

Colour Image Retrieval Fitted to «Classical» Querying

José Martinez and Sylvie Guillaume

Institut de Recherche en Informatique de Nantes (IRIN)
IRESTE, La Chantrerie, B.P. 60601
44306 NANTES Cedex 3, FRANCE
e-mail: Jose.Martinez@irin.univ-nantes.fr

Abstract. Query by image content is certainly the main problematic in the field of multimedia databases. In this paper, we investigate the usage of a perceptual colour space during and after a coarse segmentation process in order to offer to users the possibility of querying the obtained regions thanks to a query language and fuzzy descriptions of colour, spatial, and topological properties of these regions. By being understandable to the user, contrary to criteria used in comparative searches, it allows a deterministic, though more or less constrained search. Nevertheless, by maintaining a rich set of characteristics, it does not preclude the computation of simpler properties usable in such comparative searches.

1 Introduction and Motivations

From a database point of view, it is highly desirable to integrate multimedia data as conveniently as more traditional strings and numbers. We propose a fuzzy description of a colour image based on a coarse segmentation using a perceptual colour space. First, a perceptual colour space is used to allow a user to utilize terms such as «red», «dark», and «palish», i. e., strongly correlated to the natural language. Next, a coarse segmentation algorithm is used in order to minimise the number of regions in an image, i. e., to store in the database only the regions that can be of actual interest for the user. Finally, the whole description of the image is translated from the numerical domain to a fuzzy one with terms such as «large», «on the left», «rounded» ...

A clear advantage of this kind of description is that it allows a user to issue a standard query rather than providing an example image or a rough drawing which is mainly what is offered by current systems like QBIC [Flickner *et al.* 95], Virage [Bach *et al.* 96], VisualSeek [Smith & Chang 96], and other proposals [Lin *et al.* 95] [Gevers 96], Chabot (now Cypress) [Ogle & Stonebraker 95] being almost an exception. As noted by [Lin *et al.* 95], textual attributes may be insufficient to describe an image. In our opinion, they should be used for the semantic description of the image, including certainly what is not directly visible on it. What is really needed from an image description tool is to avoid the tedious task to describe everything that is directly *visible* on the picture. Consequently, textual attributes and image region attributes should be mixed together in the same query. This allows an effective retrieval ratio with a high semantical content [Ogle & Stonebraker 96], e. g., «Find all the pictures where appears the French President», and further restrictions on the set of answers by using information which is inherently present in the images, e. g., «Find [...] President and where there are blue regions at the top of the image and/or green regions» to obtain only photographs of the President on the outside.

The outline of the paper is as follows: First, the choice of the colour space is debated. Then, in Section 3, a segmentation algorithm is presented. Next, database

design issues are presented in Section 4. Section 5 introduces early experiments on top of the O_2 object-oriented database management system (OODBMS.) In the conclusion, we list several issues, some of which are already under investigation.

2 Choosing a Colour Space

For database applications where one of our goals is to offer to the user the ability to describe the kind of images that he or she wants to retrieve, it seems clear that a perceptual colour space is required. As a counter-example, a user can be asking for images containing a significant quantity of red and consequently he or she could obtain images with only blue and yellow! (This actually occurred to the authors using one of the existing systems at our disposal on Internet.)

In a perceptual colour space, three criteria express the colour: *hue* that corresponds very roughly to the wavelength, *saturation* which indicates the purity of the hue, and *intensity* that corresponds to the quantity of white. In the literature, we found out five such sets of formulæ, sometimes called HSI, HSL, or HSV, not to mention indirect translations through the CIE Luv or Lab colour spaces.

