Journal of Visual Languages and Computing (2000) **11,** 303–321
doi:10.1006/jvlc.2000.0161, available online at http://www.idealibrary.com on IDEAL®

Retrieval of Images using Rich-region Descriptions

L. CINQUE,* F. LECCA,* S. LEVIALDI* AND S. TANIMOTO†

** Department of Information Science, University of Rome 'La Sapienza', via Salaria 113, 00198 Rome, Italy, E-mail: cinque@dsi.uniroma1.it. † Department of Computer Science and Engineering, Box 352350, University of Washington, Seattle, WA 98195, U.S.A.*

Accepted 18 January 2000

Retrieval of images from databases using their visual features is a challenging and important problem. While the technical problem shares some aspects of image analysis with image understanding, the goal is not to obtain a correct interpretation of the image but to enhance the recall and precision of retrieval. The dominant visual features of an image depend on subjective interpretations and can vary from user to user. We present a technique to improve recall in region-based retrieval; the method is based upon a family of representations of images called 'rich-region descriptions'. We show in a simple experiment how this kind of representation can improve the flexibility allowed to users in obtaining desired results. We also discuss issues related to the user interface for segmentation and query systems. In this subject, the paper extends a previous work.
© 2000 Academic Press

Keywords: content query, image database, segmentation, regions, matching, similarity, distance, user interface, rich-region description.

1. Introduction

1.1. General Motivation

INFORMATION RETRIEVAL using images is an area of increasing importance, because the growing number of images online (particularly through the World Wide Web) offer opportunities to provide new services in art, archeology, advertising, real-estate, scholarly research and entertainment.

Today's techniques are limited by the lack of understanding of the best ways of characterizing images through signatures as well as the lack of understanding of certain kinds of functions that compute distances between images. These distance functions need to work with complex structures in order to reflect the perceptual and semantic features commonly used by humans.

Here we discuss a family of image signatures and related distance functions in order to provide a better foundation for the design of image query systems. We present a particular experiment with a representation we call 'rich-region descriptions' to show how these methods can work in practice. We also outline design considerations for user

1045-926X/00/060303+19 $35.00/0

interfaces that enhance the user's understanding and control of the image retrieval process using such image signatures and distance functions.

1.2. Problem Description

We define the image-query-by-content problem as follows. Given a query image I_1 and a database of images $\{I_2, \ldots, I_n\}$, find the image I_i closest to I_1. The closeness is to be computed using a distance function $D(I_i, I_j)$ which evaluates the shape, color, texture, position, and/or importance of the regions within their images.

1.3. Relationship to Pattern Recognition Problems

We note that the image-query-by-content problem involves a kind of image processing which shares some similarities with the techniques used for pattern recognition. For example, the image signatures used in recognition may use descriptive elements similar to those used in feature vectors and symbolic descriptions within image understanding applications.

However, there are important differences in the goals and techniques for image retrieval and image understanding. For example, whereas ambiguity in image representations is generally undesirable when performing image understanding, it is often desirable when performing image retrieval. Also, artifacts of image analysis algorithms, such as blockiness of regions, false contours, and the like, are the bane of image interpretation algorithms; however, they are sometimes harmless in image-retrieval applications.

The challenges of these two application areas are also different. In image understanding, the difficulty is to obtain the interpretation of the image that corresponds to the way a human would label each of the objects or regions in the image. In other words, the problem is to make the computer do what the human *does*. On the other hand, the difficulty of image information retrieval is to infer from the query the human user's intentions and preferences for images in a database. The challenge, then, is to figure out and do what the human *wants*.

1.4. Use of Regions

Most of the previous work on image retrieval by content uses descriptions of the images that are based upon statistical aspects of images (color histograms) or color and texture samples taken at fixed locations in the images. Some shape analysis has been performed when the images under consideration are simple. In order to allow better and more general kinds of retrieval, we employ descriptions of images in terms of *regions* that are obtained from a partial analysis of each image.

2. Previous Work

The fundamental principles related to the retrieval of information from textual databases using queries were studied during the 1960s and described in the work of Salton [1]. The concepts of recall and precision are key in evaluating retrieval methods, whether they be text-based or image-based.

In the literature, image query by contents refers to the process by which a user submits a query that is either pictorial in form or expressed in terms of visual properties, after which the system attempts to return to the user the image(s) from the database that best match the query. Previous work on this problem includes work on the QBIC system at IBM Almaden [2, 3], work by Jain *et al.* at UC San Diego (see [4]), and others [5–17]. A survey of these techniques has been given by Cinque *et al.* [18]. Image query by contents is a form of 'iconic indexing' [19].

The use of regions of images in computing the signatures employed for matching images has been quite limited to date. One reason for this is the relative complexity of segmentation algorithms as compared with the statistical methods for describing images. Another reason is that segmentation has been notoriously difficult to perform correctly on real images in the context of pattern recognition. However, some researchers have begun to use regions in recent years for image retrieval. Dimai and Stricker [20] use a small number of 'fuzzy' regions in fixed positions to guide the computation of the image signatures; because the shapes and positions of their regions are not data-dependent, they do not obtain the benefits that regions based on segmentation could provide.

Another use of regions is that by Carson *et al.* [21], in which a small set of regions is determined for each image using expectation maximization with color and texture features. They use a scale-selection heuristic based on directionality of edges in the image, but their regions come from only one scale of segmentation. Their method is therefore susceptible to low recall due to oversegmentation and undersegmentation of the images. An important point in their paper is that the user should be presented with a view of the signature that is computed for the query image in order to have some idea of why the retrieval is working the way it does. We suggest later in our paper how additional enhancements to the user interface can let the user be even more involved in controlling the retrieval process.

3. Image Description in Terms of Regions

In order to describe our method, we must give clear definitions of images, regions, and the structures we use to describe them.

3.1. Definitions

The following definitions for image elements and for segmentation into regions are classical. However, it is important that we review them in order to clearly explain the methods for image retrieval based on regions.

An *image* is a function $f: C \to \{0, 1, \ldots, 255\}^3$ where C is a set of cells, and $\{0, 1, \ldots, 255\}^3$ is used as a three-dimensional color space. The space of cells is two-dimensional: $C = \{0, 1, \ldots, n-1\} \times \{0, 1, \ldots, m-1\}$.

A *pixel* is an element of C together with its value given by f.

A *region* is a 4-connected set of pixels.

A *uniformity predicate* $P(R)$ is a boolean function of the region R. Such a predicate is normally designed to report True when the pixels of R all have approximately the same gray value or color. However, it could be designed to evaluate the uniformity of the texture or some other local characteristic within the region R.

