



PERGAMON

Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE @ DIRECT®

Pattern Recognition 37 (2004) 1629–1640

**PATTERN
RECOGNITION**

THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Classification with color and texture: jointly or separately?

Topi Mäenpää*, Matti Pietikäinen

Machine Vision Group, Infotech Oulu, Department of Electrical and Information Engineering, P.O. Box 4500, University of Oulu, Oulu FIN-90014, Finland

Received 15 April 2003; accepted 20 November 2003

Abstract

Current approaches to color texture analysis can be roughly divided into two categories: methods that process color and texture information separately, and those that consider color and texture a joint phenomenon. In this paper, both approaches are empirically evaluated with a large set of natural color textures. The classification performance of color indexing methods is compared to gray-scale and color texture methods, and to combined color and texture methods, in static and varying illumination conditions. Based on the results, we argue that color and texture are separate phenomena that can, or even should, be treated individually.

© 2004 Published by Elsevier Ltd on behalf of Pattern Recognition Society.

Keywords: Color indexing; Gabor; Local binary pattern; Vistex; Outex

1. Introduction

The use of joint color-texture features has been a popular approach to color texture analysis. One of the first methods allowing spatial interactions within and between spectral bands was proposed by Rosenfeld et al. [1]. Statistics derived from co-occurrence matrices and difference histograms were considered as texture descriptors. Panjwani and Healey introduced a Markov random field model for color images which captures spatial interaction both within and between color bands [2]. Jain and Healey proposed a multiscale representation including unichrome features computed from each spectral band separately, as well as opponent color features that capture the spatial interaction between spectral bands [3]. Recently, a number of other approaches allowing spatial interactions have been proposed.

In some approaches, only the spatial interactions within bands are considered. For example, Caelli and Reye proposed a method which extracts features from three spectral channels by using three multiscale isotropic filters [4].

Paschos compared the effectiveness of different color spaces when Gabor features computed separately for each channel were used as color texture descriptors [5]. Segmentation of color images by taking into account the interaction between color and spatial frequency of patterns was proposed by Mirmehdi and Petrou [6]. An interesting approach to combining color and texture with histogram ratio features was recently published by Paschos and Petrou [7]. For a related work on color constant ratio gradients, see Ref. [8].

Another way of analyzing color textures is to divide the color signal into luminance and chrominance components, and process them separately. A number of approaches using this principle have also been proposed. Tan and Kittler extracted texture features based on the discrete cosine transform from a gray level image, while measures derived from color histograms were used for color description. A color granite classification problem was used as a test bed for the method [9]. Dubuisson-Jolly and Gupta proposed a method for aerial image segmentation, in which likelihoods are computed independently in color and texture spaces. Then, the final segmentation is obtained by evaluating the certainty with which each classifier (color or texture alone) would make a decision [10].

To our knowledge, there have been no published articles on whether color and texture should be processed jointly or

* Corresponding author. Tel.: +358-8-553-2796; fax: +358-8-553-2612.

E-mail addresses: topioli@ee.oulu.fi (T. Mäenpää), mkp@ee.oulu.fi (M. Pietikäinen).

separately. Recently, the issue was touched by Drimbarean and Whelan in a paper that evaluated the contribution of color information to the classification of color textures [11]. The amount of data in the experimental evaluation was however quite limited, and different illumination conditions were not taken into account. Furthermore, opponent color texture features were not considered, and no attempts were made to combine color indexing methods and texture features.

In this paper, color image recognition schemes are empirically compared with a large collection of natural color images from two image databases. Unlike many previous studies, we do not restrict the evaluation to just texture measures. Instead, color indexing, gray-scale texture, channel-wise color texture and opponent color texture methods are considered. Furthermore, four different methods of combining separate color and texture measures on a higher level are evaluated. The performance of these methods is evaluated in constant and varying illumination conditions, and with a number of different color spaces. The goal is to find out whether the increasingly popular approach of using joint color texture measures should be preferred over separate color and texture features.

2. Perception of color textures

Research on the human visual system has provided much evidence that the image signal is composed of a luminance and a chrominance component. Both of these are processed by separate pathways [12,13], although there are some secondary interactions between the pathways [14]. The psychophysical studies of Poirson and Wandell suggest that color and pattern information are processed separately [15].

In their recent paper on the vocabulary and grammar of color patterns, Mojsilovic et al. suggest that the overall perception of color patterns is formed through the interaction of a luminance component, a chrominance component and an achromatic pattern component [16]. The luminance and chrominance components approximate signal representation in early visual cortical areas while the achromatic pattern component approximates the signal formed at higher processing levels. The luminance and chrominance components are used in extracting color-based information, while the achromatic pattern component is utilized as texture pattern information. In their study, features are extracted by combining three major domains: a nonoriented luminance domain represented by the luminance component, an oriented luminance domain represented by the achromatic pattern map, and a nonoriented color domain represented by the chrominance component [16].

Mojsilovic et al. conclude that human perception of pattern is unrelated to the color content of an image. The strongest dimensions are “overall color” (presence/absence of dominant color) and “color purity” (degree of colorfulness), indicating that, at the coarsest level of judgment, people primarily use color information to judge similarity. The

pure texture-based dimensions are “directionality and orientation” and “regularity and placement rules”. The optional fifth dimension, “pattern complexity and heaviness” (a dimension of general impression), appears to contain both chrominance and luminance information perceived, for example, as “light”, “soft”, “heavy”, “busy” and “sharp”.

In the human eye, chrominance is processed at a lower spatial frequency than luminance [17]. This fact is utilized, for example, in image compression and in imaging sensors. Furthermore, our past experience in texture analysis shows that much of the discriminative texture information is contained in high spatial frequencies such as edges [18,19]. Putting these together it seems that texture information should be extracted from the luminance component, whereas color is more a regional property.

3. Experimental setup

Three experiments with two different texture sets were arranged. The sets included 54 color textures from the Vision Texture database [20], and 68 color textures from the Outex texture database [21]. The Vision Texture database contains a large set of natural color textures taken under noncontrolled conditions. There is no knowledge of illumination sources, imaging geometries, image scale, gamma correction, camera primaries, etc. Furthermore, the images are acquired with many different cameras. Therefore, these textures must be treated “as they are”.

Outex, on the other hand, provides an even larger set of textures acquired under strictly controlled conditions. The illumination sources and the imaging geometry are known, as well as the characteristics of the imaging equipment. Gamma correction is not used, making the color channels linear. Each texture sample in the database is imaged under three different illumination sources, nine rotation angles and six spatial resolutions. Thus, there are 162 different images of each texture sample. In contrast to VisTex, the textures in Outex have been imaged with a three-CCD digital camera.

