate five features. They are the angular second moment (ASM), inverse element different moment (IEDM), correlation, variance, and entropy, respectively. In order to reduce the computation load, the authors quantify each RGB pixel value uniformly into 32 gray levels. The size of the co-occurrence matrix is thereby 32×32 , instead of 256×256 . Finally, an eight-element feature including color as well as texture is represented for each homogeneous block, where each value is normalized between 0 and 1. Further, the authors cluster all the feature vectors in order to represent a few major objects of the image for object-based retrieval. Each image in the database extracts six feature vectors at most. And, these vectors are applied in the similarity process.



(a) Decomposed result
(b) The quadtree of (a)

Fig. 2 A quadtree decomposition of the image “Lena”.

B. Edge Detection

In an image, edges provide rich information of contour content and thus are usually used as unique characteristics. Edge points can be easily detected by using some morphology operations, which aims at identifying where the intensity changes rapidly and sharply. The famous operators [13], such as Prewitt, Sobel, and Canny, utilize spatial gradient magnitude in finding the strength and direction of edges. Take the operator, Sobel, as an example. Typically, it calculates the approximate gradient magnitude G , shown in (3), at each pixel point (x, y) of a grayscale image. Note that G_V and G_H are the first order derivatives that are computed by two 3×3 convolution operations in (4), where B stands for the processing 3×3 block at location (x, y) . If the value of G exceeds a predefined threshold, the pixel (x, y) is assigned as an edge point. Moreover, it may be a vertical edge point if the value of G_V is far larger than G_H , and vice versa.

Afterwards, the 3×3 masks of (4) are slightly modified to obtain the diagonal direction. The two additional Sobel masks for detecting intensity discontinuities in diagonal directions are shown in (5), wherein G_{d+} and G_{d-} are the information of $+45^\circ$ and -45° diagonal directions, respectively. Obviously, these masks mentioned above can be applied to simply detect four distinct edge directions. Other alignments will be possibly obviated, non-directional block and chaotic block for example. Fig. 3 shows some examples of the four directions, such as the horizontal edges, vertical edges, $+45^\circ$, and -45° diagonal edges, and other block alignments. Clearly, edge points are situated where image brightness changes sharply. Therefore, edge detection is a fundamental tool that can be used in image processing, computer vision, and machine vision [13],[17], particularly in the areas of feature detection and extraction.

$$G = \sqrt{G_V^2 + G_H^2}. \quad (3)$$

$$G_V = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \times B, G_H = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \times B. \quad (4)$$

$$G_{d+} = \begin{bmatrix} 0 & +1 & +2 \\ -1 & 0 & +1 \\ -2 & -1 & 0 \end{bmatrix} \times B, G_{d-} = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & +1 \\ 0 & +1 & +2 \end{bmatrix} \times B. \quad (5)$$

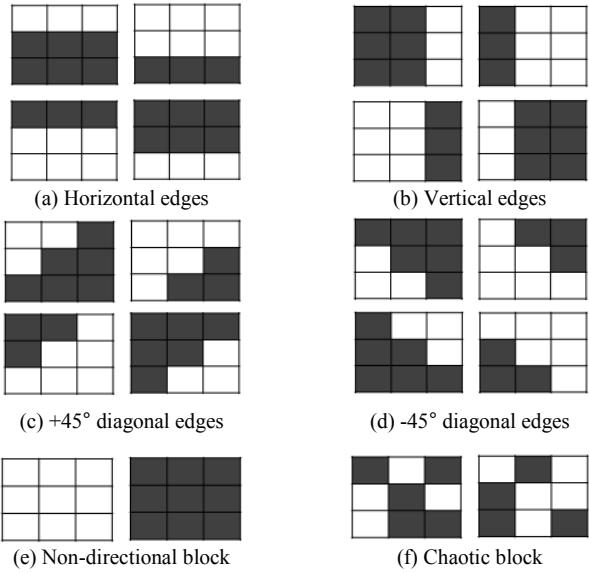


Fig. 3 Example of edge direction for a 3×3 block.

