PERGAMON

Pattern Recognition 36 (2003) 1679–1694

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Color image retrieval using multispectral random field texture model and color content features

Alireza Khotanzad*, Orlando J. Hernandez

Department of Electrical Engineering, Southern Methodist University, Dallas, TX 75275-0338, USA

Received 29 March 2002; received in revised form 19 September 2002; accepted 19 September 2002

Abstract

This paper describes a color-texture-based image retrieval system for query of an image database to find similar images to a target image. The color-texture information is obtained via modeling with the multispectral simultaneous autoregressive (MSAR) random field model. The general color content characterized by ratios of sample color means is also used. The retrieval process involves segmenting the image into regions of uniform color texture using an unsupervised histogram clustering approach that utilizes the combination of MSAR and color features. The color-texture content, location, area and shape of the segmented regions are used to develop similarity measures describing the closeness of a query image to database images. These attributes are derived from the maximum fitting square and best fitting ellipse to each of the segmented regions. The proposed similarity measure combines all these attributes to rank the closeness of the images. The performance of the system is tested on two databases containing synthetic mosaics of natural textures and natural scenes, respectively.

Keywords: Color image retrieval; Image based query; Color texture; Multispectral random field models; Similarity metrics; Color-texture segmentation

1. Introduction

An ever-increasing usage of digital images and large volume image databases gives rise to the need for organizing them according to their content so they can be retrieved easily. Retrieval of image data based on pictorial content queries is an interesting and challenging problem actively worked on by the research community [1]. With the growth of multimedia computing and the spread of the Internet, more and more people have access to large databases and would have applications for such retrieval systems.

This paper describes a color-texture-based retrieval system that finds images similar to a query image in a large database. The similarity criterion is based on the notion of containing same/similar color-texture regions. The approach involves characterizing color-texture using features derived from a class of multispectral random field models and color space. These features are then used in an unsupervised histogram clustering-based segmentation algorithm to find regions of uniform texture in the query image. The retrieval process involves identifying those images in the database that contain similar kinds of texture found in the query image. The system then considers the similarities in the size and spatial arrangement of the texture regions to carry out the retrieval process.

Texture information has been used in the past for browsing and retrieval of imagery, however, the previously proposed approaches have either considered only gray-level textures or pixels-based color content and not color-texture [2]. The utilized segmentation algorithm in some approaches has also not been completely unsupervised [3].

The contributions of this paper are as follows:

- Texture characterization by a mix of multispectral random field based features and color content features.

* Corresponding author. Fax: +1-214-768-3563.
E-mail address: kha@engr.smu.edu (A. Khotanzad).

0031-3203/03/$30.00
PII: S0031-3203(02)00292-3

- Development of a completely unsupervised color-texture-based segmentation algorithm and demonstration of its effectiveness.
- Development of a retrieval system based on novel similarity and ranking criteria that considers the type, size, location, and shape of various color-texture regions in the query and database images.
- Demonstration of the effectiveness of the proposed approach on two databases containing (1) synthetic mosaics of natural textures and (2) natural scenes.

1.1. Related studies

There are two main aspects to this work. One is segmentation of color texture images, and the other is the retrieval of like images from a database based on a query image. Accordingly, previous studies related to these two topics are reviewed in this section.

The texture segmentation algorithm in Ref. [4] considers features extracted with a 2-D moving average (MA) approach. The 2-D MA model represents a texture as an output of a 2-D finite impulse response (FIR) filter with simple input process. The 2-D MA model is used for modeling both isotropic and anisotropic textures. The maximum likelihood (ML) estimator of the 2-D MA model is used as texture features. Supervised and unsupervised texture segmentation are considered. The texture features extracted by the 2-D MA modeling approach from sliding windows are classified with a neural network for supervised segmentation, and are clustered by a fuzzy clustering algorithm for unsupervised texture segmentation.

Mirmehdi and Petrou in Ref. [5] present an approach to perceptual segmentation of color-image textures. Initial segmentation is achieved by applying a clustering algorithm to the image at the coarsest level of smoothing. The image pixels representing the core clusters are used to form 3-D color histograms that are then used for probabilistic assignment of all other pixels to the core clusters to form larger clusters and categorize the rest of the image. The process of setting up color histograms and probabilistic reassignment of the pixels to the clusters is then propagated through finer levels of smoothing until a full segmentation is achieved at the highest level of resolution.

Deng and Manjunath [2] also present a new method (JSEG) for unsupervised segmentation of color-texture regions in images. It consists of two independent steps: color quantization and spatial segmentation. In the first step, colors in the image are quantized to several representative classes that can be used to differentiate regions in the image. Applying the criterion to local windows in the class-map results in the "J-image," in which high and low values corresponded to possible boundaries and interiors of color-texture regions. A region growing method is then used to segment the images based on the multiscale J-images. This method is still pixel based, and we contrast results of image segmentation of this algorithm with our approach later in this paper.

Much work can be found in the literature regarding matching and retrieval of color patterns from a database. For instance, Mojsilović et al. [6] propose a perceptually based system for pattern retrieval and matching. The central idea is to model similarity judgement along perceptual dimensions. They detected basic visual categories that people use in judgement of similarity, and designed a computational model that accepts patterns as input, and depending on the query, produces a set of choices that follow human behavior in pattern matching.

Saber and Tekalp study in Ref. [7] presents algorithms for automatic image annotation and retrieval based on color, shape, texture, and any combination of two or more of these features. Pixel- or region (object)-based color; region-based shape; and block- or region-based texture features were considered. Automatic region selection was accomplished by integrating color and spatial edge features. Color, shape, and texture indexing was knowledge based (using appropriate training sets) or by example. The multi-feature integration algorithms were designed to: (i) offer the user a wide range of options and flexibility in order to enhance the outcome of the search and retrieval operations, and (ii) provide a compromise between accuracy and computational complexity, and vice versa. The authors demonstrated the performance of the proposed algorithms on a variety of images. It should be noted that this method uses supervised pattern recognition schemes whereas the approach described in this work is based on completely unsupervised approaches.

Fuh et al. [8] propose a model for content-based image retrieval using the idea of combining color segmentation with relationship trees and a corresponding tree-matching method. They retain the hierarchical relationship of the regions in an image during segmentation. In retrieval, they compare not only region features but also region relationships.

Retrieval of multimedia data from a database based on a seed or example for the search/query requires the use of similarity metrics and similarity assessment. Santini and Jain look at this issue in Ref. [9]. They propose a definition of similarity as an operation, and develop a similarity measure based on fuzzy logic that exhibits several features that match experimental findings in humans. The model was dubbed *Fuzzy features contrast* (FFC) and is an extension of the feature contrast model due to Tversky [10]. It is also shown how the FFC model can be used to model similarity assessment from fuzzy judgement of properties, and the use of fuzzy measures to deal with dependencies among the properties is addressed.

Li et al. [11] present a novel similarity measure for region-based image similarity comparison called integrated region matching (IRM). The images are represented by a set of regions, roughly corresponding to objects, which are characterized by features reflecting color, texture, shape, and location properties. The IRM measure for evaluating

overall similarity between images incorporates properties of all the regions in the images by a region-matching scheme. Compared with retrieval based on individual regions, the overall similarity approach reduces the influence of inaccurate segmentation, helps to clarify the semantics of a particular region, and enables simple querying interface for region-based image retrieval systems.

Smith and Chang [3] have also presented a system for image retrieval using regions and their spatial and feature attributes. This method, however, does not make use of the texture information in the images, which sometimes results in image over segmentation. Additionally, it is not a totally unsupervised query system and the user has to input some a priori determined parameters for the search. The query is not driven by the input of the query image to the system and the user has to enter a description for the image on line. The database has to be characterized for the possible types of queries that can be submitted as well. Our approach does not have any of these limitations.

