Extraction of Perceptually Important Colors and Similarity Measurement for Image Matching, Retrieval, and Analysis

Aleksandra Mojsilović, *Member, IEEE,* Jianying Hu, *Member, IEEE,* and Emina Soljanin

***Abstract*—Color descriptors are among the most important features used in image analysis and retrieval. Due to its compact representation and low complexity, direct histogram comparison is a commonly used technique for measuring the color similarity. However, it has many serious drawbacks, including a high degree of dependency on color codebook design, sensitivity to quantization boundaries, and inefficiency in representing images with few dominant colors. In this paper, we present a new algorithm for color matching that models behavior of the human visual system in capturing color appearance of an image. We first develop a new method for color codebook design in the Lab space. The method is well suited for creating small fixed color codebooks for image analysis, matching, and retrieval. Then we introduce a statistical technique to extract perceptually relevant colors. We also propose a new color distance measure that is based on the optimal mapping between two sets of color components representing two images. Experiments comparing the new algorithm to some existing techniques show that these novel elements lead to better match to human perception in judging image similarity in terms of color composition.**

***Index Terms*—Color composition matching, perceptual-based image matching, processing of color images, quantization of color spaces, weighted graph matching.**

I. Introduction

AMONG the various low-level features, color features have been extensively used in image database retrieval, especially in cases where no additional semantic information is available. Furthermore, color features are usually very robust to noise, image degradation, changes in size, resolution, or orientation. Color histogram, representing the joint probability of the intensities of the three color-channels, is the simplest and the most often used color feature. It is often employed in combination with the Euclidean distance as the color metric, providing undemanding yet efficient retrieval method. As improvements of a basic histogram search, numerous more sophisticated representations and metrics have been developed. For example, Swain and Ballard proposed histogram intersection, a L_1 metric, as similarity measure for histogram comparison [1]. To measure similarity between close but not identical colors Ioka and Niblac introduced an L_2-based metric [2], [3]. However, although the histogram representation is simple to compute, it lacks discriminatory power in retrieval of large image databases and does not match human perception very well. Another disadvantage of the traditional histogram approach is the huge amount of data needed for representation, which furthermore increases complexity of the indexing process. As alternatives to histograms, different research groups proposed more compact representations [4]–[6]. The first step in these methods is transformation from RGB space into a perceptually uniform color space. The image is then quantized and represented only through a smaller subset of colors. However, these algorithms still do not contain information about the spatial relationships of image pixels. For example, although high-frequency textured region and low-frequency natural scene can have identical color histograms, human eye performs the averaging operation within the small neighborhood, and we perceive their overall color quite differently. Therefore, many studies suggest that combination of histogram features and their spatial relations provides a better solution for the color matching applications. A commonly used approach to extend global color representation and include regional information is computation of *local histograms*. However, methods based on the local histograms are usually computationally demanding, storage expensive and very sensitive to translation and rotation. Consequently, retrieval results depend entirely on the metric used to compare the local histograms. More robust (with respect to the retrieval accuracy) are sophisticated *color layout* representations based on image segmentation [7], clustering [8], augmented histogram [9], and color correlograms [10].

Further improvement of color representation can be achieved only by recognizing that humans are the ultimate users of any retrieval system. Therefore, the study of human perception and content extraction are crucial for development of high-quality search engine. Many studies have discovered various aspects of similarity perception in humans. However, the problem of how to "simulate" subjectivity of human decisions to semantic information still remains the biggest obstacle in the design of retrieval system that equally performs across various application domains. Nevertheless, in [11] Rogowitz *et al.* performed a series of psychophysical experiments analyzing human perception of image content. According to the reported results, even though visual features do not capture semantic meaning of an image, there is a significant correlation with semantically relevant information. These findings indicate that, in the absence of

Manuscript received February 28, 2000; revised June 12, 2002. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Tsuhan Chen.

A. Mojsilović is with the IBM T. J. Watson Research Center, Hawthorne, NY 10532 USA (e-mail: aleksand@us.ibm.com).

J. Hu is with Avaya Labs Research, Basking Ridge, NJ 07920 USA (e-mail: jianhu@avaya.com).

E. Soljanin is with Bell Laboratories, Lucent Technologies, Murray Hill, NJ 07974 USA (e-mail: emina@research.bell-labs.com).

Digital Object Identifier 10.1109/TIP.2002.804260

1057-7149/02$17.00 © 2002 IEEE

semantic cues or relevance feedback, "semantically correct" retrieval results can be achieved by designing perceptually based feature extraction techniques. Therefore, in this work, we model behavior of the human visual system (HVS) in capturing the color appearance of an image. We propose a technique for extraction of perceptually relevant colors, design a compact feature vector and color metric that facilitates this representation. In addition to this, we address several other issues of vital importance in the color-based retrieval algorithms. These are: 1) selection of a suitable color-space [8], 2) design of quantization scheme to reduce the color resolution, and 3) selection of the well performing color codebook for color quantization.

This paper is organized as follows. In Section II, we present a brief overview of the method, while in Section III we give the basic definitions for color spaces and color images. In Section IV, we present a new method for color quantization. We first introduce sampling in two dimensions by cylindrical and spiral lattices and then extend the approach to color codebook design. Section V describes the statistical method for extraction of visually important colors, while Section VI introduce a new distance metric for measurement of color content similarity. Experimental results are given in Section VII.

II. Overview of Feature Extraction

The goal of feature extraction is to obtain compact, perceptually relevant representation of the color image content. So far, features based on the image histogram have been widely used in image retrieval. However, a feature set based solely on color histogram does not adequately model the way humans perceive color composition of an image. It has been shown that in the early perception stage, human visual system performs identification of dominant colors that best describe the overall color appearance [12]. Hence, when viewing the global color content, human visual system eliminates fine details and averages colors within small areas. Consequently, on the global level, humans perceive images only as a combination of few most prominent colors, even though the color histogram of the observed image might be very "busy" [12]. To obtain the dominant color representation, we have previously used a subset of histogram colors augmented by their area percentages [6]. However, while this feature set captures the global color distribution excellently, it still does not account for the spatial information. The goal of the proposed method is to determine dominant colors by taking into account both global color information (captured through the image histogram) and local information (extracted from the spatial relationships between the frequently occurring colors).

We perform extraction of perceived colors through the following steps: 1) transformation into the Lab color space, 2) reduction of all possible colors to a subset defined by the specially designed, compact color codebook, 3) estimation of the color distribution, and 4) spatial correlation processing to extract visually dominant colors. Dominant colors and their area percentages are then used as a feature vector in conjunction with a new distance function to measure color similarity between two images. Each of these steps will be addressed in full detail in succeeding sections.

