IMAGE RETRIEVAL SYSTEM IN DATABASES:
THE PROBLEM OF COLOR SIMILARITY

Aura CONCI and Everest Mathias M. M. de CASTRO
Federal Fluminense University - UFF
Niterói, Rio de Janeiro, BRAZIL

ABSTRACT

Storage of information in database is meaningful only when it can be retrieved. Nowadays databases have become too large to be handled manually. Automatic browsing on conventional databases uses alphanumeric match for retrieval. Digital images are becoming an integral part of human communication. We increasingly use visual content to express ideas, educate and entertain each other. Images demand large resources for transmission, storage and processing. Automatic browsing on image databases must be use for retrieval what the human being see in a image file. But a figure is a rich and subjective source of information. Different people extract different meaning from the same picture, and it also varies over time. These factors make the indexing, retrieval and management of visual information a great challenge. However, several systems have been designed for image retrieval using their contents like image color properties. We will discuss here how these systems work and what is the difficulties related on comparing them. We describe a prototype designed for combining interactively different color features in a common implementation. The purposes are to show the influence of these features on the performance of recovering by similarity based on color histograms and to give insight into theirs merits. Key words: Content-based indexing, content-based retrieval, image content analysis.

1. INTRODUCTION

As multimedia becomes commonplace, digital images are becoming an integral part of human communication. The ease of creating or capturing images has enabled its proliferation and influence to change many concepts on database organization. Image repositories are generated at increasing rate by a great number of applications, soon they become too large to be handled manually. Automatic browsing on conventional databases uses alphanumeric index for retrieval. Query in image databases is based on similarity assessment rather than on exact match (Santini and Jain, 1996). The items returned have similarity with the query input, but the returned items rarely have an exact match. Moreover, strings or alphanumeric indices are not closely related to human similarity judgment for pictures. Techniques of feature extraction allow queries using an intuitive manner. They are the essential part of the Content-Based Image Retrieval (CBIR) systems. The primary problem to solve is to devise image contents (color, texture, shape, etc.) capturing their essentials (Ogle and Stonebraker, 1995). A survey of classes of visual information collected on WWW shows that 85% consists of color images (Chang *et al.*, 1997). As a case study in this work we describe Content-Based Image Retrieval (CBIR) systems in terms of the image color similarity. We will discuss how these systems work and evaluate the influence of color spaces and distance measures on the performances of recovering by similarity based on color histograms.

2. FEATURE REPRESENTATION

Some systems, like QBIC (Niblack *et al.*, 1993), Virage (Bach *et al.*, 1996), Photobook (Pentland *et. al.*, 1996) and C-BIRD (Zaiane *et al.*, 1997) provide methods for retrieving digital images by using examples or visual sketches. In many approaches, feature vector is used to characterize each image in the database and to respond a user query of the form: “find more pictures like this one". The idea of exact match (on retrieval) is abandoned and the efforts are driven to meet an adequate value to represent the proximity among images. Distance between two feature vectors can represent the similarity among two images and it is then used to retrieve images. To represent color images, we use theirs color histograms as feature vectors. The assessment of similarity between images is performed via the storage of the color histogram for each image (fig 1).

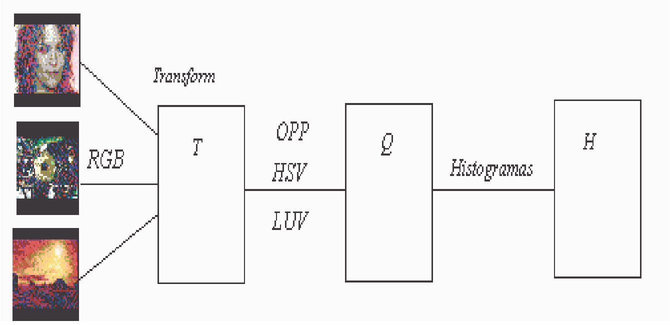


Fig. 1 - Color histograms are used to represent each image.

Color quantization transforms a continuous tone picture into a discrete image that can be stored as a data file. Two types of image file format are used on the internet: JPEG and GIF. JPEG uses lossy compression algorithms and a 24-bit RGB color representation (transformed to YIQ color space for internal codification). GIF uses LZW lossless compression algorithms (highest image quality at the expense of a low compression gain) and color maps, typically represented as 256 (for an 8-bit color map) or less colors. Both these processes can introduce distortion. In this work we use GIF files, where the visible distortion is associate only with the color used in the digitized image to represent the color in the real scene. The question in this case is how to choose the better colors to reproduce the original. A quantization algorithm should distribute any visible distortion throughout the image in order to minimize its perception by an human observer. Through the quantization the dimension of the 3D continuous color space is reduced to the discrete color map of the image file. This process precedes the image storage as a GIF file. As each image has its own color map, the query system must use the original RBG component to evaluate the histograms used for computation of the similarity between images.