Definition 1: A set of formulæ that permits to translate an image from the RGB colour space into an HSV colour space is as follows:

$$h: \begin{aligned} & [0, 1]^3 \rightarrow [0, 6[\\ & \perp \text{ if } s = 0 \\ & (r, g, b) \rightarrow \begin{cases} \frac{g - b}{\max\{r, g, b\} - \min\{r, g, b\}} \text{ if } v = r \\ \frac{b - r}{\max\{r, g, b\} - \min\{r, g, b\}} + 2 \text{ if } v = g \\ \frac{r - g}{\max\{r, g, b\} - \min\{r, g, b\}} + 4 \text{ if } v = b \end{cases} \end{aligned} \quad (\text{EQ 1})$$

$$s: \begin{aligned} & [0, 1]^3 \rightarrow [0, 1] \\ & \perp \text{ if } v = 0 \\ & (r, g, b) \rightarrow \begin{cases} \frac{\max\{r, g, b\} - \min\{r, g, b\}}{\max\{r, g, b\}} \text{ otherwise} \end{cases} \end{aligned} \quad (\text{EQ 2})$$

$$v: \begin{aligned} & [0, 1]^3 \rightarrow [0, 1] \\ & (r, g, b) \rightarrow \max\{r, g, b\} \end{aligned} \quad (\text{EQ 3})$$

It presents some advantages: (1) it is computationally inexpensive, (2) it is statistically more regular than other computations [Pujas 96], and (3) since in our current work it is heuristically partitioned, it does not require to be uniform. (In the implementation, we normalise the h , s , and v dynamics to a byte range, i. e., from 0 to 255.)

[Pujas & Aldon] introduces a heuristic dichotomy function between chromatic and achromatic pixels (i. e., grey-level ones), to be able to use classical segmentation algorithms. Our proposal is, to some extent, an extension of this dichotomy.

Definition 2: The HSV colour space is partitioned into nine colour classes, as follows:

$$\text{class}(h, s, v) = \begin{cases} \text{black if } v \leq 20 \\ \text{grey if } 20 < v \leq 230 \wedge s \leq 30 \\ \text{white if } v > 230 \wedge s \leq 30 \\ \text{orange if } v > 20 \wedge s > 30 \wedge 9 \leq h \leq 28 \\ \text{yellow if } v > 20 \wedge s > 30 \wedge 29 \leq h \leq 46 \\ \text{green if } v > 20 \wedge s > 30 \wedge 47 \leq h \leq 117 \\ \text{blue if } v > 20 \wedge s > 30 \wedge 118 \leq h \leq 184 \\ \text{magenta if } v > 20 \wedge s > 30 \wedge 185 \leq h \leq 244 \\ \text{red if } v > 20 \wedge s > 30 \wedge (0 \leq h \leq 8 \vee 245 \leq h \leq 255) \end{cases} \quad (\text{EQ } 4)$$

These limits have been empirically determined by visual inspection and dichotomic search on magnification of HSV slices. It is probably too much an *ad hoc* way to define colours. Nevertheless, experiments which are shortly presented in Section 5, show that the obtained regions are satisfactory for sufficiently contrasted images. (We modified the limits and segmented the same set of images with different versions of the class function without significant differences.)

3 Designing an Efficient Segmentation Algorithm

Even if the segmentation process is known to be an *ill-conditioned* problem, it has been for a long time at the heart of image processing and image analysis [Ballard & Brown 82].

Definition 3: The output of a segmentation algorithm is first a partitioning of an image I into a set of regions R_i , i. e., a set of sets of pixels verifying:

$$\forall i, R_i \neq \emptyset \quad (\text{EQ } 5)$$

$$\forall i, \forall j, i \neq j \Rightarrow R_i \cap R_j = \emptyset \quad (\text{EQ } 6)$$

$$\bigcup_{\forall i} R_i = I \quad (\text{EQ } 7)$$

and three other requirements: connectedness of the regions, homogeneity of the regions, and maximality of the regions,

$$\forall i, \forall p_1 \in R_i, \forall p_2 \in R_i, 4\text{connected}(p_1, p_2) \quad (\text{EQ } 8)$$

$$\forall i, H(R_i) \quad (\text{EQ } 9)$$

$$\forall i, \forall j, \text{adjacent}(R_i, R_j) \Rightarrow \neg H(R_i \cup R_j) \quad (\text{EQ } 10)$$

Definition 4: Our homogeneity criterion is based on the common membership to some class, as given by Definition 2:

$$H(R) \Leftrightarrow \forall p_1 \in R, \forall p_2 \in R, \text{class}(p_1) = \text{class}(p_2) \quad (\text{EQ } 11)$$

The choice of such a simple criterion, that mainly uses hue, is based on the fact that hue is insensitive to a great deal of physical parameters.