III. Selection of the Color Space

In this section, we briefly present the basic concepts from the color theory and introduce the notation, which will be used through the paper. An $I \times J$ *image (picture)* P is a real-valued vector function defined on a rectangular grid of points (pixels) (i, j), where $i, 0 \leq i \leq I-1$, is the row index and $j, 0 \leq i \leq I-1$, is the column index. The color of pixel (i, j), denoted by $p(i, j)$, takes values in the (possibly quantized) three-dimensional (3-D) space of colors. We write the original RGB color value of the pixel p as a column vector $c_p = [r_p \;\; g_p \;\; b_p]^T$, $c_p \in C$, where C is the set of all possible colors, the vector components are red, green, and blue tristimulus values for the pixel and superscript T denotes the transpose operator. Color quantization is sampling of the 3-D color space C, which results in a discrete subset of N_Q colors known as the *color codebook*.

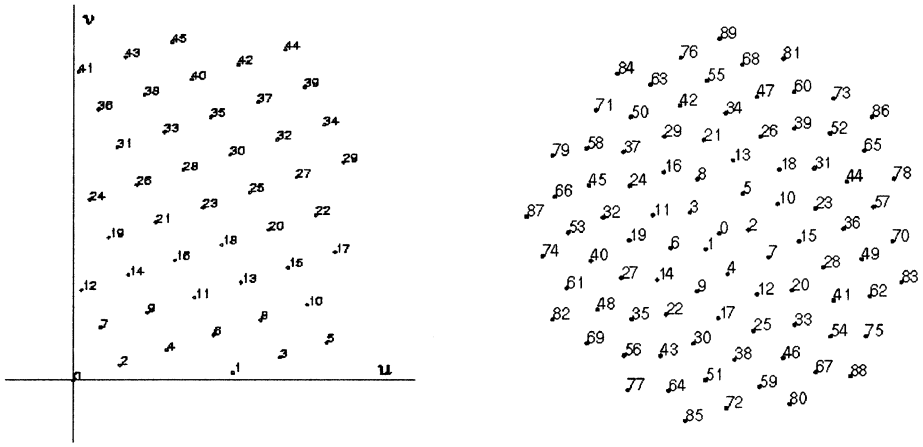
For a typical monitor, the input value of the primary and its displayed intensity are connected through the nonlinear relationship, approximated as

$$I_r = r^{\gamma} \tag{1}$$

where I_r is intensity, r is the display input, and $\gamma \approx 2.3$. Therefore, in all image capturing devices, to compensate for this nonlinearity between the input primary and its luminance, it is necessary to predistort their output by the inverse relationship

$$r = I_r^{(1/\gamma)}. \tag{2}$$

The predistorted primary color values are usually referred to as *gamma-corrected colors*. The gamma-corrected colors have the advantage that they might be displayed directly and are less susceptible to noise than the corresponding linear intensities. Therefore digital images are often represented in the gamma corrected RGB space ($R'G'B'$). However, the color difference computed as the Euclidean distance in the $R'G'B'$ space is not consistent with the human visual system; namely, just noticeable color difference is not uniform along the axes of the space [13]. It has been shown that quality of the retrieval algorithms can be substantially improved by using one of the (approximately) perceptually uniform color spaces (such as CIE Luv or Lab) designed so that fixed Euclidean distance represents a fixed perceptual distance regardless of the position in the space. Therefore, our algorithms operate in the Lab color space. The Lab representation is obtained from the $R'G'B'$ representation by first removing the gamma-correction according to (2). Next, using a linear operator the data is transformed into the XYZ color space. In the XYZ space, the data is normalized with respect to the illumination white point D65 and then converted to the Lab system via a nonlinear transform (more details on the whole procedure can be found in [13]). As a result of the nonlinear transformation, the volume of all possible colors C, approximated by a 3-D cube in the linear color space, distorts to an irregular cone. From now on we will refer to this cone as C_{Lab}. Hence, the color representation of the pixel p becomes $c_p = [L_p \;\; a_p \;\; b_p]^T$, $c_p \in C_{Lab}$.



(a) (b)

Fig. 1. (a) Points of the cylindrical lattice $w_{n,m} = u_{n,m} + jv_n = (-\alpha + j\beta)n + m$, and (b) points of the Fibonacci lattice $z_n = n^{\delta}e^{j2\pi n\tau}$ in the complex plane.

IV. Color Codebook Design

The objective of the codebook design algorithm is to sample C_{Lab} into N_Q points, where N_Q is a predetermined codebook size. In the linear color spaces (such as RGB and XYZ), approximated by 3-D cubes, color codebooks are typically obtained by performing separable uniform quantization along each coordinate axis. Unfortunately, this sampling scheme is not optimal in the Lab, due to the nonlinearity of the space. Therefore, in the Lab space color codebooks for the image retrieval applications were usually designed using vector quantization (VQ) techniques [4], [12]. When applied to the color codebook design problem, VQ algorithm determines a set of bin centers and corresponding decision boundaries so that mean-square quantization error for all pixels from all images in the particular database is minimized. Hence, in the VQ design, the training data have a crucial effect on the final result. Therefore, this approach is optimal when the codebook is designed for a particular application and the training set contains images that are representative of a given problem. However, our aim is to design a reliable color codebook to be used in different retrieval applications dealing with arbitrary input images. In that case, to obtain an accurate estimation for the color distribution, training set for the VQ algorithm has to span the whole input space. This requires a huge number of training points, resulting in a computationally expensive and possibly intractable design task. To overcome this problem, we propose a design technique that provides uniform sampling of the chromaticity planes in the Lab space and generates color codebooks that are well suited for general retrieval applications. In the proposed scheme we first sample the luminance axis into N_L levels. Then, for each discrete luminance value, we choose N_p colors in the corresponding (a, b) plane as points z_n of a hexagonal spiral lattice in the complex plane. The spiral lattices are introduced in Section IV-A, while Section IV-B describes the actual sampling algorithm. The particular sampling scheme generates the sampling points that are uniformly distributed over the area of a disk with a center at the corresponding gray point. Hence, there is neither crowding in the center nor sparsity near the edges, providing an optimal packing of the space. The proposed sampling scheme is well suited for creating small fixed color codebooks for image matching and retrieval.

A. Cylindrical and Spiral Lattices in Two Dimensions

This section gives the properties of the spiral lattices and provides the details of the sampling scheme, which was introduced in [15].