The International Business Machines (IBM) corporation has implemented a commercial system called QBIC™ (Query by Image Content) [12]. This system lets the user make queries of large image databases based on visual image content—properties such as color percentages, color layout, and textures occurring in the images. Such queries use the visual properties of images, so that the user can match colors, textures and their positions without describing them in words. This system, however, does not allow the user to query the database by providing an image example as a query for image matching and retrieval as it does ours.

2. Color-texture characterization with multispectral simultaneous autoregressive model

In this work, the texture of the color images is characterized using a class of multispectral random field image model called the multispectral simultaneous autoregressive (MSAR) model [13,14]. The MSAR model has been shown to be effective for color-texture synthesis and classification [13,14]. For mathematical simplicity, the model is formulated using a toroidal lattice assumption. A location within a two-dimensional $M \times M$ lattice is denoted by $\mathbf{s} = (i, j)$, with i, j being integers from the set $J = \{0, 1, \ldots, M-1\}$. The set of all lattice locations is defined as $\Omega = \{\mathbf{s} = (i, j) : i, j \in J\}$. The value of an image observation at location $\mathbf{s}$ is denoted by the vector value $\mathbf{y}(\mathbf{s})$, and the image observations are assumed to have zero mean. The MSAR model relates each lattice position to its neighboring pixels, both within and between image planes, according to the following model equation:

$$y_i(s) = \sum_{j=1}^{P} \sum_{r \in N_{ij}} \theta_{ij}(r) y_j(s \oplus r) + \sqrt{\rho_i} w_i(s), \quad i = 1, \ldots, P,$$

where $y_i(s)$ is the Pixel value at location s of the ith plane, s and r are the two dimensional lattices, P the number of image planes (for color images, $P = 3$, representing: red, green, and blue planes), N_{ij} the neighbor set relating pixels in plane i to neighbors in plane j (only interplane neighbor sets, i.e. $N_{ij}, i \neq j$, may include the $(0,0)$ neighbor), θ_{ij} the coefficients which define the dependence of $y_i(s)$ on the pixels in its neighbor set N_{ij}, ρ_i the noise variance of image plane i, $w_i(s)$ the i.i.d. random variables with zero mean and unit variance, and $\oplus$ denotes modulo M addition in each index.

The parameters associated with the MSAR model are θ and ρ vectors which collectively characterize the spatial interaction between neighboring pixels within and between color planes. These vectors are taken as the feature set $\mathbf{f}_T$ representing the underlying color-texture of the image.

A least-squares (LS) estimate of the MSAR model parameters is obtained by equating the observed pixel values of an image to the expected value of the model equations [13]. This leads to the following estimates:

$$\hat{\theta}_i = \left[\sum_{\mathbf{s} \in \Omega} \mathbf{q}_i(\mathbf{s}) \mathbf{q}_i^{\mathrm{T}}(\mathbf{s})\right]^{-1} \left[\sum_{\mathbf{s} \in \Omega} \mathbf{q}_i(\mathbf{s}) y_i(\mathbf{s})\right]$$

and

$$\hat{\rho}_i = \frac{1}{M^2} \sum_{\mathrm{s} \in \Omega} (y_i(\mathbf{s}) - \hat{\theta}_i^{\mathrm{T}} \mathbf{q}_i(\mathbf{s}))^2,$$

where

$$\boldsymbol{\theta}_i = [\boldsymbol{\theta}_{i1}^{\mathrm{T}} \boldsymbol{\theta}_{i2}^{\mathrm{T}} \cdots \boldsymbol{\theta}_{iP}^{\mathrm{T}}]^{\mathrm{T}}.$$

$$\mathbf{q}_i(\mathbf{s}) = [\mathbf{y}_{i1}^{\mathrm{T}}(\mathbf{s}) \mathbf{y}_{i2}^{\mathrm{T}}(\mathbf{s}) \cdots \mathbf{y}_{iP}^{\mathrm{T}}(\mathbf{s})]^{\mathrm{T}}$$

$$\mathbf{y}_{ij}(\mathbf{s}) = col\{y_j(\mathbf{s} \oplus \mathbf{r}) : \mathbf{r} \in N_{ij}\}.$$

Fig. 1 shows the effectiveness of this model for color-texture synthesis. The top row shows the original images, and the bottom row includes the images synthesized from the MSAR least-square estimates from the original image samples. The synthetic images can be visually compared to the original images, and observed to be very similar in appearance to the images from which the model was derived.

3. Color content characterization

In addition to modeling color-texture, the general color content of the image is also important. Additional features focusing on the color alone are also considered. This is done using the sample mean of the pixel values in the red, green, and blue (RGB) planes. The defined feature vector is

$$\mathbf{f}_C = \left\{\frac{\hat{\mu}_r}{\hat{\mu}_g}, \frac{\hat{\mu}_r}{\hat{\mu}_b}\right\}$$

with $\hat{\mu}_i$'s being the sample mean of the respective color component. The reason for using these specific ratios, instead of some of the other ratios formed by the other combinations of the color means, is that this same relationship was used to form ratios of the ρ parameters of the MSAR model used

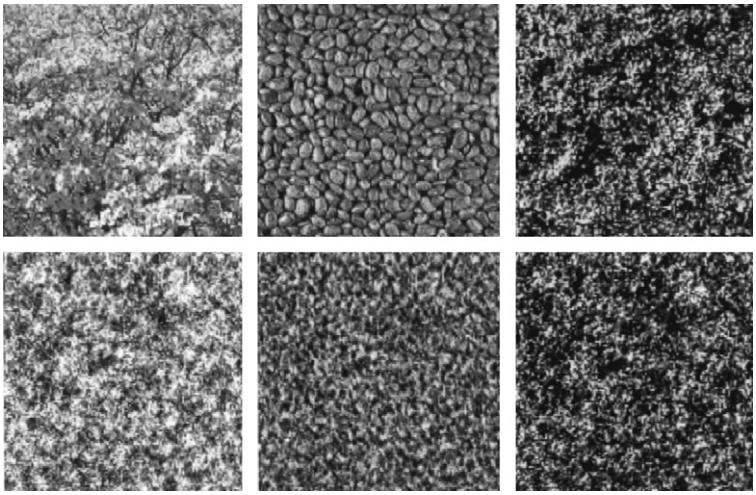

Fig. 1. MSAR model image synthesis results: (top row) original color texture images, (bottom row) synthesized images using MSAR parameters estimated from the original images.

for texture classification in Ref. [13] with very positive results. Also, the reason for using ratio of color means instead of color means themselves is that such a ratio is illumination invariant. Assuming that the observed value at each pixel is a product of illumination and spectral reflectance, the ratios of the color means are invariant to uniform changes in illumination intensity (i.e. the power of the illumination source changes uniformly across the spectrum). This kind of uniform change would cause each $\hat{\mu}_i$ to change by the same scale factor making the defined ratios invariant to illumination changes. This property makes the color-content features more robust. In the event that the denominator of any of the ratios of the color means goes to zero, the color mean with a value of zero is changed to a value of one to avoid the mathematical exception of dividing by zero. This case, however, is very unlikely, since we are dealing with textures and natural images that do not tend to have large areas (i.e. the feature extraction sliding window to be described later) with a solid color extreme.

The combination of $\mathbf{f}_C$ and $\mathbf{f}_T$ features is used to represent a color-texture region in this work. These features are collectively referred to as color content, color texture (C^3T) features.

4. Unsupervised segmentation with a histogram-based clustering algorithm

The first step in the proposed retrieval process is to segment the query image into regions of uniform color-texture. It should be noted that since the ultimate goal is to retrieve images similar to the one presented to the system, it is necessary to find dominant regions of texture in the image but locating very exact boundaries of such regions is not as critical.

