



Color-induced image representation and retrieval

Carlo Colombo, Alberto Del Bimbo*

Dipartimento di Sistemi e Informatica, Università di Firenze, Via S. Marta 3, Firenze I-50139, Italy

Abstract

In this paper, we describe a framework for pictorial content representation, query formulation and image retrieval based on color distributions. Image content is described through a set of color histograms, each relative to a region with a homogeneous color distribution. The representation also includes geometric features induced by the color distribution based segmentation process. A metric of similarity between images using the above representation and generalizing the histogram intersection operator is defined, allowing to perform retrieval by color distribution content. User query images are interactively created and modified through a graphic environment featuring color sketching, image examples and relevance feedback. The user is also provided with two novel query composition and refinement tools, based respectively on an internal query memory and on a direct manipulation of internal query representation. Examples of operation are provided, and experimental results are reported indicating that, thanks to the superiority of multiple distributions over global color histogramming, the quality of image retrieval is in good accordance with human expectation. © 1999 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Pictorial databases; Image representation and retrieval; Query by color distribution content; Color image analysis; Graphic user interfaces

1. Introduction

The efficient creation and management of large pictorial archives is a key step towards the construction of advanced digital libraries. Both the size and the non-structured nature of the information involved call for the definition of novel strategies to process, store and retrieve data in an image database [1–3].

Specifically, devising a suitable data representation – i.e., a model to extract from images, describe and manipulate only the world aspects salient to the system

– is crucial for an image database management system (IDBMS) to be really useful and effective. The most traditional way to generate a representation in an IDBMS is to allow manual annotation of pictorial information at storage time through a set of external keywords describing pictorial content [4]. Yet, relying on a high level (semantic) data representation, manual annotation and query by keywords as simple extensions of alphanumeric database technology have a limited portability to the image domain, for a number of reasons: (1) large databases cannot be annotated manually; (2) the quality of annotation is subjective, since it strictly depends on the skill and sensitivity of the human operator; (3) keywords typically do not allow search by similarity.

Actually, in the last few years, a huge research effort was devoted to the development of novel techniques for

*Corresponding author. Tel.: + 39-055-4796262; fax: + 39-055-4796363.

E-mail address: delbimbo@dsi.unifi.it (A. Del Bimbo)

an automatic, unsupervised representation of pictorial content. Several approaches based on the automatic extraction of lower level (syntactic) representations from images were proposed. Image features commonly used to describe and retrieve images from a database (query by content) are color/texture [5–7], shape [8] and spatial relationships [9]. Some systems also allow to formulate queries based on combinations of these features [10,11]. Graphic user interfaces are the most natural way of composing a pictorial query to a syntactic IDBMS, since they 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 [12].

In this paper, we describe a system for pictorial content representation and retrieval based on color distribution features. The distribution of chromatic content in an image is described through a set of color histograms, and complemented by geometric features induced by color distributions. Each histogram is related to a specific image region obtained by a color distribution based segmentation technique. An image matching strategy using histogram sets and color distribution-induced features is proposed, allowing to define a metric of similarity between image pairs.

A graphic interface allows to create 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 traditional query refinement tools (relevance feedback, i.e. generating new queries by using portions of previous outputs) and two novel mechanisms called, respectively, *internal query memory* – allowing to use past user queries for the composition of the current retrieval task – and *internal query manipulation* – allowing the user both to visually inspect and to modify the system representation of the current query image.

Examples of system operation are provided, and experimental results of both a qualitative and quantitative comparison between system output and human expectation are reported and discussed. Specifically, results witness the superiority of multiple distributions over global color histogramming as a way of representing pictorial content.

2. Image representation

2.1. Color clustering vs color segmentation

A widespread representation of the content of a color image \mathcal{I} is the *global 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 can also be used to define a metric of similarity between two images. The histogram intersection operator introduced in Ref. [5] provides a simple way to match two different images \mathcal{I} and \mathcal{I}' through their color histograms h and h' :

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

where h_i (h'_i) denotes the normalized frequency of the i th color in histogram h (h'). As shown in Fig. (1), the histogram intersection is simply the area of the surface where the two histograms overlap. Due to histogram normalization, the intersection operator assumes values in $[0, 1]$ – 1 being attained when the two histograms are equal.

Although quite straightforward, the global color histogram is simply inadequate to represent *local* characteristics of image content, since it only retains color information, the knowledge of where in the image a given color is being lost in the histogramming process. In other words, simple *clustering* in the color space does not allow to retain image geometry information: to achieve that, clustering must be extended to the image space through a suitable image *segmentation* strategy.