III. THE PROPOSED SCHEME

The Tamsui district in Taiwan is the veritable cradle of Mackay culture [18]. The place is full of centenary buildings, such as Oxford College [19], Fort San Domingo, Hobe Mackay Hospital, the Chief Taxation Officer's residence and so on. Hence, pictures of these historic spots, ancient buildings especially, are taken frequently by tourists arriving in Tamsui. In order to guide a large crowd of travelers in Tamsui, we attempt to develop a vision-based navigation system through mobile phones. Fig. 4 shows an overview of our navigation system. After capturing a scene, a traveler can use a mobile phone to upload the image taken through a wireless network using Wi-Fi or 3G. When our remote system server receives it, the proposed feature extraction scheme starts to perform the database retrieval process for the query image. Afterwards, the relevant images that are highly similar to the searched one are sequentially sent to the traveler and then displayed in the phone screen, even in a projector screen. Certainly, the contextual navigation and 3D model for each image can also be displayed.



Fig. 4 Overview of the proposed vision-based navigation system.

Fig. 5 is the proposed feature extraction scheme including the two-stage retrieval process. Firstly, it is to extract the color features of every image in the database. Note that these color features are used to filter out extremely irrelevant images. This can greatly save on the execution time for the edge feature comparison and similarity measurement procedures to be done later. Secondly, it will extract the edge features from the remaining images. In the first extraction procedure, we adopt a pixel self-clustering manner to reduce the colors and then produce the most representative RGB colors. This way is rather similar to the VQ codebook training approach [10]. Each RGB color pixel in an image is regarded as a training data. In the beginning, pick N training data randomly and let them be the initial centroid points. Next, cluster all of the training data around these points to form N new groups. Then, compute the training data of every set averagely to obtain its corresponding new centroid point. Repeat the clustering steps mentioned above until either centroid points do not change or the change is very small. In the end, the centroid points are exactly N representative colors which are then used to replace other color pixels of each group.

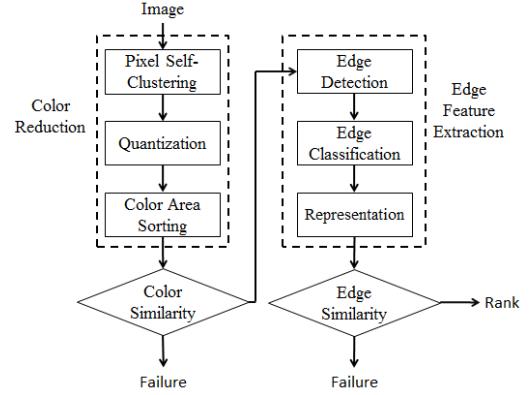


Fig. 5 The proposed feature extraction process.

Here, only t RGB colors are utilized in color similarity measurement in order to speed up the execution time, where $t < N$. For this reason, we sort the colors in decreasing order according to their area to acquire the first t important colors. Assume that the t colors for images A and B are $\{C_{a1}, C_{a2}, \dots, C_{at}\}$ and $\{C_{b1}, C_{b2}, \dots, C_{bt}\}$, respectively. In order to judge the similarity between them in terms of color, we employ a cumulative minimum distance. If that distance is not larger than a predefined threshold value TH_c , it can be said that the colors of the two images are absolutely similar to each other. The cumulative minimum distance is just the sum of the difference between each color C_{ai} and its most similar color C_{bj} . Its formula is shown in (6), where the term $\min(\cdot)$ is a function to find the smallest value in a given set of data. In cases wherein an image in the database is not similar to the query image, that is to say the inequality in (6) is violated, then that image will be excluded from the candidates for the possible desired image. This way is quite efficient to speed up the execution time of the following feature extraction process.