Sampling in the k-dimensional Euclidean space is commonly represented by a *lattice*. A k-dimensional lattice is specified by a set $\mathbf{a}_1, \ldots, \mathbf{a}_k$ of linearly independent real vectors; the set of all points

$$\mathbf{x} = u_1\mathbf{a}_1 + \cdots + u_k\mathbf{a}_k \tag{3}$$

with integer $u_1, \ldots, u_k$ comprise the lattice. The sampled subset of the Euclidean space consists of all lattice points, which belong to the subset. A point in the lattice is specified by a k-tuple $(u_1, \ldots, u_k)$.

In two dimensions we can equivalently consider the complex Euclidean plane $w = u + jv$. The points

$$w_{n,m} = u_{n,m} + jv_n = (-\alpha + j\beta)n + m, \quad \alpha, \beta \in \mathbf{R},\ n, m \in \mathbf{Z} \tag{4}$$

form a lattice known as *extended cylindrical lattice* [14]. Two numbers, m and n, are necessary to specify a lattice point. Since we are concerned with sampling finite regions of the complex plane, we can without loss of generality consider only points $w_{n,m}$ for which $0 < u_{n,m} \leq 1$. This is shown in Fig. 1(a) for $\alpha = \sqrt{2} - 1$ and $\beta = 0.03$. It is easy to see that the requirement $0 < u_{n,m} \leq 1$ translates into $m = [n\alpha] + 1$. Thus, each point in the two-dimensional (2-D) space can be identified only by its index n. Parameters α and β define the lattice, and thus the distribution of points in the plane. The *more irrational* the value of α, the *more uniform* the distribution of points is. Precisely, the point distribution is determined by the continued fractions of α, and the most uniform distribution is achieved when α is so called *Markoff irrational numbers*.

Because of the central symmetry of perceptually uniform color spaces, we consider the complex plane in polar coor-

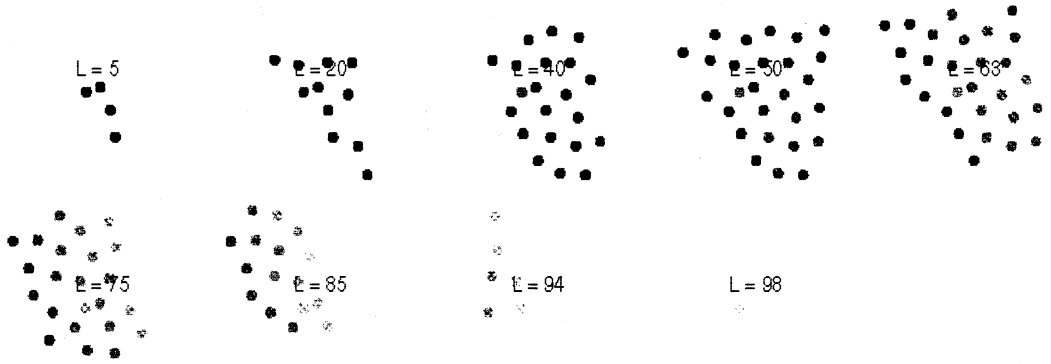


Fig. 2. Example of the Fibonacci codebook with 139 colors. The codebook is obtained by sampling the luminance planes {5, 20, 40, 50, 63, 75, 85, 94, 98} according to the spiral geometry.

dinates and define a *spiral lattice* as a convenient means for sampling. A set of points

$$z_n = n^{\delta} e^{j2\pi n\tau}, \qquad \tau, \delta \in \mathbf{R}, n \in \mathbf{Z} \tag{5}$$

constitute a spiral lattice. Fig. 1(b) shows a spiral lattice for $\tau = (\sqrt{5}-1)/2$ and $\delta = 1/2$, which we call *Fibonacci lattice*. Again, each point z_n is identified by its index n. Parameters τ and δ define the lattice and thus the distribution of points in the plane. For a given δ, parameter τ determines the axial distribution of the points. Note that for rational values of τ, e.g., $\tau = p/q$, the lattice points fall on rays radiating from the center at angles $l \cdot 2\pi p/q$, $0 \leq l < q$. Again, the axial point distribution is connected to the continued fractions of τ, and the most uniform distribution is achieved when τ is a Markoff irrational number. For a given τ, parameter δ determines the radial distribution of the points. More details about the sampling on the spiral lattices can be found in [15].

When a spiral lattice is used for sampling of color spaces, it may be advantageous to rotate it by a small angle α_o for better alignment with boundaries of the constant luminance cross-section being sampled (this will be explained in the following subsection). In this case, the lattice points are given by

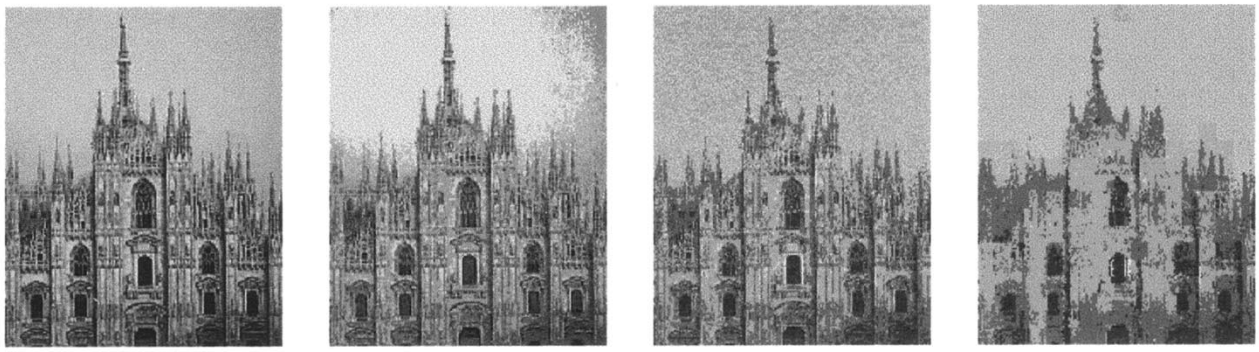
$$z_n = n^{\delta} e^{j\theta},\ \theta = 2\pi n\tau + \alpha_o, \qquad \alpha_o, \tau, \delta \in \mathbf{R}, n \in \mathbf{Z}. \tag{6}$$

B. Color Codebook Design

The objective of the codebook design algorithm is to sample the Lab space into N_Q points with respect to the spiral geometry. In the proposed scheme we first sample the luminance axis into N_L levels (unfortunately there is no generalization of the spiral sampling to three dimensions and one dimension of the color space still has to be sampled independently [15]). Then, for each discrete luminance value $L_i, i = 1, \ldots, N_L$, N_p colors in the corresponding (a, b) plane are chosen as points z_n of a hexagonal spiral lattice in the complex plane. The points are given by $z_n = s\sqrt{n}e^{j\theta}$, $\theta = 2\pi n\tau + \alpha_o$, where $\tau = (\sqrt{5}-1)/2$ is the golden mean, and s is a scaling factor determined as to ensure that the N_p-point lattice covers the region of interest in the (a, b) color plane. As a result of this sampling scheme, we obtain the set of $N_L \cdot N_p$ points. However, since a crossection of the Lab is not circular, there will be points in the Fibonacci lattice which are not valid colors in the RGB space. Therefore, in the next step, we discard all the points whose r, g, or b value is not within [0,1] range. The remaining N_Q points define the codebook. An example codebook of $N_Q = 139$ colors is shown in Fig. 2.