Property 1: Assuming a Torrance-Sparrow [Gevers & Smeulders 96] or a Phong [Gevers 96] reflection model and white illumination, hue is independant of viewpoint, surface orientation, illumination direction, illumination intensity, and highlights.

The overall processing stages are as follows: First, a pre-processing stage is responsible for providing a better image to the segmentation process. The image is firstly reduced to eliminate as much tiny details as possible. Then, we remove some remaining noise and details by using a median filter which (1) does not blur the image, (2) further diminishes small regions, and (3) does not introduce false hues. Finally, the image is translated into the HSV colour space accordingly to Definition 1.

In the second stage, we implemented a variation of a recursive region growing algorithm [Ballard & Brown 82].

The last stage consists in eliminating the small regions. As a rule of thumb, and again because we are interested by queryability, we decided to eliminate regions, the areas of which are under a threshold of 1 % of the total size of the image. For the remaining regions, various other properties are computed such as perimeter, minimal bounding rectangle, average hue, saturation, and intensity, x and y centroids, *etc.* This stage is also in $O(n)$ thanks to the inverted tree representation of a partition. These results are written down into a text file, described in the following section.

The pseudo-code of our segmentation algorithm is given in Fig. 1

Property 2: The output of this algorithm is indeed a segmentation with respect to Definition 3.

Property 3: This algorithm is linear in the size of the image.

Note that to achieve this efficiency, regions have been implemented as partitions using the known inverted tree representation of Fisher-Galler (See, for example, [Jones 90]), enhanced by the idea of reducing the depth of branches when collapsing two trees [Dijkstra 76]. In point of fact, we adopted an implementation similar to the one proposed by [Fiorio & Gustedt 94] for grey-level images. Over-segmentation does not occur thanks to our broad classification except for some fine and contrasted textures.

As an illustration, Fig. 2 (a) presents a sunset image. The segmentation algorithm generates a map of 565 regions, the contours of which are given in Fig. 2 (b). However, the thresholding decreases dramatically this number down to 5 regions of interest, corresponding to the sun and the sky above it, the horizon which is orange rather than yellow, the black mountains, and the lake.

4 Querying an Object-Oriented Image Database

We use the O_2 OODBMS as a repository for our researches. The file, which is the final result of the processing stages described before, is imported into the O_2 schema, a very simple subset of the whole C++ framework, described in Fig. 3. The class `Picture` contains the general properties of an image such as its width, height, *etc.* More importantly, an image is seen as a composite object, made up of a set of regions. In turn, these regions have a set of properties such as the minimal bounding rectangle (X_{min} , Y_{min} , X_{max} , and Y_{max}), x and y centroids, area, *etc.* We decided to maintain the exact numerical values.

The database is then filled with this summarized information, as well as with thumbnail images. Not to bother the user with these numerical values, a set of predicates is provided. For instance, a large region is one for which its exact area is at least 20 % of the area of the corresponding image. Depending on the kind of applications and user needs, other predicates can be added (e. g., hot and cold colours.)

```

function Segment:  $(I, x_1, y_1, x_2, y_2) \rightarrow \{R_i = \{(x_i, y_i)\} \mid x_1 \leq x_i \leq x_2 \wedge y_1 \leq y_i \leq y_2\}$ 
if  $x_1 = x_2 \wedge y_1 = y_2$  // the size of the area to segment is limited to one pixel
then return  $\{\{(x_1, y_1)\}\}$  // creates a new region consisting of that pixel alone
elseif  $x_2 - x_1 \leq y_2 - y_1$  // the image is higher than wider
then
    let  $y_{half} = \left\lfloor \frac{y_2 - y_1 + 1}{2} \right\rfloor$  // splits up the image into two halves along the height
    let  $R_{top} = \text{Segment}(I, x_1, y_1, x_2, y_{half})$  // segments recursively the first half,
    let  $R_{bottom} = \text{Segment}(I, x_1, y_{half+1}, x_2, y_2)$  // and the second one
    var  $R \leftarrow R_{top} \cup R_{bottom}$  // the final result will be a subset of their union
    for  $x \in \{x_1, \dots, x_2\}$  do // follows the boundary
        if  $\text{class}(x, y_{half}) = \text{class}(x, y_{half+1})$ 
            then // neighbouring pixels of the «same colour»
                let  $R_1 \in R: (x, y_{half}) \in R_1$ 
                let  $R_2 \in R: (x, y_{half+1}) \in R_2$ 
                 $R \leftarrow (R - R_1 - R_2) \cup (R_1 \cup R_2)$  // merge their respective regions
            end if
        end for
    return  $R$ 
else // the image is wider than higher
    ... // analogous to the previous case
end if

