SEGMENTATION AND CONTENT-BASED WATERMARKING FOR IMAGE INDEXING

V. Mezaris, N. V. Boulgouris, I. Kompatsiaris, D. Simitopoulos, and M. G. Strintzis

Information Processing Laboratory
Electrical and Computer Engineering Dept.
Aristotle University of Thessaloniki
Thessaloniki 54006, Greece

Informatics and Telematics Institute
1st Km Thermi-Panorama Rd,
Thessaloniki 57001, Greece
e-mail: strintzi@eng.auth.gr

ABSTRACT

In this paper, a novel approach to image indexing using content-based watermarking is presented. The proposed system uses color image segmentation and watermarking in order to facilitate content-based indexing, retrieval and manipulation of digital images and image regions. A novel segmentation algorithm is applied on reduced images and the resulting segmentation mask is embedded in the image using watermarking techniques. In each region of the image, indexing information is additionally embedded. In this way, the proposed system is endowed with content-based access and indexing capabilities which can be easily exploited via a simple watermark detection process. Several experiments have shown the potential of this approach.

Keywords: *image segmentation; watermarking; information hiding.*

1. INTRODUCTION

In recent years, the proliferation of digital media has established the need for the development of tools for the efficient access and retrieval of visual information. At the same time, watermarking has received significant attention due to its applications on the protection of intellectual property rights (IPR) [1]. However, many other applications can be conceived which involve information hiding [2, 3]. In this paper, we propose the employment of watermarking as a means to content-based indexing and retrieval of images from data bases.

In order to endow the proposed scheme with content-based functionalities, information must be hidden region-wise in digital images. Thus, the success of any content-based approach depends largely on the segmentation of the image based on its content. In this paper, a region-based approach to segmentation is presented. The segmentation of the image into regions is followed by the estimation of a set of region descriptors for each region; these serve as indexing information.

The segmentation and indexing information are subsequently embedded into the images using digital watermarking techniques. In this way, both segmentation and indexing information can be easily extracted using a fast watermark detection procedure. Embedding segmentation and indexing information in image regions [4, 5] has the following advantages:

- Each region in the image carries its own description and no additional information must be kept for its description.
- The image can be moved from a database to another without the need to move any associated description.
- Objects can be cropped from images without the requirement for employing a segmentation algorithm.

The paper is organized as follows: the overview of the proposed system is given in section 2. The segmentation algorithm is presented in section 3. In section 4, the derivation of region descriptors used for indexing is described. The information embedding process is shown in section 5. In section 6, experimental evaluation is discussed, and finally, conclusions are drawn in section 7.

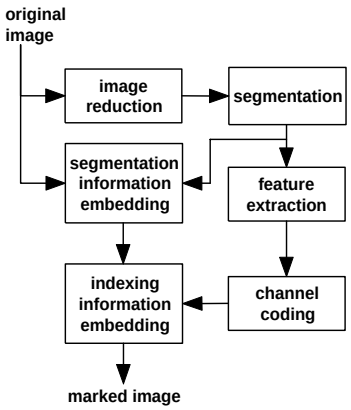

Fig. 1. Block diagram of the embedding scheme.

2. SYSTEM OVERVIEW

The block diagram of the proposed system is shown in Fig. 1. The system first segments an image into objects using a segmentation algorithm that forms connected regions. The segmentation algorithm is applied to a reduced image consisting of the mean values of the pixel intensities in 8×8 blocks of the original image. Apart from speeding the segmentation process, this approach has the additional advantage that it yields image regions

comprising a number of 8×8 blocks (since a single pixel in the reduced image corresponds to a whole block in the original image). Following segmentation, watermarking can proceed immediately. Unlike segmentation, the watermarking process is applied to the full resolution image. Specifically, the segmentation information is embedded first. The indexing information is obtained from the reduced image and the indexing bits are channel coded and then embedded in the full resolution image.

