

# WLD: A Robust Local Image Descriptor

Jie Chen, *Member, IEEE*, Shiguang Shan, *Member, IEEE*, Chu He, Guoying Zhao, Matti Pietikäinen, *Senior Member, IEEE*, Xilin Chen, *Senior Member, IEEE*, and Wen Gao, *Fellow, IEEE*

**Abstract**—Inspired by Weber’s Law, this paper proposes a simple, yet very powerful and robust local descriptor, called the Weber Local Descriptor (WLD). It is based on the fact that human perception of a pattern depends not only on the change of a stimulus (such as sound, lighting) but also on the original intensity of the stimulus. Specifically, WLD consists of two components: differential excitation and orientation. The differential excitation component is a function of the ratio between two terms: One is the relative intensity differences of a current pixel against its neighbors, the other is the intensity of the current pixel. The orientation component is the gradient orientation of the current pixel. For a given image, we use the two components to construct a concatenated WLD histogram. Experimental results on the Brodatz and KTH-TIPS2-a texture databases show that WLD impressively outperforms the other widely used descriptors (e.g., Gabor and SIFT). In addition, experimental results on human face detection also show a promising performance comparable to the best known results on the MIT+CMU frontal face test set, the AR face data set, and the CMU profile test set.

**Index Terms**—Pattern recognition, Weber law, local descriptor, texture, face detection.

## 1 INTRODUCTION

RECENTLY, there has been much interest in object and view matching using local invariant features [27], classification of textured regions using microtextures [34], and in face detection using local features [47]. There are several studies to evaluate the performance of these methods, such as [30], [31], [33], [38]. These methods can be divided into two classes: One is a sparse descriptor which first detects the interest points in a given image and then samples a local patch and describes its invariant features [30], [31]; the other is a dense descriptor which extracts local features pixel by pixel over the input image [33], [38].

For the sparse descriptors, a typical one is the scale-invariant feature transform (SIFT), introduced by Lowe [27]. It performs best in the context of matching and recognition due to its invariance to scaling and rotations [31]. Several attempts to improve the SIFT descriptor have been reported in the literature. Ke and Sukthankar developed the PCA-SIFT descriptor, which represents local appearance by principal components of the normalized gradient field [23]. Mikolajczyk and Schmid modified the SIFT descriptor by changing the gradient location orientation grid, as well

as the quantization parameters of the histograms [31]. Dalal and Triggs proposed a “histogram of oriented gradients” (HOG) [12]. Lazebnik et al. proposed a rotation invariant descriptor called the Rotation Invariant Feature Transform (RIFT) [24]. Bay et al. proposed an efficient implementation of SIFT by applying the integral image to compute image derivatives, and quantifying the gradient orientations in a small number of histogram bins [4]. Winder and Brown learned an optimal parameter setting on a large training set to maximize the matching performance [48]. Mikolajczyk and Matas developed the optimal linear projection to improve the matching quality and speed of SIFT [32]. Likewise, in order to improve the efficiency of the local descriptor, Tola et al. replaced the weighted sum rule used in SIFT by sum of convolutions [44]. In addition, Cheng et al. proposed the use of multiple support regions of different sizes surrounding a point of interest [11].

Among the most popular dense descriptors are the Gabor wavelet [28] and local binary pattern (LBP) [34]. The Gabor representation has been shown to be optimal in the sense of minimizing the joint two-dimensional uncertainty in space and frequency [28]. The Gabor filters can be considered as orientation and scale tunable edge and line (bar) detectors, and the statistics of these microfeatures in a given region are often used to characterize the underlying texture information. The Gabor wavelet has been widely used in image analysis applications, including texture classification and segmentation, image registration, motion tracking [28], and face recognition [53]. Another important dense local descriptor is LBP, which has gained increasing attention due to its simplicity and excellent performance in various texture and face image analysis tasks [34]. Many variants of LBP have been recently proposed and have achieved considerable success in various tasks. Ahonen et al. exploited the LBP for face recognition [2]. Rodriguez and Marcel proposed adapted LBP histograms for face authentication [40]. Tan and Triggs changed the thresholding means for face recognition under difficult lighting conditions [43]. Zhao and Pietikäinen proposed the local binary pattern on three orthogonal planes,

- J. Chen, G. Zhao, and M. Pietikäinen are with the Machine Vision Group, Department of Electrical and Information Engineering, University of Oulu, PO Box 4500, FI-90014, Finland. E-mail: {jiecchen, gyzhao, mkp}@ee.oulu.fi.
- S. Shan and X. Chen are with the Key Lab of Intelligent Information Processing of CAS, Institute of Computing Technology, Chinese Academy of Sciences, No. 6 Kexueyuan NanLu, Haidian District, Beijing 100190, China. E-mail: {sgshan, xlchen}@ict.ac.cn.
- C. He is with the School of Electronic Information, Wuhan University, Wuhan 430079, China. E-mail: hc@eis.whu.edu.cn.
- W. Gao is with the Key Lab of Machine Perception, Peking University, No. 5, Yiheyuan Road, Beijing 100871. P.R. China. E-mail: wgao@pku.edu.cn.

Manuscript received 15 Sept. 2008; revised 9 Dec. 2008; accepted 11 July 2009; published online 6 Aug. 2009.

Recommended for acceptance by S. Belongie.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2008-09-0620.

Digital Object Identifier no. 10.1109/TPAMI.2009.155.

and used it for dynamic texture recognition [52]. Zhang et al. proposed the local Gabor binary pattern for face representation and recognition [53]. In addition, some researchers use these descriptors in a crossway. For example, Fei-Fei and Perona used SIFT in a dense sampling way [14] and Heikkilä et al. exploit LBP in a sparse way [18].

In this paper, we propose a simple, yet very powerful and robust local descriptor. This descriptor consists of two components: differential excitation and orientation. It is inspired by Weber's Law, which is a psychological law [21]. It states that the change of a stimulus (such as sound, lighting) that will be just noticeable is a constant ratio of the original stimulus. When the change is smaller than this constant ratio of the original stimulus, a human being would recognize it as background noise rather than a valid signal. Motivated by this point, for a given pixel, the differential excitation component of the proposed Weber Local Descriptor (WLD) is computed based on the ratio between the two terms: One is the relative intensity differences of a current pixel against its neighbors (e.g.,  $3 \times 3$  square region); the other is the intensity of the current pixel. With the differential excitation component, we attempt to extract the local salient patterns in the input image. In addition, we also compute the gradient orientation of the current pixel. That is, for each pixel of the input image, we compute two components of the WLD feature (i.e., differential excitation and gradient orientation). By combining the WLD feature per pixel, we represent an input image (or image region) with a histogram, which we call a WLD histogram hereinafter. In our case, the WLD feature is computed pixelwise. Thus, WLD is a dense descriptor.

The proposed WLD descriptor employs the advantages of SIFT in computing the histogram using the gradient and its orientation, and those of LBP in computational efficiency and smaller support regions. But WLD distinctly differs from SIFT and LBP. As mentioned above, the SIFT descriptor is a 3D histogram of gradient locations and orientations in which two dimensions correspond to image spatial coordinates and the additional dimension to the image gradient orientation. As a sparse descriptor, SIFT computes only for the regions of interest (located around detected interest points) that have usually already been normalized with respect to scale and rotation. Texture classification with SIFT is performed using information in these sparsely located interest regions, as in [13]. WLD, on the contrary, is a dense descriptor computed for every pixel and depends on both the local intensity variation and the magnitude of the center pixel's intensity. Texture classification with WLD is carried out using 2D WLD histograms. Since WLD is computed around a relatively small square region (e.g.,  $3 \times 3$ ), while SIFT is computed around a relatively large region (e.g.,  $16 \times 16$ ) [27], [11], the description granularity of WLD is much smaller than that of SIFT. That is to say, WLD is computed in a finer granularity than SIFT. The smaller size of the support regions for WLD makes it capture more local salient patterns. Furthermore, WLD can be easily extended to extract the multigranularity features by multiscale analysis techniques, as presented in Section 2.4.

With regard to the LBP descriptor, it represents an input image by building statistics on the local micropattern variations. These local patterns might correspond to bright/dark spots, edges and flat areas, etc. In contrast, WLD first computes the salient micropatterns (i.e., differential

excitation), and then builds statistics on these salient patterns along with the gradient orientation of the current point.

Several researchers have used Weber's Law in computer vision, but not, as in this study, as a descriptor. Bruni and Vitulano used this law for scratch detection on digital film materials [8]. Phiasai et al. employed a Weber ratio to control the strength of a watermark [37].

This paper is an extension of our previous work [10]. In this current paper, we further extend the original WLD to a multiscale version in order to extract multigranularity features. We also provide a more in-depth theoretical analysis and more extensive evaluations on WLD. The rest of this paper is organized as follows: In Section 2, we present the details of the proposed local descriptor WLD and compare it with other existing methods. In Sections 3 and 4, we carry out the experiments dealing with the applications of WLD in texture classification and face detection. In Section 5, we discuss some issues about the proposed descriptor. Section 6 concludes the paper.

## 2 WLD FOR IMAGE REPRESENTATION

In this section, we review Weber's Law and then detail the proposed WLD. Subsequently, we develop its multiscale analysis. In addition, we compare WLD with some existing descriptors.

### 2.1 Weber's Law

Ernst Weber, an experimental psychologist in the 19th century, observed that the ratio of the increment threshold to the background intensity is a constant [21]. This relationship, known since as *Weber's Law*, can be expressed as:

$$\frac{\Delta I}{I} = k, \quad (1)$$

where  $\Delta I$  represents the increment threshold (just noticeable difference for discrimination),  $I$  represents the initial stimulus intensity, and  $k$  signifies that the proportion on the left side of the equation remains constant despite variations in the  $I$  term. The fraction  $\Delta I/I$  is known as the *Weber fraction*.

Weber's Law, more simply stated, says that the size of a *just noticeable difference* (i.e.,  $\Delta I$ ) is a constant proportion of the original stimulus value. So, for example, in a noisy environment one must shout to be heard while a whisper works in a quiet room.

### 2.2 WLD

In this part, we describe the two components of WLD: differential excitation ( $\xi$ ) and orientation ( $\theta$ ). After that, we present how to compute a WLD histogram for an input image (or image region).