3. IMPLEMENTED SYSTEM

To allow the user to compare images from different source and see the influence of color spaces and distance functions, in the implemented system, the histogram processing uses the following steps. First, each pixel of the image is originally represented as a point in the RGB-color space. That is, an image point [x,y], is specified by the values of the three channels (red, green and blue) that contain the color information:

$$(v_c) = (I_R[x,y], I_G[x,y], I_B[x,y])$$

Second, the RGB image is transformed to the selected color space by the user, T_c. Third the quantization is performed in the selected space assuming uniform number n_1, n_2, n_3 of elements per axis, Q_c. Then, each image points is analyzed to compute the normalized histogram with $m = n_1 \times n_2 \times n_3$ cells:

$$h[i] = \frac{1}{XY}\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1} f(x,y) = \begin{cases} 1 \, if\,(x,y)\,color = i \\ 0 \, if\,(x,y)\,color \neq i \end{cases}$$

$$f(x,y) = Q_c^m T_c(v_c)$$

where X and Y are the number of pixels in each direction and $i=1,2,3,\ldots m$. So color histograms are points in an m-dimensional space related "one-to-one" to the each analyzed image (see figures 2 and 3).

Human color perception is universally accepted as a 3D phenomenon but the better color space to represent this is an open point. Examining CBIR systems we see that each group works with a different space. The Opponent color space-OPP (Swain and Ballard, 1991), the Hue-Saturation-Value space-HSV (Kaiser and Boyton, 1996) and the CIE-LUV (Xiang, 1997) are the most frequently used. These three spaces (in the implemented system) can be selected by the user to compute T_c , in the last equation. Uniform quantization was used for all color spaces. Two quantization types were used for the space HSV and one quantization form for each of the other. In OPP the histograms are composed by 256 elements: the axis *wb-white/black* was partitioned in 4 cells while the axes *rg-red/green* and *by-blue/yellow* were partitioned in 8 cells each. In CIE-LUV space 4-8-8 partitions were used in L, U and V axis respectively, resulting in histograms with 256 cells. In HSV, the attribute hue (H) is associated with the dominant wavelength, it represents the more significant characteristic of the color, its values range from 0^o to 360^o and it was quantified in two forms: (1) 18 sections of 20^o each, and (2) 24 sections of 15^o each. That is the Q_c component in last equation, for HSV space, has m equals to 162 (18x3x3) or 216 (24x3x3) cells.

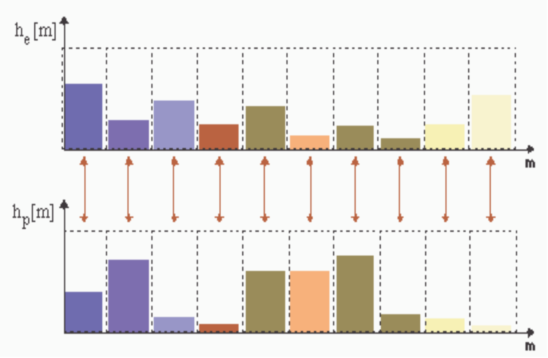


Fig. 2- Model used for d_1, d_2 and histogram intersection.

Given a sample, the distance between its histogram and each histogram of the database are computed. Images with distance less then a predefined threshold or a predefined number of nearest images are retrieved (the latter is easier to user because threshold values frequently depends on theoretic aspects). Five distance functions are used to evaluated histograms similarity. They are: (1) the d_1 metric; (2) d_2 or Euclidean metric; (3) the histogram intersection; (4) the average color distance, and (5) the quadratic distance measure. Denoting h_e[m] the histogram of the example and h_p[m] the histogram of each image present on the database (Fig. 2), then the d_1 and the Euclidean metrics (or d_2) are given by (Berman and Shapiro, 1997):

$$d_{e,p}^{\tau} = \left[\sum_{m=0}^{M-1} \left| h_e[m] - h_p[m] \right|^{\tau}\right]^{\frac{1}{\tau}}$$

