1992 National Symposium on Telecommunications
Chung-Li, Taiwan, R.O.C. Dec. 23-24, 1992

Dynamic Image Compression Using Line Edge Features by Moment-Preserving Technique

Show-Yuan Ferng (馮壽元) and Wen-Hsiang Tsai (蔡文祥)

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

ABSTRACT

A new method for feature-based dynamic image compression based on the moment-preserving principle using line edge features is proposed. To encode an image frame in an image sequence, the method first divides the image frame into small blocks, and approximates the feature in each block with a line edge using the moment-preserving technique. The parameters of the line edge can be computed analytically and so quickly. Then, by using an efficient algorithm to compute the gray and mass moments, the line edges in the overlapping blocks in the previous frame are computed. Finally, with a feature-based similarity measure, a block-matching algorithm is applied to match each block in the current frame with the blocks in a search area in the previous frame to find the displacement of the best match. The current image frame is then encoded as a set of displacement vectors. Good experimental results with low bit rates, reasonable reconstructed image quality, as well as high computation speeds prove the feasibility of the proposed approach.

1. INTRODUCTION

During the past few years, rapid advances in the developments of ISDN, HDTV, vedio conferencing, and vedio phone technologies have created high demands of low-bit-rate image sequence coding methods. Generally, image sequences contain a significant amount of redundancy between consecutive frames, and motion in the entire scene is usually low. Most of the information for the current frame can be determined from the adjacent frames. High compression rates can thus be obtained by the *motion detection and compensation coding* technique. Several methods of estimating object motion displacements in an image sequence have been proposed. One of them is the widely used *block-matching algorithm* (BMA). This algorithm predicts the motion on a *block-by-block* basis. It is assumed that all the pixels within a block undergo uniform motion. In this technique an image frame is first divided into blocks. An interframe motion detection and estimation algorithm is then applied, based upon a measure of similarity between a block of size, say, m by n in the current frame and each of the corresponding blocks in a search area with size (m + 2p) by (n + 2p) in the previous frame (as shown in Fig. 1), where the value "p" refers to the maximum displacement assumed for the block over a frame interval. The most popular measures for a BMA are the normalized cross-correlation function (NCCF), the mean-squared error (MSE), and the mean of the absolute error (MAE). The measure of the MAE is commonly used due to its simplicity. After the motion compensation process, the prediction error, defined as the difference between the current block and the motion-compensated block from the previous frame, is encoded. Then the current block can be reconstructed from the motion-compensated block and the prediction error in the reconstruction phase.

The motion of a block can be estimated by brute force on a block-by-block basis, but this requires extensive computations. Several efficient techniques which reduce the complexity have been developed. Jain and Jain [1] proposed a BMA-type technique for estimating the interframe displacements of small blocks with the minimum MSE measure. A 2D logarithmic search procedure was used by successively reducing the area of search. Koga et al. [2] proposed the MAE criteria with the advantage that no multiplication and division is required. A three-step search procedure was employed, which is closely related to the 2D logarithmic search except that at each iteration the center and the eight midpoints in the eight directions are tested. Using the MAE criteria, Kappagantula and Rao [3] proposed a relatively simple BMA for the motion detection and compensation coding technique. They assumed that the moving areas of the image are built up by a number of blocks. Then a block-difference averaging method was shown to be sufficient to produce a reasonable mismatch measure. A search method based on conjugate directions (CDS), and a simplified version of it called one-at-a-time search (OTS), were presented by Srinivasan and Rao [4]. It was shown that as much as 94% reduction in computation is possible in estimating motion vectors. In Srinivasan and Rao [5], a motion-compensated hybrid transform codec (coder/decoder) based on the OTS and the C-matrix transform (CMT) -- an efficient approximation method for discrete cosine transform (DCT) -- is presented. With OTS as the motion estimation algorithm, the CMT reduces the spatial redundancy in the transmitted prediction errors.