$$\sum_{i=1}^t \min_{\forall j} (|C_{ai} - C_{bj}|) \leq TH_c, \text{ where } 1 \leq j \leq t. \quad (6)$$

$$E(x,y) = \begin{cases} 1 (HE), & \text{if } G > TH_e \& \& G_v >> G_H \\ 2 (VE), & \text{if } G > TH_e \& \& G_v << G_H \\ 3 (PDE), & \text{if } G_D > TH_e \& \& G_{d+} >> G_{d-} \\ 4 (NDE), & \text{if } G_D > TH_e \& \& G_{d+} << G_{d-} \\ 5 (NP), & \text{if } \text{var}(B) < TH_e \\ 6 (CP), & \text{if } \text{var}(B) > TH_e \end{cases}. \quad (7)$$

$$M(i,j) = \Pr(E(x,y)=i | E(x,y+1)=j), \text{ where } 1 \leq i, j \leq 6. \quad (8)$$

During the second extraction procedure, there are three steps needed to perform on those possible candidate images. They are edge detection, classification, and representation in order. The purpose of edge detection is to acquire edge information of an image. Strictly speaking, edges are the high frequency information which describes the boundaries of objects in an image. Here, we use the Sobel operator to decide where an edge point $E(x, y)$ is and then classify it into six types by using (7). Those are horizontal edges (HE), vertical edges (VE), $+45^\circ$ diagonal edges (PDE), -45° diagonal edges (NDE), non-directional pixels (NP), and chaotic pixels (CP), respectively, where $G_D = \sqrt{G_{d+}^2 + G_{d-}^2}$ and $\text{var}(B)$ is the variance of 3×3 block B at location (x, y) . Assume the variance is smaller than a specified constant. It means the pixel at (x, y) is situated at a non-directional block, such as Fig. 3(e). On the contrary, it is situated at a chaotic block, such as Fig. 3(f). Note that it must be done in advance to convert the color image to grayscale before doing the above-mentioned edge detection.

After this, the representation step is conducted to simply take down the important edge patterns or features in an image. Here, we compute the frequency of occurrence for each edge type for a start. That will produce a six-dimensional vector $EV = (N_{HE}, N_{VE}, N_{PDE}, N_{NDE}, N_{NP}, N_{CP})$, where N_x indicates the number of edge points appearing in type x . In addition, a co-occurrence matrix is also derived to represent the edge spatial dependence of an image. This square matrix [15] adopts mainly the concept of conditional probability in statistics and it is able to further highlight the details of an image. Usually, it being a global feature, it is widely used in texture-based image retrieval. The prior edge types will also be represented by a co-occurrence matrix $M(i, j)$, which is defined in (8). In the matrix M , each entry (i, j) indicates the probability of occurrence of the pair of edge types i and j adjacently in the image. We design six edge types, so the number of matrix elements is thirty-six in total. Obviously, its size is far smaller than the original image size. That can thereby save the following similarity measurement time.

$$\text{Diff}(P, Q) = d1(P, Q) + d2(P, Q). \quad (9)$$

$$d1(P, Q) = \sum_{x \in \{HE, VE, PDE, NDE, NP, CP\}, x \neq k} (|N_x^P - N_x^Q| \times \varepsilon_1) + |N_k^P - N_k^Q| \times \varepsilon_2. \quad (10)$$

$$d2(P, Q) = \sum_{i, j=1, i \neq k, j \neq k}^6 (|M^P(i, j) - M^Q(i, j)| \times \varepsilon_1) + |M^P(k1, k2) - M^Q(k1, k2)| \times \varepsilon_2. \quad (11)$$

The procedure of edge similarity measurement is then to determine whether two given images are similar. Let Q be the query image and P be any an image, which passes the test of color similarity, in the database. We adopt a 1-norm Euclidean distance formula to compare with the resemblance of images Q and P , as shown in (9), (10), and (11). In which $d1(P, Q)$ and $d2(P, Q)$ are the differences of frequency and co-occurrence matrix of edge types, respectively. Here, we set two coefficients ε_1 and ε_2 to have a distinct important degree for edge features. N_k is the largest value of the vector EV in the query image and $M(k1, k2)$ is also the largest value in the query image's matrix. Therefore, that is really significant in terms of edge features and the value of ε_2 will be smaller than that of ε_1 . We set $\varepsilon_1=1$ as well as $\varepsilon_2=0.7$ in the later experiments. In (9), the formula was conducted to measure the distance between the two images. If the difference value $\text{Diff}(P, Q)$ is small enough, we assume that image P is quite alike to the query image. Consequently, a set of similar images with smaller 1-norm Euclidean distance is selected from the image database. Note that the proposed feature extraction scheme including color and edge is performed on only partial RGB pixels in order to accelerate the performance. These pixels are situated at the specific region of interest (ROI).