Conversely, the first step in the watermark detection process is to detect the segmentation watermark and subsequently, based on this segmentation, to extract the information bits associated with each object (see Fig. 2). If, due to unsuccessful watermark detection, the segmentation mask detected at the decoder is different than the one used at the encoder, then the detection process will not be synchronized with the embedding process and the embedded indexing information will not be retrieved correctly. To alleviate this problem, a dummy detection of the embedded segmentation information takes place at the encoder prior to the embedding of any indexing information.

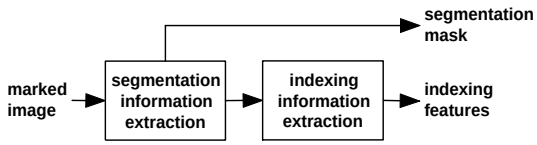

Fig. 2. Block diagram of the detection scheme.

3. COLOR IMAGE SEGMENTATION

3.1. Segmentation System Overview

The segmentation method is based on the K-Means-with-connectivity-constraint algorithm (KMCC) [6], a variant of the popular K-Means algorithm. The KMCC algorithm classifies the pixels into regions taking into account not only the intensity or texture information associated with each pixel but also the position of the pixel, thus producing connected regions rather than sets of chromatically similar pixels. Furthermore, the combination of intensity and texture information enables the algorithm to handle textured objects effectively, by forming large, chromatically non-uniform regions instead of breaking down the objects to a large number of chromatically uniform regions.

The segmentation algorithm consists of the following stages (Fig. 3):

- Stage 1. Extraction of the intensity and texture feature vectors corresponding to each pixel. These will be used along with the spatial features in the following stages.
- Stage 2. Estimation of the initial number of regions and their spatial, intensity and texture centers of the KMCC algorithm.
- Stage 3. Conditional filtering using a moving average filter.
- Stage 4. Final pixel classification using the KMCC algorithm.

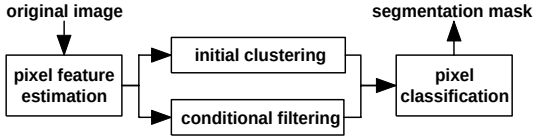

Fig. 3. Overview of the segmentation algorithm.

3.2. Color and Texture Features

The color features used are the three intensity coordinates of the CIE L*a*b* color space. This color space is related to the CIE XYZ standard through a non-linear transformation. What makes CIE L*a*b* more suitable for the proposed algorithm than the widely used RGB color space is perceptual uniformity: the CIE L*a*b* is approximately perceptually uniform, i.e. the numerical distance in this color space is approximately proportional to the perceived color difference [7]. The color feature vector of pixel $\mathbf{p} = [p_x \; p_y]^T$, $\mathbf{I}(\mathbf{p})$ is defined as

$$\mathbf{I}(\mathbf{p}) = [I_L(\mathbf{p}) \; I_a(\mathbf{p}) \; I_b(\mathbf{p})]^T$$

In order to detect and characterize texture properties in the neighborhood of each pixel, the Discrete Wavelet Frames (DWF) decomposition [8] is used. This is a similar to the Discrete Wavelet Transform (DWT) method, that employs a filter bank based on a lowpass filter $H(z)$ to decompose each intensity component of the image to a set of subbands and uses the standard deviations of all detail components calculated in a neighborhood F of pixel $\mathbf{p}$ to characterize its texture. The filter bank used in this work is based on the lowpass Haar filter

$$H(z) = \frac{1}{2}(1 + z^{-1}). \qquad (1)$$

Performing a two-dimensional DWF decomposition of two levels results in an 18-component texture feature vector estimated for pixel $\mathbf{p}$:

$$\mathbf{T}(\mathbf{p}) = [\sigma_1(\mathbf{p}) \; \sigma_2(\mathbf{p}) \; \ldots \; \sigma_{18}(\mathbf{p})]^T$$