The selected VisTex textures are shown in Fig. 1. The image set consists of different natural surfaces including grass, bark, flowers, trees, and even food. They all have quite easily distinguishable color and texture properties. The Outex textures, shown in the reference illumination in Fig. 2, are a more challenging set. There are different subsets of images that contain color and/or textures very similar to each other, in addition to some pick and mix textures. There are 15 different types of canvases, nine different carpets, seven different ceramic tiles, ten flavors of granite rock, eight sandpapers with different roughness, and 11 mixtures of barley and rice. Many of the textures have very similar color. The barley and rice mixtures also have very similar textural structures.

First, the 54 VisTex textures were split into 128×128 pixel sub-images. Since the size of the original images was 512×512 , this makes up a total of 16 sub-images per texture.



Fig. 1. VisTex textures.



Fig. 2. Outex textures.

Half of the samples from each texture were used in training while the rest served as testing data. A checkerboard pattern was used in dividing the sub-images into two sets, the upper left sub-image being the first training sample. This data was submitted to the Outex site as test suite Contrib_TC_00006.

Second, a set of 68 Outex textures were treated in a similar manner. In this case, the total number of sub-images per texture was 20 due to the fact that the original size of the images was 746×538 pixels. Again, the division into training and testing sets was accomplished using the aforementioned checkerboard pattern. Thus, there were 680 samples in both the training and the test set. The selected textures

were imaged with a 100 dpi resolution at 0° rotation, and illuminated with a 2856 K incandescent CIE A light source (reference illumination). At the Outex site, this test suite has the ID Outex_TC_00013.

Third, the 68 Outex textures were used as training data. As test samples, two differently illuminated samples of the very same textures were utilized. The illumination sources were 2300 K horizon sunlight and 4000 K fluorescent TL84. Besides the spectrum, the three illumination sources slightly differ in positions, which produces varying local shadowing. Also, there are variations in the intensities of the light sources. For these reasons, this problem can be considered a

quite challenging test of illumination invariance. The numbers of training and testing samples in this test were 680 and 1360, respectively. This test suite has the Outex ID Outex_TC_00014.

For feature extraction, a number of color, texture and color texture features were selected. The goal in selecting feature extraction methods was to collect a representative set of both straightforwardly implementable and simple methods, and commonly accepted state-of-the-art methods.

4. Color indexing

4.1. Color spaces

The *RGB* tristimulus values are very sensitive to illumination intensity and color changes. Even though good color discrimination accuracy can be achieved with *RGB* distributions under controlled illumination conditions, their performance is not stable in varying conditions. This effect can be reduced with normalization or color constancy algorithms [22]. A problem with these is that useful information may be lost. Another way to cope with the problem is to use an indexing method that can tolerate changes in *RGB* values [23].

The use of color spaces other than *RGB* may also aid in achieving invariance against illumination changes. For example, the *HSV* color space separates chromaticity and intensity information, thereby providing a way to get intensity-invariant chromaticity measures. Chromaticity coordinates are another way of eliminating the effect of intensity on color values. There are also a number of methods for deriving custom color spaces, see, for example, Ref. [24]. In the experiments, five different color spaces were investigated: *RGB*, *rg* chromaticity coordinates, the approximated K-L expansion-based color system ($I_1I_2I_3$) introduced by Ohta et al. [25], the perceptually uniform CIE 1976 ($L^*a^*b^*$), and *HSV*.

To use the $L^*a^*b^*$, one must find a way of converting *RGB* to *XYZ*. The spectral sensitivities of the primaries of the imaging equipment are often unknown. In these cases, it is customary to assume that they can be closely approximated by some standard set of primaries, like the *Recommendation ITU-R BT.709* [26]. Furthermore, if the spectral power density (SPD) of the white point is not known, D65 is typically assumed. Given these assumptions, one can use a linear transformation to obtain *XYZ* from *RGB* [27]:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{pmatrix} \times \begin{pmatrix} R_{709} \\ G_{709} \\ B_{709} \end{pmatrix}. \quad (1)$$

Obviously, the values obtained this way generally are not the “real” *XYZ* tristimuli. The situation becomes a bit different if the SPD of the white point and the spectral sensitivities of the primaries of the imaging equipment are known, because this information makes it possible to derive a transformation from custom *RGB* to *XYZ*. It should also be noted that a linear transformation is not the only possibility, but it might be the most convenient way. The “standard” transformation matrix was used in converting the *RGB* values in VisTex images to their *XYZ* correspondents.

The method of deriving *XYZ* tristimuli from known *RGB* was that presented in Ref. [28]. The transformation equation for the Outex textures (Sony DXC-755P camera and CIE A white point) becomes

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.9176 & 0.1242 & 0.0550 \\ 0.6690 & 0.3165 & 0.0145 \\ 0.0013 & 0.0475 & 0.3068 \end{pmatrix} \begin{pmatrix} R_{Outex} \\ G_{Outex} \\ B_{Outex} \end{pmatrix}. \quad (2)$$

The transformation matrix is quite different from the “standard” one (Eq. (1)), which clearly indicates there is no universal transformation from *RGB* to *XYZ*.

4.2. Color histograms

Color histograms were used as simple color features. The color indexing method of Swain and Ballard was adopted for discrimination between histograms [29]. In this method, histograms are compared using a similarity measure called histogram intersection that sums up the minimum values of two histograms for each histogram bin:

$$\begin{aligned} d(H_1(c_1, \dots, c_N), H_2(c_1, \dots, c_N)) \\ = \sum_{i_1, \dots, i_N} \min[H_1(i_1, \dots, i_N), H_2(i_1, \dots, i_N)], \end{aligned} \quad (3)$$

where H_1 and H_2 are N -dimensional color distributions. Here, normalization against changes in image size is not needed because the images in each experiment are of the same size.

