EURASIP Journal on Applied Signal Processing 2002:4, 420–433
© 2002 Hindawi Publishing Corporation

Segmentation and Content-Based Watermarking for Color Image and Image Region Indexing and Retrieval

Nikolaos V. Boulgouris
Informatics and Telematics Institute (ITI), 1st Km Thermi-Panorama Road, Thermi-Thessaloniki, P.O. Box 361, Gr-57001, Greece
Email: nblg@iti.gr

Ioannis Kompatsiaris
Informatics and Telematics Institute (ITI), 1st Km Thermi-Panorama Road, Thermi-Thessaloniki, P.O. Box 361, Gr-57001, Greece
Email: ikom@iti.gr

Vasileios Mezaris
Information Processing Laboratory, Electrical and Computer Engineering Department, Aristotle University of Thessaloniki, Thessaloniki 540 06, Greece
Email: bmezaris@iti.gr

Dimitrios Simitopoulos
Informatics and Telematics Institute (ITI), 1st Km Thermi-Panorama Road, Thermi-Thessaloniki, P.O. Box 361, Gr-57001, Greece
Email: dsim@iti.gr

Michael G. Strintzis
Information Processing Laboratory, Electrical and Computer Engineering Department, Aristotle University of Thessaloniki, Thessaloniki 540 06, Greece
Email: strintzi@eng.auth.gr

Received 30 July 2001 and in revised form 14 January 2002

An entirely novel approach to image indexing is presented using content-based watermarking. 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 and phrases: image segmentation, image analysis, 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, 2]. However, many other applications can be conceived which involve information hiding [3, 4]. In this paper, we propose the employment of watermarking as a means to content-based indexing and retrieval of images from databases.

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 the present paper, a novel segmentation algorithm is used prior to information embedding.

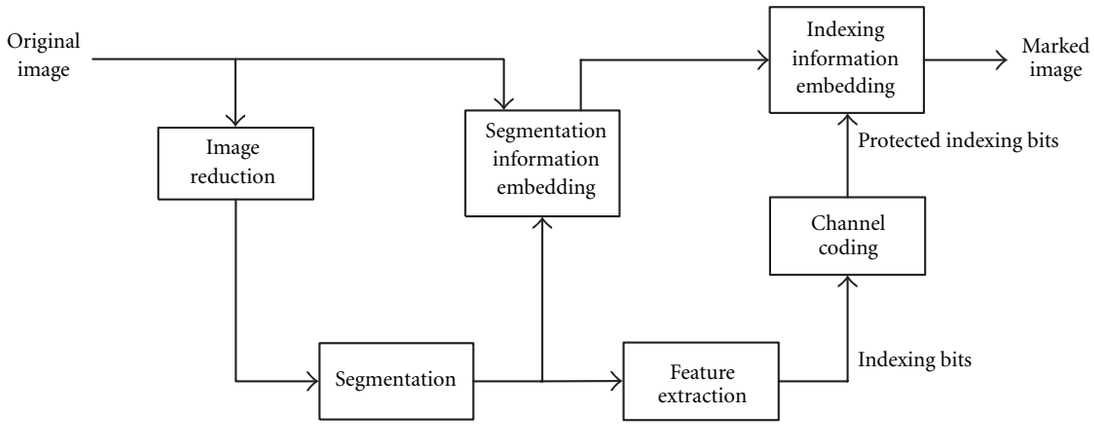


Figure 1: Block diagram of the embedding scheme.

Segmentation methods for 2D images may be divided primarily into region-based and boundary-based methods [5, 6, 7, 8]. In this paper, a region-based [9, 10] approach is presented using a combination of position, intensity, and texture information, in order to form large connected regions that correspond to the objects contained in the image. The segmentation of the image into regions is followed by the extraction 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. Specifically, segmentation information is embedded using an M-ary symbol modulation technique in which each symbol corresponds to an image region. Indexing information is embedded as a binary stream followed by channel coding. In this way, both segmentation and indexing information can be easily extracted using a fast watermark detection procedure. This is an entirely novel concept that clearly differentiates our system from classical indexing and retrieval methodologies, in which feature information for each image is separately stored in database records.

Embedding segmentation and indexing information in image regions [11, 12] 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 at the decoder from images without the requirement for employing segmentation algorithms.

The above advantages can be exploited using our watermarking methodology.

The paper is organized as follows: the system overview 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.

2. SYSTEM OVERVIEW

The block diagram of the proposed system is shown in Figure 1. The system first segments an image into objects using a segmentation algorithm that forms only 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 Figure 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.

A segmentation algorithm employing color and texture information is described in the ensuing section.

3. COLOR IMAGE SEGMENTATION

3.1. *Segmentation system overview*

The segmentation system described in this section is based on a variant of the popular K-means algorithm: the K-means-with-connectivity-constraint algorithm (KMCC) [13, 14]. This is an algorithm that 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

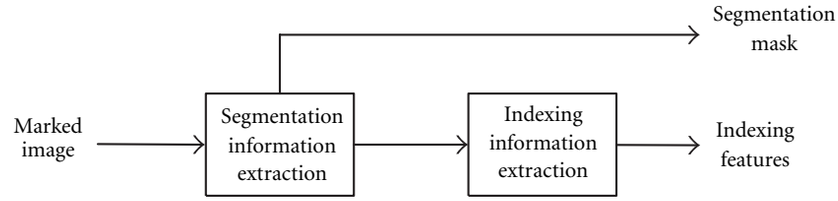


Figure 2: Block diagram of the detection scheme.

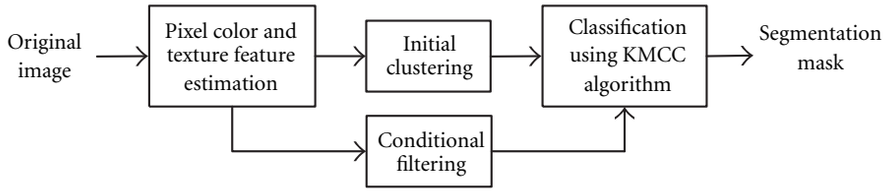


Figure 3: Overview of the segmentation algorithm.

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 nonuniform regions instead of breaking down the objects to a large number of chromatically uniform regions. To achieve this, the texture information is not only utilized by the KMCC algorithm, but is also used for determining whether and to which pixels of the image a moving average filter should be applied. Before the final application of the KMCC algorithm, a moving average filter alters the intensity information in those parts of the image where intensity fluctuations are particularly pronounced, since in these parts the KMCC algorithm does not perform efficiently. This stage of conditional filtering is described in more detail in the sequel.

The result of the application of the segmentation algorithm to a color image is the segmentation mask: a grayscale image in which different gray values correspond to different regions formed by the KMCC algorithm.

