1998 International Computer Symposium
Workshop on Image Processing and Character Recognition
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

Color Video Compression Using Fractal-Based Segmentation Scheme

蔣依吾、劉宏城、王崇任
John Y. Chiang, H. C. Liu and C. Z Wang
中山大學應用數學系資訊組
Department of Applied Mathematics
National Sun Yat-Sen University
Kaohsiung, Taiwan 80424

Abstract

Low bit rate color image sequence coding is very important for video transmission and storage applications. Color images are usually compressed in a luminance-chrominance coordinate space, with the compression performed independently for each coordinate by applying the monochrome image processing techniques. For image sequence compression, the design of an accurate and efficient algorithm for computing motion to exploit the temporal redundancy has been one of the most active research areas in computer vision and image compression [1-4]. Pixel-based motion estimation algorithms address pixel correspondence directly by identifying a set of local features and computing a match between these features across the frames [5]. These direct techniques share the common pitfall of high computation complexity resulting from the dense vector fields produced. For block matching motion estimation algorithms, the quad-tree data structure is frequently used in image coding to recursively decompose an image plane into four non-overlapping rectangular blocks [6, 7].

1. Introduction

Fractal based image and video compressions have been studied [8-11]. The theory of iterated contractive transformations is utilized to record the transformation function for every partition between the domain and range block. However, the results reported, 0.68 bit/pixel compression rate (11.76 compression ratio) with *SNR* 27.7 dB in the two-dimensional case [8], and 41.80 and 74.39 compression ratio with *PSNR* around 29dB and 33dB, respectively, in the three-dimensional one [9-11], still leave much room to be desired in the very low bit rate applications. In order to meet the compression requirements of the diverse image sequence characteristics, we propose a robust segmentation technique based on the fractal dimensions of the ensemble luminance and chrominance difference, respectively, relative to a reference frame inside a group of pictures (GOP). The segmentation criterion employing the fractal dimension of the GOP frame difference captures and exploits both the interframe and intraframe redundancy. Since the fractal dimension of the ensemble of the GOP frame differences, rather than the value of every interframe pixel difference itself, is utilized as the basis for splitting, only one segmentation topology is needed to represent the moving information along each color coordinate for a GOP. Sporadic noises, which may cause significant perturbations in a random fashion, incur large amount of intraframe and interframe pixel differences leading the traditional segmentation techniques to divide a region into many small, fractured partitions. On the other hand, a fractal based frame difference segmentation scheme is insensitive to scattered variations in either the spatial or temporal domains. A spike caused by the random noise in any single frame will not bring significant deviation in terms of the fractal dimension obtained from an ensemble of the frame differences inside a GOP [12]. Therefore, a metric measures the fractal dimension distance between two matching regions can better reflect the degree of similarity than that based on changes in pixel values.

The segmentation criterion discussed in this paper is based on the fractal dimension of the GOP frame difference. To this purpose, a modified box-counting method is introduced in Section 2 for estimating the ensemble fractal dimensions corresponding to differences in the digital luminance Y_d and chrominances C_r, C_b. In Section 3, the fractal dimensions obtained, namely, feature maps, are treated as the major feature for a variation of a quad-tree segmentation approach, namely, "slicing floorplan," by recursively partitioned a rectangle either horizontally or vertically to form two new rectangles [6]. The partitioning repeats recursively until a covering tolerance based on metric relating to fractal dimension is satisfied. The three partitioning topologies converged independently along Y-C_b-C_r color coordinates are merged into a single mask to further increase the compression ratio. The method proposed is tested on image sequences containing various motion dynamics in Section 4. The performance in terms of the compression ratio is tabulated and the peak signal-to-noise ratio (*PSNR*) in all three color channels for each reconstructed frame are illustrated. Finally, conclusions are made and future works are discussed.

2. Ensemble Fractal Dimension Estimation

Fractal dimension is a promising feature metric proposed to characterize roughness and self-similarity in an image sequence. Roughness is usually resulted from the edge components in the spatial domain and movement in the temporal domain, while self-similarity corresponds to both spatial and temporal redundancy. The above observation serves as the motivation for employing fractal

dimension as the major feature discriminator to segment an image sequence. In this section, a modified box-counting approach for estimating fractal dimension of an ensemble of the GOP pixel luminance and chrominance differences is described. The feature maps corresponding to the fractal dimensions of the luminance and chrominance components are used as the bases for the next-stage slicing floorplan segmentation in the following section.