A *segmentation* S of an image consists of a partition of the set of pixels of the image into a set of regions $\{R_1, \ldots, R_p\}$ satisfying the following:

- The regions indeed form a partition of the set of pixels of the image; each pixel of the image occurs in precisely one of the regions.
- Each region satisfies the uniformity predicate. That is, $(\forall i)\, P(R_i)$.
- No two adjacent regions can be merged and still satisfy the uniformity predicate. That is to say, each region is maximal. This is equivalent to saying,

  $$(\forall i)(\forall j)\,[i \neq j \wedge P(R_i \cup R_j)] \rightarrow \neg\,(R_i \,\mathrm{adj}\, R_j)$$

  which says that for any distinct pair of regions R_i and R_j, if the union of these regions somehow satisfies P, then the regions are not adjacent. In other words, they cannot be merged into a new region.

This definition of segmentation is the same one that has been used in the image-understanding community for many years. In order to use it for information retrieval, we build on it to obtain a richer region structure than is traditionally used for image analysis. The new representation contains regions formed from a *succession of segmentations* of the same image. This succession of segmentations is obtained in the course of running the segmentation procedure described in the following paragraphs.

3.2. Segmentation Procedure

Here we present a segmentation procedure which has the objective of providing a collection of regions, many of which are likely to be meaningful in terms of shape, color and texture, so as to permit effective information retrieval. It is not generally necessary that any or all of the regions of the image correspond directly to identifiable objects depicted in the image.

Here is the method, in three parts.

1. If the image contains more than 128 columns, build a 'reduced' image having 128 columns. This image has lower resolution and consequently fewer pixels to process. The number 128 works well with databases of images having numbers of columns predominantly in the range of 300–600, affording a reduction in resolution to approximately 10% of the original.

   Next, segment the reduced image with a region-growing algorithm that constructs regions one pixel at a time. To do this, scan the image left-to-right, top-to-bottom, and whenever an unlabelled pixel is found, conduct a depth-first search for similar pixels; each time a pixel is added to the region being grown, recursively search those of its four nearest neighbors that have not yet been searched or labelled. A pixel is judged to be similar to those in a region if the difference between its color and the mean color of those pixels in the region is less than a threshold t_1. The difference is computed using the HSV representation of the colors. The result of this phase of processing on a sample image is illustrated in Figure 1(b).

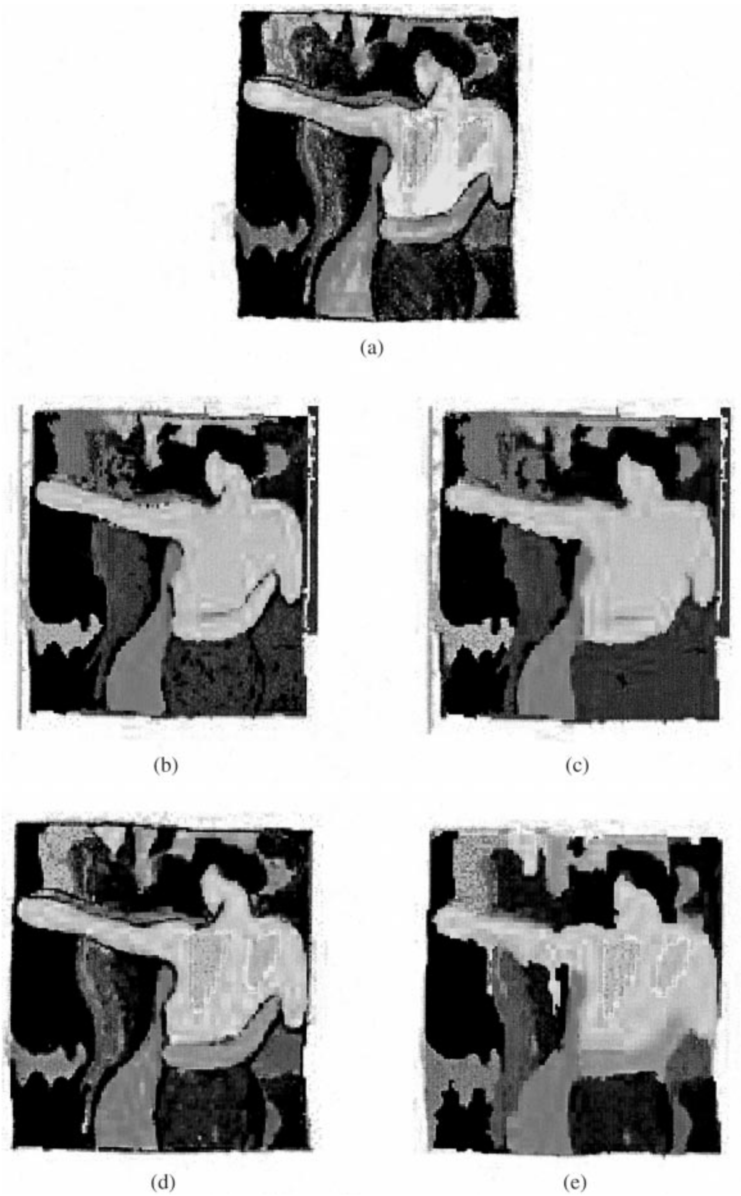


Figure 1. Steps in segmentation of an image in computing a rich-region description: **(a)** original image, **(b)** initial segmentation with liberal merging criterion, **(c)** detailed segmentation with coarse-partition criterion, **(d)–(e)** corresponding segmentation stages using a finer-partition criterion (see color page – p. 320)

2. Use the segmented image as a starting segmentation for a merging algorithm at the more detailed original level of resolution. In this phase, a region-adjacency graph is employed. There is also a region map (an image with the same dimensions as the one being analyzed, but whose pixel values are region ID values rather than

colors). One more data structure used here is a set of lists, where each list contains all the pixels currently belonging to a particular region.

Merge two adjacent regions if the difference between the mean colors of the two regions is less than a threshold t_2. After all such merges have been made, examine the size of each region, and if it is below a threshold t_3 (e.g. 15 pixels), merge the region with any of its neighbors having size above the same threshold.

3. Main loop of segmentation process repeats steps 1 and 2 with a given target number of regions to obtain. Two thresholds t_1 and t_2 are gradually increased until the number of regions obtained is less than or equal to the target number. The result of this additional processing on the sample image is shown in Figure 1(c).

This algorithm is consistent with the following strategic criteria.