The segmentation algorithm consists of the following stages (Figure 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 classification of the pixels, using the KMCC 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 nonlinear 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, that is, the numerical distance in this color space is approximately proportional to the perceived color difference [15]. The color feature vector of pixel $\mathbf{p}$, $I(\mathbf{p})$ is defined as

$$I(\mathbf{p}) = \left[I_L(\mathbf{p}), I_a(\mathbf{p}), I_b(\mathbf{p})\right]^T. \tag{1}$$

In order to detect and characterize texture properties in the neighborhood of each pixel, the discrete wavelet frames (DWF) decomposition is used. This is a method similar to the discrete wavelet transform (DWT), that uses a filter bank to decompose each intensity coordinate of the image to a set of subbands (Figure 4). The main difference between the two methods is that in the DWF decomposition, the output of the filter bank is not subsampled. The DWF approach has been proven to decrease the variability of the estimated texture features, thus improving classification performance [16].

The filter bank used is based on the lowpass Haar filter

$$H(z) = \frac{1}{2}\left(1 + z^{-1}\right), \tag{2}$$

which satisfies the lowpass condition $H(z)|_{z=1} = 1$. The complementary highpass filter $G(z)$ is defined with respect to the lowpass $H(z)$ as follows:

$$G(z) = zH\left(-z^{-1}\right). \tag{3}$$

The filters of the filter bank, $H_{L_d}(z)$, $G_i(z)$, $i = 1, \ldots, L_d$ are generated by the prototypes $H(z)$, $G(z)$, according to the following equations:

$$\begin{aligned} H_{i+1}(z) &= H\left(z^{2^i}\right)H_i(z), \\ G_{i+1}(z) &= G\left(z^{2^i}\right)H_i(z), \quad i = 0, \ldots, L_d - 1, \end{aligned} \tag{4}$$

where $H_0(z) = 1$ is the necessary initial condition and L_d is the number of levels of decomposition. The frequency responses of those filters for $L_d = 2$ are presented in Figure 5.

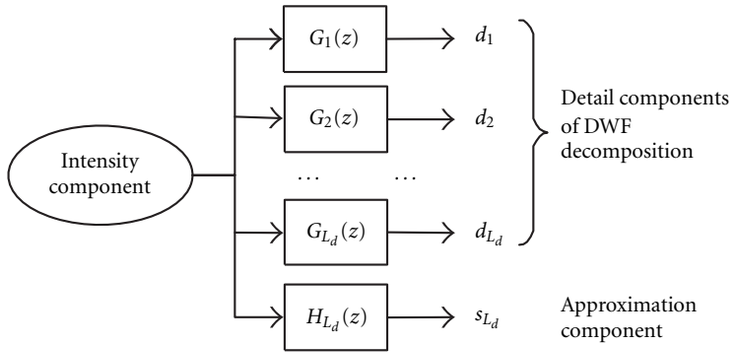


Figure 4: 1D Discrete Wavelet Frames decomposition of L_d levels.

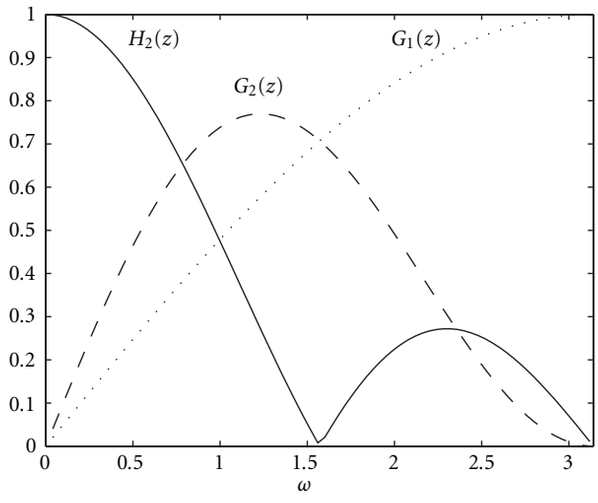


Figure 5: Frequency responses of Haar filter bank for 2 levels of decomposition.

Although the frequency localization of the filters is relatively poor, as shown by Figure 5, it has been shown in [16] that good space localization of the filter bank is more important than frequency localization; therefore simple prototype filters like the Haar filter used are good choices. The application of such simple filters also has the advantage of correspondingly reduced computational complexity.

The discrete wavelet frames decomposition can be extended in the two-dimensional space by successively processing the rows and columns of the image. In this case, for each intensity component and each level of decomposition, three detail components of the wavelet coefficients are produced. The fast iterative scheme used in our segmentation algorithm, for two levels of decomposition of the two-dimensional image, is presented in Figure 6.

The texture of pixel $\mathbf{p}$ is then characterized by the standard deviations of all detail components, calculated in a neighborhood F of pixel $\mathbf{p}$. This neighborhood F is a square of dimension $f \times f$, where in our system the value of f is odd and is chosen to be equal to the dimension of the blocks used for the initial clustering procedure (Section 3.3).

We have chosen to use a two-dimensional DWF decomposition of two levels: $L_d = 2$. Since three detail components are produced for each level of decomposition and each one of the three intensity components (Figure 6), the texture feature vector for pixel $\mathbf{p}$, $T(\mathbf{p})$, comprises 18 texture components, $\sigma_q(\mathbf{p})$, $q = 1, \ldots, 18$:

$$T(\mathbf{p}) = \left[\sigma_1(\mathbf{p}), \sigma_2(\mathbf{p}), \ldots, \sigma_{18}(\mathbf{p})\right]^T. \tag{5}$$

3.3. Initial clustering

Similarly to any other variant of the K-means algorithm, the KMCC algorithm requires initial values; in our case, an initial estimation is needed of the number of regions in the image and their spatial, intensity, and texture centers. A set of values chosen randomly could be used as initial values, since all these values can and are expected to be altered during the execution of the algorithm. Nevertheless, a well-chosen starting point can lead to a more accurate representation of the objects of the image. It can also facilitate the convergence of the K-means-with-connectivity-constraint algorithm, thus reducing the time necessary for the segmentation mask to be produced.

In order to compute the appropriate initial values, the image is broken down to square, nonoverlapping blocks of dimension $f \times f$. In this way, a total of L blocks, b_l, $l = 1, \ldots, L$ are created. In our experiments, the value of f was chosen so that the number L of blocks created would be approximately 75; this was found to be a good compromise between the need for accuracy of the initial clustering, which improves as the number L of blocks increases, and the need for its fast completion. The center of block b_l is pixel $\mathbf{p}^l_{\text{cntr}}$. A color feature vector $I(b_l)$ and a texture feature vector $T(b_l)$ are assigned to each block, as follows:

$$I(b_l) = \frac{1}{f^2}\sum_{m=1}^{f^2} I(\mathbf{p}^l_m), \qquad T(b_l) = T(\mathbf{p}^l_{\text{cntr}}), \tag{6}$$

where $\mathbf{p}^l_m$, $m = 1, \ldots, f^2$ are the pixels belonging to block b_l.