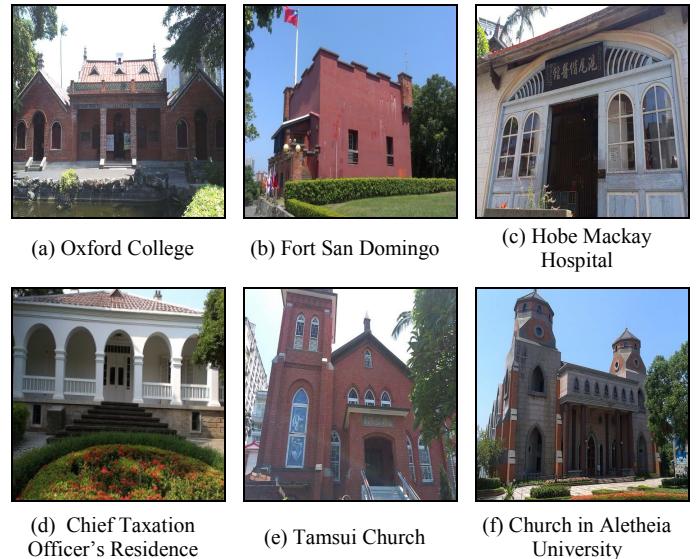
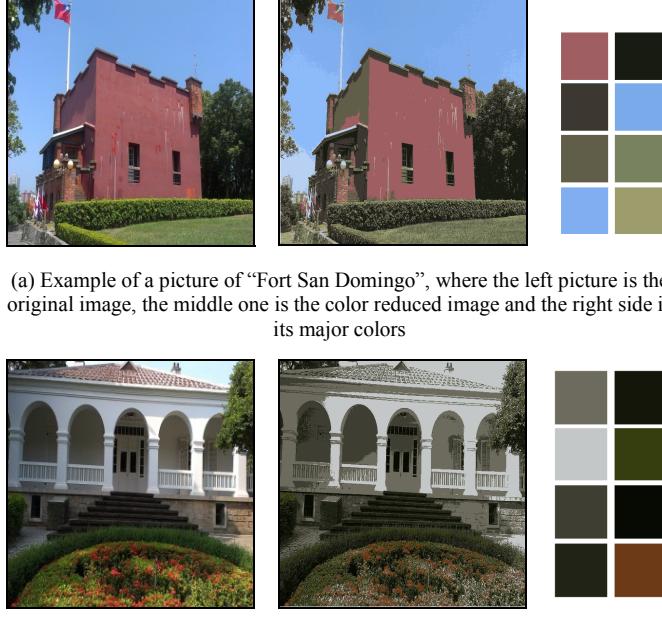


Fig. 6 Six classified Tamsui famous architectures.

IV. EXPERIMENTAL RESULTS

In our experiments, a database of 500 RGB color images is used to evaluate the performance of the proposed extraction scheme. This database has two parts collected, in which one part consists of 10 classes and each class has 13 relevant color images while the other part consisting of 370 images from the collection of the website [20] is miscellaneous to elaborate the pictures stored in the database. All of them are 512×512 color pixels. Due to the limited printed pages, Fig. 6 shows some pictures of six classes of famous building in Tamsui. An image in each class is taken as the query image in turn. The resulting images extracted by the query image will be sorted

by ranking according to the distance of (9). Certainly, the most relevant images are at the top. Our experiments were performed on the Wintel platform with the C# programming language to simulate the operation of mobile phone users.