Color-based image segmentation typically uses a predicate $P_c: \mathcal{I} \rightarrow \{\text{TRUE}, \text{FALSE}\}$ to characterize color homogeneity inside an image region \mathcal{R} . Once a color homogeneity predicate is introduced, split and merge [13] can be used to partition the image into non-overlapping regions according to the following algorithm:

COLOR-BASED SEGMENTATION

0. Let the image be initially composed of a single region \mathcal{R}
1. Split into four non-overlapping subregions $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3, \mathcal{R}_4$ each region \mathcal{R} s.t. for at least two subregions it holds $P_c(\mathcal{R}_i \cup \mathcal{R}_j) = \text{FALSE}$

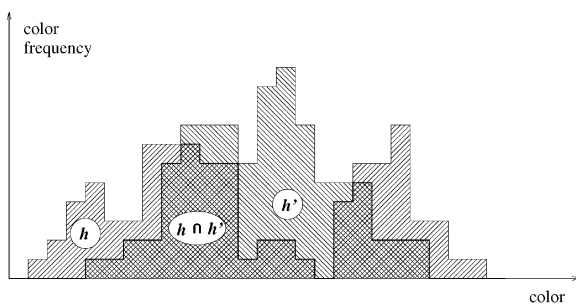


Fig. 1 The histogram intersection operator as a metric of image similarity.

2. Merge two neighboring regions $\mathcal{R}_i, \mathcal{R}_j$ s.t. $P_c(\mathcal{R}_i \cup \mathcal{R}_j) = \text{TRUE}$
3. Terminate iff there are no regions to be split (step 1) or merged (step 2)

Similar techniques for color-based image segmentation are addressed in Refs. [14–11]. Segmentation by color content is characterized by the implicit assumption that each region includes either a dominant color or a small number of characteristic colors. The next section shows how to extend color-based segmentation to color distributions, thus allowing, in principle, any color pattern homogeneously distributed inside the image to be characterized as a single region.

2.2. Segmentation based on color distributions

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, each reflecting homogeneous color distribution properties inside an image region \mathcal{R}_k . To obtain at the same time a segmentation of the image based on color distributions, and a histogram-based description of each segmented region, a color distribution homogeneity predicate $P_d: \mathcal{I} \rightarrow \{\text{TRUE}, \text{FALSE}\}$ is introduced as follows:

$$P_d(\mathcal{R}_i \cup \mathcal{R}_j) = \text{TRUE} \quad \text{iff} \quad H(\mathcal{R}_i, \mathcal{R}_j) \geq \vartheta, \quad (2)$$

where ϑ is a threshold on distribution homogeneity, and $H(\mathcal{R}_i, \mathcal{R}_j)$ extends the definition of the histogram intersection operator of Eq. (1) to image regions of general shape and size. Also notice that the unary predicate $P_d()$ is obtained from the binary operator $H()$ by computing it for the union of two image regions.

The overall image representation is obtained from the representation of each of the M segmented regions. Explicitly, the overall representation is composed by the color histogram set $F(\mathcal{I}; M, N) = \{h(\mathcal{R}_k; N, P_k), k = 1, \dots, M\}$, and an associated set of geometric features possibly describing the position and shape of each region (area, centroid, moments, etc.) and the spatial relationships between pairs of regions. These additional features are said to be “induced” by color distribution, i.e. the main feature of the representation. The parameters of the representation are, beside ϑ , the size (in pixel) of the minimum subregion (“tile”) for distribution computation, the number N of histogram bins subdividing the color space, and the area threshold φ used to retain only image regions of relevant size.

Fig. 2 illustrates the mechanism of internal representation of “Rooms by the Sea” by E. Hopper. The histogram set is composed by five elements, each related to a specific image region with a homogeneous distribution in it; region area is considered as an induced feature. In this example, the sea and the sky are distinct regions (each represented by a bimodal histogram), while the region