The distance between two blocks is defined as follows:

$$D(b_{l_1}, b_{l_2}) = \left\|I(b_{l_1}) - I(b_{l_2})\right\| + \left\|T(b_{l_1}) - T(b_{l_2})\right\|, \tag{7}$$

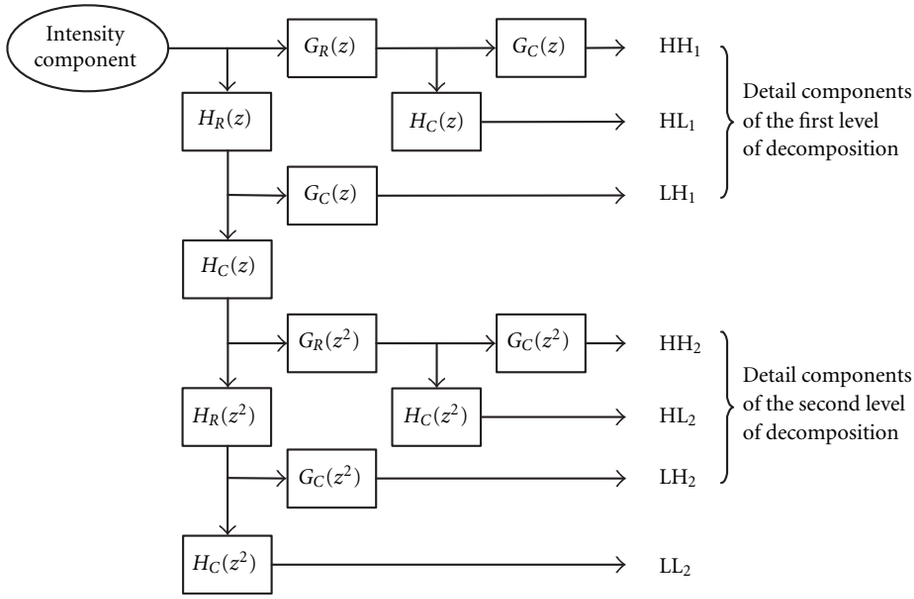


Figure 6: Fast iterative 2D discrete wavelet frames decomposition of 2 levels. Subscripts R, C denote filters applied row-wise and column-wise, respectively.

where

$$\|I(b_{l_1}) - I(b_{l_2})\| = \sqrt{(I_L(b_{l_1}) - I_L(b_{l_2}))^2 + (I_a(b_{l_1}) - I_a(b_{l_2}))^2 + (I_b(b_{l_1}) - I_b(b_{l_2}))^2},$$

$$\|T(b_{l_1}) - T(b_{l_2})\| = \sqrt{\sum_{q=1}^{18} (\sigma_q(\mathbf{p}_{\text{cntr}}^{l_1}) - \sigma_q(\mathbf{p}_{\text{cntr}}^{l_2}))^2}. \tag{8}$$

The number of regions of the image is initially estimated by applying a variant of the maximin algorithm to this set of blocks. This algorithm consists of the following steps.

Step 1. The block in the upper left corner of the image is chosen to be the first intensity and texture center.

Step 2. For each block b_l, $l = 1, \ldots, L$, the distance between b_l and the first center is calculated; the block for which the distance is maximized is chosen to be the second intensity and texture center. The distance $Db_{\max}$ between the first two centers is indicative of the intensity and texture contrast of the particular image.

Step 3. For each block b_l, the distances between b_l and all centers are calculated and the minimum of those distances is assigned to block b_l. The block that was assigned the maximum of the distances assigned to blocks is a new candidate center.

Step 4. If the distance that was assigned to the candidate center is greater than $\gamma \cdot Db_{\max}$, where γ is a predefined parameter, the candidate center is accepted as a new center and Step 3 is repeated; otherwise, the candidate center is rejected and the maximin algorithm is terminated. In our experiments the value for $\gamma = 0.4$ was used.

The number of centers estimated by the maximin algorithm constitutes an estimate of the number of regions in the image. Nevertheless, it is not possible to determine whether these regions are connected or not. Furthermore, there is no information regarding their spatial centers. In order to solve these problems, a simple K-means algorithm is applied to the set of blocks, using the information produced by the maximin algorithm as initial values. The simple K-means algorithm consists of the following steps.

Step 1. The output of the maximin algorithm is used as a starting point, regarding the number of regions s_k, $k = 1, \ldots, K$, and their intensity and texture centers, $I(s_k)$ and $T(s_k)$, respectively.

Step 2. For every block b_l, $l = 1, \ldots, L$, the distance is evaluated between b_l and all region centers. The block b_l is assigned to the region for which the distance is minimized.

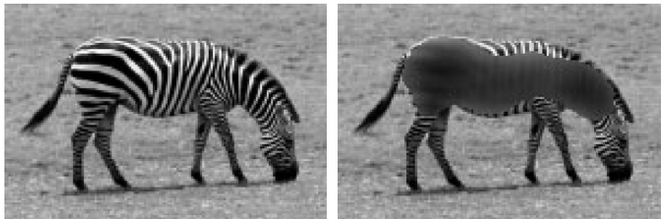
Step 3. Region centers are recalculated, as the mean values of the intensity and texture vectors over the blocks belonging to the corresponding region

$$I(s_k) = \frac{1}{M_k'} \sum_{m=1}^{M_k'} I(b_m^k), \qquad T(s_k) = \frac{1}{M_k'} \sum_{m=1}^{M_k'} T(b_m^k), \tag{9}$$

where b_m^k, $m = 1, \ldots, M_k'$ are the blocks currently assigned to region s_k.

Step 4. If the new centers are equal to those calculated in the previous iteration of the algorithm, then stop, else go to Step 2.