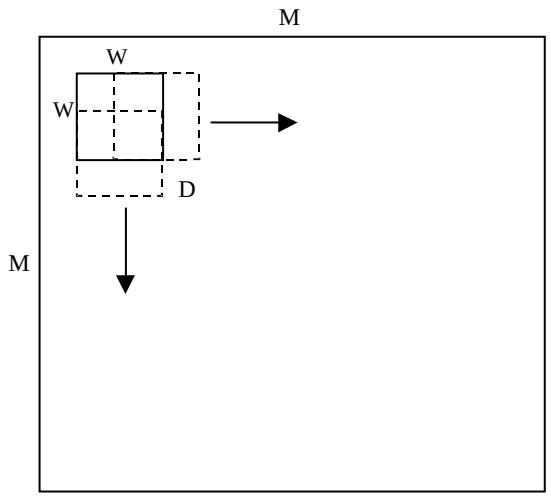

Fig. 2. Feature vector extraction with a sliding window.

The segmentation algorithm used in this work relies on scanning the image with a sliding window and extracting C^3T features from each window. These features are then clustered using an unsupervised histogram-based algorithm. Mapping the identified clusters back into the image domain results in the desired segmentation.

4.1. Feature extraction with a sliding window

The windowing operation consists of sliding a window from left to right and top to bottom across the image as illustrated in Fig. 2. M is the size of the image in pixels, W is the size of the window in pixels, and D is the size of the sliding step in pixels. After extensive experimentation,

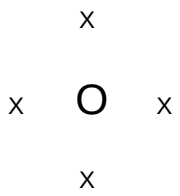

Fig. 3. Neighbor set used with the MSAR model.

where the value of D is varied from 1 pixel to W pixels, D is set to 4 pixels for this work, as this value yielded the best results. With D having a value of W pixels, the sliding windows are non-overlapping and adjacent to each other. To find the optimum window size for each case, the size of the window W varies from 4 to 28 pixels in increments of 4 pixels. The best W is found automatically as described in later sections.

As the window size is increased, the overlap between areas covered by adjacent windows increases, since D is constant, and thus the redundancy of information from feature vectors obtained from larger adjacent windows increases as W increases. It was decided not to increase D for the larger W values, and thus not to reduce the spatially adjacent vectors redundancy, because as W increases the likelihood of capturing a more heterogeneous image area increases as well. Leaving redundancy between adjacent vectors helps the cohesion and convergence of the clustering process, from a spatial perspective.

The texture bounded by each window is characterized using the C^3T features. The neighborhood used for the MSAR model is a set that contains neighbors above, below, to the left, and to the right of the pixel as illustrated in Fig. 3. The same neighbor set is used for both inter- and intra-planes of the model.

This neighbor set results in a 20-dimensional $\mathbf{f}_T$. Therefore, together with the two-dimensional color content feature set, a 22-dimensional C^3T feature vector, $\mathbf{f}$, is used to characterize each window.

4.2. Clustering algorithm

Once all 22-dimensional $\mathbf{f}$ features are extracted from the sliding window, they are clustered in the feature space using an unsupervised histogram-based peak climbing algorithm [15,16]. The 22-dimensional histogram is generated by quantizing each dimension according to the following:

$$CS(k) = \frac{f_{\max}(k) - f_{\min}(k)}{Q}, \quad k = 1, 2, \ldots, N,$$

$$d_k = INT\left\{\frac{f(k) - f_{\min}(k)}{CS(k)} + 1\right\}, \quad k = 1, 2, \ldots, N,$$

where N is the total number of features (22 in this case), $CS(k)$ the length of the kth side of histogram cell, $f_{\max}(k)$ the maximum value of the kth C^3T features, $f_{\min}(k)$ the minimum value of the kth C^3T features, Q the total number of quantization levels, and d_k the kth index for a histogram cell.

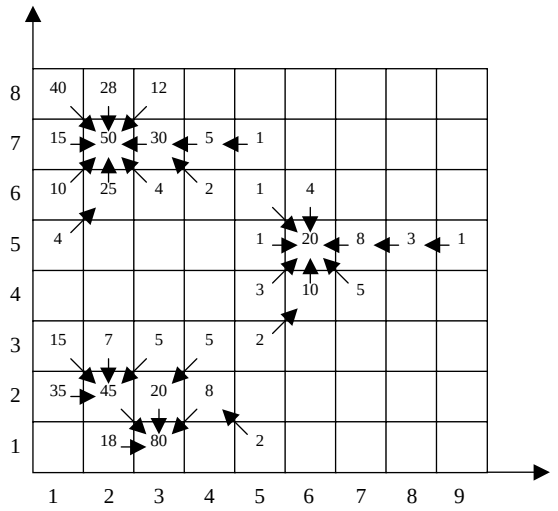

Fig. 4. Illustration of the Peak Climbing approach for a two-dimensional feature space example.

Since the dynamic range of the vectors in each dimension can be quite different, the cell size for each dimension would be different. Hence the cells will be hyperboxes. Next, the number of feature vectors falling in each hyperbox is counted and this count is associated with the respective hyperbox creating the required histogram.

After the histogram is generated in the feature space, a peak climbing clustering approach is utilized to group the features into distinct clusters. This is done by locating the peaks of the histogram. In Fig. 4, this peak climbing approach is illustrated for a two-dimensional space example.

The number in each cell (hyperbox) represents a hypothetical count for the feature vectors captured by that cell. By examining the counts of the 8-neighbors of a particular cell, a link is established between that cell and the closest cell having the largest count in the neighborhood. At the end of the link assignment, each cell is linked to one parent cell, but can be parent of more than one cell. A peak is defined as being a cell with the largest density in the neighborhood, i.e. a cell with no parent. A peak and all the cells that are linked to it are taken as a distinct cluster representing a mode in the histogram. Once the clusters are found, the windows associated with features grouped in the same cluster are tagged as belonging to the same category.

A major component of this algorithm is the number of quantization levels associated with each dimension. To decide this parameter, the total number of non-empty cells and the percentage of them capturing only one vector for each selection of quantization levels are examined. The best number of quantization levels is selected as the largest one that