3.3. Initial Clustering

In order to compute the initial values required by the KMCC algorithm, the image is broken down to square, non-overlapping blocks of dimension $f \times f$. In this way, a reduced image composed of a total of L blocks, $b_l, l = 1, \ldots, L$, is created. Let the center of block b_l be pixel $\mathbf{p}^l_{cntr}$. A color feature vector $\mathbf{I}(b_l)$ and a texture feature vector $\mathbf{T}(b_l)$ are then assigned to each block, as follows:

$$\mathbf{I}(b_l) = \frac{1}{f^2} \sum_{\mathbf{p} \in b_l} \mathbf{I}(\mathbf{p}), \;\; \mathbf{T}(b_l) = \mathbf{T}(\mathbf{p}^l_{cntr}), \qquad (2)$$

The distance between two blocks is defined as follows:

$$D(b_{l_1}, b_{l_2}) = \|\mathbf{I}(b_{l_1}) - \mathbf{I}(b_{l_2})\| + \|\mathbf{T}(b_{l_1}) - \mathbf{T}(b_{l_2})\|, \quad (3)$$

where $\|\mathbf{I}(b_{l_1}) - \mathbf{I}(b_{l_2})\|$, $\|\mathbf{T}(b_{l_1}) - \mathbf{T}(b_{l_2})\|$ are the Euclidean distances of the intensity and texture feature vectors.

The number of regions of the image is initially estimated by applying a variant of the maximin algorithm to this set of blocks. In the employed variant, the intensity and texture distance Db_{max} between the first two centers is calculated and candidate centers are accepted as region centers until the minimum distance between the candidate center and a region center is lower than $\gamma \cdot Db_{max}$, where $\gamma = 0.4$.

In order to obtain an estimate of the spatial centers of these regions, a simple K-Means algorithm is applied to the set of blocks, using the information produced by the maximin algorithm for its initialization. This is followed by the application of a recursive four-connectivity component labelling algorithm, so that a total of K' connected regions are identified. Their intensity, texture and spatial centers $\mathbf{I}_k$, $\mathbf{T}_k$, $\mathbf{S}_k$, $k = 1, \ldots, K'$, are then calculated as the mean values of the intensity, texture and position features of the pixels belonging to the blocks assigned to each region.

3.4. Conditional Filtering

Images may contain parts in which intensity fluctuations are particularly pronounced, even when all pixels in these parts of the image belong to a single object (figure 6). In order to facilitate the grouping of all these pixels in a single region based on their texture similarity, their intensity differences are reduced by conditionally applying a moving average filter.

The decision of whether the filter should be applied to a particular pixel $\mathbf{p}$ or not is made by evaluating the norm of the texture feature vector $\mathbf{T}(\mathbf{p})$; the filter is not applied if that norm is below a threshold T_{th}. The output of the conditional filtering module can thus be expressed as:

$$\mathbf{J}(\mathbf{p}) = \begin{cases} \mathbf{I}(\mathbf{p}) & \text{if } \|\mathbf{T}(\mathbf{p})\| < T_{th} \\ \frac{1}{f^2}\sum_{m=1}^{f^2} \mathbf{I}(\mathbf{p}_m) & \text{if } \|\mathbf{T}(\mathbf{p})\| \geq T_{th} \end{cases} \quad (4)$$

$$T_{th} = \max\{0.65 \cdot T_{max}, 14\} \quad (5)$$

where T_{max} is the maximum value of the norm $\|\mathbf{T}(\mathbf{p})\|$ in the image. The intensity center of region s_k calculated from the filtered intensity features is denoted $\mathbf{J}_k$.

The output of the conditional filtering stage is used as input by the KMCC algorithm.

3.5. The K-Means with Connectivity Constraint Algorithm

Clustering based on the K-Means algorithm is a widely used region segmentation method [9] which, however tends to produce unconnected regions. This is due to the propensity of the classical K-Means algorithm to ignore spatial information about the intensity values in an image, since it only takes into account the global intensity or color information. In order to alleviate this problem, the K-Means-with-connectivity-constraint algorithm was proposed [6]. In this algorithm the *spatial proximity* of each region is also taken into account by defining a new center for the K-Means algorithm and by integrating the K-Means with a component labeling procedure.

