Contents-based Image Retrieval Using Color & Edge Information

Dong-Won Park[†††] · Syungog An[††] · Ming Ma[†] · Kulwinder Singh[†]

Abstract

In this paper we present a novel approach for image retrieval using color and edge information. We take into account the HSI (Hue, Saturation and Intensity) color space instead of RGB space, which emphasizes more on visual perception. In our system colors in an image are clustered into a small number of representative colors. The color feature descriptor consists of the representative colors and their percentages in the image. An improved cumulative color histogram distance measure is defined for this descriptor. And also, we have developed an efficient edge detection technique as an optional feature to our retrieval system in order to surmount the weakness of color feature. During the query processing, both the features (color, edge information) could be integrated for image retrieval as well as a standalone entity, by specifying it in a certain proportion. The content-based retrieval system is tested to be effective in terms of retrieval and scalability through experimental results and precision-recall analysis.

Key Words: Edge detection, Color histogram, K-Means Clustering Algorithm, Content-Based Image Retrieval (CBIR).

칼라와 에지 정보를 이용한 내용기반 영상 검색

박동원[†††] · 안성옥[††] · Ming Ma[†] · Kulwinder Singh[†]

요 약

본 논문에서는 칼라와 에지 정보를 이용한 내용기반 영상검색 기법을 제안하였다.기존의 RGB 공간 정보를 이용하기 보다는,시각적 인식에 보다 중점을 둔 HSI칼라 공간에서 고찰하였다. 비슷한 류의 색을 대표색으로 통합 표현하여, 개선된 칼라 정보 이용법을 본 연구에서 제안하였다. 또한 칼라 정보만을 이용했을 때의 시스템 성능상의 결점을 보완하기 위하여, 효율적인 에지 디텍션 기법을 함께 사용하였다. 칼라와 에지 기법을 통합함에 있어서, 각각의 기법에 적절한 가중치를 배분함으로써 시스템 성능을 실험적으로 향상시켰다.

1. Introduction

With tremendous growth in the field of the visual media, there is an increase and an urgent demand for the better accessibility of the visual media. Hence, many approaches have been suggested, in order to improve the query performance of the CBIR system. These approaches can be segmented into two categories [1]:

Manual: In this approach, a human expert may identify and annotate the essences of an image such as marks objects, regions, shapes, foreground and background, and so on for storage and retrieval.

Computational: Automatic extraction of visual information from an image, like color, shape and texture. There are many disadvantages, which exist in manual approach, such as extremely time-consuming, subjective view of the annotator is reflected, not scalable to large archives and inconsistency problem. Thus, automatic content-based approaches have received more attention in recent years.

In our system, color and edge feature has been taken into account. Color feature is widely used in CBIR systems, as it is the most important element that enables human to recognize images and thus it is the fundamental characteristic of the image features.

Many objects in an image have well defined geometric outlines or contours, which can often be helpful for human perception. Moreover, the contours in an image correspond to edges or boundaries of regions, so if a retrieval process is to be performed on the contours, an efficient method is required to obtain the contour's shape. We attained this objective using edge detection process.

The remainder of the paper is organized as follows. Section 2 focuses on the common methods used in feature extraction. Section 3 presents the content-based retrieval system that we developed is presented. Section 4 illustrates the experimental results of feature extraction techniques followed by conclusion in Section 5.

2. Feature Extraction

2.1 Color Feature

In image retrieval, the color feature has been one of the most widely used visual features in virtue of that it is relatively robust to background complication and independent of image size and orientation. Some representative studies of color perception and color spaces can be found in [2, 3].

There are several color feature

†비회 원: 배재대학교 IT공학부 박사과정
††종신 회 원: 배재대학교 IT공학부 교수
†††비회원: 배재대학교 IT공학부 부교수(교신저자)

representations that have been developed and applied in image retrieval such as color histogram, color moments and color sets.

