Histogram Families for Color-Based Retrieval in Image Databases

Carlo Colombo, Alessandro Rizzi and Ivan Genovesi

Dipartimento di Elettronica per l'Automazione
Università di Brescia
Via Branze 38, I-25123 Brescia, Italy

Abstract. A system for image representation and retrieval in a pictorial database using color distribution features is presented. Images are internally described and matched one against the other by means of a set of color histograms taking into account the local characteristics of chromatic image structure. A graphic environment allows the user to compose interactively pictorial queries by both color sketch and image examples. It is also possible to the user to exploit the history of previous queries to affect current system output. Experimental evidence relating system performance to human expectation is presented and discussed.

1 Introduction

Recent advances in information technology have made it possible to process, store and share huge amounts of heterogeneous digital data. The efficient creation and management of large pictorial archives is a key step towards the costruction of modern digital libraries [1, 2].

The complex structure of images requires that novel techniques be devised in order to both represent and recover information in a pictorial database. Relying on a high level (semantic) data representation, manual annotation and query by keywords as simple extensions of traditional alphanumeric database technology have a limited portability to the image domain [3]. In the last few years, several approaches based on the automatic extraction of lower level (syntactic) representations from images have been proposed. Image features commonly used to describe and retrieve images from a database (query by content) are color/texture [4, 5, 6], shape [7] and 2D spatial relationships [8].

Graphic user interfaces (GUI) are the most natural way of composing a pictorial query to an image database management system. GUIs allow to specify the query directly in pictorial form (query by example), and provide the user with a visual feedback of the system response to the query. Differently from traditional databases, the system output for image databases is usually not a partitioning of the archive into two classes (elements satisfying or not satisfying the query), but a reordering of the archive itself according to a measure of similarity of each image with the query image [9].

In this paper, we describe a system for pictorial content representation and retrieval based on color distribution features [10]. The distribution of chromatic

content in an image is described through a collection of color histograms, referred to as "histogram family." Each histogram in a family is related to a specific image region obtained by a clustering technique. A matching strategy between histogram families is proposed, allowing to define a metric of similarity between images.

A graphic interface allows to compose image-like queries to the system both through user-made color sketches and portions of example images. The interactive nature of the retrieval process is emphasized by providing the system with a memory of past queries and outputs. The results of both a qualitative and quantitative comparison between system output and human expectation are presented and discussed.

2 Organizing Pictorial Content

This section describes the system aspects related to internal database organization and image search (retrieval subsystem).

2.1 Color Histogram Families

The color content of an image $\mathcal{I}$ can be characterized by the color histogram $h(\mathcal{I}; N, P)$ giving the frequency of occurrence – normalized w.r.t. the overall image pixel number P – of each of the N colors quantizing the image color space. Beside being effective for characterizing the *global* color properties of an image, the color histogram representation is also useful to define a measure of similarity between two images. The histogram intersection operator introduced in [4] provides a simple way to match two different images $\mathcal{I}$ and $\mathcal{I}'$ through their color histograms h and h':

$$H(\mathcal{I}, \mathcal{I}') = \sum_{i=1}^{N} \min(h_i, h'_i) \quad , \tag{1}$$

where h_i (h'_i) denotes the normalized frequency of the i-th color in histogram h (h').

A natural way of extending the use of global image histograms to characterize the *local* color properties of an image is to use a set of color histograms $\mathrm{F}(\mathcal{I}; M_{\mathcal{I}}) = \{h(\mathcal{R}_k; N, P_k),\ k = 1, \ldots M_{\mathcal{I}}\}$ each reflecting homogeneous color distribution properties inside an image region $\mathcal{R}_k$. Such a histogram set is referred to as *histogram family*. A histogram family is obtained by dividing the image into small square non-overlapping pixel tiles, and clustering them with a "split and merge" technique [11] using $H(\mathcal{R}, \mathcal{R}')$ in eq. (1) as a homogeneity predicate.

Using the concept of histogram family it is possible to take into account the color (and also the shape and spatial) information of image regions with well defined color distributions: such distributions are usually much different from the overall image color distribution and are possibly related to different objects contained in the image.