1. *Work from large regions to small*: by merging very small regions into their neighboring large regions, the large regions are allowed to dominate the elaboration of the shapes in the image.
2. *Do all splitting before merging*: by following the initial segmentation by the merging, we not only bias the process towards more meaningful regions, but we avoid the possible redundancy of work that can be possible when splitting and merging processes are interleaved.
3. *Be deterministic and repeatable*: This is achieved by avoiding inherently random processes such as simulated annealing.

The result of applying this procedure once to an image is a set of regions that satisfies the definition of segmentation given above. Despite one's efforts to carefully model the features of good regions using contrast, texture, color, and shape concerns, the resulting automatic segmentations from real-world images are typically too fine or too coarse to provide the basis for a sound interpretation of the image. They may even be too fine in some areas of the image and too coarse in others. A segmentation with too many regions is known as an 'oversegmentation', whereas one with not enough regions is an 'undersegmentation'.

We can compensate for this natural tendency to over and/or undersegment the image by computing several segmentations of the image at different levels of granularity. This is accomplished simply by running the above segmentation several times on the same image, using more and more liberal merging criteria. The result is a set of segmentations

$$
\begin{aligned}
S_1 &= \{R_{1,1}, \mathrm{R}_{1,2}, \ldots, R_{1,p_1}\} \\
S_2 &= \{R_{2,1}, \mathrm{R}_{2,2}, \ldots, R_{2,p_2}\} \\
\vdots \quad &\vdots \quad \vdots \\
S_q &= \{R_{q,1}, R_{q,2}, \ldots, R_{q,p_q}\}
\end{aligned}
$$

The union of these sets forms the candidate pool of regions from which the 'rich-region description' is constructed. Note that sometimes the same region may appear in more than one segmentation. Such a region only appears once in the candidate pool, because the set union operation eliminates duplicates.

To illustrate, a second segmentation, and its intermediate stage of processing for the sample image are presented in Figure 1(d) and (e). The difference between the two segmentations is the result of different tolerances. These tolerances resulted from requesting different target numbers of regions in the segmentations.

3.3. Description in Terms of Regions

Since the candidate pool may contain many regions, a selection is made to retain only those regions with an appropriate likelihood of being useful. The selection is made according to estimates of 'importance', which is described later. The result of selection is a set of regions which is not necessarily a segmentation of the image. For example, it may cover some pixels more than once, and there may be some pixels that are not covered by any of the regions. This is not a disadvantage but actually an advantage, because this description is used for retrieval, not for reconstruction of the image.

The description for an image, termed *desc*(*I*), a set of region descriptions $\mathscr{R}_i$, where each of the regions is made up of pixels of *I*. It is not necessary that every pixel be represented in a region. Neither is it required that no pixel be represented more than once. Thus regions may overlap, and they do not necessarily cover the entire image:

$$desc(I) = \{\mathscr{R}_1, \mathscr{R}_2, \ldots, \mathscr{R}_k\}$$

$$\begin{aligned}\mathscr{R}_i = \langle x_i, y_i, &\text{ red}_i, \text{ green}_i, \text{ blue}_i,\\ &\text{color covariance}_i, \text{ perimeter}_i,\\ &\text{convexity}_i, \text{ number of bays}_i\rangle\end{aligned}$$

The object *desc*(*I*) is what we term the *rich-region description* for image *I*.

Selecting only those regions whose importance (measured as the ratio of the region's area to the total image area) exceeds 0·1, a rich-region description for the image of Figure 1 is shown in Figure 2.

4. Distance Metrics and Functions

4.1. Design Considerations

The purpose of a distance computation for image retrieval is to evaluate how closely each database image matches the user's query. The distance function should therefore compute a measure of the dissimilarity of the query and the database image. A relatively small value of this function would then indicate a relatively good match between query and database image. The design of distance functions is problematical, because query images are almost never 'close' to their intended targets, when closeness is considered in classical mathematical forms. (An example of a classical distance metric for images is the Euclidean distance between a pair of images considered as *N*-dimensional vectors,

np	*rm*	*gm*	*bm*	*x*1	*x*2	*y*1	*y*2	*bx*	*by*
91 933	128	115	129	0	325	0	358	153	180
74 038	190	160	142	0	325	0	358	168	167
24 205	222	189	111	20	309	55	262	209	159
19 697	114	97	128	58	325	5	346	249	263
17 367	218	182	93	22	309	58	262	205	152
14 941	238	205	102	20	309	58	231	213	149
8649	24	22	26	15	98	94	251	52	180
8259	21	19	22	15	98	106	242	52	181
7964	20	18	21	17	95	117	259	52	187
7242	105	93	136	76	154	128	329	112	207
7064	61	47	63	83	268	24	110	174	67
6795	152	135	163	20	266	5	110	93	51
5987	80	66	117	162	253	257	343	207	302
5794	245	244	245	0	325	0	358	121	52
5379	254	253	254	0	325	0	358	116	55
5095	204	179	213	4	95	5	358	29	210
4829	22	19	23	15	100	263	343	56	306
4759	230	116	66	99	161	226	340	133	295
4628	233	119	67	101	164	235	343	134	298
4507	88	232	232	101	161	226	340	133	294
4507	233	117	65	101	161	226	340	133	294
4484	82	44	57	124	182	145	343	155	212
3879	67	52	93	185	268	257	340	233	297
3627	113	78	124	15	92	229	340	48	267
3578	239	237	240	7	255	69	358	74	284

Figure 2. Rich-region description for the image of Figure 1. Some 25 regions from the contributing segmentations have been selected whose importance values are greatest. For reasons of space we show only 10 features per region: number of pixels; mean values of red, green and blue; bounding box and barycenter coordinates

where N is the number of pixels in each image.) The objective is to design distance functions that capture the users' notions of similarity or that satisfy the users' needs.

In order to take advantage of the results of segmentation, additional analysis of the data is usually necessary. We suggest three approaches to sorting through the regions.

1. The results of segmentation can be used to classify images into 'types'. Determine the 'type' of image(s) being compared using rules that examine the region description(s). Then use the type as a feature in matching.
   (a) If the largest region has an area smaller than 10% of the overall image area, then this is a 'small-regions image'.
   (b) Otherwise, if this image has fewer than 10 regions, it is a 'few regions' image.
   (c) Otherwise, it is a 'normal-regions image'.
2. Some regions are more likely than others to be important for matching. Order regions by decreasing size, and either eliminate all regions smaller than a fixed percentage of the image area or look for locations of major size reductions in the sequence and use that as the cutoff point.

3. Find the most distinctive regions of the image using other criteria, such as strength of contrast, distance to nearest-neighboring region in color space where each region is represented by its average color, etc.