#### 2.2.1 Differential Excitation

We use the intensity differences between its neighbors and a current pixel as the changes of the current pixel. By this means, we hope to find the salient variations within an image to simulate the pattern perception of human beings. Specifically, a differential excitation  $\xi(x_c)$  of a current pixel  $x_c$  is computed as illustrated in Fig. 1. We first calculate the differences between its neighbors and the center point using the filter  $f_{00}$ :

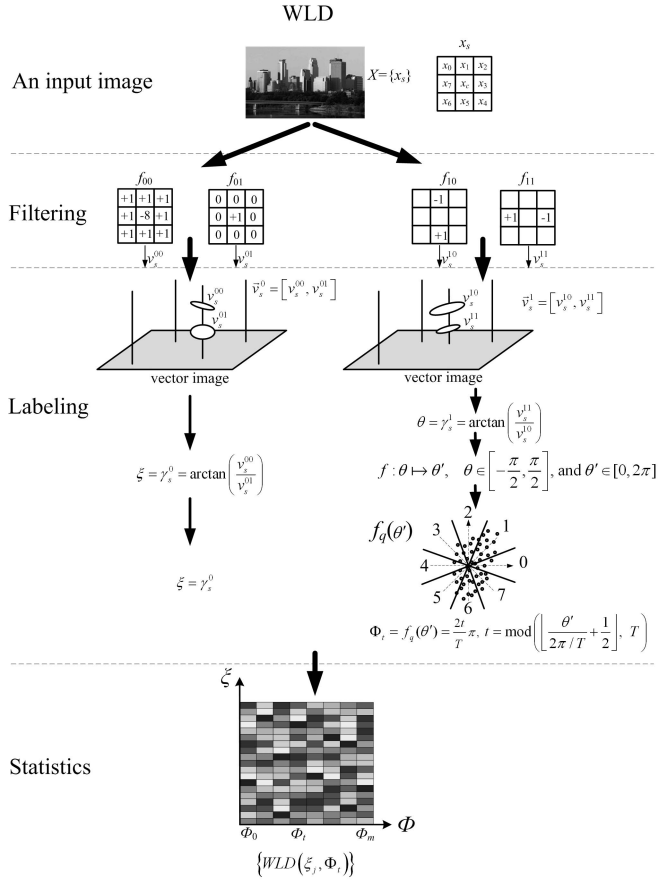


Fig. 1. Illustration of the computation of the WLD descriptor.

$$\nu_s^{00} = \sum_{i=0}^{p-1} (\Delta x_i) = \sum_{i=0}^{p-1} (x_i - x_c), \quad (2)$$

where  $x_i$  ( $i = 0, 1, \dots, p-1$ ) denotes the  $i$ th neighbors of  $x_c$  and  $p$  is the number of neighbors. Following hints in Weber's Law, we then compute the ratio of the differences to the intensity of the current point by combining the outputs of the two filters  $f_{00}$  and  $f_{01}$  (whose output  $\nu_s^{01}$  is the original image in fact):

$$G_{ratio}(x_c) = \nu_s^{00} / \nu_s^{01}. \quad (3)$$

We then employ the arctangent function on  $G_{ratio}(\cdot)$ :

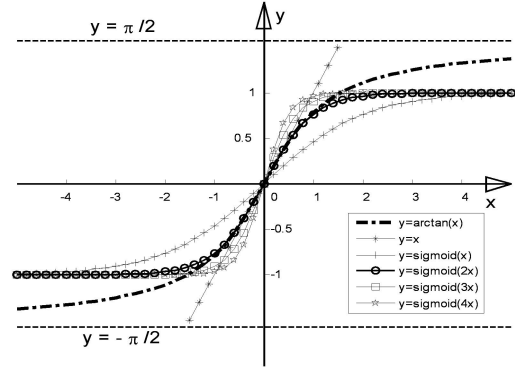
$$G_{arctan}[G_{ratio}(x_c)] = \arctan[G_{ratio}(x_c)]. \quad (4)$$

Combining (2), (3), and (4), we have:

$$\begin{aligned} G_{arctan}[G_{ratio}(x_c)] &= \gamma_s^0 = \arctan\left[\frac{\nu_s^{00}}{\nu_s^{01}}\right] \\ &= \arctan\left[\sum_{i=0}^{p-1} \left(\frac{x_i - x_c}{x_c}\right)\right]. \end{aligned} \quad (5)$$

So, the differential excitation of the current pixel  $\xi(x_c)$  is computed as:

$$\xi(x_c) = \arctan\left[\frac{\nu_s^{00}}{\nu_s^{01}}\right] = \arctan\left[\sum_{i=0}^{p-1} \left(\frac{x_i - x_c}{x_c}\right)\right]. \quad (6)$$


 Fig. 2. Comparison of the arctangent function and some sigmoid functions. Note that the output of  $\arctan(\cdot)$  is in radian measure.

Note that  $\xi(x)$  may take a minus value if the neighbor intensities are smaller than that of the current pixel. By this means, we attempt to preserve more discriminating information in comparison to using the absolute value of  $\xi(x)$ . Intuitively, if  $\xi(x)$  is positive, it simulates the case that the surroundings are lighter than the current pixel. In contrast, if  $\xi(x)$  is negative, it simulates the case that the surroundings are darker than the current pixel.

**Discussion on using arctangent function.** As shown in (4), we use the arctangent function  $G_{arctan}(\cdot)$  to compute  $\xi(x_c)$ . We plot the curve of this function in Fig. 2. Here, we use this function since it can limit the output to prevent it from increasing or decreasing too quickly when the input becomes larger or smaller.

One optional filter is a logarithm function, which matches a human being's perception well. However, it cannot be used here since many outputs of (2) are negative. Another optional filter is a sigmoid function:

$$\text{sigmoid}(\beta x) = \frac{1 - e^{-\beta x}}{1 + e^{-\beta x}}, \quad (7)$$

where  $\beta > 0$ . It is a typical neuronal nonlinear transfer function and is widely used in artificial neural networks [3]. Both arctangent and sigmoid functions have similar curves, as shown in Fig. 2, especially when  $\beta = 2$ . In our case, we use the former for simplicity.

As shown in Fig. 3, we plot an average histogram of the differential excitations on 2,000 texture images. One can find that there are more frequencies at the two sides of the average histogram (e.g.,  $[-\pi/2, -\pi/3]$  and  $[\pi/3, \pi/2]$ ). It results from two factors: One is the delimitation effect of the

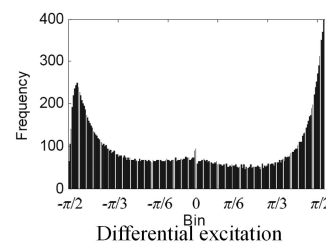


Fig. 3. The average histogram of the differential excitations on 2,000 texture images.

arctangent function, as shown in Fig. 2; the other is the approach used in computing the differential excitation  $\xi$  of a pixel (i.e., a sum of the difference ratios of  $p$  neighbors against a central pixel), as shown in (6). However, it is valuable for a classification task. For more details, please refer to Sections 2.2.4, 3, and 4.

### 2.2.2 Orientation

As shown in Fig. 1, the orientation component of WLD is the gradient orientation as in [27], which is computed as:

$$\theta(x_c) = \gamma_s^1 = \arctan\left(\frac{\nu_s^{11}}{\nu_s^{10}}\right), \quad (8)$$

where  $\nu_s^{10}$  and  $\nu_s^{11}$  are the outputs of the filters  $f_{10}$  and  $f_{11}$ :

$$\nu_s^{10} = x_5 - x_1 \quad \text{and} \quad \nu_s^{11} = x_7 - x_3. \quad (9)$$

For simplicity,  $\theta$  is further quantized into  $T$  dominant orientations. Before the quantization, we perform the mapping  $f: \theta \mapsto \theta'$ :

$$\theta' = \arctan 2(\nu_s^{11}, \nu_s^{10}) + \pi, \quad \text{and} \quad \arctan 2(\nu_s^{11}, \nu_s^{10}) = \begin{cases} \theta, & \nu_s^{11} > 0 \text{ and } \nu_s^{10} > 0, \\ \pi + \theta, & \nu_s^{11} > 0 \text{ and } \nu_s^{10} < 0, \\ \theta - \pi, & \nu_s^{11} < 0 \text{ and } \nu_s^{10} < 0, \\ \theta, & \nu_s^{11} < 0 \text{ and } \nu_s^{10} > 0, \end{cases} \quad (10)$$

where  $\theta \in [-\pi/2, \pi/2]$  and  $\theta' \in [0, 2\pi]$ . This mapping considers the value of  $\theta$ , computed using (8), and the sign of  $\nu_s^{10}$  and  $\nu_s^{11}$ . The quantization function is then as follows:

$$\Phi_t = f_q(\theta') = \frac{2t}{T}\pi, \quad \text{and} \quad t = \text{mod}\left(\left\lfloor \frac{\theta'}{2\pi/T} + \frac{1}{2} \right\rfloor, T\right). \quad (11)$$

For example, as shown in Fig. 1, if  $T = 8$ , these  $T$  dominant orientations are  $\Phi_t = (t\pi)/4$ , ( $t = 0, 1, \dots, T-1$ ). In other words, those orientations located within the interval  $[\Phi_t - \pi/T, \Phi_t + \pi/T]$  are quantized to  $\Phi_t$ .

### 2.2.3 WLD Histogram

The idea of representing an image by histogram of gradient and orientations has been used in biologically plausible vision systems and in object detection and recognition [1], [5], [10], [32]. Motivated by this idea, as shown in Fig. 1, we first compute each pixel's differential excitation ( $\xi_j$ ) using (6), and orientation ( $\Phi_t$ ) using (11). As shown in Fig. 2, we then compute the 2D histogram  $\{WLD(\xi_j, \Phi_t)\}$ ,  $j = 0, 1, \dots, N-1$ ,  $t = 0, 1, \dots, T-1$ ,  $N$  is the dimensionality of an image and  $T$  is the number of the dominant orientations, as mentioned in Section 2.2.2. Note that the size of this 2D histogram is  $T \times C$ , where  $C$  is the number of cells in each orientation (for more details of this parameters, please refer to the following part of this section). In other words, in this 2D histogram, each column corresponds to a dominant orientation  $\Phi_t$ , and each row corresponds to a differential excitation histogram with  $C$  bins. Thus, the intensity of each cell corresponds to the frequencies of a certain differential excitation interval on a dominant orientation.

To obtain a more discriminative descriptor, the 2D histogram  $\{WLD(\xi_j, \Phi_t)\}$  is further encoded into a 1D histogram  $H$ . Specifically, given the 2D histogram

$\{WLD(\xi_j, \Phi_t)\}$  of an image, as shown in Fig. 4a, we project each column of the 2D histogram to form a 1D histogram  $H(t)$  ( $t = 0, 1, \dots, T-1$ ). That is, we regroup the differential excitations  $\xi_j$  into  $T$  subhistograms  $H(t)$ , each subhistogram  $H(t)$  corresponding to a dominant orientation (i.e.,  $\Phi_t$ ). Subsequently, each subhistogram  $H(t)$  is evenly divided into  $M$  segments, i.e.,  $H_{m,t}$  ( $m = 0, 1, \dots, M-1$ , and in our implementation we set  $M = 6$ ). All of these subhistogram segments  $H_{m,t}$  form a histogram matrix. Each column corresponds to a dominant orientation, and each row corresponds to a differential excitation segment (i.e., having similar differential excitation values). The histogram matrix is then reorganized as a 1D histogram  $H$ . Specifically, each row of the histogram matrix is concatenated as a subhistogram  $H_m$  (i.e.,  $H_m = \{H_{m,t}\}, t = 0, 1, \dots, T-1$ ). Concatenating the resulting  $M$  subhistograms, we have the 1D histogram:  $H = \{H_m\}, m = 0, 1, \dots, M-1$ .

Note that, after each subhistogram,  $H(t)$  is evenly divided into  $M$  segments, the range of differential excitations  $\xi_j$  (i.e.,  $l = [-\pi/2, \pi/2]$ ) is also evenly divided into  $M$  intervals  $l_m$  ( $m = 0, 1, \dots, M-1$ ). Thus, for each interval  $l_m$ , we have  $l_m = [\eta_{m,l}, \eta_{m,u}]$ . Here, the lower bound  $\eta_{m,l} = (m/M - 1/2)\pi$  and the upper bound  $\eta_{m,u} = [(m+1)/M - 1/2]\pi$ . For example,  $l_0 = [-\pi/2, -\pi/3]$ .

Furthermore, as shown in Fig. 4b, each subhistogram segment  $H_{m,t}$  is composed of  $S$  bins, i.e.,

$$H_{m,t} = \{h_{m,t,s}\}, s = 0, 1, \dots, S-1.$$

Herein,  $h_{m,t,s}$  is computed as:

$$h_{m,t,s} = \sum_j \delta(S_j == s), \quad \left(S_j = \left\lfloor \frac{\xi_j - \eta_{m,l}}{(\eta_{m,u} - \eta_{m,l})/S} + \frac{1}{2} \right\rfloor\right), \quad (12)$$

where the subscripts  $m, t$  of  $h_{m,t,s}$  are computed as follows:  $m$  is determined according to the interval to which the value of  $\xi_j$  belongs, i.e.,  $\xi_j \in l_m$ ,  $t$  is the index of quantized orientation, i.e.,  $\Phi_t = f_q(\theta'_j)$ , and  $t$  is computed as in (11); and  $\delta(\cdot)$  is the Kronecker delta function defined as follows:

$$\delta(X) = \begin{cases} 1, & X \text{ is true,} \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

Intuitively,  $h_{m,t,s}$  means the number of the pixels whose differential excitations  $\xi_j$  belong to the same interval  $l_m$ , and orientations  $\theta'_j$  are quantized to the same dominant orientation  $\Phi_t$  and that the computed index  $S_j$  is equal to  $s$ . Meanwhile,  $(\eta_{m,u} - \eta_{m,l})/S$  is the width of each bin, and

$$\frac{1}{(\eta_{m,u} - \eta_{m,l})/S} (\xi_j - \eta_{m,l})$$

is linear mapping, used to map the differential excitation to its corresponding bin since the values of  $\xi_j$  are of real. Note that, as shown in Fig. 4a, for the number of the cells in each column of the 2D histogram  $\{WLD(\xi_j, \Phi_t)\}$ , we have  $C = M \times S$ .