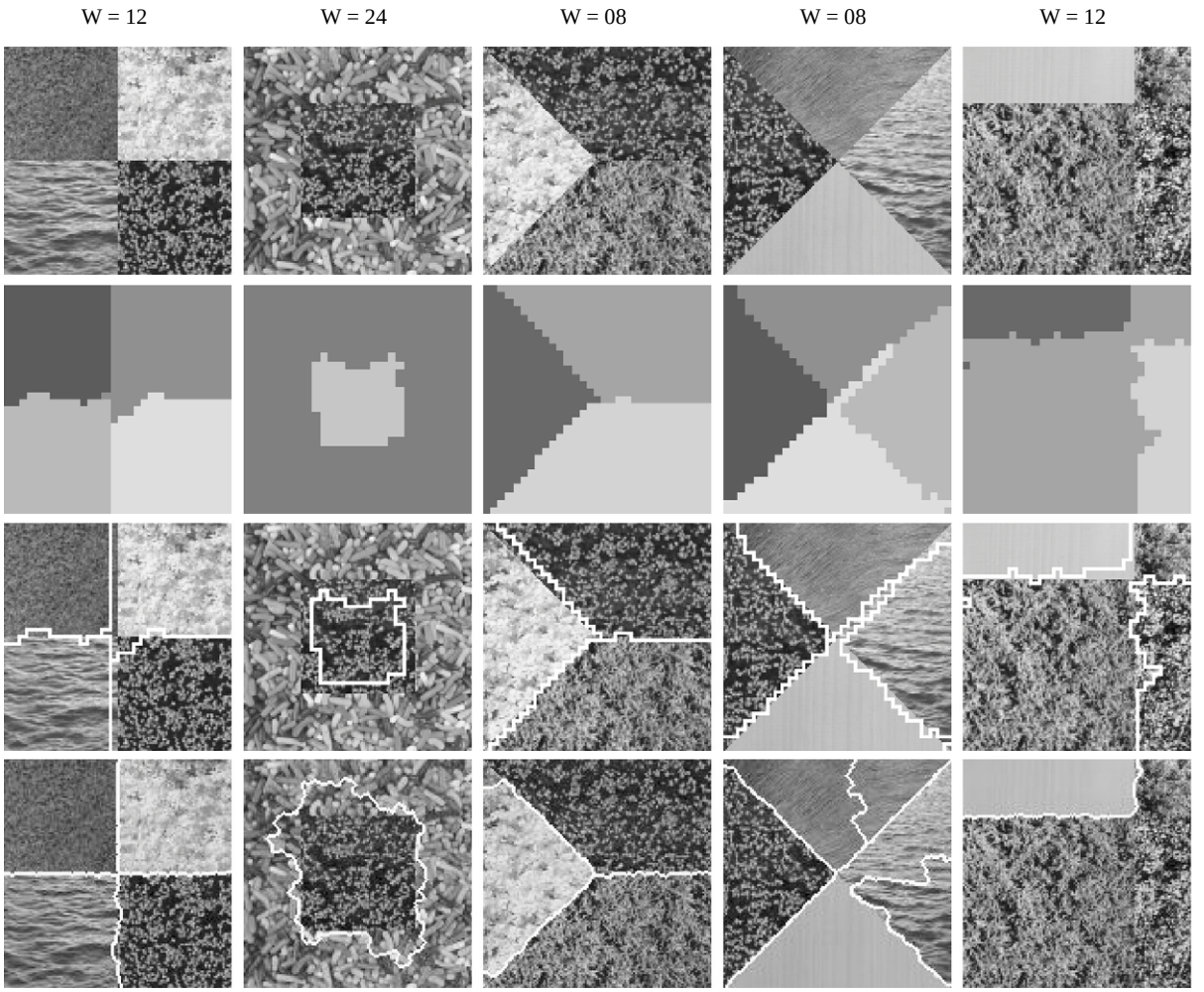

Fig. 5. Segmentation results for four natural texture mosaic images: (1st row) original image, (2nd row) segmentation results, (3rd row) texture boundaries corresponding to segmentation results, (4th row) segmentation using JSEG method.

maximizes the measure below [16].

$$M_i = (N_{ci} - N_{ui}) \times N_{ui},$$

where N_{ci} is the number of non-empty cells, and N_{ui} the number of cells capturing only one sample.

The algorithm also includes a spatial domain cluster validation step. This step involves constructing a matrix B for each cluster m as

$$B_m(i,j) = 1 \quad \textit{if sample} \in \textit{cluster } m$$

$$B_m(i,j) = 0 \quad \textit{otherwise.}$$

The (i,j) index corresponds to the location of a sliding window. A cluster is considered compact if only a very small number of its 1-elements have a 0-element neighbor, i.e. a cluster is considered valid (compact) if only a very small number of its elements have neighboring elements that do not belong to that cluster. A cluster that does not pass this test is merged with a valid cluster that has the closest centroid to it.

During the segmentation process, the best window size for scanning the image is chosen in an unsupervised fashion. The optimum window size is obtained by sweeping the image with varying window sizes (4 to 28 pixels in steps of 4 pixels), and choosing the smallest one out of at least two consecutive window sizes that produce the same number of clusters.

4.3. Segmentation results

The performance of the proposed segmentation algorithm and the associated features is illustrated in Figs. 5 and 6. Fig. 5 shows five images each containing a number of different textures. These image mosaics are created from texture samples available in Ref. [17]. Below each image the

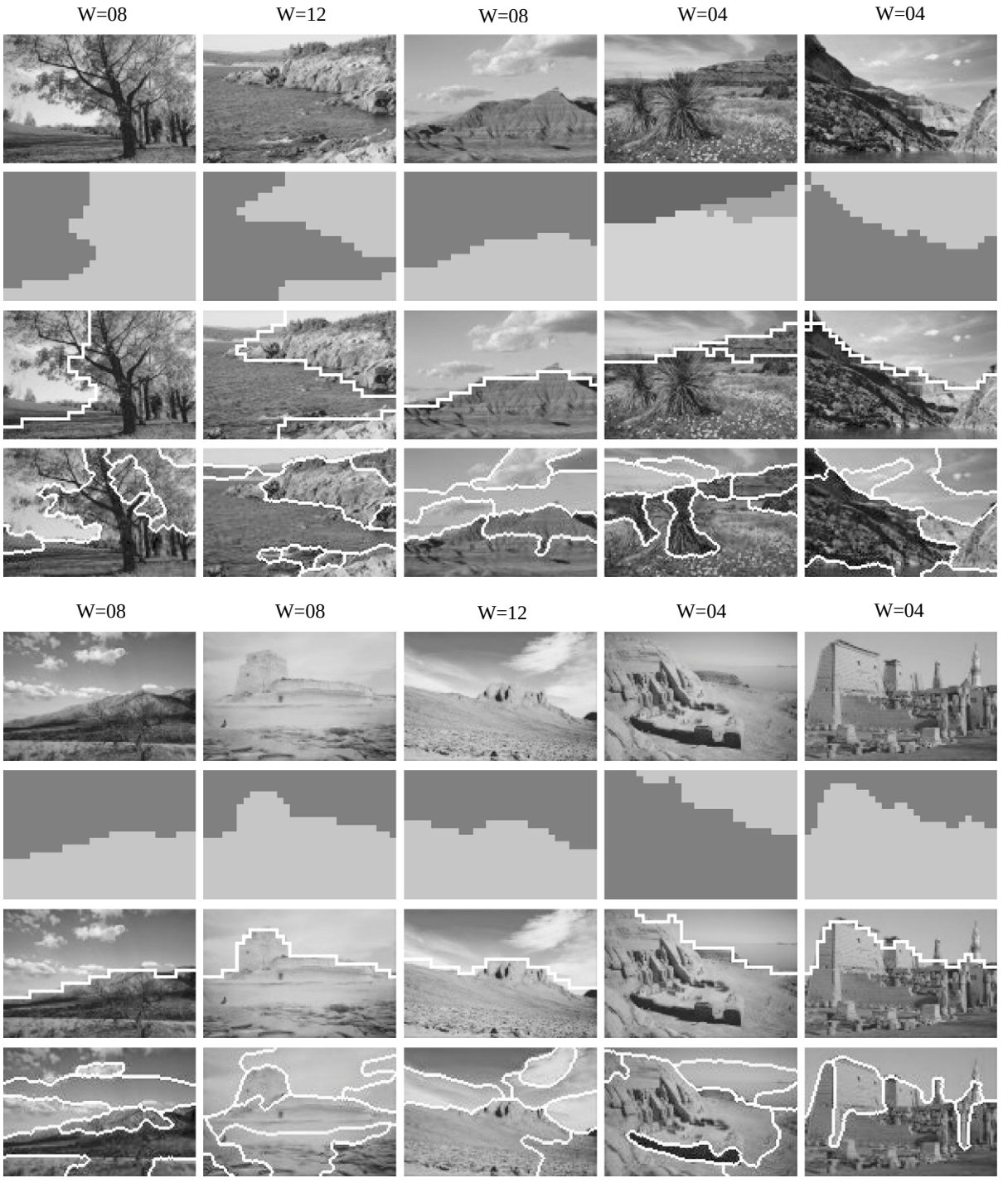

Fig. 6. Segmentation results for eight natural scene images, (1st row) original image: (2nd row) segmentation results, (3rd row) texture boundaries corresponding to segmentation results, (4th row) segmentation using JSEG method.

