

Simplified Gabor wavelets for human face recognition

Wing-Pong Choi, Siu-Hong Tse, Kwok-Wai Wong, Kin-Man Lam*

Centre for Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

Received 14 November 2006; received in revised form 31 May 2007; accepted 23 July 2007

Abstract

Gabor wavelets (GWs) are commonly used for extracting local features for various applications such as object detection, recognition and tracking. However, extracting Gabor features is computationally intensive, so the features are impractical for real-time applications. In this paper, we propose a simplified version of Gabor wavelets (SGWs) and an efficient algorithm for extracting the features based on an integral image. We evaluate the performance of the SGW features for face recognition. Experimental results show that using SGWs can achieve a performance level similar to using GWs, while the runtime for feature extraction using SGWs is, at most, 4.39 times faster than that of GWs implemented by using the fast Fourier transform (FFT).

© 2007 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.

Keywords: Gabor wavelets; Simplified Gabor wavelets; Feature extraction

1. Introduction

The Gabor wavelet (GW) [1,2] is well known for its effectiveness as a feature for image processing and pattern recognition. Its kernels are similar to the response of the two-dimensional receptive field profiles of the mammalian simple cortical cell [3], and exhibit the desirable characteristics of capturing salient visual properties such as spatial localization, orientation selectivity, and spatial frequency selectivity [4]. In the spatial domain, a GW is a complex exponential modulated by a Gaussian function, which is defined as follows [5]:

$$\psi_{\omega,\theta}(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x \cos \theta + y \sin \theta)^2 + (-x \sin \theta + y \cos \theta)^2}{2\sigma^2}} \cdot \left[e^{i(\omega x \cos \theta + \omega y \sin \theta)} - e^{-\frac{\omega\sigma^2}{2}} \right], \quad (1)$$

where x, y denote the pixel position in the spatial domain, ω is the radial center frequency of the complex exponential, θ is the orientation of the GW, and σ is the standard deviation of the Gaussian function. By selecting different center frequencies

and orientations, we can obtain a family of Gabor kernels from Eq. (1), which can then be used to extract features from an image.

GWs can effectively abstract local and discriminating features. In textural analysis [6,7] and image segmentation [8], GW features have achieved outstanding results, while in machine vision, they are found to be effective in object detection [9,10], recognition [7,10,11] and tracking [12–14]. The most successful application of the GWs is for face recognition. In Refs. [15–19], GWs are employed for face recognition, and achieve very high performance levels. As the dimension of the feature vectors using GWs is very large, linear subspace methods such as PCA and LDA are used to reduce the dimension. To further improve the performance, kernel methods are also used with the Gabor features. The improvement of both the linear methods and the kernel methods is due to the fact that the GW features are robust to illumination, rotation, and scale [2].

In spite of its superior performance, extracting GW features is highly computational. Given an image $f(x, y)$, GW features are extracted by convolving $f(x, y)$ with $\psi_{\omega,\theta}(x, y)$ as follows:

$$Y_{\omega,\theta}(x, y) = f(x, y) * \psi_{\omega,\theta}(x, y), \quad (2)$$

where $*$ denotes the convolution operator. Usually, convolution is implemented by the fast Fourier transform (FFT) to reduce the computation required for feature extraction. However,

* Corresponding author. Tel.: 852 27666207; fax: +852 23628439.

E-mail addresses: enwp.choi@polyu.edu.hk (W.-P. Choi), enshtse@polyu.edu.hk (S.-H. Tse), enkwwong@polyu.edu.hk (K.-W. Wong), enkmlam@polyu.edu.hk (K.-M. Lam).

the computation required is still very intensive; this, in turn, creates a bottleneck for real-time processing. Hence, an efficient method for extracting Gabor features is important for many practical applications.

The main contribution of this paper is to propose a simplified version of Gabor wavelets, whose features can be computed efficiently and can achieve a similar performance level for face recognition. These simplified Gabor wavelets (SGWs) can be viewed as an approximation of the original Gabor wavelets (GWs). An SGW is generated by quantizing a corresponding GW into a certain number of levels. With SGWs, features can be computed efficiently using an integral image. Our proposed SGWs can replace the GWs for the purpose of real-time processing and applications. The rest of this paper will describe the structure and the properties of SGWs. Fast algorithms for extracting features by using SGWs will be described, and their corresponding computational complexity will be analyzed. Finally, we will compare the performances of the SGW features and the GW features for face recognition, and discuss the discriminative power of these features.

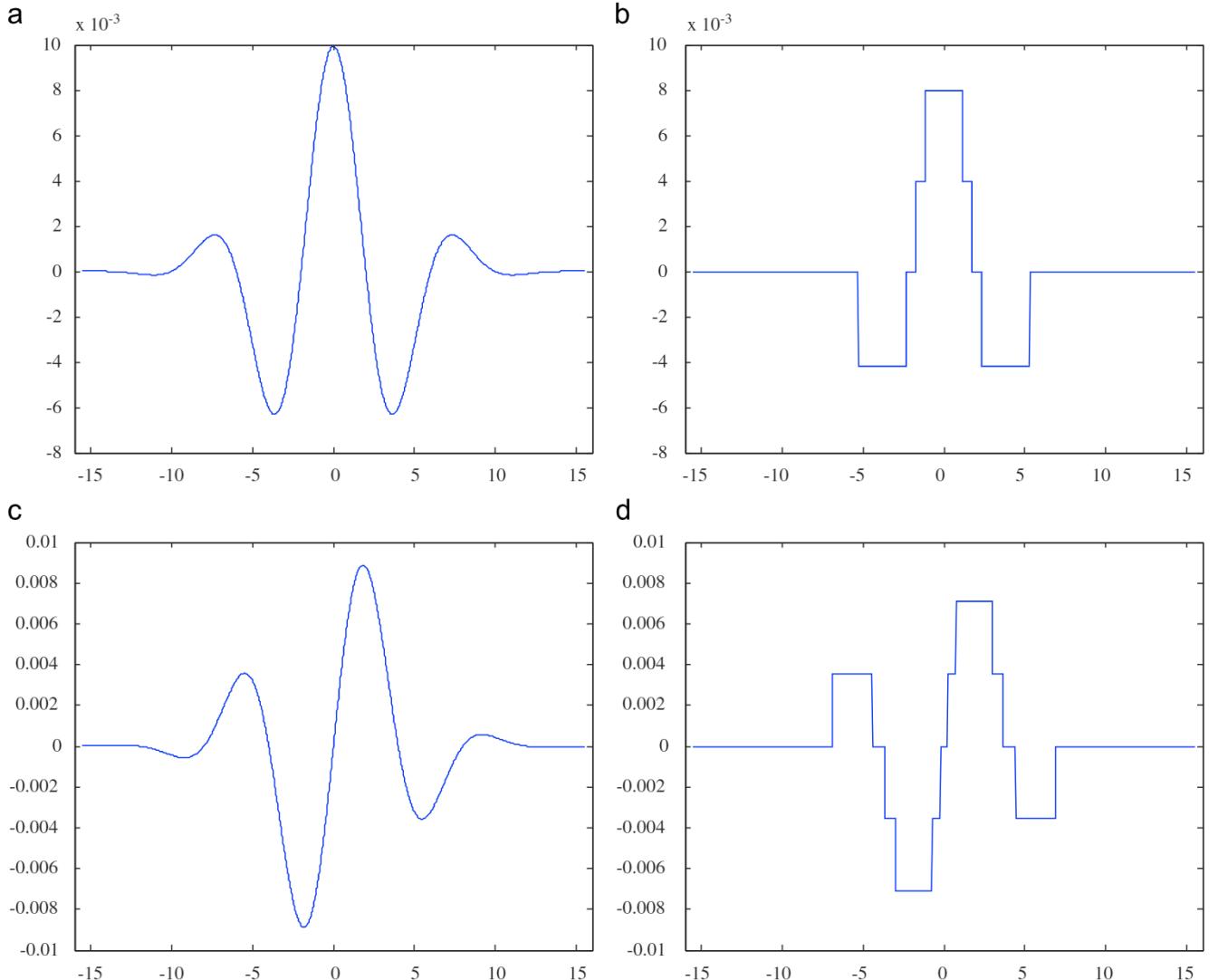


Fig. 1. (a) The real part of a one-dimensional GW; (b) the simplified version of (a); (c) the imaginary part of the wavelet; and (d) the simplified version of (c).

2. Simplified Gabor wavelets

In this section, we will describe the structure of our proposed SGW. This includes the shape of the SGW, the number of quantization levels, and the methods which determine the respective quantization values.