An important property of Fibonacci codebooks is that they are completely specified by a set of parameters that define and control the sampling process. Consequently, regardless of the codebook size, we do not need to store the real color values for each codebook entry. Instead, we need to store only a small set of parameters and use them in real time to generate the required color values. This property is of great importance for storage, transmission, and color processing applications. In addition to the compact representation and uniform distribution of points, another advantage of our sampling scheme is that, for a given index n, one can easily determine which points z_i are in the neighborhood of z_n. Hence, the structure of the Fibonacci codebook and the numbering of its points: a) significantly reduces computational cost of color quantization, b) provides easy manipulation with the colors from the codebook, and c) simplifies many image processing operations on the color quantized image [15].

Depending on the choice of the parameters, we generate codebooks with different properties. The choice of N_L, N_p, and $\{L_i\}$ is most important for the appearance of colors in the Fibonacci codebook. The value of the initial angle α_o is not crucial in the design of larger codebooks, because of the density of the sampling. However, this parameter should not be neglected in the design of smaller codebooks, since some chromaticities could be missed due to the sparse sampling in the plane. In case of a color codebook design for image retrieval, segmentation, or analysis, we need a small numbers of colors to facilitate compact representation and fast searches. In this case, one possible choice of parameters is the following: $\alpha_o = 0.05$, $N_L = 7$, $N_p = 56$, and $\{L_i\} = \{0, 10, 40, 65, 85, 94, 100\}$. The set of luminance planes is chosen through the vector quantization of the luminance values, with the large set of training images. As we already mentioned, there is no generalization of the proposed sampling scheme to three dimensions, and one dimension of the color space still has to be sampled independently. However, for a fairly general training set, the quantization of the luminance axis is less sensitive to the selection of training data than the quantization of the entire color space, to obtain a set of the best-representing colors. Therefore, the spiral sampling



(a) (b) (c) (d)

Fig. 3. Color quantization and spatial processing: (a) original image, (b) color quantized version using a codebook with 125 colors obtained by uniformly sampling the RGB space, (c) color quantized version using the Fibonacci codebook with 92 colors, and (d) image after color remapping with $N = 20$.

in the chromaticity planes provides advantage over the methods based on training data.

These parameters above generate a codebook with 92 colors, which was used in all experiments. The quality of this codebook is nicely illustrated in Fig. 3, which compares the proposed quantization algorithm to the image-independent quantization scheme based on the uniform sampling of the RGB color space.

V. Extraction of Visually Dominant Colors

After the color quantization, the resulting images often contain speckle noise. These are characterized as isolated spots of a particular color randomly distributed among the background of a different color. While commonly used smoothing methods such as median filtering or low pass filtering of the original image may reduce such artifacts, they cannot eliminate them entirely, as some of these artifacts are produced during the nonlinear quantization process. We developed a statistical method to identify colors of speckle noise and remap them to the surrounding dominant color (after the quantization). The method identifies speckle colors by evaluating neighborhood color distributions.

Let I be an image, quantized into m colors $c_1, c_2, \ldots, c_m$. For a pixel $p = (x, y) \in I$, $I(p)$ denotes its color, and $W(p, D)$ denotes its $D \times D$ neighborhood. Let $p \in I_c$ denote $p \in I$, $I(p) = c$. Then, the *neighborhood color histogram (NCH) matrix* is defined as

$$H^D_{c_i, c_j} = \Pr_{\substack{p \in I_{c_i} \\ q \in W(p, D)}} [q \in I_{c_j}]$$

and is calculated as the number of pixels of color c_j occurring within the $D \times D$ neighborhood of any pixel of color c_i, divided by the total number of pixels within the $D \times D$ neighborhood of any pixel of color c_i. The size of the neighborhood, D, depends on the estimated maximum speckle size and is typically set to 3–5. The difference between the NCH matrix and the well known color correlogram [10] is that while the color correlogram captures color distribution among pixels **at** distance k from any pixel of a particular color, NCH matrix is designed to measure color distribution **within** a predefined neighborhood of any pixel of a particular color. While the color correlogram has been shown to be effective in capturing certain spatial information in color based image indexing, the proposed NCH matrix is more suitable for the purpose of speckle color detection. Note that instead of being a square window, $W(p, D)$ could also be defined as a mask, which could be used to remove speckle noise while preserving certain structures (e.g., thin lines).

Based on the NCH matrix, speckle colors are detected and remapped in the following manner. For each color c_i, we examine row c_i of the matrix and find the entry $H^D_{c_i, c_k}$ which has the maximum value. This indicates that color c_k dominates the neighborhood of any pixel of color c_i. When $i \neq k$ and the ratio $H^D_{c_i, c_k} / H^D_{c_i, c_i}$ exceeds a predetermined threshold, color c_i occurs mostly in isolation among the background of color c_k. In that case, c_i is determined to be a speckle color and all pixels of color c_i are remapped to color c_k. Otherwise, c_i is considered a *perceptually relevant* color and no remapping is done.

Natural images typically have nonuniform color distribution and a speckle color in one region may be a perceptually revelant color in another. Therefore, each image is first divided into nonoverlapping $N \times N$ subimages (N is typically 20), and the NHC matrix computation, speckle color detection and remapping procedures are carried out independently within each subimage. The occasional blocking effect caused by such windowing does not cause a problem in this case, since we compare images by color composition only, and do not consider texture or edge features. Finally, the remaining colors from all subimages are pooled together and each color that occupies more than a predefined area percentage (i.e., 3%) is determined to be a dominant color. Typically 3–10 dominant colors are detected in each image. Non-dominant colors are than mapped to the closest dominant color. Fig. 3 illustrates the complete process of color feature extraction. Fig. 3(a) is the original image, Fig. 3(c) is the image after color quantization using a color codebook with 92 colors, and Fig. 3(d) shows the result after speckle removal and dominant color extraction.

