Matching and Retrieval Based on the Vocabulary and Grammar of Color Patterns

Aleksandra Mojsilović, *Member, IEEE,* Jelena Kovačević, *Senior Member, IEEE,* Jianying Hu, *Member, IEEE,* Robert J. Safranek, *Senior Member, IEEE,* and S. Kicha Ganapathy, *Member, IEEE*

***Abstract*—We propose a perceptually based system for pattern retrieval and matching. There is a need for such an "intelligent" retrieval system in applications such as digital museums and libraries, design, architecture, and digital stock photography. The central idea of the work is that similarity judgment has to be modeled along perceptual dimensions. Hence, we detect basic visual categories that people use in judgment of similarity, and design 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. There are two major research aspects to our work. The first one addresses the issue of how humans perceive and measure similarity within the domain of color patterns. To understand and describe this mechanism we performed a subjective experiment. The experiment yielded five perceptual criteria used in comparison between color patterns (vocabulary), as well as a set of rules governing the use of these criteria in similarity judgment (grammar). The second research aspect is the actual implementation of the perceptual criteria and rules in an image retrieval system. Following the processing typical for human vision, we design a system to: 1) extract perceptual features from the vocabulary and 2) perform the comparison between the patterns according to the grammar rules. The modeling of human perception of color patterns is new—starting with a new color codebook design, compact color representation, and texture description through multiple scale edge distribution along different directions. Moreover, we propose new color and texture distance functions that correlate with human performance. The performance of the system is illustrated with numerous examples from image databases from different application domains.**

***Index Terms*—Color and texture classification, color and texture extraction, image database retrieval.**

I. Introduction

FLEXIBLE retrieval and manipulation of image databases has become an important problem with application in video editing, photo-journalism, art, fashion, cataloguing, retailing, interactive CAD, geographic data processing, etc. Until recently, content-based retrieval systems (CBR's) have asked people for key words to search image and video databases. Unfortunately, this approach does not work well since different people describe what they see or what they search for in different ways, and even the same person might describe the same image differently depending on the context in which it will be used. These problems stimulated the development of innovative content-based search techniques as well as new types of queries.

A. Previous Work

One of the earliest CBR systems is ART MUSEUM [1], where retrieval is performed entirely based on edge features. The first commercial content-based image search engine with profound effects on later systems was QBIC [2]. As color representation, this system uses a k-element histogram and average of (R, G, B), (Y, i, q), and (L, a, b) coordinates, whereas for the description of texture it implements Tamura's feature set [3]. In a similar fashion, color, texture, and shape are supported as a set of interactive tools for browsing and searching images in the Photobook system developed at the MIT Media Lab [4]. In addition to these elementary features, systems such as VisualSeek [5], Netra [6], and Virage [7] support queries based on spatial relationships and color layout. Moreover, in the Virage system [7], the user can select a combination of implemented features by adjusting the weights according to his own "perception." This paradigm is also supported in RetrievalWare search engine [8]. A different approach to similarity modeling is proposed in the MARS system [9], where the main focus is not in finding a best representation, but rather on the relevance feedback that will dynamically adapt multiple visual features to different applications and different users. Hence, although great progress has been made, none of the existing search engines offers a complete solution to the general image retrieval problem, and there are still many open research issues, preventing their use in a real application. *Why is that so?*

B. Motivation

While it is recognized that images can be described at a metalevel through color, texture, and shape of the objects within the image, general image understanding is a hard problem. Thus, one challenge is to accomplish image retrieval based on similarities in the feature space without necessarily performing full-fledged scene analysis. Many of the existing systems [7], [8], accomplish this task by expecting the user to assign a set of weights to color, shape, and texture features, thus specifying the way these attributes are going to be combined in the algorithm. Unfortunately, certain problems arise from this approach: First, this is certainly not the way matching is performed in the human visual system. Further, humans have no general notion of similarity; instead, they possess a functional notion of similarity within a particular domain. Therefore, to

Manuscript received November 20, 1998; revised July 7, 1999. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. B. S. Manjunath.

The authors are with Bell Laboratories, Lucent Technologies, Murray Hill, NJ 07974 USA (e-mail: saska@research.bell-labs.com).

Publisher Item Identifier S 1057-7149(00)00175-5.

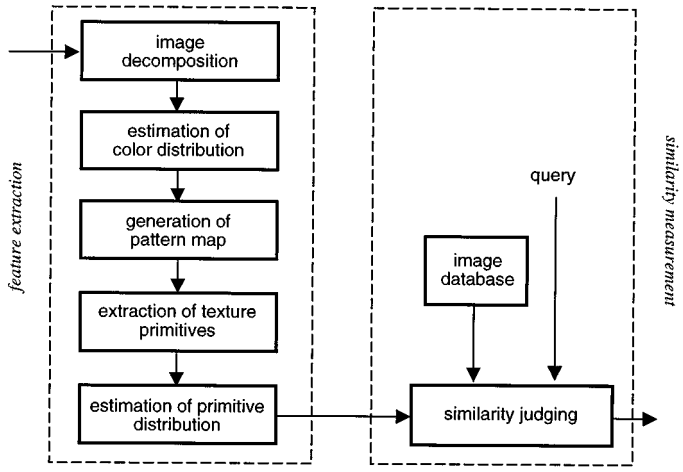

Fig. 1. Overview of the system. The two main parts deal with feature extraction and similarity measurement. Both the feature extraction and similarity measurment parts mimic the behavior of the human visual system. Within the feature extraction part, color and texture are processed separately.

perform similarity matching in a human-like manner one has to: 1) choose a specific application domain, 2) understand how users judge similarity within that domain, and then 3) build a system that will replicate human performance.

Since color and texture are fundamental aspects of human visual perception, we developed a set of techniques for search and manipulation of color patterns. Moreover, there are a great many applications for pattern retrieval in: arts and museums, fashion, garment and design industry, digital libraries, and digital stock photography. Therefore, there is a need for an "intelligent" visual information retrieval system that will perform pattern matching in these applications. However, regardless of the application domain, to accomplish retrieval successfully it is necessary to understand what type of color and texture information humans actually use and how they combine them in deciding whether two patterns are similar. In this paper, we are focusing on the integration of color and texture features for pattern retrieval and matching. Our aim is to detect basic visual categories that people use in judgment of similarity, and then to design a computational model which accepts one (or more) texture images as input, and depending on the type of query, produces a set of choices that follow human behavior in pattern matching.

There are two major research aspects in our work: The first one addresses the issue of how humans perceive and measure similarity within the domain of color patterns. To understand and describe this mechanism we performed a subjective experiment. The experiment yielded five perceptual criteria important for the comparison between the color patterns, as well as a set of rules governing the use of these criteria in the similarity judgment. The five perceptual criteria are considered to be the *basic vocabulary*, whereas the set of rules is considered as the *basic grammar* of the "color pattern language." The second research aspect is the actual implementation of the perceptual criteria and rules in the image retrieval system illustrated in Figs. 1 and 2. Following the processing typical for human vision, we design a system to 1) extract perceptual features from the vocabulary and 2) perform the comparison between the patterns according to the grammar rules. The modeling of human perception of color patterns is new—starting with a new color codebook design, compact color representation, and texture description through multiple scale edge distribution along different directions. Finally, to model the human behavior in pattern matching, instead of using the traditional Euclidean metric to compare color and texture feature vectors, we propose new distance functions that correlate with human performance.

The outline of the paper is as follows. Section II describes the subjective experiment and analytical tools we used to interpret the data. At the end of this section we list and describe in detail the five perceptual categories (vocabulary) and five rules (grammar) used by humans in comparison of color patterns. Section III gives an overview of the system together with its psychophysical background. Sections IV and V present the implementation of feature extraction based on color and texture, respectively, and the development of new color and texture metrics. Section VI describes how these features and distances are used in similarity measurement and presents numerous examples. Section VII gives examples of different queries and the corresponding search results. The final section includes discussion and conclusions.

II. Vocabulary and Grammar of Color Patterns

Our understanding of color patterns is very modest compared to our understanding of other visual phenomena such as color, contrast or even gray-level textures. That is mainly due to the fact that the basic dimensions of color patterns have not yet been identified, a standardized set of features for addressing their important characteristics does not exist, nor are there rules defining how these features are to be combined. Previous investigations in this field concentrated mainly on gray-level natural textures [3], [10], [11]. Particularly interesting is work of Rao and Lohse [11]: their research focused on how people classify textures in meaningful, hierarchically structured categories, identifying relevant features used in the perception of gray-level textures. Similarly, here we determine the basic categories—*vocabulary*—used by humans in judging similarity of color patterns, their relative importance and relationships, as well as the hierarchy of rules—*grammar*. Later in the paper, through numerous search examples (see Figs. 8–13), we will show that these attributes are applicable to a broad range of textures, starting from simple patterns, all the way up to complex, high-level visual texture phenomena.

This section describes the subjective experiment, and gives a brief overview of multidimensional scaling and hierarchical clustering techniques we used to interpret the experimental data. Multidimensional scaling was applied to determine the most important dimensions of pattern similarity, while hierarchical clustering helped us understand how people combine these dimensions when comparing color patterns. The results obtained are listed and explained at the end of this section, while the details can be found in [14].

A. Experimental Setup

During the subjective testing, we used 25 patterns from interior design catalogs. Twenty patterns were used in the actual study, while five patterns were used as a "warm-up" before

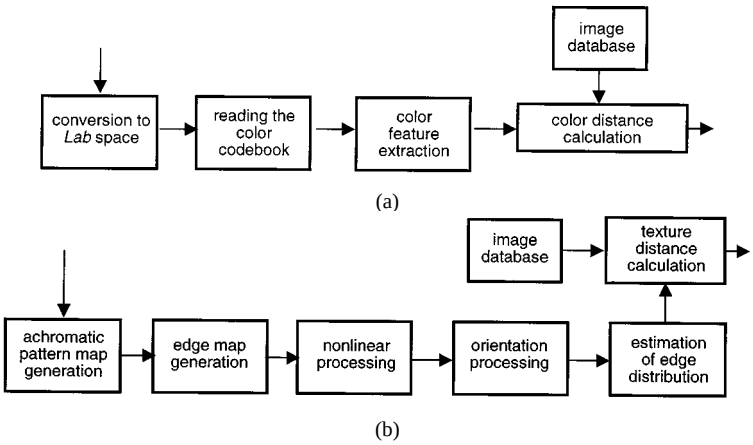

Fig. 2. Two basic blocks of the feature extraction part from Fig. 1. (a) Color representation and modeling. (b) Texture representation and modeling.