segmentation result is presented in the form of a gray-level image with pixels belonging to the same texture having the same gray level. In the next row, the boundaries of the segmented regions are shown as superimposed white lines. At the top of the figures, the size of the optimal window found by the algorithm is shown. Fig. 6 shows the segmentation

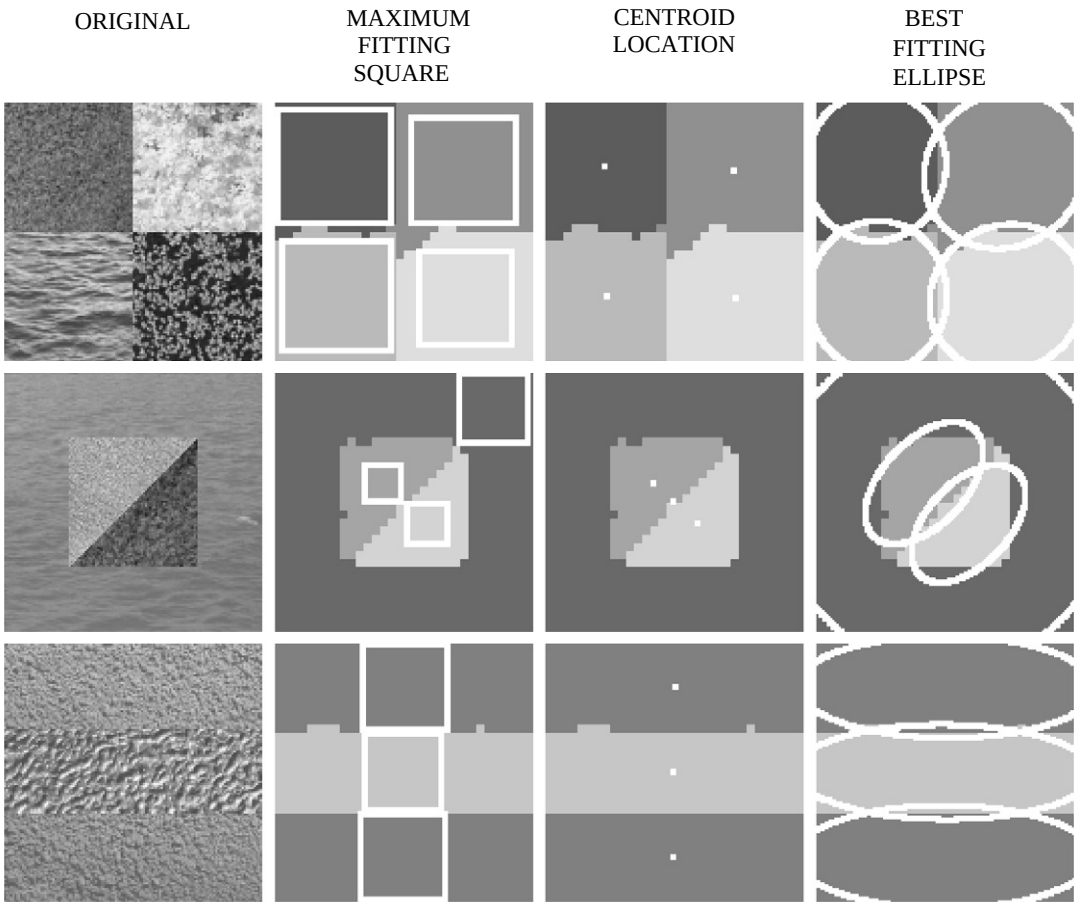

Fig. 7. Examples of texture and shape attribute computation for texture mosaic images.

results for several natural scene images. These natural scene images are available in Ref. [18]. It is observed that the proposed algorithm performs quite well and is capable of localizing uniform color textures in each image.

In Figs. 5 and 6, we also compare the results of our approach with the image segmentation results achieved using the JSEG method described in Ref. [2]. The JSEG results were obtained from applying the images to the programs made available by the JSEG authors at the Internet site http://maya.ece.ucsb.edu/JSEG/. The obtained region boundaries are superimposed on the original images. The JSEG results are displayed in the last rows of Figs. 5 and 6. It can be seen that our segmentation results have a better match with perceptual boundaries in the images. The JSEG method over segments most of the natural scene images and misses or mislabels some boundaries in mosaic images. However, the approach proposed in this work is 3 to 5 times more computationally intensive than the JSEG method, e.g. it takes about 15 s to segment a 128×128 pixels image with this method on a Pentium II 400MHz processor versus about 5 s with the JSEG method.

5. Attributes used in similarity metrics

Once the images are segmented into regions of distinct color texture, similarity metrics need to be developed to measure how close two images are with respect to their color texture content. The attributes used for similarity computation are described in this section.

The foremost attribute is the type of color texture of each segmented region. To characterize this texture, the largest square that could be fitted to the region is found. This square will be referred to as the "maximum fitting square" (MFS). The C^3T features are then extracted from the MFS and used to characterize the color texture of the entire segmented region. The reason for using the MFS is that the shape of the segmented region could be irregular which will not be suitable for the MSAR model computation.

Several shape-related parameters are then extracted from the entire region. These are:

1. Centroid location, which is the mean of the row and column positions of the pixels contained in the region, i.e.

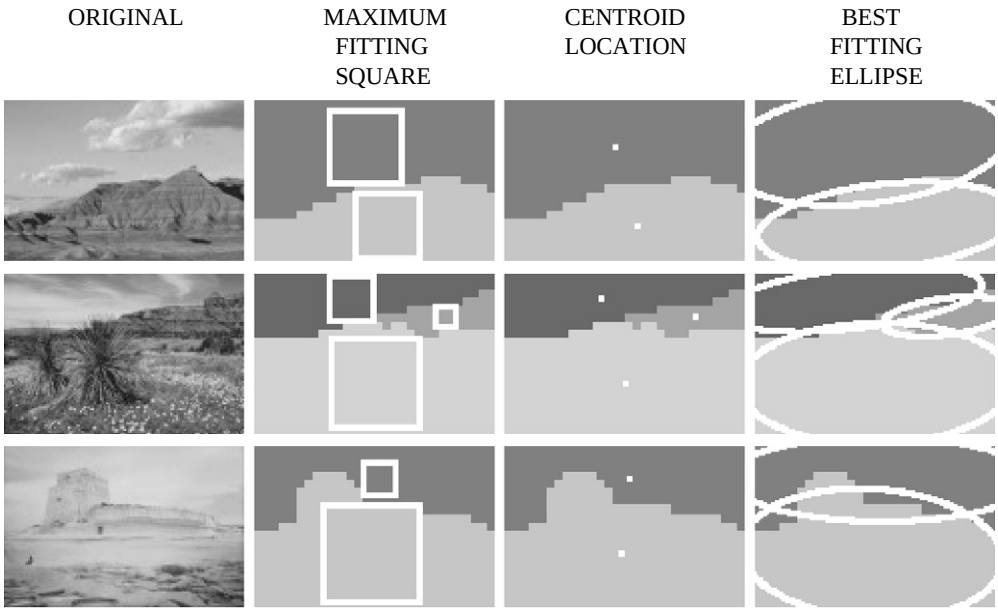

Fig. 8. Examples of texture and shape attribute computation for natural scene images.

$(\bar{x}, \bar{y})$. The centroid location conveys information about the overall position of the region.