We segment the range of  $\xi$  into several intervals due to the fact that different intervals correspond to the different variances in a given image. For example, given two pixels  $P_i$  and  $P_j$ , if their differential excitations  $\xi_i \in l_0$  and  $\xi_j \in l_2$ , we say that the intensity variance around  $P_i$  is larger than that of

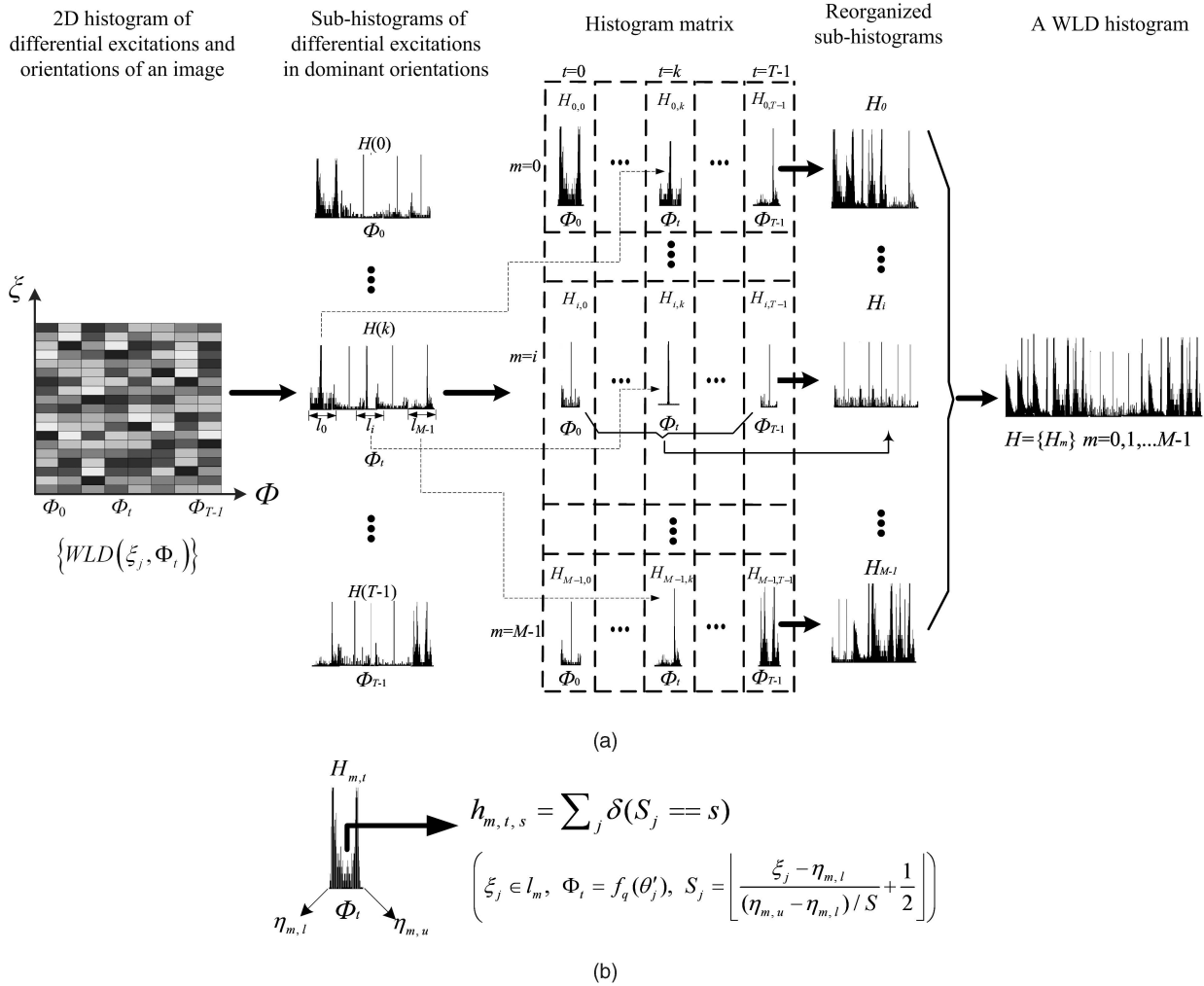


Fig. 4. An illustration of a WLD histogram for a given image. (a)  $H$  is concatenated by  $M$  subhistograms  $\{H_m\}$  ( $m = 0, 1, \dots, M-1$ ). Each  $H_m$  is concatenated by  $T$  subhistogram segments  $H_{m,t}$  ( $t = 0, 1, \dots, T-1$ ). Meanwhile, for each column of the histogram matrix, all of the  $M$  segments  $H_{m,t}$  ( $m = 0, 1, \dots, M-1$ ) have the same dominant orientation  $\Phi_t$ . In contrast, for each row of the histogram matrix, the differential excitations  $\xi_j$  of each subhistogram segment  $H_{m,t}$  ( $t = 0, 1, \dots, T-1$ ) belong to the same interval  $l_m$ . (b) A subhistogram segment  $H_{m,t}$ . Note that if  $t$  is fixed, for any  $m$  or  $s$ , the dominant orientation of a bin  $h_{m,t,s}$  is fixed (i.e.,  $\Phi_t$ ).

$P_j$ . That is, flat regions of an image produce smaller values of  $\xi$  while nonflat regions produce larger values. However, besides the flat regions of an image, there are two kinds of intensity variations around a central pixel which might lead to smaller differential excitations. One is the clutter noise around a central point; the other is the “uniform” patterns as shown in [34]. Meanwhile, the latter provides a majority of variations in comparison to the former, and the latter can be discriminated by the orientations.

Here, we let  $M = 6$  for the reason that we attempt to use these intervals to approximately simulate the variances of high, middle, or low frequency in a given image. That is, for a pixel  $P_i$ , if its differential excitation  $\xi_i \in l_0$  or  $l_5$ , we say that the variance near  $P_i$  is of high frequency; if  $\xi_i \in l_1$  or  $l_4$ , or  $\xi_i \in l_2$  or  $l_3$ , we say that the variance near  $P_i$  is of middle frequency or low frequency, respectively.

#### 2.2.4 Weight for a WLD Histogram

Intuitively, one often pays more attention to the regions of high variances in a given image compared with the flat

regions. So, the different frequency segments  $H_m$  should play different roles in a classification task. Thus, we can assign different weights to different segments  $H_m$  for a better classification performance.

For weight selection, a heuristic approach is to take into account the different contributions of the different segments  $H_m$  ( $m = 0, 1, \dots, M-1$ ). First, by computing the recognition rate on a collected texture data set from the Internet for each subhistogram  $H_m$  separately, we obtain  $M$  rates  $R = \{r_m\}$ ; then, we set each weight  $\omega_m = r_m / \sum_i r_i$ , as shown in Table 1. Simultaneously, in this table, we also collect statistics on the percentage of frequencies of each

TABLE 1  
Weights for a WLD Histogram

	$H_0$	$H_1$	$H_2$	$H_3$	$H_4$	$H_5$
Frequency percent	0.2519	0.1179	0.1186	0.0965	0.0875	0.3276
Weights ( $\omega_m$ )	0.2688	0.0852	0.0955	0.1000	0.1018	0.3487

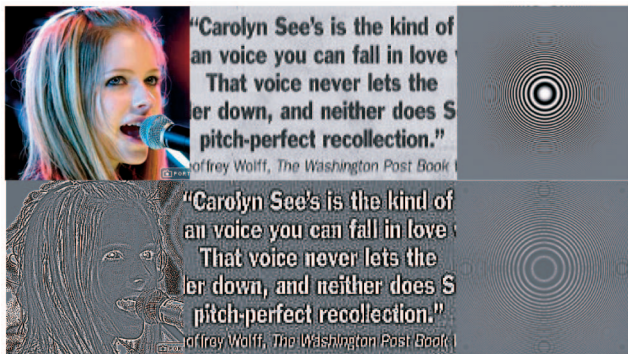


Fig. 5. The upper row contains original images, and the bottom row shows the filtered images by the proposed WLD. The value of the intensity in the filtered image is the differential excitation scaled to  $[0, 255]$  for visualization purposes.

subhistogram. From this table, one can find that these two groups of values (i.e., frequency percentage and weights) are similar. In addition, a high frequency segment  $H_0$  or  $H_5$  includes more frequencies (cf. Fig. 3 for more details) than the middle or low frequency segments.

Note that a high frequency segment  $H_0$  or  $H_5$  taking a larger weight is consistent with the intuition that a good classification feature should pay more attention to the salient variations of an object. Here, besides the large changes at the edges or occlusion boundaries within an image, the approach of computing differential excitation of a pixel (i.e., a sum of the differences of  $p$  neighbors against a central pixel) further contributes to a larger weight of  $H_0$  or  $H_5$ . However, a side effect of the weighting approach might enlarge the influence of noise. One can avoid this disadvantage by removing a few bins at the end of high frequency segments, that is, the left end of  $H_{0,t}$  and the right end of  $H_{M-1,t}$  ( $t = 0, 1, \dots, T-1$ ).

Although the weights shown in Table 1 are computed from the texture classification, our experiments show that they are also useful for face detection, since faces can be seen as a composition of micropatterns which are well described by a local operator [2].

### 2.3 Characteristics of WLD

The proposed descriptor, WLD, is based on Weber's Law. It has several advantages, such as detecting edges elegantly, robustness to noise and illumination change, and its powerful representation ability.

WLD is based on a physiological law. It extracts features from an image by simulating a human sensing his/her surroundings. Specifically, as shown in Fig. 1, a WLD uses the ratio of the intensity differences  $\nu_s^{00}$  to  $\nu_s^{01}$ , motivated by Weber's Law. As expected, WLD gains powerful representation ability for textures.

The detected edges match the subjective criterion elegantly since WLD depends on the perceived luminance difference. For example, as shown in (2), WLD preserves the differences ( $\nu_s^{00}$ ) between its neighbors and a center pixel. Sometimes  $\nu_s^{00}$  may be quite large. But if  $\nu_s^{00}/\nu_s^{01}$  is smaller than a noticeable threshold, there is not a noticeable edge. In contrast,  $\nu_s^{00}$  may be quite small. But if  $\nu_s^{00}/\nu_s^{01}$  is larger than a noticeable threshold, there is a noticeable edge. In Fig. 5, we show some filtered images produced by WLD, from which

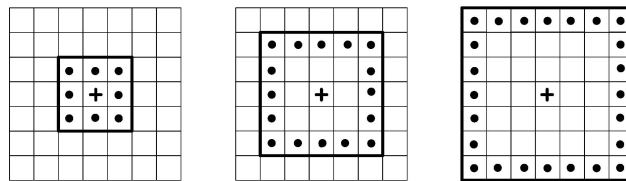


Fig. 6. Squared symmetric neighborhood for different  $(P, R)$ .

one could conclude that a WLD extracts the edges of images perfectly even with heavy noise (Fig. 5, middle column). Furthermore, the results of texture analysis show that much of the discriminative texture information is contained in high spatial frequencies such as edges [34]. Thus, the WLD works well to obtain a powerful feature for textures.