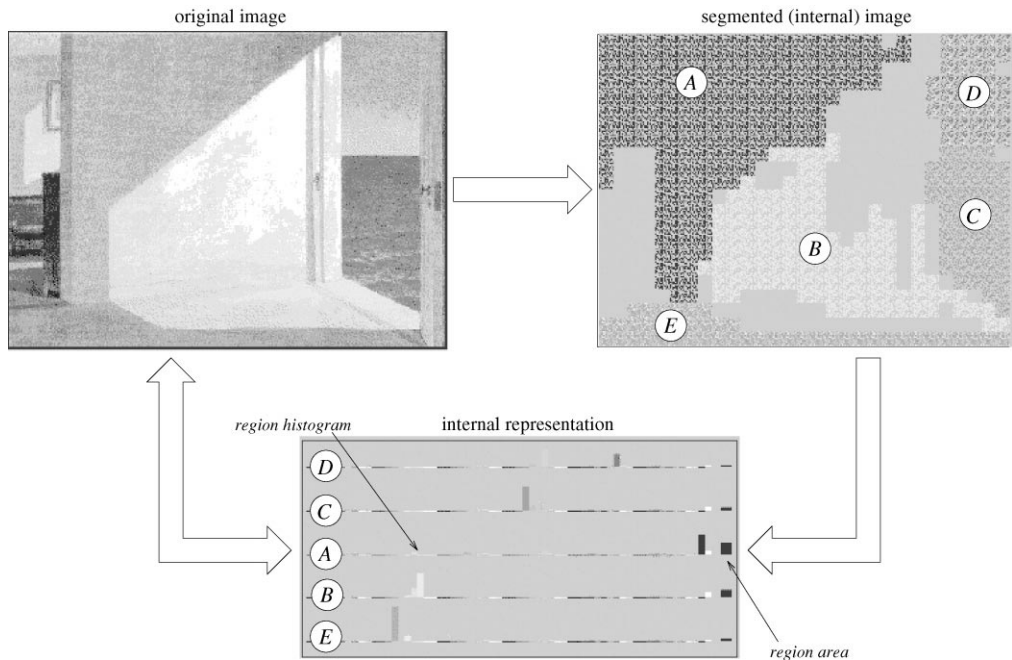


Fig. 2. Original image, and its internal 5-region, 5-set representation. Only image regions with an area exceeding a predefined threshold of 4% of the original image area are retained.



Fig. 3. An image and the two regions obtained from segmentation based on color distributions.

with largest area is formed by the shadow on the walls. Generally speaking, each distribution is usually much different from the overall image distribution, and is possibly related to a different object contained in the image; besides, notice that it is not required that the regions have only a few dominant colors: Fig. 3 clearly shows that multimodal distributions like the cloudy sky and the meadow full of flowers in C. Monet’s “Coquelicots” are successfully segmented as homogeneous regions.

2.3. A metric for image comparison

After extending the global histogram representation to histogram sets, it is required to extend the histogram intersection operator to a metric of comparison between image descriptions in the new representation. The new metric should also take into account induced features for the assessment of a global similarity score. In the following, a weighted multi-feature three-step algorithm extending the criterion of Eq. (1) to the histogram set representation augmented with the area of each segmented region is expounded.

Color distributions matching. Let \mathcal{I} denote the query image and \mathcal{J} be a generic database image to be matched against it, and let their histogram sets be, respectively, $F(\mathcal{I}; M', N)$ and $F(\mathcal{J}; M, N)$. In the matching process, since the two sets may have a different number of elements ($M \neq M'$), query and database histograms are connected in pairs and some histograms are preliminarily discarded from further match. To this aim, a *one-to-one coupling function* $\gamma: \mathbb{N} \rightarrow \mathbb{N}$ such that

$$H(\mathcal{R}_i, \mathcal{R}'_{\gamma(i)}) = \max_{j=1, \dots, \min(M, M')} H(\mathcal{R}_i, \mathcal{R}'_j) \quad (3)$$

for all $i = 1, \dots, \min(M, M')$ is used to connect regions \mathcal{R}_i and \mathcal{R}'_j in the two images. Notice that maximizing homogeneity for each histogram pair introduces a *conservation criterion* in the retrieval strategy, which is most appropriate in the case of image databases, where a false

alarm in the output is to be preferred over a missed detection. Since the coupling function has a relevant role in the representation, its computation and impact on run-time performance are addressed in detail in the appendix.

Geometric features matching. A similarity measure related to the image area occupied by color-coupled regions (and extendable to other induced features) is evaluated. Denoted with $area(\mathcal{R})$ the area occupied by an image region \mathcal{R} , geometric similarity is defined as

$$A(\mathcal{R}_i, \mathcal{R}'_{\gamma(i)}) = \frac{\min[area(\mathcal{R}_i), area(\mathcal{R}'_{\gamma(i)})]}{\max[area(\mathcal{R}_i), area(\mathcal{R}'_{\gamma(i)})]} \quad (4)$$

giving a number always positive and less or equal to 1 - unity being reached in the case of equal areas.

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:

$$K(\mathcal{I}, \mathcal{J}) = \frac{1}{\min(M, M')} \sum_{i=1}^{\min(M, M')} w_H \cdot H(\mathcal{R}_i, \mathcal{R}'_{\gamma(i)}) + w_A \cdot A(\mathcal{R}_i, \mathcal{R}'_{\gamma(i)}), \quad (5)$$

with the normalization constraint $w_H + w_A = 1$, $w_H \in [0, 1]$. The weight w_H controls the relative importance of chromatic distributions w.r.t. color-induced geometric features.

After the introduction of a metric for image similarity matching, the representation engine based on image color distribution content is complete. It is in fact possible both to produce automatically descriptions of input images at storage time, and match user query images against stored images at retrieval time based on description similarity. The next section provides an insight into the interaction aspects of image retrieval by color distributions.

3. Image retrieval

3.1. Interface and operation

Query images directly reflecting the user's current retrieval task are created interactively by visual composition (see Fig. 4). The graphic interface also provides the user with a visual feedback of both query image and system output: these can be used to update – refine, modify, etc. – the task. The interface is divided into three main areas. The *query composition area* (center and upper right) is used to construct 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 to be considered as a query to the system. The *composition tools area* (upper and center left) contains icons related to painting tools (“brush,” “rubber,” “spray” and “ink”), region handling tools (“select,” “cut,” “paste” and “move”), and general tools (“open window,” “close window,” “undo,” “exit”). A color palette for user sketch drawing is also available, together with buttons for specifying modes and parameters for retrieval. System output resulting from a query is displayed in the form of 20 image thumbnails in the *system output area* (bottom). Thumbnails are displayed in raster order according to their similarity w.r.t. the query (thumbnails to be displayed at interface setup are selected at random from the entire database).