2.1. Shape of an SGW

To simplify our discussion, a one-dimensional GW is first considered, whose equation is shown as follows:

$$\psi_{\omega,0}(x) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right) \exp(i\omega x), \quad (3)$$

where the term $\exp(-\omega\sigma^2/2)$ in Eq. (1) is ignored. Fig. 1(a) shows the real part of this GW, whose values are continuous. To simplify the GW, its values are quantized to a certain number of levels. Fig. 1(b) illustrates a quantized SGW with 2 quantization levels for the positive values and 1 quantization level for

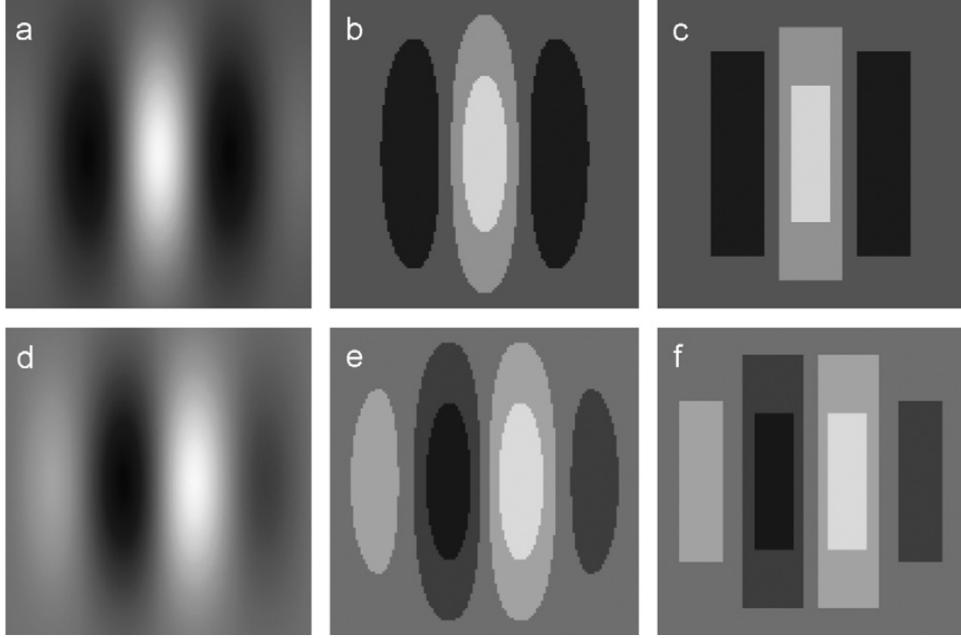


Fig. 2. (a) The real part of a two-dimensional GW; (b) the contours of the quantized GW of (a); (c) the approximation of the contours in (b) by rectangles; (d) the imaginary part of the two-dimensional GW; (e) the contours of the quantized GW of (d); and (f) the approximation of the contours in (e) by rectangles.

the negative values. Including a level of zero value, the wavelet is said to be quantized into 4 levels. Figs. 1(c) and (d) illustrate the corresponding imaginary part of the GW and its simplified version, respectively. The same number of quantization levels is used for the positive and the negative values of the wavelet, because their magnitudes are the same. In Fig. 1(d), the total number of quantization levels used is 5. For two-dimensional cases, Figs. 2(a) and (d) show the real and imaginary parts of the original two-dimensional GWs with the gray-level intensities representing the magnitudes of the wavelet. The contours of $\Psi_{\omega,\theta}(x, y)$ whose values equal those quantization levels in Figs. 1(b) and (d) are illustrated in Figs. 2(b) and (e), respectively. In SGWs, the contours are approximated by rectangles. We have derived two approximation methods for forming the rectangles, as shown in Figs. 3(a) and (b), respectively. The first method is to use a rectangle of a size just large enough to contain the corresponding contour of the quantized GW. The second method is to choose a rectangle such that the squared error between the elliptical contour of the GW and the corresponding rectangle is a minimum. To simplify the approximation, we adopt the first method in our algorithm. Figs. 2(c) and (f) illustrate the corresponding quantized GWs in Figs. 2(b) and (e), respectively, approximated by rectangles.

2.2. Number of quantization levels

The number of rectangles in an SGW depends on the number of quantization levels used to quantize the GW. If more quantization levels are employed, the SGWs will be more similar to the original GW, but more computation will then be involved for feature extraction. In other words, there is a trade-off between computation and approximation accuracy. In Section

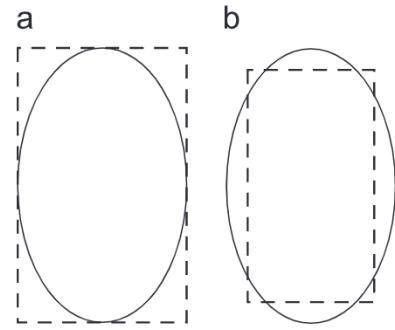


Fig. 3. (a) Approximation of an elliptical contour using a rectangle just large enough to enclose it; and (b) approximation of the elliptical contour using a rectangle such that the squared error between the rectangle and the contour is a minimum.

4, the computational analysis of using SGWs and GWs for feature extraction will be performed, and the experiments to evaluate the relative performances of SGWs and GWs with different numbers of quantization levels for face recognition will be conducted in Section 5.

2.3. Determination of quantization levels

In this paper, we describe two methods for determining the quantization levels to be used in constructing the SGWs. One of the quantization levels of the SGW is set to zero. Assume that the number of quantization levels for the positive and negative values are n_p and n_n , respectively. Then, the total number of quantization levels is $n_p + n_n + 1$.

Uniform quantization: In this method, the positive and negative parts of a GW are quantized uniformly according to

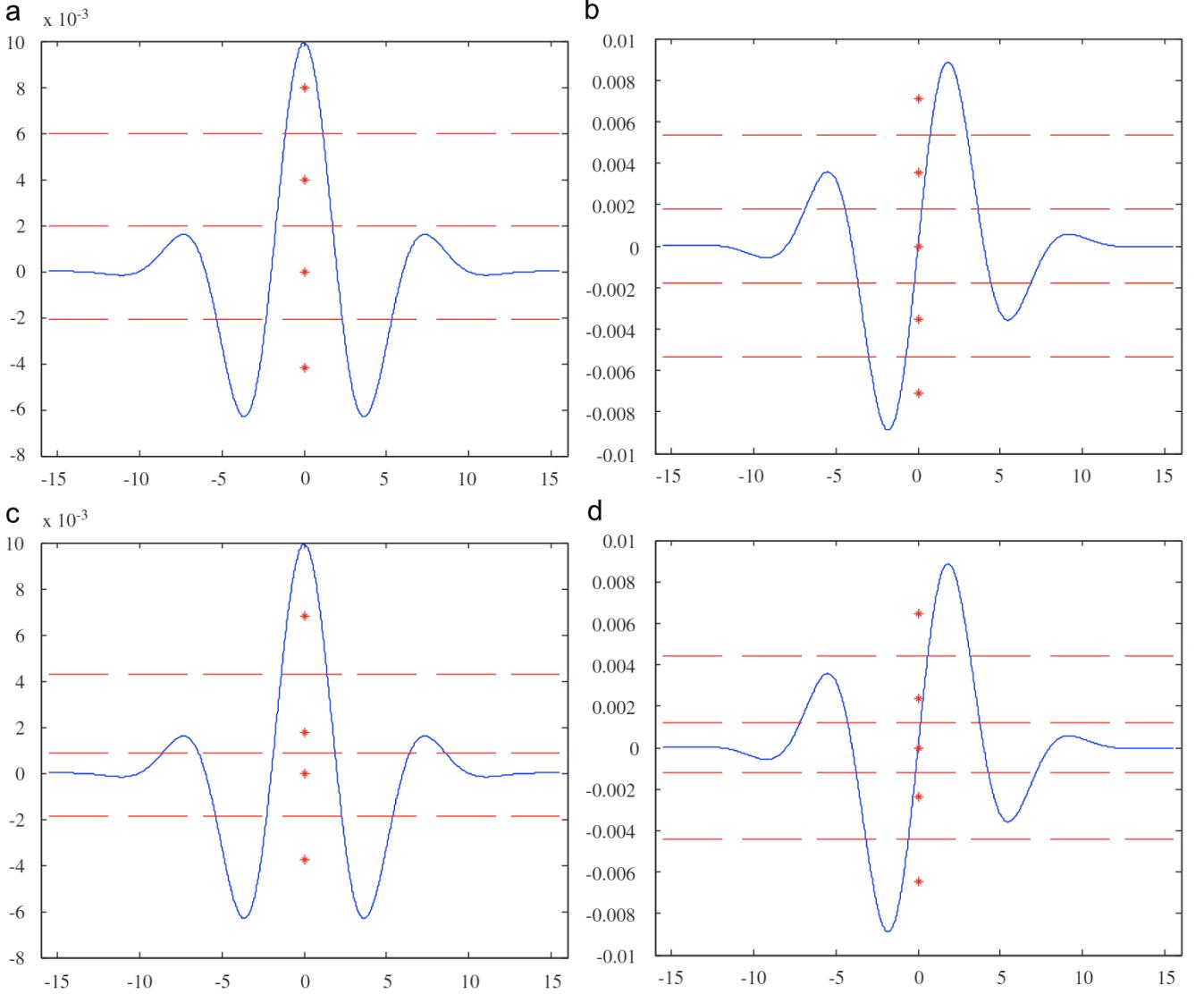


Fig. 4. (a) The quantization levels for the real part of a GW based on uniform quantization with $n_p = 2$ and $n_n = 1$, (b) the quantization levels for the imaginary part of the GW based on uniform quantization with $n_p = 2$ and $n_n = 2$, (c) the quantization levels for the real part of the GW based on k -means clustering with $n_p = 2$ and $n_n = 1$, and (d) the quantization levels for the imaginary part of the GW based on k -means clustering with $n_p = 2$ and $n_n = 2$.