Fig. 3. Pattern set used in the experiment. The patterns are obtained from an interior design database, containing 350 patterns. Twenty were selected capturing a variety of features. Another five were used as a "warm up" in the study. The patterns are numbered from 1 through 20, starting at the upper left-hand corner.

each trial. This allowed the subjects to get comfortable with the testing procedure and to sharpen their own understanding of similarity. The digitized version of the twenty patterns selected are displayed in Fig. 3. We selected patterns that capture a variety of different image features and their combinations. The selection of stimuli is crucial for interpretation of the data. Since we postulated that visual similarity needs to be modeled by a high number of dimensions, it was vital for this experiment to select the stimuli so that there is sufficient variation of potential dimensions.

Twenty eight subjects participated in the study. The subjects were not familiar with the input data. They were presented with all 190 possible pairs of stimuli. For each pair, the subjects were asked to rate the degree of overall similarity on a scale ranging from zero for "very different" to 100 for "very similar." There were no instructions concerning the characteristics on which these similarity judgments were to be made since this was the very information we were trying to discover. The order of presentation was different for each subject and was determined through the use of a random number generator. This was done to minimize the effect on the subsequent ratings of both the same presentation order for all the subjects (group effect) as well as the presentation order for one subject (individual effect).

At the end of experiment, half of the subjects were presented with pairs they thought the most similar, and asked to explain why. Their explanations were used later as an aid in the interpretation of the experimental results, as well as for the development of the retrieval system. Experimental data were interpreted using multidimensional scaling techniques yielding the vocabulary and the hierarchical clustering analysis which, in turn, led to the grammar rules.

B. Multidimensional Scaling

Multidimensional scaling (MDS) is a set of techniques that enables researchers to uncover the hidden structures in data [12]. MDS is designed to analyze distance-like data called *similarity* data; that is, data indicating the degree of similarity between two items. Traditionally, similarity data is obtained via subjective measurement. It is acquired by asking people to rank similarity of pairs of objects—*stimuli*—on some scale (as in our experiment). The obtained similarity value connecting stimulus i to stimulus j is denoted by δ_{ij}. Similarity values are arranged in a similarity matrix Δ, usually by averaging δ_{ij} obtained from all measurements. The aim of MDS is to place each stimulus from the input set into an n-dimensional stimulus space (the optimal dimensionality of the space, n, should be also determined in the experiment). The points $\boldsymbol{x}_i = [x_{i1}x_{i2}\cdots x_{\mathrm{in}}]$ representing each stimulus are obtained so that the Euclidean distances d_{ij} between each pair of points in the stimulus space match as closely as possible the subjective similarities δ_{ij} between corresponding pairs of stimuli. The coordinates of all stimuli (i.e., the *configuration*) are stored in the matrix $\boldsymbol{X}$, also called the *group configuration matrix.*

Depending on the type of the MDS algorithm, one or several similarity matrices are analyzed. The simplest algorithm is the classical MDS (CMDS), where only one similarity matrix is analyzed. The central concept of CMDS is that the distance d_{ij} between points in an n-dimensional space will have the strongest possible relation to the similarities δ_{ij} from a single matrix Δ. The traditional way to describe a desired relationship between the distance d_{ij} and the similarity δ_{ij} is by the relation $d = f(\delta)$ such as

$$d = f(\delta) = a\delta + b \tag{1}$$

where, for a given configuration, values a and b must be discovered using numerical optimization. There are many different computational approaches for solving this equation [12]. Once the best f is found, we then search for the best configuration $\boldsymbol{X}$ of points in the stimulus space. This procedure is repeated for different n's until further increase in the number of dimensions does not bring a reduction in the following error function (also known as *stress formula 1 or Kruskal's stress formula*):

$$\text{stress}(\Delta,\, \boldsymbol{X},\, f) = \sqrt{\frac{\sum_i \sum_j [f(\delta_{ij}) - d_{ij}]^2}{\sum_i \sum_j f(\delta_{ij})^2}}. \qquad (2)$$

A detailed introduction to the CMDS together with many important implementation aspects can be found in [12]. Once the CMDS configuration is obtained we are left with the task of interpreting and labeling the dimensions we have. Usually, we aim to interpret each dimension of the space. However, the number of dimensions does not necessarily reflect all the relevant characteristics. Also, although a particular feature exists in the stimulus set, it may not contribute strongly enough to become visible as a separate dimension. Therefore, one useful role of MDS is to indicate which particular features are important.

Another important MDS type is weighted multidimensional scaling (WMDS). It generalizes CMDS Euclidean distance model, so that several similarity matrices can be used. This model assumes that individuals vary in the importance they attach to each dimension of the stimulus space. In that way WMDS accounts for individual differences in human responses. WMDS analyzes several similarity matrices, one for each of m subjects. In the WMDS model, δ_{ijk} indicates the similarity between stimuli i and j, as judged by subject k. The notion of "individual taste" is incorporated into the model through weights w_{kl}, for each subject $k = 1, \cdots, m$ and each dimension $l = 1, \cdots, n$. Just as in CMDS, WMDS determines the configuration of points in the group stimulus space $\boldsymbol{X}$. However, in order to find the best possible configuration, WMDS does not use distances among the points in the group space. Instead, a configuration for each subject is made by altering the group configuration space according to the weights w_{kl}. Algebraically, given a point $\boldsymbol{x}_i$ from the group space, the points for subject k are obtained as

$$x_{ilk} = \sqrt{w_{lk}} \cdot x_{il}. \qquad (3)$$

In WMDS, the formula for stress is based on the squared distances calculated from each of m individual similarity matrices

$$\text{stress}(\Delta,\, X_k,\, f) = \sqrt{\frac{1}{m} \sum_k \frac{\sum_i \sum_j [f(\delta_{ijk}) - d_{ijk}]^2}{\sum_i \sum_j f(\delta_{ijk})^2}} \qquad (4)$$

where d_{ijk} are weighted Euclidean distances between stimuli i and j, for the subject k. In that way, the WMDS model accommodates very large differences among the individual ratings, and even very different data from two subjects can fit into the same space.

An important characteristic of CMDS is that once a configuration of points is obtained, it can be rotated, implying that the dimensions are not meaningful. Thus, when interpreting the results, higher-dimensional CMDS soon becomes impractical. As opposed to CMDS, due to the algebra of the weighted Euclidian model, once the WMDS configuration is obtained, it cannot be rotated [12], [28]. However, the stability of configuration depends heavily on the accuracy of the model; if the model fits that data well, the dimensions are meaningful which makes our job of interpreting them much easier.

C. Hierarchical Cluster Analysis

Given a similarity matrix, hierarchical cluster analysis (HCA) organizes a set of stimuli into similar units [13]. Therefore, HCA help us discover the rules and the hierarchy we use in judging similarity and pattern matching. This method starts from the stimulus set to build a tree. Before the procedure begins, all stimuli are considered as separate clusters, hence there are as many clusters as there are ranked stimuli. The tree is formed by successively joining the most similar pairs of stimuli into new clusters. At every step, either an individual stimulus is added to the existing clusters, or two existing clusters are merged. The grouping continues until all stimuli are members of a single cluster. How the similarity matrix is updated at each stage of the tree is determined by the joining algorithm. There are many possible criteria for deciding how to merge clusters. Some of the simplest methods use *nearest neighbor technique*, where the first two objects combined are those that have the smallest distance between them. Another commonly used technique is the farthest neighbor technique where the distance between two clusters is obtained as the distance between their farthest points. The centroid method calculates the distances between two clusters as the distance between their means. Also, since the merging of clusters at each step depends on the distance measure, different distance measures can result in different clustering solutions for the same clustering method [13].

Clustering techniques are often used in combination with MDS, to clarify the obtained dimensions. However, in the same way as with the labeling of the dimensions in the MDS algorithm, interpretation of the clusters is usually done subjectively and strongly depends on the quality of the data.

D. Vocabulary: Most Important Dimensions of Color Patterns

The first step in the data analysis was to arrange subjects' ratings into a similarity matrix Δ to be an input to the two-dimensional (2–D) and three-dimensional (3–D) CMDS. Also, WMDS procedure was applied to the set of 28 individual similarity matrices. WMDS was performed in two, three, four, five, and six dimensions. The stress index (4) for the 2-D solution was 0.31, indicating that a higher-dimensional solution is necessary, that is, the error is still substantial. The stress values for the three-, four-, five-, and six-dimensional configurations were: 0.26, 0.20, 0.18, and 0.16, respectively. We stopped at six dimensions since further increase did not result in a noticeable decrease of the stress value. The 2–D CMDS configuration is shown in Fig. 4. Dimensions derived from this configuration are: 1) presence/absence of a dominant color, or as we are going to call it "the dimension of overall color," and 2) color purity. It

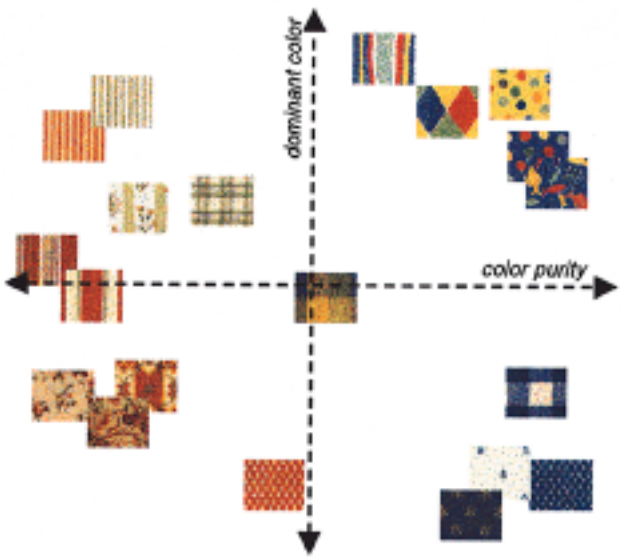

Fig. 4. Multidimensional scaling results. Two-dimensional CMDS configuration is shown. Horizontal axis represents the dimension of color pruity whereas the vertical axis is the dimension of dominant color.

is interesting that both dimensions are purely color based, indicating that, at the coarsest level of judgment, people primarily use color to judge similarity. As will be seen later, these dimensions remained in all solutions. Moreover, the 2-D configuration strongly resembles one of the perpendicular projections in the three-, four-, and five-dimensional solutions. The same holds for all three dimensions from the 3-D solution, indicating that these features could be the most general in human perception. Both for CMDS and WMDS, the same three dimensions emerged from 3-D configurations. They are

1) overall color,;
2) color purity;
3) regularity and placement.