These approaches may require additional interpretation when the time comes to implement a method, and an interpretation inevitably has consequences that may favor some kinds of applications and fail to perform as desired on others. We have chosen to implement a version of the second approach in our experiment. However, it must be kept in mind that rather than using only *one* segmentation of an image, we include regions from multiple segmentations, at differing levels of granularity, in a description of the image, and this provides a wider net in which the retrieval algorithm can snare the description of the desired image(s) in the database.

4.2. Distance Metrics on Regions

Each region R of I_1 that is well matched with a region of I_2 lowers the distance between I_1 and I_2. If all regions can be well matched, the distance is zero. The extent to which a matched pair reduces the distance depends on the *importance* of the regions making up the pair, as well as the closeness of the match.

We define the distance between two region descriptions as follows:

$$\mathscr{D}(\mathscr{R}_1, \mathscr{R}_2) = \alpha_1 d_{\text{shape}}(\mathscr{R}_1, \mathscr{R}_2) + \alpha_2 d_{\text{color\&texture}}(\mathscr{R}_1, \mathscr{R}_2) \\ + \alpha_3 d_{\text{position}}(\mathscr{R}_1, \mathscr{R}_2) + \alpha_2 d_{\text{importance}}(\mathscr{R}_1, \mathscr{R}_2)$$

4.3. Correspondences Between Sets of Regions

Let ϕ be a correspondence between a subset $\mathscr{Q}_1$ of region descriptions in *desc*(I_1) and a subset of $\mathscr{Q}_2$ of region descriptions in *desc*(I_2) such that either $\mathscr{Q}_1 = desc(I_1)$ or $\mathscr{Q}_2 = desc(I_2)$. In other words, ϕ puts as many regions into correspondence as possible, but no region is put into correspondence with more than one region in the other description. We may say, without loss of generality, that $\mathscr{Q}_1 = desc(I_1)$.

Now, we define the cost of ϕ as follows:

$$\text{Cost}(\phi) = \frac{1}{r} \sum_{\mathscr{R}_i \in \mathscr{Q}_1} \mathscr{D}(\mathscr{R}_i, \phi(\mathscr{R}_i))(\mu(R_i) + \mu(\phi(R_i)))/2$$

where $\mu(R_i)$ is the ratio of the area of R_i to that of I_1, and $\mu(\phi(R_i))$ is the ratio of the area of $\phi(R_i)$ to that of I_2. Thus, the cost of ϕ is a weighted sum of region distances, where the weights are the average relative areas of the regions involved. Here r is the 'redundancy factor' that indicates how many 'pyramid levels' are represented in *desc*(I_1) and *desc*(I_2). If only one level of regions is involved, then $r = 1$. If every region at the finest level is also a part of exactly one parent region and no other levels are involved, then $r = 2$. A simple way to compute r is to take r as the ratio of the sum of the areas of the regions in *desc*(I_1) and *desc*(I_2) to the sum of the areas of I_1 and I_2:

$$r = \frac{\sum_{\mathscr{R}_i \in \mathscr{Q}_1} area(R_i) + \sum_{\mathscr{R}_j \in \mathscr{Q}_2} area(R_j)}{area(I_1) + area(I_2)}$$

4.4. Distances Between Images

Using $Cost(\phi)$, we can now express a definition for the distance between two images based on their regions. Knowing this distance precisely requires knowing that particular correspondence ϕ which minimizes the cost

$$D(desc(I_1), desc(I_2)) = \min_{\phi} Cost(\phi)$$

Now the main difficulty in computing D, given $desc(I_1)$ and $desc(I_2)$ is finding the best correspondence ϕ between the regions of (I_1) and the regions of I_2.

4.5. Computational Considerations

To facilitate computation of D, we may make some assumptions:

1. The number of regions in $desc(I)$ is limited to some number n. For example, n could be set to 16.
2. The component distance matrices d_{shape}, etc., can be rapidly computed (in constant time) from the region descriptions.

Further computational savings may be had by using a *greedy algorithm* that does not necessarily find the optimal ϕ but finds a correspondence ϕ' likely to be almost as good as the optimal ϕ.

5. The Component Distance Metrics

5.1. Shape

The shapes of objects and contours in images are intuitively very important in human recognition of objects and retrieval of associations. A major difficulty of using shape in automatic recognition and/or retrieval is that the correctness of shape of any region depends greatly on the correctness of the segmentation. By using the rich-region description, it is possible to increase the likelihood of including a description of the correct shape of an object or contour, because a greater collection of the possible segmentations of the image is provided.

In the current study, we used only the most rudimentary representation of shape—the bounding box for a region. Using rich-region descriptions that contain more sophisticated shape descriptions is planned as future work.

5.2. Color and Texture

As in the case of shape, the colors and textures of regions depend on which pixels make up the regions, and that depends upon the segmentation. We employ separate color and texture descriptors for each of the regions in the rich-region description. (These color and texture features were used previously by Dimai and Stricker [20].)

We assume that each pixel is represented with three values: one each for red, green, and blue color components. For each region R, we compute three mean values $\mu_r(R)$, $\mu_g(R)$, and $\mu_b(R)$. Each is given by

$$\mu_i(R) = \frac{1}{N} \sum_{P \in R} C_i(P)$$

Then we compute six covariance values: σ^2_{rr}, σ^2_{rg}, σ^2_{rb}, σ^2_{gg}, σ^2_{gb}, and σ^2_{bb}. Each is given by

$$\sigma^2_{ij} = \frac{1}{N} \sum_{P \in R} [C_i(P) - \mu_i(R)][C_j(P) - \mu_j(R)]$$

These nine values give us a description of the color and the texture within the region.

5.3. Position

Position information consists of the following values for each region R: the minimum and maximum x and y coordinates (i.e. bounding box information), and the centroid of the region $(\bar{x}, \bar{y})$.