(a) Example of a picture of “Fort San Domingo”, where the left picture is the original image, the middle one is the color reduced image and the right side is its major colors
(b) Example of a picture of the “Chief Taxation Officer’s Residence”, where the left picture is the original image, the middle one is the color reduced image and the right side is its major colors

Fig. 7 The 8-color quantization results by using self clustering

In practice, we first processed all 500 images using the pixel self-clustering technique to reduce the colors. Fig. 7 is the result of the color reduction process, where that produced 8 major colors ($N=8$). It can be seen clearly that the resulting images in the middle of Fig. 7(a) and Fig. 7(b) are a bit similar to the original image, respectively. For this reason, the N colors became the important features for the image. Table I displays the results of elimination rate (ER) by using the color features under different values of N and t colors, where ER is defined as follows and r is the number of relevant but excluded images:

$$ER = \frac{\text{the number of excluded images to the query}}{\text{the number of images in the database}}. \quad (12)$$

Obviously, the value of ER is associated to the coefficients N and t . When the values of N and t are larger, the number of images to be excluded will be increased gradually. The reason is that, during the color comparison, more information of image content will be referred to such that the performance is more accurate. However, that will also entail a longer computation time when the values of N and t are larger. Here, the values $N=8$ and $t=3$ are thereby suggested. In brief, the proposed color features are able to filter out most of the extremely impossible images to improve the efficiency.

TABLE I AVERAGE ELIMINATION RATE OF SELF-CLUSTERING COLOR REDUCTION UNDER DIFFERENT VALUES OF N AND t

N	$t=1$		$t=2$		$t=3$	
	ER	r	ER	r	ER	r
4	266	2	387	4	324	1
8	337	2	372	4	308	1
16	297	2	366	5	388	4
Average ER	60.0%		75.0%		68%	

TABLE II PRECISION RESULTS UNDER DIFFERENT EDGE THRESHOLDS

Class	1	2	3	4	5	6
$TH_e = 40$	23.0%	76.9%	61.5%	53.8%	38.4%	46.1%
$TH_e = 80$	30.7%	69.2%	61.5%	61.5%	38.4%	46.1%
$TH_e = 120$	30.7%	84.6%	61.5%	53.8%	38.4%	46.1%
$TH_e = 200$	23.0%	69.2%	69.2%	53.8%	30.7%	38.4%
Average	26.9%	75.0%	63.4%	55.7%	36.5%	44.2%

Table II displays the precision results of the six classes under different edge thresholds TH_e while we adopt the values of $N=8$ and $t=3$ in the color features and let the number of retrieved image be thirteen. Note that for the precision and recall rates, the most frequently used evaluation criterion [15],[16] are defined as follows:

$$\text{precision} = \frac{\text{the number of relevant images to the query}}{\text{the number of retrieved images}}. \quad (13)$$

$$\text{recall} = \frac{\text{the number of relevant images to the query}}{\text{the number of all relevant images}}. \quad (14)$$

Table II is used to evaluate the ability of our proposed edge features. We observed that larger or smaller threshold values produce more inexact results in precision. Hence, it is suggested to use the value ranging from 80 to 120.

Figs. 8 and 9 reveal the retrieval results of our proposed scheme using $TH_e=120$ when searching images of “Fort San Domingo” and the “Chief Taxation Officer’s Residence”, respectively. In which, the image at the top left corner is the query image and the others are the most relevant images in ranking. Clearly, the one in rank 1 must be the query image and most of the first ten ranks are quite similar to the query image. The precision of the two query image are 84.6% and 53.8%, respectively. In addition, we found that the precisions of “Oxford College” and “Tamsui Church” images are relatively worse. The reason is because that the illumination in the two image classes is so bright to dim the buildings. This way makes the color of building objects darker, and further decrease the ability of color reduction. Hence, our proposed feature extraction maybe not suitable for those back-lighted images. Fig.10 shows the precision and recall curves of the proposed feature extraction scheme for the six classes. We submitted thirteen queries in total for each class of pictures and varied the number of retrieved image. Precision and recall typically go in opposite directions. When the query is broad, the recall is high but precision is low, and vice versa. As can

be seen in this figure, some of six classes have good precision and recall, “Fort San Domingo” and “Hobe Mackay Hospital” images especially.