WLD is robust to noise appearing in a given image. Specifically, a WLD reduces the influence of noise, as it is similar to the smoothing in image processing. As shown in Fig. 1, a differential excitation is computed by a sum of its  $p$ -neighbor differences to a current pixel. Thus, it reduces the influence of noisy pixels. Moreover, the sum of its  $p$ -neighbor differences is further divided by the intensity of the current pixel, which also decreases the influence of noise in an image. For more details, please refer to Section 5.5.

WLD has been developed to reduce the effects of illumination change. On the one hand, it computes the differences  $\nu_s^{00}$  between its neighbors and a current pixel. Thus, a brightness change in which a constant is added to each image pixel will not affect the differences values. On the other hand, WLD performs the division between the differences  $\nu_s^{00}$  and  $\nu_s^{01}$ . Thus, a change in image contrast in which each pixel value is multiplied by a constant will multiply differences by the same constant, and this contrast change will be canceled by the division. Therefore, the descriptor is robust to changes in illumination.

Furthermore, regrouping the differential excitation and orientation into a 2D histogram and then weighting the different frequency segments can further improve the performance of the WLD descriptor.

### 2.4 Multiscale Analysis

WLD features described above are extracted from the  $3 \times 3$  neighborhood, which implies a single and fixed granularity. Motivated by the idea of [34], we also develop the multiscale WLD for characterizing local salient patterns in different granularities. It is computed using a square symmetric neighbor set of  $P$  pixels placed on a square whose sides have the length  $(2R + 1)$ , as shown in Fig. 6. Parameter  $P$  denotes the number of the neighbors, whereas  $R$  determines the spatial resolution of the operator.

With the neighborhood definition in Fig. 6, multiscale analysis of WLD can be accomplished by combining the information provided by multiple operators of varying  $(P, R)$ . We denote the operator as  $WLD_{P,R}$ . Although we derive the operator for a general case based on a squared symmetric neighbor set of  $P$  members on a square with side length  $(2R + 1)$ , one can also generalize them to a circular one. In addition, a straightforward approach for multiscale analysis is to concatenate the histograms from multiple operators realized with different  $(P, R)$ . In general, it can improve the discrimination of a single resolution of  $(P, R)$  (cf. Section 3.4).

TABLE 2  
Comparison of WLD with other Descriptors  
According to the FLS Framework

	filtering	labeling	statistics
LBP	Intensity difference	Thresholding at zero	Histogram over the binary strings
SIFT	Intensity difference	Orientation quantization	Histogram over the weighted gradient on dominant orientation
WLD	Weber ratio	Orientation quantization	Histogram over the differential excitation on dominant orientation

## 2.5 Comparison with Existing Descriptors

Using the filtering, labeling, and statistics (FLS) framework described in [17], we can easily compare our descriptor with the existing ones, as shown in Table 2. Note that, in the FLS framework, the step filtering depicts the interpixel relationship in a local image region; the step labeling (which includes quantization and mapping) describes the intensity variations which cause psychology redundancies; the step statistics capture the attribute which is not in adjacent regions [17].

For LBP, it computes the intensity differences between the center pixel  $x_c$  and its neighbors in the first stage, and the responses of each neighbor are thresholded at zero and are then concatenated to a binary string in the second stage. Each binary string corresponding to each pixel is then used to compute a histogram feature in the last stage. For SIFT, it computes the gradient magnitude and orientation at each image sample point in a region around the keypoint location in the first stage. The orientations are quantized to eight dominant ones in the second stage. For the third stage, the gradient magnitudes are weighted by a Gaussian window, and then accumulated into orientation histograms by summarizing the contents over denoted subregions.

Although WLD also computes the difference between the center pixel  $x_c$  and its neighbors like LBP in the first stage, these differences are added together and then divided by the center pixel  $x_c$  to obtain the differential excitation like the Weber fraction. Differently from LBP, WLD uses the gradient orientations to describe the direction of edges. The gradient orientations are then quantized to eight dominant orientations in the second stage. Differently from SIFT, we use the differential excitation but not the weighted gradient to compute the histogram. Moreover, differential excitations are not accumulated over denoted subregions around the keypoint location. In contrast, we compute the frequency of the occurrence of differential excitations for each bin of the histogram. Although we weighed the WLD histogram as shown in Table 1, the weighted object is the frequency of each bin, and the weights are computed according to the recognition performance based on the statistics, not weighting the values of gradients in terms of the distances between the neighbors and the keypoint, as does SIFT [27].

Furthermore, we also compare the time complexity of WLD with LBP and SIFT theoretically. Given an image in

TABLE 3  
Comparison of the Average Time Consumption  
with LBP and SIFT

Methods	LBP	WLD	SIFT
Time (s)	0.0015	0.0027	0.5419

$m \times n$ , the time complexity for WLD is very low. It is as follows:

$$O_{WLD} = C_1 mn, \quad (14)$$

where  $C_1$  is a constant. We use  $C_1$  for the computation of each pixel in WLD through several additions, divisions, and filtering with an arctangent function.

Likewise, the time complexity for LBP is also very simple:

$$O_{LBP} = C_2 mn, \quad (15)$$

where  $C_2$  is also a constant. We use  $C_2$  for the computation of each pixel in LBP through several additions.

However, the time complexity for SIFT is a little complicated:

$$O_{SIFT} = C_{31}(a\beta)(pq)(mn) + C_{32}k_1 + C_{33}k_2st + C_{34}k_2st. \quad (16)$$

Here, the four terms correspond to the four steps: detection of scale-space extrema, accurate keypoint localization, orientation assignment, and building the local image descriptor. Meanwhile,  $C_{3i}$  ( $i = 1, \dots, 4$ ) are four constants. For the first term, it represents the convolution of a variable-scale Gaussian with the given image. Here, the size of the convolution template is  $p \times q$ ;  $a, \beta$  represent the levels of octave and scales of each octave, respectively. For the second term,  $k_1$  represents the number of the keypoint candidates. For the third and fourth terms,  $k_2$  represents the number of the keypoints and  $s, t$  represent the size of the support regions for each keypoint (e.g.,  $s, t = 16$ ). So, we have:

$$O_{SIFT} \approx C_{31}(a\beta)(pq)(mn). \quad (17)$$

Comparing (14) and (15) with (17), one can find that both LBP and WLD are more efficient than SIFT. For the quantitative comparison of time consumptions of these three descriptors, please refer to Section 3.4 and Table 3.

## 3 APPLICATION TO TEXTURE CLASSIFICATION

In this section, we use the WLD histogram feature for texture classification and compare both the performance and computational efficiency with those of the state-of-the-art methods.

### 3.1 Background

Texture classification plays an important role in many applications, such as robot vision, content-based access to image databases, and automatic tissue recognition in medical images. Several approaches to the extraction of texture features have been proposed. On the one hand, there are some recent attempts using sparse descriptors for this

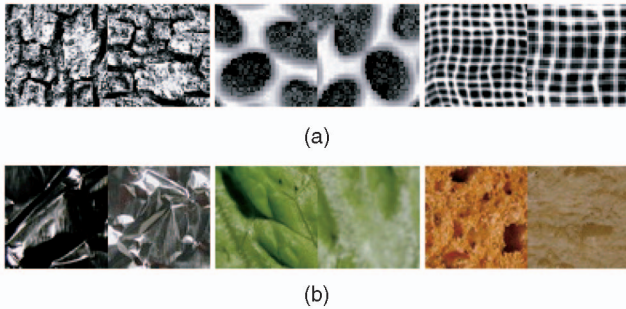


Fig. 7. Some examples from two texture data sets (a) Brodatz and (b) KTH-TIPS2-a.

task, such as [13], [25]. Dorkó and Schmid optimized the keypoint detection and then used SIFT for the image representation [13]. Lazebnik et al. presented a probabilistic part-based approach to describe the texture and object [25]. On the other hand, there are also several attempts using dense descriptors for this task, such as [22], [28], [33], [34], [35], [45], [46]. For example, Manjunath and Ma used Gabor filters for texture analysis [28]. Ojala et al. proposed the use of signed gray-level differences and their multidimensional distributions for texture description [35]. The original LBP is its simplification, discarding the contrast of local image texture [33], [34].

### 3.2 Data Set and Evaluation Protocol

Experiments are carried out on two different texture databases: Brodatz [7] and KTH-TIPS2-a [9]. Examples of the 32 Brodatz [35] textures used in the experiments are shown in Fig. 7a. The images are  $256 \times 256$  pixels in size, and they have 256 gray levels. Each image was divided into 16 disjoint samples of size  $64 \times 64$  pixels, which were independently histogram-equalized to remove luminance differences between textures. To make the classification problem more challenging and generic and to make a comparison possible, we use the same experimental setups as [22], [35], [45]. Three additional samples were generated from each sample: 1) a sample rotated by 90 degrees, 2) a  $64 \times 64$  scaled sample obtained from the  $45 \times 45$  pixels in the middle of the original sample, and 3) a sample that was both rotated and scaled. Consequently, the entire data set, which we refer to as the Brodatz data set, is comprised of 2,048 samples, with 64 samples in each of the 32 texture categories.

The KTH-TIPS2-a database contains four physical, planar samples of each of 11 materials under varying illumination, pose, and scale. Some examples from each sample are shown in Fig. 7b. The KTH-TIPS2-a texture data set contains 11 texture classes with 4,395 images. The images are  $200 \times 200$  pixels in size (we did not include those images which are not of this size), and they are transformed into 256 gray levels. The database contains images at nine scales, under four different illumination directions, and three different poses.

Note that we use different evaluation setups for the Brodatz and KTH-TIPS2-a texture databases for fair comparison with other typical evaluations. Specifically, for the Brodatz textures we use the same setup as that used in [22], [35], [45]. Experiments are carried out with 10-fold cross validation to avoid bias. For each round, we randomly

divide the samples in each class into two subsets of the same size, one for training and the other for testing. In this fashion, the images belonging to the training set and to the test set are disjoint. The results are reported as the average value and standard deviation over the 10 runs. In contrast, for the KTH-TIPS2-a textures, we use the same evaluation setup proposed by Caputo et al. [9]. Specifically, in our experiment, only three samples are available during training, while testing is subsequently performed on all the images of all the remaining samples. Similarly, this experiment is also repeated four times by randomly selecting three different samples for training. The results are also reported as the average value over the four runs.

### 3.3 The WLD Histogram for Classification

To perform the texture classification, there are two essential issues: texture representation and classifier design. We use WLD histogram feature as a representation and build a system for texture classification. For texture representation, given an image, we extract the WLD histogram as shown in Fig. 4. Here, we experientially set  $M = 6$ ,  $T = 8$ ,  $S = 20$ . In addition, we also weighed each subhistogram  $H_m$  using the same weights, as shown in Table 1.

As the classifier we use the  $K$ -nearest neighbor, which has been successfully utilized in classification. In our case,  $K = 3$ . To compute the distance between two given images  $I_1$  and  $I_2$ , we first obtain their WLD histogram features  $H^1$  and  $H^2$ . We then measure the similarity between  $H^1$  and  $H^2$ . In our experiments, we use the normalized histogram intersection  $\Pi(H^1, H^2)$  as a similarity measurement of two histograms [42]:

$$\Pi(H^1, H^2) = \sum_{i=1}^L \min(H^{1,i}, H^{2,i}), \quad (18)$$

where  $L$  is the number of bins in a histogram. The intuitive motivation for this measurement is to calculate the common parts of two histograms.