A pixel $I(x, y, t)$, $1 \le x, y \le n, t > 0, x, y, t \in N$, is a discrete sample in both the spatial and temporal domains of a continuous color image sequence. Digital color images are often acquired in an RGB (red-green-blue) coordinate system, with the RGB values quantized to integers within a fixed range. We first convert the original RGB image to Y-C_b-C_r system through the following formulation:

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}.$$

Fig. 1 (a) is the first frame F_1 in consecutive 33 frames of the color Claire sequence. Fig. 1 (b) – (d) are the corresponding Y, C_b and C_r color component images of the first frame F_1 in (a), respectively.

For a fixed t value, the two-dimensional vector composed of $n \times n$ pixels is denoted as an image frame F_t. An ensemble of $m+1$ successive image frames $F_1, F_2, \ldots, F_{m+1}$, is termed as a group of pictures (GOP). A GOP containing $m+1$ temporally consecutive image frames is the basic processing unit considered in our scheme. Among them, a frame F_r is chosen as the reference for all the other constituent frames within the same GOP. The reference frame for the aforementioned Claire sequence in a GOP is the 17th frame, as shown in Fig.7 (a). For every Y, C_b and C_r color components, a total of m interframe differences derived from the matching pixel values in the same spatial location between F_r and the remaining frames can be derived. The interframe differences Y-$Diff(x, y, t)$, C_b-$Diff(x, y, t)$ and C_r-$Diff(x, y, t)$ represent the pixel luminous and chromonous variations between the reference and the t^{th} frames at spatial coordinate (x, y), respectively. A small value of Y-$Diff(x, y, t)$, C_b-$Diff(x, y, t)$ or C_r-$Diff(x, y, t)$ corresponds to a similar intensity or color component at the same coordinate position along the temporal direction, while a large one indicates a dynamic scene caused by motion or sporadic noises. The interframe differences of the three color components Y-$Diff(x, y, 1)$, C_b-$Diff(x, y, 1)$ and C_r-$Diff(x, y, 1)$ for the Claire sequence are shown in Fig.2 (a) – (c), respectively, after normalizing to 255 for ease of observation.

Given the frame differences Y-$Diff(x, y, t)$, C_b-$Diff(x, y, t)$ and C_r-$Diff(x, y, t)$, how to effectively split the image plane into rectangular blocks corresponding to regions moving in unison while at the same time maintain high compression rate and signal-to-noise ratio is the major task facing the segmentation scheme. Rather than directly utilize pixel difference Y-$Diff(x, y, t)$, C_b-$Diff(x, y, t)$ and C_r-$Diff(x, y, t)$ as the measure for segmentation, the fractal dimensions of the frame differences within a GOP are employed as the discriminating metric for the slicing floorplan segmentation scheme in our approach.

A modified box-counting approach, extended to an image sequence by following Sarkar and Chaudhuri's method, is used to estimate the fractal dimensions of the frame differences Y-$Diff(x, y, t)$, C_b-$Diff(x, y, t)$ and C_r-$Diff(x, y, t)$ inside a GOP relative to the reference frame F_r. For each color component, a three-dimensional difference volume V is formed by tiling up the planes of the interframe difference together, where the value of each voxel represents the interframe luminous or chromonous difference between matching pixels $I(x, y, r)$ in the reference frame F_r and $I(x, y, t)$ in the t^{th} frame F_t, $1 \le x, y \le n$ and $1 \le t \le m+1, t \ne r$. For the sake of notational clarity, the third parameter t in all interframe differences Y-$Diff(x, y, t)$, C_b-$Diff(x, y, t)$ and C_r-$Diff(x, y, t)$ is re-indexed from 1 to m, i.e., $1 \le t \le m$. The scaling factor s, used in the calculation of fractal dimension below, between the temporal and spatial dimensions is defined as $s = m/n$. Fig. 3 illustrates the notations used in this section.