Among them the color histogram which statistically denotes the joint probability of the intensities of the three color channels is the most commonly used color feature representation. As the similarity measure for the color histogram, Swain[4], Ioka[5] and Niblack et al.[6] proposed L1 metric and L2 metric respectively. Furthermore, considering that most color histograms are very sparse and thus sensitive to noise, Stricker and Orengo[7] introduced the cumulated color histogram that demonstrated the advantages of the proposed approach over the conventional color histogram approach.

The color moments approach proposed by Stricker and Orengo[7] has been used to overcome the quantization effects, as in the color histogram. The mathematical foundation of this approach is that any color distribution can be characterized by its moments. Furthermore, since most of the information is concentrated on the low-order moments, only the first moment (mean), and the second and third central moments (variance and skewness) were extracted as the color feature representation. Weighted Euclidean distance was used to calculate the color similarity.

To facilitate fast search over large-scale image collections, Smith and Chang[8, 9]proposed color sets as an approximation to the color histogram. They first transformed the (R, G, B) color space into a perceptually uniform space, such as HSV, and then quantized the transformed color space into M bins. A color set is defined as a selection of colors from the quantized color space. Because color set feature vectors were binary, a binary search tree was constructed to allow a fast search.

2.2 Edge Feature

To obtain the edge feature, the edge detection methods which are fundamental to computer vision are always used. Basically, the idea underlying most edge detection methods is the computation of a local derivative operator. We simply define a profile perpendicular to the edge direction at any desired point and interpret the results as in the preceding discussion. The first derivative at any point in an image is obtained by using the magnitude of the gradient at that point. The second derivative is similarly obtained by using the Laplacian.

2.2.1 Gradient operators (First order derivative)

From Equation 2.1, the gradient of an image f(x,y) at location (x,y) is the vector

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \qquad (2.1)$$

It is well known from vector analysis that the gradient vector points in the direction of maximum rate of change of f at (x, y). In edge detection an important quantity is the magnitude of this vector, generally referred to simply as the gradient and denoted ∇f, where:

$$\nabla f = mag(\nabla f) - [G_x^2 + G_y^2]^{1/2}. \qquad (2.2)$$

This quantity equals the maximum rate of increase of f(x,y) per unit distance in the direction of ∇f. Common practice is to approximate the gradient with absolute values:

$$\nabla f \approx |G_x| + |G_y| \qquad (2.3)$$

Which is much simpler to implement, particularly with dedicated hardware.

The direction of the gradient vector also is an important quantity. Let $\alpha(x,y)$ represent the direction angle of the vector ∇f at (x,y). Then, from vector analysis,

$$\alpha(x,y) = \tan^{-1}\left(\frac{G_y}{G_x}\right) \qquad (2.4)$$

where the angle is measured with respect to the x axis.

Several commonly used Gradient operators such as the Sobel, and Prewitt, Roberts operators are based in some way on measuring the intensity gradient at a point in the image. It is common to express these operators as two masks, one for computing partial derivative along rows and the second for computing partial derivative along columns.

2.2.2 Laplacian (Second order derivative)

The Laplacian of a 2-D function f(x, y) is a second-order derivative defined as

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \qquad (2.5)$$

As in the case of the gradient, Equation 2.5 may be implemented in digital form in various ways. For a 3×3 region, the form most frequently encountered in practice is