The four-dimensional (4-D) WMDS solution revealed following dimensions:

1) overall color;
2) color purity,;
3) regularity and placement;
4) directionality.

The five-dimensional (5–D) WMDS solution came with the same four dominant characteristics with the addition of a dimension that we called "pattern heaviness." Hence, as a result of the experiment, the following five important similarity criteria emerged.

Dimension 1—Overall Color: Overall color can be described in terms of the presence/absence of a dominant color. At the negative end of this axis are patterns with an overall impression of a single dominant color (patterns 4, 5, 8, 15). This impression is created mostly because the percentage of one color is truly dominant. However, a multicolored image can also create an impression of dominant color. This happens when all the colors within this image are similar, having similar hues but different intensities or saturation (pattern 7). At the positive end of this dimension are patterns where no single color is perceived as dominant (such as in true multicolored patterns 16–20).

Dimension 2—Directionality and Orientation: This axis represents a dominant orientation in the edge distribution, or a dominant direction in the repetition of the structural element. The lowest values along this dimension have patterns with a single dominant orientation, such as stripes and then checkers (2, 4, 11–13). Midvalues are assigned to patterns with a noticeable but not dominant orientation (5, 10), followed by the patterns where a repetition of the structural element is performed along two directions (3, 8, 9, 15). Finally, completely nonoriented patterns and patterns with uniform distribution of edges or nondirectional placement of the structural element are at the positive end of this dimension.

Dimension 3—Regularity and Placement Rules: This dimension describes the regularity in the placement of the structural element, its repetition and uniformity. At the negative end of this axis are regular, uniform, and repetitive patterns (with repetition completely determined by a certain set of placement rules), whereas at the opposite end are nonrepetitive or nonuniform patterns.

Dimension 4—Color Purity: This dimension arose somehow unexpectedly, but it remained stable in all MDS configurations, clustering results, even in the subjects' explanations of their rankings. This dimension divides patterns according to the degree of their colorfulness. At the negative end are pale patterns (1, 10), patterns with unsaturated overtones (7), patterns with dominant "sandy" or "earthy" colors (5, 6, 11). At the positive end are patterns with very saturated and very pure colors (9, 13, 19, etc.). Hence, this dimension can also be named the dimension of overall chroma or overall saturation within an image.

Dimension 5—Pattern Complexity and Heaviness: This dimension showed only in the last, 5–D configuration, hence it can be seen as optional. Also, as we will show in the next section, it is not used in judging similarity until the very last level of comparison. For that reason we have also named it "a dimension of general impression." At one end of this dimension are patterns that are perceived as "light" and "soft" (1, 7, 10) while at the other end are patterns described by subjects as "heavy," "busy," and "sharp" (2, 3, 5, 17, 18, 19).

E. Grammar: Rules for Judging Similarity

Having determined the dimensions of color patterns, we need to establish a set of rules governing their use. HCA achieves that by ordering groups of patterns according to the degree of similarity, as perceived by subjects. Fig. 5 shows the ordering of clusters obtained as a result of the HCA, arising from the complete similarity matrix for 20 patterns used in the study. As a result of the HCA, we derived a list of similarity rules and the sequence of their application based on the analysis given below. For example, we observed that the very first clusters were composed of pairs of equal patterns (clusters 21–23). These were followed by the clusters of patterns with similar color and dominant orientation. Thus, from the early stages of clustering we were able to determine the initial rules used by humans in judging similarity (Rules 1 and 2). These were followed by rules emerging from the middle stages (Rules 3 and 4). Finally, at the

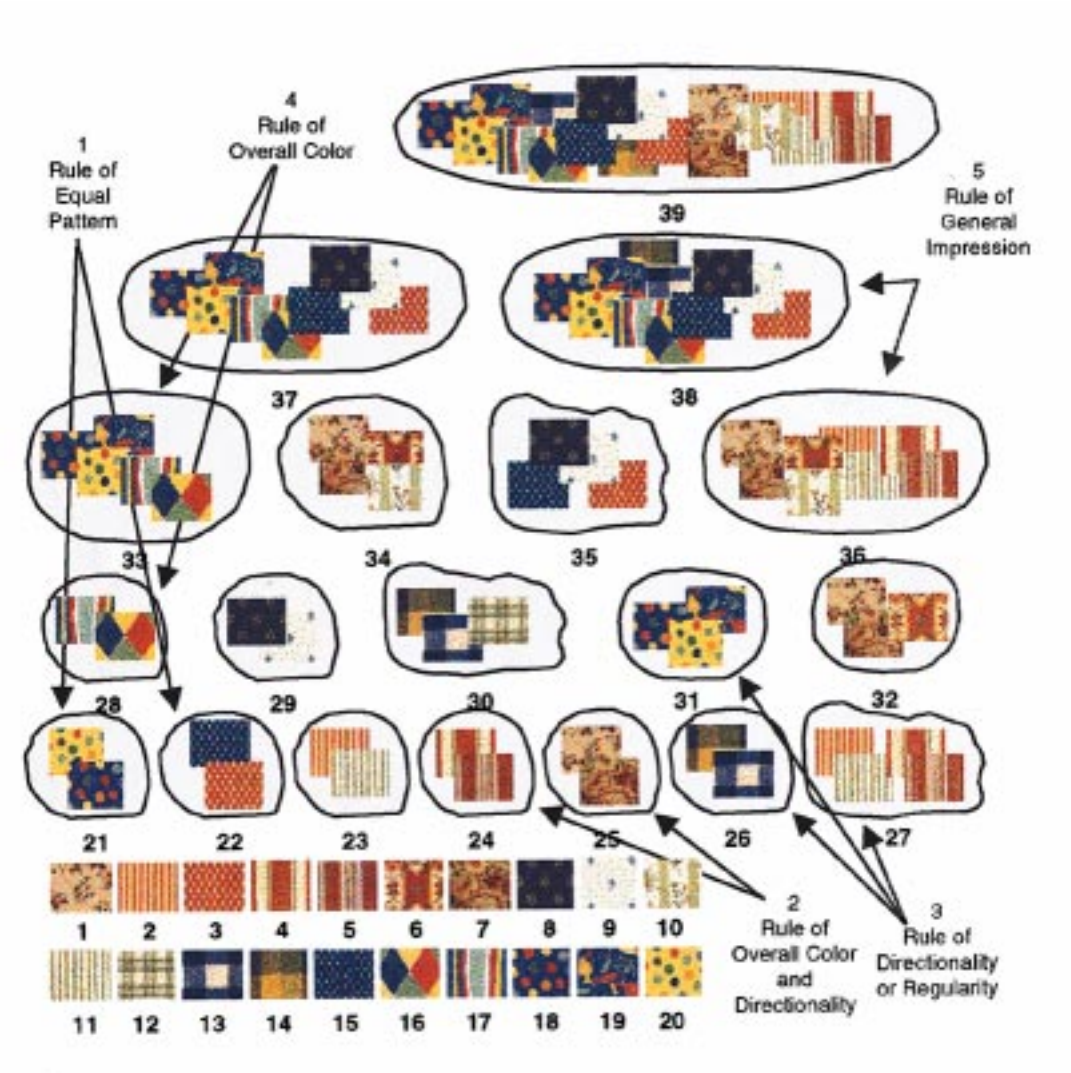

Fig. 5. Result of the HCA applied to the complete set of stimuli. Clusters 1 to 20 are original patterns, clusters 21 to 37 represent successive nodes of the tree. In the last step, clusters 36 and 38 are joined to form the top cluster. The ordering of clusters was used to determine the rules and the sequence of their application in pattern matching.

coarsest level of comparison we use Rule 5 (clusters 36–38 in Fig. 5).

In addition, to confirm the stability of rules, we have split the original data in several ways and performed separate HCA's for each part. As suggested in [12], we eliminated some of the stimuli from the data matrix and determined the HCA trees for the remaining stimuli. The rules remained stable through various solutions; thus we conclude that the 5–D configuration can be used for modeling the similarity metrics of the human visual system, together with the following rules:

Rule 1: The strongest similarity rule is that of equal pattern. Regardless of color, two textures with exactly the same pattern such as pairs (17, 18), (2, 11), and (3, 15) are always judged to be the most similar. Hence, this rule uses Dimensions 3 and 2 (pattern regularity and directionality).

Rule 2: The second rule in the hierarchy is that of overall appearance. It uses the combination of Dimension 1 (dominant color) *and* Dimension 2 (directionality). Two patterns that have similar values in both dimensions, such as pairs (10, 11), (1, 7), and the triplet (2, 4, 5) are also perceived as similar.

Rule 3: The third rule is that of similar pattern. It concerns either dimension 2 (directionality) *or* dimension 3 (pattern regularity and placement rules). Hence, two patterns which are dominant along the same direction (or directions) are seen as similar, regardless of their color. One such example is the cluster (12–14). In the same manner, seen as similar are patterns with the same placement or repetition of the structural element, even if the structural element is not exactly the same (see patterns 8 and 9, or 17, 18 and 19).

Rule 4: In the middle of the hierarchy comes the rule of dominant color. Two multicolored patterns are perceived as similar if they possess the same color distributions regardless of their content, directionality, placement, or repetition of a structural element (patterns 16–20). This also holds for patterns that have the same dominant or overall color (patterns 2–6). Hence, this rule involves only the Dimension 1 (dominant color).

Rule 5: Finally, at the very end of the hierarchy, comes the rule of general impression (Dimensions 4 and 5). This rule divides patterns into "dim," "smooth," "earthy," "romantic," or "pale" (at one end of the corresponding dimension) as opposed

to "bold," "bright," "strong," "pure," "sharp," "abstract," or "heavy" patterns (at the opposite end). This rule represents the complex combination of color, contrast, saturation, and spatial frequency, and therefore applies to patterns at the highest, abstract level of understanding.

This set of rules represents the basic grammar of pattern matching. For actual implementation of the grammar it is important to observe the way these rules are applied: Each rule can be expressed as a logical combination (logical OR, AND, XOR, NOT) of the pattern values along the dimensions involved in it. For example, consider cluster 24 composed of patterns 4 and 5 in Fig. 5. These patterns have similar overall color and dominant orientation, thus their values both along the dimensions 1 and 2 are very close. Consequently, they are perceived as similar according to the Rule 2, which is expressed in the following way:

$$(DIM_1(\text{pattern } 4) \textbf{ similar to } DIM_1(\text{pattern } 5)) \\ \textbf{AND}(DIM_2(\text{pattern } 4) \textbf{ similar to } DIM_2(\text{pattern} 5)). \quad (5)$$

III. Overview of the System

We will summarize our findings thus far. To model the human perception of similarity:

1) we determined the basic vocabulary V of color patterns consisting of dimensions 1–5: $V = \{DIM_1, \cdots, DIM_5\}$;
2) we determined the grammar G, that is, the rules governing the use of the dimensions from the vocabulary V. Five rules (R_1–R_5) were discovered so that $G = \{R_1, R_2, R_3, R_4, R_5\}$.

