Pergamon

Pattern Recognition, Vol. 27, No. 11, pp. 1439–1449, 1994
Elsevier Science Ltd

0031-3203/94 $7.00 + .00

0031-3203(94)E0022-D

IMAGE COMPRESSION BY MOMENT-PRESERVING EDGE DETECTION

SHYI-CHYI CHENG† and WEN-HSIANG TSAI‡§

† Institute of Computer Science and Information Engineering, and ‡ Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan 300, Republic of China

(*Received* 27 *January* 1992; *received for publication* 24 *February* 1994)

Abstract—A new image compression algorithm based on a new edge detection technique is proposed. A given image is compressed by dividing the image into non-overlapping square blocks and coding the edge information in each non-uniform block. The edge feature in a given window is detected by applying the moment-preserving principle to the image data. The edge directions are approximated by multiples of 45° to speed up the direction process without introducing obvious distortion. For a larger block size, a more accurate model of a two-step edge is employed in the detection process. The proposed algorithm offers excellent reconstructed image quality agreeing with human perception, high compression ratios, and greatly reduced coding complexity. The solution to the edge detection problem in a given block is also analytic. The algorithm can be performed very fast for real-time applications with no need for special hardware.

Image compression Gray moment Moment-preserving principle Edge detection
Human visual system

1. INTRODUCTION

The objective of an image compression technique is to remove as much redundant information as possible without destroying the image integrity. These two goals, however, are in general conflicting. The amount of information discarded can be easily stated in terms of compression ratios or bits per pixel (BPP); however, it is difficult to define a quantitative term to evaluate the quality of a coded image. Previous approaches have utilized error norms, such as the mean-squared error (MSE) and the signal to noise ratio (SNR), for this purpose. However, they do not meet well with the perceptual notion of visual fidelity. The measurement of the accuracy of coded images from a visual perspective has remained difficult, particularly at high compression ratios when much information is discarded.

To maintain good image fidelity in the sense of information theory limits the compression ratios to being in the order of about 10. To obtain higher compression ratios without increasing computational expenses, it is better to take into account the properties of the human visual system. Certain low-level aspects of vision can be accurately modeled. Some well-known psychophysical and physiological properties of the human visual system have been utilized in the design of certain coding algorithms. For example, it was found that the human eye assigns different attention to the low-frequency and high-frequency components of images and this fact leads to the development of the sub-band coding algorithms.[1,2]

Decomposition of images into low-frequency blocks and blocks containing visually important features (such as edges or lines) suggests visual continuity and visual discontinuity constraints. A block is visually continuous if the values of all the pixels in the block are almost the same. In contrast, if the variations of the pixel values in the block are noticeable, it is a visually discontinuous block. The mean of a visually continuous block is enough to represent the block. If a block is visually discontinuous and if a strong orientation is associated with it, then it should be coded as a kind of visually important feature.

Another biologically visual property helping to design an image compression algorithm is the spatial frequency constraint. Visual information is processed over specified frequency ranges, normally measured in cycles per degree of viewing angle (c/deg). The passband of the receptive fields of a normal human eye is in the range 3–6 c/deg. The response falls off quickly outside that range.[3] Spatial frequencies are also related to the amounts of visual response time. A higher spatial frequency requires a longer response time. The contribution of perception for a high spatial frequency block is thus small. Accordingly, it is accurate enough to model a visually discontinuous block of a small size as an edge.

In this paper, a new image compression algorithm is proposed based on the application of the moment-preserving technique to detect the visually important feature, namely, edge, in a given image block. Each given image is divided into non-overlapping square blocks and coded block by block. Edge features used to code an ordinary image, as found in this study,

§ Author to whom all correspondence should be addressed.

produce excellent image quality in accordance with human perception; and high image compression ratios comparable to or exceeding those offered by the vector quantization (VQ)[4] and block truncation coding (BTC)[5] algorithms can also be obtained. Furthermore, simple and analytical formulae to compute the parameters of the edge feature in an image block are also derived in this study. The computation is therefore very fast. This implies that the proposed algorithm can be employed for real-time applications.

The remainder of this paper is organized as follows. Section 2 includes an overview of the moment-preserving principle. Section 3 presents the use of the moment-preserving principle to detect edges. Experimental results are shown in Section 4 to illustrate the effectiveness of the proposed image coding algorithm. Finally, a conclusion is drawn in Section 5.

2. REVIEW OF MOMENT-PRESERVING PRINCIPLE

In digital images, a pixel represents the smallest image element and is usually regarded as a unit grid square. If f is a given image with N pixels whose gray value at (x, y) is denoted by $f(x, y)$, then the ith gray moment of f is defined as

$$m_i = \frac{1}{N} \sum_x \sum_y f^i(x, y), \quad i = 1, 2, \ldots, \tag{1}$$

or equivalently,

$$\begin{aligned} m_i &= \frac{1}{N} \sum_j N_j h_j^i, \\ &= \sum_j p_j h_j^i, \quad i = 1, 2, \ldots, \end{aligned} \tag{2}$$

where N_j is the total number of pixels in f with gray value h_j, and $p_j = N_j/N$. Let R be a region with certain characteristics in a gray-valued image, G be an operator which is applied to R, and S be a corresponding ideal output with the same characteristics as those of R. The gray moments of R and those of S can then be computed. The moment-preserving principle is to equate the ith gray moment of S with that of R for all $i = 0, 1, 2, \ldots, k$, where k is an integer. For example, if the first $2N$ gray moments $m_0, m_1, \ldots,$ and m_{2N-1} of R are preserved in S, then we can obtain the following equations:

$$\begin{aligned} &p_1 h_1^0 + p_2 h_2^0 + p_3 h_3^0 + \cdots + p_N h_N^0 = m_0, \\ &p_1 h_1^1 + p_2 h_2^1 + p_3 h_3^1 + \cdots + p_N h_N^1 = m_1, \\ &\qquad \vdots \\ &p_1 h_1^{2N-1} + p_2 h_2^{2N-1} + p_3 h_3^{2N-1} + \cdots + \\ &\quad p_N h_N^{2N-1} = m_{2N-1}, \end{aligned} \tag{3}$$

where the left-hand sides are the gray moments of S, and p_i is the fractions of the pixels with gray values h_i in S, $i = 1, 2, \ldots, N$. For $N \leq 4$, analytical solutions to the unknown parameters p_i and h_i have been derived by Tsai.[6]