the corresponding number of levels, as shown in Figs. 4(a) and (b). Suppose the most positive and negative values of a GW are A_+ and A_- , respectively, the corresponding quantization levels for positive levels $q_+(k)$ and negative levels $q_-(k)$ are as follows:

$$q_+(k) = \frac{A_+}{2n_p + 1} \cdot 2k \quad \text{where } k = 1, \dots, n_p,$$

and $q_-(k) = \frac{A_-}{2n_n + 1} \cdot 2k \quad \text{where } k = 1, \dots, n_n.$ (4)

k-means clustering: As the GWs are not evenly distributed, so the k -means algorithm is used to determine the respective optimal quantization levels. The positive values and the negative values are sampled, and are then partitioned into $n_p + 1$ and $n_n + 1$ clusters, respectively. However, after each iteration, the cluster whose centroid is the closest to zero will be set at zero.

Fig. 5 illustrates the real part and the imaginary part of a GW and their corresponding simplified versions. These SGWs are then convolved with an image to extract the SGW features at different center frequencies and orientations, which then form a simplified Gabor jet.

2.4. Demeaned SGW (DMSGW)

The term $e^{-\frac{\omega\sigma^2}{2}}$ in Eq. (1) makes the GW have a zero mean. An SGW formed by quantizing a GW has a non-zero mean; this makes the SGW features sensitive to the lighting conditions of an image. Hence, each of the SGWs has to be demeaned. The mean of an SGW is computed by summing all of its values, and then dividing this sum by the size of the filter. A demeaned simplified Gabor wavelet (DMSGW) is obtained by subtracting the SGW from its mean value. In the rest of this paper, we

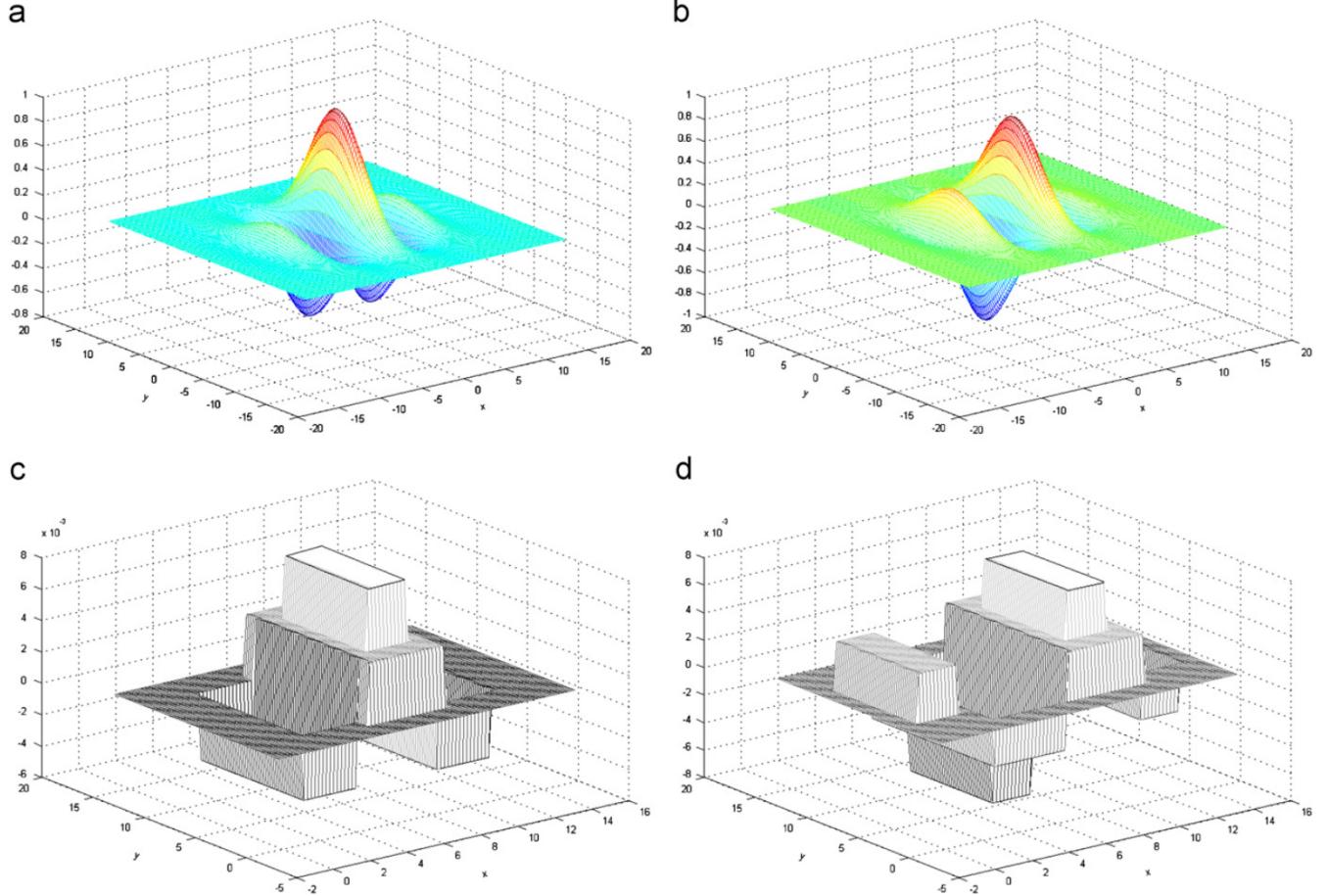


Fig. 5. The three-dimensional structures of (a) the real part and (b) the imaginary part of a two-dimensional GW, and (c) the real part and (d) the imaginary part of the corresponding SGW.

will use SGW to refer to a demeaned SGW, and the mean of an SGW is denoted as q_m . The next section will describe an efficient algorithm for computing the SGW features using our proposed SGWs.

3. Fast algorithm for feature extraction

The feature extraction process with an SGW is far more efficient than that with a GW. This section will, firstly, describe the extraction of GW features using the FFT, and then devise the fast algorithms for extracting features using the SGWs. The computational complexities of using the GW and the proposed SGW for different orientations will be analyzed in Section 4, and their respective runtimes will be measured in Section 5. In addition to requiring less computation, the SGW features for any pixel position can be extracted. This is particularly an advantage if the SGW features are used for object tracking. To use the FFT, the size of the image must be a power of 2.

3.1. Feature extraction using the original GWs

By selecting different center frequencies and orientations, we can obtain a family of GW kernels from Eq. (1), which can be used for extracting features from images. Given a gray-level image $f(x, y)$, the convolution of $f(x, y)$ and $\psi_{\omega, \theta}(x, y)$ is

given by Eq. (2). The convolution can be computed efficiently by performing the FFT, then point-by-point multiplications, and finally the inverse FFT (IFFT). By concatenating the convolution output, we can obtain a GW feature vector $\mathbf{Y}_{\omega, \theta}$ of dimension $N_w \cdot N_H$:

$$\mathbf{Y}_{\omega, \theta} = [\mathbf{Y}_{\omega, \theta}(0, 0), \mathbf{Y}_{\omega, \theta}(0, 1), \dots, \mathbf{Y}_{\omega, \theta}(0, N_H - 1), \\ \mathbf{Y}_{\omega, \theta}(1, 0), \dots, \mathbf{Y}_{\omega, \theta}(N_w - 1, N_H - 1)]^T, \quad (5)$$

where T represents the transpose operation, and N_w and N_H are the width and height of the image, respectively. In this paper, we consider only the magnitude of the GW representations, which can provide a measure of the local properties of an image [20] and is less sensitive to the lighting conditions [21] (for convenience, we denote it as $\mathbf{Y}_{\omega, \theta}$). $\mathbf{Y}_{\omega, \theta}$ is normalized to have zero mean and unit variance distribution; and then the Gabor representations with different ω and θ are concatenated to form a high-dimensional vector, as shown in Eq. (6), and are used for face recognition,

$$\mathbf{Y} = [\mathbf{Y}_{\omega_1, \theta_1}^T \ \mathbf{Y}_{\omega_1, \theta_2}^T \ \dots \ \mathbf{Y}_{\omega_1, \theta_n}^T \ \mathbf{Y}_{\omega_2, \theta_1}^T \ \dots \ \mathbf{Y}_{\omega_l, \theta_n}^T]^T, \quad (6)$$

where l and n are the number of center frequencies and the number of orientations used, respectively. Although the FFT is employed so as to reduce the computational complexity, it is still very computationally intensive because a total of $l \times n$

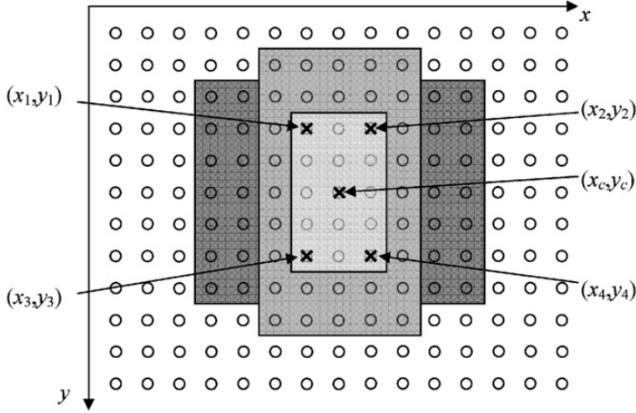


Fig. 6. Image $f(x, y)$ is convolved with an SGW whose center is shifted to the pixel position (x_c, y_c) .