In this work, pixels are classified into regions by a variant of the KMCC algorithm, using a distance function of a pixel $\mathbf{p}$ from a region s_k defined as:

$$D(\mathbf{p}, s_k) = \|\mathbf{J}(\mathbf{p}) - \mathbf{J}_k\| + \\ + \|\mathbf{T}(\mathbf{p}) - \mathbf{T}_k\| + \lambda \frac{\bar{A}}{A_k} \|\mathbf{p} - \mathbf{S}_k\|$$

where $\|\mathbf{J}(\mathbf{p}) - \mathbf{J}_k\|$, $\|\mathbf{T}(\mathbf{p}) - \mathbf{T}_k\|$ and $\|\mathbf{p} - \mathbf{S}_k\|$ are the Euclidean distances of the intensity, texture and spatial feature vectors respectively, A_k is the area of region s_k, $\bar{A}$ is the average area of all regions and λ is a regularization parameter. The KMCC algorithm features splitting of non-connected regions and merging of neighboring regions with similar intensity or texture centers. The region centers are recalculated on every iteration as the mean values of the intensity, texture and spatial features of the pixels assigned to each region. Centers corresponding to regions that fall below a size threshold $th_{size} = 0.75\%$ of the image area, are omitted.

4. REGION DESCRIPTORS

As soon as the segmentation mask is produced, a set of descriptors that will be used for querying are calculated for each region. These descriptors compactly characterize each region's color, position and shape.

The **color and position descriptors** of a region are based on the intensity and spatial centers that were calculated for the region in the last iteration of the KMCC algorithm. In particular, the color descriptors of region s_k are its intensity centers, $\mathbf{I}_k = [I_{k,L}\ I_{k,a}\ I_{k,b}]$, whereas the position descriptors $P_{k,x}, P_{k,y}$ are the spatial centers normalized by the dimensions of the image.

The **shape descriptors** of a region are its area, eccentricity and orientation. The area E_k is expressed by the number of pixels M_k that belong to region s_k, divided by the total number of pixels of the image:

$$E_k = \frac{M_k}{x_{max} \cdot y_{max}}$$

The other two shape descriptors are calculated using the covariance or scatter matrix $\mathbf{C}_k$ of the region. This is defined as:

$$\mathbf{C}_k = \frac{1}{M_k} \sum_{\mathbf{p} \in s_k} (\mathbf{p}_m^k - \mathbf{S}_k)(\mathbf{p}_m^k - \mathbf{S}_k)^T .$$

Let ρ_i, $\mathbf{u}_i, i = 1, 2$ be its eigenvalues and eigenvectors: $\mathbf{C}_k\, \mathbf{u}_i = \rho_i\, \mathbf{u}_i$ with $\mathbf{u}_i^T\, \mathbf{u}_i = 1$, $\mathbf{u}_i^T\, \mathbf{u}_j = 0$, $i \neq j$ and $\rho_1 \geq \rho_2$. As is known from Principal Component

Analysis (PCA), the principal eigenvector $\mathbf{u}_1$ defines the orientation of the region and $\mathbf{u}_2$ is perpendicular to $\mathbf{u}_1$. The two eigenvalues provide an approximate measure of the two dominant directions of the shape. Using these quantities, an approximation of the eccentricity ε_k and orientation θ_k of the region are calculated: orientation θ_k is the argument of the principal eigenvector of $\mathbf{C}_k$, $\mathbf{u}_1$, and eccentricity ε_k is defined as follows:

$$\varepsilon_k = 1 - \frac{\rho_1}{\rho_2}$$

The eight region descriptors mentioned above form a region descriptor vector $\mathbf{D}_k$:

$$\mathbf{D}_k = [I_{k,L}\ I_{k,a}\ I_{k,b}\ P_{k,x}\ P_{k,y}\ E_k\ \theta_k\ \varepsilon_k]$$

Using eight bits to express each one of the region descriptors, a total of 64 bits is required for the entire region descriptor vector $\mathbf{D}_k$. This information, along with the segmentation mask, will be embedded in the image using digital watermarking techniques, as described in the ensuing section.