Fig. 4 shows, in the query composition area, a *query by sketch* of the painting in Fig. 2 obtained by using the brush and selecting two different palette colors. While query by sketch can be used to recover images by heart by specifying some memorable color combinations, *query by example* is useful to rank the database images in order of similarity w.r.t. a template image. Fig. 5 shows the results of a query by example using the painting in Fig. 2 as template and the two retrieval modes embedded in the system, namely, by global histogram (*left*) and by histogram sets (*right*). Apart from retrieving the template image as best ranked, the two retrieval modes provide quite different outputs: while retrieval by histogram sets recovers a number of canvases by the same painter featuring similar color combinations (the 3rd through 8th, 15th and 16th ranked are by Hopper, of which the 3rd, 5th–7th, 15th and 16th are interiors as the template), retrieval by global histograms only recovers a Hopper's interior in 19th position.

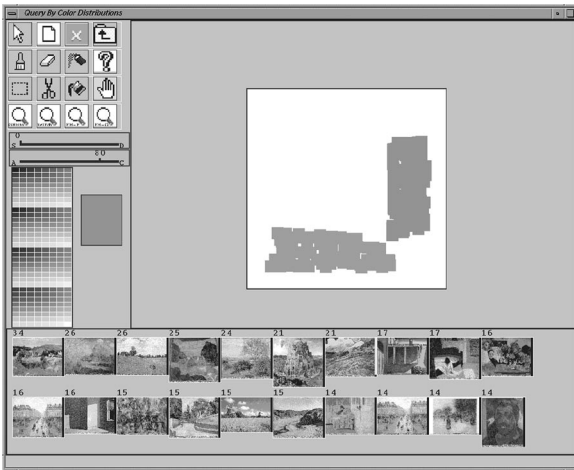


Fig. 4. System interface. In the query composition area, a sketch obtained using the “brush” selected from the graphic tools area.

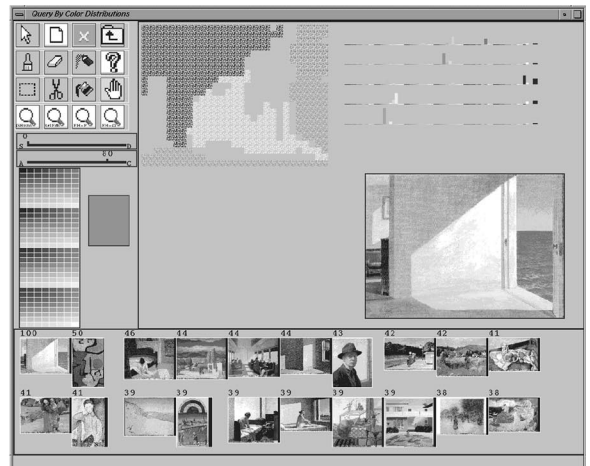
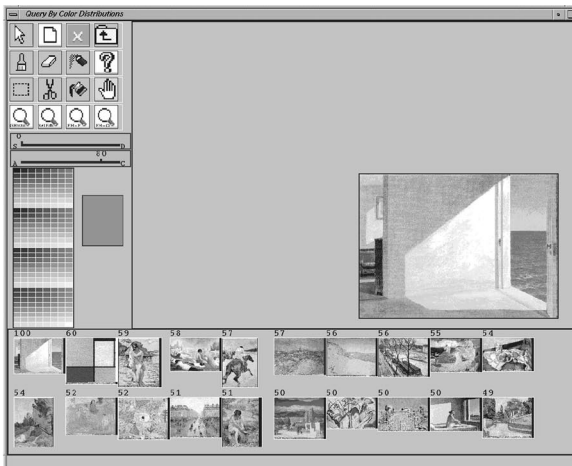


Fig. 5. Query by example. *Left*: query by global histograms. *Right*: query by histogram sets.

3.2. Query refinement and interaction examples

The system embeds three different modes – referred to respectively as relevance feedback, internal query memory and internal query manipulation – for interactive query refinement, allowing to complement and/or to modify the content of a query composition window by the specification of additional pictorial information. In the following, these three modes are addressed in detail, and illustrated via some retrieval and user-system interaction examples.

Relevance feedback. Besides modifying the query using the composition tools described earlier, information relevant for the user’s task can be taken directly from the output of the previous query. This mechanism, referred to as relevance feedback (see e.g. Ref. [10]), is based on

combining the current query content (Q_0) with the color distribution information common (according to the intersection operator) to a selected number of output images ($Q_j, j = 1, \dots, K$):

$$Q = Q_0 \cap (Q_1 \cap Q_2 \cap \dots \cap Q_K). \tag{6}$$

Relevance feedback can be iterated to progressively refine a query.