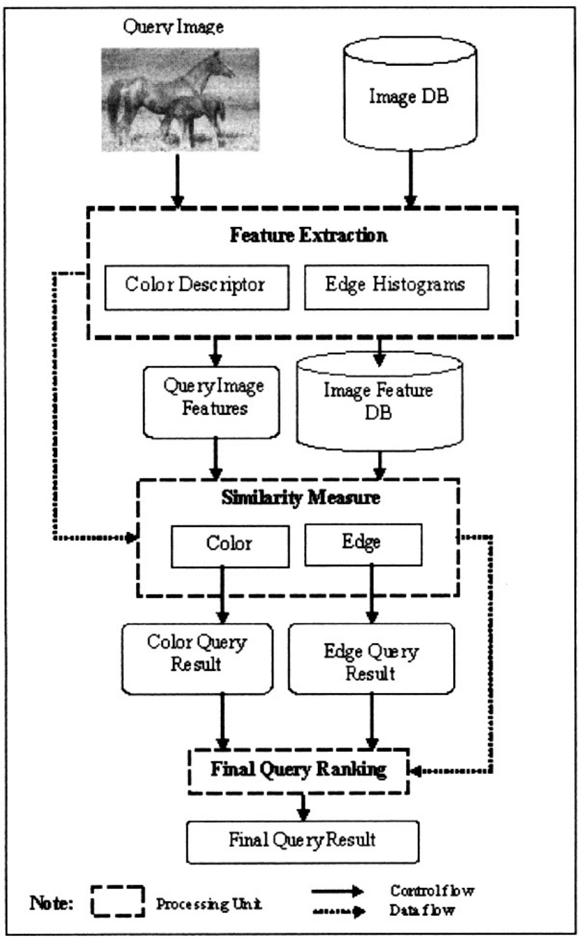
$$\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8) \qquad (2.6)$$

Where the z's have been defined already. The basic requirement in defining the digital Laplacian is that the coefficient associated with the center pixel is positive and the coefficients associated with the outer pixels are negative. Because the Laplacian is a derivative, the sum of the coefficients has to be zero. Hence the response is zero whenever the point in question and its neighbors have the same value.

3. Proposed Measures

3.1 Overview of Our CBIR System

Our CBIR system offers retrieval by any combination of color and edge features and image query is specifying an example query image. As it is illustrated in Figure 3.1, the system extracts and stores color and edge features from each image added to the database. At search time, the system matches appropriate features from query and stored images, calculates a similarity score between the query and each stored image examined, and displays the most similar images on the screen as thumbnails.



3.2 Color Descriptor

The color feature extraction starts first with color model transformation. Each pixel in an image has a three-dimensional color vector and different color space approaches exist to represent color information. One of these color space is the hardware-oriented Red-Green-Blue Model (RGB), where the color vector of a pixel p is the compound of red, green and blue channels $v_p = (r, g, b)$. Another color space model is the Hue-Saturation-Intensity Model (HSI) that is based on color descriptions rather than individual color components $v_p = (h, s, i)$. The RGB model has a major drawback: it is not perceptually uniform. Therefore, most of the systems use color space models other than RGB, such as HSI. Color clustering is performed on each image to obtain its representative colors.

After clustering, only a small number of colors remain and the percentages of these colors are calculated. Each representative color and its corresponding percentage form a pair of attributes that describe the color characteristics in an image. The dominant color descriptor is defined to be

$$F = \{\{c_i, p_i\}, i = 1, \dots, N\} \tag{3.1}$$

Where N is the total number of color clusters in the image region, c_i is a color vector, p_i is its percentage, and $\sum_i p_i = 1$.

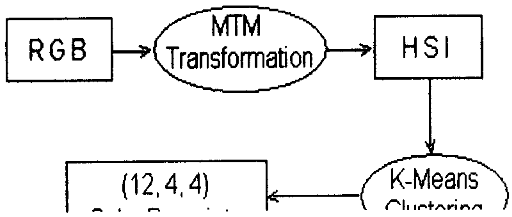


Figure 3.2 Procedure diagram for color clustering

The procedure for color clustering is shown in Figure 3.2. Firstly, we convert the RGB color model in to HIS model using the MTM transformer formulas.

Secondly, an improved K-Means clustering algorithm proposed by Alsabti et al.[10]is used on these HSI values to get representative colors and their percentages in the image [11].