### 3.4 Experimental Results

Experimental results on Brodatz and KTH-TIPS2-a textures are illustrated in Fig. 8. Herein, the accuracy of our method is given as a percentage of correct classifications. It is computed as follows:

$$\text{accuracy} = \frac{\# \text{ correct classification}}{\# \text{ total images}}. \quad (19)$$

As shown in Fig. 8a, we compare our method with others on the classification task of Brodatz textures: SIFT, Jalba et al. [22], Ojala et al. [35] (i.e., signed gray-level difference (SD) and LBP), Urbach et al. [45], and Manjunath and Ma [28] (i.e., Gabor). Note that all of the results from other methods in Fig. 8a are quoted directly from the original papers except for those of Gabor [28] and SIFT. The approach in [28] is a “traditional” texture analysis method using a global mean and standard deviation of the responses of Gabor filters. We use the results in [35] for a substitution, in which Ojala et al. use the same setups for Gabor filters of six orientations and four scales. In addition, SIFT is reimplemented by us according to [13], in which they optimized the keypoint detection to achieve stable local descriptors. In addition, we

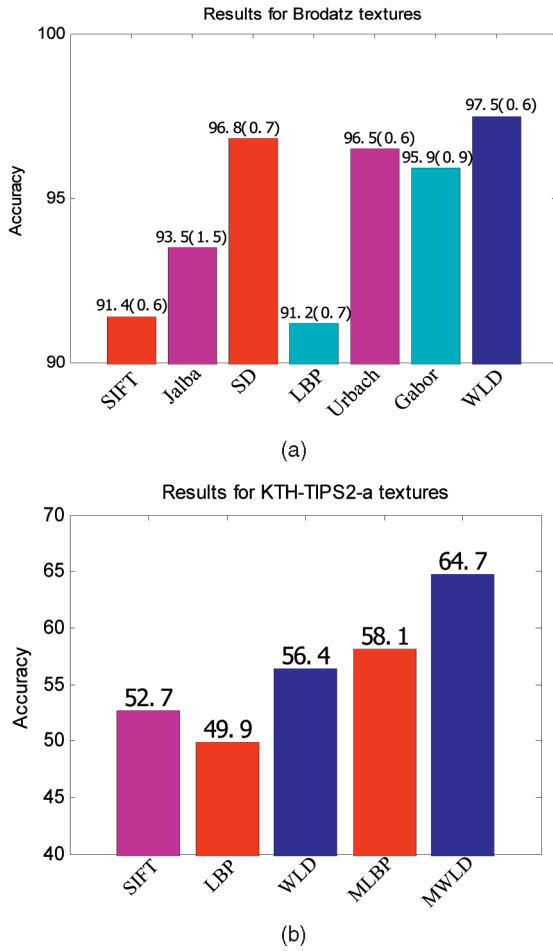


Fig. 8. Results comparison with state-of-the-art methods on Brodatz and KTH-TIPS2-a textures, where the values above the bars are the accuracy (both (a) and (b)) and corresponding standard deviations ((a) only).

also refer to the code by Vedaldi (<http://vision.ucla.edu/~vedaldi/code/siftpp/siftpp.html>).

As shown in Fig. 8b, we also compare our method with SIFT and LBP on the classification task of KTH-TIPS2-a textures. Likewise, both SIFT and LBP are reimplemented by us. However, in the implementation of SIFT, we use the Laplacian to detect keypoints in Fig. 8a, while in Fig. 8b, we employ the Harris detector, following the idea in [13].

From Fig. 8, one can find that our approach works in a very robust way in comparison to other methods. Moreover, the standard deviation of WLD shown in Fig. 8a is smaller compared with other methods. Although we have rotated and scaled the subimages of Brodatz textures, we also obtain favorable results. It shows that WLD extracts powerful discriminating features which are robust to rotation and scaling. The poorer performance of SIFT can be partly explained by the small image size (e.g.,  $64 \times 64$  in Brodatz database) for a sparse descriptor, from which too small a number of keypoints may be located, leading to the performance decrease. In addition, the performance of LBP can be improved by combining the contrast of images [34]. Note that due to the fact that the variations in the KTH-TIPS2-a set (i.e., pose, scale, and illumination) are much more diverse than those of the Brodatz set, the accuracies of all the tested descriptors (WLD, LBP, and SIFT) are

accordingly lower than those on the Brodatz. Note that we use the  $K$ -nearest neighbor classifier in this experiment because we attempt to compare the performances of different descriptors. However, using the proposed support-vector machine (SVM) based classification techniques proposed by Caputo et al. [9] might improve the performance significantly.

Since we have further extended WLD to multiscale (cf. Section 2.4), we also compared the performances of the multiscale versions of LBP (or multiresolution in [34]) and WLD. In Fig. 8b, we denote them by MLBP and MWLD, respectively. Specifically, MLBP denotes  $LBP_{8,1} + LBP_{16,2} + LBP_{24,3}$  and MWLD denotes  $WLD_{8,1} + WLD_{16,2} + WLD_{24,3}$ . The features of MLBP and MWLD for each image is to concatenate the histograms from multiple operators realized with different  $(P, R)$  as discussed in Section 2.4. From Fig. 8b, one can find that, compared with their single-resolution counterparts, both MWLD and MLBP significantly improve the accuracies (by 8.3 and 8.2 percent, respectively).

Besides the performance comparison with other methods, we also carried out an experiment on the Brodatz data set to compare the efficiency of WLD with LBP and SIFT. The experiments are performed on a 1.86 GHz Intel Pentium 4 processor using 1.50 GB RAM by executing C/C++ code. As shown in Table 3, to extract the features for an image of this data set, the average time consumptions of LBP and WLD are 0.0015 and 0.0027 seconds, respectively, while that of SIFT is 0.5419 seconds. Clearly, the computation of both LBP and WLD is much faster than that of SIFT. Herein, the codes of both LBP and SIFT are reimplemented by us. In addition, most of the time consumed by SIFT is spent on the first step (i.e., scale-space extrema detection), which needs to compute the convolution of a variable-scale Gaussian with the given image at different levels of octaves and scales (cf. (17)). Hence, inspired by SIFT, some speeded up SIFT-like descriptors are proposed, such as [4], [18], and [44].

## 4 APPLICATION TO FACE DETECTION

In this section, we use a WLD histogram for human face detection. Although we train only one classifier, we use it to detect frontal, occluded, and profile faces. Furthermore, experimental results show that this classifier obtains comparable performance to state-of-the-art methods.

### 4.1 Background

The goal of face detection is to determine whether there are any faces in a given image and return the location and extent of each face in an image if one or more faces are present. Recently, many methods for detecting faces have been proposed, and most of them extract the features densely [51]. Among these methods, learning-based approaches to capture the variations in facial appearances have attracted much attention, such as [39], [41]. One of the most important steps forward is the appearance of the boosting-based method, such as [6], [19], [20], [26], [36], [47], [49], [50]. In addition, Garcia and Delakis use a convolutional face finder for fast and robust face detection [15]. Hadid et al. use LBP for face detection and recognition [16].

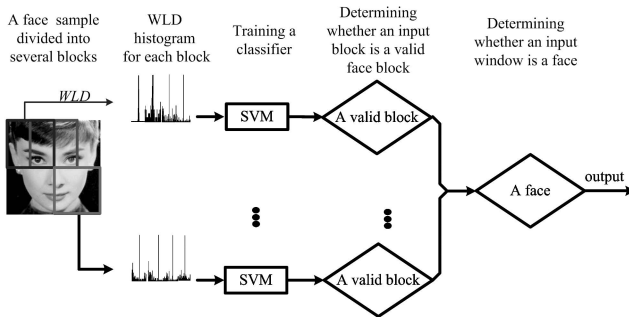


Fig. 9. An illustration of a WLD histogram feature for face detection.

## 4.2 WLD Histogram for Face Samples

Based on WLD, as shown in Fig. 9, we propose a new facial representation for face detection. Specifically, we divide an input sample into overlapping regions, and use a  $P$ -neighborhood WLD operator ( $P = 8$  and  $R = 1$ ). In our case, we normalize each sample to  $w \times h$  (e.g.,  $32 \times 32$ ) and derive a WLD histogram representation as follows:

We divide a face sample of size  $w \times h$  into  $K$  overlapping blocks ( $K = 9$  in our experiments) of size  $(w/2) \times (h/2)$  pixels. The overlapping size is equal to  $w/4$  pixels in a column and  $h/4$  pixels in a row. For each block, we compute a concatenated histogram  $H^k$ ,  $k = 0, 1, \dots, K - 1$ . Herein, each  $H^k$  is computed as shown in Fig. 4. That is, each  $H^k$  is a concatenated histogram with  $M$  subhistograms  $H_m^k$ ,  $m = 0, 1, \dots, M - 1$ , and  $H_m^k$  is also concatenated with  $T$  histogram patches  $H_{m,t}^k$ ,  $t = 0, 1, \dots, T - 1$ . Moreover,  $H_{m,t}^k$  is an  $S$ -bin histogram patch. In addition, for this group of experiments, we experientially set  $M = 6$ ,  $T = 4$ ,  $S = 3$ . Note that for each subhistogram  $H_m^k$ , we use the same weights as shown in Table 1.

For each block, we train an SVM classifier using an  $H^k$  histogram feature to verify whether the  $k$ th block is a valid face block (in our case, we use a second degree polynomial kernel function for the SVM classifier). If the number of the valid face blocks is larger than a given threshold  $\Xi$ , we say that a face exists in the input window. As to the parameter  $\Xi$ , its value is a trade-off between the detection rate and false alarms for a face detector. That is, when the value of  $\Xi$  becomes larger, the detection rate decreases but the false alarms also decrease. In contrast, when the value of  $\Xi$  becomes smaller, the detection rate increases but the false alarms also increase. The value of  $\Xi$  also varies with the pose of faces. For more details, please refer to Section 4.4.

## 4.3 The Data Set

The training set is composed of two sets, i.e., a positive set  $S_f$  and a negative set  $S_n$ . The positive set consists of 50,000 frontal face samples. They are collected from Web, video, and digital cameras, and cover wide variations in poses, facial expressions, and also in lighting conditions. To make the detection method robust to an affine transform, the training samples are often rotated, translated, and scaled [39]. After such preprocessing, we obtain the set  $S_f$  including 100,000 face samples. The negative set  $S_n$  consists of 31,085 images containing no faces, and they are collected from the Internet.

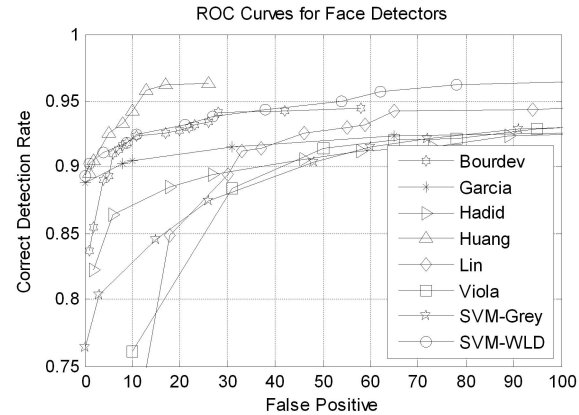


Fig. 10. A performance comparison of our method with some existing methods on the MIT+CMU frontal face test set.

As for the test sets, we use three: The first one is the MIT+CMU frontal face test set, which consists of 130 images showing 507 upright faces [39]. The second one is a subset from the Aleix Martinez-Robert (AR) face database [29]. The AR face database consists of over 3,200 color images of the frontal view faces from 126 subjects. However, we choose those images with occlusions (i.e., conditions of 8-13 from the first session, and conditions of 21-26 from the second session). The resulting test set consists of 1,512 images. The third one is the CMU profile testing set [41] (441 multiview faces in 208 images).

Note that the face samples are of the size  $32 \times 32$ . In order to detect some faces smaller or larger than the sample size, we enlarge and shrink each input image.

## 4.4 Classifier Training

As described in Section 4.3, the  $S_f$  set is composed of a large number of face samples. Furthermore, we can also extract hundreds of thousands of nonface samples from the  $S_n$  set. Thus, it is extremely time consuming to train an SVM classifier using the two sets  $S_f$  and  $S_n$ . For this problem, we use the resampling methods to train an SVM classifier. Specifically, motivated by [50], we also resample both the positive and negative samples during classifier training.