When the K-means algorithm converges, the connectivity of the regions that were formed is evaluated; those that are



(a) (b)

Figure 7: (a) Original image "zebra." (b) Filtered image.

not connected are easily broken down to the minimum number of connected regions using a recursive four-connectivity component labelling algorithm [17], so that a total of K' connected regions are identified. Their centers, including their spatial centers $S(s_k) = [S_x(s_k), S_y(s_k)]^T$, $k = 1, \ldots, K'$, will now be calculated. In order to obtain other useful information as well, such as the current size M_k of each region in pixels, we choose to perform the center calculation process not in the block domain but in the pixel domain, as we will do during the execution of the KMCC algorithm. These centers will be used as initial values by the KMCC algorithm:

$$\begin{aligned} I(s_k) &= \frac{1}{M_k}\sum_{m=1}^{M_k} I(\mathbf{p}_m^k), \\ T(s_k) &= \frac{1}{M_k}\sum_{m=1}^{M_k} T(\mathbf{p}_m^k), \\ S_x(s_k) &= \frac{1}{M_k}\sum_{m=1}^{M_k} p_{m,x}^k, \\ S_y(s_k) &= \frac{1}{M_k}\sum_{m=1}^{M_k} p_{m,y}^k, \end{aligned} \tag{10}$$

where M_k is the number of pixels $\mathbf{p}_m^k$, $m = 1, \ldots, M_k$ that belong to region s_k.

3.4. Conditional filtering

In some images, there are parts of the image where intensity fluctuations are particularly pronounced, even when all pixels in that part of the image belong to a single object (Figure 7). In order to facilitate the grouping of all these pixels in a single region based on their texture similarity, which is our objective, it would be of great importance to somehow reduce their intensity differences. This can be achieved by applying a moving average filter to the appropriate parts of the image, thus altering the intensity information of the corresponding pixels.

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 $T(p)$ (see Section 3.2); 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:

$$J(\mathbf{p}) = \begin{cases} I(\mathbf{p}) & \text{if } \|T(\mathbf{p})\| < T_{\text{th}}, \\ \dfrac{1}{f^2}\displaystyle\sum_{m=1}^{f^2} I(\mathbf{p}_m) & \text{if } \|T(\mathbf{p})\| \geq T_{\text{th}}. \end{cases} \tag{11}$$

An appropriate value of the threshold T_{th} was experimentally found to be

$$T_{\text{th}} = \max\{0.65 \cdot T_{\max}, 14\}, \tag{12}$$

where $T_{\max}$ is the maximum value of the norm $\|T(\mathbf{p})\|$ in the image. For computational efficiency purposes, the maximum of $\|T(\mathbf{p})\|$ can be sought only among the pixels that served as block centers during the initial clustering described in Section 3.3. The term $0.65 \cdot T_{\max}$ in the threshold definition is used to make sure that the filter will not be applied outside the borders of the textured objects. In this way, the boundaries of the textured objects will not be corrupted, thus enabling the KMCC algorithm to accurately detect those boundaries. The constant term 14, on the other hand, is necessary for the system to be able to handle images composed of chromatically uniform objects; in such images, the value of $T_{\max}$ is expected to be relatively small and would correspond to pixels on edges between objects, where the application of a moving average filter is obviously undesirable.

The output of the conditional filtering stage will be now used 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 [18, 19, 20] 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, we propose the use of an extended K-Means algorithm: the K-means-with-connectivity-constraint algorithm. 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.

The K-means with connectivity constraint (KMCC) algorithm, that is, applied on the pixels of the image consists of the following steps (Figure 8).

Step 1. An initial clustering is produced, using the estimation procedure in Section 3.3; thus the number of regions K is initialized as $K = K'$.

Step 2. For every pixel $\mathbf{p}$, the distance between $\mathbf{p}$ and all region centers is calculated. The pixel is then assigned to the region for which the distance is minimized. A generalized distance of a pixel $\mathbf{p}$ from a region s_k is defined as follows:

$$\begin{aligned} D(\mathbf{p}, s_k) = {} & \|J(\mathbf{p}) - J(s_k)\| \\ & + \|T(\mathbf{p}) - T(s_k)\| + \lambda\frac{\bar{A}}{A_k}\|\mathbf{p} - S(s_k)\|, \end{aligned} \tag{13}$$

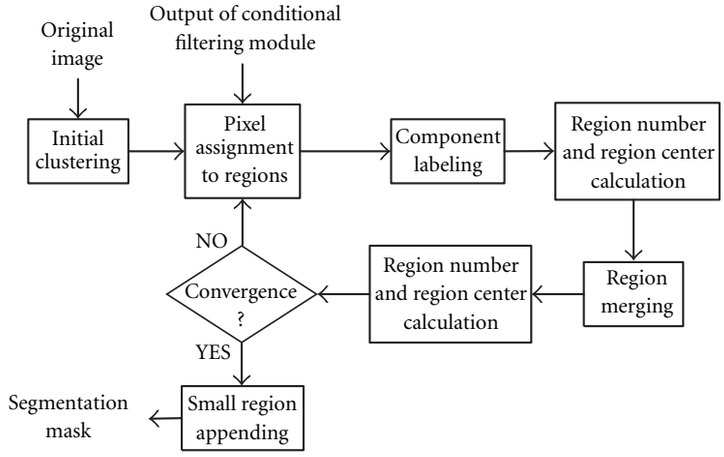


Figure 8: Block diagram of the KMCC algorithm.

where

$$\begin{aligned}
&\|J(\mathbf{p}) - J(s_k)\| \\
&= \sqrt{(J_L(\mathbf{p}) - J_L(s_k))^2 + (J_a(\mathbf{p}) - J_a(s_k))^2 + (J_b(\mathbf{p}) - J_b(s_k))^2}, \\
&\|T(\mathbf{p}) - T(s_k)\| = \sqrt{\sum_{q=1}^{18} (\sigma_q(\mathbf{p}) - T_q(s_k))^2}, \\
&\|\mathbf{p} - S(s_k)\| = \sqrt{(p_x - S_x(s_k))^2 + (p_y - S_y(s_k))^2},
\end{aligned} \tag{14}$$

the area A_k of each region is defined as

$$A_k = M_k, \tag{15}$$

where M_k is the number of pixels assigned to region s_k, and $\bar{A}$ is the average area of all regions:

$$\bar{A} = \frac{1}{K} \sum_{k=1}^{K} A_k. \tag{16}$$

The regularization parameter λ is defined as:

$$\lambda = 0.4 \cdot \frac{Db_{\max}}{\sqrt{p_{x,\max}^2 + p_{y,\max}^2}}. \tag{17}$$

Normalization of the spatial distance, $\|\mathbf{p} - S_k\|$ with the area of each region, $\bar{A}/A_k$ is necessary in order to encourage the creation of large connected regions; otherwise, pixels would tend to be assigned to smaller rather than larger regions due to greater spatial proximity to their centers. In this case, large objects would be broken down to more than one neighboring smaller regions instead of forming one single, larger region.

The regularization parameter λ is used to make sure that a pixel is assigned to a region primarily due to their intensity and texture similarity. Being proportional to the intensity and texture contrast $Db_{\max}$ of the image, it ensures that even in low-contrast images, where intensity and texture differences are small, these will not become insignificant compared to spatial distances. The opposite would result in the formation of regions that would not correspond to the objects of the image.