Having found the vocabulary and grammar, we need to design a system that will, given an input image A and a query Q:

1) measure the dimensions $DIM_i(A)$ from the vocabulary, $i = 1, \cdots, 5$;
2) for each image B from the database, apply rules R_1–R_5 from G and obtain corresponding distance measures $dist_1(A, B), \cdots, dist_5(A, B)$, where $dist_i(A, B)$ is the distance between the images A and B according to the rule i;

Therefore, the system has two main parts: 1) the feature extraction part, measuring the dimensions from V and 2) the search part, where similar patterns are found according to the rules from G. The feature extraction part is designed to extract dimensions 1 to 4 of pattern similarity. Dimension 5 (pattern complexity and heaviness) is not implemented, since our experiments have shown that people use this criterion only at a higher level of judgment, while comparing groups of textures [14]. Feature extraction is followed by judgment of similarity according to Rules 1–4 from G. Rule 5 is not supported in the current implementation, since it is only used in combination with dimension 5 at a higher level of pattern matching (such as subdividing a group of patterns into romantic, abstract, geometric, bold, etc.).

Let us now examine the system in more detail. It is important to note that the feature extraction part is developed according to the following assumptions derived from psychophysical properties of the human visual system and conclusions extracted from our experiment.

1) The overall perception of color patterns is formed through the interaction of luminance component L, chrominance component C and achromatic pattern component AP. The luminance and chrominance components approximate signal representation in the early visual cortical areas while the achromatic pattern component approximates signal representation formed at higher processing levels [15]. Our experimental results confirm this fact: we found that at the coarsest level of judgment only color features are used (2-D MDS) whereas texture information is added later and used in the detailed comparison. Therefore, our feature extraction simulates the same mechanism; it decomposes the image map into luminance and chrominance components in the initial stages, and models pattern information later in the system.
2) As in the human visual system the first approximation is that each of these components is processed through separate pathways [16], [29]. While luminance and chrominance components are used for the extraction of color-based information, the achromatic pattern component is used for the extraction of purely texture-based information. However, if we want to be more precise, we need to account for residual interactions along the pathways [17]. As will be shown in Section V, we accomplish this by extracting the achromatic pattern component from the color distribution, instead of using the luminance signal as in previous models. Moreover, the discrete color distribution is estimated through the use of a specially designed perceptual codebook allowing the interaction between the luminance and chrominance components (see Section IV).
3) Features are extracted by combining three major domains: a) nonoriented luminance domain represented by the luminance component of an image, b) oriented luminance domain represented by the achromatic pattern map, and c) nonoriented color domain represented by the chrominance component. The first two domains are essentially color blind, whereas the third domain carries only the chromatic information. These three domains are well documented in the literature [18] and experimentally verified in perceptual computational models for segregation of color textures [19]. Purely color-based dimensions (1 and 4) are extracted in the nonoriented domains and are measured using the color feature vector. Texture-based dimensions (2 and 3) are extracted in the oriented luminance domain, through the scale-orientation processing of the achromatic pattern map.

In summary, our computational model is implemented as in Fig. 1 and contains the following parts.

1) *Feature extraction block* with the following components.
   - *Image Decomposition*: Input image is transformed into the Lab color space and decomposed into luminance L and chrominance $C = (a, b)$ components.
   - *Estimation of Color Distribution*: Both L and C maps are used for the color distribution estimation and

extraction of color features. We are thus performing feature extraction along the color-based dimensions 1 and 4.
- *Pattern Map Generation*: Color features extracted in the second stage are used to build the achromatic pattern map.
- *Texture Primitive Extraction and Estimation*: The achromatic pattern map is used to estimate the spatial distribution of texture primitives. We are thus performing feature extraction along the texture-based dimensions 2 and 3.

2) *Similarity Measurement*: Here similar patterns are found according to the rules from G. Given an input image A, for every image B in the database, rules R_1–R_4 are applied and corresponding distance measures are computed. Then, depending on a query Q, a set of best matches is found.

IV. Feature Extraction Based on Color Information

The color information is used both for the extraction of color-related dimensions (color features), and for the construction of the achromatic pattern map (used later in texture processing), therefore we aim for compact, perceptually-based color representation. As illustrated in Fig. 2(a), this representation is obtained through the following steps.

1) The input image is transformed into the Lab color space.
2) Its color distribution is determined using a vector quantization-based histogram technique,.
3) Significant color features are determined from the histogram.
4) These color features are used in conjunction with a new distance measure to determine the perceptual similarity between two color distributions.

A. Color Representation

Our goal is to produce a system that performs in accordance with human perception, hence we need a representation (color space) based on human color matching. CIE Lab is such a color space, since it was designed so that intercolor distances computed using the $||\,||_2$ norm correspond to subjective color matching data [20]. After transforming an input image into the Lab color space, the next step is to estimate the color distribution by computing a histogram of the input color data. Since linear color spaces (such as RGB) can be approximated by 3-D cubes, histogram bin centers can be computed by performing separable, equidistant discretizations along each of the coordinate axes. Unfortunately, by going to the nonlinear Lab color space, the volume of all possible colors distorts from cube to an irregular cone and consequently, there is no simple discretization that can be applied to this volume.

To estimate color distributions in the Lab space, we have to determine the set of bin centers and decision boundaries that minimize some error criterion. In the Lab color system, $||\,||_2$ norm corresponds to perceptual similarity, thus representing the optimal distance metric for that space [20]. Therefore, to obtain an optimal set of bin centers and decision boundaries, we have to find Lab coordinates of N bin centers so that the overall mean-square classification error is minimized. This is exactly the underlying problem in vector quantization (VQ). Hence, we used the LBG vector quantization algorithm [21] to obtain a set of codebooks which optimally represent the valid colors in the Lab space. In any VQ design, the training data have a large effect on the final result. A commonly used approach is to select training images that are either representative of a given problem so the codebook is optimally designed for that particular application, or span enough of the input space so the resulting codebook can be used in different applications. The following problem occurs with both approaches: In order to obtain an accurate estimation for the color distribution, a large number of training images is required, resulting in a computationally expensive and possibly intractable design task. To overcome this problem, we have taken a different approach. Since we are dealing with an arbitrary input, we can assume that every color is equiprobable. Hence, a synthetic set of training data was generated by uniformly quantizing the XYZ space. The data was transformed into the Lab space and used as input to the standard VQ design algorithm. This resulted in a set of codebooks ranging in size from 16 to 512 colors. When used in the standard image retrieval task, these codebooks performed quite well. For our task, however, these codebooks have one drawback; They are designed as a global representation of the entire color space and consequently, there is no structure to the bin centers. Our purpose is to design a system which allows a user to interact with the retrieval process. Therefore, the color representation must provide manipulation with colors in a "human-friendly manner." To simulate human performance in color perception, a certain amount of structure on the relationships between the L, a, and b components must be introduced. One possible way to accomplish this is by separating the luminance L, from the chrominance (a, b) components. Starting from this assumption, we first applied one-dimensional (1–D) quantization on luminance values of the training data (using a Lloyd–Max quantizer). Then, after partitioning the training data into slices of similar luminance, a separate chrominance codebook was designed for each slice by applying the LBG algorithm to the appropriate (a, b) components.

This color representation better mimics human perception and allows the formulation of functional queries such as looking for "same but lighter color," "paler," "contrasting," etc. For example, the formulation of a query vector to search for a "lighter" color can be accomplished through the following steps:

1) extract the luminance L_Q and the (a_Q, b_Q) pair for the query color;
2) find the codebook for a higher luminance level $L > L_Q$;
3) in this codebook, find the cell which corresponds to (a, b) entry which is the closest to (a_Q, b_Q) in the $||\,||_2$ sense;
4) retrieve all images having (L, a, b) as a dominant color.

Moreover, starting from the relationship between L, a, and b values for a particular color, and its hue H and saturation S

$$H = \arctan\frac{b}{a}, \qquad S = \sqrt{a^2 + b^2}. \tag{6}$$

Similar procedures can be applied to satisfy queries such as "paler color," "bolder color," "contrasting color," etc. Finally,

in applications where the search is performed between different databases or when the query image is supplied by the user, separation of luminance and chrominance allows for elimination of the unequal luminance condition. Since the chrominance components contain the information about the type of color regardless of the intensity value, color features can be extracted only in the chrominance domain $C(i, j) = \{a(i, j), b(i, j)\}$, for the corresponding luminance level, thus allowing for comparison between images of different quality.

B. Color Feature Extraction

Color histogram representations based on color codebooks have been widely used as a feature vector in image segmentation and retrieval [22], [23]. Although good results have been reported, a feature set based solely on the image histogram may not provide a reliable representation for pattern matching and retrieval. This is due to the fact that most patterns are perceived as combinations of a few dominant colors. For example, subjects who participated in our previously reported subjective experiments [14], were not able to perceive nor distinguish more than six or seven colors, even when presented with very busy or multicolored patterns. For that reason, we are proposing color features and associated distance measures consisting of the subset of colors (which best represent an image), augmented by the area percentage in which each of these colors occur.

In our system we have used a codebook with $N = 71$ colors denoted by $C_{71} = \{C_1, C_2, \cdots, C_{71}\}$ where each color $C_i = \{L_i, a_i, b_i\}$ is a three-dimensional Lab vector. As the first step in the feature extraction procedure (before histogram calculation) input image is convolved with a B-spline smoothing kernel. This is done to refine contours of texture primitives and foreground regions, while eliminating most of the background noise. The B-spline kernel is used since it provides an optimal representation of a signal in the $||\ ||_2$ sense, hence minimizing the perceptual error [24]. The second step (after the histogram of an image was built) involves extraction of dominant colors to find colors from the codebook that adequately describe a given texture pattern. This was done by sequentially increasing the number of colors until all colors covering more than 3% of the image area have been extracted. The remaining pixels were represented with their closest matches (in $||\ ||_2$ sense) from the extracted dominant colors. Finally, the percentage of each dominant color was calculated and the color feature vectors were obtained as

$$f_c = \{(i_j, p_j) | j \in [1, N], p_j \in [0, 1]\} \tag{7}$$

where i_j is the index in the codebook, p_j is the corresponding percentage and N is the number of dominant colors in the image. Another similar representation has been successfully used in image retrieval [25].