The moment-preserving principle has been employed to perform many image processing works.[5–16] Among these, Delp and Mitchell[5] developed a block truncation (BTC) method for image compression. Tabatabai and Mitchell[7] developed a subpixel edge detector based on a thresholding method that preserves the first three gray moments. Tsai[6] generalized the moment-preserving concept and proposed a multilevel thresholding approach that can be employed to threshold a given image into N gray levels by preserving the first $2N - 1$ gray moments. Chen and Tsai applied further the principle to several image processing operations, such as sharpening, line detection, curve detection and curve edge detection.[8–10] Later, the moment-preserving principle was used to detect other features[11,12] in a given circular window. Studies employing the moment-preserving principle to detect features in a square window were also conducted.[13,14] Recently, Delp and Mitchell[15] applied the moment-preserving concept to design a generalized moment-preserving quantizer, and a neural network implementation of the general case of the moment-preserving technique was also proposed[16] for real-time consideration.

Most of the feature detectors mentioned above suffer from having either no analytical solution[8–11] or no general form solution.[13,21] To overcome these difficulties, a new moment-preserving edge detector is proposed in this paper. Its specific advantages include at least the following: (1) the solutions are analytic and easy to compute; (2) general form solutions for a square image block can be obtained; and (3) it can be generalized to detect other features (lines or curves), and analytical solutions are still possible.

3. PROPOSED MOMENT-PRESERVING EDGE DETECTION FOR IMAGE COMPRESSION

The proposed image compression algorithm basically partitions an image into a set of non-overlapping square blocks. Each block is coded as either a uniform block or an edge block. The edge in each block is detected by the proposed edge detection technique. The image can be reconstructed according to the parameters of these blocks.

3.1. *Proposed edge detector*

All existing moment-preserving edge detectors can detect an edge along any direction in a given image block (called a window henceforth). However, the possible orientations of an edge in an image block of a digital image are limited. For example, only eight possible directions of an edge in a 3×3 window can be identified, as shown in Fig. 1. Furthermore, the difference between two edges with very close directions is not perceivable when the image resolution is high and the window size is small. According to this fact, the orientation of an edge to be detected in a small window need not be any direction for some applications. In this paper, the window size 4×4 or 5×5 is adopted, which can be assumed to be small for high-resolution images. Instead of detecting edges in any direction, the

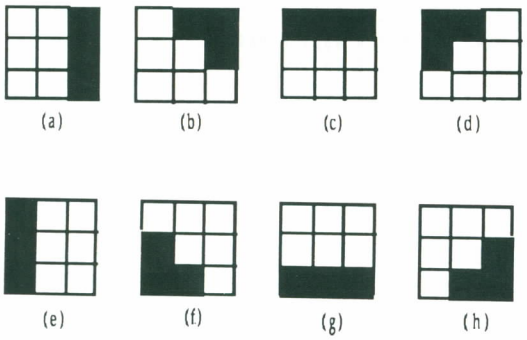


Fig. 1. Eight possible directions of an edge in a 3 × 3 window.

proposed moment-preserving edge detector assumes that the possible directions of an edge in a 4 × 4 or 5 × 5 window are limited to be multiples of 45°, or equivalently, $i \times 45°$, $i = 0, 1, \ldots, 7$. If the actual direction of an edge is not a multiple of 45°, it is quantized to be the nearest multiple of 45°. This means that if the direction of an edge is 35°, for example, it is assumed to be a 45° edge. Under this constraint, the difficulty of being unable to derive analytical and/or general form solutions of the moment-preserving feature detectors proposed before can be solved. For simplicity, the window size is assumed to be 4 × 4 in the subsequent discussion, unless specified otherwise.

A continuous two-dimensional edge model specified by four parameters, two gray values h_1 and h_2, an edge translation l, and an orientation angle θ, for an edge in a square window W shown in Fig. 2 is employed for edge detection in this approach. The edge is simply a step transition from gray value h_1 to gray value h_2. The edge translation l is defined as the length from the center of the edge model to the transition and is confined within the range of -1 to $+1$. The parameter θ specifies the direction of the edge and is confined within the range of 0 to 360 degrees. The value of θ can be computed individually first, as described next.

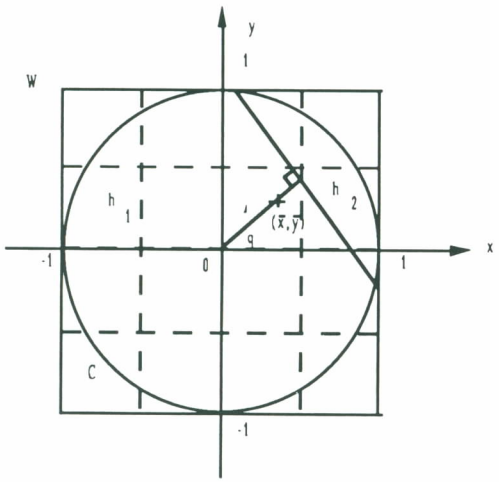


Fig. 2. An edge model in a 4 × 4 window W. The circle C is inscribed in W, and $(\bar{x}, \bar{y})$ are the coordinates of the centers of gravity of the gray values inside C.