```

Fig. 1: The segmentation algorithm

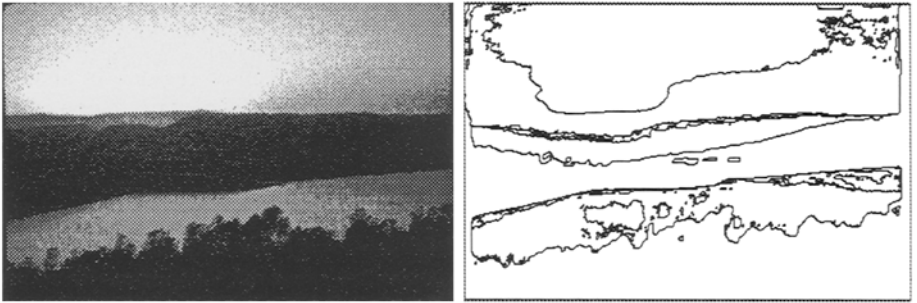


Fig. 2: (a) A sunset on a lake; (b) The contours of the 565 regions

In image retrieval by image example, various kinds of criteria have been proposed, from fast calculated but imprecise ones (Euclidian distance between histograms being a must) to much more expensive but also more effective ones. However, in all these cases, the criteria are hidden to the naive user. Therefore, he or she often does not understand why some images are retrieved, and why those that look to him or her more

adequate are proposed after less convincing ones. Our proposal offers a way to retrieve images that respond *exactly* to the user's query. An early experiment has been conducted with a subset of the properties introduced in the previous section and only 147 images. The goal was to obtain rapidly a feedback on the effectiveness of our approach.

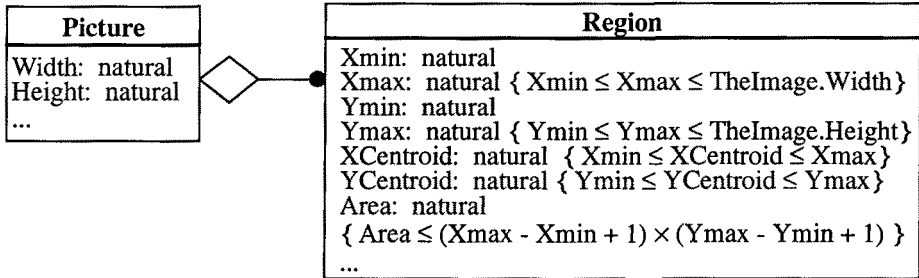


Fig. 3: A summary of our database schema

Example: A sunset could be described by the user as a yellow region surrounded by an orange one, the bottom of the image being dark. This may translate into the following OQL query [Cattel 94]:

```

select p
from p in Pictures
where exists sun in p.Regions:
    (sun.yellow and exists sky in p.Regions:
        (sky.orange and sky.adjacent(sun)) and
    exists ground in p.Regions:
        (ground.dark and ground.bottom and not ground.small))
  