For the positive samples, we first randomly draw a subset  $S_{f1}$  with the size  $N_p$  (in our experiments,  $N_p = 3,000$ ). Likewise, we also randomly crop out a subset  $S_{n1}$  with the size  $N_n$  (in our experiments,  $N_n = 3,000$ ) from the nonface database  $S_n$ . Note that for the samples in  $S_{n1}$ , we normalize their sizes to  $w \times h$  (i.e.,  $32 \times 32$ ). Subsequently, we extract a WLD histogram of both the face and nonface samples as shown in Fig. 9. Using the extracted features of faces and nonfaces, we train a lower performance SVM classifier. Simultaneously, we obtain a support-vector set  $S_1$ , which includes a face support-vector subset  $S_f^1$  and a nonface support-vector subset  $S_n^1$ .

Using the resulting lower performance SVM classifier, we test it on the two training subsets (i.e.,  $S_f$  and  $S_n$ ) to collect  $N_p$  misclassified face samples  $S_{f2}$  and  $N_n$  misclassified nonface samples  $S_{n2}$ . Combining the newly collected sample sets (i.e.,  $S_{f2}$  and  $S_{n2}$ ) and the two support-vector subsets obtained last time (i.e.,  $S_f^1$  and  $S_n^1$ ), we obtain two new training sets:  $(S_f^1 + S_{f2})$  and  $(S_n^1 + S_{n2})$ . We then train another SVM classifier with a better performance. After

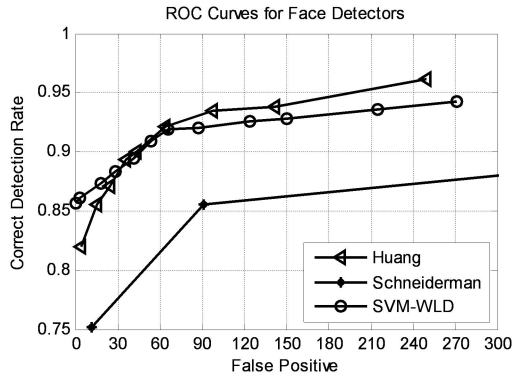


Fig. 11. A performance comparison of our method with some existing methods on the CMU profile testing set.

several iterations of the aforementioned procedure, we finally train a well-performed SVM classifier.

Note that we actually train  $K$  subclassifiers of SVM. Each subclassifier corresponds to a block as shown in Fig. 9. Combining these  $K$  subclassifiers, we obtain a final strong classifier.

#### 4.5 The Experiment Results

The resulting final strong SVM classifier is tested on the three testing sets described in Section 4.3. The experimental results are shown in Figs. 10 and 11 and Table 4, respectively. Herein, we compare the performance of the resulting SVM classifier (we call it “SVM-WLD”) with some existing methods. Note that all of the results from the original papers except for Hadid et al. [16] (which is implemented by us following their idea). During testing on these sets, the parameter  $\Xi$  takes the different values as described in Section 4.2. For the MIT+CMU frontal test set, the AR test set, and the CMU profile test set,  $\Xi$  is equal to 8, 7, and 6, respectively.

As shown in Fig. 10, we compare the performance of our method with some existing methods on the MIT+CMU frontal face test set, such as Bourdev and Brandt [6], Garcia and Delakis [15], Hadid et al. [16], Huang et al. [20], Lin et al. [26], and Viola and Jones [47]. Meanwhile, Lin et al. also proposed a method for detecting occluded faces. SVM-Gray denotes that we only use the gray intensities as input for the SVM classifier, and other experimental setups are the same as for the SVM-WLD. From Fig. 10, one can find that SVM-WLD locates 89.3 percent faces without any false alarm, and works much better than SVM-Gray (76.3 percent faces without a

TABLE 4  
Performance of Our Method on the AR Test Set

Detection rate	False Alarms
99.7%	0
100%	3

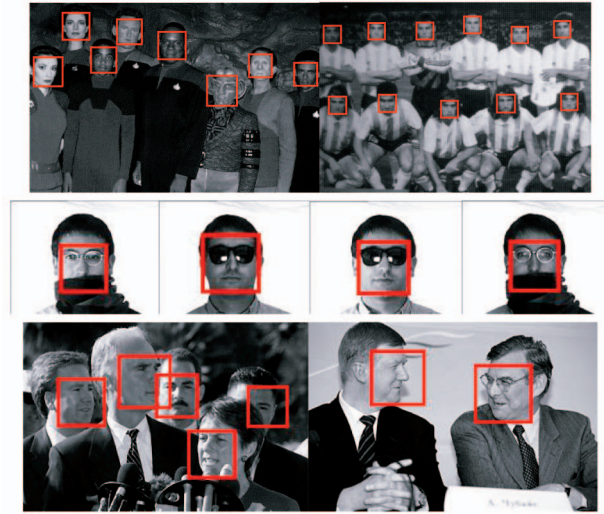


Fig. 12. Some experimental results from our detector on the MIT+CMU frontal test set (first row), the AR database (second row), and the CMU profile test set (third row).

false alarm). Furthermore, SVM-WLD is comparable to the existing methods, e.g., Lin et al. [26].

In Table 4, we show the detection results on the AR test set. From this table, one can find that SVM-WLD locates 99.7 percent faces without any false alarm and locates all faces with only three false alarms. In addition, in Fig. 11, we compare SVM-WLD with some existing methods, such as Huang et al. [20] and Schneiderman and Kanade [41] on the CMU profile test set. To locate those profile faces with in-plane rotation, we also rotate the testing images. From Fig. 11, one can find that SVM-WLD locates 85.7 percent faces without any false alarm.

However, different criteria (e.g., the training examples involved and the number of scanned windows during detection, etc.) can be used to favor one over another, which makes it difficult to evaluate the performance of different methods, even though they use the same benchmark data sets [51]. Thus, the results shown in Figs. 10 and 11 just illustrate that our method works robustly and achieves a performance comparable to the state-of-the-art methods. Some results from our detector on these three test sets are shown in Fig. 12.

## 5 MORE EXPERIMENTAL VALIDATIONS AND DISCUSSIONS

In this section, by conducting more experimental validations, we will discuss some issues about our method, including the relationship between WLD and Weber’s Law, the influence of the parameter setting on the performance of WLD, the different filter functions for the computation of the differential excitation, the influence of the WLD components on its performance, and the robustness of WLD to noise.

### 5.1 WLD and Weber’s Law

In this section, we compare the performance of the proposed descriptor WLD with another descriptor which also follows Weber’s Law, but employs a logarithm filter

TABLE 5  
Performance Comparison of  $WLD_{arctan}$  and  $WLD_{log}$

Method	$WLD_{arctan}$	$WLD_{log}$
Brodatz	97.5	93.3
KTH-TIPS2-a	56.4	50.2

function. We analyze the reason for the performance difference between these two descriptors, and provide evidence to support our results and analyses.

As presented in Section 2.1, WLD is motivated by Weber's Law. However, we should point out that WLD is not the only possible way to follow the Law. Its computing approach also does not exactly match the Law well. Moreover, an alternative method exactly following the Law might be as follows, employing a Logarithm operator: Specifically, Weber's Law can be described as  $dp = C(dI/I)$ , where  $dp$  is the differential change in perception, and  $dI$  is the differential change in image intensity. By integrating this equation, one can get  $p = C \log(I/I_r)$ , where  $I_r$  is the threshold below which no change can be perceived. Hence, the 2D histogram of  $\log(I/I_r)$  and gradient orientation can also be employed as an image descriptor. Furthermore, to improve the robustness of  $\log(I/I_r)$  to the variations due to illumination and noise, one can use  $\log(I/I_m)$  (i.e.,  $\log(I/I_r) - \log(I_m/I_r)$ ) instead, where  $I_m$  is the mean in a local neighborhood. We call this alternative descriptor  $WLD_{log}$ , and denote the descriptor presented in Section 2 as  $WLD_{arctan}$  for clarity.

We compare the performance of the two descriptors  $WLD_{log}$  and  $WLD_{arctan}$  on the Brodatz and KTH-TIPS2-a textures. The results are shown in Table 5. Note that  $I_r$  is set to 5 in the experiments. From the table, one can find that  $WLD_{arctan}$  outperforms  $WLD_{log}$  in both databases, although the latter matches Weber's Law well. One explanation for  $WLD_{arctan}$  outperforming  $WLD_{log}$  in both databases is that the performance gain of  $WLD_{arctan}$  benefits from the gradient computation with the filter  $f_0$  when we compute the differential excitations in (2) since the gradient is relatively more robust to the illumination variations compared with the image intensity. The observation that the performance gap for the KTH-TIPS2-a textures (6.2 percent) is a little larger than that for the Brodatz textures (4.2 percent)

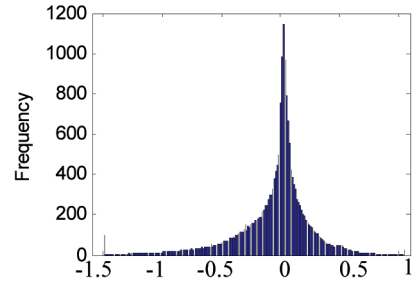


Fig. 13. Average histogram of  $\log(I/I_m)$  on 2,000 texture images. Note that for  $\log(I/I_m)$ , we only plot the values in the interval  $[-1.5, 1]$ , and the smaller and larger values are counted into the first bin and the last bin.

also supports this point since the illumination variations of the KTH-TIPS2-a textures are much more diverse than those of the Brodatz textures.

We provide more evidence to validate that  $WLD_{arctan}$  outperforms  $WLD_{log}$  with the distribution of the average histogram. Specifically, as shown in Fig. 13, we plot the average histogram of  $\log(I/I_m)$  on 2,000 textures images. By comparing Figs. 13 and 3, one can find that the frequencies of the differential excitations of  $WLD_{arctan}$  distribute more evenly than those of the  $\log(I/I_m)$ , which also provides evidence that the discrimination of  $WLD_{arctan}$  outperforms that of  $WLD_{log}$ .

## 5.2 The Effects of Parameters

In this section, we discuss the influence of the parameter setting of  $M$ ,  $T$ , and  $S$ . For a histogram-based method, the setting of  $M$ ,  $T$ , and  $S$  is a trade-off between discriminability and statistical reliability. In general, if these parameters (i.e.,  $M$ ,  $T$ , and  $S$ ) become larger, the dimensionality of the histogram (i.e., the number of its bins) becomes larger and thus the histogram becomes more discriminable. However, in a real application, the entries of each bin become smaller because the size of the input image/patch is fixed. This degrades the statistical reliability of the histogram. If the entries of each bin become too small, it in turn degrades the discriminability of the histogram because of its poor statistical reliability. In contrast, if these parameters (i.e.,  $M$ ,  $T$ , and  $S$ ) become smaller, the entries of each bin become larger, and the histogram becomes statistically more reliable. However, if these parameters are too small, the dimensionality of the histogram also becomes too small, and it degrades the discriminability of the histogram. The experiment results with the Brodatz textures when varying the parameters are plotted in Fig. 14.

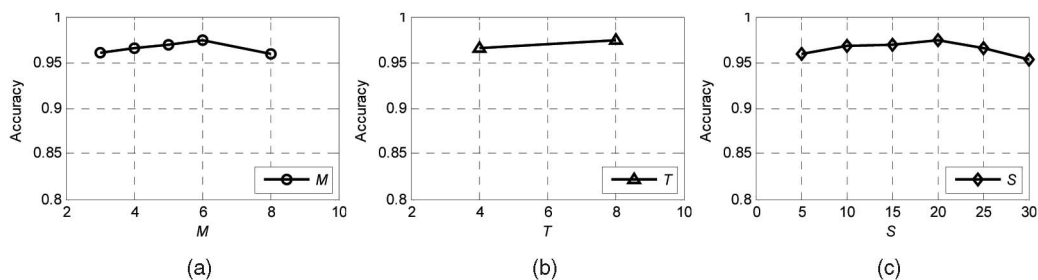


Fig. 14. The effects of using different  $M$ ,  $T$ , and  $S$  parameters.