VI. Optimal Color Composition Distance

This section introduces a new color distance called *optimal color composition distance* (OCCD). We first define a color component of an image as a pair $CC_i(c_i, p_i)$, where c_i is a color in the selected codebook and p_i is the area percentage occupied by that color. As explained in Section V, a color component CC_i

is considered to be dominant if c_i represents perceptually relevant color and p_i exceeds a predefined threshold. Hence, the color composition is represented by the set of dominant color components (DCC) found in the image. Based on human perception, for two images to be considered similar in terms of color composition, two conditions need to be satisfied. First, the colors of dominant color components of the two images need to be similar. Second, the color components with similar colors need to have similar area percentage. Therefore, the distance between two images in terms of color composition should be a measure based on the optimal mapping between the DCCs of the two images.

Previously proposed color metrics all fail to fully capture both factors. The most naive metric, the Euclidean distance between the color histograms, captures the second factor well. However, it is too rigid with regard to the first factor: the area percentages are only compared for color components with exactly the same color. As a result, it overestimates distances [19]. Swain and Ballard proposed a modified metric called the histogram intersection (HI) method [1]. In this method the similarity between two images is measured by

$$\frac{\sum_{i=1}^{N} \min\left(\hat{H}_Q(i), \hat{H}_T(i)\right)}{\sum_{i=1}^{N} \hat{H}_Q(i)} \tag{7}$$

where: $\hat{H}_Q(i)$ and $\hat{H}_T(i)$ represent number of pixels in the bin i, for the query and target image, respectively. While achieving improved results compared to L_1 or L_2 norm, this method still suffers from the same fundamental problem, which is that no consideration is given to two colors that are similar but are in different histogram bins. The weighted quadratic distance metric [3], [21] developed for the QBIC system, aimed at alleviating this problem. In this metric the distance between two color histograms is computed as

$$d^2_{hist}(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})' \mathbf{A} (\mathbf{x} - \mathbf{y})$$

where: $\mathbf{A} = [a_{ij}]$ is a matrix and weight a_{ij} denotes similarity between color bins i and j. This method allows comparison between different bins through weighting using the similarity matrix. However, it underestimates distances because it tends to accentuate the similarity of color distributions without a pronounced mode [19]. Pei and Cheng proposed a method, which allows more flexible color matching using a dynamic programming approach [5]. However they disregard the area percentage factor completely. In [4], Ma *et al.* proposed a more balanced metric (referred to as MDM metric henceforth). They define the distance between two color components to be the product of the difference in area percentage and the difference in color space, i.e.,

$$D(CC_i, CC_j) = |p_i - p_j| \times W(c_i, c_j) \tag{8}$$

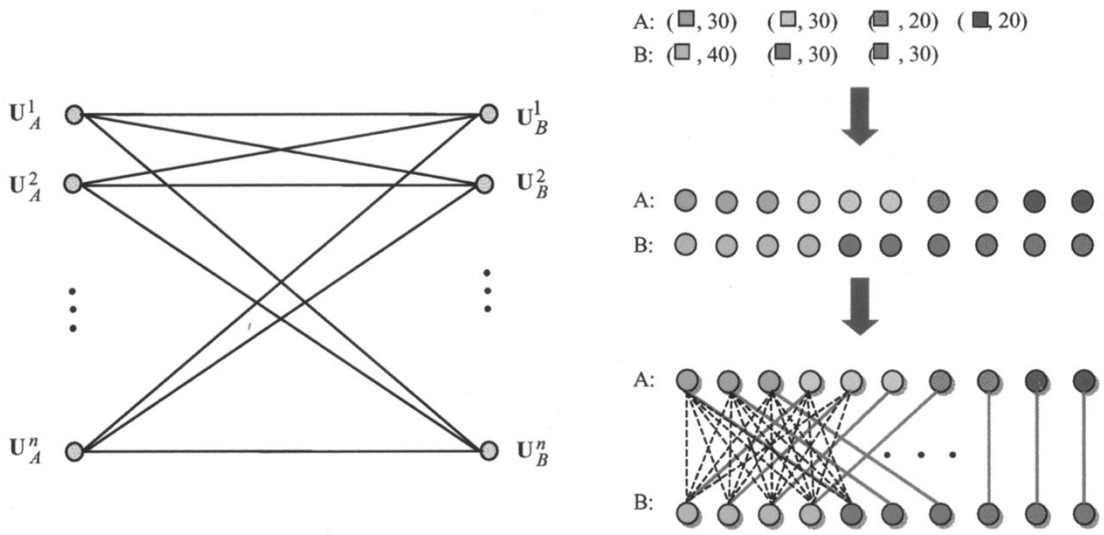
where $W(c_i, c_j)$ is the distance between colors c_i and c_j in a given color space. Then, for each DCC in image A, the best matching DCC in image B is defined to be the DCC with the closest color. Since the resulted match is not symmetric, the same procedure is performed to find the matching DCC in the image A for each DCC in the image B. Finally, the distance between the images A and B is then defined as the sum of the distances between the matching DCCs. This method has two serious drawbacks. First, the definition of the distance between two color components is problematic. Since it is defined to be the product of two components, when either component is very small the remaining component becomes irrelevant. For example, in an extreme case, two color components with the same area percentage but very different colors will have distance 0, which is very counter-intuitive. The second drawback is that the distance used in the search for matching DCCs (which is the color distance alone) is different from the distance used in the final metric (which is the product of color distance and area percentage distance). This discrepancy often leads to unexpected results. In [12] Mojsilovic *et al.* proposed a modified version of this metric. They defined the distance between two color components to be the sum of the difference in percentage area and the difference in color space

$$D(CC_i, CC_j) = |p_i - p_j| + W(c_i, c_j) \tag{9}$$

where both terms are normalized to range [0,1]. Then, a greedy search is performed where for each DCC in A, the DCC in B with smallest color component distance is chosen as the matching DCC. The procedure is then repeated for each DCC in B. Finally, the distance between the two images is computed as the sum of the distances between the color components. This metric does not have the drawbacks of MDM metric. However, the greedy search does not always find the optimal match between the color components of two images.

Both MDM and modified MDM methods are early attempts at providing a distance measure that takes into account both color and area percentage factor. Unfortunately, in both cases neither the metric itself nor the distance computation is well defined. This drawback often offsets the advantage brought by considering both factors and eventually leads to unsatisfactory results, as can be seen in some examples provided later.