Since research of feature-based image sequence coding is still in an early stage, the existing techniques developed for this area usually require very complicated computations. In such techniques, scene analysis techniques are employed to extract features such as line, edge, corner, etc., from an image. Motion estimation is accomplished by feature matching. In Kretz [6], an edge-adaptive pel-recursive motion compensation method is proposed. With the derived motion parameters of edges, the precise motion of edges can be estimated and used for motion compensation coding. In Labit and Benveniste [7], a pel-by-pel motion estimation and compensation scheme is presented. It is assumed that abrupt changes in motion are located along the boundaries of objects. The motion estimator uses adaptive algorithms and instantaneous detection of motion changes.

In this paper, a new feature-based dynamic image compression method is proposed. The feature type is line edge. The basic idea is to encode each image in an image sequence block by block, with each block represented by a line edge and two adjacent areas with constant gray levels. The line feature in each block is found by computing the line parameters according to the moment-preserving principle. With a new line-based similarity measure, a block-matching algorithm is applied to match each block in the current frame with the blocks in a search area in the previous frame. The displacement of the best match is computed. The current image frame is then encoded as a set of displacement vectors. Good experimental results with low bit rates, reasonable reconstructed image quality, as well as high computation speeds prove the feasibility of the proposed approach.

2. REVIEW OF MOMENT-PRESERVING PRINCIPLE

Moments can be classified as gray moments and mass moments. Given an image f with n pixels, let the gray value at pixel (x,y) be denoted by f(x,y). The i-th gray moment m_i of f is defined as

$$m_i = \frac{1}{n}\sum_x \sum_y f^i(x,y), \qquad i = 1, 2, 3, \ldots \qquad (1)$$

Gray moments can also be computed from the histogram of f in the following way

$$\begin{aligned} m_i &= \frac{1}{n}\sum_j n_j h_j^i \\ &= \sum_j P_j h_j^i, \qquad i = 1, 2, 3, \ldots \qquad (2) \end{aligned}$$

where n_j is the total number of the pixels in f with gray values h_j and $P_j = n_j / n$. We also define m_0 to be 1.

And the zeroth- and first-order mass moments, M, M_x, and M_y, of a region R in f are defined as

$$M = \iint_R \rho(x,y)\, dx\, dy \qquad (3)$$

$$M_x = \iint_R y\, \rho(x,y)\, dx\, dy \qquad (4)$$

$$M_y = \iint_R x\, \rho(x,y)\, dx\, dy \qquad (5)$$

where $\rho(x,y)$ is the density function value of R at (x,y) which is taken to be the gray value of f at (x,y) in this study.

The moment-preserving principle for image processing is to transform an image into another form by preserving the gray and/or mass moments of the original image. The purpose of doing such a type of transformation depends on its applications. Without loss of generality, it can be said that the moment-preserving transformation aims to group the gray values of the pixels in an image into a number of classes and represent the gray values of all the pixels in each class with a single gray value.

3. OVERVIEW OF PROPOSED APPROACH

The proposed approach essentially is based on a combination of a moment-preserving line edge detection technique and a BMA. To encode the image frames in an image sequence, the first image in the sequence must be encoded first using an intraframe coding technique. For this, an image data compression method based on the gray moment preserving principle for extracting the line edge feature from an n x n block of an image is proposed. With this method, the first image is compressed as a set of line edge parameters.

The remaining images then are encoded sequentially. To encode each remaining image, an efficient algorithm for computing the gray and mass moments is proposed so that the line edge in each of the overlapping blocks in the previous image frame can be computed in constant time. The current image frame is divided into small blocks with size n x n, say, B_1, B_2, and so on. For each B_i, by measuring the similarity between B_i and the corresponding block B' at the same location in the previous image using the MAE criterion, B_i is identified to be either still or moved. If B_i is identified to be still, it is encoded as a 1-bit flag to indicate that Bi can be reconstructed by the content of B'.

Otherwise, some gray moments of B_i are computed. Then, the existence of a line edge in B_i is tested by checking if the pixel gray values of B_i are uniform. For the reason of simplicity and efficiency, the variance of the pixel gray values of B_i, Var(B_i), is used here. If Var(B_i) is less than a predefined threshold T_{Var}, it is decided that B_i is uniform and contains no feature in B_i; otherwise, it is decided that a line edge feature exists in B_i, which is then detected using the proposed moment-preserving edge detector. The value of Var(B_i) can be directly computed from the following formula with constant time,

$$\mathrm{Var}(B_i) = m_2(B_i) - m_1^2(B_i), \qquad (6)$$

where $m_i(B_i)$ is the i-th gray moment of Bi.