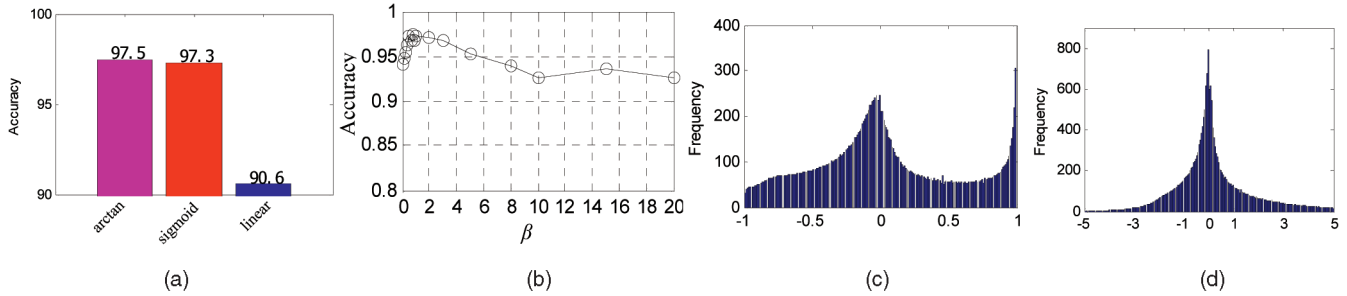


Fig. 15. (a) Performance comparison of WLD using different filter functions; (b) performance comparison of WLD using a sigmoid function and the parameter  $\beta$  takes different values; (c) and (d) are the plots of average histograms of the differential excitations using different functions on 2,000 texture images, i.e., (c) for sigmoid ( $\beta = 1$ ) and (d) for the Weber fraction  $G_{ratio}(x_c)$  (we only plot the values in the interval  $[-5, 5]$ . For those smaller and larger values, we use the first and last bin to compute their frequencies, respectively).

In these experiments, we just vary one parameter and fix the other ones. One can discover that the performance changes slightly, which shows that the histogram-based method is relatively robust, although obviously its dimensionality changes.

### 5.3 Performance Comparison Using Different Filters

We compare the influence of different filters, as mentioned in Section 2.2.1 (i.e., arctangent and sigmoid functions) on the performance of the WLD descriptor. The test data set is the Brodatz data set, and the performances of three different filters are shown in Fig. 15a. In the figure, “linear” means that we directly use the  $G_{ratio}(x_c)$  (shown in (3)) as the differential excitation of the current pixel  $\xi(x_c)$ .

From Fig. 15a, one can find that both the WLD descriptors using the sigmoid ( $\beta = 1$ ) and arctangent functions obtain very similar performance, and both outperform the WLD descriptor using the linear form (i.e.,  $G_{ratio}(x_c)$ ). As shown in Figs. 15c and 15d, we plot the average histograms of the differential excitations using a sigmoid function and  $G_{ratio}(x_c)$  on 2,000 texture images. One can observe that both the average histograms of the differential excitations using sigmoid function (as shown in Fig. 15c) and using arctangent function (as shown in Fig. 3) are more even than that using the linear form.

We further compare the performance of the WLD descriptor using a sigmoid function and the parameter  $\beta$  takes different values. As shown in Fig. 15b, one can see that the accuracy decreases slightly as  $\beta$  increases. Therefore, we set  $\beta = 1$  in Fig. 15a.

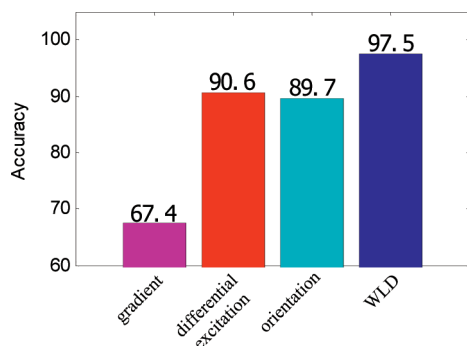


Fig. 16. Performance comparison of different components of WLD.

### 5.4 Performance Comparison of Components

As described in Section 2, WLD consists of two components: differential excitation and orientation. In this section, we compare the contributions of these two components to the performance of WLD. In addition, we also test the performance of directly using gradient, which is computed as shown in (2). In these comparisons, one can make thorough observations on the contributions of these components.

Tests were conducted on the Brodatz textures, and the performances are shown in Fig. 16. In the figure, “gradient” denotes the method using gradients directly computed by (2), “differential excitation” denotes that using the differential excitation  $\xi(x)$  computed by (6); “orientation” means that using the gradient orientation  $\theta(x)$  computed by (8). Each of these components from each image is grouped as a histogram for classification.

From Fig. 16, one can see that the performance gap between differential excitation and gradient (23.2 percent) is much larger than that between WLD and differential excitation (6.9 percent). Furthermore, the performance of orientation is close to that of differential excitation. It shows that the ratio used for the computation of differential excitation and orientation is important for the performance of the WLD descriptor. We believe that the ratio can remove the multiplicative noise further compared with gradient, which is helpful for the discrimination of a descriptor.

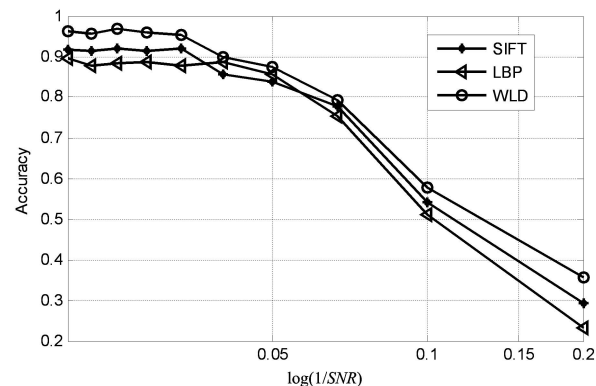


Fig. 17. A performance comparison between WLD, LBP, and SIFT on the Brodatz textures with added white Gaussian noise.

## 5.5 Robustness to Noise

As discussed in Section 2.2, both of the two components of WLD (i.e., differential excitation and orientation) are computed using the ratio (cf. (6) and (8)). Thus, WLD is robust to multiplicative noise. In this section, we test the robustness of WLD to additive noise, and also compare its performance with that of SIFT and LBP.

As can be seen from Fig. 17, we compare the performances of WLD, LBP, and SIFT on the Brodatz textures with added white Gaussian noise. Here, the  $x$  axis is  $\log(1/SNR)$ , i.e., the logarithm of the inverse of the signal-to-noise ratio (SNR). SNR is computed as:  $SNR = f(I)/f(N)$ , where  $f(I)$  and  $f(N)$  are the power of the input image  $I$  and the noise image  $N$ :  $f(I) = \frac{1}{n} \sum_{i=0}^{n-1} |I_i^2|$ ,  $f(N) = \frac{1}{n} \sum_{i=0}^{n-1} |N_i^2|$ , where  $n$  is the dimensionality of the input image and noise image.

From Fig. 17, one can see that the three descriptors WLD, LBP, and SIFT are equally robust to the added white Gaussian noise when the noise strength is smaller than 5 percent. The performances of these three descriptors decrease when the noise strength is larger than 5 percent. However, the performance of WLD is still better than that of SIFT and LBP. For WLD, we believe that the reasons lie in the following issues: the computation means of differential excitation using the gradient and ratio, the suppression toward the ends of arctangent function, and the histogram computation clustering the near differential excitations.

## 6 CONCLUSION

We propose a novel discriminative descriptor called WLD. It is inspired by Weber's Law, which is a law developed according to the perception of human beings. We organize WLD features to compute a histogram by encoding both differential excitations and orientations at certain locations. Experimental results show that WLD illustrates a favorable performance on both Brodatz and KTH-TIPS2-a textures compared with the state-of-the-art methods (e.g., SIFT and LBP). Besides the performance comparison with the other methods, we also compare the computational cost of WLD with LBP and SIFT. The analysis shows that the computation of WLD is much faster compared with that of SIFT, and is comparable to that of LBP.

For the face detection task, we train only one classifier, but it can accurately detect the frontal, occluded, and profile faces. The results on the three data sets, i.e., the MIT+CMU frontal face test set, the AR face database, and the CMU profile testing set, demonstrate the effectiveness of the proposed method through experiments and comparisons with other existing face detectors.

The current work has been developed for texture classification and face detection. Future interest lies in how to exploit the proposed descriptor for the domain of face recognition and object recognition.

## ACKNOWLEDGMENTS

This work has been partially supported by the National Natural Science Foundation of China under contract Nos. 60772071, 60833013, U0835005, and 60702041, the National Basic Research Program of China (973 Program) under contract No. 2009CB320902, and the Academy of Finland. Jie Chen was with the Key Lab of Intelligent Information

Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, China, where most of the work was done, as a research assistant and when he was also a PhD candidate of the School of Computer Science and Technology, Harbin Institute of Technology.

## REFERENCES

- [1] M. Agrawal, K. Konolige, and M.R. Blas, "CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching," *Proc. European Conf. Computer Vision*, 2008.
- [2] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face Description with Local Binary Patterns: Application to Face Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037-2041, Dec. 2006.
- [3] J. Anderson, *An Introduction to Neural Networks*. The MIT Press, 1995.
- [4] H. Bay, T. Tuytelaars, and L. van Gool, "SURF: Speeded Up Robust Features," *Proc. European Conf. Computer Vision*, 2006.
- [5] J. Begard, N. Allezard, and P. Sayd, "Real-Time Human Detection in Urban Scenes: Local Descriptors and Classifiers Selection with AdaBoost-like Algorithms," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition Workshops*, 2008.
- [6] L. Bourdev and J. Brandt, "Robust Object Detection via Soft Cascade," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2005.
- [7] P. Brodatz, *Textures: A Photographic Album for Artists and Designers*. Dover Publications, 1966.
- [8] V. Bruni and D. Vitulano, "A Generalized Model for Scratch Detection," *IEEE Trans. Image Processing*, vol. 13, no. 1, pp. 44-50, Jan. 2004.
- [9] B. Caputo, E. Hayman, and P. Mallikarjuna, "Class-Specific Material Categorisation," *Proc. IEEE Int'l Conf. Computer Vision*, 2005.
- [10] J. Chen, S. Shan, G. Zhao, X. Chen, W. Gao, and M. Pietikäinen, "A Robust Descriptor Based on Weber's Law," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2008.
- [11] H. Cheng, Z. Liu, N. Zheng, and J. Yang, "A Deformable Local Image Descriptor," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2008.
- [12] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2005.
- [13] G. Dorkó and C. Schmid, "Maximally Stable Local Description for Scale Selection," *Proc. European Conf. Computer Vision*, 2006.
- [14] L. Fei-Fei and P. Perona, "A Bayesian Hierarchical Model for Learning Natural Scene Categories," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2005.
- [15] C. Garcia and M. Delakis, "Convolutional Face Finder: A Neural Architecture for Fast and Robust Face Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1408-1423, Nov. 2004.
- [16] A. Hadid, M. Pietikäinen, and T. Ahonen, "A Discriminative Feature Space for Detecting and Recognizing Faces," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2004.
- [17] C. He, T. Ahonen, and M. Pietikäinen, "A Bayesian Local Binary Pattern Texture Descriptor," *Proc. Int'l Conf. Pattern Recognition*, 2008.
- [18] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of Interest Regions with Local Binary Patterns," *Pattern Recognition*, vol. 42, no. 3, pp. 425-436, 2009.
- [19] C. Huang, H. Ai, Y. Li, and S. Lao, "High-Performance Rotation Invariant Multiview Face Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 671-686, Apr. 2007.
- [20] C. Huang, H. Ai, T. Yamashita, S. Lao, and M. Kawade, "Incremental Learning of Boosted Face Detector," *Proc. IEEE Int'l Conf. Computer Vision*, 2007.
- [21] A.K. Jain, *Fundamentals of Digital Image Processing*, p. 51. Prentice-Hall, 1989.
- [22] A.C. Jalba, M.H.F. Wilkinson, and J.B.T.M. Roerdink, "Morphological Hat-Transform Scale Spaces and Their Use in Pattern Classification," *Pattern Recognition*, vol. 37, no. 5, pp. 901-915, 2004.
- [23] Y. Ke and R. Sukthankar, "PCA-SIFT: A More Distinctive Representation for Local Image Descriptors," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2004.