2. Area (A) which is the total number of pixels contained by the region.
3. Overall shape as measured by the moments that define the best fitting ellipse to a region, i.e. the largest ellipse that could be fitted to the region. Such an ellipse could be found by computing three geometrical moments, μ_{xx}, μ_{yy}, and μ_{xy} where

$$\mu_{xx} = \sum_{OVER\ ROWS} \sum_{OVER\ COLUMNS} \frac{(x-\bar{x})^2}{A},$$

$$\mu_{yy} = \sum_{OVER\ ROWS} \sum_{OVER\ COLUMNS} \frac{(y-\bar{y})^2}{A},$$

$$\mu_{xy} = \mu_{yx} = \sum_{OVER\ ROWS} \sum_{OVER\ COLUMNS} \frac{(x-\bar{x})(y-\bar{y})}{A}.$$

These three shape-based measures along with the C^3T features extracted from the MFS are used in the similarity metric developed for the retrieval process.

Figs. 7 and 8 illustrate this process for three texture mosaics and three natural scene images, respectively. In each figure, the original images are shown in the first column followed by the computed MFS' superimposed on the segmented regions. The centroid locations, and the best fitting ellipses in images are shown in columns three and four, respectively.

6. Search space reduction

The next step after extracting the texture and shape attributes is to reduce the number of database images that are to be considered in the retrieval process. This is done by finding those images that contain similar textures to that of the query image. This task is carried out by using the C^3T features extracted from the MFS. The C^3T features of the database images are first clustered using the clustering algorithm described earlier resulting in M clusters, C_i, $i = 1, 2, \ldots, M$. Then, the C^3T features of each of the segmented regions of the query image are considered in this clustered 22-dimensional space. Let us denote the C^3T features of the kth segmented region of the query image as $\mathbf{f}_k$. For each $\mathbf{f}_k$, the closest cluster center $C_j(\mathbf{f}_k)$ is found and all the database images that are associated with C_j cluster are tagged as images that need to be considered in the search process. After all $\mathbf{f}_k$'s are considered, all the database images that are not tagged are removed from further consideration resulting in a reduced search space. In other words, only those database images that potentially have one or more similar color texture to those of the query image are retained. Note that this process also identifies the likely number of common textures between the query image and each of the retained database images.

7. Region association

In the next phase, the retained database images are considered one at a time and an association is established

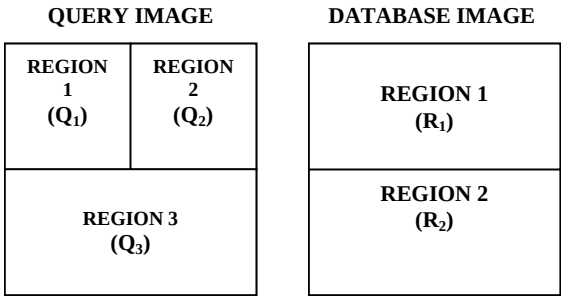

Fig. 9. An example of the Region Association process.

between each region of the query image and one of the regions of the considered database image. This process is illustrated by an example.

Let us assume that the query image contains three segmented regions, Q_1, Q_2, and Q_3 and a database image has two regions denoted by R_1 and R_2 as shown in Fig. 9.

The Euclidean distance between C^3T features of each possible Q_i, R_j pair ($d(Q_i, R_j)$) is computed resulting in the following six distances:

$$\{d(Q_1,R_1), d(Q_1,R_2), d(Q_2,R_1), d(Q_2,R_2), d(Q_3,R_1), d(Q_3,R_2)\}.$$

These six distances are then sorted from smallest to largest. Let us assume that this sorting results in the following order:

$$\{d(Q_3,R_1), d(Q_2,R_1), d(Q_3,R_2), d(Q_1,R_2), d(Q_1,R_1), d(Q_2,R_2)\}.$$

An association between each Q_i and R_j is established by considering this sorted list and making the association according to the first occurrence of each Q_i in the list. In this example, the following associations are made

$$Q_3 \leftrightarrow R_1 \quad Q_2 \leftrightarrow R_1 \quad Q_1 \leftrightarrow R_2.$$

Upon completion of this step, the best potential match between each region of the query image and one of the regions of each of the database images is established.

8. Similarity computation

The similarity between the query image and a database image is computed using the region associations established in the previous phase. In this stage, the shape-based attributes are utilized to arrive at a final similarity measure. The Euclidean distances between the position, area, and shape attributes of each associated pair are computed. Continuing with the example from the previous section, the computed distances will be:

Q_3 associated with R_1 : $d(P_{Q_3}, P_{R_1})$, $d(A_{Q_3}, A_{R_1})$, $d(S_{Q_3}, S_{R_1})$,
Q_2 associated with R_1: $d(P_{Q_2}, P_{R_1})$, $d(A_{Q_2}, A_{R_1})$, $d(S_{Q_2}, S_{R_1})$,
Q_1 associated with R_2: $d(P_{Q_1}, P_{R_2})$, $d(A_{Q_1}, A_{R_2})$, $d(S_{Q_1}, S_{R_2})$,

where $d(P_{Q_i}, P_{R_j})$ is the Euclidean distance between centroid locations of Q_i and R_j, $d(A_{Q_i}, A_{R_j})$ the Euclidean distance between areas of Q_i and R_j, and $d(S_{Q_i}, S_{R_j})$ the Euclidean distance between 3-D vector of $(\mu_{xx}, \mu_{yy}, \mu_{xy})$ of Q_i and R_j.

Next, the position, area, and shape distances for all Q_i's are added together to form total position, area, and shape difference measures denoted by PD, AD, and SD, respectively.

$$PD = d(P_{Q_3}, P_{R_1}), +d(P_{Q_2}, P_{R_1}) + d(P_{Q_1}, P_{R_2}),$$
$$AD = d(A_{Q_3}, A_{R_1}) + d(A_{Q_2}, A_{R_1}) + d(A_{Q_1}, A_{R_2}),$$
$$SD = d(S_{Q_3}, S_{R_1}) + d(S_{Q_2}, S_{R_1}) + d(S_{Q_1}, S_{R_2}).$$

At this point a total texture difference measure based on the Euclidean distance of the C^3T features is also computed and denoted as TD.

$$TD = d(C^3T_{Q_3}, C^3T_{R_1}) + d(C^3T_{Q_2}, C^3T_{R_1}) + d(C^3T_{Q_1}, C^3T_{R_2}).$$

Finally, a composite distance measure used as the overall similarity measure between the two images is computed. This measure denoted as S is

$$S = \sqrt{PD^2 + AD^2 + SD^2 + TD^2}.$$

9. Retrieval

The S measure reflects the degree of similarity between the query image and the considered database image. As such, the retrieval process involves sorting all the S measures computed between the query image and each of the database images in the reduced search space. The database images ordered from smallest to largest S will be the most similar to least similar to the query image, respectively.

The performance of the system on two different databases is reported in the next three sections. In each experiment, the query image is taken to be one of the database images, i.e., there is an exact copy of the query image in the database to be searched. In each case, the exact copy is always found as the most similar image. This is an important affirmation of the validity of the proposed algorithm.

9.1. Test databases

Two databases are used to test the performance of the proposed approach. These databases are referred to as Natural Texture Mosaics Database, and Natural Scenes Database.