If there is a line edge in B_i, with a feature-based similarity measure SM_1 constructed from line edge parameters, the BMA is applied to match each block in the current frame with the overlapping blocks of a search area in the previous frame to find the displacement of the best match. If B_i is uniform, with a similarity measure SM_2 constructed from the difference of the mean gray values of blocks, the BMA is applied to find the displacement of the best match block B'. If the similarity (SM_1 or SM_2) between Bi and B' is less than a predefined threshold (T_{sim1} or T_{sim2} according to which similarity is used), the displacement is stored and used to reconstruct B_i; otherwise, B_i is compressed by storing its line edge parameters.

The details of the encoding process of an image block by the BMA as shown as a flowchart in Fig. 2.

4. PROPOSED LINE EDGE DETECTOR

The proposed line edge detector estimates an edge location in a block by a line edge whose parameters can be calculated analytically according to the moment-preserving principle. For an n x n block in a given image, the detector generates as output an ideal line edge and two intensity values h_1 and h_2 (assume $h_2 > h_1$) with the line edge separating the block into two subregions A_1 and A_2, and h_1 and h_2 being the representative intensity values for A_1 and A_2, respectively, as shown in Fig. 3. Also, the detector identifies the location of the line edge by two intersection points on the block boundary with coordinates $Q_1=(x_1,y_1)$, $Q_2=(x_2,y_2)$ (only two of the four coordinates x_1, y_1, x_2 and y_2, are unknown; the other two are 0 or n), assuming that the origin of the Cartesian coordinate system is located at the upper-left corner of the block (see Fig. 3 and Fig. 6 for detailed illustrations). When the two unknown coordinates are determined, the location of the line edge is determined. The details of the line edge detection process is described in the remainder of this section, which consists essentially of three major steps: (1) determining by gray moment preserving the representive intensity values h_1 and h_2 of the two subregions A_1 and A_2; (2) determining the type of intersection of the line edge with the block boundary (see Fig. 6); and (3) determining by mass moment preserving the two unknown coordinates of the intersection points of the line edge with the block boundary.

4.1 Determination of representative gray values of subareas

By preserving the first three moments m_1, m_2, and m_3 in the output O of the detector, four equalities are obtained:

$$\begin{aligned} P_1h_1 + P_2h_2 &= m_1, \\ P_1h_1^2 + P_2h_2^2 &= m_2, \\ P_1h_1^3 + P_2h_2^3 &= m_3, \\ P_1 + P_2 &= 1, \end{aligned} \tag{7}$$

where P_1 and P_2 are the fractions of the pixels with gray values h_1 and h_2 in O, respectively. The four parameters P_1, P_2, h_1, and h_2 in (7), can be solved [9] to be

$$\begin{aligned} h_1 &= \left[-C_1 - \sqrt{C_1^2 - 4C_0}\right] / 2, \\ h_2 &= \left[-C_1 + \sqrt{C_1^2 - 4C_0}\right] / 2, \\ P_1 &= (1/P_d)\begin{vmatrix} 1 & 1 \\ m_2 & h_1 \end{vmatrix}, \\ P_2 &= 1 - P_1, \end{aligned} \tag{8}$$

where

$$\begin{aligned} P_d &= \begin{vmatrix} 1 & 1 \\ h_1 & h_2 \end{vmatrix}, \\ C_0 &= \begin{vmatrix} -m_2 & m_1 \\ -m_3 & m_2 \end{vmatrix} / C_d, \\ C_1 &= \begin{vmatrix} 1 & -m_2 \\ m_2 & -m_3 \end{vmatrix} / C_d, \\ C_d &= m_2 - (m_1)^2. \end{aligned} \tag{9}$$