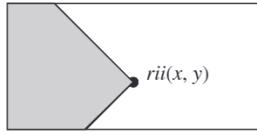


Fig. 7. Rotated integral image $rii(x, y)$, which is equal to the sum of pixel intensities inside the shaded and rotated rectangle.

GWs are involved. In addition, the size of the image must be a power of 2, so that the FFT can be used to implement the convolution for saving the computation.

3.2. Fast algorithms for feature extraction based on SGWs

In this section, we will present fast algorithms for feature extraction with the SGW at different orientations. Consider an SGW that is convolved with an image $f(x, y)$, and the SGW is shifted to the pixel position (x_c, y_c) , as shown in Fig. 6. The convolution output at this point is given as follows:

$$Y(x_c, y_c) = \sum_{k=1}^{NR_p} q_+(k)S_+(k) + \sum_{k=1}^{NR_n} q_-(k)S_-(k) + q_m S_F, \quad (7)$$

where $S_+(k)$, $S_-(k)$ and S_F are the sum of the gray-level intensities of those pixels covered by the rectangles with quantization values $q_+(k)$, $q_-(k)$, and the rectangular region of the filter, respectively. NR_p and NR_n are the numbers of rectangles with positive quantization values and negative quantization values, respectively. As an example in Fig. 2(c), $n_p = 2$ and $n_n = 1$, then $NR_p = 2$ and $NR_n = 2$.

$S_+(k)$, $S_-(k)$ and S_F are computed based on the idea of an integral image [22], which can calculate the sum of pixel values within a rectangle efficiently. In addition, a fast algorithm for rectangles rotated by 45° or 135° is also available [23]. Consequently, our SGW considers four orientations only, which are 0° , 45° , 90° , and 135° . Denote $ii(x, y)$ as the integral image, then its value at location (x, y) is the sum of the pixel values above and to the left of (x, y) inclusive, i.e.

$$ii(x, y) = \sum_{x' \leqslant x, y' \leqslant y} f(x', y'). \quad (8)$$

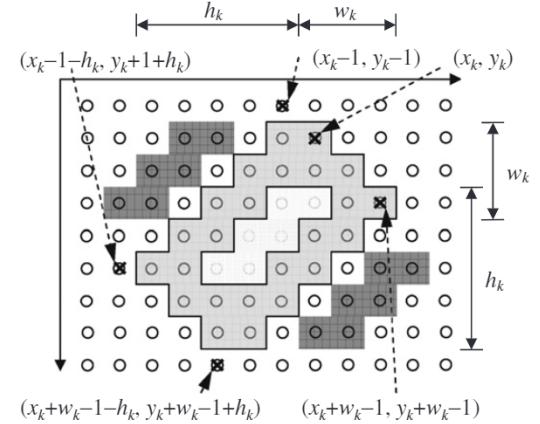


Fig. 8. The computation scheme for a rotated rectangle.

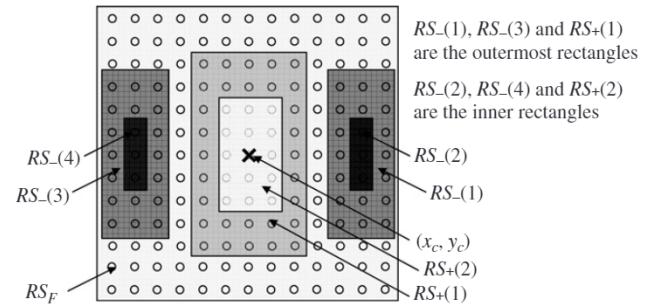


Fig. 9. The rectangles in an SGW.

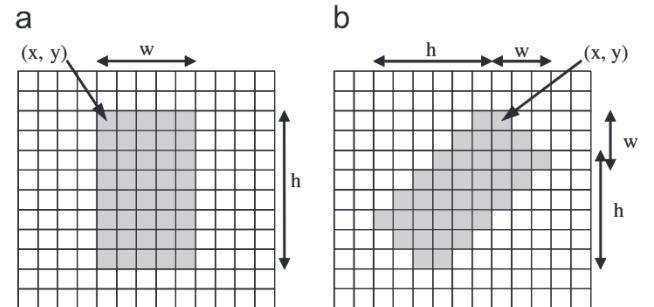


Fig. 10. Definition of the (x, y) -coordinates, width and height of a rectangle in an SGW at an orientation of (a) 0° , and (b) 45° .

The following pair of recursive equations is used to compute the integral image in one pass over the image:

$$\begin{aligned} s(x, y) &= s(x, y - 1) + f(x, y) \quad \text{and} \\ ii(x, y) &= ii(x - 1, y) + s(x, y), \end{aligned} \quad (9)$$

where $s(x, -1) = ii(-1, y) = 0$. Let us denote (x_k^1, y_k^1) , (x_k^2, y_k^2) , (x_k^3, y_k^3) , and (x_k^4, y_k^4) as the respective coordinates of the four corners of the rectangle for the k th quantization level. Fig. 6 shows the four corners for $k = n_p$. Hence, we have

$$S_+(k) = \begin{cases} ii(x_{n_p}^4, y_{n_p}^4) + ii(x_{n_p}^1 - 1, y_{n_p}^1 - 1) \\ - ii(x_{n_p}^2, y_{n_p}^2 - 1) - ii(x_{n_p}^3 - 1, y_{n_p}^3), & k = n_p, \\ ii(x_k^4, y_k^4) + ii(x_k^1 - 1, y_k^1 - 1) - ii(x_k^2, y_k^2 - 1) \\ - ii(x_k^3 - 1, y_k^3) - S_+(k + 1), & k < n_p. \end{cases} \quad (10)$$

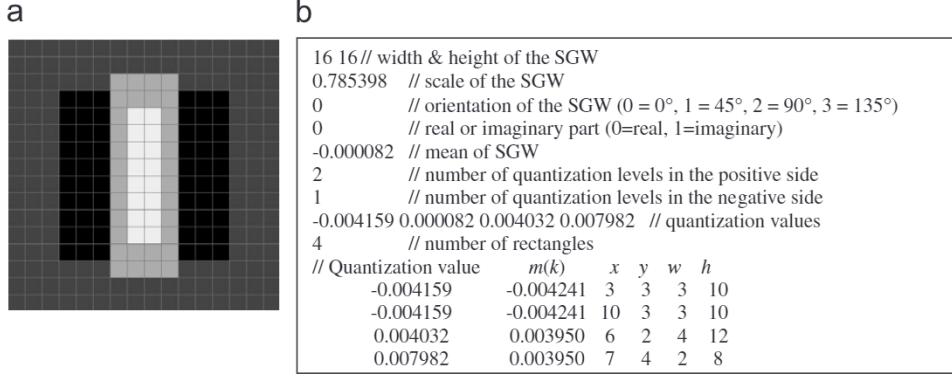


Fig. 11. (a) An SGW and (b) the corresponding parameters of this wavelet.