5. CONTENT-BASED INFORMATION EMBEDDING

The information obtained for each image using the techniques of the preceding sections is embedded in the image itself. Two kinds of watermarks are embedded; one containing segmentation information and another that carries indexing information. Both are embedded in the spatial domain.

5.1. Segmentation Information Embedding

The segmentation watermark consists of a number of symbols. A different symbol is embedded in each image region in order to make possible the identification of the regions during the watermark detection process. In a sense, the segmentation watermark symbols can be seen as labels that are tagged to each image object (see fig. 4). We chose to embed the segmentation watermark in the Blue component of the RGB images because the Human Visual System is less sensitive to blue color [10]. Even though the methodology that will be developed is entirely general and an arbitrary number of regions may be labelled, we shall describe it here for the sake of brevity and simplicity for the case where only four regions of each segmented image need be labelled i.e. three main objects and the background. It will be assumed that any other objects determined using the segmentation algorithm are small and insignificant objects which are labelled using the same label as the background and therefore cannot be indexed and retrieved independently.

The pixels of block (l_1, l_2) in the Blue component I_B of the image will be modified due to the watermarking process as follows:

$$I'_{(l_1,l_2)B}[i,j] = I_B[8l_1 + i, 8l_2 + j] + a_{l_1,l_2} \cdot w[i,j] \quad (6)$$

where l_1, l_2 are the block indices, i, j are the indices specifying the position of the pixel inside the block and

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	0	0	2	2	2	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	0	2	2	2	2	2	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0	2	2	2	2	2	0	0	0
0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	0	0
0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	0	0
0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	2	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	3	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0
0	0	0	3	3	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0
0	0	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0	0	0
0	0	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0	0	0
0	0	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0	0	0
0	0	0	3	3	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 4. Example of embedding labels to each image block.

a_{l_1,l_2} is a modulating factor valued as follows

$$a_{l_1,l_2} = \begin{cases} -3, & \text{if the block } (l_1,l_2) \text{ belongs to the background or is not classified} \\ -1, & \text{if the block } (l_1,l_2) \text{ belongs to the 1st region} \\ +1, & \text{if the block } (l_1,l_2) \text{ belongs to the 2nd region} \\ +3, & \text{if the block } (l_1,l_2) \text{ belongs to the 3rd region} \end{cases} \quad (7)$$

and $w[i,j]$ is the watermark matrix (see fig. 5), which is the same for all blocks in the image and is given by $w[i,j] = (-1)^{i+j}$, i.e.

$$w[i,j] = \begin{cases} 1, & \text{if } i+j = \text{even} \\ -1, & \text{if } i+j = \text{odd} \end{cases} \quad (8)$$

+1	-1	+1	-1	+1	-1	+1	-1
-1	+1	-1	1+	-1	+1	-1	+1
+1	-1	+1	-1	+1	-1	+1	-1
-1	+1	-1	+1	-1	+1	-1	+1
+1	-1	+1	-1	+1	-1	+1	-1
-1	+1	-1	+1	-1	+1	-1	+1
+1	-1	+1	-1	+1	-1	+1	-1
-1	+1	-1	+1	-1	+1	-1	+1

Fig. 5. Watermark matrix. The modulating factor corresponding to the embedded information symbol is multiplied by the matrix elements and the resulting signal is added to the image.

The extraction of the label of each block is achieved via a simple correlation-based watermark detection process. The watermark detector is applied to each block. To detect the watermark, we calculate the correlation between the intensities $I'_{(l_1,l_2)B}[i,j]$ of the watermarked pixels in a block and $w[i,j]$. The detector output for block (l_1,l_2) is calculated by

$$q_{l_1,l_2} = \frac{1}{N} \sum_i \sum_j I'_{(l_1,l_2)B}[i,j] \cdot w[i,j] \quad (9)$$