The x-mass moment M_x and y-mass moment M_y defined within the cycle C inscribed in W are as follows:

$$M_x = \iint_C x f(x, y)\, dy\, dx, \tag{4}$$

$$M_y = \iint_C y f(x, y)\, dy\, dx, \tag{5}$$

where $f(x, y)$ is the gray level at (x, y). The value of θ can be easily obtained from the values of M_x and M_y. Let $(\bar{x}, \bar{y})$ be the coordinates of the center of gravity of the gray values inside C. Then,

$$\bar{x} = \frac{M_x}{M_0}, \tag{6}$$

$$\bar{y} = \frac{M_y}{M_0}, \tag{7}$$

where M_0 is the mean of C and is computed by

$$M_0 = \iint_C f(x, y)\, dy\, dx. \tag{8}$$

It has been found that the direction of the edge is perpendicular to the direction of the vector from the origin to $(\bar{x}, \bar{y})$.[14] So,

$$\sin\theta = \frac{\bar{y}}{\sqrt{\bar{x}^2 + \bar{y}^2}}, \tag{9}$$

$$\cos\theta = \frac{\bar{x}}{\sqrt{\bar{x}^2 + \bar{y}^2}}, \tag{10}$$

and angle θ can be calculated by

$$\theta = \tan^{-1}\left(\frac{\bar{y}}{\bar{x}}\right) = \tan^{-1}\left(\frac{M_y}{M_x}\right). \tag{11}$$

The moments M_0, M_x and M_y are also useful to judge whether C and therefore W is visually uniform. Notice that if C is uniform, $(\bar{x}, \bar{y})$ will be very close to the origin of C. So, if the distance between the center of gravity $(\bar{x}, \bar{y})$ and the origin of C is less than a threshold τ, then it is considered that there is no edge involved in C, and W is decided to be a uniform block. This criterion for judging the uniformity of a given window is actually equivalent to the following:

$$\sqrt{M_x^2 + M_y^2} < \tau M_0. \tag{12}$$

To compute each of the moments M_0, M_x and M_y, a more convenient way is to correlate the pixel gray values in the window with a mask. Each mask represents the value resulting from performing the integration (4), (5), or (8) over each pixel in C, assuming $f(x, y)$ is constant over that pixel. The circular limits are also included in the integrations. The resulting set of masks is shown in Fig. 3.

Once the value of θ is obtained, the solutions for h_1, h_2, and l can be found by applying the moment-preserving technique. As mentioned in the previous sections, the proposed moment-preserving edge detector only

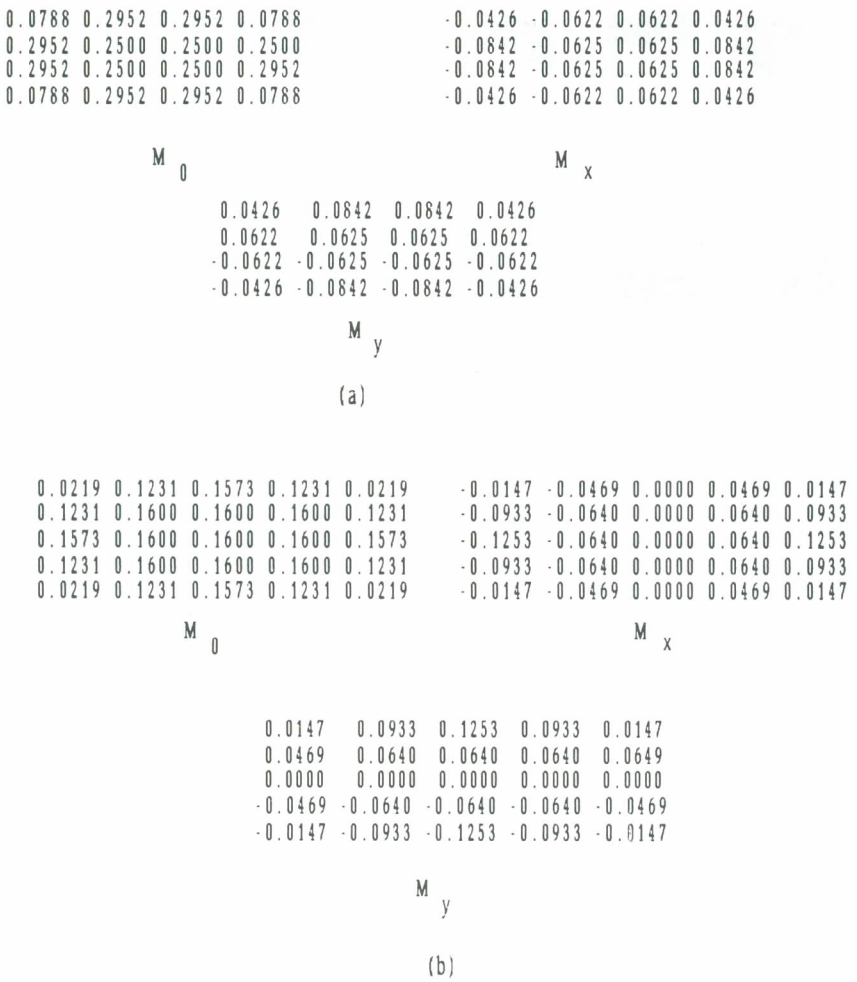


Fig. 3. The set of three masks for moment computation. (a) For window size 4 × 4, (b) for window size 5 × 5.