To overcome these problems we define a new metric, called *optimal color composition distance* (OCCD). The optimal color composition distance measures the difference between two images in terms of color components based on the optimal mapping between the two corresponding sets of color components. First, the set of dominant color components of each image is quantized into a set of n color units, each with the same area percentage p, where $n \times p = 100$. We call this set the *quantized color component set*. Apparently different color units in this quantized set may have the same color, and the number of color units labeled with a particular color c_i is proportional to the corresponding area percentage p_i. Since every unit now has the same area percentage, it suffices to label each by the color alone. Thus, the color composition of the image is now represented as a set of n labeled units. Suppose we have two images A and B, with the quantized color component sets $\{C_A|U^1_A, U^2_A, \ldots, U^n_A\}$ and $\{C_B|U^1_B, U^2_B, \ldots, U^n_B\}$. Let $I(U^k_x)$, $x = A, B$ and $k = 1, \ldots, n$ denote the color of unit U^k_x, and let $\{M_{AB}|m_{AB}\colon C_A \rightarrow C_B\}$ be the set of one-to-one mapping



(a) (b)

Fig. 4. (a) Graph G_{AB} used to compute the color distance. (b) An illustration of OCCD computation.

functions from set C_A to set C_B. Each mapping function defines a mapping distance between the two sets

$$MD(C_A, C_B) = \sum_{i=1}^{N} W(I(U_A^i), I(m_{AB}(U_A^i))) \qquad (10)$$

where $W(i, j)$ is the distance between color i and color j in a given color codebook. Our goal is to find the optimal mapping function, denoted $\overline{m}_{AB}$, which minimizes $MD(C_A, C_B)$. The distance between the images A and B is then defined to be the *minimal mapping distance*, denoted by $\overline{MD}(C_A, C_B)$. Note that this distance is symmetric, i.e.,

$$\overline{MD}(C_A, C_B) = \overline{MD}(C_B, C_A). \qquad (11)$$

This optimization problem can be formulated as the so called *minimum cost graph matching problem*. Given an undirected graph defined by a set of nodes and edges associated with costs, the minimum cost graph matching problem is the problem of finding the set of disjoint edges (i.e., no two edges in the set share the same node) with the minimum total cost. This problem can also be equivalently formulated as the maximum weighted graph matching. It is a well studied problem in the computer algorithms community. The complexity of the solution for a general bipartite graph is $O(n^3)$, where n is the number of nodes in the graph [17].

Given two images A and B with their corresponding quantized color component sets C_A and C_B defined above, we create a graph G_{AB} as shown in Fig. 4(a). The graph contains $2n$ nodes, one for each color unit in C_A or C_B, and n^2 edges, one between each node in C_A and each node in C_B. The cost for an edge is defined to be the distance between the two corresponding colors. The resulting graph is an undirected, bipartite graph. Clearly, the solution to the minimum cost graph matching problem for graph G_{AB} gives us the solution to the minimum mapping distance.

To compute the minimum mapping distance in our system we use Rothberg's implementation (available at ftp://dimacs.rutgers.edu/pub/netflow/matching/weighted/solver-1) of Gabow's algorithm [18]. Since Gabow's algorithm solves the maximum weighted graph matching problem and assumes positive weights, G_{AB} is first modified such that each edge is assigned a weight which is inversely proportional to the cost (e.g., $C_{\max} - cost$, where $C_{\max}$ is the maximum possible cost). The algorithm applies to general graphs as well as bipartite graphs with complexity $O(n^3)$.

Fig. 4(b) demonstrates the OCCD computation between two hypothetical images A and B, assuming $p = 10$. On the top are the color components (including the area percentages) for the two images. In the middle are the quantized color units. The graph matching process is illustrated at the bottom, where the blue edges indicate the optimal mapping between A and B. As seen in this example, OCCD can handle images with different numbers of dominant color components and it can handle partial matching down to the size of the color units (p).

In real applications, the parameters n and p are chosen through perceptual experiments. Subjective experiments with the images from our databases have shown that, with the current codebook, colors that occupy less than 5% of image are usually not perceived. Also, previously reported subjective experiments demonstrate that humans are not able to perceive large number of colors within an image (typically not more than 10) [6]. Hence, we found it sufficient to assign $n = 20$ and $p = 5$, allowing up to 20 dominant color components. As we mentioned before, images with less than 20 dominant colors will simply result in representations containing multiple color units with identical color labels.

The OCCD metric described above is similar in spirit to the so-called Earth Mover's Distance (EMD) proposed recently [20]. The main difference is that in EMD, the optimal mapping between two sets of color components (called signatures in [20]) is directly computed using linear programming, while in OCCD the color components are first quantized into color units and the optimal mapping is computed using weighted graph matching. The advantage of the proposed metric is that unlike

linear programming, the weighted graph matching problem can be solve by a deterministic algorithm with guaranteed complexity of $O(n^3)$. Furthermore, because the color components are already quantized, OCCD can be more easily extended to include measures comparing local shape or texture.

Although the $O(n^3)$ complexity of OCCD appears much higher than that required to compute metrics such as histogram intersection [1] (linear) or the weighted quadratic distance [21] (quadratic), a closer examination reveals that it is not so. In OCCD, the base n is the number of quantized color units, which is completely independent of the codebook size and typically a small number (20 or less). On the other hand, the base in histogram intersection or quadratic distance corresponds to the codebook size, which is often very large (typically from $\sim$100 to $\sim$1000). Considering this difference, the complexity of OCCD is in fact comparable to that of quadratic distances. Further more, the fact that the execution time of OCCD is independent of the codebook size is attractive, since it enables the use of very large codebooks if required by an application. For use on large databases, standard speed-up methods, such as pre-filtering using a coarser but faster metric [20], can be applied.

VII. Experimental Results

In this section, we illustrate the performance of the new OCCD algorithm and compare it with several previously proposed methods. The comparison includes: 1) the simplest retrieval scheme based on the Euclidean distance between the target and query histograms, 2) Swain's histogram intersection method [1], 3) MDM metric [4], and 4) modified MDM metric as defined in [6]. For the comparison we used an interior design database, consisting of 335 color patterns. This database was chosen because there is little meaning attached to it—in that way we were able to test our scheme and compare it to other methods without too much bias of the semantic information. Also, since our main purpose is to test how well the algorithms match human perception in terms of identifying similar (not identical) images based on appearance as opposed to efficiency of the algorithms, we feel that a larger database is not needed. Furthermore, a database of moderate size is more suitable for carrying out subjective experiments, which are often considered as the best approach for assessing the perceptual qualities of an algorithm.

In the examples shown below, all retrieval results are displayed with the query image at the upper-left corner. The five retrieved images are arranged from left to right and top to bottom in order of decreasing similarity.