```

The reader can note that this query does not accurately take into account each and every sunset image. Part of the answer to this query is given in Fig. 4. Eleven images are retrieved from which height represent really a sunset and only three something else. Note that these images are *exact* answers to the query, however. In particular, there is no ranking. Furthermore, all the sunsets are not retrieved: only height out of fifteen. In one case, the sun is not visible, so there is no yellow region. In another image, the sea is reflecting the light of the sun, therefore there is no dark region at the bottom of the picture. As another cause, there is also a *drawing* of a sunset, where the sky has been painted in night blue rather than orange ... Considering this example, it is not possible to judge that the search has not been as successful as one might expect. In effect, either the user did not take the time to write a lengthy query that takes into account much more kinds of sunsets, or he or she really wanted those artistically classical images of sunsets. Only in the first case should the system be extended in order to help the user to construct a more complex query.

5 Conclusion and Future Work

In this paper, we presented the design, implementation, and preliminary results of querying an image database by using a query language. The objective was to offer to the user a set of predicates that he or she can introduce in his or her queries.

A segmentation algorithm was designed with the goals of segmenting colour images into large regions. The first experiments showed good results in general,

whereas expected difficult images need indeed further reflection on the way that they can be queried.

Fuzzyness could also be introduced in the segmentation process to deal with smooth colour transitions as well as in the querying process, e. g., fuzzy transition from a large region to a medium one.