The efficient K-Means algorithm is proved able to enhance the computational speed of the direct K-Means algorithm by an order to two orders of magnitude in the total number of distance calculations and the overall time of computation. This has been shown to be effective in producing good clustering results for many practical applications. The efficient K-Means algorithm consists of the following steps:

1. A k-d tree is generated to organize the pattern vectors, by which one can find all the patterns that are closest to a given prototype efficiently. There are some information kept by each node of the tree:
   a. The number of points *(m)*
   b. The linear sum of the points $(\overline{LS})$, i.e. $\sum_{i=1}^{m} \overline{p}_i$
   c. The square sum of the points *(SS)*, i.e. $\sum_{i=1}^{m} \overline{p}_i^{\,2}$
2. The initial prototypes are derived as in **the direct K-Means algorithm.**
3. A number of iterations are performed as in the direct K-Means algorithm until the termination condition is met. And the number of points C^i_n, the linear sum of the points $\overline{C^i_{LS}}$ and the square sum of the point C^i_{SS} are maintained for each cluster *i*.

During each iteration, we traverse the k-d tree from the root node with all *k* candidate prototypes. A pruning function is in turn applied on the candidates prototypes. The traversal stopped when the number of candidate prototypes is equal to one. The cluster statistics are updated based on the information about he number of points, linear sum and square sum stored for the internal node. A direct K-Means algorithm is applied on the leaf node if there is more than one candidate prototype.

In addition, because the human visual system is more sensitive to hues than to saturation or intensity, the H axis is quantized more finely than the *S* axis and the *I* axis. In our experiments, we cluster the HSI color space into 12 bins for hue, 4 bins for saturation, and 4 bins for intensity.

3.3 Edge Histogram

As mentioned above, the image was transformed to HSI color space for generating the color histograms, and the value of intensity which corresponds to the gray-scale representation of the image, was processed for edge detection. Thus the intensity channel is convolved separately with the two Sobel gradient operators to produce separate measurements of the gradient component in each orientation (call these *Gx* and *Gy*). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The orientations are in turn grouped into 8 directions ($0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°$). Finally the edge histograms are calculated by summing up the number in each direction, i.e.

$$H(i) = \left| \sum \alpha(x, y) \middle| \alpha(x, y) = i \right|, i = 1,2 \ldots 8 \tag{3.2}$$

where $\alpha(x, y)$ is the direction value of a pixel in the image.

The Histogram Intersection method (denotes in Equation 3.3, 3.4) is employed for similarity calculations as a result of texture vector and edge histogram comparisons between database images and query image.

intersection

$$(h(I), h(M)) = \sum_{j=1}^{K} \min\{ h(I)[j], h(M)[\tag{3.3}$$

match

$$(h(I), h(M)) = \frac{\sum_{j=1}^{K} \min\{ h(I)[j], h(M)[j]\}}{\sum_{j=1}^{K} h(M)[j]} \tag{3.4}$$

4. Performance Experiments

The performance of our content-based image retrieval system was tested on the database having 1000 images which used in SIMPLIcity [12].

4.1 Evaluating Effectiveness

In order to evaluate effectiveness of our retrieval systems, two well-known metrics, *precision* and *recall* [13], are used. Since different queries may lead to different precision and recall values, the computation of average effectiveness is needed. To ease this computation, the interpolation procedure is performed in order to facilitate the computation of average of precision and recall values [13], which is showed below:

- Individual precision values are interpolated to a set of 11 standard recall levels: 0%, 10%, 20%, ..., 100%.
- Let $j \in \{0,1,2,\ldots,10\}$ be a reference to the j-th standard recall level. Then,

$$P(r_j) = \max_{r_j \le r \le r_{j+1}} P(r) \tag{4.1}$$

- So, the interpolated precision at the

j-th standard recall level is the maximum known precision at any recall level between the j-th recall level and the (j+1)-th recall level.

In this section, the interpolated precision-recall graphs for an image are given (Figure 4.1). The experiment is to show the different effectiveness of the system by using three features separately and the combination of them. There is no specific reason for presenting the query images, but they provide a comprehensive way to evaluate the retrieval effectiveness of the content-based image retrieval system. To conclude that the system is effective, the basic expectation from the interpolated precision-recall graphs is the fact that they have to be non-increasing. In all of these graphs, the precision value is 1 for the first a few standards recall levels, and while standard recall levels are increasing, the precision values continue in a non-increasing manner.