The proposed feature extraction scheme has several advantages: It provides an optimal representation of the original color content by minimizing the MSE introduced when using a small number of colors. Then, by exploiting the fact that the human eye cannot perceive a large number of colors at the same time, nor is it able to distinguish close colors well, we provide a very compact feature representation. This greatly reduces the size of the features needed for storage and indexing. Furthermore, because of the codebook used, this representation facilitates queries containing an overall impression of patterns expressed in a natural way, such as "find me all blue-yellow fabrics," "find me the same color, but a bit lighter," etc. Finally, in addition to storing the values of the dominant colors and their percentages, we are also storing the actual number of dominant colors. This information is useful in addressing the more complex dimensions of pattern similarities as suggested in [14]. Namely, by using this feature we can search for simple and single colored patterns, versus heavy, multicolored ones.

C. Color Metric

The color features described above, represented as color and area pairs, allow the definition of a color metric that closely matches human perception. The idea is that the similarity between two images in terms of color composition should be measured by a combination of color and area differences.

Given two images, a query image A and a target image B, with N_A and N_B dominant colors, and feature vectors $f_c(A) = \{(i_a, p_a) | \forall a \in [1, N_A]\}$, and $f_c(B) = \{(i_b, p_b) | \forall b \in [1, N_B]\}$, respectively, we first define the similarity between these two images in terms of a single dominant color. Suppose that i is the dominant color in image A. Then, we measure the similarity between A and B in terms of that color using the minimum of distance measures between the color element (i, p) and the set of color elements $\{(i_b, p_b) | \forall\ b \in [1, N_B]\}$:

$$d(i, B) = \min_{b \in [1, N_B]} D((i, p), (i_b, p_b)) \tag{8}$$

where

$$D((i, p), (i_b, p_b)) = |p - p_b| + \sqrt{(L - L_b)^2 + (a - a_b)^2 + (b - b_b)^2}. \tag{9}$$

Once the distance $d(i, B)$ has been calculated, besides its value we also use its argument to store the color value from B that, for a particular color i from A, minimizes (8). We denote this color value by $k(i, B)$ as

$$k(i, B) = \arg\ d(i, B). \tag{10}$$

Note that the distance between two color/area pairs is defined as the sum of the distance in terms of the area percentage and the distance in the Lab color space, both within the range [0, 1]. In [25], Ma *et al.* used a different definition where the overall distance is the product of these two components. That definition, while being more intuitive, has the drawback that when either component distance is very small the remaining component becomes irrelevant. Consider the extreme case, when the color distance between two color/area pairs is zero. This is not unusual, since the color space has been heavily quantized. Then, even if the difference between the two area percentages is very large, the overall distance is zero yielding a measure that does not match human perception. Our definition is a simple and effective remedy to that problem—it guarantees that both color

and area components contribute to the perception of color similarity.

Given the distance between two images in terms of one dominant color as defined above, the distance in terms of overall color composition is defined as the sum over all dominant colors from both images, in the following way.

1) For image A, for $\forall\ a \in [1,\ N_A]$ find $k_A(i_a,\ B)$ and the corresponding distance $d(i_a,\ B)$.
2) Repeat this procedure for all dominant colors in B, that is, for $\forall\ b \in [1,\ N_B]$ find $k_B(i_b,\ B)$ and $d(i_b,\ A)$.
3) calculate the overall distance as

$$\text{dist}(A,\ B) = \sum_{a \in [1,\ N_A]} d(i_a,\ B) + \sum_{b \in [1,\ N_B]} d(i_b,\ A). \quad (11)$$

V. Feature Extraction Based on Texture Information

Having obtained the color feature vector, the extraction of texture features involves the following steps [see Fig. 2(b)]:

1) spatial smoothing, to refine texture primitives and remove background noise;
2) building the achromatic pattern map;
3) building the edge map from the achromatic pattern map;
4) application of a nonlinear mechanism to suppress nontextured edges;
5) orientation processing to extract the distribution of pattern contours along different spatial directions;
6) computation of a scale-spatial texture edge distribution.

Spatial smoothing of the input image is performed during the extraction of color features. Then, the color feature representation is used for construction of the achromatic pattern map. The achromatic map is obtained in the following manner: For a given texture, by using the number of its dominant colors N, a gray level range of 0–255 is discretized into N levels. Then, dominant colors are mapped into gray levels according to the following rule: Level 0 is assigned to the dominant color with the highest percentage of pixels, the next level is assigned to the second dominant color, etc., until the level 255 has been assigned to a dominant color with the lowest area percentage. In other words, the achromatic pattern map models the fact that human perception and understanding of form, shape, and orientation is completely unrelated to color. Furthermore, it resolves the problem of secondary interactions between the luminance and chrominance pathways. As an example, consider a pair of textures in Fig. 6(a). The values in the luminance map are much higher for the texture on top, hence the edge amplitudes, and edge distributions are different for these two images [see Fig. 6(b)]. Moreover, the dominant colors are not close, which makes the classification of these two patterns as similar (either using luminance, chrominance, or color features) extremely difficult. However, in our model, the way that luminance and chrominance are coupled into a single pattern map guarantees that both textures will have identical achromatic pattern maps [see Fig. 6(c)], leading to almost identical texture feature vectors.

The objective of edge and orientation processing is to extract information about the pattern contours from the achromatic pattern map. Instead of applying a bank of oriented filters, as in previous models, we decided to compute polar edge maps and use them to extract distribution of edges along different directions. This approach allowed us to obtain the edge distribution for an arbitrary orientation with low computational cost. It also introduced certain flexibility in the extraction of texture features since, if necessary, the orientation selectivity can be enhanced by choosing an arbitrary number of orientations. In our system, we used edge-amplitude and edge-angle maps, calculated at each image point. Edge maps were obtained by convolving an input achromatic pattern map with the horizontal and vertical derivatives of a Gaussian and converting the result into polar coordinates. The derivatives of a Gaussian along x and y axes were computed as

$$g_x(i,\ j) = ie^{-(i^2+j^2)}, \qquad g_y(i,\ j) = je^{-(i^2+j^2)} \quad (12)$$

while the derivatives of the achromatic pattern map along x and y axes were computed as

$$\begin{aligned} A_x(i,\ j) &= (g_x * AP)(i,\ j), \\ A_y(i,\ j) &= (g_y * AP)(i,\ j) \end{aligned} \quad (13)$$

where $*$ stands for 2-D convolution. These derivatives were then transformed into their polar representation as

$$\begin{aligned} A(i,\ j) &= \sqrt{A_x(i,\ j)^2 + A_y(i,\ j)^2}, \\ \theta(i,\ j) &= \tan^{-1}\frac{A_y(i,\ j)}{A_x(i,\ j)}, \quad \theta(i,\ j) \in \left(-\frac{\pi}{2},\ \frac{\pi}{2}\right]. \end{aligned} \quad (14)$$

Texture phenomenon is created through the perception of image "edgeness" along different directions, over different scales. Hence, to estimate the placement and organization of texture primitives, we do not need information about the edge strength at a certain point; rather, we only need to know whether an edge exists at this point and the direction of the edge. Therefore, after the transformation into the polar representation, the amplitude map is nonlinearly processed as

$$A_Q(i,\ j) = \begin{cases} 1, & \text{med}\,(A(i,\ j)) \geq T \\ 0, & \text{med}\,(A(i,\ j)) < T \end{cases} \quad (15)$$

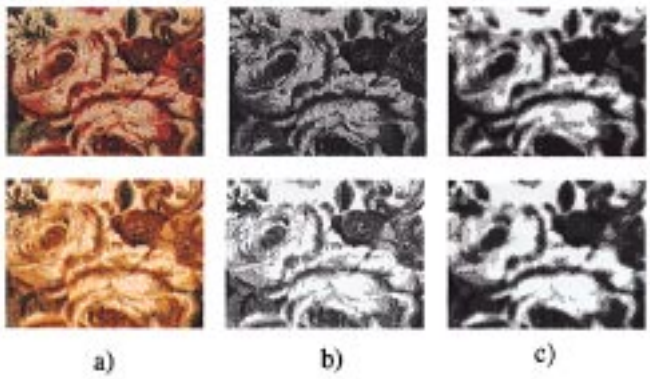

Fig. 6. Human perception and understanding of form, shape, and orientation is unrelated to color. The system models this through the use of the achromatic pattern map. (a) Two identical textures with different color distributions are perceived as identical. (b) However, modeling of these patterns by their luminance components results in different feature vectors. (c) The solution is to map dominant colors from both patterns into the same gray-scale values, resulting in an achromatic pattern map. This representation corresponds to human perception. Consequently, the feature vectors extracted from the achromatic pattern maps are almost identical.

where med$(\cdot)$ represents the median value calculated over a 5 × 5 neighborhood. Nonlinear median operation was introduced to suppress false edges in the presence of stronger ones, and eliminate weak edges introduced by noise. The quantization threshold T is determined as

$$T = \mu_A - 2\sqrt{\sigma_A^2} \tag{16}$$

where μ_A and σ_A^2 are the mean and variance of the edge amplitude, estimated on a set of 300 images. This selection allowed all the major edges to be preserved. After quantizing the amplitude map, we perform the discretization of the angle space, dividing it into the six bins corresponding to directions 0°, 30°, 60°, 90°, 120°, and 150°, respectively. For each direction an amplitude map $A_{\theta_i}(i, j)$ is built as

$$A_{\theta_i}(i, j) = \begin{cases} 1, & A_Q(i, j) = 1 \wedge \theta(i, j)) \in \theta_i \\ 0, & A_Q(i, j) = 0 \vee \theta(i, j)) \notin \theta_i \end{cases}, \quad i = 1, \cdots, 6. \tag{17}$$

To address the textural behavior at different scales, we estimate mean and variance of edge density distribution, by applying overlapping windows of different sizes to the set of directional amplitude maps. For a given scale, along a given direction, edge density is calculated simply by summing the values of the corresponding amplitude map within the window, and dividing that value by the total number of pixels in the window. We used four scales, with the following parameters for the sliding window:

$$\begin{aligned} &\text{Scale 1:} \quad WS_1 = \tfrac{3}{4}W \times \tfrac{3}{4}H, \quad N_1 = 30, \\ &\text{Scale 2:} \quad WS_2 = \tfrac{2}{5}W \times \tfrac{2}{5}H, \quad N_2 = 56, \\ &\text{Scale 3:} \quad WS_3 = \tfrac{1}{5}W \times \tfrac{1}{5}H, \quad N_3 = 80, \\ &\text{Scale 4:} \quad WS_4 = \tfrac{1}{10}W \times \tfrac{1}{10}H, \quad N_4 = 224 \end{aligned}$$

where WS_i and N_i are window size and number of windows for scale i, and W and H are the width and height of the input texture. Note that the above approach is scale (zoom) invariant. In other words, the same pattern at different scales will have similar feature vectors.