detects edges with directions $i \times 45°$, where $i = 0, 1, \ldots, 7$. If the value of θ is not exactly equal to one of the multiples of 45°, it is quantized to be the nearest one. For convenience, let the eight multiples of 45°, called the detectable directions of edges, be denoted as e_0 to e_7. These edges can be categorized further into two classes, namely, the even and odd classes. Each e_i with an odd (or even) value of i belongs to the odd (or even) class, and so the even class $= \{e_0, e_2, e_4, e_6\}$, and the odd class $= \{e_1, e_3, e_5, e_7\}$. A special window W', which is adapted from the given window W and is shown in Fig. 4(a), is employed to detect edges in the odd class, while edges in the even class are detected using the original window W. The gray values of the pixels of W' outside the original window W can be found in the original image. Notice that a rotation of the window W or W' by $-i \times 45°$ with respect to the origin will align the orientation of the detected edge e_i along the x-axis, as shown in Fig. 4(b). For convenience, let the rotated window be denoted by R. In this study, the solutions of h_1, h_2 and l can be obtained by preserving the first three moments of the input window in the output of the edge detector in the following manner:

$$m_k = p_1 h_1^k + p_2 h_2^k, \quad k = 1, 2, 3, \tag{13}$$

where $p_1 = A_1/A$, A_1 is the area of R covered by intensity h_1, A is the total area of R and is equal to 4, $p_2 = 1 - p_1$, $R = \{(x, y): |x| \le 1 \text{ and } |y| \le 1\}$, and m_k is the kth gray moment of the original window T ($T = W$ (or W') if the edge is in the even (or odd) class) and is computed as follows:

$$m_k = \frac{1}{A'} \int\int_T f^k(x, y)\, dy\, dx. \tag{14}$$

The value of A', which is the area of T, is also 4.

By assuming that $f(x, y)$ takes a constant value over each grid, the integral in (14) becomes a weighted sum of the gray values in window T and can be rewritten as

$$m_k = \sum_x \sum_y w_{xy} f^k(x, y), \quad k = 1, 2, 3, \tag{15}$$

where $f(x, y)$ is the gray value of (x, y), and w_{xy} is the weighting coefficient associated with (x, y) (if a grid is totally included in W, then $w_{xy} = 1/16$). The values of w_{xy} for W and W' are shown in Fig. 5. The unknown

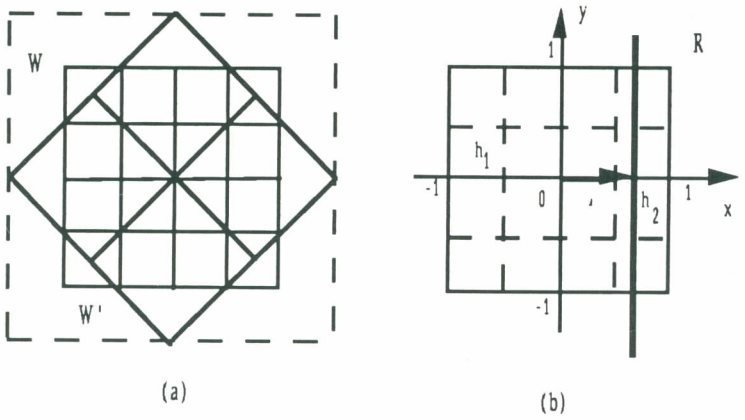
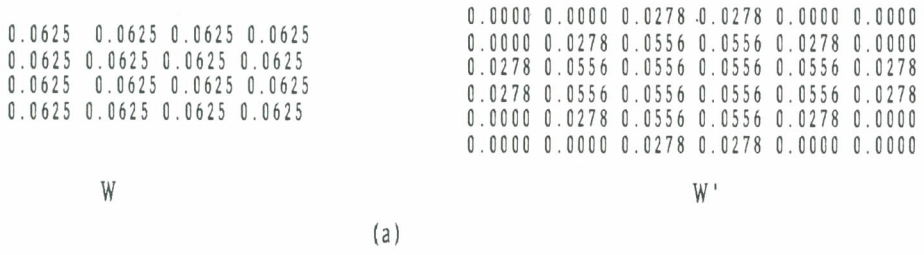


Fig. 4. Different rectangular-shaped windows for different edge classes. (a) The edges of the even class are detected in W and those of the odd class are detected in W'; (b) the appearance of the common edge model after rotating the window w or w' through $i \times 45°$ with respect to the origin.



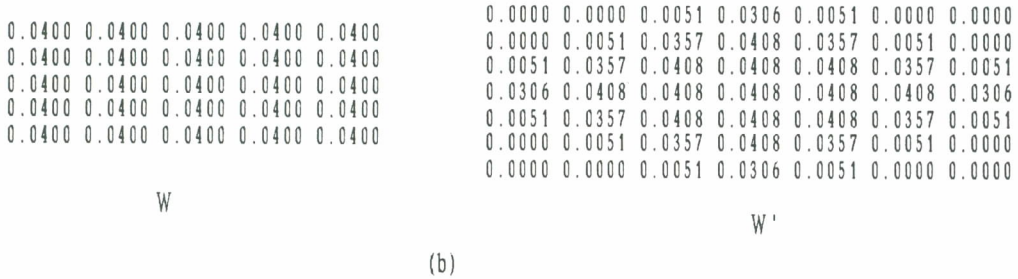


Fig. 5. The weighting coefficients w_{ij} for both types of window W and W'. (a) For window size 4×4; (b) for window size 5×5.

parameters h_1, h_2, p_1 and p_2 in equation (13) can be solved in closed form to be as follows:[6]

$$c_d = \begin{vmatrix} m_0 & m_1 \\ m_1 & m_2 \end{vmatrix}; \tag{16}$$

$$c_0 = \frac{1}{c_d}\begin{vmatrix} -m_2 & m_1 \\ -m_3 & m_2 \end{vmatrix}; \tag{17}$$

$$c_1 = \frac{1}{c_d}\begin{vmatrix} m_0 & -m_2 \\ m_1 & -m_3 \end{vmatrix}; \tag{18}$$