The difference volume V is partitioned into cubes of size $a \times a \times a$. On each cube partitioned, we can follow the fractal paradigm using a "yardstick" of size a to measure the magnitude of the interframe difference inside the cube. Let the minimum and maximum values of the interframe difference in the partitioned cube enclosing the voxel (x, y, t) fall into the i^{th}- and the j^{th}-fold measurements, respectively. Then the contribution attributable to this specific cube is defined as $cont(x, y, t) = j - i + 1$. Summing up the contributions from all cubes over an overlapping volumes of size $m \times m \times m$ centered at voxel (x, y, t), i.e., the volume including voxels $(x+p, y+q, t+r)$, $(-m/2 + 1) \le p, q, r \le m/2$, we have the total contribution C from this overlapping volume as follows:

$$C = \sum_{(-m/2+1) \le p,q,r \le m/2} cont(x+p, y+q, t+r).$$

The fractal dimension $FD_{(x,y,t)}$ for the voxel (x, y, t) inside the difference volume V can be derived from the contribution C and the scaling ratio s according to the following fractal formulation:

$$FD_{(x,y,t)} = \frac{\log(C)}{\log(1/s)}.$$

A higher value of $FD_{(x,y,t)}$ implies abrupt changes in the neighborhood of the voxel (x, y, t) along the corresponding Y, C_b or C_r coordinate in the temporal domain. The phenomenon can be attributed to the relative movement between the reference frame and the rest of the frames belonging to the same GOP or drastic changes in scene contents. A small $FD_{(x,y,t)}$ indicates a relatively stationary scene following the Y, C_b or C_r axis along the temporal direction. According to the above formulation, the calculation of the fractal dimension is repeated for every voxel (x, y, t) on the $t = m/2$ plane inside the difference

volume V. The dynamic range of the fractal dimension obtained for a three-dimensional volume is between 3 and 4. The fractal dimension $FD_{(x,y,t)}$ is organized as a two-dimensional array $F(x, y)$, called feature map. This feature map represents the overall motion dynamics of a GOP by taking frame F_r as the reference. Based on the value of the $F(x, y)$, the image plane can be classified into intense moving and stationary areas. A single feature map $F(x, y)$ for each color component representing the fractal dimension of the ensemble difference, rather than multiple interframe differences *Y-Diff*(x, y, t), C_b-*Diff*(x, y, t) and C_r-*Diff*(x, y, t), $1 \le t \le m$, is utilized as the feature descriptor in the following segmentation processing. The feature maps representing the fractal dimensions of the corresponding interframe differences for each color axis are shown in Fig. 4 (a) – (c), respectively, after normalizing to 255 for ease of observation. The head portion, especially the area surrounding the eyes and mouth, corresponds to intense motion in the GOP, while the background and the majority of the torso remain stationary.

The slicing floorplan coding scheme, introduced in the next section, segments the image frames into non-overlapping, varying-size rectangles according to the ensemble fractal dimension $F(x, y)$ reached for each color coordinate. The three independently converged splitting topologies are merged into one single partitioning mask. The blocks partitioned are then compared with those of the reference frame. The relative displacements between matching blocks are recorded and used in the later reconstruction process.

3. Slicing Floorplan Segmentation

The feature map $F(x, y)$, representing the degree of variation along specific color coordinate in both temporal and spatial domains, is partitioned by applying vertical or horizontal cutting lines into rectangular blocks according to a variation of the quad-tree paradigm, namely, "slicing floorplan" [6]. The slicing floorplan representation is created by recursively splitting a single rectangle into two smaller rectangles, not necessarily the same size, as opposed to the four equal-sized subsquares in the quad-tree case. A binary tree can be constructed to represent the segmentation topology, where the coordinates of the rectangles are stored in the nodes and the structure of the rectangles can be derived from the hierarchy of the tree. The convention that the slicing floorplan algorithm always splits first along the larger side of a rectangle is observed. A cutting line L separates a rectangular plane R into two neighboring ones, R_L^1 and R_L^2, having the dissimilarity between them maximized. The potential cutting line is tested by comparing the mean value of the constituent elements on both sides of the line. The difference in terms of the motion dynamics in the rectangles partitioned, R_L^1 and R_L^2, is discriminated through the amount of disparity in the mean value of the fractal dimension. A cutting line therefore differentiates regions with high contrast of motion dynamics, i.e. stationary versus intense movement, into two smaller rectangles. A cutting operation is performed on $F(x, y)$ when the maximum mean difference is reached by following:

A cutting line L along larger side of rectangle R is chosen, *if and only if,*

for $L \in R$, $Maximize\left(\left| M_{R_L^1} - M_{R_L^2} \right| \right)$,

where $R_L^1 \cap R_L^2 = L$, $R_L^1 \cup R_L^2 = R$, and $M_R = \sum_{(x,y) \in R} \frac{F(x,y)}{\| R \|}$,

$F(x, y)$ is the fractal value corresponds to the $(x, y)^{th}$ position in the feature map and $\|R\|$ the total number of elements in R.

Once segmented, the variance of the fractal dimension on each partitioned rectangle is compared with a threshold to determine whether additional splitting operation is required.

$V_R \ge T \Rightarrow$ rectangle R will be segmented in the next iteration,

where $V_R = \sum_{x,y \in R} \frac{(F(x,y) - M_R)^2}{\| R \|}$, $F(x, y)$ is the feature map and T the threshold.

This procedure is repeated until the variances of all the blocks partitioned are less than the threshold T selected, representing pixels inside a block having a homogeneous motion dynamics. A homogeneous block satisfying the variance requirement may subject to different interpretations depending on the block fractal dimension. A block with low value of fractal dimension indicates a portion with less intense activity across the entire GOP. A rectangular block with uniformly high fractal dimension corresponds to intense movement and is subjected to further splitting operation. The rectangular pattern finally reached is called the partitioning mask for the group of pictures examined. The slicing floorplan algorithm isolates relatively stationary portions within an ensemble into larger rectangular areas and regions with more intense activities into finer partitioned rectangles. The partitioning masks corresponding to the Claire sequence for each color coordinate are shown in Fig. 5(a) – (c), respectively. The partitioning masks for both the C_b and C_r axes are much simpler than that of Y axis. This observation remains valid for all the image sequences tested and serves as the motivation for merging the three partitioning masks into a single one to further decrease the amount of data stored.

The locations and sizes of the blocks determined from the recursive splitting process are used as a mask for the later motion compensation processing. In order to reduce the amount of partitioning data stored, the three partitioning masks for all color coordinates are grouped together into one single mask. Fig. 5 (d) is the final partitioning mask obtained after grouping the three Y-C_b-C_r masks in Fig. 5 (a) – (c). Every frame within a GOP is partitioned according to the final mask obtained. Each rectangle partitioned is associated with an independent motion vector in each frame. The motion vector of a rectangle for a frame represents the relative displacement between the corresponding rectangles in the reference image and the specific frame currently considered. Only one partitioning mask is required for the entire GOP in our scheme as opposed to the general practice that one segmentation topology is needed to represent the moving

information for each frame.

The image frames within the same GOP are split according to the partitioning mask obtained. Each block is compared with its counterpart in the reference frame within a limited search region. A match is reached with the least amount of error for the whole search region and the corresponding displacement is recorded. Let b_i be the i^{th} block in the mask M and S the search region. For each block $b_i \in M$, a motion vector $(dx_i, dy_i)_t$ for the t^{th} frame is determined by exhaustively shifting the pixels within block b_i of the t^{th} frame F_t with an amount $(dx_i, dy_i)_t$, such that the summed squared error of all three color components between the pixels inside the displaced block and those in the matching location of the reference frame F_r is minimized, i.e.

$(dx_i, dy_i)_t$ is the motion vector for the block b_i of mask M in the t^{th} frame if and only if $\sum_{c=Y,C_b,C_r} \{ \sum \sum_{(x,y)\in b_i} (I^c(x+dx_i, y+dy_i, t) - I^c(x,y,r))^2 \}$ is minimized, where $(dx_i, dy_i) \in b_i$.

The mask, the motion vector, and the reference frame are used in the decompression process by segmenting the image frames into blocks according to the stored mask topology. Each block b_i inside the t^{th} frame is reconstructed by copying the contents of a matching rectangular area in the reference frame with a displacement specified by the motion vector $(dx_i, dy_i)_t$. The three color components of the first frame in the Claire sequence after decompression are shown in Fig. 6 (a) – (c).