4.2 Determination of Line Edge Type

As mentioned in [8], a line edge may intersect a block (denoted as B henceforth) on any of the four different boundary edges, and twelve types of line edges can thus be derived (see Fig. 4). A new method to determine the type of the line edge in B is performed. First, the mass moments M, Mx, and My of B are computed, using Eqs. (3), (4), and (5) with the density values $\rho(x,y)$ being taken to be h_1 and h_2 for the subregions A_1 and A_2, respectively. Then the centroid $(\bar{x}, \bar{y})$ by definition is

$$\bar{x} = \frac{M_y}{M}, \qquad \bar{y} = \frac{M_x}{M}. \tag{11}$$

The centroid may be located in any of the four quadrants of B, as shown in Fig. 5. After a rotation of 90°, 180°, or 270°, the centroid of B', the rotated version of B, can be made to lie in the first quadrant. Then the line edge of B' can only be of one of the four types shown in Fig. 6.

Let $\bar{x}(A_i)$ and $\bar{y}(A_i)$ be the x and y coordinates of the centroid of the uniform region A_i (A_1 or A_2), respectively. It is found in this study that the type of the line edge of block B' can be determined by the following rule:

(1) if $\bar{x}(A_2) \le \frac{n}{3}$ and $\bar{y}(A_2) \le \frac{n}{3}$, then it is a type I edge;
(2) if $\bar{x}(A_1) \ge \frac{2n}{3}$ and $\bar{y}(A_1) \ge \frac{2n}{3}$, it is a type IV edge;
(3) if $\bar{y}(A_2) \ge \bar{x}(A_2)$, it is a type II edge;
(4) otherwise, it is a type III edge.

The rule listed above can be obtained by observing the bounds of the centroid for each of the four types as shown in Table 1. The proof of these bounds can be found in [13].

Table 1. The bounds of the centroid for the four line edge types.

	$\bar{x}(A_1)$	$\bar{y}(A_1)$	$\bar{x}(A_2)$	$\bar{y}(A_2)$
type I	$\le \frac{2n}{3}$	$\le \frac{2n}{3}$	$\le \frac{n}{3}$	$\le \frac{n}{3}$
type II	$\ge \bar{y}(A_1)$	$\le \frac{2n}{3}$	$\le \bar{y}(A_2)$	$\ge \frac{n}{3}$
type III	$\le \frac{2n}{3}$	$\ge \bar{x}(A_1)$	$\ge \frac{n}{3}$	$\le \bar{x}(A_2)$
type IV	$\ge \frac{2n}{3}$	$\ge \frac{2n}{3}$	$\ge \frac{n}{3}$	$\ge \frac{n}{3}$

4.3 Determination of line edge parameters

In the following, only the type II edge of Fig. 6 are investigated; the others can be derived in similar ways. The goal is to derive analytic formulas for computing the two unknown coordinates x_1 and x_2 of the intersection points of the line edge with the block boundary (see Fig. 6II). Recall that A_1 and A_2 are two uniform subregions of region A separated by the line edge. By preserving the zeroth- and the first-order mass moments M and M_X of A and after some geometric reasoning, we get

$$\begin{aligned} M(A) &= M(A_1) + M(A_2) \\ &= \iint_{A_1} \rho(x,y)\,dx\,dy + \iint_{A_2} \rho(x,y)\,dx\,dy \\ &= h_1 n^2 + (h_2 - h_1)\, n \frac{x_1 + x_2}{2} \end{aligned} \tag{12}$$

$$\begin{aligned} M_x(A) &= M_x(A_1) + M_x(A_2) \\ &= \iint_{A_1} y\,\rho(x,y)\,dx\,dy + \iint_{A_2} y\,\rho(x,y)\,dx\,dy \\ &= \frac{n\,M(A)}{2} + (h_2 - h_1)\, n^2 \frac{x_1 - x_2}{12} \end{aligned} \tag{13}$$

where the mass moments M(A) and $M_X(A)$ of region A can be computed directly from the original input image f as follows:

$$M(A) = \sum_x \sum_y f(x,y), \tag{14}$$

$$M_x(A) = \sum_x \sum_y y\, f(x,y). \tag{15}$$

With (4-12) and (4-13), the two unknowns parameters x_1 and x_2 of the type II line edge (see Fig. 6II) can be solved to be:

$$\begin{aligned} x_1 &= A - B, \\ x_2 &= A + B, \end{aligned} \tag{16}$$

where

$$A = \frac{M(A) - h_1 n^2}{n(h_2 - h_1)},$$

$$B = \frac{6M_X(A) - 3nM(A)}{n^2(h_2 - h_1)}. \tag{17}$$

5. SIMILARITY MEASURES

In this section, the similarity measure of two blocks are defined in terms of line edge parameters. The similarity measure used here is constructed from the parameters of line edges. For a block B belonging to the current frame and a block B' in the previous frame, useful line edge parameters to measure the similarity of the two blocks are:

(a) the coordinates of the endpoints Q_1 and Q_2 of the line edge;
(b) the representative intensity values h_1 and h_2 of the "background" (region A_1) and the "object" (region A_2), respectively;
(c) the areas of the regions of "object" and "background", a_1 and a_2; and
(d) the centroid, $(\bar{x}, \bar{y})$.

The area a_1 and a_2 can be computed by $a_1 = P_1 n^2$ and $a_2 = P_2 n^2$.

If two blocks B and B' are similar, the values of the parameters of them would be much alike. The similarity measure of the two blocks can thus be defined as a function of the four sets of parameters. Since the complexity of computation is the major consideration, a simple and efficient similarity measure is desired. The similarity measure SM_1 used in this study is defined as a linear weighted combination of the four sets of parameters as follows:

$$\begin{aligned} &SM(B,B') \\ &= w_1 \bullet (\text{distance of endpoints of B and B'}) + \\ &\quad w_2 \bullet (\text{difference of intensity values } h_i \text{ of B and } h_i' \text{of B'}) + \\ &\quad w_3 \bullet (\text{difference of the areas of the features of B and B'}) + \\ &\quad w_4 \bullet (\text{distance of the centroids of B and B'}), \end{aligned}$$

where the value of each parameter is normalized to be between 0 and 1 by dividing the maximum possible value of that parameter and the summation of the weights $\sum_i w_i = 1$. The values w_i are determined experimentally. For this study, they are selected to be identical and so are equal to 1/4.

6. EXPERIMENTAL RESULTS

The proposed approach has been tested on several 720 x 480 image sequences. It takes about 6 to 7 seconds of CPU time per image on a Sun workstation, depending on the selected block size. The sequence used to illustrate the compression result here is a 29-frame gray image sequence with resolution 720 x 480 x 8 bits, called "Suzie". Let the frames be denoted as "Suzie1" through "Suzie29". The block size is chosen to be 6 x 6 in the experiment. Fig. 9 shows the intermediate compression results. The first image, "Suzie1", of the sequence, is shown in Fig. 9(a), and its reconstruction after the intraframe coding process is shown in Fig. 9(b). The original image of "Suzie10" and the result after interframe encoding are shown in Fig. 9(c) and Fig. 9(d), respectively. The original image of the last frame, "Suzie29", in the sequence and its reconstruction are shown in Fig. 9(e) and Fig. 9(f), respectively. It can be seen that the quality of the reconstructed image of the 29th frame is still satisfactory. In Table 2, the numbers of bits required to compress the sequence "Suzie" and the compression ratios for compression by the proposed approach with two different block sizes are shown.

Table 2 The number of bits required to store the image sequence "Suzie" and its compression ratio.

	Bits required	Compression ratio
size 6 x 6	2783325	28.80
size 8 x 8	1750067	45.82

REFERENCES

[1] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe coding," *IEEE transactions on Communications*, vol. COM-29, pp. 1799-1808, Dec. 1981.

[2] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and

T. Ishiguro, "Motion compensated interframe coding for video conferencing," *Proceedings of 1981 NTC*, New Orleans, LA, pp. G5.3.1-5.3.5, 1981.

[3] S. Kappagantula and K. R. Rao, "Motion compensated interframe prediction," *IEEE Transactions on Communications*, vol. COM-33, pp. 1011-1015, Sept. 1985.

[4] R. Srinvisan and K. R. Rao, "Predictive coding based on efficient motion estimation," *IEEE Transactions on Communications*, vol. COM-33, pp 888-896, Aug. 1985.