4.2 Query Examples

In Figure 4.1~4.3 shows the result sets for three example queries using color edge and an integration of them.

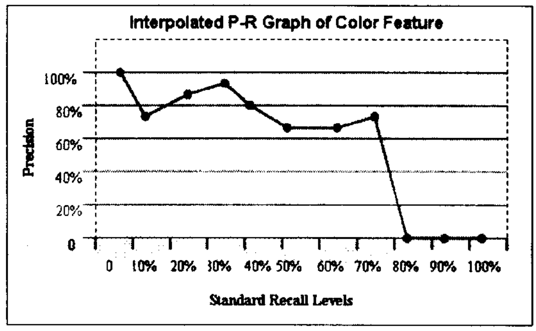


Figure 4.1 Interpolated Precision Recall Graph of Searching Result by Using Color (Image 435.jpg)

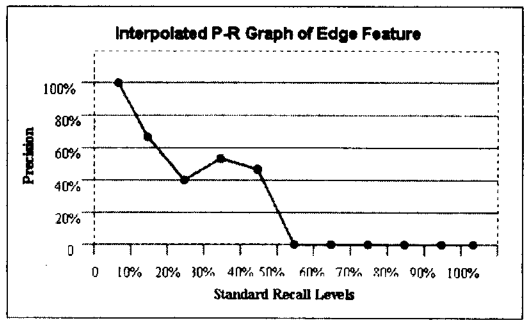


Figure 4.2 Interpolated Precision Recall Graph of Searching Result by Using Edge (Image 435.jpg)

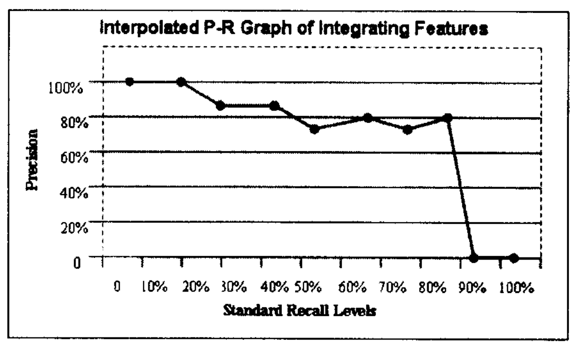


Figure 4.3 Interpolated Precision Recall Graph of Searching Result by Using Integration Feature (Image 435.jpg).

5. Conclusions and Future Work

We have designed and implemented a content-based image retrieval system that evaluates the similarity of each image in its data store to a query image in terms of color and edge characteristics, and returns the images within a desired range of similarity.

For the color content extraction, the color descriptor is used. The expressiveness of this technique is accelerated via color space transformation and quantization, and the color features are extracted by the help of an improved K-Means Clustering algorithm, an effective method for determining the optimal number of classes. For the edge feature extraction, a well-known and powerful technique, histograms, are used. The histogram intersection method has been used as the similarity measure between two feature vectors.

Our system has been tested on images used in SIMPLIcity and shown to be an efficient tool for image retrieval. Based on the experimental results, the retrieval based on the color feature is relatively robust to background complication and independent of image size and orientation compare to edge features. But edge feature works as the accessorial component of color feature by virtue of the capability of spatial discrimination. Finally, the retrieval by integrating these features with a proper proportion shows the best performance.

A crucial future work to be done on our system is to further enhance its capability of querying with respect to the certain object regions in images, in which the better results can be reached by using texture feature. Thus, the semiautomatic and fully automatic feature extraction processes will be included. For the former, the user is provided a drawing facility that is used for the specification of regions of interest. For the latter, the system tries to capture a rectangular region that represents the interested content of the image.

References

[1] A. Celentano and V. Di Lecce, A FFT based technique for image signature generation.

[2] C. S. McCamy, H. Marcus, and J. G. Davidson, A color-rendition chart, Journal of Applied Photographic Engineering 2(3), 1976.