$$h_1 = \tfrac{1}{2}[-c_1 - \sqrt{c_1^2 - 4c_0}]; \tag{19}$$

$$h_2 = \tfrac{1}{2}[-c_1 + \sqrt{c_1^2 - 4c_0}]; \tag{20}$$

$$p_d = \begin{vmatrix} 1 & 1 \\ h_1 & h_2 \end{vmatrix}; \tag{21}$$

$$p_1 = \frac{1}{p_d}\begin{vmatrix} 1 & 1 \\ m_1 & h_2 \end{vmatrix}; \tag{22}$$

$$p_2 = 1 - p_1. \tag{23}$$

The remaining unknown parameter l is simple to calculate from the value of p_1 as follows. First,

$$p_1 = \frac{A_1}{A}, \tag{24}$$

where A_1 is the area covered by gray value h_1 in R and $A = 4$ is the area of R. Next, the value of A_1 can be computed by

$$A_1 = 2(1 + l), \tag{25}$$

as seen from Fig. 4(b). From equations (24) and (25),

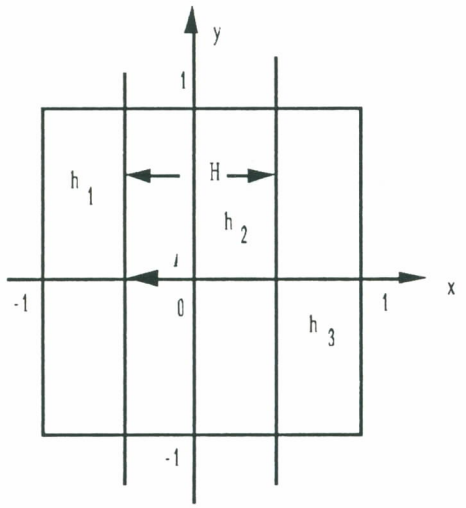


Fig. 6. A two-step edge model for a 5 × 5 window.

we get

$$l = 2p_1 - 1. \tag{26}$$

If the size of the window is changed to 5 × 5, the visually important feature of the window can be modeled as a more accurate two-step edge, as shown in Fig. 6. For this case, the number of unknown parameters is five, namely, three gray values h_1, h_2 and h_3, an edge translation l, and finally an edge transition width H. These values can be easily obtained by preserving the first five moments, that is,

$$\begin{aligned} p_1 + p_2 + p_3 &= 1; \\ p_1h_1 + p_2h_2 + p_3h_3 &= m_1; \\ p_1h_1^2 + p_2h_2^2 + p_3h_3^2 &= m_2; \\ p_1h_1^3 + p_2h_2^3 + p_3h_3^3 &= m_3; \\ p_1h_1^4 + p_2h_2^4 + p_3h_3^4 &= m_4; \\ p_1h_1^5 + p_2h_2^5 + p_3h_3^5 &= m_5. \end{aligned} \tag{27}$$

The above equations can also be solved in closed form as follows:[6]

$$c_d = \begin{vmatrix} m_0 & m_1 & m_2 \\ m_1 & m_2 & m_3 \\ m_2 & m_3 & m_4 \end{vmatrix}; \tag{28}$$

$$c_0 = \begin{vmatrix} 1 & -m_3 & m_1 & m_2 \\ c_d & -m_4 & m_2 & m_3 \\ & -m_5 & m_3 & m_4 \end{vmatrix}; \tag{29}$$

$$c_1 = \frac{1}{c_d} \begin{vmatrix} m_0 & -m_3 & m_2 \\ m_1 & -m_4 & m_3 \\ m_2 & -m_5 & m_4 \end{vmatrix}; \tag{30}$$

$$c_2 = \frac{1}{c_d} \begin{vmatrix} m_0 & m_1 & -m_3 \\ m_1 & m_2 & -m_4 \\ m_2 & m_3 & -m_5 \end{vmatrix}; \tag{31}$$

$$h_1 = -c_2/3 - A - B; \tag{32}$$

$$h_2 = -c_2/3 - W_1A - W_2B; \tag{33}$$

$$h_3 = -c_2/3 - W_2A - W_1B; \tag{34}$$

$$p_d = \begin{vmatrix} 1 & 1 & 1 \\ h_1 & h_2 & h_3 \\ h_1^2 & h_2^2 & h_3^2 \end{vmatrix}; \tag{35}$$

$$p_1 = \frac{1}{p_d} \begin{vmatrix} m_0 & 1 & 1 \\ m_1 & h_2 & h_3 \\ m_2 & h_2^2 & h_3^2 \end{vmatrix}; \tag{36}$$

$$p_2 = \frac{1}{p_d} \begin{vmatrix} 1 & m_0 & 1 \\ h_1 & m_1 & h_3 \\ h_1^2 & m_2 & h_3^2 \end{vmatrix}; \tag{37}$$

$$p_3 = 1 - p_1 - p_2, \tag{38}$$

where A, B, W_1 and W_2 are as follows:

$$\begin{aligned} A = \{&(c_0/2 - c_1c_2/6 + c_2^3/27) \\ &- [(c_0/2 - c_1c_2/6 \\ &+ c_2^3/27)^2 + (c_1/3 - c_2^2/9)^3]^{1/2}\}^{1/3}; \\ B &= -(c_1/3 - c_2^2/9)/A; \\ W_1 &= -1/2 + i(\sqrt{3}/2); \\ W_2 &= -1/2 - i(\sqrt{3}/2); \\ i &= \sqrt{-1}. \end{aligned}$$

The two unknown parameters l and H of the edge model shown in Fig. 6 can be computed as follows:

$$H = 2p_2, \quad l = 2p_1 - 1. \tag{39}$$

An algorithm is given below to summarize the proposed moment-preserving edge detector.

Algorithm 1. Proposed moment-preserving edge detection technique.

Input. Square window R containing an edge feature.

Output. Parameters of the edge.

Method.

1. Compute the values of M_x, M_y and M_0 from the circle inscribed inside R.
2. Calculate the value of θ using equation (11). Quantize the value of θ to be the nearest detectable direction, which is one of the multiples of 45°.
3. Apply the moment-preserving technique to find the solutions of the parameters of the edge. These values can be computed according to equations (19), (20) and (26) for window size 4 × 4, or equations (32), (33), (34) and (39) for window size 5 × 5.