[5] R. Srinvisan and K. R. Rao, "Motion-compensated coder for videoconferencing," *IEEE Transactions on Communications*, vol. COM-35, pp 297-304, Mar. 1987.

[6] F. Kretz, "Edge in visual scenes and sequences: applications to filtering, sampling and adaptive DPCM coding," *Image Sequence Processing and Dynamic Scene Analysis*, T. S. Huang, Ed., Berlin, Germany: Springer-Verlag, 1983, pp. 125-155.

[7] C. Labit and A. Benveniste, "Motion estimation in a sequence of television pictures," *Image Sequence Processing and Dynamic Scene Analysis*, T. S. Huang, Ed., Berlin, Germany: Springer-Verlag, 1983, pp. 292-306.

[8] H. S. Hsu and W. H. Tsai, "Moment preserving edge detection and its application to image data compression," *Proceedings of International Computer Symposium*, Hsinchu, Taiwan, R.O.C., pp. 428-433, 1990.

[9] W. H. Tsai, "Moment-preserving thresholding: a new approach," *Computer Vision, Graphics, and Image Processing*, v. 29, pp. 377-393, 1985.

[10] H. G. Musmann, P. Pirsch and H. J. Grallert, "Advances in picture coding," *Proceedings of the IEEE*, vol. 73, n. 4, pp. 523-548, Apr. 1985.

[11] T. R. Hsing, "Motion detection and compensation coding for motion video coders: technical review and comparison," *IEEE/IEICE Global Telecommunications Conference*, vol. 1, pp. 2.6.1-2.6.5, 1987.

[12] C. T. Chen and T. R. Hsing, "Digital coding techniques for visual communications," *Journal of Visual Communication and Image Representation*, vol. 2, n. 1, pp. 1-16, Mar. 1991.

[13] S. Y. Ferng and W. H. Tsai, "Dynamic image compression using line edge features by moment-preserving technique," *Technical Report*, Insititute of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan, Republic of China, June 1992.

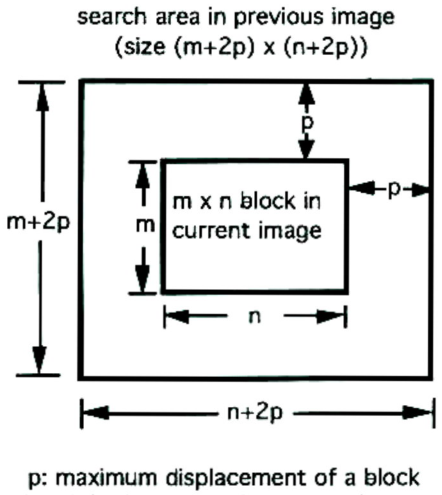

Fig. 1 A block in the current image and its corresponding search area in the previous image.