To make the implicit assumption of uniformly distributed color values true, the quantization method presented in Refs. [18,30] was used. The method works by selecting a large amount of random pixels—or all the pixels—from training data. An N -dimensional probability density function (histogram) $H(c_1, \dots, c_N)$ is constructed out of these pixels. Obviously, in the case of the *RGB* color space, N equals to three. Let us denote the N marginal distributions (channel-wise histograms) of this pdf by $h_n(c)$, $n \in [1, N]$. When a quantization to M levels is needed, $M + 1$ pieces of histogram bin boundaries b_m , $m \in [0, M]$ are derived from $h_n(c)$ so that, in the case of a discrete color distribution, each

bin contains an equal number of entries:

$$\sum_{c=b_i}^{b_{i+1}-1} h_n(c) = \sum_{c=b_j}^{b_{j+1}-1} h_n(c), \quad \forall i, j \in \{0, \dots, M-1\}, \quad (4)$$

where $b_0 = 0$ and b_M is one larger than the original number of quantization levels.

Later on, the derived bin boundaries were used in quantizing both sample and model distributions. Each color channel was quantized separately, and a number of distributions were created. An N -dimensional quantized histogram for each color space was created with 16 and 32 levels per channel. The histograms are denoted by $RGB16^3$, $RGB32^3$, $rg32^2$, $I_1I_2I_316^3$, $I_1I_2I_332^3$, $L^*a^*b^*16^3$, $L^*a^*b^*32^3$, $HSV16^3$, and $HSV32^3$. Furthermore, the marginal distributions with 256 histogram bins per channel were concatenated into one $256 \times N$ -bin histogram. The resulting histograms are denoted by $RGB256 \times 3$, $I_1I_2I_3256 \times 3$, and so on. As a third variation, the chrominance and luminance dimensions of the $I_1I_2I_3$, $L^*a^*b^*$, and HSV spaces were used separately, resulting in $I_2I_332^2$, I_1256 , $a^*b^*32^2$, L^*256 , $HS32^2$, and $V256$. These histograms were used in demonstrating the effect of illumination change on chrominance and luminance.

Summing up all the variations, a total of 19 different color histograms were used. Furthermore, the RGB histograms were used both with and without the comprehensive normalization algorithm of Finlayson et al. [22]. To see the contribution of first- and second-order statistics of gray values to the classification result, normalized luminance ($Y-Z256$) was also included for reference.

4.3. Color ratio histograms

It is well known that the indexing performance of color histograms drops drastically if illumination conditions cannot be kept constant. There are, however, color descriptors that are in some sense invariant to illumination color and intensity changes [23,31]. The color constant color indexing method of Funt and Finlayson [23] was chosen for its theoretically appealing yet simple construction. The authors also report promising results on retrieval problems.

The color constant color indexing method is implemented by taking the logarithm of an RGB image and by calculating a distribution of the derivative values of the resulting image. The authors propose three ways of approximating the derivative, from which the Laplacian of Gaussian was chosen because it was reasoned to work well on our high-resolution test images. The calculation of color ratios can be defined as follows:

$$d_c(x, y) = \nabla^2 G * \ln i_c(x, y), \quad (5)$$

where i_c stands for color channel c , $c \in \{1, 2, 3\}$, and G stands for a Gaussian filter. A three-dimensional distribution of derivative values is created out of $d_c(x, y)$ after quantization. Color ratios are actually neither color nor texture features, but something in between. In some sense, the color ratios measure local texture on each color channel by utilizing

the differences between neighboring pixels. The authors do however consider it a color indexing method. Consequently, this method was treated separately.

The color constant color indexing method requires a uniform quantization of the derivatives. Even though the authors describe a different method, the uniform quantization scheme described in Section 4.2 was used in achieving the very same goal. This time, however, the distributions were essentially continuous, and a slight modification of Eq. (4) is needed in defining bin boundaries:

$$\int_{b_i}^{b_{i+1}} D_c(t) dt = \int_{b_j}^{b_{j+1}} D_c(t) dt, \quad \forall i, j \in \{0, \dots, M-1\}, \quad (6)$$

where $D_c(t)$ represents the probability density function of derivative values on color channel c . b_0 and b_M can be set to $-\infty$ and ∞ , respectively. In the experiments, quantization into 16 and 32 levels was used. The corresponding operators are denoted by $CR16^3$ and $CR32^3$.

5. Texture features

5.1. Gray-scale texture

With gray-scale texture operators, only the luminance of the images was considered. For VisTex images, no knowledge of camera sensitivities or light sources was available, rendering any simple way of calculating the luminance useless. Therefore, the average of the RGB values was used: $Y = (R + G + B)/3$. For Outex textures, all necessary information was available. Following the derivation represented in Ref. [28], the equation for calculating luminance becomes $Y = 0.6534R + 0.3190G + 0.0277B$. The weighting factors are not equal to those in the middle row of the transformation matrix in Eq. (2) because the $L^*a^*b^*$ values must be calculated from the 1931 specification of XYZ , whereas the more recent 1964 version can be used for luminance.

In both cases, the luminance values were further scaled so that the mean and standard deviation of the luminance in each image were 127 and 20, respectively. This transformation (Z -normalization) removes the effect of mean luminance and overall contrast changes, but may fail in normalizing the images against illumination color or geometry variations. The normalization was applied to make sure texture operators really depend on texture information and not on the first-order statistics of gray values.

As gray-scale texture operators, the Gabor filter design of Manjunath and Ma [32] and the local binary pattern (LBP) operator [18,19] were selected because they have been demonstrated to perform very well on different texture discrimination problems. Furthermore, they represent two different approaches to texture analysis: filtering and pattern (micro-texton) statistics. Both Gabor filters and LBP also have a generalization for opponent colors. The use of the

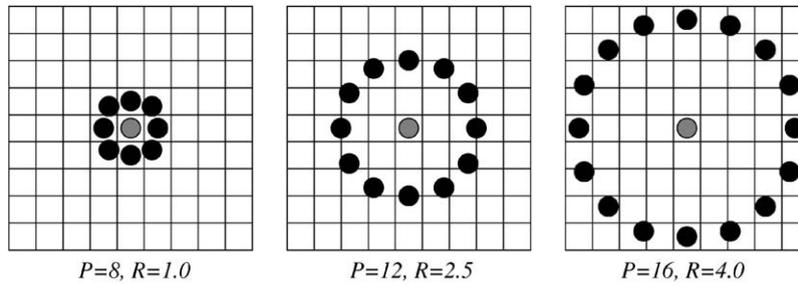


Fig. 3. Three circular neighborhoods.

term “opponent color” here follows the convention adopted by Jain and Healey [3]: all pairs of color channels are called “opponent colors”. It should not be confused with the opponent color mechanism of the human eye.