Fig. 5 illustrates the performances of traditional histogram method [Fig. 5(a)] and histogram intersection [Fig. 5(b)] compared to that of OCCD [Fig. 5(c)]. In this example, both histogram based methods fail to retrieve some of the very similar images, because they ignore colors that are close to each other and yet happen to fall into different quantization bins. OCCD succeeds to retrieve these images by allowing flexibility in comparing both the colors and the area percentages they occupy. Fig. 6 shows a comparison of the performances of the MDM method [Fig. 6(a)], the modified MDM method [Fig. 6(b)], and the new OCCD method [Fig. 6(c)]. In this case both of the MDM and modified MDM methods perform poorly, with the former biased toward the background and the latter biased toward the

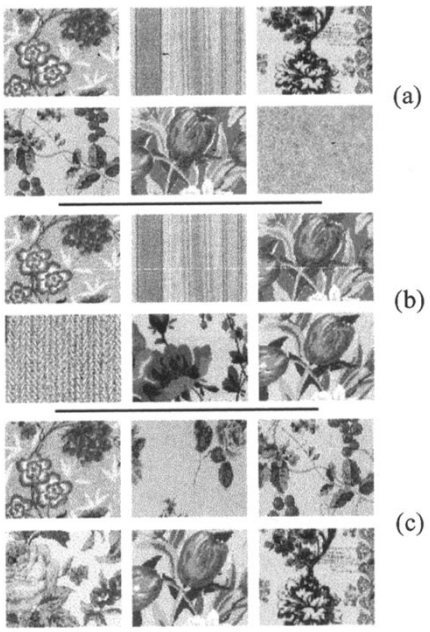


Fig. 5. Retrieval results using: (a) traditional histogram, (b) histogram intersection, and (c) OCCD methods.

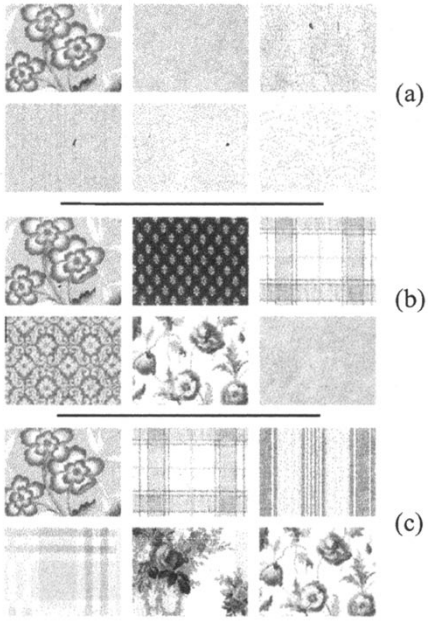


Fig. 6. Retrieval results using: (a) MDM, (b) modified MDM, and (c) OCCD methods.

TABLE I
AVERAGE RANK FOR THE IMAGE RETRIEVAL SCHEMES COMPARED IN THE SUBJECTIVE EXPERIMENT. IN THE SUBJECTIVE RANKINGS, 1 INDICATES THE BEST SCHEME, WHILE 4 INDICATES THE WORST SCHEME

HI	MDM	Modified MDM	OCCD
2.3	2.8	3.3	1.6



Fig. 7. Retrieval results from "Nature" database. The query image is on the left, followed by six best matches.

foreground. As mentioned before, such unpredictable behavior is likely caused by the fact that the metric is ill defined in both methods. OCCD provides a well defined metric and guaranteed optimality in distance computation, thus achieving much better results.

A subjective experiment was carried out to quantitatively compare the different methods. The traditional histogram method was not included in this experiment because its problems are well established. Moreover, including too many schemes in the experiment tends to get the subjects more confused. Fifteen representative patterns were chosen from the interior design database as query images. For each query image, four different retrieval results were generated using the histogram intersection method, MDM method, modified MDM method, and the OCCD method, respectively. Each query result contained the top five matching images retrieved from the pattern database, displayed in the same manor as the examples in Figs. 3 and 4. Thirteen subjects (seven men and six women) were asked to evaluate these results. Each subject was presented the four retrieval results for each query image, arranged in random order on one page, and asked to rank order them based on how they matched their own judgment. A ranking of 1 was given to the best retrieval result, 2 to the second best result, etc.

We expected large variations in the evaluation results due to several factors. First, even though the subjects were instructed to concentrate on color similarity alone and ignore other factors such as pattern and spatial arrangement of colors, many subjects reported afterward that it was difficult to do so. An examination of the results showed that some subjects were better at separating the different factors than others. Second, comparing rankings in general is a difficult task. Different people often use slightly different rules. Finally, people differ in judging color similarity itself. For example, some may concentrate more on foreground while others more on background. Despite these factors, the evaluation results showed significant amount of consistency among different subjects for most images. Ten of the 15 images used yielded majority votes (e.g., >6) for a single scheme as the best, indicating that the corresponding rankings were reasonably consistent. We discarded the remaining five images, since the rankings were too scattered to be interpreted reliably. Of the ten query images yielding consistent rankings, the results produced by OCCD method had majority votes as the best method for eight images, and the results produced by histogram intersection were voted best for two images. Table I gives the average rank for each scheme computed from these ten query images, where OCCD clearly has the best ranking. These results demonstrate that the OCCD method indeed best matches human perception overall.

To demonstrate the effectiveness of the proposed algorithm in general retrieval applications, we include example from the Corel database, shown in Fig. 7, and two examples from the database of paintings (600 images) shown in Figs. 8 and 9. However, we would like to emphasize that color features alone (although very important) do not provide sufficient information for searching general, broad-context image databases. It has been shown that color features correlate with some semantic concepts, but cannot be used as a single measure to capture the semantic meaning of images [11], [22]. Hence, the proposed method should be seen as a method for measuring and comparing color similarity and color composition, rather than a complete solution for content based image retrieval.

VIII. Discussion and Conclusions

Due to very compact representation and low complexity, color quantization with respect to the fixed codebook, followed by the histogram computation is the most often used feature extraction technique in image database retrieval. However, despite these desirable properties there are some major drawbacks to this approach. First, discrimination ability of the color histogram largely depends on the selection of the color quantization method, as well as on the size of the color codebook.



Fig. 8. Results from the database of paintings (600 images). The query image is on the left, followed by six best matches.



Fig. 9. Results from the database of paintings (600 images). The query image is on the left, followed by six best matches.