[3] M. Miyahara, Mathematical transform of (r,g,b) color data to munsell (h,s,v) color data, SPIE Visual Commun. Image Process. 1001, 1988.

[4] M. Swain and D. Ballard, Color indexing, International Journal of Computer Vision 7(1), 1991.

[5] M. Ioka, A Method of Defining the Similarity of Images on the Basis of Color Information, Technical Report RT-0030, IBM Research, Tokyo

Research Laboratory, Nov. 1989.

[6] W. Niblack, R. Barber, and et al., The QBIC project: Querying images by content using color, texture and shape, in Proc. SPIE Storage and Retrieval for Image and Video Databases, Feb. 1994.

[7] M. Stricker and M. Orengo, Similarity of color images, in Proc. SPIE Storage and Retrieval for Image and Video Databases, 1995.

[8] J. R. Smith and S.-F. Chang, Single color extraction and image query, in Proc. IEEE Int. Conf. on Image Proc., 1995.

[9] J. R. Smith and S.-F. Chang, Tools and techniques for color image retrieval, in IS & T/SPIE Proceedings, Vol. 2670, Storage & Retrieval for Image and Video Databases IV, 1995.

[10] K.Singh, M. Ma, D-W. Park, A Content-Based Image Retrieval Using FFT & Cosine Similarity Coefficient, Proc. of. The 5th IASTED International Conference Signal and Image processing, (SIP 2003), Honolulu, Hawaii, U.S.A. (13-15 August,2003).

[11] M. Ma, K.Singh, D-W. Park, Image Retrieval By Contents / Using Color & Edge, Proc. of. The 3rd International Workshop on Content-Based Multimedia Indexing Sep.22-24, 2003 IRISA, Rennes, France.

[12] J. R. Smith and S.-F. Chang, Single color extraction and image query, in Proc. IEEE Int. Conf. on Image Proc., 1995.

[13] K.S. Jones. Information Retrieval Experiment. Butterworth and Co., 1981.

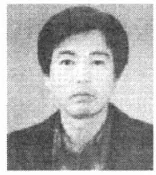
안 성 옥

1983년 2월 고려대학교수학교육과졸업(이학사).
1985년 2월 고려대학교 대학원 수학과 졸업
(전산학전공 석사).
1989년 8월 고려대학대학원 수학과 졸업
(전산학전공 박사).
1991년~현재 배재대학교 IT공학부 교수.
1993년~1994년 펜실베이니아 주립대학교
컴퓨터학과 방문교수.
1994년~1995년 뉴욕주립대학교
컴퓨터학과 방문교수
관심분야:데이터베이스,컴퓨터교육, 영상처리
게임공학
E-Mail: sungohk@pcu.ac.kr

박 동 원

1983 고려대학교 전기공학과
(공학사)
1985 Florida Institute of
Technology 공학석사
(전산학 전공)
1993 Texas A&M University
공학박사 (전산학 전공)
1994 ~ 현재 배재대학교 IT공학부 부교수

관심분야 : 실시간통신, 영상처리
E-Mail : dwpark@pcu.ac.kr



Ming Ma

1997 Dept. of Computer
Science and Information
Engineering, Inner
Mongolia Agricultural
University, China B.E
2002 Dept. of Information and
Communications Engineering Paichai
University, Korea M.S
2004 ~ Dept. of Information and
Communications Engineering Paichai
University, Korea
Research Interests:
Multimedia information retrieval,
Image processing and computer vision.
E-Mail : ming@pcu.ac.kr

Kulwinder Singh

1996 Dept. of Electronic
Engineering,
Ravishanker Shukla
University, India B.E
2002 Dept. of Information and
Communications Engineering Paichai
University, Korea M.S
2004 ~ Dept. of Information and
Communications Engineering Paichai
University, Korea
Research Interests:
Multimedia information retrieval,
Biometric Authentication.
E-Mail : singh@pcu.ac.kr