In addition to the four scales and six orientations used by Manjunath and Ma, a Gabor filter bank with three scales and four orientations was used. This choice was made to keep the filter bank designs for the gray scale and the opponent color Gabor methods as close to each other as possible. These operators are later denoted by $Gabor_{4,6}$ and $Gabor_{3,4}$. As a feature vector, the means and standard deviations of each filtered input image were used:

$$f = (\mu_{11}\sigma_{11}\mu_{21}\sigma_{21} \cdots \mu_{SO}\sigma_{SO}), \quad (7)$$

where each μ_{so} and σ_{so} represent the mean and the standard deviation of the input image convolved with the Gabor filter at scale s and orientation o . The capital letters S and O are used to denote the total numbers of scales and orientations, respectively.

As suggested by the authors, a city-block distance scaled with the standard deviations of the features was used as a dissimilarity measure for the gray-scale Gabor features:

$$d(s, m) = \sum_{i=0}^{N-1} \frac{|s_i - m_i|}{\sigma_i}, \quad (8)$$

where s and m are feature vectors, σ_i is the standard deviation of feature i , and N is the length of the feature vectors, in this case either 48 or 24.

The LBP is a gray scale invariant texture primitive statistic. For each pixel in an image, a binary code is produced by thresholding a circularly symmetric neighborhood with the value of the center pixel. A histogram is created to collect up the occurrences of different binary patterns. The basic version of the LBP operator considers only the eight-neighbors of a pixel, but the definition can be extended to include all circular neighborhoods with any number of pixels. By extending the neighborhood one can collect larger-scale texture primitives. However, the spatial support of the LBP operator is much smaller than that of the Gabor filters.

In Fig. 3, three circular neighborhoods are shown. The radius of the neighborhood is denoted by R , and the number of samples in it by P . Values for the samples that do not exactly

match pixels are calculated using bilinear interpolation. The variations of the LBP operator are denoted by $LBP_{P,R}$. In the experiments, circular neighborhoods with 8, 16 and 24 samples and neighborhood radii 1, 2, 3, and 5 were used. It is also possible to create multi-resolution LBP operators by concatenating the histograms produced by operators with different parameters. For example, $LBP_{(8,1+16,2)}$ stands for a multi-resolution LBP operator in which the first operator uses a neighborhood radius of one pixel and eight neighborhood samples. The second operator uses two as the neighborhood radius, and considers 16 neighborhood samples.

The $LBP_{24,R}$ operator produces histograms with over 16 million bins, which would be overwhelmingly sparse for any image of reasonable size. For that reason, a majority of the bins was discarded, and only “uniform” LBP codes were selected for further scrutiny [19]. This procedure was applied to the $LBP_{16,R}$ versions as well. The resulting operators are denoted by $LBP_{16,R}^{u2}$ and $LBP_{24,R}^{u2}$, and their histograms contain 242 and 554 entries, respectively. As a dissimilarity measure for LBP distributions, the suggested log-likelihood measure was utilized:

$$d(s, m) = - \sum_{i=0}^{N-1} s_i \log m_i, \quad (9)$$

where s and m are sample and model distributions, respectively, and N stands for the number of bins in the histograms.

5.2. Color texture

Most gray-scale texture operators can be straightforwardly applied on color images by combining the results of channelwise operations. Often, this simple technique works well even though the operators have been designed for gray-scale textures (See, for example, Ref. [11]). Since this method does not consider inter-channel differences, it is supposed to be robust against illumination color changes, compared to the opponent color texture methods that utilize cross-channel differences. Furthermore, as the goal is not to model biological signal processing, practically any color space can be used. We experimented with the $Gabor_{3,4}$ and $LBP_{8,1}$ operators in the RGB , $I_1I_2I_3$, $L^*a^*b^*$, and HSV spaces. The feature vectors obtained from different color

channels were simply concatenated into a color texture description. To account for illumination intensity changes, texture features were also calculated from Z -normalized color channels ($RGB-Z$).

The opponent color Gabor filtering technique of Jain and Healey [3] and an opponent color version of the LBP operator were used in opponent color texture analysis. There are some minor differences between the filter bank designs of Manjunath and Ma and that of Jain and Healey, but these were minimized by using the same number of filter scales and orientations for both. The main difference between the gray-scale method and its opponent color counterpart is that, in addition to unichrome features derived from filtered color channels, the latter also uses differences between filtered color channels to mimic the opponent color processing of the eye. If the filtered color channels of an image are denoted by $i_{so}^{(c)}(x, y)$, where c , s , and o stand for *channel*, *scale*, and *orientation*, respectively, the unichrome features can be written as

$$\mu_{so}^{(c)} = \sqrt{\sum_{x,y} i_{so}^{(c)}(x, y)^2}. \quad (10)$$

The total number of unichrome features becomes $4 \times 3 \times 3 = 36$.

Opponent color features are calculated from the differences of filtered color channels. The differences are calculated between all color channel pairs in the RGB space (3), for all orientations (4), and for each pair of scales i, j satisfying $|i - j| \leq 1$ (7). The total number of opponent color features thus becomes 84. Opponent color features are defined as

$$\mu_{s_1 s_2 o}^{(c_1 c_2)} = \sqrt{\sum_{x,y} \left(\frac{i_{s_1 o}^{(c_1)}(x, y)}{\mu_{s_1 o}^{c_1}} - \frac{i_{s_2 o}^{(c_2)}(x, y)}{\mu_{s_2 o}^{c_2}} \right)^2}. \quad (11)$$

The unichrome and opponent color features are concatenated into one 120-dimensional feature vector. In the experiments, a squared Euclidean distance scaled with feature variances was used as the dissimilarity measure, as suggested by the authors:

$$d(s, m) = \sum_{i=0}^{N-1} \frac{(s_i - m_i)^2}{\sigma_i^2}, \quad (12)$$

where σ_i^2 is the variance of feature i .

The LBP operator can be extended to color texture in a similar manner. Given an RGB image, the LBP operator is applied on each color channel separately. In addition, each pair of color channels is used in collecting opponent color patterns so that the center pixel for a neighborhood and the neighborhood itself are taken from different color channels. Since opposing pairs, like $R - G$ and $G - R$, are highly redundant, only three opponent color pairs need to be considered, in addition to the three intra-channel operators. The resulting six feature distributions are concatenated into a single distribution. In the experiments, the log-likelihood dissimilarity measure (Eq. (9)) was used.