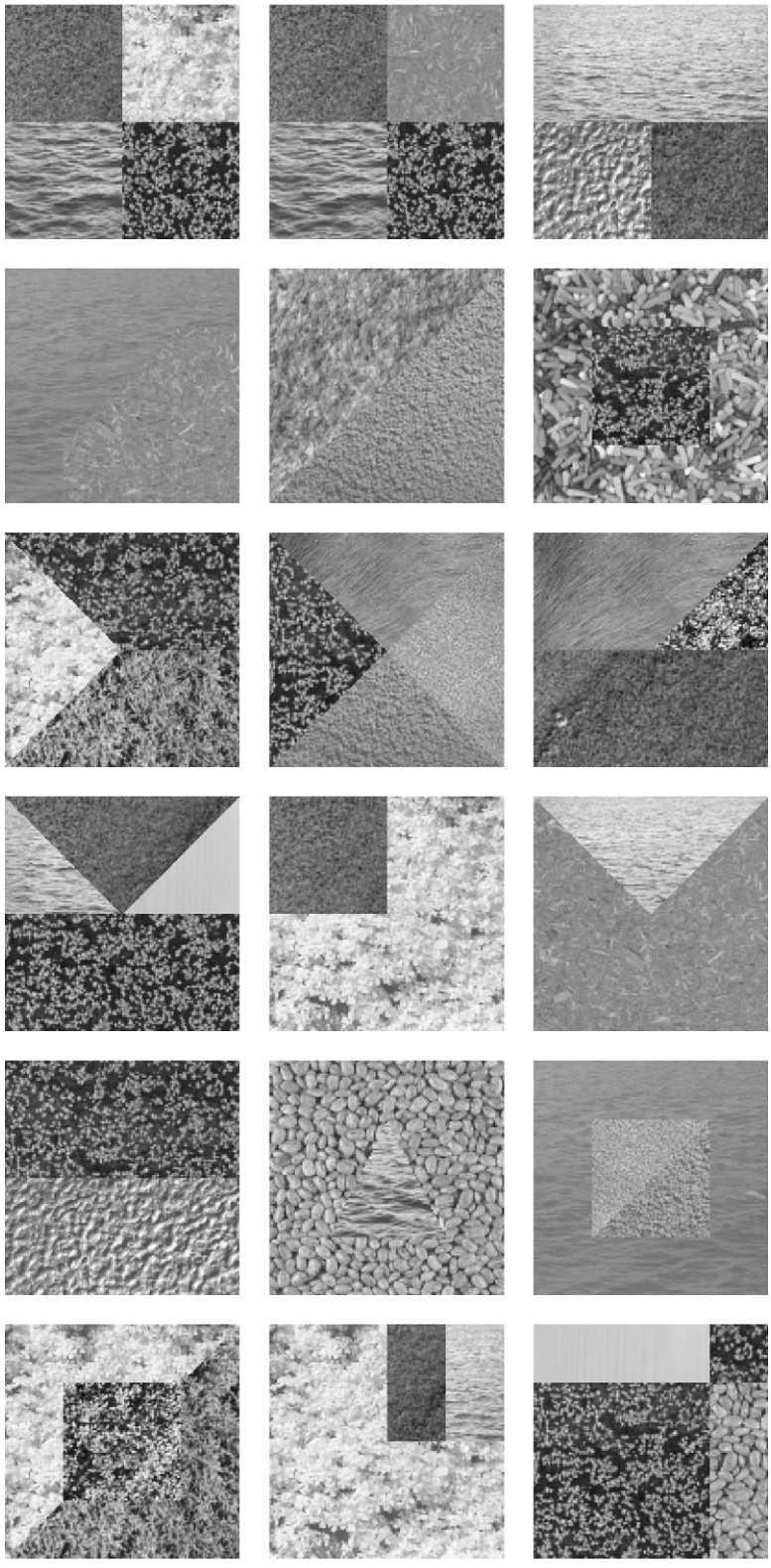

Fig. 10. Samples of images in the Natural Textures Mosaics Database.

Fig. 11. Samples of images in the Natural Scenes Database.

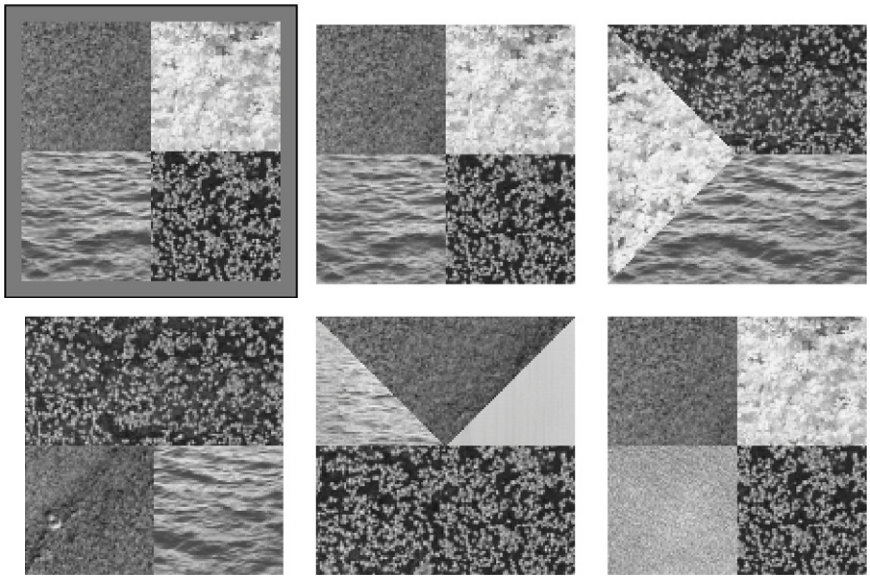

Fig. 12. First example of retrieval results for query of the Natural Texture Mosaics Database, the image at the top left with blue border is the query image, the other images from left to right, top to bottom are retrieved images with decreasing similarity.

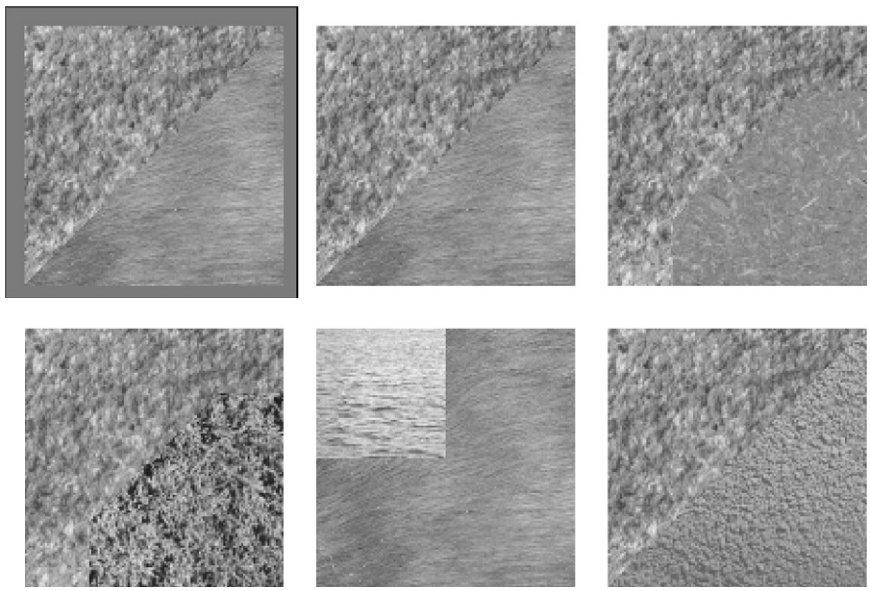

Fig. 13. Second example of retrieval results for query of the Natural Texture Mosaics Database.

9.1.1. Natural Texture Mosaics Database

The Natural Texture Mosaics Database is a collection of images that are mosaics of natural textures constructed from texture images available in Ref. [17]. This collection includes 200 128×128 images with mosaics put together at random in terms of the arrangement and the type of the texture regions used. This database is constructed to measure and investigate the effectiveness of the algorithm in a controlled environment. Samples of images of this database are shown in Fig. 10.