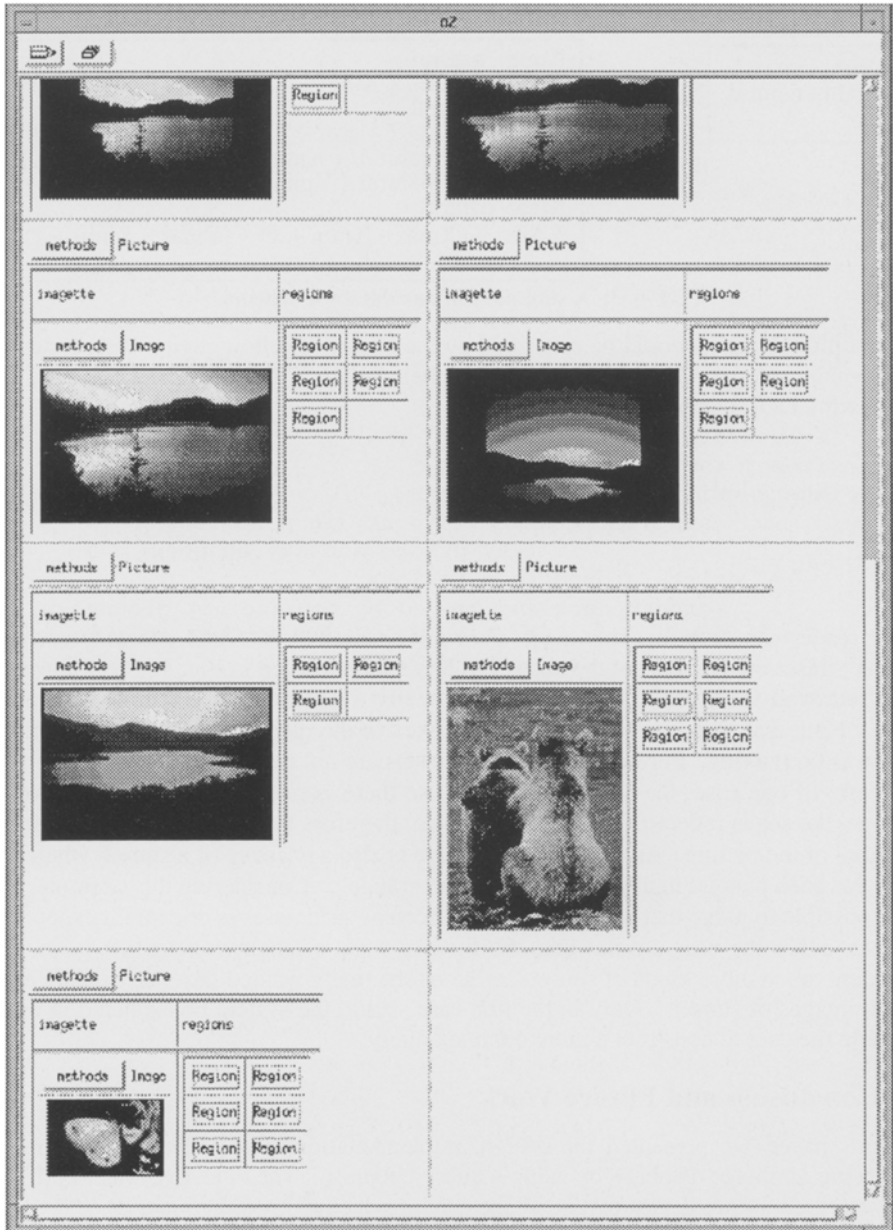


Fig. 4: Excerpt from the sunset query answer

Also, comparative searches can be based on the extracted characteristics. It may also be desirable to start from the first returned set of solutions to retrieve other images by indicating which ones are more or less interesting. A new OQL request will then be generated automatically by the system, taking into account much more properties than what can be reasonably expected from the user to type in!

References

- [Bach *et al.* 96] Bach, J. R., Fuller, C., Gupta, A., Hampapur, A., Horowitz, B., Humphrey, R., Jain, R. C., Shu, C.; *Virage Image Search Engine: An Open Framework for Image Management*; Proc. of the IS&T/SPIE Int. Symposium on Electronic Imaging: Science and Technology, Storage & Retrieval for Image and Video Databases IV, 1996, pp. 76-87
- [Ballard & Brown 82] Ballard, D. H., Brown, C. M.; *Computer Vision*; Prentice-Hall, 1982, 523 p.
- [Cattell 94] Cattell, R. G. G. (ed.); *The Object Database Standard: ODMG-93*; Morgan Kaufmann, San Francisco, 1994, 176 p.
- [Dijkstra 76] Dijkstra, E. W.; *A Discipline of Programming*; Prentice Hall, 1976
- [Fiorio & Gustedt 94] Fiorio, C., Gustedt, J.; *Two Linear-Time Union-Find Strategies for Image Processing*; Research Report No. 375, Technische Universität Berlin, Fachbereich 3, Mathematik, 1994, 12 p.
- [Flickner *et al.* 95] Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., Yanker, P.; *Query by Image and Video Content: The QEIC System*; IEEE Computer, September 1995, pp. 23-32
- [Gevers 96] Gevers, T.; *Colour Image Invariant Segmentation and Retrieval*; Ph. D. Thesis, University of Amsterdam, Netherlands, May 1996, 142 p.
- [Gevers & Smeulders 96] Gevers, T., Smeulders, A. W. M.; *Color-Metric Pattern-Card Matching for Viewpoint Invariant Image Retrieval*; Proceedings the IEEE Int. Conf. on Pattern Recognition (ICPR), 1996, pp. 3-7
- [Jones 90] Jones, C. B.; *Systematic Software Development using VDM, 2nd Edition*; Prentice Hall, 1990
- [Lin *et al.* 95] Lin, H.-C., Wang L.-L., Yang, S.-N.; *Color Image Retrieval Based on Hidden Markov Models*; Proceedings of the IEEE Int. Conf. on Image Processing (VCIP), 1995, pp. 342-345
- [Ogle & Stonebraker 95] Ogle, V. E., Stonebraker, M.; *Chabot: Retrieval from a Relational Database of Images*; IEEE Computer, September 1995, pp. 40-48
- [Pujas 96] Pujas, P.; *Analyse d'images couleur et fusion d'images 3D et couleur* (in french); Ph. D. Thesis, University of Montpellier II, France, February 1996, 249 p.
- [Pujas & Aldon 94] Pujas, P., Aldon, M.-J.; *Robust Colour Image Segmentation*; Proc. of ICAR, Sant Feliu de Guixols, Catalunya, Spain, 1994
- [Smith & Chang 96] Smith, J. R., Chang, S.-F.; *VisualSEEK: a Fully Automated Content-Based Image Query System*; ACM Multimedia, Boston, Massachusetts, November 1996