6. Combining separate color and texture features

The fusion of separate color and texture descriptions can happen on three levels. They will be called feature vector, similarity measure, and classifier levels. On feature vector level, color and texture features can be simply concatenated into a single feature vector. This method, however, requires that feature values are properly scaled, and that feature vectors of different lengths are weighted. Furthermore, using a different similarity measure for color and texture measures would be a problem. For these reasons, fusion on feature vector level was not considered

6.1. Similarity measure level

On similarity measure level, a single classifier is used to classify samples with color and texture descriptions. A separate similarity measure is used for both feature vectors. The results of the similarity measurements between corresponding feature vectors are then scaled to make them commensurable, and combined to obtain a final similarity estimate. If it is possible to scale the similarity measures in the range $[0, 1]$, they can be treated as probabilities or as fuzzy decisions. For fuzzy decisions, a number of aggregation methods are available, see, for example, Ref. [33]. If the results of the similarity measures are treated as probabilities, the measurement of the similarity between a sample and a model is equivalent to assigning a probability to the class the model represents. Kuncheva et al. have carried out a comprehensive comparison of fusion methods for this kind of soft decisions [34]. A method based on maximum likelihood classification and certainty based fusion criterion was presented by Dubuisson-Jolly and Gupta [10]. Also, Horikawa has empirically evaluated combining methods in a color texture classification problem [35].

The similarity measures used in this study are not directly interpretable as probabilities. Even though one can derive estimations of minimum and maximum values for a similarity measure from training data, the distribution of the values may not be quite suitable for a probabilistic interpretation. Besides that, this type of scaling is very sensitive to outliers. Therefore, scaling the measures to the correct value range poses a serious problem. For this reason, the fuzzy and probabilistic interpretations were discarded. Instead, the values given by different similarity measures were normalized using their mean and variance:

$$d'_i = (d_i - \mu_i) / \sigma_i^2, \quad (13)$$

where d_i and d'_i represent the original and the normalized similarity between the i th feature vector, respectively. μ_i and σ_i^2 are the mean and variance of the similarity values over a representative training set. Despite the fact that the scaled similarities cannot be treated as probabilities, and that the methods represented in Ref. [34] actually handle class assignment, some of the methods of combining probabilities are still applicable for combining similarity measures.

Among the represented methods the applicable ones, namely *minimum*, *maximum* and *average* were selected. These were also used in Ref. [35]. In classification, average is equivalent to sum. Thus, the total similarity can be derived from any of the three following equations:

$$d_{\min} = \min(d'_1, d'_2), \quad (14)$$

$$d_{\max} = \max(d'_1, d'_2), \quad (15)$$

$$d_{\text{sum}} = d'_1 + d'_2, \quad (16)$$

where d'_1 and d'_2 are the scaled similarities between color and texture feature vectors, respectively. Using dissimilarities instead of similarities causes no problems, as one can just find the minimum dissimilarity instead of the maximum similarity.

6.2. Classifier level

In a case where complementary color and texture information is used, it is unlikely that both color and texture make exactly the same mistakes. Therefore, a method of combining these two that can take the strengths and weaknesses of each feature type into account is needed. With scaled similarity measures the danger of incorrect scaling is always present. The problems with incompatible similarity measurements can be reduced by combining representations on classifier level. In this case, a separate classifier is used for both features. The classifiers may now use incompatible similarity measures, and one does not need to worry about scaling. The final classification result is derived by combining classification results—or class rankings, to be exact. Here, the method of Ho et al. [36] was used in combining the classification results with color and texture features. The Borda count was used as a decision criterion because, in addition to the average (or sum, see Eq. (16)), it was reported to give the best results in Ref. [35]. The Borda count for a class is defined as the sum of the number of classes that are ranked below it by each classifier. Therefore, it can be considered a generalization of the majority vote.

7. Results and discussion

The classification (retrieval) principle in both [3,32] was the nearest neighbor rule, with a custom dissimilarity measure. With the LBP, the k -NN classifier has been successfully utilized. Using the NN classifier was considered fair play, and it was consequently used in all experiments.

The color histograms, the color constant color indexing method, the gray-scale texture features, and the color texture features were all used in classifying the three color texture sample sets. The classification results are summarized in Tables 1–5. For the Outex 14 test suite, results are reported both with and without the comprehensive normalization, where applicable.

Table 1
Results for color histograms

Histogram type	VisTex	Outex 13	Outex 14
$RGB16^3$	99.1	94.6	9.3/34.3
$RGB32^3$	99.5	94.3	9.3/32.1
$RGB256 \times 3$	95.8	91.5	19.0/28.9
$I_1I_2I_316^3$	99.8	94.7	11.8
$I_1I_2I_332^3$	100	94.1	12.0
$I_1I_2I_3256 \times 3$	98.1	90.7	21.8
$L^*a^*b^*16^3$	99.8	95.1	8.8
$L^*a^*b^*32^3$	99.8	94.1	8.1
$L^*a^*b^*256 \times 3$	98.4	93.7	12.2
$HSV16^3$	99.8	95.4	9.7
$HSV32^3$	99.8	94.6	10.7
$HSV256 \times 3$	97.5	94.7	14.0
$rg32^2$	99.1	93.4	6.0
$I_2I_332^2$	97.7	87.9	6.9
$a^*b^*32^2$	98.8	91.9	4.3
$HS32^2$	99.3	93.3	5.8
I_1256	87.3	91.3	36.6
$L256$	85.4	80.9	34.6
$V256$	89.1	75.6	39.9
$Y-Z256$	47.7	30.1	11.0

Table 2
Results for color ratio histograms

Feature	VisTex	Outex 13	Outex 14
$CR16^3$	98.4	84.0	42.7
$CR32^3$	98.8	74.9	39.2

Table 3
Results for gray-scale texture

Feature	VisTex	Outex 13	Outex 14
$Gabor_{4,6}$	89.6	77.1	66.0
$Gabor_{3,4}$	89.8	78.4	64.2
$LBP_{8,1}$	97.7	81.0	59.3
$LBP_{16,2}^{u2}$	96.8	79.7	69.3
$LBP_{(8,1+u2_{16,2})}$	98.6	83.4	66.5
$LBP_{(8,1+u2_{16,3}+u2_{24,5})}$	98.6	82.4	69.5

The VisTex textures are almost faultlessly classified by most of the features tested, and $I_1I_2I_332^3$ and $LBP_{16,2}^{u2}$ yield a 100% score. Thus, these textures give not much useful information for comparison purposes. It suffices to summarize that with such a diverse set of natural textures, most methods work. Even a two-dimensional distribution of chromaticities (e.g. $HS32^2$ or $rg32^2$) is enough. Compared to the simple color indexing methods, no performance increase is achieved with any of the gray-scale or color texture methods. The differences between different methods can be better analyzed with the Outex textures, the classification of which reveals a number of interesting phenomena. The next