Fig. 8 Retrieved results when searching for “Fort San Domingo” image.

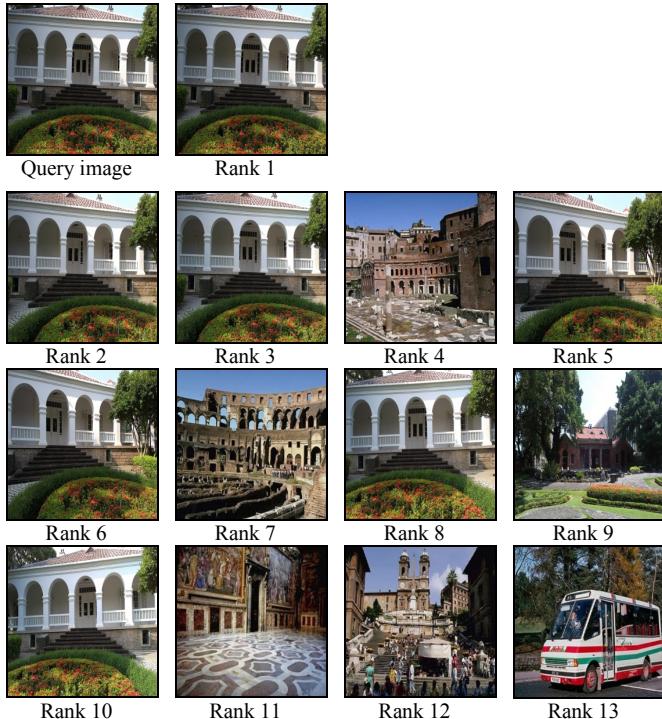


Fig. 9 Retrieved results when searching the image of the “Chief Taxation Officer’s Residence”.

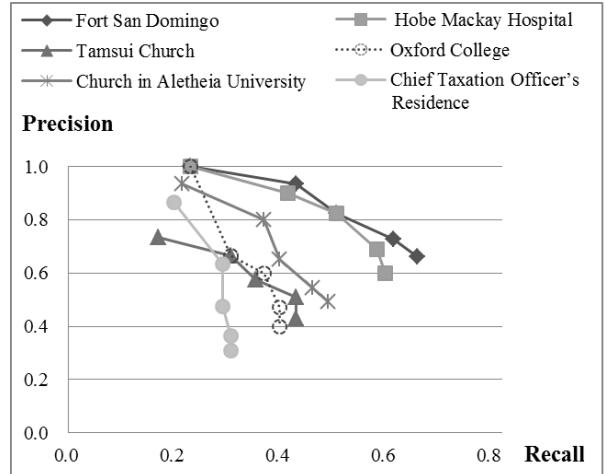


Fig. 10 Precision and recall curves

V. CONCLUSIONS

This paper presents a vision-based navigation system for historical buildings in the Tamsui region. It utilizes the technique of content-based image retrieval to identify a similar building image and then display its historical information apart from the 3D model. When searching for an image, the color features are first acquired by using self-clustering color reduction. These features are used to efficiently filter out a pile of irrelevant images in a database. The edge features are then drawn out according to its edge classification and are represented in the format of the co-occurrence matrix. As shown in the experiments, the proposed feature extraction scheme retrieves the target images effectively. In terms of color reduction, 68% of the images in the database are usefully filtered out on average. At the same time, the precision of the proposed feature extraction scheme in the first three ranks is quite good. Future work will focus on building object shape descriptors in order to abate the interference of obstacles.

ACKNOWLEDGMENT

This work was supported by National Science Council, Taiwan, Republic of China, under the Grant NSC 99-2632-H-156-001-MY3.