The computational complexity on the encoding side, mostly attributed to the calculation of GOP fractal dimension and motion compensation processes, is much higher than that of the decoding operation. However, the decompression process involving only duplication of the pixel contents from the reference blocks to the matching locations according to the motion vector is extremely simple. Due to this desirable property, not only real-time decompression on the receiving ends is feasible, but also the implementing hardware cost can be lowered. The performance of this proposed fractal video compression algorithm is studied in the next section by employing color image sequences with diverse motion dynamics.

4. Experimental Results

Computer simulation is performed using the approach described in the previous sections to demonstrate the feasibility and efficiency of our approach. Four standard image sequences - Claire, Susie, Train and Football, with a resolution of 352 x 240 pixels and a depth of 8 bits per color channel for a total of 24 bits per pixel - are employed as test sequences. Fig. 7 (a) – (d) show one typical frame for each of the four test image sequences.

In our implementation, a GOP is consisted of 33 successive frames. The 17th frame is chosen as the reference frame. The fractal dimension of the 32 intensity differences between the reference and the remaining frames is calculated according to Eq. (1). The dynamic range of the fractal dimension obtained for the piling of two-dimensional planes is between 3 and 4. The number of blocks partitioned relates directly to the compression ratio and *PSNR*. As the number of blocks increases, the higher the *PSNR* and the lower the compression ratio, and vice versa. The number of partitioned blocks can be controlled by adjusting the variance thresholds as formulated in Eq. (2). The peak signal-to-noise ratio (*PSNR*) per color channel is defined as:

$$per-channel-PSNR = 10 \log \frac{255^2}{MSE} dB,$$

where $MSE = \left(\frac{1}{n \times n}\right) \sum_{i=1}^{n} \sum_{j=1}^{n} (x_{ij} - \hat{x}_{ij})^2$, x_{ij} is the value of a specific color channel in the original image and $\hat{x}_{ij}$ the decompressed one. The *per-channel-PSNR* of every frame within a GOP for the four test sequences are illustrated in Fig. 8 (a) – (d). Since the reference frame itself is not motion compensated, the reference frame has the highest fidelity and *PSNR* value across a GOP. The average *PSNR* of each color channel is calculated by taking the average of *per-channel-PSNR* for all 33 decompressed frames within a GOP.

In order to decompress the image sequence at the decoding end, the global partitioning mask, the block motion vector with respect to the reference frame, and the reference frame itself needs to be stored. The partitioning topology contains only rectangles and can be easily encoded. The size of each motion vector will reflect the range of search region selected. In our implementation, we define a window of size 32 x 32 as the search area. Therefore, 10 bits are sufficient for each motion vector. The reference frame itself is encoded by the JPEG standard. The results show consistent low bit rates ranging from 0.06 ~ 0.08 bit/pixel (corresponding to a compression ratio 300 ~400) for videos with low motion dynamics, e.g., Claire and Susie.

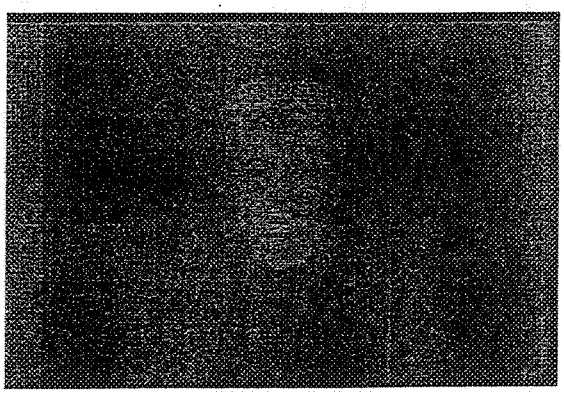
5. Conclusion

The utilization of the fraction dimension of the frame difference, rather than the intensity value of the frame difference, as the major means for segmentation allows a more robust and consistent compression result. The computational complexity on the decoding side is simpler than that of transform-based algorithms, such as MPEG, since only duplication of the pixel contents in the reference blocks to the matching locations is required. Real-time decompression can be achieved with simplified hardware requirements rendering lower system cost. As a practical video compression algorithm, the proposed method can be utilized in a wide range of image sequence compression, transmission and archiving applications.