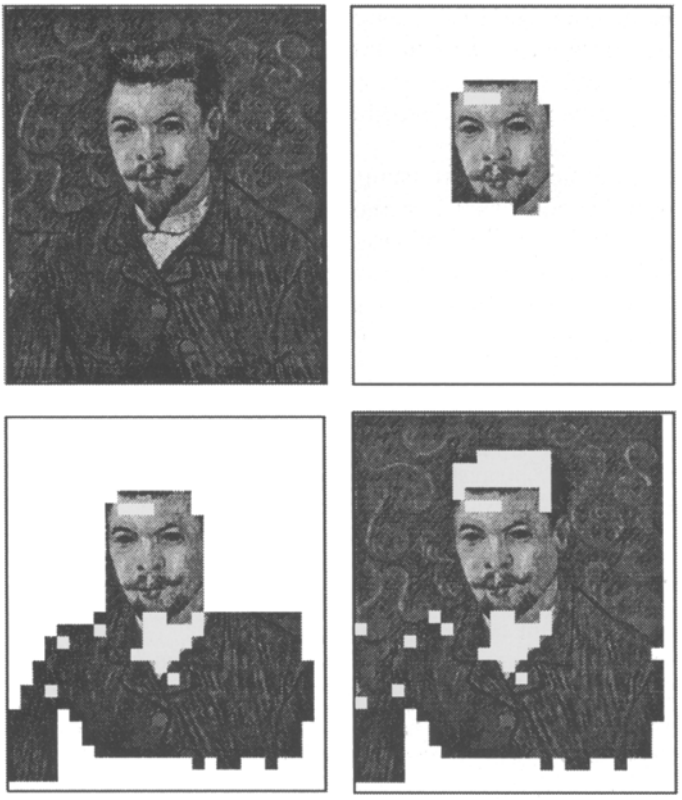

Fig. 1. An image and its three clustered regions: face, jacket and background wall.

Fig. 1 shows the homogeneous regions produced by clustering two different images using the method described with a tile size of 8×8 pixels. Only image regions with an area exceeding a predefined threshold of 4% of the overall image area are considered.

2.2 Image Matching

To evaluate the similarity between a query image $\mathcal{I}'$ and a database image $\mathcal{I}$ through their corresponding histogram families $\mathrm{F}(\mathcal{I}'; M_{\mathcal{I}'})$ and $\mathrm{F}(\mathcal{I}; M_{\mathcal{I}})$ a three-step algorithm extending the criterion of eq. (1) has been devised.

1. Chromatic Matching. A correspondence is established between histogram pairs of the query and database families. Since the two families have in general a different number of elements, a *coupling function* $\gamma : \mathcal{I} \longrightarrow \mathcal{I}'$ is used to connect regions $\mathcal{R}_k$ and $\mathcal{R}'_j$ in the two images maximizing the histogram intersection functional $H(\mathcal{R}_i, \gamma(\mathcal{R}_i))$, $i = 1, \ldots \min(M_{\mathcal{I}}, M_{\mathcal{I}'})$.

2. Geometric Matching. A similarity measure related to the image area occupied by color-coupled regions is evaluated. Denoted with $\sigma(\mathcal{R})$ the area occupied by an image region $\mathcal{R}$, geometric similarity is defined as

$$A(\mathcal{R}_i, \gamma(\mathcal{R}_i)) = \frac{\min[\sigma(\mathcal{R}_i), \sigma(\gamma(\mathcal{R}_i))]}{\max[\sigma(\mathcal{R}_i), \sigma(\gamma(\mathcal{R}_i))]} \quad . \tag{2}$$

3. Similarity Score Evaluation. The overall similarity score between the query and database images is computed as the weighted average of the chromatic and geometric scores:

$$K(\mathcal{I}, \mathcal{I}') = \frac{1}{M_{\mathcal{I}'}} \sum_{i=1}^{\min(M_{\mathcal{I}}, M_{\mathcal{I}'})} w_H \cdot H(\mathcal{R}_i, \gamma(\mathcal{R}_i)) + w_A \cdot A(\mathcal{R}_i, \gamma(\mathcal{R}_i)) \quad , \tag{3}$$

with the normalization constraint $w_H + w_A = 1$, $w_H \in [0, 1]$.

It is crucial to system speed performance that the number of images matched against the query be lower than the overall number of images present in the database. This is achieved via indexing the database by sorting each of the image representation attributes (histogram color frequencies and region areas) and using an "interest window," centered on the current query representation, from which to select the images being compared with the query.