3.2. *Application to image compression*

A goal of the proposed moment-preserving edge detector is to meet the perception of an observer so that it can be used to compress an image with good performance. An algorithm using the proposed moment-preserving edge detector is given below for image compression.

Algorithm 2. Image compression using proposed moment-preserving edge detector.

Input. An image I.

Output. Parameters of coded image blocks.

Method.

1. Partition I into a set of $n \times n$ non-overlapping windows where $n = 4$ or $n = 5$, and for each window R, perform the following steps.
2. Calculate the values of M_x, M_y and M_0 from the circle inscribed inside R.
3. Test the uniformity of R according to equation (12). If R is a uniform block, code it as the mean of R; otherwise, continue.
4. Perform the proposed moment-preserving edge detector (Algorithm 1) to find the solutions of the parameters of the edge model in R.
5. Code the edge as the parameter sets $\{h_1, h_2 - h_1, l, \theta\}$ or $\{h_1, h_2 - h_1, h_3 - h_2, H, l, \theta\}$ for $n = 4$ or $n = 5$, respectively.

The proposed compression scheme classifies windows as uniform blocks or edge blocks, so it needs a bit to represent the type of a block. Although the parameters to represent an edge block are real numbers, they can be quantized into integers without sacrificing the coded image quality. In this paper, the number of bits assigned to code a block is shown in Table 1, in which the total number of bits used to code a uniform block is denoted as B_{uniform} and that of an edge block as B_{edge}. Let N_{uniform} and N_{edge} denote the numbers of uniform blocks and edge blocks, respectively. Then, the compression ratio (CR) can be computed by

Table 1. Parameters used in the proposed compression scheme

Type	Parameter	Number of bits per block: Window size = 5	Number of bits per block: Window size = 4
Uniform block	Block type	1	1
	Mean intensity	6	6
	B_{uniform}	7	7
Edge block	Block type	1	1
	h_1	4	4
	$h_2 - h_1$	3	3
	$h_3 - h_2$	3	—
	$l + \theta$	—	4
	$l + H + \theta$	4	—
	B_{edge}	15	12

$$\text{CR} = \frac{512 \times 512 \times 8}{N_{\text{uniform}} \times B_{\text{uniform}} + N_{\text{edge}} \times B_{\text{edge}}} \tag{40}$$

if the size of a tested image is 512×512 with 8 bits for each pixel.

Image reconstruction is trivial using the proposed compression scheme since the coded image blocks are represented either as uniform blocks or as edge blocks.

(a)

(b)

(c)

Fig. 7. Performing the proposed compression scheme on tested image "girl". (a) Original image; (b) reconstructed image using 4×4 windows; (c) reconstructed image using 5×5 windows.

Simple table look-up methods can be used to reconstruct each edge block. The low overhead incurred in decoding implies that a certain amount of postprocessing may be feasible. At high compression ratios, the "blockiness" effect is obvious using the proposed compression scheme, owing to the coarse quantization of the average intensities in uniform blocks. The blockiness effect is also a problem in the high compression ratios of other methods, like the VQ or DCT schemes. A simple method to eliminate partially the blockiness effect in the reconstructed image is to average the gray values of the pixels along the borders between every two neighboring blocks. This method is effective without degrading the edges in the images.