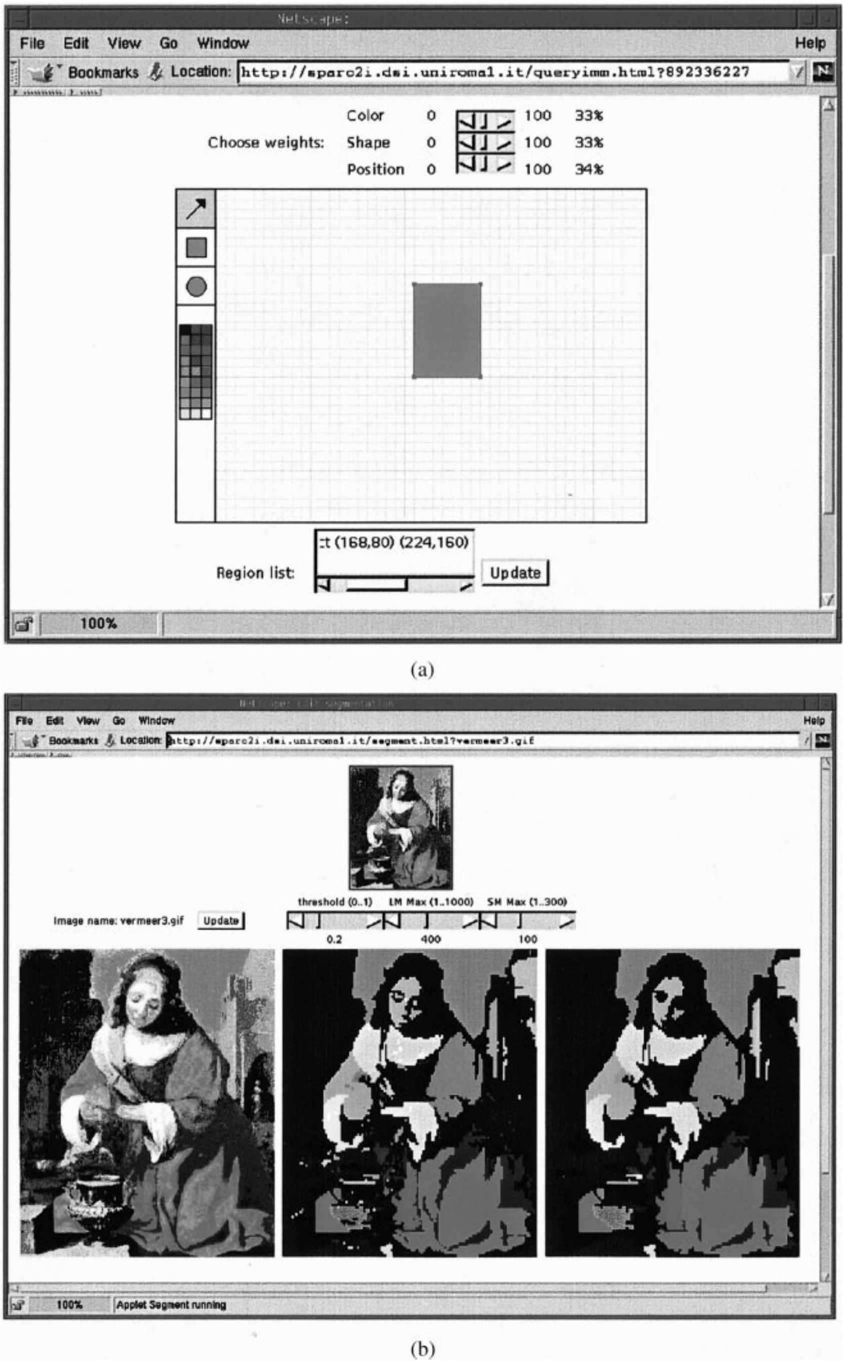
5.4. Importance

The importance of a region is a heuristically computed value that attempts to measure the relative significance of the region within its image. One way to compute this is to divide the region's area by the total area of its image. Additional factors may be taken into account as well, such as the relative contrast between this region and its neighbors, in comparison with the average level of contrast between regions and their neighbors within the image.

6. User Interface

The distance functions described in this paper are intended for use in image information retrieval systems. In a typical situation, there is a query image which the user selects or constructs. Then there are many database images which may be carefully preprocessed or may not be preprocessed at all. The user may be able to assist in the processing of the query image by aiding during the segmentation process or during the assignment of importance values to regions. In order to facilitate this interaction, the user interface should present the query image in several stages of segmentation and allow the user to select the best segmentation, adjust a segmentation, and assign importance values to regions.

When the retrieval system has selected one or more database images that match the user's query, these images should be displayed for the user together with their segmentations which have been used in the matching process. The user should be shown the best correspondences found among the regions of the database images and the query image,



Location: http://sparc2i.dsi.uniroma1.it/queryimm.html?892336227

Choose weights: Color 0 100 33%

Shape 0 100 33%

Position 0 100 34%

Region list: :t (168,80) (224,160) Update

(a)

Location: http://sparc2i.dsi.uniroma1.it/segment.html?vermeer3.gif

Image name: vermeer3.gif Update

threshold (0..1) 0.2

LM Max (1..1000) 400

SM Max (1..300) 100

(b)

Figure 3. The current web-based interface to the retrieval system: **(a)** query sketching facility, and **(b)** controls for constructing rich-region descriptions using segmentation (see color page – p. 321)

so that she/he may verify that the segmentation and retrieval processes are working as expected and desired.

We have implemented a simple web interface to our retrieval method. This interface currently supports the sketch-based query generation as well as user adjustment of the segmentation parameters used in constructing the rich-region descriptions for either database images or query images. A screen shot of this interface is shown in Figure 3. Planned improvements include means for detailed user interaction with the segmentations and rich-region descriptions.

7. Experimental Results

We have implemented a matching algorithm that computes an approximation of the region-based inter-image distance function described above. The algorithm takes as input a pair of rich-region descriptions and produces as output a numeric distance value. It begins by building a set of two-dimensional matrices whose elements are region–region distance values for component distance functions. These matrices are then weighted and added to obtain region-to-region values for the combined criteria.

Next, the columns of the new matrix are scanned, and the minimal value in each column is determined, so as to pair the regions of one image with those of the other. The columns are assumed to be sorted in order of region importance, and this assumption that the assignments are made in order of region importance justifies the greedy, no-backtracking implementation.

For our experiments, a small database of 15 images of paintings was used. For each image, a rich–region description was computed using either two or three segmentations. For some images, the thresholds were adjusted manually to obtain a diversity of regions in the resulting segmentations. It was found that fixed thresholds produced good results for many images but some need human attention in order to have good rich-region descriptions.

Queries were generated in two ways. In one way, a sketching program was used to obtain a small list of geometrically shaped regions from the user. This list was used as if it were a rich-region description obtained through a segmentation process. In the other queries, an actual image from the database was used; its rich-region description was compared, using the region-based distance function, with the rich-region descriptions of the other images in the database.

The result of processing a query is a list of images from the database, sorted in order of increasing distance from the query.

In order to demonstrate the value of the rich-region description method, we processed each of two queries in two ways—first using the rich-region description and second using a similar region description but based on only a single segmentation of the image. Both queries consisted of one rectangular region. In the first query, a small green region was used; it was intended to match one of the green shadows on the back of the dancer in the 'Tango' image (Figure 1). The other query used a large yellow region intended to match the entire back of the dancer.