REFERENCES

- [1] K.T. Chen, K.H. Lin, Y.H. Kuo, Y.L. Wu, and W.H. Hsu, “Boosting image object retrieval and indexing by automatically discovered pseudo-objects,” *Journal of Visual Communication and Image Representation*, Vol. 21, No. 8, pp. 815-825, 2010.
- [2] H.H. Chen, H.T. Sheu, and J.J. Ding, “Quadtree classified vector quantization based image retrieval scheme,” *Proceedings of the 18th IEEE International Conference on Image Processing*, Brussels Belgium, pp. 3625-3628, 2011.
- [3] C.C. Chang, W.C. Wu, and Y.C. Hu, “Content-based color image retrieval system using color difference features,” *Proceedings of the 2nd International Conference on Future Generation Communication and Networking Symposia*, Vol. 3, Hainan Island China, pp. 181-184, 2008.

- [4] E. El-Qawasmeh, "A quadtree-based representation technique for indexing and retrieval of image databases," *Journal of Visual Communication and Image Representation*, Vol. 14, No. 3, pp. 340-357, 2003.
- [5] M. Flicker, H. Sawhney, W. Niblack, J. Ashley, 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, 1995.
- [6] A. Gupta and R. Jain, "Visual information retrieval," *Communications of the ACM*, Vol. 40, No. 5, pp. 70-79, 1997.
- [7] C.W. Hanson, "Chapter 2: mobile devices in 2011," *Library Technology Reports*, Vol. 47, No. 2, pp. 11-23, 2011.
- [8] D. Hoiem, R. Sukthankar, H. Schneiderman, and L. Huston, "Object-based image retrieval using the statistical structure of images," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, Washington DC USA, pp. 490-497, 2004.
- [9] T. Kato, "Database architecture for content-based image retrieval," *Proceedings of SPIE on Image Storage and Retrieval Systems*, Vol. 1662, California USA, pp. 112-123, 1992.
- [10] Y. Linde, A. Buzo, and R.M. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, Vol. 28, No. 1, pp. 84-95, 1980.
- [11] J. Lebrun, P.H. Gosselin, and S. Philipp-Foliguet, "Inexact graph matching based on kernels for object retrieval in image databases," *Image and Vision Computing*, Vol. 29, No. 11, pp. 716-729, 2011.
- [12] C.H. Lin, J.C. Liu, and C.W. Liao, "Energy analysis of multimedia video decoding on mobile handheld devices," *Computer Standards & Interfaces*, Vol. 32, No. 1-2, pp. 10-17, 2010.
- [13] Y.D. Qu, C.S. Cui, S.B. Chen, and J.Q. Li, "A fast subpixel edge detection method using Sobel-Zernike moments operator," *Image and Vision Computing*, Vol. 23, No. 1, pp. 11-17, 2005.
- [14] M.J. Swain and D.H. Ballard, "Color indexing," *International Journal of Computer Vision*, Vol. 7, No. 1, pp. 11-32, 1991.
- [15] S.Y. Tseng, Z.Y. Yang, W.H. Huang, C.Y. Liu, and Y.H. Lin, "Object feature extraction for image retrieval based on quadtree segmented blocks," *Proceedings of World Congress on Computer Science and Information Engineering*, Vol. 6, Los Angeles USA, pp. 401-405, 2009.
- [16] S. Wu, M.K.M. Rahman, and T.W.S. Chow, "Content-based image retrieval using growing hierarchical self-organizing quadtree map," *Pattern Recognition*, Vol. 38, No. 5, pp. 707-722, 2005.
- [17] W.C. Wu, W.T. Wong, and Z.W. Lin, "Reversible difference expansion embedding using two edge directions," *Proceedings of the Seventh International Conference on Complex, Intelligent, and Software Intensive Systems*, Taichung Taiwan, pp. 197-200, 2013.
- [18] George Leslie mackay:
http://en.wikipedia.org/wiki/George_Leslie_Mackay
- [19] Tamsui District in Taiwan:
http://en.wikipedia.org/wiki/Tamsui_District
- [20] Test database used in SIMPLIcity paper:
<http://wang.ist.psu.edu/docs/related/>