Similarly, if $k = n_n$, we have

$$S_-(k) = \begin{cases} ii(x_{n_n}^4, y_{n_n}^4) + ii(x_{n_n}^1 - 1, y_{n_n}^1 - 1) \\ -ii(x_{n_n}^2, y_{n_n}^2 - 1) - ii(x_{n_n}^3 - 1, y_{n_n}^3), & k = n_n, \\ ii(x_k^4, y_k^4) + ii(x_k^1 - 1, y_k^1 - 1) - ii(x_k^2, y_k^2 - 1) \\ -ii(x_k^3 - 1, y_k^3) - S_-(k+1), & k < n_n. \end{cases} \quad (11)$$

For a rectangle at an orientation of 45°, the rotated integral image, $rii(x, y)$ at location (x, y) contains the sum of the pixel values of the rectangle rotated by 45°, with the rightmost corner at (x, y) and extended to the boundaries of the image, as shown in Fig. 7, i.e.

$$rii(x, y) = \sum_{x' \leq x, x' \leq x - |y - y'|} f(x', y'). \quad (12)$$

Two passes over an image are required to compute the rotated integral image. The first pass is performed from left to right and top to bottom as follows:

$$rii(x, y) = rii(x - 1, y - 1) + rii(x - 1, y) + f(x, y) - rii(x - 2, y - 1), \quad (13)$$

where $rii(x, -1) = rii(-1, y) = rii(-2, y) = 0$. The second pass is performed from right to left and bottom to top as follows:

$$rii(x, y) = rii(x, y) + rii(x - 1, y + 1) - rii(x - 2, y). \quad (14)$$

Let us denote (x_k, y_k, w_k, h_k) as the x -coordinate, y -coordinate, width, and height, respectively, of the rotated rectangle in Fig. 8. Then, we have

$$S_+(k) = \begin{cases} rii(x_{n_p} + w_{n_p} - 1, y_{n_p} + w_{n_p} - 1) \\ + rii(x_{n_p} - h_{n_p} - 1, y_{n_p} + h_{n_p} - 1) \\ - rii(x_{n_p} - 1, y_{n_p} - 1) \\ - rii(x_{n_p} + w_{n_p} - h_{n_p} - 1, y_{n_p} + w_{n_p} \\ + h_{n_p} - 1), & k = n_p, \\ rii(x_k + w_k - 1, y_k + w_k - 1) \\ + rii(x_k - h_k - 1, y_k + h_k - 1) \\ - rii(x_k - 1, y_k - 1) - rii(x_k + w_k \\ - h_k - 1, y_k + w_k + h_k - 1) \\ - S_+(k+1), & k < n_p. \end{cases} \quad (15)$$

Similar formulation can be derived for the computation of $S_-(k)$, as well as for the case when a rectangle is at an orientation of 135°.

To further speed up feature extraction, let us denote $RS(k)$ as the sum of pixel intensities inside a rectangle with the coordinates of its four corners being (x_k^1, y_k^1) , (x_k^2, y_k^2) , (x_k^3, y_k^3) , and (x_k^4, y_k^4) , respectively. Thus

$$RS(k) = ii(x_k^4, y_k^4) + ii(x_k^1 - 1, y_k^1 - 1) - ii(x_k^2, y_k^2 - 1) - ii(x_k^3 - 1, y_k^3). \quad (16)$$

Let $RS_+(k)$, $RS_-(k)$ and RS_F be the sum of the gray-level intensities of those pixels inside the rectangles with quantization values $q_+(k)$, $q_-(k)$ and the rectangular region covered by the SGW, respectively. Fig. 9 shows the real part of an SGW with $n_n = n_p = 2$ or $NR_n = 4$ and $NR_p = 2$. Then, the convolution output at the pixel position (x_c, y_c) is:

$$\begin{aligned} Y(x_c, y_c) &= \sum_{k=1}^{NR_n} [q_-(k) \cdot S_-(k)] + \sum_{k=1}^{NR_p} [q_+(k) \cdot S_+(k)] + q_m \cdot S_F \\ &= q_{-(1)} \cdot S_{-(1)} + q_{-(2)} \cdot S_{-(2)} + q_{-(3)} \cdot S_{-(3)} \\ &\quad + q_{-(4)} \cdot S_{-(4)} + q_+(1) \cdot S_+(1) \\ &\quad + q_+(2) \cdot S_+(2) + q_m \cdot S_F \\ &= q_{-(1)} \cdot [RS_{-(1)} - RS_{-(2)}] + q_{-(2)} \cdot RS_{-(2)} \\ &\quad + q_{-(3)} \cdot [RS_{-(3)} - RS_{-(4)}] + q_{-(4)} \cdot RS_{-(4)} \\ &\quad + q_+(1) \cdot [RS_+(1) - RS_+(2)] + q_+(2) \cdot RS_+(2) \\ &\quad + q_m \cdot (RS_F - RS_{-(1)} - RS_{-(3)} - RS_+(1)) \\ &= [q_{-(1)} - q_m] \cdot RS_{-(1)} + [q_{-(2)} - q_{-(1)}] \cdot RS_{-(2)} \\ &\quad + [q_{-(3)} - q_m] \cdot RS_{-(3)} + [q_{-(4)} \\ &\quad - q_{-(3)}] \cdot RS_{-(4)} + [q_+(1) - q_m] \cdot RS_+(1) \\ &\quad + [q_+(2) - q_+(1)] \cdot RS_+(2) \\ &\quad + q_m \cdot RS_F \\ &= \sum_{k=1}^{NR_n} [m_-(k) \cdot RS_-(k)] + \sum_{k=1}^{NR_p} [m_+(k) \cdot RS_+(k)] \\ &\quad + m_F \cdot RS_F \end{aligned} \quad (17)$$

where

$$m_+(k) = \begin{cases} q_+(k) - q_m & k, \text{ refer to the outermost rectangles} \\ q_+(k) - q_+(k-1) & k, \text{ refer to the inner rectangles.} \end{cases}$$

$$m_-(k) = \begin{cases} q_{-(k)} - q_m & k, \text{ refer to the outermost rectangles} \\ q_{-(k)} - q_{-(k-1)} & k, \text{ refer to the inner rectangles.} \end{cases}$$

$$m_F = q_m.$$

Hence, instead of using $q(k)$ directly, the $m(k)$ s are employed in the computation.

For implementation, a number of parameters are required to describe a rectangle, which govern the computation of $RS_+(k)$, $RS_-(k)$ and RS_F . These parameters include the orientation, $m_+(k)$, $m_-(k)$, m_F , (x, y) coordinates, and the width and height of each rectangle. Fig. 10 defines the (x, y) coordinates, and the width and height of an upright rectangle and a rotated rectangle, which is similar to that in Ref. [22] and [23]. Fig. 11(a) shows an SGW, while Fig. 11(b) describes the parameters of this SGW.

4. Computational analysis for feature extraction

In this section, we will analyze and compare the computations required for extracting features using GW and SGW, respectively. Within our context, computations refer to the number of real additions and real multiplications required for extracting the GW features of an image using a GW. In our analysis, we assume that the image size is a power of 2 so that the FFT can be applied when using GWs for faster feature extraction. Actually, for the use of SGW, the image may be of any size and the features at any individual pixel position can be computed efficiently.

4.1. Feature extraction with GW

Given an $N \times N$ image, f , and a GW, g , with an arbitrary scale and orientation, GW features can be extracted by convolution, i.e. f^*g . The convolution is implemented by using the FFT, then point-by-point multiplications, and finally the IFFT. In our analysis, we assume that the FFTs of the GWs are pre-computed.

The FFT of an $N \times N$ image requires $N^2 \log_2 N^2$ complex additions and $0.5N^2 \log_2 N^2$ complex multiplications. The IFFT requires the same amount of computation as the FFT. The point-by-point multiplications involve N^2 complex multiplications. Performing one complex addition requires 2 real additions, while one complex multiplication requires 2 real additions and 4 real multiplications. Therefore, feature extraction based on a GW requires a total of $2N^2 \log_2 N^2$ complex additions and $N^2 \log_2 N^2 + N^2$ complex multiplications; this is equivalent to a total of $6N^2 \log_2 N^2 + 2N^2$ real additions and $4N^2 \log_2 N^2 + 4N^2$ real multiplications.

4.2. Feature extraction with SGW

As described in Section 3, fast algorithms are available for extracting SGW features using SGWs at 4 different orientations. These fast algorithms are based on the use of integral images and rotated integral images, such that features at any position in an image can be computed efficiently. Our algorithm will first perform a table look-up operation to compute the sum of pixel values for the respective rectangles of the SGW. Then, each of the pixel sums is multiplied by the quantization value of the corresponding rectangle. The sum of these products is the SGW feature at a given pixel position.

The computation for extracting features using an SGW at orientation 0° or 90° (a non-rotated SGW) is different from that when using an SGW at orientation 45° or 135° (a rotated SGW). This is because, for feature extraction, the non-rotated SGW uses the integral image, while the rotated SGW uses the rotated integral image. The computations involved are different for different orientations. Consequently, we separate our analysis into two parts: the non-rotated SGW (NR-SGW) and the rotated SGW (R-SGW).

4.2.1. The non-rotated SGW (NR-SGW)

Before extracting features using an NR-SGW, the integral image must be computed. From Eq. (9), 4 real additions are required to compute an entry of the integral image. For an image of size $N \times N$, $4N^2$ real additions are required for the whole integral image. Suppose that the SGW contains a total of N_{rect}^t rectangles. From Eqs. (16) and (17), $3N_{rect}^t$ real additions are required to compute all the rectangular pixel sums, and N_{rect}^t real multiplications and $(N_{rect}^t - 1)$ real additions are required to compute the SGW feature for a given pixel position. The coordinates of the four corners in Eq. (16) can be generated by a table look-up operation. Consequently, a total of $4N^2 N_{rect}^t + 3N^2$ real additions and $N^2 N_{rect}^t$ real multiplications are required to extract the SGW feature.