The system output sequences in the upper row of Fig. 6 show the results of retrieval by sketch using the query of Fig. 4 and global histograms (left) and histogram sets (right), respectively. Notice that, although retrieval by global histograms fails to recover the target (the painting in Fig. 2) due to a rather poor color sketch, using histogram sets succeeds in retrieving the target (ranked 4th). Generally speaking, retrieval by histogram sets is successful provided that a *strong partial match* exists with



Fig. 6. Query by sketch. In raster order: query by global histograms, query by histogram sets, relevance feedback selection, relevance feedback output.

one or more query image regions, despite the presence of even large regions with colors dissimilar from query colors. Besides, the second through fourth images are all interiors by Hopper. The lower row of Fig. 6 shows the selection of the 3rd and 4th Hopper images for query refinement by relevance feedback (*left*), and the retrieval result (*right*). After query refinement, three more Hopper interiors are retrieved in 13th, 14th, and 18th position, respectively. The paintings in 10th, 12th, and 20th position are also by the same artist.

Internal query memory. To further improve the interaction characteristics of the system, internal query memory is introduced as an additional mechanism for keeping track of previous user queries. 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

$\{U(\tau), \tau = 0, 1, \dots, t - 1, t\}$ of user queries as

$$Q(t) = v \cdot U(t) + (1 - v) \cdot Q(t - 1), \tag{7}$$

where $v \in [0, 1]$ and $Q(0) \doteq U(0)$. The case of memoryless system output, i.e. of 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 interface – together with the relative color/area weight w_H – by a scroll bar (see again Fig. 4).

Fig. 7 illustrates the principle of internal query memory. While in the case of a memoryless query (*upper left*) featuring a blue region all best ranked images have a prevailing blue color and, similarly, a memoryless query on green (*upper right*) induces to retrieve images with dominant green areas, in the presence of query memory a green query followed by repeated blue queries (*lower row*) determines images featuring *both* colors to be