where N is the number of pixels in a block. In our case $N = 64$ since 64 pixels are included in a 8×8 block. The symbol that is extracted from each block depends on the detector output. The probability density function of the detector output can be approximated by a gaussian distribution with mean equal to -3, -1, 1, or 3, depending on the symbol that was embedded. In this case, the optimal rule for extracting the label of block (l_1, l_2) is

$$s = \begin{cases} 0, & \text{if } q_{l_1,l_2} < -2 \\ 1, & \text{if } -2 < q_{l_1,l_2} < 0 \\ 2, & \text{if } 0 < q_{l_1,l_2} < 2 \\ 3, & \text{if } 2 < q_{l_1,l_2} \end{cases} \quad (10)$$

since the above choice minimizes the probability of erroneous symbol detection. Although this is a small probability, there are some cases in which even such a small error could affect the synchronization capability of the system and the subsequent indexing information extraction. Such a case may occur if a block on region boundaries is misinterpreted. For this reason, immediately before embedding indexing information, a dummy detection of segmentation information takes place in order to identify blocks which yield ambiguous segmentation labels. In such blocks, no indexing information is embedded.

5.2. Indexing Information Embedding

Indexing information is embedded in the Red component of each image using binary symbols. For each region, eight feature values described by 8 bits each are ordered in a binary vector of 64 bits. Each bit of this vector is embedded in a block of the corresponding region. After the embedding of the watermark, the Red component of the block (l_1, l_2) of the image is as follows:

$$I'_{(l_1,l_2)R}[i,j] = I_R[8l_1 + i, 8l_2 + j] + a_{l_1,l_2} \cdot w[i,j]$$

where w is the watermark matrix given in eq. (8) and a_{l_1,l_2} is a modulating factor valued as follows

$$a_{l_1,l_2} = \begin{cases} 1, & \text{if the embedded bit is 1} \\ -1, & \text{if the embedded bit is 0} \end{cases} \quad (11)$$

The Green component is not altered. The detection is correlation-based i.e.

$$q_{l_1,l_2} = \frac{1}{N} \sum_i \sum_j I'_{(l_1,l_2)R}[i,j] \cdot w[i,j]$$

If the output q of the detector is less than zero then the extracted bit is 0, otherwise 1. Using this rule, the resulting probability of erroneous detection is very small. However, in order to achieve lossless extraction, error correcting codes can be used. Error correcting codes can detect and correct errors that may occur during the extraction of the embedded bitstream. In this paper, a simple Hamming code is used that adds three error control bits B_{C1}, B_{C2}, B_{C3} for every four information bits B_{I1}, B_{I2}, B_{I3}, B_{I4}. Thus, the embedding bitstream takes the form B_{C1}, B_{C2}, B_{I1}, B_{C3}, B_{I2}, B_{I3}, B_{I4} for every four indexing bits. If only a single error occurs while detecting the four indexing bits, the error can be corrected. The protection achieved using this approach is so strong (for the given application) that practically guarantees the correct extraction of all indexing bits.

6. EXPERIMENTAL RESULTS

The segmentation and content-based watermarking algorithms presented in the previous sections were tested for embedding information in a variety of test images [11]. As seen in Fig. 6, the segmentation algorithm is endowed with the capability to handle efficiently both textured and non-textured objects. This is due to the combined use of intensity, texture and position features for the image pixels. The derivation of the segmentation mask was followed by the extraction of indexing features for the formed regions as described in section 4. The above segmentation and indexing information was subsequently embedded in the images. Alternatively, instead of indexing information, any other kind of object-related information could be embedded, including a short text describing the object.

The segmentation information was embedded in the Blue component of RGB images using the procedure described in the previous section. The indexing information was embedded in the Red component of RGB images. Moreover, if the object was large enough, the same indexing bits were embedded twice or even more, until all available region blocks are used. The average time for watermarking an image was 0.07 seconds and the average time for the extraction of indexing information was 0.035 seconds on a computer with a Pentium-III processor. No perceptual degradation of image quality was observed due to watermarking. The 0.07 seconds include both mask and indexing information embedding but exclude the time needed for segmentation and feature extraction. The processes of segmenting an image and extracting indexing features from the formed regions are more time-consuming; the entire process in Fig. 1 takes roughly 15 seconds. However, this process is performed only once (at the time an image is segmented and marked) whereas the detection process (Fig. 2), which takes place many times (once for each different query), still needs 0.035 seconds/image.