4.2.2. The rotated SGW (R-SGW)

The rotated integral image is computed for extracting feature with a rotated SGW. From Eqs. (13) and (14), 9 real additions are required to compute an entry in the rotated integral image. For an image of size $N \times N$, $9N^2$ real additions are required to compute the whole rotated integral image.

Feature extraction with an R-SGW is computed in a similar way to that with the NR-SGW. The rotated pixel sums covered by the rotated rectangles of the R-SGW are computed. Suppose that the R-SGW contains N_{rect}^t rectangles, then from Eqs. (16) and (17), $3N_{rect}^t$ real additions are required to compute all the rotated rectangular pixel sums, and N_{rect}^t real multiplications and $(N_{rect}^t - 1)$ real additions are required to compute the R-SGW feature at a pixel position. Therefore, a total of $4N^2 N_{rect}^t + 8N^2$ real additions and $N^2 N_{rect}^t$ real multiplications is required to extract the feature from the whole image. Table 1 shows the summarization of the computational complexities of feature extraction using GW and SGW.

To illustrate the computational advantage of using SGWs over GWs, Table 2 tabulates the respective numbers of arithmetic operations required for extracting GW features and SGW features, and Table 3 shows the respective numbers of rectangles used to represent the different level quantized SGWs. It is found that about 2.85 times and 2.44 times the arithmetic operations are saved if a 3-level quantized NR-SGW and R-SGW, respectively, are used. Moreover, the number of multiplications required for SGW feature extraction is reduced significantly when compared to that for GW. In general, the runtime required for multiplication is longer than that for addition. Furthermore, the runtime consumed by a floating point arithmetic operation is longer

Table 1

Computational complexities of feature extraction using GW and SGW

		+	×
GW	A: Compute FFT of image (floating point operations)	$3N^2 \log_2 N^2$	$2N^2 \log_2 N^2$
	B: Compute feature by multiplying FFT image and FFT GW (floating point operations)	$2N^2$	$4N^2$
	C: Compute IFFT of feature (floating point operations)	$3N^2 \log_2 N^2$	$2N^2 \log_2 N^2$
	Total	$6N^2 \log_2 N^2 + 2N^2$	$4N^2 \log_2 N^2 + 4N^2$
NR-SGW	D: Compute SAT (integer additions)	$4N^2$	0
	E: Compute rectangular pixel sums (integer additions)	$3N^2 N_{rect}^t$	0
	F: Compute feature by multiplying rectangular pixel sums and quantization value of rectangles (floating point multiplications)	0	$N_{rect}^t N^2$
	G: Add all products in F (floating point additions)	$N^2(N_{rect}^t - 1)$	0
R-SGW	Total	$4N^2 N_{rect}^t + 3N^2$	$N^2 N_{rect}^t$
	H: Compute RSAT (integer additions)	$9N^2$	0
	I: Compute rotated rectangular pixel sums (integer additions)	$3N^2(N_{rect}^t - 1)$	0
	J: Compute SGW background pixel sums (integer additions)	$3N^2$	0
R-SGW	K: Compute feature by multiplying rectangular pixel sums and quantization value of rectangles (floating point multiplications)	0	$N_{rect}^t N^2$
	L: Add all products in K (floating point additions)	$N^2(N_{rect}^t - 1)$	0
	Total	$8N^2 + 4N^2 N_{rect}^t$	$N_{rect}^t N^2$

^a Image dimension = $N \times N$, where N must be to the power of 2 in order to speed up the GW feature extraction process. ^b N_{rect}^t is the total number of rectangles in an SGW, which is listed in Table 3.

Table 2

Number of arithmetic operations required for extracting GW features from a 64×64 pixel image using a GW and an SGW with different numbers of quantization levels

	GW	+	×	Total
		303,104	212,992	516,096
NR-SGW	No. of quantization levels used	3 levels	147,844	32,768
		5 levels	229,764	53,248
		7 levels	344,452	81,920
R-SGW	No. of quantization levels used	3 levels	179,049	32,768
		5 levels	260,969	53,248
		7 levels	375,657	81,920

Table 3

The number of rectangles of an SGW with different numbers of quantization levels, where n_n and n_p are the number of negative quantization levels and the number of positive quantization levels in an SGW

Number of quantization levels ($n_n + n_p + 1$)	Number of rectangles in the <i>real part</i> of an SGW (N_{rect}^r)	Number of rectangles in the <i>imaginary part</i> of an SGW (N_{rect}^i)	Total number of rectangles in an SGW, including the background of SGW ($N_{rect}^r + N_{rect}^i + 1$)
$3(n_n = 1, n_p = 1)$	$(N_{rect}^r) = (n_n \times 2 + n_p) = 3$	$(N_{rect}^i) = ((n_n + 1) + (n_p + 1)) = 4$	8
$5(n_n = 2, n_p = 2)$	$(N_{rect}^r) = (n_n \times 2 + n_p) = 6$	$(N_{rect}^i) = ((n_n + 1) + (n_p + 1)) = 6$	13
$7(n_n = 3, n_p = 3)$	$(N_{rect}^r) = (n_n \times 2 + (n_p + 2)) = 11$	$(N_{rect}^i) = ((n_n + 1) + (n_p + 1)) = 8$	20

than that for an integer arithmetic operation. Feature extraction with SGW involves fewer floating point operations than does GW, therefore, the runtime for SGW feature extraction should in practice have a speed-up rate higher than 2.85 times.

5. Experimental results

In this section, we will evaluate the respective performances of the proposed SGWs with different numbers of quantization levels. The two different methods for determining the

Table 4

The number of distinct subjects, the number of images and the characteristics of the face databases

Databases	Characteristics	Number of distinct subjects	Number of images	Number of images per subject
Yale	Variations in facial expression	15	150	10
YaleB	Large variations in lighting	10	640	64
AR	Variations in facial expression	121	605	5
	Overall	146	1395	

Table 5

Face recognition performances of SGW1, SGW2 and GW with different scales, orientations, and quantization levels (SGW1: uniformly quantized SGWs, SGW2: k -means quantized SGWs, GW: Gabor wavelets)

	Different combinations of scales-orientations-quantization levels	Recognition rate		
		Yale(%)	YaleB(%)	AR(%)
SGW1	5 scales 4 orientations 3 quantization levels	82.00	90.16	92.40
	5 scales 4 orientations 5 quantization levels	84.67	92.19	92.40
	5 scales 4 orientations 7 quantization levels	82.67	92.66	92.89
	4 scales 4 orientations 3 quantization levels	82.67	93.13	92.07
	4 scales 4 orientations 5 quantization levels	82.00	94.69	91.74
	4 scales 4 orientations 7 quantization levels	82.67	94.84	92.07
	3 scales 4 orientations 3 quantization levels	82.67	92.97	92.23
	3 scales 4 orientations 5 quantization levels	82.67	93.91	92.23
	3 scales 4 orientations 7 quantization levels	83.33	94.69	92.23
SGW2	5 scales 4 orientations 3 quantization levels	82.67	91.09	91.90
	5 scales 4 orientations 5 quantization levels	82.67	92.50	92.23
	5 scales 4 orientations 7 quantization levels	82.67	92.50	92.56
	4 scales 4 orientations 3 quantization levels	82.67	93.91	91.74
	4 scales 4 orientations 5 quantization levels	82.67	95.00	91.74
	4 scales 4 orientations 7 quantization levels	83.33	95.47	91.90
	3 scales 4 orientations 3 quantization levels	82.00	93.59	92.40
	3 scales 4 orientations 5 quantization levels	82.67	95.00	91.90
	3 scales 4 orientations 7 quantization levels	83.33	94.53	92.23
GW	5 scales 4 orientations	80.00	94.69	92.73
	4 scales 4 orientations	78.00	97.50	92.23
	3 scales 4 orientations	74.00	99.22	89.92

quantization values of an SGW will also be evaluated. Then, we will compare the performances of the SGW features and the GW features for face recognition. Finally, we will compare the runtimes for extracting the SGW features and the GW features.

5.1. Face databases and experimental set-up

The standard face databases used include the Yale database, YaleB database and AR database. The number of distinct subjects, the number of testing images and the characteristics of the databases are tabulated in Table 4.

For face recognition, a frontal-view image of each subject in the databases is selected as a training image, and the remaining faces are used for testing. Each face image is normalized to a size of 64×64 , and is aligned based on the position of the two eyes for matching. In order to enhance the global contrast of the images and reduce the effect of uneven illuminations, histogram equalization is applied to all images. As described in Section 2.3, we have two different ways to determine the quantization levels of SGWs. The SGWs derived based on uniform

quantization and on k -means clustering are denoted as SGW1 and SGW2, respectively. The GW and SGW adopt 3–5 center frequencies with 4 orientations. In other words, 12–20 GWs and SGWs are used for feature extraction. The extracted features with each Gabor filter are concatenated to form a feature vector, which is then normalized to have zero mean and unit variance. These Gabor jets are then used directly to compute the distance between two images, pixel position by pixel position.