9.1.2. Natural scenes database

The Natural Scenes Database is a collection of images, which is comprised of natural scenes available in [18]. This collection has 400 images of various sizes that include $120 \times 80, 80 \times 120, 128 \times 85$, and 85×128 pixels. This

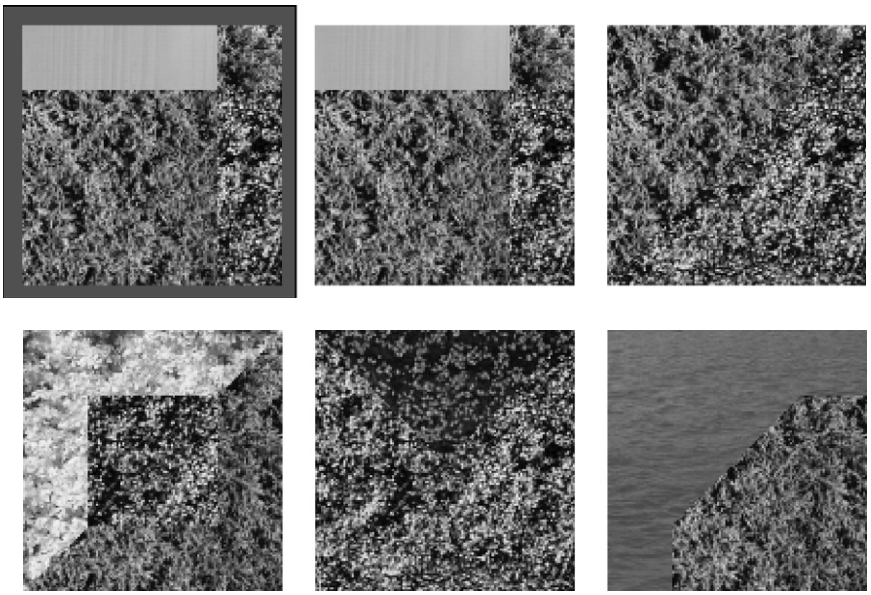

Fig. 14. Third example of retrieval results for query of the Natural Texture Mosaics Database.

Fig. 15. First example of retrieval results for query of the Natural Scenes Database, the image at the top left with blue border is the query image, the other images from left to right, top to bottom are retrieved images with decreasing similarity.

database was constructed to test the performance on real imagery. Samples of images of this database are shown in Fig. 11.

9.2. Retrieval results

A series of experiments is carried out by taking an image from the considered database and using it as the query image. In all the experiments the database image that is the same as the query image is found as the most similar image to the query image.

In Figs. 12–14, examples of the performance for the Natural Texture Mosaics database are shown. In each figure, the image at the top left with the blue border is the query image. The five most similar images retrieved from the database are shown in the order of similarity from left to right and top to bottom. Note that all retrieved images have common texture types with the query image.

In Figs. 15–17, similar examples are shown for Natural Scene database images. Again note that the exact match of the query image is ranked as the closest retrieved image. The other retrieved images all contain similar scenery as the query image.

It should be noted that the segmentation of database images can be done offline. The online retrieval process involving the segmentation of the query image and computation and ranking of the corresponding S measures takes around 60 s on a 400 MHz Intel Pentium III class machine.

Fig. 16. Second example of retrieval results for query of the Natural Scenes Database.

Fig. 17. Third example of retrieval results for query of the Natural Scenes Database.

10. Conclusions

In this work, a color-texture-based approach to image retrieval from a large database is developed. Features derived from the multispectral autoregressive (MSAR) random field model and the RGB color space characterize the color-texture content of the image. These features are used in conjunction with an unsupervised clustering-based segmentation algorithm to segment the images into regions of uniform color texture. Similarity measures based on the color texture content, shape, size, and position of the segmented regions are developed to measure the closeness of a query image to the database image. The effectiveness of the approach has been demonstrated using two different databases containing synthetic mosaics of natural textures and natural scenes.

References

[1] Y. Rui, T.S. Huang, Image retrieval; current techniques, promising directions, and open issues, J. Visual Comm. Image Representation 10 (1999) 39–62.

[2] Y. Deng, B.S. Manjunath, Unsupervised segmentation of color-texture regions in images and video, IEEE Trans. Pattern Anal. Machine Intell. 23 (8) (2001) 800–810.

[3] J.R. Smith, S-F. Chang, Integrated Spatial and Feature Image Query, Multimedia Systems, Vol. 7 (2), ACM © Springer, Berlin, 1999, pp. 129–140.

[4] K.B. Eom, Segmentation of monochrome and color textures using moving average modeling approach, Image Vision Comput. 3 (17) (1999) 233–244.

[5] M. Mirmehdi, M. Petrou, Segmentation of color textures, IEEE Trans. Pattern Anal. Machine Intell. 22 (2) (2000) 142–159.

[6] A. Mojsilović, J. Kovačević J. Hu, R.J. Safranek, S.K. Ganapathy, Matching and retrieval based on the vocabulary and grammar of color patterns, IEEE Trans. Image Process. 9 (1) (2000) 38–54.

[7] E. Saber, A.M. Tekalp, Integration of color, shape, and texture for automatic image annotation and retrieval, Electronic Imaging (special issue) 7 (3) (1998) 684–700.

[8] C.S. Fuh, S.W. Cho, K. Essig, Hierarchical color image region segmentation for content-based image retrieval system, IEEE Trans. Image Process. 9 (1) (2000) 156–162.

[9] S. Santini, R. Jain, Similarity measures, IEEE Trans. Pattern Anal. Machine Intell. 21 (9) (1999) 871–883.

[10] A. Tversky, Features of similarity, Psychol. Rev. 84 (4) (1977) 327–352.
[11] J. Li, J.Z. Wang, G. Wiederhold, IRM: integrated region matching for image retrieval, Proceedings of the ACM Multimedia 2000 Conference, Los Angeles, CA, 2000, pp. 147–156.
[12] International Business Machines (IBM) Company, QBIC™—IBM's query by image content, http://wwwqbic.almaden.ibm.com/.
[13] J.W. Bennett, Modeling and analysis of gray tone, color, and multispectral texture images by random field models and their generalizations, Ph.D. Dissertation, Southern Methodist University, 1997.
[14] J.W. Bennett, A. Khotanzad, Multispectral random field models for synthesis and analysis of color image, IEEE Trans. Pattern Anal. machine Intell. 20 (3) (1998) 327–332.
[15] A. Khotanzad, J.Y. Chen, Unsupervised segmentation of textured images by edge detection in multidimensional features, IEEE Trans. Pattern Anal. Machine Intell. 11 (4) (1989) 414–421.
[16] A. Khotanzad, A. Bouarfa, Image segmentation by a parallel, non-parametric histogram based clustering algorithm, Pattern Recognition 23 (9) (1990) 961–963.
[17] Vision and Modeling Group, MIT Media Laboratory, Vision Texture (VisTex) database, http://www-white.media.mit.edu/vismod/, 1995.
[18] Corel Corporation, Professional Photos CD-ROM Sampler—SERIES 200000, 1994.

About the Author—ALIREZA KHOTANZAD received his Ph.D. degree in Electrical engineering from Purdue University in 1983. He is currently Professor of Electrical Engineering at Southern Methodist University, Dallas, Texas. His research interests include intelligent systems, computer vision, and neural networks. He served as an associate editor of the IEEE Transactions on Neural Networks from 1995 to 1997. He is currently an associate editor of Pattern Recognition and the IEEE Transactions on Pattern Analysis and Machine Intelligence.

About the Author—ORLANDO J. HERNANDEZ was born on July 8th, 1969 in Havana, Cuba. He attended University of South Florida (USF), where he received a Bachelor of Science in Electrical Engineering and a Master of Science in Electrical Engineering in 1991 and 1993, respectively. He subsequently received his Ph.D. degree in Electrical Engineering from Southern Methodist University, Dallas, Texas in 2002. Since 1993, he has been with Texas Instruments, Inc. in Dallas, Texas where he has held positions in Product Engineering, Applications Engineering, Product Development, Program Management, and Design Management. His research interests include color image segmentation and retrieval, computer vision, and image processing.