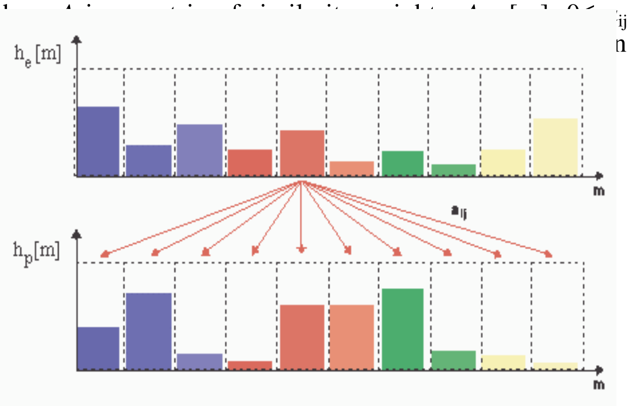
where if $\tau = 1$, it represents d_1 metric and if $\tau = 2$ it corresponds to Euclidian metric. If the images has the same number of pixels, XxY then the histogram intersection distance function, d_3, is given by (Swain and Ballard, 1991):

$$d_{e,p} = 1 - \frac{\sum_{m=0}^{M-1} \min(h_e[m], h_p[m])}{\{XY\}}$$

The d_4 average colors distance (Hafner *et al.*, 1997) uses the average magnitude along the three channel of the used space color to characterize the average color of each image. Then the distance between two images are computed by the Euclidean distance between their average colors. Finally, the quadratic distance measure, d_5, is calculated by (Deng and Manjunath, 1997):

$$(d_{e,p})^2 = (h_e - h_p)^T A(h_e - h_p)$$

w ij
≤ n



distance between colors i and j in the used color space (here HSV, OPP or LUV) and d_{max} is the greater distance between colors on the same space. This distance function is the only one that consider the similarity between color elements (figure 3) of the histogram (Ma *et al.*, 1997).

Fig. 3 - Considering similarity between histogram cells.

The implemented system is designed to permit combination of these (distance functions and color space) to evaluate the query results when an image is used as sample. It collects these options and through the routine of communications (using HTTP-Hypertext Transfer Protocol) transfers they to the server. On the server, programs writing in C, Java and CGI (Common Gateway Interface) receive the customer's information, accomplish searches in image database, and return the results to the customer, in the form of selected images (figure 4 shows the user interface). It can be used on the site: http://www.ic.uff.br/~mathias/visual.htm.

Fig. 4 - A retrieval example with the user interface

4. EVALUATING QUERY RESULTS

Figure 5 shows partial retrieval results using different combination of the parameters color and distance function. The query image is the left image out of line, followed by the retrieval result from left to right according to its distance to the query image. In this example same database are used but the results change with the change of color space and distance function. That is, the result of a query depends on the selected parameters. In the literature there is no method or benchmarks to compare approaches. To evaluate the influence of each feature, the ideal measure is the human evaluation. By a visual examination, the classification of the images depends on the observer's subjectivity to characterize them. That is to say, it needs to be accompanied by an exhaustive visual verification of the recovered images. For this purpose a simulator engine was implemented with the objective of computing the grade of adequate recovery of the images. Figure 6 shows how it is designed. It permits evaluation of the influence of each aspect determining the relative merits of each combination.

The projected simulator consists basically of a database with groups of similar images and a search simulation engine (Castro and Conci, 1998). It contains images with all the chromatic nuances. Groups of similar images were identified visually by 3 different users. When an image is recovered, such identification can be used to compute evaluation parameters. Whenever an image of the same group is present in a query, it is considered that a significant recovery has been made. Two features are proposed to measure how well the results satisfy the query (Conci and Castro, 1999). The first feature, effectiveness, evaluates the amount of false negative that a group of color/distance produces in a recovery. It tries to evaluate the possibility of the system to show all the significant images. Other feature, efficiency, evaluates the false positives: when less similar images present closer distance than other that are real similar. The graphs on figure 7 show how predominant color of the query affects the effectiveness. In this figure group 2 corresponds to images with green/yellow nuances and group 4 are composed by gray/blue nuances. More graphs considering the 20 possible combinations of metrics and spaces. can be seen on http://ic.uff.br/~aconci/colors/ efficiency.htm or ic.uff.br/~aconci/colors/effectiveness.htm.

As results, show evaluation through computer of the visual perception of similar colored images is a complex process. Besides this difficulty some simplifications have to be adopted in the global process. The perception of a color is supposed not to be influenced by the other colors that compose the image. Conditions as illumination, observer's visual adaptation, distance of the observer to the image, and quality of the video can not be considered. When a search of images is requested, a selection process will recover the closest images in the database. The central focus is to accomplish a search on the database for proximity. That is to say, to find those images that, inside a