5.2. Relative performances of SGW1 and SGW2

Table 5 shows the recognition rates based on SGW1 and SGW2 with different numbers of quantization levels for the different databases. For the real part of a GW, the dynamic range of the positive values is usually larger than that of the negative values. Hence, n_p should be set larger than n_n . However, for the imaginary part of the GW, the dynamic ranges of the positive values and negative values are the same, so n_p should be equal to n_n . To simplify the experiment, we set n_p equal to n_n for both the real and imaginary parts. Consequently, including the

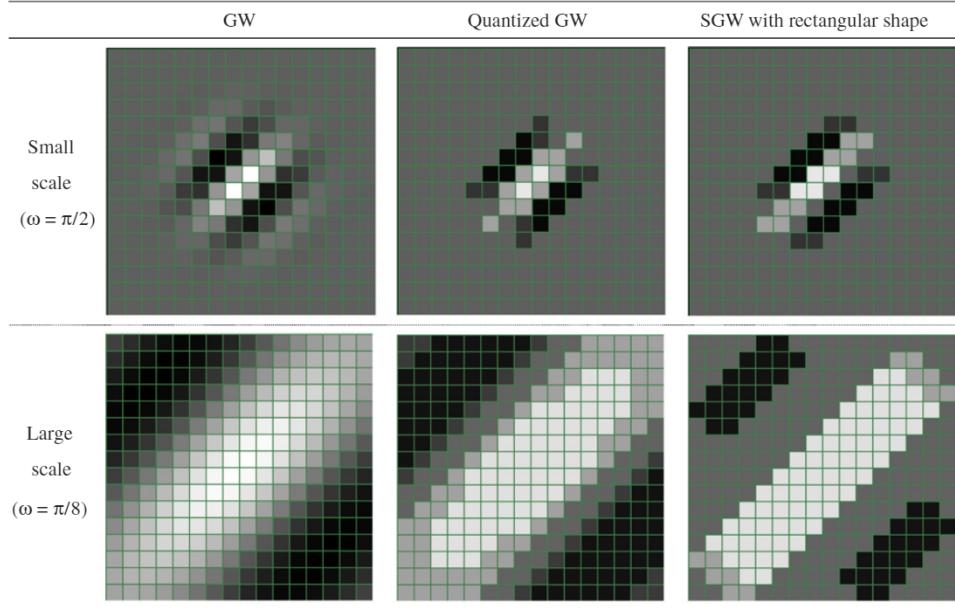


Fig. 12. The first column is the GW, the second column is the quantized form of GW, and the third column is the SGW with a rectangular shape. The top row is the small-scale ($\omega = \pi/2$) GW being quantized and formed into a rectangular-shaped SGW. The bottom row is the large-scale ($\omega = \pi/8$) GW being quantized and formed into a rectangular-shaped SGW.

level for zero, the numbers of quantization levels considered in the experiments are 3, 5, and 7.

From Table 5, the relative performances of SGW1 and SGW2 are very similar. The face recognition rate increases slightly with an increase in the number of quantization levels. If more quantization levels are used, the SGW can better approximate the GW, and its performance will then be closer to that of the GW. However, using the SGW with more quantization levels will involve more computations.

We have also investigated the effect of using more scales of the SGW with a fixed number of quantization levels. Experimental results show that using 4 scales of SGW results in the best recognition rate. Theoretically, using 5 scales should produce a better performance than using 4 scales only. However, the error in representing a GW is large when its scale is large. As discussed in Sections 2.1 and 3, in order to utilize fast algorithms to extract the features, the SGWs must be approximated with rectangles after quantizing the GWs. This constraint will alter the effective regions of the SGWs. Fig. 12 shows a GW, a GW after quantization, and an SGW approximated by rectangles. We can observe that part of the effective regions of the quantized GW is removed or extended in order to form a rectangular shape, which will therefore introduce quantization errors. As the size of an SGW is 16×16 pixels only, large rectangles cannot be formed. As a result, the quantization errors in forming the rectangles are significant for those large-scale SGWs. On the contrary, for small-scale SGWs, small rectangles will be formed without requiring much of the original shape of the quantized GW to be changed. This will introduce fewer quantization errors. For SGWs with 5 scales, the approximation of some of the large-scale GWs is not accurate. This, in turn, will degrade the overall recognition performance.

5.3. Performances of the SGW and the GW

The use of the SGW can save a lot of computation when compared to the GW, while maintaining a comparable performance to the GW. Table 5 tabulates the performances using SGW1, SGW2 and GW for face recognition with different numbers of center frequencies and orientations. The face recognition results show that, with the same number of center frequencies and orientations, the relative performances of the SGW and the GW are very similar; and in some cases, the SGW outperforms the GW. Actually, the center frequency of an SGW should be very similar to its original GW. An SGW is a quantized version of its GW; their rates of variation should be maintained. Hence, in the frequency domain, the center frequencies of the SGW and the GW should be very close, while the shape of their spectra will differ. The features extracted by a GW and the corresponding SGW should be similar. Fig. 13 shows the magnitudes of the GW features and the SGW features at 3 scales and 4 orientations. We can observe that the general shapes of SGW features and GW features are similar; however, SGWs introduce a directional pattern on the features, which is a drawback with quantizing GWs coefficients to a certain number of levels.

From Table 5, the performance of the SGW is slightly worse than that of the GW with the YaleB database, while the SGW has a very similar performance to the GW with the other databases. The reason for this is that the images in the YaleB database have a wide variation in lighting conditions. As we discussed in Section 2.4, an SGW is the quantized version of a GW, so the values of the SGWs are changed in step. Therefore, when two images of the same person have a significant difference in lighting conditions, the features extracted by GWs and SGWs

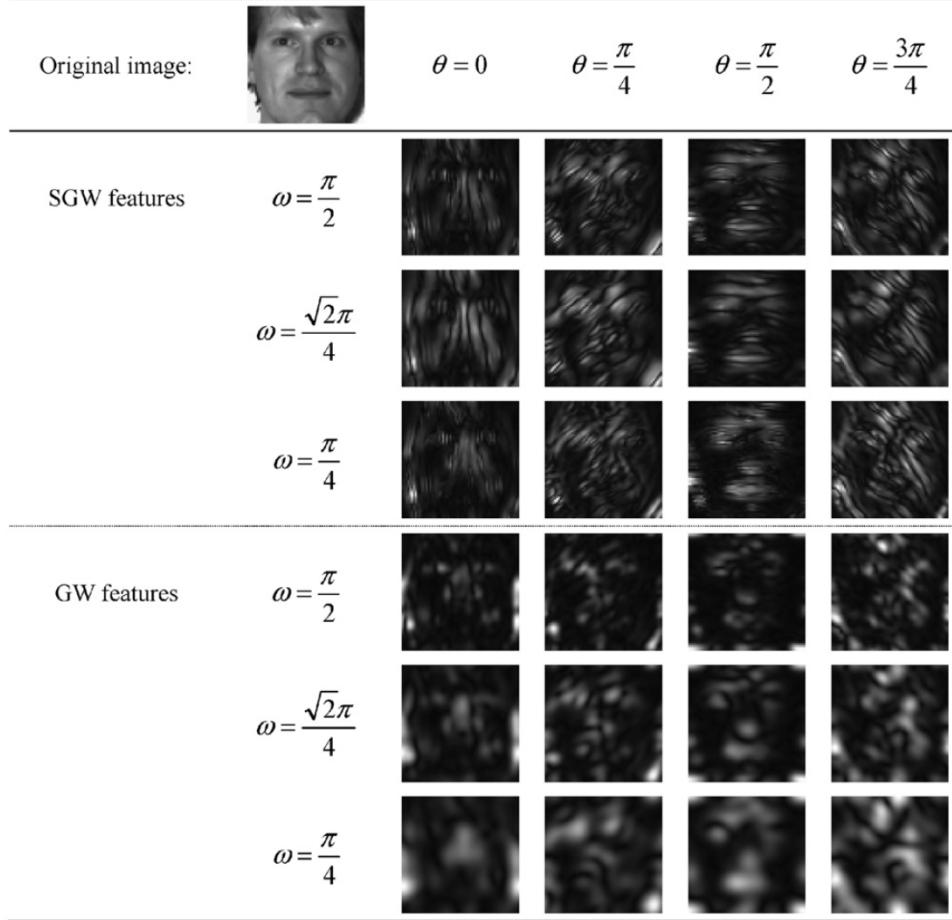


Fig. 13. The magnitudes of SGW features and GW features at 3 scales and 4 orientations.

Table 6

The average runtimes for feature extraction using GW and SGW with different scales, orientations, and numbers of quantization levels

SGW	5 scales, 4 orientations			4 scales, 4 orientations			3 scales, 4 orientations		
	3-Lv	5-Lv	7-Lv	3-Lv	5-Lv	7-Lv	3-Lv	5-Lv	7-Lv
	16.09 ms	27.50 ms	37.97 ms	12.81 ms	22.50 ms	30.94 ms	9.37 ms	17.50 ms	23.44 ms
GW			70.64 ms			56.73 ms			42.67 ms
Speed-up rate	4.39	2.57	1.86	4.43	2.52	1.83	4.55	2.44	1.82

The speed-up rate is equal to the runtime required by GW divided by that of the SGW.

will also differ greatly. Hence, the performance of the SGW will be degraded in this circumstance.