The proposed system was subsequently tested for the retrieval of image regions using 1000 images of the Corel [11] database. In all cases, due to the channel coding, 100% of the embedded indexing bits were reliably extracted from the watermarked image. In most cases, the system was able to respond in less than 20 seconds and present the image region which was close to the one required by the user. However, for applications in which the speed of the system in its present form is considered not satisfactory, a separate file could be built off-line containing the features values that are embedded in the images. In this way, the feature values could be accessed much faster than extracting them from the images on-line.

Fig. 6. Images segmented into regions.

7. CONCLUSIONS

A methodology was presented for the segmentation and content-based embedding of indexing information in digital images. The segmentation algorithm combines pixel position, intensity and texture information to segment the image into a number of regions. Two types of watermarks are subsequently embedded in each region: a segmentation watermark and an indexing watermark. The proposed system is appropriate for building flexible data bases in which no side information is needed to be kept for each image. Moreover, the semantic regions comprising each image can be easily extracted using the segmentation watermark detection procedure.

8. ACKNOWLEDGEMENTS

This material is based upon work supported by the EU IST project IST-2000-32795 SCHEMA. The assistance of COST211 quat and COST276 is also gratefully acknowledged.

9. REFERENCES

[1] W. Zeng and B. Liu, "A Statistical Watermark Detection Technique Without Using Original Images for Resolving Rightful Ownerships of Digital Images," *IEEE Trans. Image Processing*, vol. 8, no. 11, pp. 1534–1548, Nov. 1999.

[2] A. M. Alattar, "Smart images using Digimarc's watermarking technology," in *Security and Watermarking of Multimedia Contents II, Proc SPIE,* Jan. 2000, vol. 3971, pp. 264–273.

[3] N. V. Boulgouris, I. Kompatsiaris, V. Mezaris, and M. G. Strintzis, "Content-based Watermarking for Indexing Using Robust Segmentation," in *Proc. Workshop on Image Analysis For Multimedia Interactive Services*, Tampere, Finland, May 2001.

[4] P. Bas, N. V. Boulgouris, F. D. Koravos, J.-M. Chassery, M. G. Strintzis, and B. Macq, "Robust Watermarking of Video Objects for MPEG-4 Applications," in *Proc. SPIE International Symposium on Optical Science and Technology*, San Diego, July 2001.

[5] N. V. Boulgouris, F. D. Koravos, and M. G. Strintzis, "Self-synchronizing Watermark Detection for MPEG-4 Objects," in *Proc. IEEE International Conference on Electronics, Circuits and Systems*, Malta, Sept. 2001.

[6] I. Kompatsiaris and M. G. Strintzis, "Spatiotemporal Segmentation and Tracking of Objects for Visualization of Videoconference Image Sequences," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 10, no. 8, Dec. 2000.

[7] S. Liapis, E. Sifakis, and G. Tziritas, "Color and/or Texture Segmentation using Deterministic Relaxation and Fast Marching Algorithms," in *Intern. Conf. on Pattern Recognition*, Sept. 2000, vol. 3, pp. 621–624.

[8] M. Unser, "Texture classification and segmentation using wavelet frames," *IEEE Trans. on Image Processing*, vol. 4, no. 11, pp. 1549–1560, Nov. 1995.

[9] S. Sakaida, Y. Shishikui, Y. Tanaka, and I. Yuyama, "Image segmentation by integration approach using initial dependence of k-means algorithm," in *Picture Coding Symposium 97*, Berlin, Germany, Sept. 1997, pp. 265–269.

[10] M. Kutter, *Digital image watermarking: hiding information in images*, PhD Thesis, EPFL, 1999.

[11] *Corel stock photo library*, Corel Corp., Ontario, Canada.