Step 3. The connectivity of the regions formed is evaluated; those which are not connected are easily broken down to the minimum number of connected regions using a recursive four-connectivity component labelling algorithm [17].

Step 4. Region centers are recalculated. Regions with areas below a size threshold th_{size} are dropped. The number of regions K is also recalculated, taking into account only the remaining regions.

Step 5. Two regions are merged if they are neighbors and if their intensity and texture distance is not greater than an appropriate merging threshold:

$$\begin{aligned}
D(s_{k_1}, s_{k_2}) = {} & \|J(s_{k_1}) - J(s_{k_2})\| \\
& + \|T(s_{k_1}) - T(s_{k_2})\| \le \text{th}_{\text{merge}}.
\end{aligned} \tag{18}$$

Step 6. Region number K and region centers are once again reevaluated.

Step 7. If the region number K is equal to the one calculated in Step 6 of the previous iteration and the difference between the new centers and those in Step 6 of the previous iteration is below the corresponding threshold for all centers, then stop, else go to Step 2. If index "old" characterizes the region number and region centers calculated in Step 6 of the previous iteration, the convergence condition can be expressed as:

$$\begin{aligned}
K &= K^{\text{old}}, \\
\|J(s_k) - J(s_k^{\text{old}})\| &\le \text{th}_I, \\
\|T(s_k) - T(s_k^{\text{old}})\| &\le \text{th}_T, \\
\|S(s_k) - S(s_k^{\text{old}})\| &\le \text{th}_S,
\end{aligned} \tag{19}$$

for $k = 1, \ldots, K$.

Even though the region centers of particularly small regions are omitted in Step 4 and the formation of large regions is encouraged in Step 2, there is no guarantee that no

Table 1: Threshold values.

Threshold description	Threshold value

Texture threshold	$T_{\text{th}} = \max\{0.65 \cdot T_{\max}, 14\}$
Region size threshold	$\text{th}_{\text{size}} = 0.75\%$ of the total image area
Region merging threshold	$\text{th}_{\text{merge}} = \begin{cases} 20 & \text{if } Db_{\max} > 75 \\ 15 & \text{if } Db_{\max} \leq 75 \text{ and } T_{\max} \geq 14 \\ 10 & \text{if } Db_{\max} \leq 75 \text{ and } T_{\max} < 14 \end{cases}$
Convergence thresholds	$\text{th}_I = 4.0$ $\text{th}_T = 1.0$ $\text{th}_S = 2.0$

such small regions will be present in the segmentation mask after the convergence of the algorithm. Since these regions are not wanted, they are forced to merge with one of their neighboring regions, based on intensity and texture similarity: a small region s_{k_1}, $M_{k_1} < \text{th}_{\text{size}}$ is appended to region s_{k_2}, $k_2 = 1, \ldots, K$, $k_2 \neq k_1$, for which the distance

$$D(s_{k_1}, s_{k_2}) = \|J(s_{k_1}) - J(s_{k_2})\| + \|T(s_{k_1}) - T(s_{k_2})\| \quad (20)$$

is minimum. This procedure is performed for all small regions of the segmentation mask, until all such small regions are absorbed.

In Table 1, a summary of the thresholds required by the segmentation algorithm and the corresponding values used in our experiments is presented.

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 region 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 the intensity centers of that region, $I_L(s_k)$, $I_a(s_k)$, $I_b(s_k)$, whereas the position descriptors $P_{k,x}$, $P_{k,y}$ are the spatial centers normalized by the dimensions of the image:

$$P_{k,x} = \frac{S_{k,x}}{x_{\max}}, \qquad P_{k,y} = \frac{S_{k,y}}{y_{\max}}. \quad (21)$$

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:

$$E_k = \frac{M_k}{x_{\max} \cdot y_{\max}}. \quad (22)$$

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

$$\mathbf{C}_k = \frac{1}{M_k} \sum_{m=1}^{M_k} (\mathbf{p}_m^k - S_k)(\mathbf{p}_m^k - S_k)^T, \quad (23)$$

where $\mathbf{p}_m^k = [p_{m,x}^k, p_{m,y}^k]^T$, $m = 1, \ldots, M_k$ are the pixels belonging to region s_k. 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:

$$\varepsilon_k = 1 - \frac{\rho_1}{\rho_2}. \quad (24)$$

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

$$\mathbf{D}_k = [I_L(s_k), I_a(s_k), I_b(s_k), P_{k,x}, P_{k,y}, E_k, \theta_k, \varepsilon_k]. \quad (25)$$

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.

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

Figure 9: Example of embedding labels to each image block.

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 Figure 9). 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 [21]. 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, that is, 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:

$$I'_{(l_1,l_2)B}[i, j] = I_B\left[8l_1 + i, 8l_2 + j\right] + a_{l_1,l_2} \cdot w[i, j], \tag{26}$$

where l_1, l_2 are the block indices, i, j are the indices specifying the position of the pixel inside the block and a_{l_1,l_2} is a modulating factor valued as follows:

$$a_{l_1,l_2} = \begin{cases} -3, & \text{if the block } (l_1, l_2) \text{ belongs to the} \\ & \text{background or is not classified,} \\ -1, & \text{if the block } (l_1, l_2) \text{ belongs to the 1st region,} \\ +1, & \text{if the block } (l_1, l_2) \text{ belongs to the 2nd region,} \\ +3, & \text{if the block } (l_1, l_2) \text{ belongs to the 3rd region} \end{cases} \tag{27}$$

+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

Figure 10: 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.

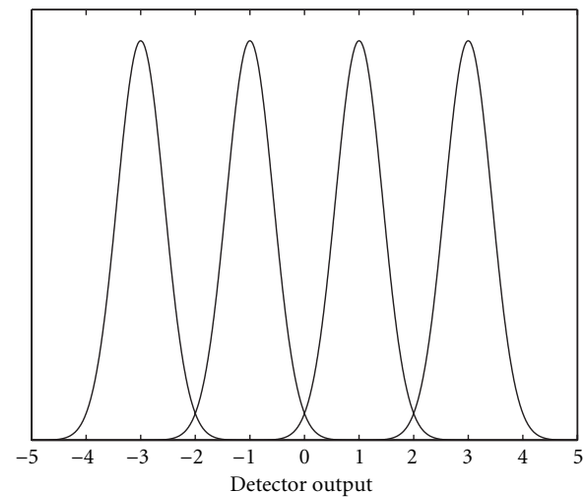


Figure 11: Distributions of the detector statistic for the four symbols.