5.4. Runtimes for feature extraction with the SGW and the GW

In our experiments, we also measure the runtimes required for feature extraction using the SGW and the GW. One of the images from the Yale database was used, and the size of each face region is 64×64 pixels. Feature extractions using the SGW and the GW at 5 scales and 4 orientations were performed for 100 times, and the respective total runtimes were measured. Table 6 tabulates the runtimes for extracting features using the

SGW and the GW. With a 3-level quantized SGW, the speed-up rate for feature extraction is 4.39 times that of a GW. The reduction in runtime will decrease if the SGW uses more quantization levels. For SGWs with 5 and 7 quantization levels, the runtimes for feature extraction are 27.5 and 37.97 ms, respectively, and the corresponding speed-up rates are 2.57 and 1.86, respectively.

To conclude our experiment results, the performance of the SGW is comparable to that of the GW, while the computation required by the SGW is significantly less than that for the GW. GWs can extract features which are discriminative and useful for many applications, but they are impractical for real-time

applications due to their high complexity in feature extraction. Consequently, SGWs can be propelled to replace GWs for real-time applications and processing.

6. Conclusion

In this paper, we have proposed a simplified version of GWs, which can achieve a performance level similar to the original GWs for face recognition. We have also described fast algorithms for feature extraction based on SGWs at different orientations. In addition, we have presented how to construct these SGWs and their performance with different numbers of quantization levels, center frequencies and orientations. When 5 center frequencies and 4 orientations are employed, the relative performances of the SGWs and the GWs are very similar, while, at most, a speed-up rate of 4.39 times can be achieved if 3-level quantized SGWs are used. The runtimes required for feature extraction in a 64×64 image, based on an SGW with 3 quantization levels and a GW, are 16.09 and 70.64 ms, respectively. These results can propel SGWs to replace GWs for realizing real-time applications and processing. However, the simplified Gabor features are slightly more sensitive to lighting variations than the original Gabor features are.

Acknowledgment

The work described in this paper was supported by a grant from the Research Grants Council of the HKSAR, China (Project No. PolyU 5220/03E).

References

- [1] T.S. Lee, Image Representation Using 2D Gabor Wavelets, *IEEE Trans. Pattern Anal. Mach. Intell.* 18 (10) (1996) 959–971.
- [2] J.K. Kamarainen, V. Kyriki, H. Kalviainen, Invariance properties of Gabor filter-based features—overview and applications, *IEEE Trans. on Image Process.* 15 (5) (2006) 1088–1099.
- [3] C.K. Chui, An Introduction to Wavelets, Academic Press, Boston, 1992.
- [4] C. Liu, H. Wechsler, Independent component analysis of Gabor features for face recognition, *IEEE Trans. Neural Networks* 14 (4) (2003) 919–928.
- [5] D. Liu, K.M. Lam, L.S. Shen, Optimal sampling of Gabor features for face recognition, *Pattern Recognition Lett.* 25 (2) (2004) 267–276.
- [6] A.C. Bovik, M. Clark, W.S. Geisler, Multichannel texture analysis using localized spatial filters, *IEEE Trans. Pattern Anal. Mach. Intell.* 12 (1) (1990) 55–73.
- [7] B.S. Manjunath, W.Y. Ma, Texture feature for browsing and retrieval of image data, *IEEE Trans. Pattern Anal. Mach. Intell.* 18 (8) (1996) 837–842.
- [8] Y. Chen, R.S. Wang, Texture segmentation using independent component analysis of Gabor features, in: 18th International Conference on Pattern Recognition, vol. 2, August 2006, pp. 20–24.
- [9] H. Cheng, N. Zheng, C. Sun, Boosted Gabor features applied to vehicle detection, in: 18th International Conference on Pattern Recognition, vol. 1, 2006, pp. 662–666.
- [10] M. Valstar, M. Pantic, Fully automatic facial action unit detection and temporal analysis, in: IEEE Conference on CVPRW, June 2006, p. 149.
- [11] C. Liu, H. Wechsler, Gabor feature based classification using the enhanced Fisher linear discriminant model for face recognition, *IEEE Trans. Image Process.* 11 (4) (2002) 467–476.
- [12] E. Painkras, C. Charoensak, A framework for the design and implementation of a dynamic face tracking system, in: IEEE Region 10 TENCON, November 2005, pp. 1–6.
- [13] S. Dubuisson, An adaptive clustering for multiple object tracking in sequences in and beyond the visible spectrum, in: IEEE Conference on CVPRW, June 2006, p. 142.
- [14] D. Tao, X. Li, S.J. Maybank, X. Wu, Human carrying status in visual surveillance, in: IEEE Computer Society Conference on CVPR, vol. 2, 2006, pp. 1670–1677.
- [15] L. Wiskott, J.M. Fellous, N. Krüger, C. Malsburg, Face recognition by elastic bunch graph matching, *IEEE Trans. Pattern Anal. Mach. Intell.* 19 (7) (1997) 775–779.
- [16] C. Liu, Gabor-based kernel PCA with fractional power polynomial models for face recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* 26 (5) (2004) 572–581.
- [17] C. Liu, Capitalize on dimensionality increasing techniques for improving face recognition grand challenge performance, *IEEE Trans. Pattern Anal. Mach. Intell.* 28 (5) (2006) 725–727.
- [18] X. Xie, K.M. Lam, Gabor-based kernel PCA with doubly nonlinear mapping for face recognition with a single face image, *IEEE Trans. on Image Process.* 15 (9) (2006) 2481–2492.
- [19] L. Shen, L. Bai, M. Fairhurst, Gabor wavelets and general discriminant analysis for face identification and verification, *Image Vision Comput.* 25 (5) (2007) 553–563.
- [20] M. Lades, J.C. Vorbruggen, J. Buhmann, J. Lange, C.V.D. Malsburg, R.P. Wurtz, W. Konen, Distortion invariant object recognition in the dynamic link architecture, *IEEE Trans. Comput.* 42 (3) (1993) 300–311.
- [21] L. Shams, C. Malsburg, The role of complex cells in object recognition, *Vision Res.* 42 (22) (2002) 2547–2554.
- [22] P. Viola, M.J. Jones, Rapid object detection using a boosted cascade of simple features, *IEEE CVPR*, 2001.
- [23] R. Lienhart, J. Maydt, An extended set of Haar-like features for rapid object detection, *Proceedings of IEEE ICIP*, 2002, pp. I-900–I-903.

About the Author—WING-PONG CHOI received his BEng degree in Electronic and Information Engineering from the Hong Kong Polytechnic University in 2003. He is currently pursuing his MPhil degree in the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University. His main research interests include human face detection, tracking and verification in image and video.

About the Author—SIU-HONG TSE received his BEng degree in Electronic and Information Engineering from The Hong Kong Polytechnic University in 2003. He is currently pursuing his MPhil degree in the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University. His main research interests include image processing and pattern recognition.

About the Author—KWOK-WAI WONG received his BEng degree and MPhil degree in Electronic and Information Engineering from The Hong Kong Polytechnic University, Hong Kong, in 1998 and 2001, respectively. He is currently working in the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University as a Research Associate. His research interests include human face detection, image and video processing, pattern recognition with applications in computer vision and inspection systems.

About the Author—KIN-MAN LAM received his Associateship in Electronic Engineering with distinction from The Hong Kong Polytechnic University (formerly called Hong Kong Polytechnic) in 1986. He won the S.L. Poa Scholarship for overseas studies and was awarded an MSc degree in communication engineering from the Department of Electrical Engineering, Imperial College of Science, Technology and Medicine, England, in 1987. In August 1993, he undertook a PhD degree program in the Department of Electrical Engineering at the University of Sydney, Australia, and won an Australia Postgraduate Award for his studies. He completed his PhD studies in August 1996, and was awarded the IBM Australia Research Student Project Prize.

From 1990 to 1993, he was a lecturer at the Department of Electronic Engineering of The Hong Kong Polytechnic University. He joined the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University again as an Assistant Professor in October 1996, and has been an Associate Professor since February 1999. Dr Lam has also been a member of the organizing committee and program committee of many international conferences. In particular, he was the Secretary of the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03), the Technical Chair of the 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing (ISIMP 2004), and a Technical Co-Chair of the 2005 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS 2005). In addition, Dr Lam was a Guest Editor for the Special Issue on Biometric Signal Processing, EURASIP Journal on Applied Signal Processing.

Currently, Dr Lam is the Chairman of the IEEE Hong Kong Chapter of Signal Processing and an Associate Editor of EURASIP Journal of Image and Video Processing. His current research interests include human face recognition, image and video processing, and computer vision.