Table 4
Results for color texture

Feature	VisTex	Outex 13	Outex 14
Gabor _{3,4} <i>RGB-Z</i>	92.6	84.0	62.4
Gabor _{3,4} <i>RGB</i>	96.1	86.9	35.4/43.5
Gabor _{3,4} $I_1 I_2 I_3$	98.6	83.8	44.0
Gabor _{3,4} $L^* a^* b^*$	99.1	86.8	55.2
Gabor _{3,4} <i>HSV</i>	97.5	86.5	25.4
LBP _{8,1} <i>RGB-Z</i>	98.4	86.8	54.0
LBP _{8,1} <i>RGB</i>	97.9	87.8	53.9/43.8
LBP _{8,1} $I_1 I_2 I_3$	98.8	84.7	57.1
LBP _{8,1} $L^* a^* b^*$	99.3	82.9	60.1
LBP _{8,1} <i>HSV</i>	98.8	85.9	44.9
LBP _{16,2} ^{u2} $L^* a^* b^*$	100	85.3	63.2
LBP _(8,1+^{u2}_{16,3}+^{u2}_{24,5}) $L^* a^* b^*$	99.5	87.8	67.8

Table 5
Results for opponent color texture

Feature	VisTex	Outex 13	Outex 14
Gabor	97.9	81.2	53.3/47.4
LBP _{8,1}	98.8	92.5	13.5/45.1
LBP _{16,2} ^{u2}	99.5	92.4	15.4/47.1

three sections discuss the findings. The best results for each experiment are summarized in Table 7.

7.1. Static illumination

With the Outex_TC_00013 test set (static illumination), even the 256-level intensity histogram ($I_1 256$) gives a 91.3% score, showing that the first- and second-order statistics of gray values carry lots of useful information if they can be relied on. If the images are normalized with respect to these variations, the descriptive power of the gray level histogram drops drastically, as shown by the 30.1% score of Z -normalized luminance ($Y-Z256$). On the other hand, good accuracy can be achieved even when the intensity information is totally discarded: $HS32^2$ or $rg32^2$ both provide a better accuracy than any of the gray-scale or color texture methods.

With gray-scale texture, the best score (83.4%) was achieved by the multi-resolution LBP_(8,1+^{u2}_{16,2}). Channelwise color texture descriptors perform somewhat better, but with a tripled computational overhead and a three times longer feature vector. It should be noted that the color channels also carry intensity information. Therefore, the channelwise texture features contain practically the same information as the gray-scale features, supplemented with some additional information from the chromaticities. The best results with channelwise Gabor and LBP features are obtained in the *RGB* color space, while the color spaces with intensity and chromaticity separated work slightly

worse. This is important because it shows that the texture information obtained from the chromaticity channels is not very useful. In the *RGB* space, intensity is present on each color channel.

Neither Gabor nor LBP features are significantly affected by the Z -normalization of color channels. Unexpectedly, the opponent color Gabor features are weaker than the channelwise features. With the LBP, the opponent color version performs clearly better, coming close to the performance of the color histograms. Anyhow, the use of complex color texture methods is not easily justified because no significant performance increase can be achieved compared even to the gray-scale histogram, not to speak about color histograms.

In the Outex_TC_00013 experiment, the difference between color and texture descriptors can be mainly attributed to the last 11 texture classes (mixtures of barley and rice), and partially to the eight different sandpapers. While color descriptors are able to distinguish between these classes quite well, the differences in their textural structures seems not to be prominent enough for the texture measures.

7.2. Varying illumination

Gray-scale texture measures show their teeth with the Outex_TC_00014 experiment in which the light source changes. The changes in the color of the illumination immediately render all color histograms practically useless. With the *RGB* color space, the comprehensive normalization however helps a bit. The color ratios also survive somewhat better than color histograms, but are no match for the texture measures. It is the chromaticity that causes most of the erroneous classifications, which is shown with the color histograms with intensity and chromaticity separated. For example, the score of $HS32^2$ is 5.8%, whereas the $V256$ histogram achieves a 39.9% accuracy.

The best score (69.5%) in this experiment was achieved by the multi-resolution LBP_(8,1+^{u2}_{16,3}+^{u2}_{24,5}). The performance increase compared to the LBP_{8,1} is over 10 percentage units, showing that a large neighborhood is needed to cope with the changes caused by varying illumination. The gray-scale Gabor features stand the illumination change quite well, most probably for the same reason.

The color texture measures perform clearly worse than their gray-scale counterparts. Especially the opponent color texture descriptors suffer from the illumination change. Again, the channelwise Gabor features outperform the opponent color version, but only if the Z -normalization is applied. With the opponent color LBP, the comprehensive normalization helps a lot, but the result is still weaker than that of the channelwise features. The color spaces with intensity and chromaticity separated now work better with the channelwise features, because the distorted chromaticity information is in a sense limited to just two channels. An exception is the *HSV* space, which seems to be very instable in varying illumination.

Table 6
Results for combined color and texture

Combination method	VisTex	Outex 13	Outex 14
<i>RGB16³</i> and Gabor _{3,4}			
Minimum	89.8	78.4	64.2/64.2
Maximum	99.1	94.6	14.0/40.2
Sum	91.0	80.6	64.2/64.3
Borda count	98.1	91.8	33.9/59.1
<i>I₁I₂I₃256 × 3</i> and Gabor _{3,4}			
Minimum	89.8	78.4	64.2
Maximum	99.1	91.2	28.2
Sum	90.5	78.8	64.3
Borda count	97.9	90.9	55.6
<i>RGB16³</i> and LBP _{16,2} ^{u₂}			
Minimum	97.5	88.8	49.2/58.4
Maximum	98.4	85.4	31.7/47.1
Sum	97.2	81.3	69.4/69.4
Borda count	99.8	91.2	35.9/59.7
<i>I₁I₂I₃256 × 3</i> and LBP _{16,2} ^{u₂}			
Minimum	97.5	85.7	53.7
Maximum	97.7	85.1	39.7
Sum	97.0	80.1	69.3
Borda count	99.3	90.7	56.2

7.3. Combined color and texture

So far, it seems that neither color nor texture features are successful in both static and varying illumination conditions. Unfortunately, none of the methods of combining color and texture on a higher level was universally applicable either. In the first two experiments where both color and texture give reasonable results, the Borda count and the maximum dissimilarity method work best, as shown in Tables 6 and 7. In the third experiment, however, the sum of dissimilarities seems to be the most robust method. Therefore, if these methods are used, they must be selected based on the application at hand.