and $w[i, j]$ is the watermark matrix (see Figure 10), which is the same for all blocks in the image and is given by $w[i, j] = (-1)^{i+j}$, that is,

$$w[i, j] = \begin{cases} 1, & \text{if } i + j = \text{even}, \\ -1, & \text{if } i + j = \text{odd}. \end{cases} \tag{28}$$

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

$$q_{l_1,l_2} = \frac{1}{N} \sum_i \sum_j I'_{(l_1,l_2)B}[i, j] \cdot w[i, j], \tag{29}$$

where N is the number of pixels in a block. In our case, $N = 64$ since 64 pixels are included in an 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 (see Figure 11). By choosing the decision boundaries to coincide with the value of q at which adjacent distributions in Figure 11 cross, the probability of misclassification is minimized [22]. Clearly, the optimal rule for extracting the label of block (l_1, l_2) is

$$s = \begin{cases} 0, & \text{if } q_{l_1,l_2} < -2, \\ 1, & \text{if } -2 < q_{l_1,l_2} < 0, \\ 2, & \text{if } 0 < q_{l_1,l_2} < 2, \\ 3, & \text{if } 2 < q_{l_1,l_2} \end{cases} \tag{30}$$

since the above choice minimizes the probability of erroneous symbol detection. This probability will be shown to be very small. In fact, using the conditional distributions of the detector statistic for each of the four symbols, the probabilities P_i, $i = 0, \ldots, 3$, that the ith symbol is extracted erroneously are

$$P_0 = \frac{1}{\sqrt{2\pi}\sigma} \int_{-2}^{\infty} e^{-(x+3)^2/2\sigma^2} dx, \tag{31}$$

$$P_1 = \frac{1}{\sqrt{2\pi}\sigma} \left(\int_{-\infty}^{-2} e^{-(x+1)^2/2\sigma^2} dx + \int_{0}^{\infty} e^{-(x+1)^2/2\sigma^2} dx \right), \tag{32}$$

$$P_2 = P_1, \tag{33}$$

$$P_3 = P_0. \tag{34}$$

The mean value of the detector output is

$$\begin{aligned} m &= \frac{1}{N} \sum_i \sum_j \left(I_{(l_1,l_2)B}[i,j] + a_{l_1,l_2} \cdot w[i,j] \right) \cdot w[i,j] \\ &= \frac{1}{N} \sum_i \sum_j \left(I_{(l_1,l_2)B}[i,j] \cdot w[i,j] + a_{l_1,l_2} \cdot w[i,j]^2 \right) \\ &= \frac{1}{N} \left(\sum_i \sum_j I_{(l_1,l_2)B}[i,j] \cdot w[i,j] + a_{l_1,l_2} \sum_i \sum_j w[i,j]^2 \right). \end{aligned} \tag{35}$$

From (28),

$$\begin{aligned} &\frac{1}{N} \sum_i \sum_j I_{(l_1,l_2)B}[i,j] \cdot w[i,j] \\ &\quad = \frac{1}{N} \left(\sum_{i+j:\text{even}} I_{(l_1,l_2)B}[i,j] - \sum_{i+j:\text{odd}} I_{(l_1,l_2)B}[i,j] \right) \end{aligned} \tag{36}$$

which is a very small quantity. Furthermore,

$$\sum_i \sum_j w[i,j]^2 = N. \tag{37}$$

Thus, (35) yields

$$m = \frac{1}{N} \cdot a_{l_1,l_2} \cdot N = a_{l_1,l_2} \tag{38}$$

and the variance of the correlator output is

$$\begin{aligned} \sigma^2 &= \left(\frac{1}{N} \sum_i \sum_j \left(I_{(l_1,l_2)B}[i,j] \cdot w[i,j] \right.\right. \\ &\qquad \left.\left. + a_{l_1,l_2} \cdot w[i,j]^2 \right) - a_{l_1,l_2} \right)^2 \\ &= \left(\frac{1}{N} \sum_i \sum_j I_{(l_1,l_2)B}[i,j] \cdot w[i,j] \right. \\ &\qquad \left. + \frac{1}{N} a_{l_1,l_2} \sum_i \sum_j w[i,j]^2 - a_{l_1,l_2} \right)^2 \\ &= \left(\frac{1}{N} \sum_i \sum_j I_{(l_1,l_2)B}[i,j] \cdot w[i,j] + \frac{1}{N} \cdot a_{l_1,l_2} \cdot N - a_{l_1,l_2} \right)^2 \\ &= \left(\frac{1}{N} \sum_i \sum_j I_{(l_1,l_2)B}[i,j] \cdot (-1)^{i+j} \right)^2 \\ &= \frac{1}{N^2} \left(\sum_{i+j:\text{even}} I_{(l_1,l_2)B}[i,j] - \sum_{i+j:\text{odd}} I_{(l_1,l_2)B}[i,j] \right)^2 \\ &= \frac{1}{64^2} \left(\sum_{i+j:\text{even}} I_{(l_1,l_2)B}[i,j] - \sum_{i+j:\text{odd}} I_{(l_1,l_2)B}[i,j] \right)^2. \end{aligned} \tag{39}$$

Since the term in brackets is far lower than 64^2 (intensities in a block are highly correlated), the resulting variance σ^2 will be a very small quantity (see Figure 11).

In most practical cases, if no attack is performed in the image, the variance σ^2 of the segmentation watermark detector is approximately equal to 0.09. By substituting this value in (31) and (32), the probability that the label of a block in an object is misinterpreted is less than 10^{-3}. Although this is a very 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 (see Figure 12). As seen, detection errors occurring at blocks within a region can be easily corrected since isolated blocks obviously cannot be considered objects. However, errors at block boundaries cannot be corrected since there is ambiguity regarding the region in which they belong. 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

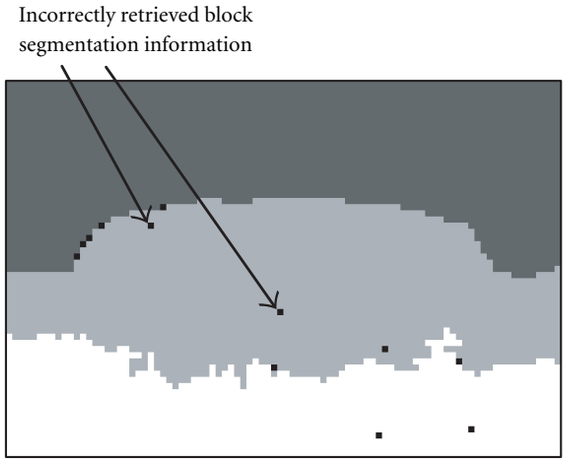


FIGURE 12: Incorrectly retrieved block segmentation information.