Using the rich-region description a user was able to retrieve the 'Tango' image (Figure 1 with either of the two queries. On the other hand, using a region description

Query	Method	Database	Result	Value
Green rect.	Single Seg.	Full	Vermeer1	0·169
Green rect.	RRD	Full	Tango	0·132
Yellow rect.	Single Seg.	Full	Tango	0·064
Yellow rect.	RRD	Full	Tango	0·054
Tango	Single seg.	Full	Tango	0·000
Tango	RRD	Full	Tango	0·000
Vermeer1	RRD	¬ Vermeer	Vermeer2	0·004
Vermeer2	RRD	¬ Tango	Vermeer1	0·004

Figure 4. Results of query processing

based on only a single segmentation, the query failed in one case. In the failing case, the image deemed closest to the querry is the Vermeer painting shown in Figure 3.

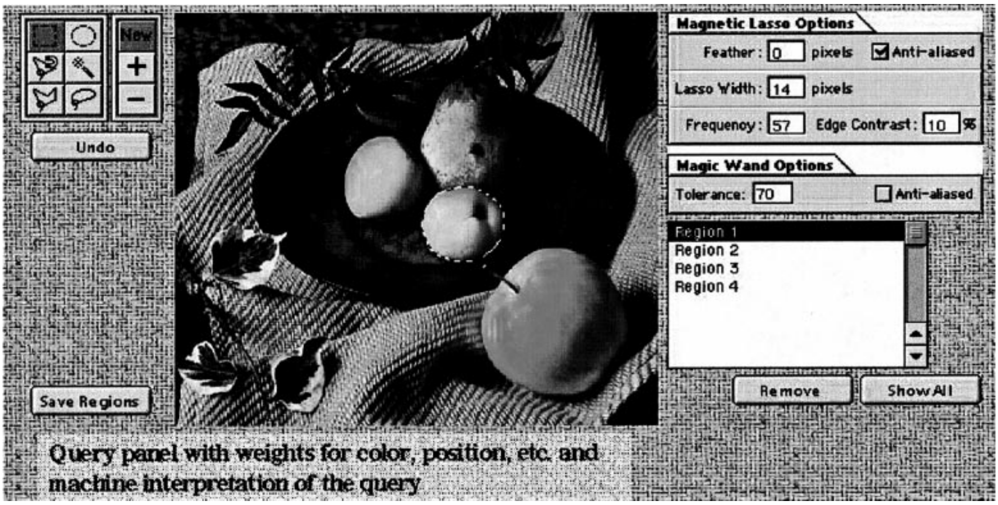
The results for a variety of queries and conditions are presented in Figure 4. The green rectangle and yellow rectangle queries were generated with the sketching interface, while the Tango, Vermeer1 and Vermeer2 images were used as queries in the remaining cases. Methods used include rich-region description (RRD) and description using single segmentations. The full database consisted of 15 images of paintings whereas the database ¬Vermeer1 consisted of the 14 images other than the Vermeer1 image. Similarly, the ¬ Vermeer2 database excluded the Vermeer2 image. In the value column are the scores for the best matching images for each query condition. A score of 0·0 is the lowest possible value. As can be seen, both RRD and single segmentation methods correctly retrieved an image A from the database when A itself is the query. When Vermeer1 is the query but is not in the database, then Vermeer2 is retrieved, and *vice versa*; these two images are relatively similar in terms of the region-based distance function.

8. Manual Segmentation for Storing and Querying Regions

The automatic segmentation procedure previously described constitutes only one possible method for building the rich-region description of an image. Automatic segmentation by computers poses several problems (for example, the selection of the initial threshold value for certain algorithms, and sometimes the results can be poor. Human beings instead are pretty good at segmenting images and finding zones of interest, and the best of both worlds can be obtained by providing users with good tools for segmentation.

Here we describe a user interface (Figure 5) for specifying regions, which will then be indexed and stored in the rich description. Our current system can handle region data regardless of the method used for creating them. Multiple users (or single users in different sessions) interactively segmenting the same image contribute to build the rich-region description for that image.

People dealing with images are most probably familiar with tools for editing them, so the best way to design a user interface for segmentation is to provide tools users are already familiar with; for example, the image selection tools that are common in



Feather: 0 pixels ☑ Anti-aliased

Lasso Width: 14 pixels

Frequency: 57 Edge Contrast: 10 %

Tolerance: 70 □ Anti-aliased

Region 1
Region 2
Region 3
Region 4

Query panel with weights for color, position, etc. and machine interpretation of the query

Figure 5. Proposed GUI for building rich-region descriptions (see color page – p. 320)

commercial programs like Adobe Photoshop: the rectangle or oval selection tool, the 'lasso', the 'magnetic lasso', and the 'magic wand' (see the tool bar in Figure 5).

The use of the first two tools is almost obvious; the lasso allows the user to draw with the mouse the contour of the desired region; the magnetic lasso uses gradient information to 'adjust' the mouse to follow the borders of object, resulting in a very satisfying experience of region selection. The magic wand uses a region growing method with a configurable tolerance: the seed point is selected by the user with the mouse.

Each of these tools has its own parameters which can be customized by means of an option panel. Moreover, a tool modifier is used to determine the behavior of the tools the next time they will be used: whether they will add to the current region, subtract from the current region, or just start a new region.

Finally, we have: a button for undoing the last action, the list of all the regions built so far, and a button for saving them.

In a typical work session, the user would retrieve an image and the associated list of regions. Then, she/he would select an existing region to be modified, or start a new one using the tools. Then, using the + or − modifiers, the region can be shaped in detail by including and excluding areas with many different methods: typically, the user will want to start a region with the magic wand, then he would add or subtract regions with different tools, and in the end take care of the details of the borders with the magnetic lasso.

A possible improvement might include developing a method whereby the computer can learn how to do better automatic segmentations, by analyzing the user's interactive segmentations.

Once the user has gained experience in the use of the tools for segmentation, it would be important to integrate them with the query process: the proposed interface can simply replace the current query-by-sketch applet in our system.

Various query-composition strategies are possible: a possible one is the fully automatic construction of a query object (rich-region description) from the user's image;

otherwise, a query object can be built by the user with the proposed tools (with additional widgets for selecting the color of the desired regions, and with weights for shape, color and position). Starting a query will give a list of results, and clicking on one result image, the user will go back to the query interface with the new image: such a process can be a starting activity for a longer interactive query process, where the results are considered feedback for making another attempt.