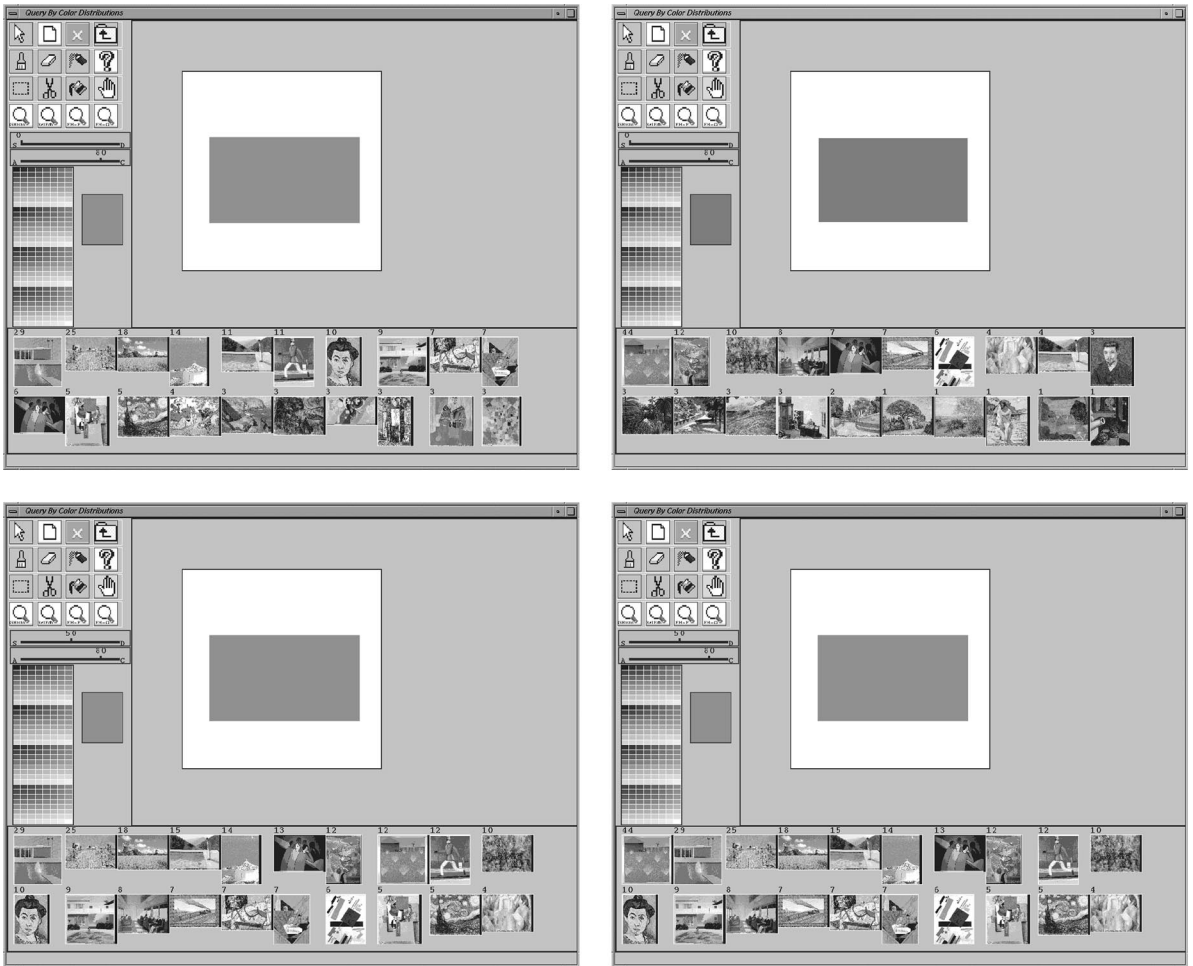


Fig. 7. Internal query memory. *In raster order:* memoryless query on blue, memoryless query on green, query memory on blue (time steps 1 and 3).



Fig. 8. Query composition and refinement. *In raster order:* query by example, query image manipulation, internal query manipulation.

ranked in the first positions. As time goes by, the memory of the green query gradually fades away, and the output sequence becomes more and more similar to that resulting from the memoryless blue query.

Internal query manipulation. The user is also allowed to refine his query by direct manipulation of the color/area histogram set (*internal query*) using the mouse. After a local histogram change is made, color frequencies are renormalized. The effect of the manipulation is back-projected onto the image space as a useful feedback information, also providing an insight into system semantics, i.e. the real way the system internally represents an image.

Fig. 8 shows the result of a query by example featuring Van Gogh's "Portrait of Doctor Félix Rey" as template. The template is characterized by a 3-set representation (*top*), where the three histograms correspond to the doctor's jacket and face and to the background wall, respectively. Retrieval by histogram sets is robust even in the presence of heavy template manipulations; in fact, after removing the background wall using the "rubber" graphic tool (*middle*), the target is still ranked first. After manipulating the jacket histogram by introducing a bias towards the red band (*bottom*), the target is ranked fifth, and three more portraits of men in red jacket are retrieved, one of which is ranked first.

It should be appreciated that the query refinement modes defined above refer to different levels of user control of internal operation. The most accurate control is achieved by internal query manipulation, but this mode is likely to be exploited only by expert users. On the other hand, internal query memory transfers most of the control from the user side to the system side, leading to less accurate results but also to a minimum of mental effort. Finally, relevance feedback realizes a trade-off between the two modes, where the system and the user equally share the computational resources to a common goal.

4. Tests and results

Diverse tests were performed with the help of a number of naive users and a database including over 250 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.) The experiments aimed at assessing (1) the impact on system performance of using histogram sets in the place of plain global histograms; (2) system accordance with human similarity judgement; (3) performance sensitivity to some important parameters such as the number of colors in the representation and the relative weight between color and geometric features.

Parameter settings. To represent colors, the *HSL* color space [13] was preferred to the conventional *RGB* space, as it agrees better with human chromatic perception. The overall number of colors used in the system is $N = N_H \cdot N_S \cdot N_L + 1$, 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 sets, a tile size of 8×8 pixel was used. Also, the homogeneity and area thresholds – controlling the number of set elements obtained for each database image – $\vartheta = 0.5$ (50%) and $\varphi = 0.04$ (4%) were chosen as a good trade-off between region representativity and significance.

Experiments. 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. Tables 1 and 2 report the results for the two 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 – the greater the numbers in the rightmost categories of each table line, the better the system performance for that line.

From the results it appears that histogram sets 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, but the former is more important than the latter: this result also provides an insight into the way users classify and memorize color images.

System performance was also assessed qualitatively by testing the system capability to recover 30 database images from user color sketches ($w_H = 0.8$). Table 3 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 (the greater the numbers in the leftmost categories of each table line, the better the system performance for that line). A position after the 20th is considered as a retrieval miss. Also in this case histogram sets 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 opposite is

Table 1
System performance: *precision* parameter

Precision	Precision categories (%)			
	0–25	25–50	50–75	75–100
Search Type	0–25	25–50	50–75	75–100
Global histograms	51	29	20	0
Sets, $w_H = 0.3$	46	54	0	0
Sets, $w_H = 0.6$	30	40	29	1
Sets, $w_H = 0.8$	16	29	28	27
Sets, $w_H = 1.0$	18	38	19	25

Table 2
System performance: *recall* parameter

Recall	Recall categories (%)			
	0–25	25–50	50–75	75–100
Search Type	0–25	25–50	50–75	75–100
Global histograms	49	30	12	9
Sets, $w_H = 0.3$	44	33	23	0
Sets, $w_H = 0.6$	23	7	48	22
Sets, $w_H = 0.8$	18	23	35	24
Sets, $w_H = 1.0$	11	42	23	24

Table 3
System performance: retrieval by user sketch

Retrieval by user	Position in system output			
	1–3	4–10	11–20	Missed
Sketch Search Type	1–3	4–10	11–20	Missed
Global histograms, $N = 61$	8	1	5	16
Global histograms, $N = 7$	8	3	5	14
Histogram sets, $N = 61$	13	4	7	6
Histogram sets, $N = 7$	17	3	4	6

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. Conclusions

In this paper, 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 segmented from the images. A specific metric has been developed in order to perform comparisons with such multi-feature image

representation. The internal query description is created graphically by composing the current image query by possibly mixing tools such as color sketching, visual examples, relevance feedback, and a memory of previous queries. Finally, a novel method for query refinement was introduced, based on accessing and manipulating through graphics directly the internal query representation, so as to improve and exploit the user's knowledge of system behavior.