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:

$$I'_{(l_1,l_2)R}[i, j] = I_R\left[8l_1 + i, 8l_2 + j\right] + a_{l_1,l_2} \cdot w[i, j], \tag{40}$$

where w is the watermark matrix given in (28) and a_{l_1,l_2} is a modulating factor valued as follows:

$$a_{l_1,l_2} = \begin{cases} 1, & \text{if the embedded bit is 1,} \\ -1, & \text{if the embedded bit is 0.} \end{cases} \tag{41}$$

The green component is not altered. The detection is correlation-based, that is,

$$q_{l_1,l_2} = \frac{1}{N} \sum_i \sum_j I'_{(l_1,l_2)R}[i, j] \cdot w[i, j]. \tag{42}$$

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 [23] 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}. The control bits are computed from the information bits in the following way:

$$\begin{aligned} B_{C1} &= B_{I1} \oplus B_{I2} \oplus B_{I4}, \\ B_{C2} &= B_{I1} \oplus B_{I3} \oplus B_{I4}, \\ B_{C3} &= B_{I2} \oplus B_{I3} \oplus B_{I4}, \end{aligned} \tag{43}$$

where $\oplus$ denotes the XOR operation. 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 watermarking algorithms described in the previous sections were tested for embedding information in a variety of test images [24]. As seen in Figures 13 and 14, 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 (these features are also used by the ISTORAMA[1] content-based image retrieval system). 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 (see Figure 15) 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 and in our system take several seconds/image. In practice, the entire process in Figure 1 takes roughly 15 seconds. However, the process in Figure 1 is performed only once (at the time an image is segmented and marked) whereas the detection process (Figure 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 from the ISTORAMA database. In all cases, due to the channel coding, 100% of the embedded indexing bits were reliably extracted from the watermarked image. A simple retrieval example is shown in Figure 16. 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

[1] http://uranus.ee.auth.gr/Istorama/.

Figure 13: Images segmented into regions.

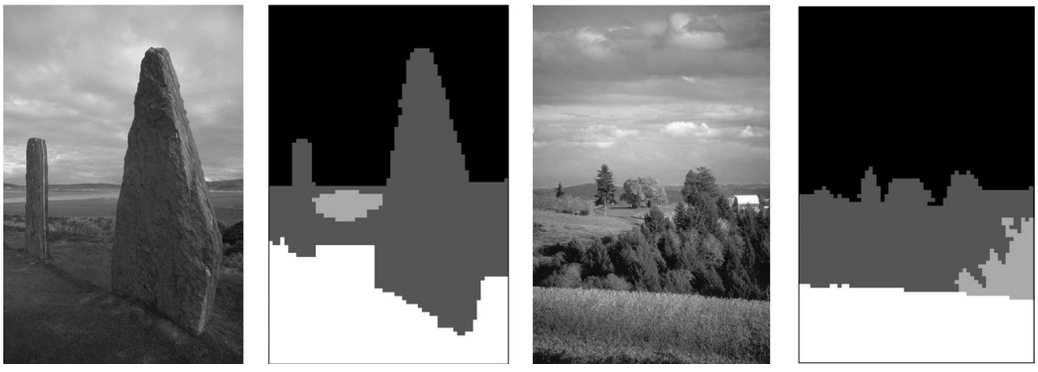


Figure 14: Images segmented into regions.

(a) (b)

Figure 15: (a) Original image. (b) Watermarked image. No perceptual degradation can be observed.

applications in which the speed of the system in its present form is considered not satisfactory, a separate file could be built offline 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 online.

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 in order 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 databases 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.

ACKNOWLEDGMENTS

The authors are with the Information Processing Laboratory, Electrical and Computer Engineering Department, Aristotle University of Thessaloniki, Thessaloniki, Greece, and the Informatics and Telematics Institute, Thessaloniki, Greece. This work was supported by the COST 211quat and the EU IST project ASPIS.

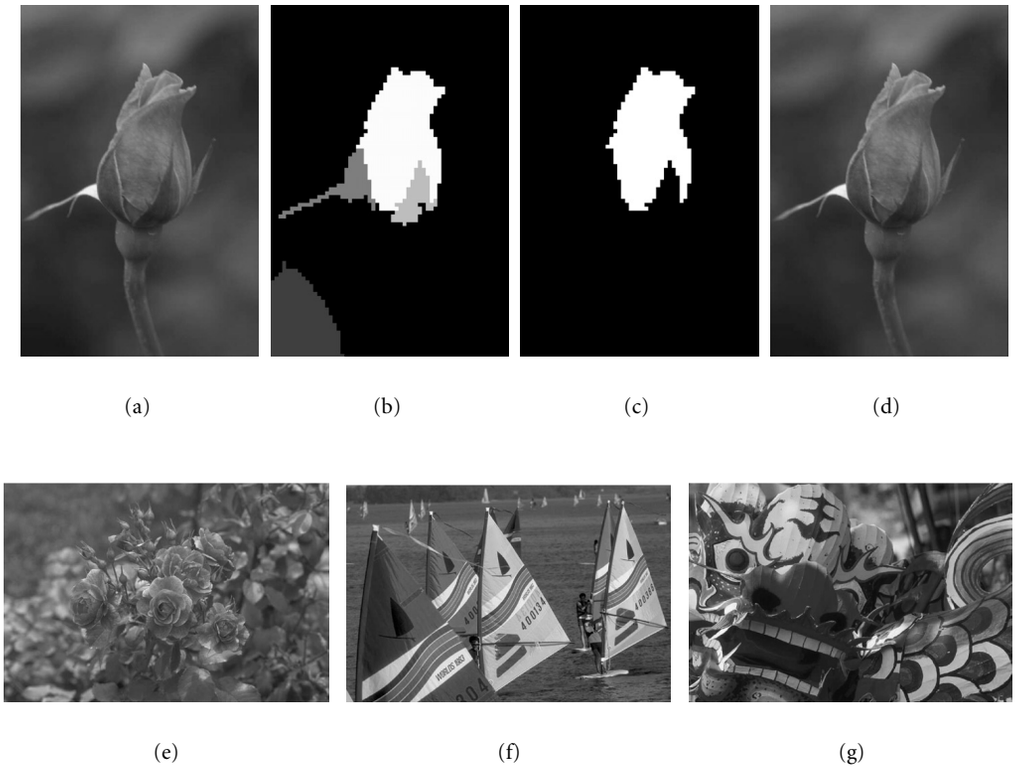


Figure 16: (a) Original image (730 × 490). (b) Segmentation mask. (c) The region used for querying is presented in white. (d)–(g) Retrieved images.

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, 1999.