Although all combinations of color and texture features were not tested, it seems that the utility obtained in combining color and texture features is at most minimal. In most cases, either of the combined features would have produced a better result by itself.

8. Conclusions

Many previous studies have compared gray-scale texture features with their color counterparts, concluding that adding color information to texture measures increases accuracy. According to our results, this is indeed the case, but only in static illumination conditions. It has been mostly forgotten that much of the difference can be explained by the multiple amount of information. One can pick some numbers to

Table 7
Best results for each problem

	Result	Best method
<i>VisTex</i>		
Color histogram	100	<i>I₁I₂I₃32³</i>
Color ratios	98.8	<i>CR32³</i>
Gray-scale texture	98.6	LBP _(8,1+u₂,2)
Color texture	100	LBP _{16,2} <i>L*a*b*</i>
Opponent color texture	99.5	LBP _{16,2} ^{u₂}
Color + texture	99.8	<i>RGB16³</i> + LBP _{16,2} ^{u₂} Borda count
<i>Outex 13</i>		
Color histogram	95.4	<i>HSV16³</i>
Color ratios	84.0	<i>CR16³</i>
Gray-scale texture	83.4	LBP _(8,1+u₂,2)
Color texture	87.8	LBP _{8,1} <i>RGB</i>
Opponent color texture	92.5	LBP _{8,1}
Color + texture	94.6	<i>RGB16³</i> + Gabor _{3,4} Max dissimilarity
<i>Outex 14</i>		
Color histogram	39.9	<i>V256</i>
Color ratios	42.7	<i>CR16³</i>
Gray-scale texture	69.5	LBP _(8,1+u₂,3+u₂,5)
Color texture	67.8	LBP _(8,1+u₂,3+u₂,5) <i>L*a*b*</i>
Opponent color texture	53.3	Gabor
Color + texture	69.4	<i>RGB16³</i> + LBP _{16,2} ^{u₂} Dissimilarity sum

demonstrate this with the results presented above. For example, LBP_(8,1+u₂,2) scores 82.8% on average with a feature distribution of 599 bins (see Table 3). The same mean score is achieved by LBP_{16,2}^{u₂} *L*a*b**, but with 729 bins (Table 4). It is tempting to compare this result with that of LBP_{16,2}^{u₂} (81.9%) to demonstrate that adding color information increases accuracy. The increase is, however, obtained with a three times longer feature vector.

Overall, comparing gray-scale texture to color texture is a rather limited approach. The results here show that the cases in which color texture is better than gray-scale texture could be better resolved by very simple color indexing methods. In static illumination conditions one can rely on color measurements, which is a necessary precondition to the functioning of color texture and color histograms. In varying illumination, gray-scale texture works clearly better than either color, channelwise color texture or opponent color texture.

The evidence presented in this study suggests that using color and texture in parallel is not the most powerful way of utilizing this complementary information. All joint color texture descriptors and all methods of combining color and texture on a higher level are outperformed by either color or gray-scale texture alone. The utility attained with a joint

description is at most minimal. However, even a minimal performance increase may be important in some applications [37].

There are applications where both color and texture need to be used for maximum performance [38,39]. Typically, they have been used in parallel. It should however be considered whether a sequential use could better exploit the capabilities of each feature space. Either can be used in drawing a preliminary decision, which is then refined with the other.

Acknowledgements

The financial support provided by Academy of Finland, the national Graduate School in Electronics, Telecommunications and Automation, and Nokia Foundation is gratefully acknowledged.