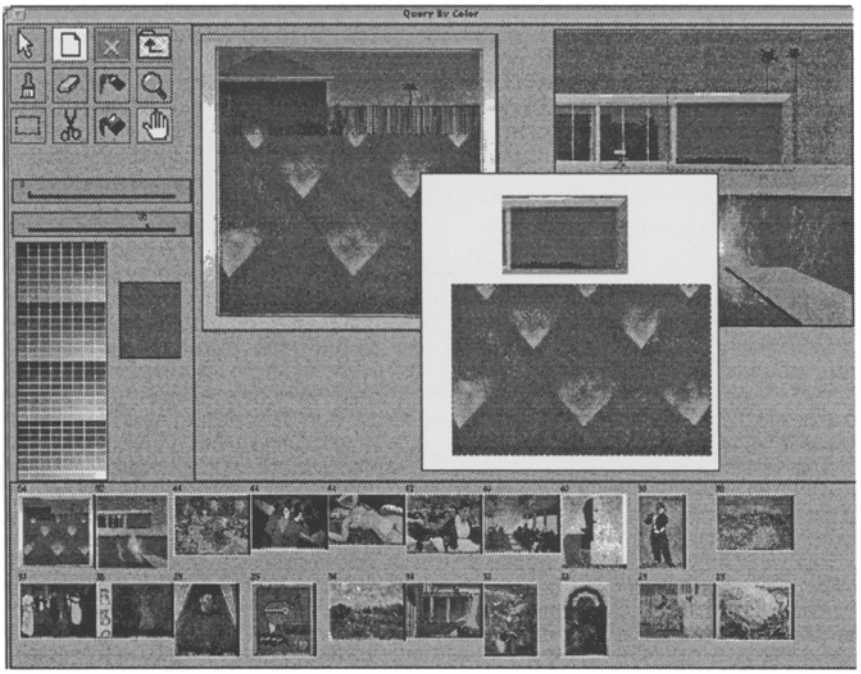

Fig. 2. Visual query composition and corresponding system output.

3 Interacting with the System

An interactive query subsystem provided with an XWindow interface has been implemented taking into account both user friendliness and expressivity GUI requirements.

3.1 System Interface

The system interface is divided into three main areas (Fig. 2). The *query composition area* (center and upper right) is used to compose image-like queries. Complex queries can be generated by using multiple windows and mixing user color sketches and portions of pre-existent database images. Clicking on a window makes its pictorial content be considered as a query to the system. The *composition tools area* (upper left) contains icons related to painting tools such as "brush," "spray" and "fill," and image handling tools such as "cut," "paste" and "move." A color palette for user sketch drawing is also made available (center left). System output resulting from a query is displayed in the form of image thumbnails in the *system output area* (bottom). Thumbnails are displayed in raster order according to their similarity w.r.t. the query: they provide the user with a visual feedback which can be helpful for subsequent queries.

Fig. 2 shows (center of the query composition area) a pictorial query obtained by combining portions of database images "lawn" (left) and "swimming pool" (right), previously imported in the query composition area. The system output sequence presents the original images in the first two positions: such a result would not have been obtained by a similarity search based on global histograms. The "lawn" image is classified before the "swimming pool" image, thanks to the larger region used for the query.

3.2 Query Memory

To further improve the interaction characteristics of the system, a mechanism for keeping track of previous user queries is introduced (state of the query). According to such a mechanism, system output at query time t is the result of a *system query* $q(t)$ obtained from the entire sequence $\{u(\tau),\ \tau = 0, 1, \ldots t-1, t\}$ of *user queries* as

$$q(t) = v \cdot u(t) + (1 - v) \cdot q(t-1) \quad , \tag{4}$$

where $v \in [0, 1]$ and $q(0) \doteq u(0)$. The case of "memoryless" system output, i.e. an output function of the current query only, is obtained for $v = 1$. The desired degree v of query memory can be selected directly from the GUI, together with the relative color/area weight w_H, by a scroll bar (Fig. 2, upper left). System query memory is well suited to iterative query refinement, since it allows the user to progressively adjust his target.

4 Tests and Results

Diverse tests have been performed with a database including over 200 digitized reproductions of Renaissance through modern art paintings. Paintings are among the most challenging kinds of pictorial data to process automatically, due to the large variability of styles and subjects to deal with.

To represent colors, the *HSL* color space [11] was preferred to the conventional *RGB* space, as it agrees betters with human chromatic perception. The overall number of colors considered was $N = N_H \cdot N_S \cdot N_L + 1$, i.e. the product of the quantization levels for hue, saturation and luminance, respectively, with an additional category for non colored pixels. Tests were performed using the two color quantizations 6–1–1 ($N = 7$) and 6–2–5 ($N = 61$): in each of these cases, hue is considered more important than saturation or luminance to color-based retrieval. To construct the database representation based on families, an area similarity threshold – controlling the number of families obtained for each database image – of 50% was chosen as a trade-off value between region representativity and significance.