Hence, at the output of the texture processing block, we have a texture feature vector of length 48:

$$f_t = [\mu_1^{\theta_1}\sigma_1^{\theta_1}\mu_1^{\theta_2}\sigma_1^{\theta_2} \cdots \mu_1^{\theta_6}\sigma_1^{\theta_6}\mu_2^{\theta_1}\sigma_2^{\theta_1} \cdots \mu_4^{\theta_6}\sigma_4^{\theta_6}] \tag{18}$$

where $\mu_i^{\theta_j}$ and $\sigma_i^{\theta_j}$ stand for mean and standard deviation of texture edges at scale i along the direction θ_j. Each feature component is normalized so that it assumes the mean value of zero and standard deviation of one over the whole database. In that way this feature vector essentially models both texture-related dimensions (directionality and regularity): The distribution estimates along the different directions address the dimension of directionality. At any particular scale, the mean value can be understood as an estimation of the overall pattern quality, whereas the standard deviation estimates the uniformity, regularity and repetitiveness at this scale, thus addressing the dimension of pattern regularity.

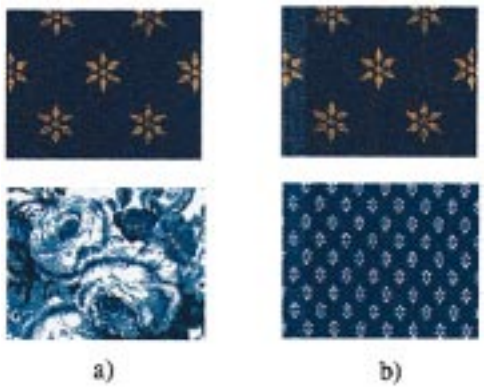

Fig. 7. Discrimination of textures based on the mean and variance of texture edge distribution. (a) If two textures have different degrees of regularity, characterized by different variances, they are immediately perceived as different. (b) However, if two textures have similar degrees of regularity, characterized by similar variances, perception of similarity depends on pattern quality, which is modeled by the mean values of edge distribution.

A. Texture Metric

As previously mentioned, at any particular scale, the mean values measure the overall edge pattern and the standard deviations measure the uniformity, regularity and repetitiveness at this scale. Our experiments [14] demonstrate that the perceptual texture similarity between two images is a combination of these two factors in the following way: If two textures have very different degrees of uniformity [as in Fig. 7(a)] they are immediately perceived as different. On the other hand, if their degrees of uniformity, regularity and repetitiveness are close [as in Fig. 7(b)], their overall patterns should be further examined to judge similarity. The smooth transition between these two factors can be implemented using the logistic function, commonly used as an excitation function in artificial neural networks [26]. Thus, the distance between the query image A and the target image B, with texture feature vectors

$$f_t(A) = [\mu_{1A}^{\theta_1} \cdots \sigma_{4A}^{\theta_6}] \quad \text{and} \quad f_t(B) = [\mu_{1B}^{\theta_1} \cdots \sigma_{4B}^{\theta_6}] \tag{19}$$

respectively, is defined as

$$\begin{aligned} M_i^{\theta_j} &= |\mu_{iA}^{\theta_j} - \mu_{iB}^{\theta_j}|, \\ D_i^{\theta_j} &= |\sigma_{iA}^{\theta_j} - \sigma_{iB}^{\theta_j}| \end{aligned} \tag{20}$$

$$\begin{aligned} d_i^{\theta_j} &= w_M(i, \theta_j)M_i^{\theta_j} + w_D(i, \theta_j)D_i^{\theta_j} \\ &= \frac{e^{-\alpha(D_i^{\theta_j} - Do)}}{1 + e^{-\alpha(D_i^{\theta_j} - Do)}} M_i^{\theta_j} \\ &\quad + \frac{1}{1 + e^{-\alpha(D_i^{\theta_j} - Do)}} D_i^{\theta_j}, \end{aligned} \tag{21}$$

$$\text{dist}(A, B) = \sum_i \sum_j d_i^{\theta_j}. \tag{22}$$

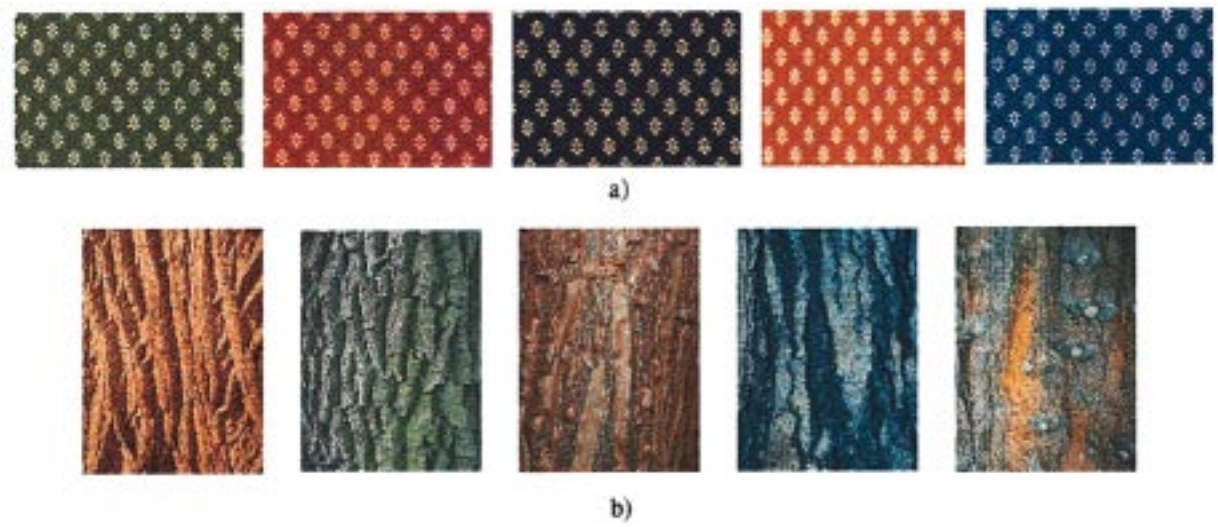

Fig. 8. Examples of the search mechanism using Rule 1 (the rule of equal pattern). This is the strongest rule people use when judging similarity. The leftmost image is the query pattern following by four best matches. (a) Example from the Interior Design database. (b) Example from the Corel database: bark textures.

At each scale i and direction θ_j, the distance function $d_i^{\theta_j}$ is the weighted sum of two terms: the first $M_i^{\theta_j}$, measuring the difference in mean edge density and the second $D_i^{\theta_j}$, measuring the difference in standard deviation, or regularity. The weighting factors, $w_M(i, \theta_j)$ and $w_D(i, \theta_j)$, are designed such that when the difference in standard deviation is small, the first term is more dominant; as it increases, the second term becomes dominant, thus matching human perception as stated above. The parameters α and Do control the behavior of the weighting factors, where α controls the sharpness of the transition, and Do defines the transition point. These two parameters are currently trained using 40 images taken from an interior design database, in the following way: First, ten images were selected as representatives of the database. Then, for each representative, three comparison images were chosen as the most similar, close, and least similar to the representative. For each representative image I_i, $i = 1, \cdots, 10$, the comparison images $C_{i,j}$, $j = 1, \cdots, 3$ are ordered in decreasing similarity. Thus, sets $\{I_i\}$ and $\{C_{i,j}\}$ represent the ground truth. For any given set of parameters (α, Do), the rankings of the comparison images as given by the distance function can be computed. Let $\text{rank}_{ij}(\alpha, Do)$ represents the ranking of the comparison image $C_{i,j}$ for representative image I_i. Ideally, we would like to achieve

$$\text{rank}_{ij}(\alpha, Do) = j, \quad \forall\, i, j | i \in [1, 10], j \in [1, 3]. \tag{23}$$

The deviation from ground truth is computed as

$$D(\alpha, Do) = \sum_{i=1}^{10} d_i(\alpha, Do) \tag{24}$$

where

$$d_i(\alpha, Do) = \sum_{j=1}^{3} \left| \text{dist}(I_i, C_{i,j}) - \text{dist}(I_i, C_{i,\text{rank}_{ij}(\alpha, Do)}) \right|. \tag{25}$$

The goal of parameter training is to minimize function $D(\alpha, Do)$. Many standard optimization algorithms can be used to achieve this. We used Powell's algorithm [27] and the optimal parameters derived were: $\alpha = 10$ and $Do = 0.95$.

VI. Similarity Measurement

In this part of the system, we perform similarity measurement based on the rules from our grammar G. The system was tested on the following databases: Corel (more than 2000 images), interior design (350 images), architectural surfaces (600 images), stones (350 images), historic ornaments (110 images), and oriental carpets (100 images).

The current implementation of our system supports four strongest rules for judging the similarity between patterns. Here we briefly summarize the rules and their implementation in the system. For more details on rules, see Section II or [14].

Applying Rule 1: The first similarity rule is that of *equal pattern.* Regardless of color, two textures with exactly the same pattern are always judged to be similar. Hence, this rule concerns the similarity only in the domain of texture features, without actual involvement of any color-based information. Therefore, this rule is implemented by comparing texture features only, using the texture metric (20)–(22). The same search mechanism supports Rule 3 (*equal directionality, regularity or placement*) as well. According to that rule, two patterns that are dominant along the same directions are seen as similar, regardless of their color. In the same manner, seen as similar are textures with the same placement or repetition of the structural element, even if the structural element is not exactly the same. Hence, the value of the distance function in the texture domain reflects either pattern identity or pattern similarity. For example, very small distances mean that two patterns are exactly the same (implying that the rule of identity was used), whereas somewhat larger distances imply that the similarity was judged by the less rigorous rules of equal directionality or regularity. Examples of the equal pattern search mechanism are given in Fig. 8, while the examples of similar pattern search mechanism are given in Fig. 10.