The proposed system has a number of potential application fields, ranging from art paintings to aerial and satellite imagery, and geographic information systems. The system can be augmented and enhanced in several ways. For instance, a sharper sensitivity to geometric features such as 2D shape and spatial relationships can be introduced by considering higher-order moments of the extracted image regions. Also, to improve further system memory and user adaptation, the concept of query memory can be extended to define a state of the database itself (active index).

Acknowledgements

The authors would like to thank Dr. Ivan Genovesi for his contribution in the development of this work.

Appendix: Coupling function and search complexity

In order to determine how system parameters affect the overall search complexity and time, a few words have to be spent on the way the coupling function γ is computed.

Let \mathcal{I}' denote the query image and \mathcal{I} be the generic database image to be matched against it, and let their histogram sets be, respectively, $F(\mathcal{I}'; M', N)$ and $F(\mathcal{I}; M, N)$. Let also the histogram intersection $H(\mathcal{R}_i, \mathcal{R}'_j)$ between regions $\mathcal{R}_i \subset \mathcal{I}$ and $\mathcal{R}'_j \subset \mathcal{I}'$ be indicated by the number σ_{ij} . The first step in the computation of the coupling function is to construct an $M \times M'$ matrix featuring all the possible intersections σ_{ij} :

$$\Sigma(\mathcal{I}, \mathcal{I}') = \begin{matrix} & \mathcal{R}'_1 & \mathcal{R}'_2 & \dots & \mathcal{R}'_{M'} \\ \begin{matrix} \mathcal{R}_1 \\ \mathcal{R}_2 \\ \vdots \\ \mathcal{R}_M \end{matrix} & \begin{pmatrix} \sigma_{11} & \sigma_{12} & \dots & \sigma_{1M'} \\ \sigma_{21} & \sigma_{22} & \dots & \sigma_{2M'} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{M1} & \sigma_{M2} & \dots & \sigma_{MM'} \end{pmatrix} \end{matrix} \quad (A.1)$$

The second step selects from Σ a subset $\{\sigma_{i(k)j(k)}, k = 0, \dots, \min(M, M') - 1\}$ of entries such that $i(k_1) \neq i(k_2)$ and $j(k_1) \neq j(k_2)$ if $k_1 \neq k_2$ (coupling subset). This is a way to ensure that the coupling function is one-to-one; in fact, each region \mathcal{R}_i as well as each region \mathcal{R}'_j is encoded at most in one of the $\sigma_{i(k)j(k)}$'s. Operationally speaking, the elements of the coupling subset are ob-

tained as follows. First, $\sigma_{i(0)j(0)}$ is obtained as the maximum among the elements of Σ , and row $i(0)$ and column $j(0)$ are deleted from Σ . Then $\sigma_{i(1)j(1)}$ is obtained as the maximum of the remaining entries of Σ , row $i(1)$ and column $j(1)$ are further deleted from Σ , and so forth.

It is not difficult to show (see e.g. Ref. [17]) that, since the algorithm for extracting the maximum from an array of n data has a linear – i.e. $\mathcal{O}(n)$ – complexity, then the second step in the computation of the coupling function is $\mathcal{O}(\overline{M}^3)$, where for simplicity we assume $M' = M = \overline{M}(\mathcal{G}, \varphi)$ the average number of regions obtained using the segmentation thresholds \mathcal{G} and φ . The complexity of the first computational step depends instead on both the average number of regions, and on the number N of colors used in the histogram representation, as $\mathcal{O}(\overline{M}^2 N)$, so that the computational complexity of image search from a database set of L images is

$$\mathcal{C}(L, \overline{M}, N) = \mathcal{O}(L\overline{M}^2 N) + \mathcal{O}(L\overline{M}^3). \quad (A.2)$$

Thus, the overall search time depends linearly on both the number of images being matched against the query and the number of histogram bins, while it depends at most cubically on the average number of segmented regions. Specifically, if $N > \overline{M}$, then the first term on the right-hand side of Eq. (A.2) prevails, and the dependence on the number of regions is quadratic; actually though, since N is usually smaller than \overline{M} , it is the second term which prevails. Anyway, from Eq. (A.2) it appears that \overline{M} has a crucial impact on search-time performance: this number should be kept as small as possible by accurately setting the selectivity thresholds \mathcal{G} and φ , taking into account that \overline{M} increases with increasing values of \mathcal{G} and with decreasing values of φ . The average number of image regions obtained with $\mathcal{G} = 0.5$ and $\varphi = 0.04$ (the parameter values used in the Section 4) is about 7.