[24] S. Lazebnik, C. Schmid, and J. Ponce, "A Sparse Texture Representation Using Local Affine Regions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1265-1278, Aug. 2005.

[25] S. Lazebnik, C. Schmid, and J. Ponce, "A Maximum Entropy Framework for Part-Based Texture and Object Recognition," *Proc. IEEE Int'l Conf. Computer Vision*, 2005.

[26] Y.Y. Lin, T.L. Liu, and C.S. Fuh, "Fast Object Detection with Occlusions," *Proc. European Conf. Computer Vision*, 2004.

[27] D. Lowe, "Distinctive Image Features from Scale Invariant Key Points," *Int'l J. Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[28] B. Manjunath and W. Ma, "Texture Features for Browsing and Retrieval of Image Data," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837-842, Aug. 1996.

[29] A.M. Martinez and R. Benavente, "The AR Face Database," CVC Technical Report, #24, June 1998.

[30] P. Moreels and P. Perona, "Evaluation of Features Detectors and Descriptors Based on 3D Objects," *Int'l J. Computer Vision*, vol. 73, no. 3, pp. 263-284, 2007.

[31] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615-1630, Oct. 2005.

[32] K. Mikolajczyk and J. Matas, "Improving Descriptors for Fast Tree Matching by Optimal Linear Projection," *Proc. IEEE Int'l Conf. Computer Vision*, 2007.

[33] T. Ojala, M. Pietikäinen, and D. Harwood, "A Comparative Study of Texture Measures with Classification Based on Feature Distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51-59, 1996.

[34] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution Gray Scale and Rotation Invariant Texture Analysis with Local Binary Patterns," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971-987, July 2002.

[35] T. Ojala, K. Valkealahti, E. Oja, and M. Pietikäinen, "Texture Discrimination with Multidimensional Distributions of Signed Gray-Level Differences," *Pattern Recognition*, vol. 34, no. 3, pp. 727-739, 2001.

[36] M.T. Pham and T.J. Cham, "Fast Training and Selection of Haar Features Using Statistics in Boosting-Based Face Detection," *Proc. IEEE Int'l Conf. Computer Vision*, 2007.

[37] T. Phiasai, P. Kumhom, and K. Chamnongthai, "A Digital Image Watermarking Technique Using Prediction Method and Weber Ratio," *Proc. Int'l Symp. Comm. and Information Technologies*, 2004.

[38] T. Randen and J.H. Husoy, "Filtering for Texture Classification: A Comparative Study," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 291-310, Apr. 1999.

[39] H.A. Rowley, S. Baluja, and T. Kanade, "Neural Network-Based Face Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23-38, Jan. 1998.

[40] Y. Rodriguez and S. Marcel, "Face Authentication Using Adapted Local Binary Pattern Histograms," *Proc. European Conf. Computer Vision*, 2006.

[41] H. Schneiderman and T. Kanade, "A Statistical Method for 3D Object Detection Applied to Faces and Cars," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2000.

[42] M. Swain and D. Ballard, "Color Indexing," *Int'l J. Computer Vision*, vol. 7, no. 1, pp. 11-32, 1991.

[43] X. Tan and B. Triggs, "Enhanced Local Texture Feature Sets for Face Recognition under Difficult Lighting Conditions," *Proc. IEEE Int'l Workshop Analysis and Modelling of Faces and Gestures*, 2007.

[44] E. Tola, V. Lepetit, and P. Fua, "A Fast Local Descriptor for Dense Matching," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2008.

[45] E.R. Urbach, J.B.T.M. Roerdink, and M.H.F. Wilkinson, "Connected Shape-Size Pattern Spectra for Rotation and Scale-Invariant Classification of Gray-Scale Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 272-285, Feb. 2007.

[46] M. Varma and A. Zisserman, "Texture Classification: Are Filter Banks Necessary?" *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2003.

[47] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2001.

[48] S. Winder and M. Brown, "Learning Local Image Descriptors," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2007.

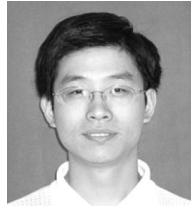
[49] R. Xiao, H. Zhu, H. Sun, and X. Tang, "Dynamic Cascades for Face Detection," *Proc. IEEE Int'l Conf. Computer Vision*, 2007.

[50] S. Yan, S. Shan, X. Chen, W. Gao, and J. Chen, "Matrix-Structural Learning (MSL) of Cascaded Classifier from Enormous Training Set," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2007.

[51] M.H. Yang, D. Kriegman, and N. Ahuja, "Detecting Faces in Images: A Survey," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34-58, Jan. 2002.

[52] G. Zhao and M. Pietikäinen, "Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 915-928, June 2007.

[53] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang, "Local Gabor Binary Pattern Histogram Sequence (LGBPHS): A Novel Non-Statistical Model for Face Representation and Recognition," *Proc. IEEE Int'l Conf. Computer Vision*, 2005.



**Jie Chen** received the MS and PhD degrees from Harbin Institute of Technology, China, in 2002 and 2007, respectively. Since September 2007, he has been a senior researcher in the Machine Vision Group at the University of Oulu, Finland. His research interests include pattern recognition, computer vision, machine learning, dynamic texture, and watermarking. He has authored more than 20 papers in journals and conferences, and he is a member of the IEEE.



**Shiguang Shan** received the MS degree in computer science from Harbin Institute of Technology, China, in 1999, and the PhD degree in computer science from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, in 2004. He has been with ICT, CAS, since 2002, and has been an associate professor since 2005. He is also the vice director of the ICT-ISVISION Joint Research and Development Laboratory for Face Recognition, ICT, CAS. His research interests cover image analysis, pattern recognition, and computer vision. He is focusing especially on face recognition related research topics, and has published more than 100 papers on the related research topics. He received the China's State Scientific and Technological Progress Awards in 2005 for his work on face recognition technologies. He is a member of the IEEE.



**Chu He** received the BS, MS, and PhD degrees from Wuhan University, China, in 1996, 1999, and 2007, respectively. He has been with the School of Electronic Information, Wuhan University, since 1999, and has been an associate professor since 2007. He was a visiting scholar in the Machine Vision Group at the University of Oulu, Finland, from 2007 to 2008. His research interests include image processing, computer vision, and synthetic aperture radar image interpretation. He has authored more than 30 papers in journals and conferences and received the Best Paper Award at CSAR 2005.



**Guoying Zhao** received the PhD degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2005. Since July 2005, she has been a senior researcher in the Machine Vision Group at the University of Oulu, Finland. Her research interests include gait analysis, dynamic texture recognition, facial expression recognition, human motion analysis, and person identification. She has authored more than 50 papers

in journals and conferences, and has served as a reviewer for many journals and conferences. With Professor Pietikäinen, she gave a tutorial: "Local Binary Pattern Approach to Computer Vision" at the 18th ICPR, August 2006, Hong Kong. With Professor Pietikäinen again, she gave a tutorial: "Local Texture Descriptors in Computer Vision," at the 12th ICCV, September 2009, Kyoto, Japan. She is authoring/editing three books: the Springer book *Computer Vision Using Local Binary Patterns*, the Springer book *Machine Learning for Vision-Based Motion Analysis*, and the IGI global book *Machine Learning for Human Motion Analysis: Theory and Practice*. She is a guest editor of the special issue on new advances in video-based gait analysis and applications: challenges and solutions in the *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*. She was a cochair of the ECCV 2008 Workshop on Machine Learning for Vision-based Motion Analysis (MLVMA), and was a cochair of the MLVMA workshop at ICCV '09.



**Matti Pietikäinen** received the DrSc degree in technology from the University of Oulu, Finland, in 1982. In 1981, he established the Machine Vision Group at the University of Oulu. This group has achieved a highly respected position in its field, and its research results have been widely exploited in industry. Currently, he is a professor of information engineering, a scientific director of the Infotech Oulu Research Center, and a 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. His research interests include texture-based computer vision, face analysis, human motion analysis, human-computer interaction, person identification, and visual surveillance. He has authored more than 250 refereed papers in international journals, books, and conference proceedings, and about 100 other publications or reports. His publications have more than 1,800 citations in the Science Citation Index. He was an associate editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* and *Pattern Recognition*. He was the president of the Pattern Recognition Society of Finland from 1989 to 1992. From 1989 to 2007, he 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 regularly serves on program committees of the top conferences and workshops of his field. Recently, he was an area chair of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '07), a cochair of the Workshops of the International Conference on Pattern Recognition (ICPR '08), a cochair of the ECCV 2008 Workshop on Machine Learning for Vision-based Motion Analysis (MLVMA), and a cochair of the MLVMA workshop at ICCV 2009. He is a senior member of the IEEE, a member of the IEEE Computer Society, and was the vice chair of the IEEE Finland Section.



**Xilin Chen** received the BS, MS, and PhD degrees in computer science from Harbin Institute of Technology, China, in 1988, 1991, and 1994, respectively. He was a professor with Harbin Institute of Technology from 1999 to 2005. He was a visiting scholar with Carnegie Mellon University, Pittsburgh, Pennsylvania, from 2001 to 2004. He has been a professor with the Institute of Computing Technology, Chinese Academy of Sciences (CAS), Beijing,

since August 2004. He is the director of the Key Laboratory of Intelligent Information Processing, CAS. He has published one book and more than 150 papers in refereed journals and conference proceedings in the areas of computer vision, pattern recognition, image processing, and multimodal interfaces. He is an associate editor of the *IEEE Transactions on Image Processing* and an area editor of the *Journal of Computer Science and Technology*. He has served as a program committee member for more than 30 international conferences. He has received several awards, including the China's State Scientific and Technological Progress Award in 2000, 2003, and 2005 for his research work, and he is a senior member of the IEEE and a member of the IEEE Computer Society.



**Wen Gao** received the PhD degree in electronics engineering from the University of Tokyo, Japan. He is a professor of computer science at Peking University, China, and the chief scientist at China Unioncom Labs. Before joining Peking University, he was a full professor of computer science at Harbin Institute of Technology from 1991 to 1995 and, at the Chinese Academy of Sciences (CAS) from 1996 to 2005, he served as a professor, the managing director of the

Institute of Computing Technology of CAS, the executive vice president of the Graduate School of CAS, and a vice president of the University of Science and Technology of China. He is an associate editor of the *IEEE Transactions on Circuits and Systems for Video Technology*, an associate editor of the *IEEE Transactions on Multimedia*, an associate editor of the *IEEE Transactions on Autonomous Mental Development*, an area editor of the *EURASIP Journal of Image Communications*, and an editor of the *Journal of Visual Communication and Image Representation*. He chaired a number of prestigious international conferences on multimedia and video signal processing, and also served on the advisory and technical committees of numerous professional organizations. He has published extensively, including four books and more than 500 technical articles in refereed journals and conference proceedings in the areas of image processing, video coding and communication, pattern recognition, multimedia information retrieval, multimodal interface, and bioinformatics. He is a fellow of the IEEE.

► For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).