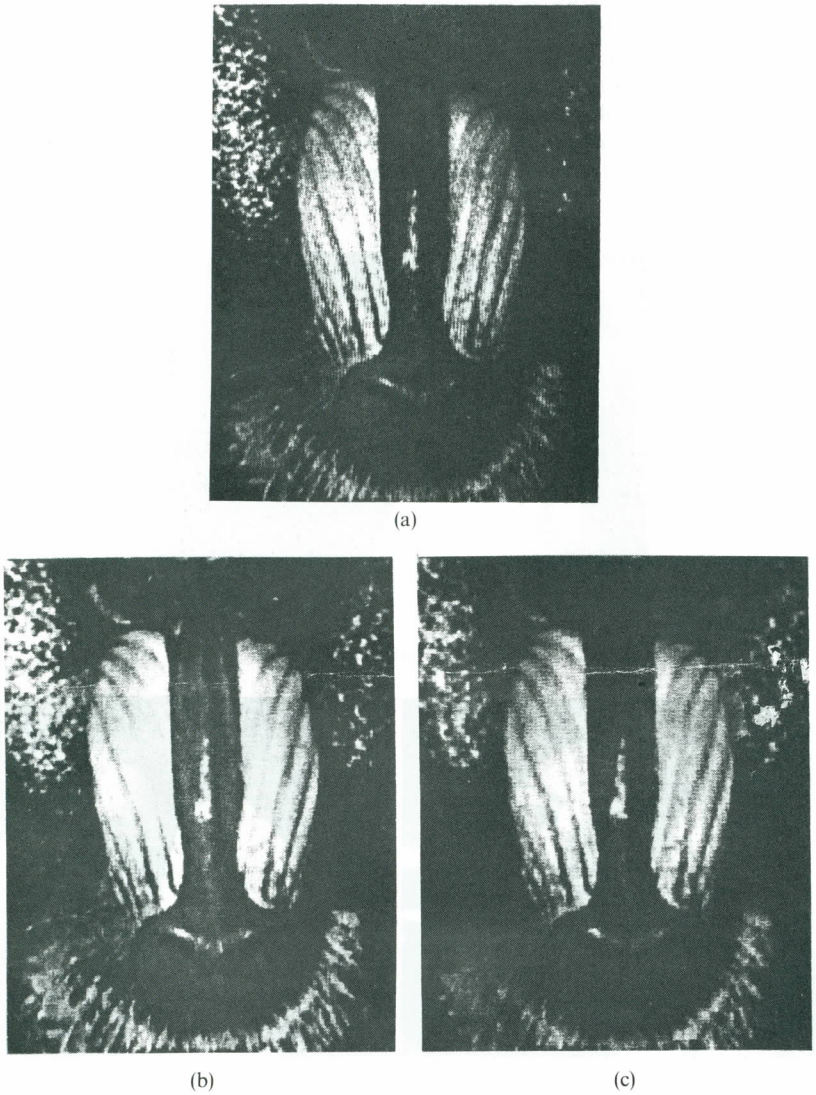
4. EXPERIMENTAL RESULTS

The proposed approach has been tested on several images. The experiments were conducted on a Sun/4 Spark workstation using the C language. Two window sizes are used in this study. The compression ratios are in the range of (10.7, 18.3) if the window size is chosen to be 4×4, and in the range of (13.33, 28.6) if the window size is 5×5. Figures 7 and 8 show two results using the proposed compression scheme.

The performance of the proposed method has been compared with the BTC and DCT methods. The BTC method is known to spend less computational expense, and the DCT is a widely used method in image compression. The reconstructed image quality is compared in terms of the SNR, which is defined as follows:

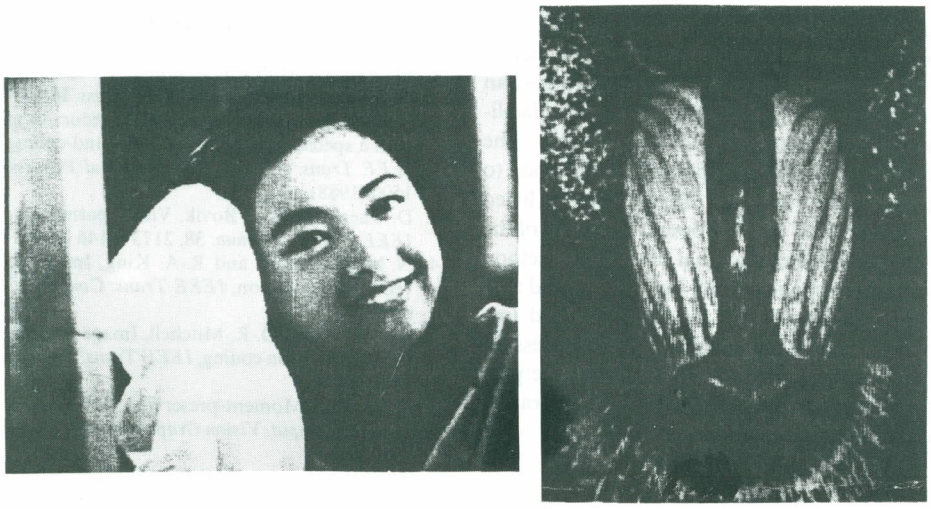
$$\mathrm{SNR} = 10 \log_{10} \left(\sum_x \sum_y \frac{[g(x,y) - f(x,y)]^2}{f^2(x,y)} \right), \qquad (41)$$

where $f(x, y)$ denotes the gray value of an input image at (x, y), and $g(x, y)$ denotes that of the reconstructed image. The block size of the BTC method is selected to be 4×4; however, the block size of the DCT method



(a) (b) (c)

Fig. 8. Performing the proposed compression scheme on tested image "monkey". (a) Original image; (b) reconstructed image using 4×4 windows; (c) reconstructed image using 5×5 windows.



(a) (b)

Fig. 9. Performing the BTC method on images "girl" and "monkey". (a) Reconstructed image of "girl"; (b) reconstructed image of "monkey". The compression ratios of both images are 5.33.

(a) (b)

Fig. 10. Performing the DCT method on images "girl" and "monkey". (a) Reconstructed image of "girl"; (b) reconstructed image of "monkey". The compression ratios of both images are 15.60.

Table 2. Computed results of performing three different methods to tested images

Images	Parameters	Proposed method		BTC	DCT
		Window size = 4	Window size = 5		
Girl	CPU time (s)	2.9	1.9	3.9	153.9
	SNR (d.b.)	31.56	29.41	31.82	29.85
	CR	16.23	23.11	5.33	15.60
Monkey	CPU time (s)	3.0	1.9	3.9	153.9
	SNR (d.b.)	24.82	23.03	26.03	22.41
	CR	15.27	23.03	5.33	15.60

is selected to be 16 × 16. For the DCT method, only the DC term and 11 AC coefficient (11 bits per coefficient) are retained such that the compression ratios can be close to those of the proposed method. This facilitates comparison of reconstructed image quality. The computed results of applying these three methods to the tested images shown in Figs 7(a) and 8(a) are listed in Table 2. It is noticed that the operation speed of the proposed method is faster than that of the BTC method with a comparable image quality (see Fig. 9), and that the distortion introduced by the DCT method (see Fig. 10) is large under the range of the compression ratios of the proposed method. Accordingly, the proposed compression scheme is efficient in both operation speed and compression quality.

5. CONCLUSION

A new image compression algorithm based on the moment-preserving principle has been proposed. The major contributions of this study include: (a) a new moment-preserving edge detector is proposed; (b) both one-step and two-step edges are modeled; (c) reconstructed images with good quality in accordance with human perception can be obtained; (d) the operation speed is fast and thus suitable for real-time applications. Some possible improvements and further research topics are as follows.

First, it can be observed the proposed edge detector only detects edges in directions of multiples of 45°. If the orientation of the edge in an image block is not actually equal to these values, distortion will be produced. This disadvantage can be modified easily by increasing the possible directions from 8 to 16. Furthermore, the proposed method can be modified to detect more complicated features, e.g. curves and lines. The size of an image block also need not be fixed. The use of variable block sizes in the compression scheme will yield generally higher compression ratios.