References

- [1] A. Rosenfeld, C. Ye-Wang, A. Wu, Multispectral texture, *IEEE Trans. Systems, Man, Cybern.* 12 (1) (1982) 79–84.
- [2] D. Panjwani, G. Healey, Unsupervised segmentation of textured color images, *IEEE Trans. Pattern Anal. Mach. Intell.* 17 (10) (1995) 939–954.
- [3] A. Jain, G. Healey, A multiscale representation including opponent color features for texture recognition, *IEEE Trans. Image Process.* 7 (1) (1998) 124–128.
- [4] T. Caelli, D. Reye, On the classification of image regions by colour, texture and shape, *Pattern Recognition* 26 (4) (1993) 461–470.
- [5] G. Paschos, Perceptually uniform color spaces for color texture analysis: an empirical evaluation, *IEEE Trans. Image Process.* 10 (6) (2001) 923–937.
- [6] M. Mirmehdi, M. Petrou, Segmentation of color textures, *IEEE Trans. Pattern Anal. Mach. Intell.* 22 (2) (2000) 142–159.
- [7] G. Paschos, M. Petrou, Histogram ratio features for color texture classification, *Pattern Recogn. Lett.* 24 (2003) 309–314.
- [8] T. Gevers, A. Smeulders, Color constant ratio gradients for image segmentation and similarity of texture objects, in: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, Kauai, Hawaii, 2001, pp. 18–25.
- [9] T. Tan, J. Kittler, Colour texture classification using features from colour histogram, in: *Proceedings of the 8th Scandinavian Conference on Image Analysis*, Vol. 2, Tromsø, Norway, 1993, pp. 807–813.
- [10] M.-P. Dubuisson-Jolly, A. Gupta, Color and texture fusion: application to aerial image segmentation and GIS updating, *Image Vision Comput.* 18 (2000) 823–832.
- [11] A. Drimbarean, P. Whelan, Experiments in colour texture analysis, *Pattern Recogn. Lett.* 22 (2001) 1161–1167.
- [12] T. Pappathomas, R. Kashi, A. Gorea, A human vision based computational model for chromatic texture segregation, *IEEE Trans. Systems, Man, Cybern. B* 27 (3) (1997) 428–440.
- [13] E. DeYoe, D. van Essen, Concurrent processing streams in monkey visual cortex, *Trends. Neurosci.* 11 (1996) 219–226.
- [14] R. DeValois, K. DeValois, *Spatial Vision*, Oxford University Press, Oxford, 1990.
- [15] B. Poirson, B. Wandell, Pattern-color separable pathways predict sensitivity to simple colored patterns, *Vision Res.* 36 (4) (1996) 515–526.
- [16] A. Mojsilovic, J. Kovacevic, D. Kall, R. Safranek, S. Ganapathy, Matching and retrieval based on the vocabulary and grammar of color patterns, *IEEE Trans. Image Process.* 9 (1) (2000) 38–54.
- [17] B. Wandell, *Foundations of Vision*, Sinauer Associates, Sunderland, MA, 1995.
- [18] T. Ojala, M. Pietikäinen, D. Harwood, A comparative study of texture measures with classification based on feature distributions, *Pattern Recognition* 29 (1996) 51–59.
- [19] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray scale and rotation invariant texture analysis with local binary patterns, *IEEE Trans. Pattern Anal. Mach. Intell.* 24 (7) (2002) 971–987.
- [20] MIT Media Lab, Vision texture—VisTex database, <http://www-white.media.mit.edu/vismod/imagery/VisionTexture/vistex.html> (1995).
- [21] T. Ojala, T. Mäenpää, M. Pietikäinen, J. Viertola, J. Kyllönen, S. Huovinen, Outex—new framework for empirical evaluation of texture analysis algorithms, in: *Proceedings of the 16th International Conference on Pattern Recognition*, Vol. 1, Québec, Canada, 2002, pp. 701–706, <http://www.outex.oulu.fi/>.
- [22] G. Finlayson, B. Schiele, J. Crowley, Comprehensive colour image normalization, in: *Proceedings of the 5th European Conference on Computer Vision*, Vol. 1, Freiburg, Germany, 1998, pp. 475–490.
- [23] B. Funt, G. Finlayson, Color constant color indexing, *IEEE Trans. Pattern Anal. Mach. Intell.* 17 (5) (1995) 522–529.
- [24] C. Vertan, N. Boujemaa, Color texture classification by normalized color space representation, in: *Proceedings of the 15th International Conference on Pattern Recognition*, Vol. 3, Barcelona, Spain, 2000, pp. 580–583.
- [25] Y. Ohta, T. Kanade, T. Sakai, Color information for region segmentation, *Comput. Graphics Image Process.* 13 (1980) 222–241.
- [26] International Telecommunications Union, Parameter values for the HDTV standards for production and international programme exchange, ITU-R Recommendation BT.709-4 (March 2000).
- [27] C. Poynton, *A Technical Introduction to Digital Video*, Wiley, Inc., New York, 1996.
- [28] R. Hunt, *Measuring Colour*, Ellis Horwood Ltd., Chichester, 1987.
- [29] M. Swain, D. Ballard, Color indexing, *Int. J. Comput. Vision* 7 (1991) 11–32.
- [30] M. Pietikäinen, S. Nieminen, E. Marszalec, T. Ojala, Accurate color discrimination with classification based on feature distributions, in: *Proceedings of the 13th International Conference on Pattern Recognition*, Vol. 3, Vienna, Austria, 1996, pp. 833–838.
- [31] T. Gevers, A. Smeulders, Color-based object recognition, *Pattern Recognition* 32 (1999) 453–464.
- [32] B. Manjunath, W. Ma, Texture features for browsing and retrieval of image data, *IEEE Trans. Pattern Anal. Mach. Intell.* 18 (8) (1996) 837–842.

- [33] H.-M. Hsu, C.-T. Chen, Aggregation of fuzzy opinions under group decision making, *Fuzzy Sets Systems* 79 (1996) 279–285.
- [34] L. Kuncheva, J. Bezdek, R. Duin, Decision templates for multiple classifier fusion: an experimental comparison, *Pattern Recognition* 34 (2001) 299–314.
- [35] Y. Horikawa, Comparison of combining methods in invariant color texture classification with cross-bispectral features, in: *The First European Conference on Colour in Graphics, Image and Vision*, Poitiers, France, 2002, pp. 424–428.
- [36] T. Ho, J. Hull, S. Srihari, Decision combination in multiple classifier systems, *IEEE Trans. Pattern Anal. Mach. Intell.* 16 (1) (1994) 66–75.
- [37] T. Mäenpää, J. Viertola, M. Pietikäinen, Optimizing color and texture features for real-time visual inspection, *Pattern Anal. Appl.* 6 (3) (2003) 169–175.
- [38] C. Boukouvalas, F.D. Natale, G.D. Toni, J. Kittler, R. Marik, M. Mirmehdi, M. Petrou, P.L. Roy, R. Salgari, G. Vernazza, Automatic system for surface inspection and sorting of tiles, *J. Mater. Process. Technol.* 82 (1–3) (1998) 179–188.
- [39] R. Bolle, J. Connell, N. Haas, R. Mohan, G. Taubin, Veggievision: a produce recognition system, in: *Proceedings of the 3rd IEEE Workshop on Applications of Computer Vision*, Sarasoto, FL, 1996, pp. 244–251.

About the Author—TOPI MÄENPÄÄ received the M.Sc. (diploma engineer) degree in electrical engineering from the university of Oulu, Finland, in 1999 with honors. In 2003, he received the Doctor of Technology degree in information engineering, with honors. He is currently working as the R&D manager of Intopii Ltd. and as a researcher in the Machine Vision Group of University of Oulu. His research interests include color and texture analysis, visual inspection with efficient texture methods, biometric identification, and control architectures of mobile robots.

About the Author—MATTI PIETIKÄINEN received his Doctor of Technology degree in electrical engineering from the University of Oulu, Finland, in 1982. In 1981, he established the Machine Vision Group at the University of Oulu. The research results of his group have been widely exploited in industry. Currently, he is a professor of information technology, the scientific director of Infotech Oulu Research Center, and the leader of the Machine Vision Group at the University of Oulu. From 1980 to 1981 and from 1984 to 1985, he visited the Computer Vision Laboratory at the University of Maryland, USA. His research interests are in machine vision and image analysis. His current research focuses on texture analysis, color and face image analysis, learning in machine vision, and machine vision for sensing and understanding human actions. He has authored over 145 papers in international journals, books, and conference proceedings, and nearly 100 other publications or reports. He is Associate Editor of *IEEE Transactions on Pattern Analysis and Machine Intelligence* and *Pattern Recognition* journals. He has edited two books and has served as a reviewer for numerous journals and conferences. He was the president of the Pattern Recognition Society of Finland from 1989 to 1992. Since 1989, he has served as a member of the governing board of the International Association for Pattern Recognition (IAPR) and became one of the founding fellows of the IAPR in 1994. He has also served on committees of several international conferences. He is a senior member of the IEEE.