In a first set of tests a quantitative comparison between system output and human expectation was carried out ($N = 61$). Several subjects were asked to select, from the database, images with significant chromatic analogies to images from a test set of 100. Defined T as the number of images selected by the system from the database as significant (according to a threshold on the similarity score), S as the number of images significant to human subjects, and R as the number of images significant to both the system and subjects, the performance figures *precision* R/T and *recall* R/S are introduced. Tab. 1 reports the results for both performance figures, giving the number of test images yielding a figure falling into one of the four categories 0.0–0.25, 0.25–0.5, 0.5–0.75 and 0.75–1.0.

PRECISION & RECALL	*Precision Categories (%)*				*Recall Categories (%)*			
Search Type	*0–25*	*25–50*	*50–75*	*75–100*	*0–25*	*25–50*	*50–75*	*75–100*
Global Histogram	51	29	20	0	49	30	12	9
Families, $w_H = 0.3$	46	54	0	0	44	33	23	0
Families, $w_H = 0.6$	30	40	29	1	23	7	48	22
Families, $w_H = 0.8$	16	29	28	27	18	23	35	24
Families, $w_H = 1.0$	18	38	19	25	11	42	23	24

Table 1. Quantitative results: precision and recall parameters (see text).

From the results it appears that histogram families perform always better than global histograms save for $w_H = 0.3$, in which case the system actually relies more on area than on color similarity. The best performance is achieved for $w_H = 0.8$, i.e. when both color and area are taken into account, and the former is much more important than the latter.

System performance was also assessed qualitatively by testing the system capability to recover database images from user color sketches ($w_H = 0.8$). Tab. 2 reports the frequency with which a desired image retrieved from user sketch was present in the system output, and its position in the output sequence. A position after the 20th is considered as a retrieval miss.

RETRIEVAL BY USER SKETCH	*Position in System Output*			
Search Type	*1-3*	*4-10*	*11-20*	*missed*
Global Histogram, $N = 61$	8	1	5	16
Global Histogram, $N = 7$	8	3	5	14
Histogram Families, $N = 61$	13	4	7	6
Histogram Families, $N = 7$	17	3	4	6

Table 2. Qualitative results: retrieval by user sketch (see text).

Also in this case histogram families are evidently superior to global histograms. Besides, while a finer color quantization usually provides better results using as query a visual example, in the case of queries based on color sketches the reverse is true (compare rows $N = 7$ with rows $N = 61$). This is easily explained by the fact that to the user it is a harder task to keep in mind and specify an exact color rather than an approximate hue value.

5 Conclusion and Future Work

We have presented and discussed an interactive system for the organization of and retrieval from image databases based on color distributions and simple geometric properties of regions clustered from the images. The system creates its internal query representation based on the current user query – composed by possibly mixing color sketches and visual examples – and on a memory of previous queries.

The system can be augmented and enhanced in several ways. For instance, a sharper sensitivity to geometric features such as 2D shape can be introduced by considering higher order moments of the extracted image regions. To improve further system memory and user adaptation, the concept of state of the query can also be extended to define a state of the database itself (active index).

Acknowledgement

The authors would like to thank Prof. Alberto Del Bimbo for his advice and support.

References

1. V.N. Gudivada and V.V. Raghavan, eds. Content-Based Image Retrieval Systems. *IEEE Computer* 28(9), September 1995. (Special Issue.)
2. R.W. Picard and A.P. Pentland, eds. Digital Libraries: Representation and Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18(8), August 1996. (Special Issue.)
3. V.E. Ogle and M. Stonebraker. CHABOT: Retrieval from a Relational Database of Images. *IEEE Computer* 28(9), 1995.
4. M. Swain and D. Ballard. Color Indexing. *International Journal of Computer Vision* 7(11), 1991.
5. K. Hirata and T. Kato. Query by Visual Example – Content-Based Image Retrieval. In Proc. *EDBT'92*, pages 56–71, Springer LNCS 1992.
6. W. Niblack *et al.* The QBIC Project: Querying Images by Content using Color, Texture and Shape. *Research Report 9203*, IBM Research Division, Almaden Research Center, 1993.
7. A. Del Bimbo and P. Pala. Visual Image Retrieval by Elastic Matching of User Sketches. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(2), 1997.
8. S.-K. Chang and E. Jungert. Pictorial Data Management based upon the Theory of Symbolic Projections. *Journal of Visual Languages and Computing* 2(2), 1991.
9. S. Santini and R. Jain. The Graphical Specification of Similarity Queries. *Journal of Visual Languages and Computing* 4(5), 1996.
10. I. Genovesi. *Ricerca Interattiva per Distribuzioni di Colore in Database di Immagini.* Master's Thesis, Polytechnic of Milan, April 1997. (In Italian.)
11. D. Ballard and C. Brown. *Computer Vision.* Prentice-Hall, Engelwood Cliffs NJ, 1982.