Second, for most natural images histogram representation is very sparse, and therefore matching based entirely on the comparison between the query and target histogram is not efficient. Furthermore, when images are quantized with respect to the fixed color codebook, histogram comparison becomes sensitive to quantization boundaries. Namely, if two very similar colors are located on different sides of the quantization boundary, they will be quantized into different bins. Thus, if a matching function performs very simple histogram comparison these two colors will not be compared at all, resulting in significant decrease in the performance of the retrieval system. Therefore, the objective of this work was to design a compact color representation and computationally inexpensive yet effective comparison scheme, having all the desirable properties, while overcoming the drawbacks of the traditional histogram-based measures of image similarity. It is important to point that the proposed statistical procedure and color remapping can be seen as the preprocessing technique to generate color histogram that better matches human perception. Hence, it can be used with any histogram-based feature extraction algorithm to improve the quality of representation. Also, although the proposed OCCD metric is used with a fixed codebook here, it can be equally applied to features extracted using multiple codebooks such as image dependent codebooks. Finally, the proposed technique represents an initial step in computing the spatial relationships between the colors, generating the color layout or segmenting the image for further analysis.

REFERENCES

[1] M. Swain and D. Ballard, "Color indexing," *Int. J. Comput. Vis.*, vol. 7, no. 1, pp. 11–32, 1991.

[2] 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.

[3] W. Niblack, R. Berber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, and P. Yanker, "The QBIC project: Quering images by content using color, texture and shape," *Proc. SPIE*, pp. 172–187, 1994.

[4] 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.

[5] S. Pei and C. Cheng, "Color image processing by using binary quaternion moment preserving technique," *IEEE Trans. Image Processing*, vol. 8, pp. 614–628, May 1999.
[6] A. Mojsilovic, J. Kovacevic, D. Kall, R. J. Safranek, and K. Ganapathy, "Toward vocabulary and grammar of color patterns," IEEE Trans. Image Processing, to be published.
[7] J. R. Smith and S.-F. Chang, "Single color extraction and image query," in *Proc. IEEE Int. Conf. Image Processing*, Washington, DC, Oct. 1995.
[8] X. Wang and C.-C. J. Kuo, "A new approach to image retrieval with hierarchical color clustering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 628–643, Sept. 1998.
[9] Y. Chen and E. K. Wong, "Augmented image histogram for image and video similarity search," *Proc. SPIE*, Jan. 1999.
[10] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image indexing using color correlogram," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, PR, June 1997.
[11] B. Rogowitz, T. Frese, J. Smith, C. A. Bouman, and E. Kalin, "Perceptual image similarity experiments," *Proc. SPIE*, 1997.
[12] A. Mojsilovic, J. Kovacevic, J. Hu, R. J. Safranek, and K. Ganapathy, "Matching and retrieval based on the vocabulary and grammar of color patterns," *IEEE Trans. Image Processing*, vol. 9, pp. 38–54, Jan. 2000.
[13] G. Wyszecki and W. S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*. New York: Wiley, 1982.
[14] F. Rothen and A. Koch, "Phylotaxis or the properties of spiral lattices," *J. Phys.*, vol. 50, pp. 633–657, Mar. 1989.
[15] A. Mojsilovic and E. Soljanin, "Color quantization and processing by fibonacci lattices," *IEEE Trans. Image Processing*, vol. , pp. 1712–1725, Nov. 2001.
[16] R. M. Haralick, "Statistical and structural approaches to texture," *Proc. IEEE*, vol. 67, pp. 786–804, May 1979.
[17] E. Lawler, *Combinatorial Optimization: Networks and Matroids*. New York: Holt, Rinehart, and Winston, 1976.
[18] H. Gabow, "Implementation of algorithms for maximum matching on nonbipartite graphs," Ph.D. dissertation, Stanford Univ., Stanford, CA, 1973.
[19] M. Stricker and M. Orengo, "Similarity of color images," *Proc. SPIE*, pp. 381–392, 1995.
[20] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distribution with applications to image databases," in *Proc. 1998 IEEE Int. Conf. Computer Vision*, Bombay, India, 1998.
[21] J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack, "Efficient color histogram indexing for quadratic form distance functions," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, no. 7, July 1995.
[22] A. Mojsilovic and B. Rogowitz, "Capturing image semantics with low-level descriptors," in *Proc. Int. Conf. Image Processing 2001*, Thessaloniki, Greece, Oct. 2001.



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

From 1994 to 1998, she was a Member of Academic Staff at the University of Belgrade. From 1998 to 2000, she was with Bell Laboratories, Lucent Technologies, Murray Hill, NJ. She is currently a Research Staff Member at the IBM T. J. Watson Research Center, Hawthorne, NY. Her main research interests include computer vision, image processing, multimedia, multidimensional signal processing, medical imaging, and human perception.

Dr. Mojsilovic received the IEEE Signal Processing Society Young Author Best Paper Award for her paper on image retrieval coauthored with J. Kovačević, J. Hu, R. Safranek, and K. Ganapathy. She currently serves as an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING.



Jianying Hu (S'91–M'93) studied electrical engineering at Tsinghua University, Beijing, China, before coming to the United States in 1988. She received the M.S. and Ph.D. degrees from the Computer Science Department, State University of New York, Stony Brook, in 1991 and 1993, respectively.

She joined Bell Labs in April 1993 as a Member of Technical Staff and transferred to one of its spin-offs, Avaya Labs Research, in October 2000. Her main research interests include image and video analysis, document understanding, pen computing, and multimedia indexing, summarization, and retrieval.

Dr .Hu is an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING and a member of the Industrial Liaison Committee of the International Association for Pattern Recognition (IAPR). She co-chaired the 1st International Workshop on Web Document Analysis (WDA2001), and is a co-chair for the 5th International Workshop on Document Analysis Systems (DAS2002), and the SPIE Conference on Document Recognition and Retrieval X. She is a member of the ACM.



Emina Soljanin received the B.S. degree in electrical engineering from Sarajevo University, Bosnia and Herzegovina in 1986, and the M.S. and Ph.D. degrees in electrical engineering from Texas A&M University, College Station, in 1989 and 1994.

From 1986 to 1988, she worked in the Energoinvest Company, Bosnia and Herzegovina, developing optimization algorithms and software for power system control. After grduating from Texas A&M, she joined Bell Laboratories, Murray Hill, NJ, as a Member of Technical Staff, Mathematical Sciences Research Center. Her research interests are in the broad area of communications, information and coding theory, their applications to storage systems, and more recently quantum information theory.

Dr. Soljanin was the recipient of the 1992 Texas A&M University Electrical Engineering Department Fouraker fellowship. She served as a Technical Proofreader (1990–1992) and Associate Editor for Coding Techniques (1997–2000) for the IEEE TRANSACTIONS ON INFORMATION THEORY.