[2] D. Simitopoulos, N. V. Boulgouris, A. Leontaris, and M. G. Strintzis, "Scalable detection of perceptual watermarks in JPEG 2000 images," in *Conference on Communications and Multimedia Security*, Darmstadt, Germany, May 2001.

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

[4] 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.

[5] "Special issue," *IEEE Trans. Circuits and Systems for Video Technology, Special Issue on Image and Video Processing*, vol. 8, no. 5, September 1998.

[6] "Special issue," *Signal Processing, Special Issue on Video Sequence Segmentation for Content-Based Processing and Manipulation*, vol. 66, no. 2, April 1998.

[7] K. S. Fu and J. K. Mui, "A survey on image segmentation," *Pattern Recognition*, vol. 13, no. 1, pp. 3–16, 1981.

[8] R. M. Haralick and L. G. Shapiro, "Image segmentation techniques," *Comput. Vision Graphics Image Process.*, vol. 29, no. 1, pp. 100–132, 1985.

[9] A. A. Alatan, L. Onural, M. Wollborn, R. Mech, E. Tuncel, and T. Sikora, "Image sequence analysis for emerging interactive multimedia services—the European COST 211 framework," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 8, no. 7, pp. 19–31, 1998.

[10] D. Geiger and A. Yuille, "A common framework for image segmentation," *International Journal of Computer Vision*, vol. 6, no. 3, pp. 227–243, 1991.

[11] 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, Calif, USA, July–August 2001.

[12] 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, September 2001.

[13] I. Kompatsiaris and M. G. Strintzis, "Content-based representation of colour image sequences," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Salt Lake City, Utah, USA, May 2001.

[14] I. Kompatsiaris and M. G. Strintzis, "Spatiotemporal segmentation and tracking of objects for visualization of videoconference image sequences," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 10, no. 8, 2000.

[15] S. Liapis, E. Sifakis, and G. Tziritas, "Color and/or texture segmentation using deterministic relaxation and fast marching algorithms," in *International Conference on Pattern Recognition*, vol. 3, pp. 621–624, September 2000.

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

[17] R. Jain, R. Kasturi, and B. G. Schunck, *Machine Vision*, McGraw-Hill International Editions, New York, USA, 1995.

[18] S. Z. Selim and M. A. Ismail, "K-means-type algorithms," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 6, no. 1, pp. 81–87, 1984.

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

[20] I. Kompatsiaris and M. G. Strintzis, "3D representation of videoconference image sequences using VRML 2.0," in *European Conference for Multimedia Applications Services and Techniques (ECMAST '98)*, pp. 3–12, Berlin, Germany, May 1998.

[21] M. Kutter, *Digital image watermarking: hiding information in images*, Ph.D. thesis, 1999.

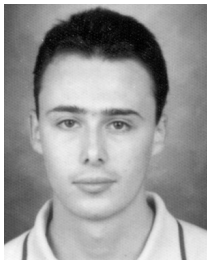
[22] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley, New York, USA, 1973.

[23] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1983.

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

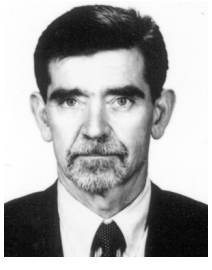
Nikolaos V. Boulgouris was born in Greece in 1975. He received the Diploma and the Ph.D. degrees from the Electrical and Computer Engineering Department, Aristotle University of Thessaloniki, Thessaloniki, Greece, in 1997 and 2002, respectively. He is currently a Researcher in the Informatics and Telematics Institute, Thessaloniki, Greece. Since 1997, Dr. Boulgouris has participated in several research projects funded by the European Union and the Greek Secretariat of Research and Technology. His research interests include image/video communication, networking, content-based indexing and retrieval, wavelets, pattern recognition and multimedia copyright protection. Nikolaos V. Boulgouris is a member of the Technical Chamber of Greece and a member of IEEE.

Ioannis Kompatsiaris received the Diploma degree in electrical engineering and the Ph.D. degree in 3D model based image sequence coding from Aristotle University of Thessaloniki (AUTH), Thessaloniki, Greece in 1996 and 2001, respectively. He is a Senior Researcher with the Informatics and Telematics Institute, Thessaloniki. Previously, he was a Leading Researcher on 2D and 3D Imaging at AUTH. I. Kompatsiaris has participated in many research projects funded by the EC and the GSRT. His research interests include image processing, computer vision, model-based monoscopic and multiview image sequence analysis and coding, medical image processing and video coding standards (MPEG-4, MPEG-7, MPEG-21). He is a representative of the Greek National Standardization Body (ELOT) to the ISO JTC 1/SC 29/WG 11 MPEG group. In the last 4 years, he has authored 6 articles in scientific journals and delivered over 20 scientific conference presentations in these and similar areas. I. Kompatsiaris is a member of the Technical Chamber of Greece.



Vasileios Mezaris was born in Athens, Greece, in 1979. He received the Diploma degree in Electrical and Computer Engineering in 2001 from the Electrical and Computer Engineering Department, Aristotle University of Thessaloniki, Greece, where he is currently working towards the Ph.D. degree. He is also a Graduate Research Assistant at the Informatics and Telematics Institute, Thessaloniki, Greece. His research interests include still image segmentation, video segmentation and object tracking, video streaming.

Dimitrios Simitopoulos was born in Greece in 1977. He received his Diploma in Electrical and Computer Engineering from Aristotle University of Thessaloniki, Greece, in 1999. He is currently working towards the Ph.D. degree in the Department of Electrical and Computer Engineering in Aristotle University of Thessaloniki, where he holds a teaching assistanship position. Since 2000, he is working as a research assistant in Informatics and Telematics Institute. His research interests include watermarking and multimedia security and image indexing and retrieval.



Michael G. Strintzis received the Diploma in Electrical Engineering from the National Technical University of Athens, Athens, Greece in 1967, and the M.A. and Ph.D. degrees in Electrical Engineering from Princeton University, Princeton, NJ, USA in 1969 and 1970, respectively. He then joined the Electrical Engineering Department at the University of Pittsburgh, Pittsburgh, Pa, USA, where he served as Assistant (1970–1976) and Associate (1976–1980) Professor. Since 1980 he is Professor of Electrical and Computer Engineering at the University of Thessaloniki, and since 1999 Director of the Informatics and Telematics Research Institute in Thessaloniki, Greece. Since 1999 he serves as an Associate Editor of the IEEE Transactions on Circuits and Systems for Video Technology. His current research interests include 2D and 3D image coding, image processing, biomedical signal and image processing and DVD and Internet data authentication and copy protection. In 1984, Dr. Strintzis was awarded one of the Centennial Medals of the IEEE.