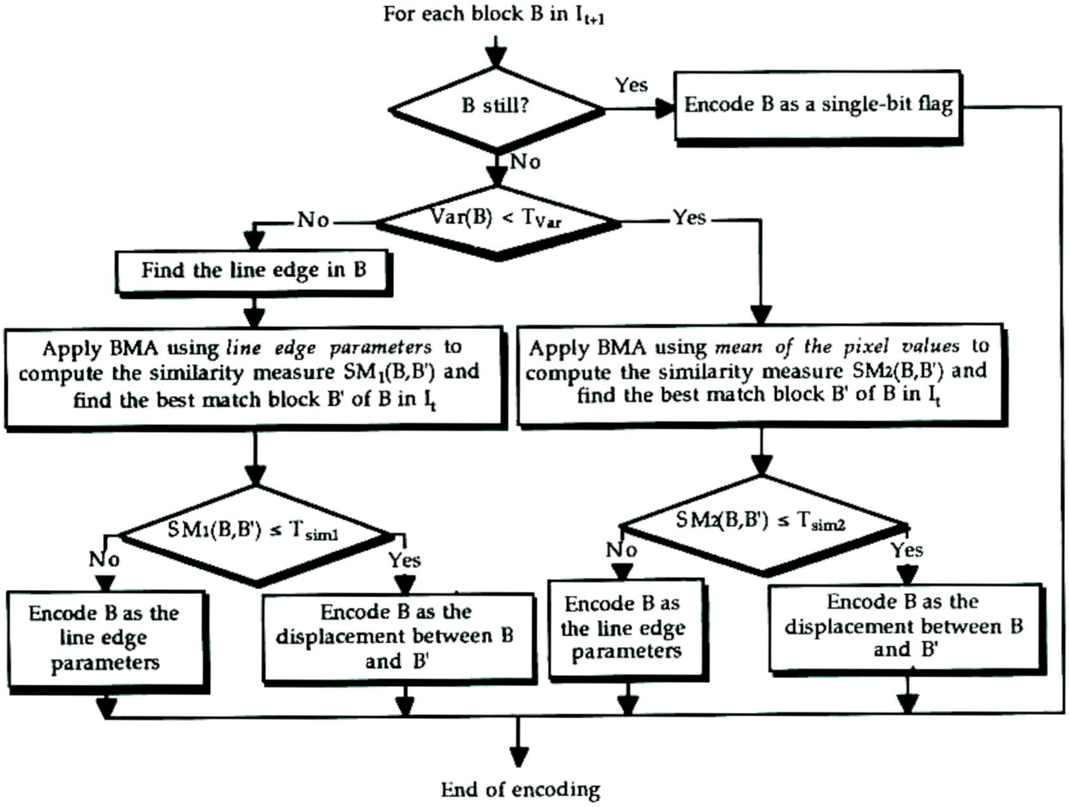

Fig. 2 The flowchart of encoding an image block by the BMA in the proposed approach.

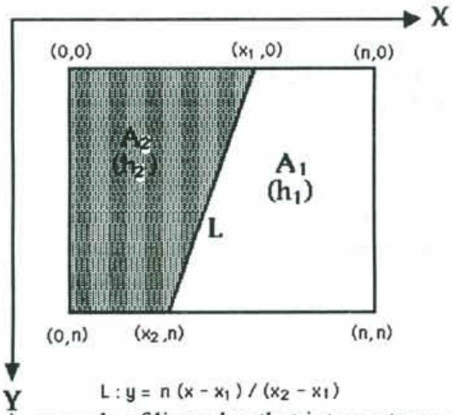

Fig. 3 A example of line edge that intersects an n x n block at two points $(x_1,0)$ and (x_2,n).

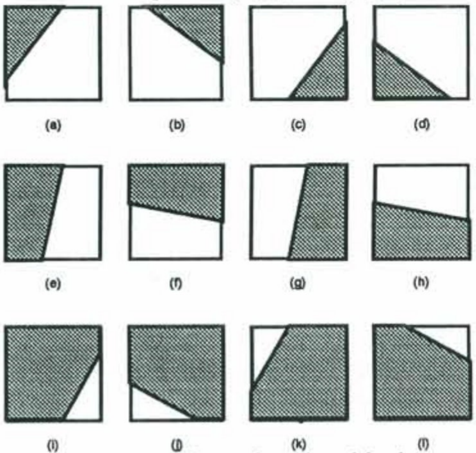

Fig. 4 Twelve types of line edges in a block.

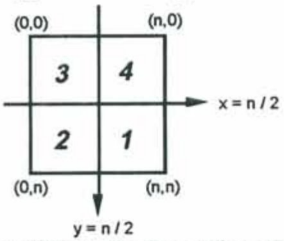

Fig. 5 The centroid may be located on one of the four quadrants.

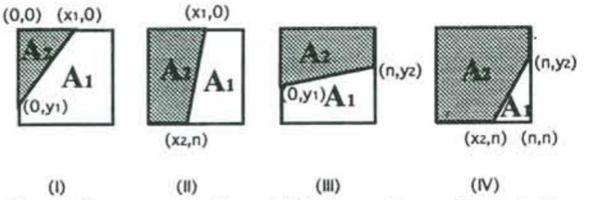

Fig. 6 Four types of possible line edges after rotation.

(a) Original "Suzie1".

(b) Reconstructed "Suzie1".

(c) Original "Suzie10".

(d) Reconstructed "Suzie10".

(e) Original "Suzie29".

(f) Reconstructed "Suzie29".

Fig. 7 The image sequence "Suzie" is compressed with 6 x 6 block size.