9. Conclusions

Retrieval of images from databases using their visual features is a challenging and important problem. One of the most difficult visual features to take properly into consideration is shape, because the shapes present in an image depend on subjective interpretations and can vary from user to user. We have presented a technique to improve recall in region-based retrieval; the method is based upon a family of representations of images called 'rich-region descriptions'. We have shown in a simple experiment how this kind of representation can improve the flexibility allowed to users in obtaining desired results. We have also discussed issues related to the user interface for such query systems.

References

1. G. Salton (1989) *Automatic Text Processing*. Addison-Wesley, Reading, MA.
2. W. Niblack *et al.* (1993) The QBIC project: querying images by content using color, texture, and shape. *Proceedings Storage and Retrieval for Image and Video Databases*, Vol. 1, **908,** SPIE, Bellingham, WA, pp. 173–187.
3. M. Flickner *et al.* (1995) Query by image and video content: the QBIC system. Special issue on content-based image retrieval systems. *Computer* **28,** 23–32.
4. S. Santini & R. Jain (1996) Similarity matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21,** 871–883.
5. G. Bordogna, I. Gagliardi, D. Merelli, P. Mussio, M. Padula & M. Protti (1989) Iconic queries on pictorial data. *Proceedings of the IEEE Workshop on Visual Languages*, pp. 38–42.
6. A. Califano & R. Mohar (1994) Multidimensional indexing for recognizing visual shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **4,** 373–392.
7. C. C. Chang & S. Y. Lee (1991) Retrieval of similar pictures in pictorial databases. *Pattern Recognition* **7,** 675–680.
8. S.-K. Chang & C. C. Yang (1982) Picture information measures for similarity retrieval. *Computer Vision*, *Graphics and Image Processing* **23,** 366–375.
9. D. Daneels *et al.* (1993) Interactive outlining: An improved approach using active contours. In: *Proceedings of the SPIE*, *Image and Video Storage and Retrieval* **1908,** 226–233.
10. A. Del Bimbo, P. Pala & S. Santini (1994) Visual image retrieval by elastic deformation of object sketches. In: *Proceedings of the IEEE Symposium on Visual Languages*, St. Louis (Missouri), pp. 216–223.
11. F. Fierens, J. Van Cleynenbreugel, P. Suetens & A. Oosterlinck (1992) A software environment for image database research. *Journal of Visual Languages and Computing* **3,** 49–68.
12. V. N. Gudivada & V. V. Raghavan (1995) Design and evaluation of algorithms for image retrieval by spatial similarity. *ACM Transactions on Information Systems* **13,** 115–144.
13. V. N. Gudivada & V. V. Raghavan (Guest Editors) (1995) Special issue on content-based image retrieval systems. *Computer* **28,** 18–62.

14. C. E. Jacobs, A. Finkelstein & D. H. Salesin (1995) Fast multiresolution image querying. Technical Report 95-01-06, Department of Computer Science and Engineering. University of Washington, Seattle, WA.
15. T. Kato, T. Turita, N. Otsu & K. Hirata (1992) A sketch retrieval method for full color image database. In: *Proceedings of the 11th International Conference on Pattern Recognition*, The Hegue (Netherlands), pp. 530–533.
16. S. Y. Lee, M. K. Shan & W. P. Yang (1989) Similarity retrieval of iconic image database. *Pattern Recognition* **6,** 675–682.
17. A. Pentland, R. W. Picard & S. Sclaroff (1994) Photobook: tools for content-based manipulation of image databases. Media Lab Technical Report **255,** MIT, Cambridge, MA.
18. L. Cinque *et al.* (1997) Survey of image retrieval. *Image and Vision Computing* **15,** 119−141.
19. S. L. Tanimoto (1976) An iconic/symbolic structuring scheme. In: *Pattern Recognition and Artificial Intelligence*, (C. H. Chen, ed.), Academic Press, Orlando, FL, pp. 452–471.
20. A. Dimai & M. Stricker (1996) Spectral covariance and fuzzy regions for image indexing. Communications Technology Lab Technical Report BIWI-TR-173, Swiss Federal Institute of Technology, ETH, Zurich, Switzerland.
21. C. Carson, S. Belongie, H. Greenspan & J. Malik (1997) Region-based image querying. In: *Proceedings of the CVPR'97 Workshop on Content-Based Access of Image and Video Libraries*, IEEE Computer Society, Alameda, CA.

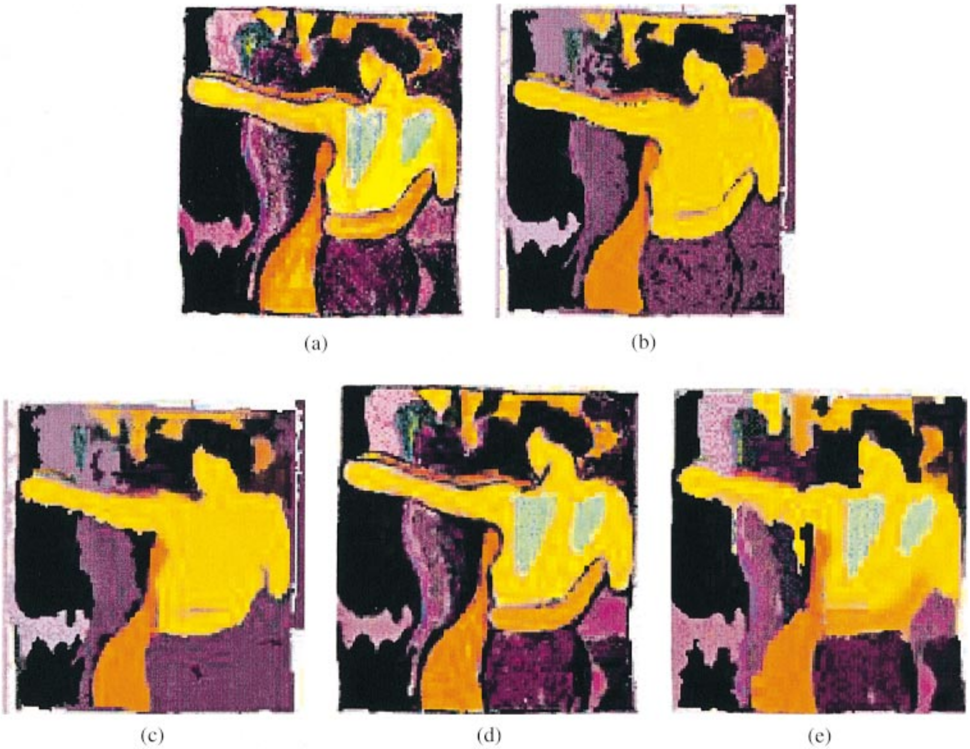
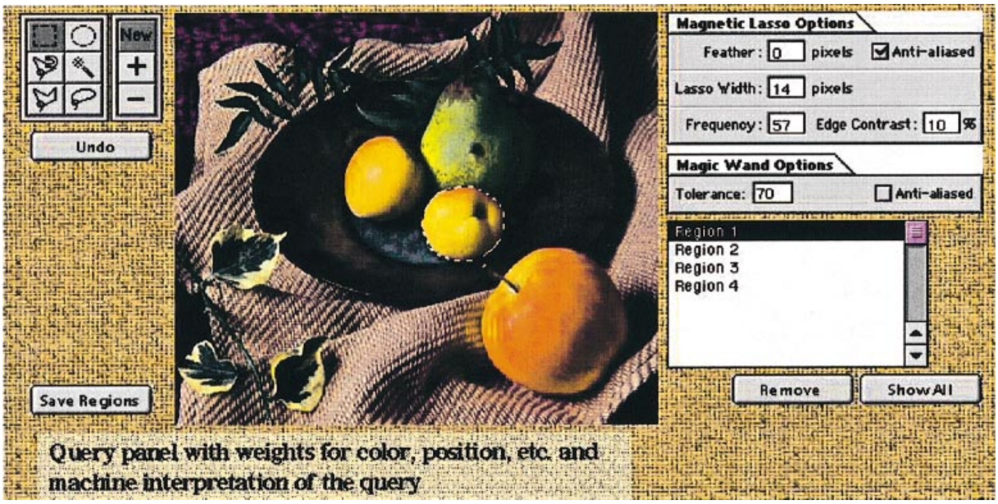