It should also be noted that, since usually an image database features thousands of images, it is the linear term L which has the most dramatic impact on performance. This term can be kept smaller than the overall database dimension by appropriate indexing techniques, allowing to search for similar database images only in a suitable “neighborhood” of the current query image.

References

[1] V.N. Gudivada, V.V. Raghavan (Eds.), Content-based image retrieval systems, IEEE Comput. 28 (9) (1995) (special issue).
 [2] R.W. Picard, A.P. Pentland (Eds), Digital libraries: representation and retrieval, IEEE Trans. Pattern Anal. Machine Intelligence 18 (8) (1996) (special issue).
 [3] A. Gupta, R. Jain, Visual information retrieval, Commun. ACM 40 (5) (1997).
 [4] V.E. Ogle, M. Stonebraker, CHABOT: retrieval from a Relational Database of Images, IEEE Comput. 28 (9) (1995).

- [5] M. Swain, D. Ballard, Color indexing, *Int. J. Comput. Vision* 7 (11) (1991).
- [6] K. Hirata, T. Kato, Query by visual example – content-based image retrieval, in: *Proc. EDBT'92. Lecture Notes in computer Science*, Springer, Berlin, 1992, pp. 56–71.
- [7] 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.
- [8] A. Del Bimbo, P. Pala, Visual image retrieval by elastic matching of user sketches, *IEEE Trans. Pattern Anal. Machine Intelligence* 19 (2) (1997).
- [9] S.-K. Chang, E. Jungert, Pictorial data management based upon the theory of symbolic projections, *J. Visual Languages Comput.* 2 (2) (1991).
- [10] T.P. Minka, R.W. Picard, Interactive learning with a ‘Society of Models’, *Pattern Recognition* 30 (4) (1997).
- [11] A. Del Bimbo, M. Mugnaini, P. Pala, F. Turco, Visual querying by color perceptive regions, *Pattern Recognition* 31 (9) (1998).
- [12] S. Santini, R. Jain, The graphical specification of similarity queries, *J. Visual Languages Comput.* 4 (5) (1996).
- [13] D. Ballard, C. Brown, *Computer Vision*, Prentice-Hall, Englewood Cliffs, NJ, 1982.
- [14] R. Schettini, A segmentation algorithm for color images, *Pattern Recognition Lett.* 14, 1993.
- [15] J.R. Smith, S.F. Chang, VisualSEEK: a fully automated content-based Image query system, in: *Proc. ACM Multimedia '96*, ACM 1996.
- [16] W.Y. Ma, B.S. Manjunath, NeTra: a toolbox for navigating large image databases, in: *Proc. ICIP'97*, IEEE, New York, 1997, pp. 568–571.
- [17] Donald E. Knuth, *The Art of Computer Programming. Vol. 3: Sorting and Searching*, 2nd ed. Addison-Wesley, 1998. Reading, MA.

About the Author—CARLO COLOMBO holds a M.S. in Electronic Engineering from the University of Firenze, Italy (1992) and a Ph.D. in Robotics from the Scuola Superiore di Studi Universitari e di Perfezionamento Sant’Anna, Pisa, Italy (1996). In 1994 he was a visiting scientist at LIFIA-IMAG, Polytechnic of Grenoble, France. From 1996 to 1998 he was assistant Professor at the University of Brescia, Italy. In 1999 he joined the Dipartimento di Sistemi e Informatica of the University of Florence, Italy. Dr. Colombo is a member of IEEE – the Institute of Electrical and Electronics Engineers – and IAPR – the International Association for Pattern Recognition – and presently serves as the Secretary of the IAPR Italian Chapter. He is Guest Editor of the *Robotics and Autonomous Systems* Journal (special issue on “Intelligent Robotic Systems ’95”, March 1997). His main research activities are in the field of computer vision, with specific interests in image and video analysis, human–machine interfaces, robotics and multimedia.

About the Author—ALBERTO DEL BIMBO graduated in Electronic Engineering at the University of Florence, Italy in 1977. From 1977 to 1988 he was with IBM Italy SpA. Presently, he is Full Professor at the Faculty of Engineering of the University of Florence, Italy. His scientific interests cover image sequence analysis, shape and object recognition, image databases and multimedia, visual languages and advanced man–machine interaction. Prof. A. Del Bimbo is presently an Associate Editor of *Pattern Recognition* and *Journal of Visual Languages and Computing*, and *IEEE Transactions on multimedia*. He is a member of IEEE and IAPR. He presently serves as the Chairman of the Italian Chapter of IAPR. He is the General Chair of IAPR *ICIAP'97* – Int. Conf. on Image Analysis and Processing – and of IEEE *ICMCS'99* – Int. Conf. on Multimedia Computing and Systems – and has been in the Program Committee of many International Conferences in the field of Imaging Technology and Multimedia Computing. Prof. A. Del Bimbo is the author of over 120 scientific publications appeared in international journals and conference proceedings and has been Guest Editor of several special issues on Imaging Technologies on some of the most important journals in the field.