REFERENCES

1. M. Vetterli, Multi-dimensional sub-band coding: some theories and algorithms, *Signal Process.* **16**, 97–112 (1984).
2. T. Kronander, A new approach to recursive mirror filters with a special application in sub-band coding of images, *IEEE Trans. Accoust. Speech Signal Process.* **36**, 1496–1500 (1988).
3. D. Chen and A. C. Bovik, Visual pattern image coding, *IEEE Trans. Commun.* **38**, 2173–2146 (1990).
4. N. M. Nasrabadi and R. A. King, Image coding using vector quantization, *IEEE Trans. Commun.* **36**, 957–971 (1988).
5. E. J. Delp and O. R. Mitchell, Image compression using block truncation coding, *IEEE Trans. Commun.* **27**, 1335–1342 (1979).
6. W. H. Tsai, Moment-preserving thresholding: a new approach, *Comput. Vision Graphics Image Process.* **29**, 377–393 (1985).
7. A. J. Tabatabai and O. R. Mitchell, Edge location to subpixel values in digital imagery, *IEEE Trans. Pattern Anal. Mach. Intell.* **6**, 188–201 (1984).
8. L. H. Chen and W. H. Tsai, Moment-preserving sharpening: a new approach to digital picture deblurring, *Comput. Vision Graphics Image Process.* **41**, 1–13 (1988).
9. L. H. Chen and W. H. Tsai, Moment-preserving line detection, *Pattern Recognition* **21**, 45–53 (1988).
10. L. H. Chen and W. H. Tsai, Moment-preserving curve detection, *IEEE Trans. Syst. Man Cybern.* **18**, 148–158 (1988).
11. S. T. Liu and W. H. Tsai, Moment-preserving corner detection, *Pattern Recognition* **23**, 441–460 (1990).
12. R. Lee, P. K. Lu and W. H. Tsai, Moment-preserving detection of elliptical shapes in gray-scale images, *Pattern Recognition Lett.* **11**, 405–414 (1990).
13. H. S. Hsu and W. H. Tsai, Moment-preserving edge detection and its application to image data compression, *Optical Engineering* **32**, 1596–1608 (1993).
14. I. H. Kuo and W. H. Tsai, Moment-preserving feature detection for image compression, *1991 Workshop Comput. Vision Graphics Image Process.* Tainan, Taiwan, pp. 66–71 (1991).
15. E. J. Delp and O. R. Mitchell, Moment-preserving quantization, *IEEE Trans. Commun.* **39** (1991).
16. S. C. Cheng and W. H. Tsai, A neural network implementation of moment-preserving technique and its application to thresholding, *IEEE Trans. Comput.* **42**, 501–507 (1993).

About the Author—SHYI-CHYI CHENG was born in Changhua, Taiwan, Republic of China, on 28 September 1963. He received the B.S. degree in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, Republic of China, in 1986, and the M.S. degree in electronics and the Ph.D. degree in computer science and information engineering, both from National Chiao Tung University, Hsinchu, Taiwan, Republic of China, in 1988 and 1992, respectively. He was a winner of the 1988 AceR Long-Term Outstanding M.S. Thesis Scholarship and the 1992 Xerox Foundation Ph.D. Dissertation Study Scholarship. From July to October 1992, he served in the Chinese Army as an officer. In October 1992, he joined the Laboratories of Telecommunications, Ministry of Communications, Republic of China, at Chungli, Taiwan, as a research member. His current research interests include image/Video coding, digital signal processing, and multimedia databases.

About the Author—WEN-HSIANG TSAI was born in Tainan, Taiwan, Republic of China, on 10 May 1951. He received the B.S. degree from National Taiwan University, Taipei, Taiwan, Republic of China, in 1973, the M.S. degree from Brown University, Providence, Rhode Island, U.S.A., in 1977, and his Ph.D. degree from Purdue University, West Lafayette, Indiana, U.S.A., in 1979, all in electrical engineering. Since November 1979, he has been on the faculty of the Institute of Computer Science and Information Engineering at National Chiao Tung University, Hsinchu, Taiwan. From 1984 to 1986, he was an Assistant Director and later an Associate Director of the Microelectronics and Information Science and Technology

Research Center at National Chiao Tung University. He joined the Department of Computer and Information Science at National Chiao Tung University in August 1984, acted as the Head of the Department from 1984 to 1988, and is currently a Professor there. He serves as a consultant to several research institutes and industrial companies. His current research interests include computer vision, image processing, pattern recognition and Chinese information processing. Dr Tsai is an Associate Editor of *Pattern Recognition*, the *International Journal of Pattern Recognition and Artificial Intelligence*, the *Journal of the Chinese Institute of Engineers*, and the *Journal of Information Science and Engineering*, and was an Associate Editor of *Computer Quarterly* and *Proceedings of National Science Council of the Republic of China (Part A)*. He was elected as an Outstanding Talent of Information Science of the Republic of China in 1986. He was the winner of the 13th Annual Best Paper Award of the Pattern Recognition Society of U.S.A. He obtained the 1987 Outstanding Research Award and the 1988–1989, 1990–1991 and 1992–1993, Distinguished Research Awards of the National Science Council of the Republic of China. He also obtained the 1989 Distinguished Teaching Award of the Ministry of Education of the Republic of China. He was the winner of the 1989 and 1992 AceR Long Term Award for Outstanding Ph.D. and Master Thesis Supervision, the 1991 Xerox Foundation Award for Ph.D. Dissertation Study Supervision, and the 31st S. K. Chuang Foundation Award for Science Research. He was also the winner of 1990 Outstanding Paper Award of the Computer Society of the Republic of China. Dr Tsai has published more than 70 papers in well-known international journals. Dr Tsai is a senior member of the IEEE, and a member of the Chinese Image Processing and Pattern Recognition Society, the Computing Linguistics Society of the Republic of China, and the Medical Engineering Society of the Republic of China.