Applying Rule 2: The second in the hierarchy of similarities is the combination of dominant colors and texture directionality,

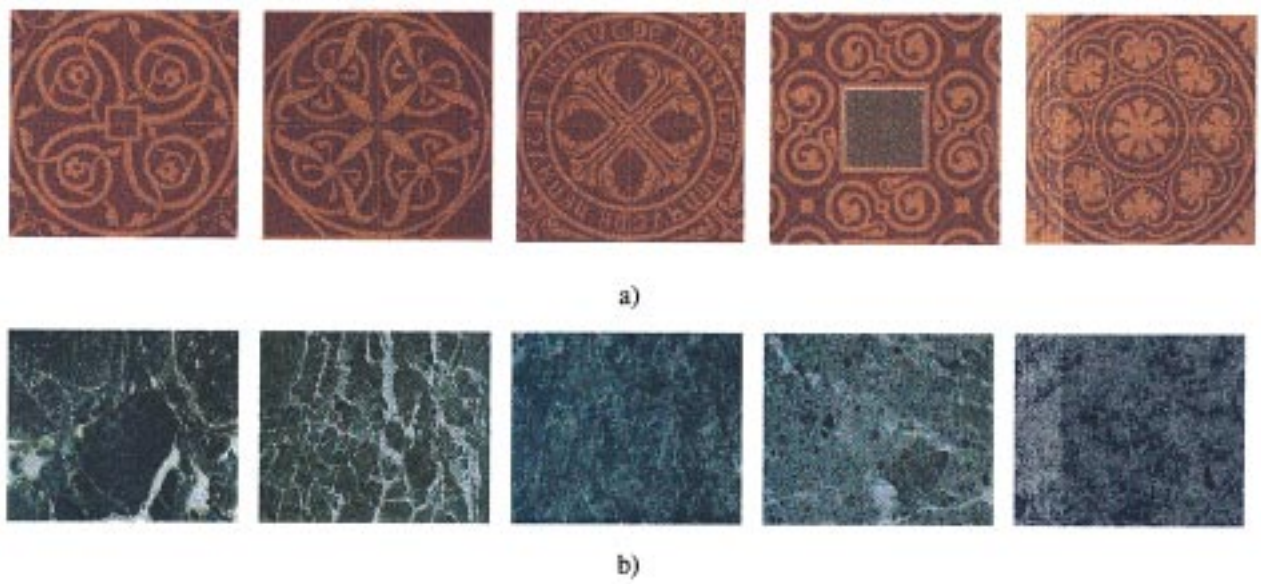

Fig. 9. Examples of the search mechanism using Rule 2 (the rule of similar overall appearance). This is the second strongest rule people use when judging similarity. This rule comes into play when there are no identical patterns. The leftmost image is the query pattern followed by four best matches. (a) Example from the Historic Ornaments database. (b) Example from the Stones database: various types of green marble.

Fig. 10. Examples of the search mechanism using Rule 3 (the rule of similar pattern). The leftmost image is the query pattern following by four best matches. (a) Example from the Oriental Carpets database. (b) Example from the Architectural Surfaces database.

yielding images with similar overall appearance. The actual implementation of this rule involves comparison of both color and texture features. Therefore the search is first performed in the texture domain, using texture features and metrics (20)–(22). A set of selected patterns is then subjected to another search, this time in the color domain, using color features (7) and color metric (8)–(11). Examples of this search mechanism are given in Fig. 9.

Applying Rule 3: The same mechanism as in Applying Rule 1 is used here, and the search examples are given in Fig. 10.

Applying Rule 4: According to the *rule of dominant color,* two patterns are perceived as similar if they possess the same color distributions regardless of texture quality, texture content, directionality, placement, or repetition of a structural element. This also holds for patterns that have the same dominant or overall color. Hence, this rule concerns only similarity in the color domain and is applied by comparing color features only. An example of the search is given in Fig. 11.

VII. Query Types and Other Search Examples

As explained in the introduction, one of the assumptions about the model is that chromatic and achromatic components are processed through mostly separate pathways. Hence, by separating color representation and color metric from texture representation and texture metric, we add a significant amount of flexibility into the system in terms of manipulation of image features. This is an extremely important issue in many practical applications, since it allows for different types of queries. As input into the system the user is expected to supply: a) a query and b) patterns to begin the search. The rules explained in the previous section model typical human queries, such

Fig. 11. Example of the search mechanism using Rule 4 (the rule of dominant color). The leftmost image is the query pattern followed by four best matches. Example is from the Historic Ornaments database: Islamic designs with lettering from an illuminated Koran, 14th or 15th century.

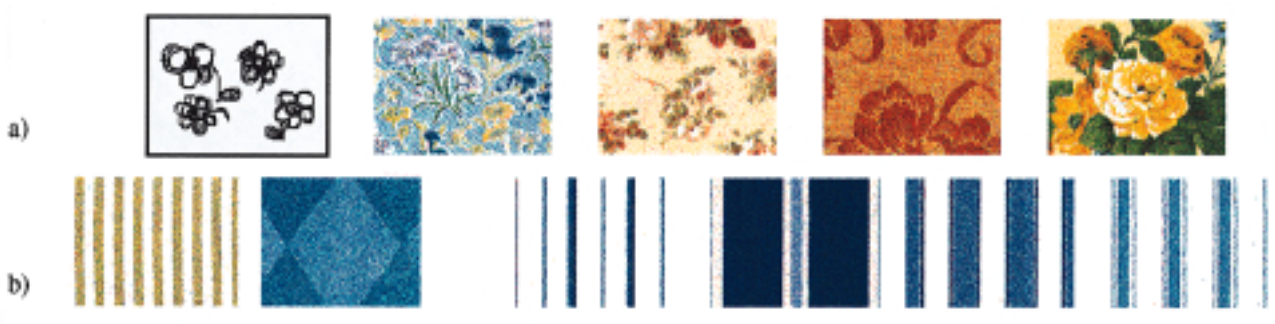

Fig. 12. Different types of queries supported by the system. (a) Query by sketch. The user supplies a sketch (bitmap image) of a desired pattern (the leftmost image). Four best matches are given from the interior Design database. (b) Combination query. The desired pattern (stripes) is taken from one input image (first from left) and the desired color (blue) from another (second from left). Four best matches are given on the right.

as: "find the same pattern" (Rule 1), "find all patterns with similar overall appearance" (Rule 2), "find similar patterns" (Rule 3), "find all patterns of similar color," "find all patterns of a given color," "find patterns that match a given pattern" (Rule 4). Moreover, due to the way the color codebook is designed, the system supports additional queries such as "find darker patterns," "find more saturated patterns," "find simple patterns," "find multicolored patterns," and "find contrasting patterns." The input pattern the user provides can be supplied by the user, selected from a database, or given in the form of a sketch. If the user has color preferences, they can be specified either from the color codebook, or from another pattern.

As an example, let us discuss query by sketch. There are certain situations when the user is unable to supply an image of the pattern he is trying to find. Hence, instead of browsing through the database manually, our system provides tools for sketching the pattern and formulating a query based on the obtained bitmap image. In that case, without any lowpass prefiltering, only texture feature vector is computed for the bitmap image and used in the search. One such query and four best matches are given in Fig. 12(a). Furthermore, this search mechanism allows the user to specify a desired color, by selecting a color $i = \{L_i, a_i, b_i\}$ from the codebook. Then, the search is performed in two iterations. First a subset of patterns is selected based on color similarity. Color similarity between the color i and target image B, with the color feature vector $f_c(B) = \{(i_b, p_b) | \forall\ b \in [1, N_B]\}$ is calculated as

$$d(i, B) = \min_{b \in [1, N_B]} D_c(i, i_b),$$

$$D_c(i, i_b) = \sqrt{(L_i - L_b)^2 + (a_i - a_b)^2 + (b_i - b_b)^2}. \quad (26)$$

Next, within the selected set, a search based on texture features is performed to select the best match. A similar search mechanism is applied for combination query, where the desired pattern is taken from one input image and the desired color from another image [see Fig. 12(b)], or in a search where the desired pattern is specified by an input image and the desired color is selected from the color map.

To conclude this section, we present retrieval results on general class of images from the Corel database. Although our system was designed specifically for color patterns, the search results demonstrate robustness of the algorithm to other types of images (such as natural scenes and images with homogeneous regions as in Fig. 13).

VIII. Discussion and Conclusions

It is our belief that a good working system for image retrieval must accomplish visual similarity along perceptual dimensions. With this as the central thrust of our research, we performed subjective experiments and analyzed them using multidimensional scaling techniques to extract the relevant dimensions. We then interpreted these dimensions along perceptual categories, and used hierarchical clustering to determine how these categories are combined in measuring similarity of color patterns. Having discovered the psychophysical basis of pattern matching, we developed algorithms for feature extraction and image retrieval in the domain of color patterns. As part of this research we realized a need for distance metrics that are better matched to human perception. Distance metrics that we developed for color matching (8)–(11) and texture matching (20)–(22) satisfy this criterion.

While most of our research has been directed at color patterns, we believe that the underlying methodology has greater significance beyond color and texture. We believe that such a methodology, if applied to other retrieval tasks (such as shape and object understanding), will result in a system that is better matched to human expectations. A major advantage of such an approach is that it eliminates the need for selecting the visual primitives for image retrieval and expecting the user to assign weights to them, as in most current systems. Furthermore, as can

Fig. 13. Examples of the serch algorithms applied to the general class of images. The leftmost image is the query pattern followed by four best matches. (a) Application of Rule 2 (the rule of overall appearance). Example from the Corel database: Tulips. (b) Application of Rule 3 (the rule of similar pattern). Example from the Corel database: Alaska. (c) Application of Rule 4 (the rule of dominant color). Example from the Corel database: Vegetables.

be seen from the results, our rules of pattern matching are robust enough to work in various domains, including digital museums [Figs. 9(a) and 11], architecture [Figs. 8(b) and 10(b)], interior design [Fig. 9(b)], and fashion and design industry [Figs. 8(a) and 12]. In general, as long as there is no meaning attached to the patterns (or even images) our approach should work well. However, when building any system dealing with image similarity, one should be aware of the importance of image content or domain specific information, and additional studies addressing this issue need to be conducted.

The important reason for the success of our system is that it implements the following experimental, biological, and physiological observations.

1) The perception of color patterns can be modeled by a set of visual attributes and rules governing their use.
2) This same perception is formed through the interaction of luminance and chrominance components (in the early stages of the human visual system), and achromatic pattern component (in the later stages of the human visual system).
3) Each of these components is processed through separate pathways.
4) Perception and understanding of patterns is unrelated to color and relative luminance.
5) Patterns are perceived through the interaction of image edges of different orientations and at different scales.

Each of these assumptions has its equivalent in the system, and is accomplished by

1) determining the basic vocabulary and grammar of color patterns through a subjective experiment;
2) decomposing an image into luminance, chrominance, and pattern maps;
3) processing the color information first, and then texture;
4) modeling an image pattern with its achromatic pattern map;
5) extracting texture features from edge representation of the achromatic pattern map at different scales, along different directions.