Figure 1. (see p. 307)



Magnetic Lasso Options

Feather: 0 pixels ☑ Anti-aliased

Lasso Width: 14 pixels

Frequency: 57 Edge Contrast: 10 %

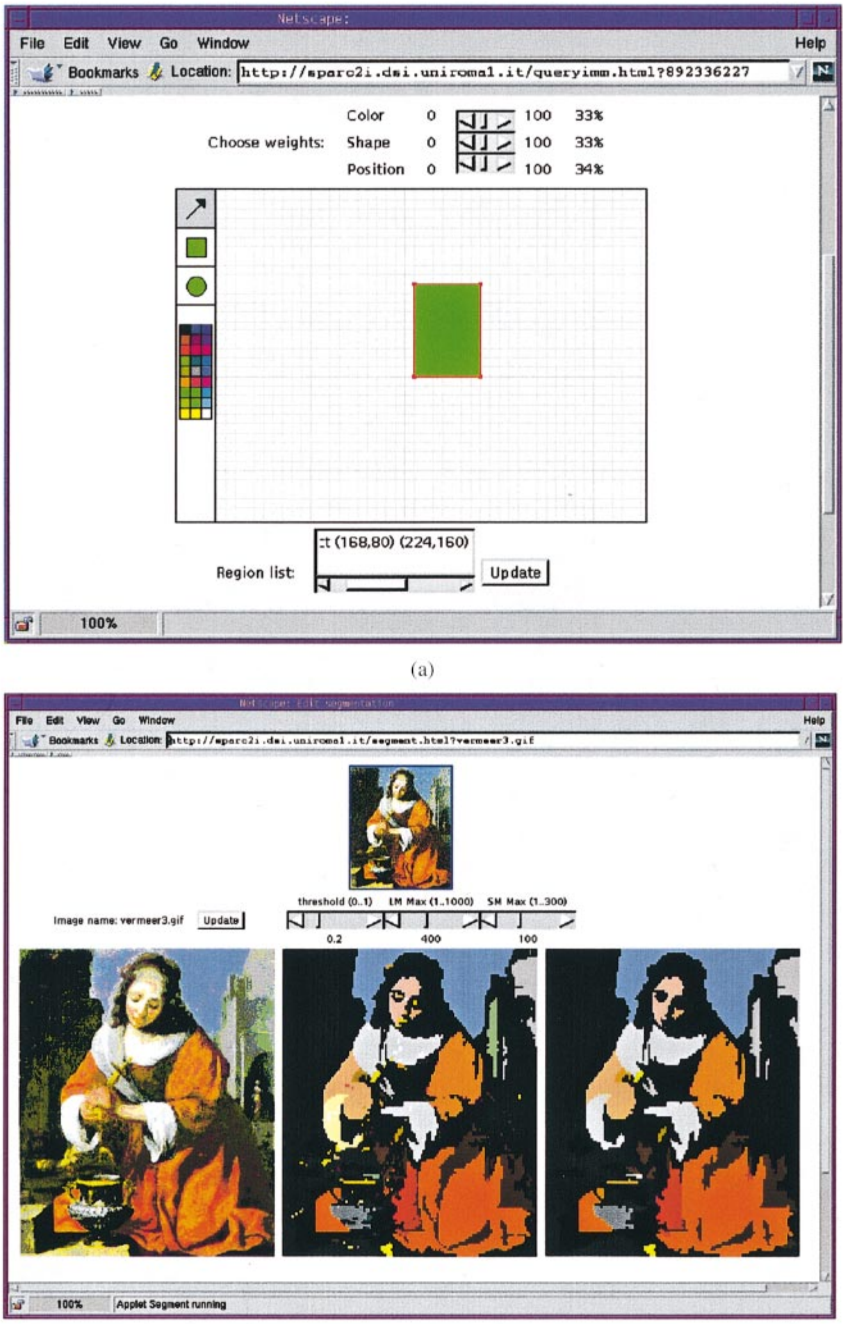
Magic Wand Options

Tolerance: 70 □ Anti-aliased

Region 1
Region 2
Region 3
Region 4

Query panel with weights for color, position, etc. and machine interpretation of the query

Figure 5. (see p. 317)



Netscape:
File Edit View Go Window Help
Bookmarks
Update
100%

Location: http://sparc2i.dsi.uniroma1.it/queryimm.html?892336227

Choose weights:

Color	0		100	33%
Shape	0		100	33%
Position	0		100	34%

Region list: :t (168,80) (224,160)

(a)

Netscape: Edit segmentation
File Edit View Go Window Help
Bookmarks
Update
100%
Applet Segment running

Location: http://sparc2i.dsi.uniroma1.it/segment.html?vermeer3.gif

Image name: vermeer3.gif

threshold (0..1)	LM Max (1..1000)	SM Max (1..300)
0.2	400	100

(b)

Figure 3. (see p. 314)