This has been the approach we have taken toward building an image retrieval system that has human like performance and behavior. Besides image retrieval, the proposed model can be utilized in other areas such as perceptually based segmentation and coding, pattern recognition and machine vision as well as for effectively employing perceptual characteristics in scientific visualization of large data sets.

ACKNOWLEDGMENT

The authors wish to thank A. Stanley-Marbell for his work on color search, D. Kall for helping design the experiment, J. Hall for his tremendous help with the multidimensional scaling and for many useful suggestions, D. Davis for providing software for the subjective experiment, and J. Pinheiro for his help with the statistical analysis of the data. The authors also thank F. Juang for technical discussions and J. Mazo for insightful comments.

REFERENCES

[1] K. Hirata and T. Katzo, "Query by visual example, content based image retrieval," in *Advances in Database Technology-EDBT'92*, vol. 580, A. Pirotte, C. Delobel, and G. Gottlob, Eds., 1992.
[2] W. Niblack *et al.*, "The QBIC project: Quering images by content using color, texture and shape," in *Proc. SPIE Storage and Retrieval for Image and Video Data Bases*, 1994, pp. 172–187.
[3] H. Tamura, S. Mori, and T. Yamawaki, "Textural features corresponding to visual perception," *IEEE Trans. Syst., Man, Cybern.,*, vol. 8, pp. 460–473, 1982.
[4] A. Pentland, R. W. Picard, and S. Sclaroff, "Photobook: Content-based manipulation of image databases," *Int. J. Comput. Vis.*, vol. 18, pp. 233–254, 1996.

[5] J. R. Smith and S. Chang, "Visualseek: A fully automated content-based query system," in *Proc. ACM Multimedia'96*, pp. 87–98.
[6] W. Y. Ma and B. S. Manjunath, "Netra: A toolbox for navigating large image databases," in *Proc. IEEE Int. Conf. Image Processing*, 1997, pp. 568–571.
[7] A. Gupta and R. Jain, "Visual information retrieval," *Commun. ACM*, vol. 40, pp. 70–79, 1997.
[8] J. Dowe, "Content based retrieval in multimedia imaging," in *Proc. SPIE Conf. Storage and Retrieval for Image and Video Databases*, 1993.
[9] Y. Rui, T. S. Huang, and S. Mehrotra, "Content-based image retrieval with relevance feed-back in Mars," in *Proc. IEEE Conf. Image Processing*, 1997, pp. 815–818.
[10] Amadsun and R. King, "Textural features corresponding to texture properties," *IEEE Trans. Syst., Man, Cybern.,*, vol. 19, pp. 1264–1274, 1989.
[11] A. R. Rao and G. L. Lohse, "Toward a texture naming system: Identifying relevant dimensions of texture," *Vis. Res.*, vol. 36, no. 11, pp. 1649–1669, 1996.
[12] J. Kruskal and M. Wish, *Multidimensional Scaling*. London, U.K.: Sage , 1978.
[13] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
[14] A. Mojsilovic *et al.*, "Toward vocabulary and grammar of color patterns," IEEE Trans. Image Processing, to be published.
[15] T. N. Cornsweet, *Visual Perception*. New York: Academic , 1970.
[16] E. A. DeYoe and D. C. VanEssen, "Concurrent processing streams in monkey visual cortex," *Trends. Neurosci.*, vol. 11, pp. 219–226, 1988.
[17] R. L. DeValois and K. K. DeValois, *Spatial Vision*. Oxford, U.K.: Oxford Univ.Press, 1990.
[18] M. S. Livingstone and D. H. Hubel, "Segregation of form, color, movement and depth: Anatomy, physiology and perception," *Science*, vol. 240, pp. 740–749, 1988.
[19] T. V. Papathomas, R. S. Kashi, and A. Gorea, "A human vision based computational model for chromatic texture segregation," *IEEE Trans. Syst., Man, Cybern. B*, vol. 27, pp. 428–440, June 1997.
[20] G. Wyszecki and W. S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*, New York: Wiley, 1982.
[21] A. Gersho and R. M. Gray, *Vector Quantization and Signal Processing*. Boston, MA: Kluwer, 1992.
[22] M. Ioka, "A method of defining the similarity of images on the basis of color information," IBM Res., Tokyo Res. Lab., Tech. Rep. RT-0030, Nov. 1989.
[23] M. Swain and D. Ballard, "Color indexing," *Int. J. Comput.Vis.*, vol. 7, pp. 11–32, 1991.
[24] M. Unser, A. Aldroubi, and M. Eden, "Enlargement or reduction of digital images with minimum loss of information," *IEEE Trans. Image Processing*, vol. 4, pp. 247–257, Mar. 1995.
[25] W. Y. Ma, Y. Deng, and B. S. Manjunath, "Tools for texture/color base search of images," *Proc. SPIE*, vol. 3016, pp. 496–505, 1997.
[26] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York: Macmillan , 1994.
[27] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, 2nd ed, New York: Cambridge Univ. Press, 1992, pp. 412–420.
[28] SPSS Inc., , , SPSS Professional Statistics, Chicago, IL, 1997.
[29] A. B. Poirson and B. A. Wandell, "Appearance of colored patterns: Pattern-color separability," *J. Opt. Soc. Amer. A*, vol. 10, Dec. 1993.
[30] S. Santini and R. Jain, "Similarity Matching," *IEEE Trans. Pattern Anal. Machine Intell.*, to be published.

Aleksandra Mojsilović (S'93–M'98) was born in Belgrade, Yugoslavia, in 1968. She received the B.S.E.E., M.S.E.E., and Ph.D. degrees from the University of Belgrade, Belgrade, Yugoslavia, in 1992, 1994, and 1997, respectively.

From 1994 to 1998, she was a member of the academic staff at Department of Electrical Engineering, University of Belgrade. Since 1998, she has been a Member of Technical Staff, Bell Laboratories, Lucent Technologies, Murray Hill, NJ. Her main research interests include multidimensional signal processing, computer vision, and image analysis.

Jelena Kovačević (S'88–M'91–SM'96) received the Dipl. Electr. Eng. degree from the Electrical Engineering Department, University of Belgrade, Yugoslavia, in 1986, and the M.S. and Ph.D. degrees from Columbia University, New York, NY, in 1988 and 1991, respectively.

In November, 1991, she joined AT&T Bell Laboratories (now Lucent Technologies), Murray Hill, NJ, as a Member of Technical Staff. In the fall of 1986, she was a Teaching Assistant at the University of Belgrade. From 1987 to 1991, she was a Graduate Research Assistant at Columbia University. In the summer of 1985, she worked for Gaz de France, Paris, France, during the summer of 1987 for INTELSAT, Washington, DC, and in the summer of 1988 for Pacific Bell, San Ramon, CA. Her research interests include wavelets, multirate signal processing, data compression, and signal processing for communications. She is the coauthor of the book (with M. Vetterli) *Wavelets and Subband Coding*, (Englewood Cliffs, NJ: Prentice-Hall, 1995).

Dr. Kovacevic served as an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING and as the Guest Co-editor (with I. Daubechies) of the special issue on wavelets of the PROCEEDINGS OF THE IEEE. She is on the editorial boards of the *Journal of Applied and Computational Harmonic Analysis*, *Journal of Fourier Analysis and Applications*, and *Signal Processing*. She is on the IMDSP technical committee of the Signal Processing Society of the IEEE and was the General Co-chair (with J. Allerbach) of the Ninth Workshop Image and Multidimensional Signal Processing. She received the Belgrade October Prize for student scientific achievements awarded for the Engineering Diploma thesis in October 1986, and the E. I. Jury Award at Columbia University for outstanding achievement as a graduate student in the areas of systems, communication, or signal processing.

Jianying Hu (M'93) studied electrical engineering at Tsinghua University in Beijing, China, from 1984 to 1988. She received the M.S. and Ph.D. degrees in computer science from the State University of New York at Stony Brook in 1991 and 1993, respectively.

Since 1993, Dr. Hu has been a Member of Technical Staff at Bell Laboratories, Lucent Technologies, Murray Hill, NJ. Her current research interests include document structure analysis, content based image retrieval, information retrieval, multimedia information systems, and handwriting recognition. She is a member of the ACM.

Robert H. Safranek (M'80–SM'92) was born in Manitowoc, WI, in 1958. He received the B.S.E.E. degree in 1980, the M.S.E.E. degree in 1982, and the Ph.D. degree in 1986, all from Purdue University, West Lafayette, IN. While at Purdue, he worked in the areas of parallel processing, speech coding, computer vision, and robotics. Since 1986, he has been with Bell Laboratories, Lucent Technologies, Murray Hill, NJ. His research interests include human and machine perception, digital video, networked multimedia, film and video production, and hardware/software systems for signal processing.

While at Bell Labs, he worked on a variety of problems in the area of image and video processing. He was a member of the team that developed the AT&T/Zenith submission to the U.S. HDTV standardization process. He has also been active in investigating and promoting the use of human perception in signal processing systems. Currently, he is working in the areas of networked multimedia and multimedia information retrieval.

Dr. Safranek holds 17 U.S. patents and numerous foreign patents and has published extensively. In 1995, he was awarded the IEEE Donald G. Fink Prize for his paper "Signal compression based on models of human compression" (co-authored with J. D. Johnston and N. S. Jayant). He is a member of the Eta Kappa Nu, SMPTE, and a Member of the IEEE Image and Multidimensional Signal Processing Technical Committee.

S. Kicha Ganapathy (M'82) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology (IIT), Bombay, in 1967, the M.Tech. degree in electrical engineering from IIT, Kanpur, in 1969, and the Ph.D. degree in computer science from Stanford University, Stanford, CA, in 1976 in the area of computer vision.

He was on the faculty at the University of Michigan, Ann Arbor, from 1976 to 1982, and conducted research in the areas of computer vision, computer graphics, and robotics. Since 1982, he has been with Bell Laboratories, Lucent Technologies, Murray Hill, NJ, and became a Research Department Head in 1985. In that capacity, he codirected work that put Bell Laboratories in the forefront of robotics. These include robot ping-pong player, telerobotics, and the pioneering work in silicon micromachines. Subsequently, he initiated and led the team that created virtual reality based video games, an example of which can be seen at the AT&T pavilion at Epcot Center. Currently, he is a Member of Technical Staff in the Multimedia Communications Research Laboratory, and is engaged in the area of multimedia information retrieval based on psychophysics of human perception. In particular, he is leading an R&D effort at Lucent Technologies that is applying these research ideas to the emerging field of electronic retailing and related business in e-commerce.