(a)

(b)

(c)

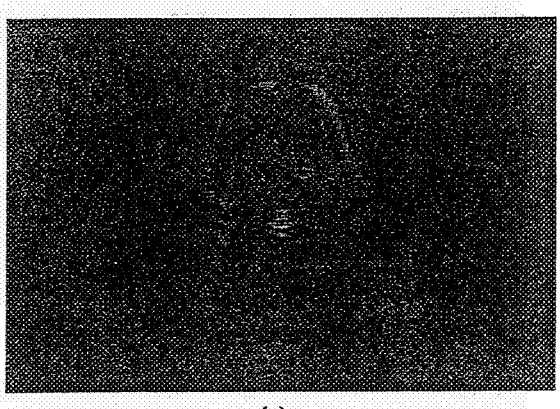


(d)

Fig. 1 (a) The first frame of a GOP containing 33 consecutive frames in the color Claire sequence. (b) – (d) The Y, C_b and C_r components of the first frame, respectively.

(a)

(b)



(c)

Fig. 2 (a) – (c) The interframe differences Y-$Diff(x, y, 1)$, C_b-$Diff(x, y, 1)$ and C_r-$Diff(x, y, 1)$ for each color coordinate between the first frame and reference frame in Fig. 6 (a), after normalizing to 255 for ease of observation.

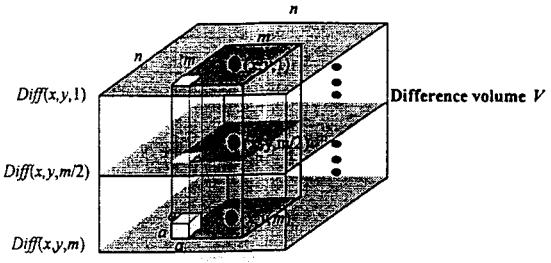
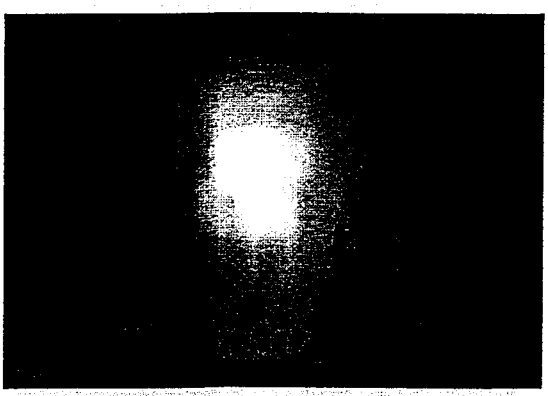
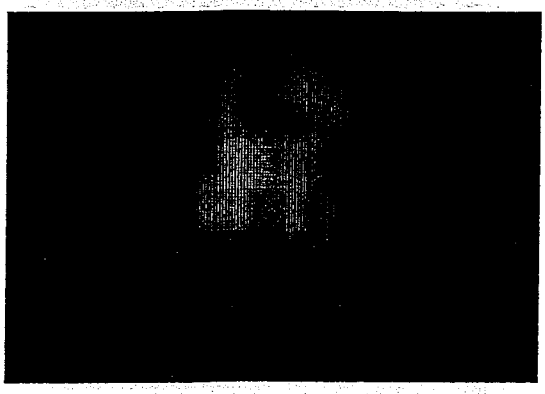


Fig. 3 The relationship between the frame dimension n, interframe difference $Diff(x, y, t)$ along each color coordinate, the number of interframe difference m in a group of pictures (GOP), and the dimension of the measuring cube a. A difference volume is formed for each color coordinate.



(a)



(b)

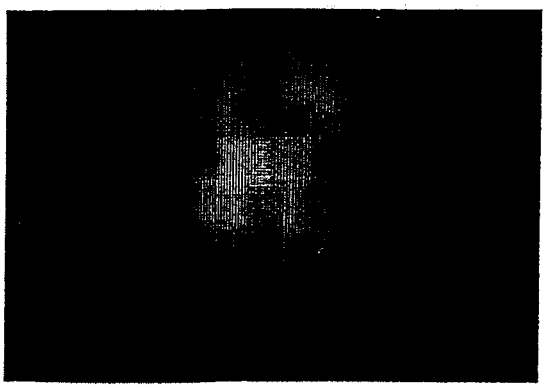
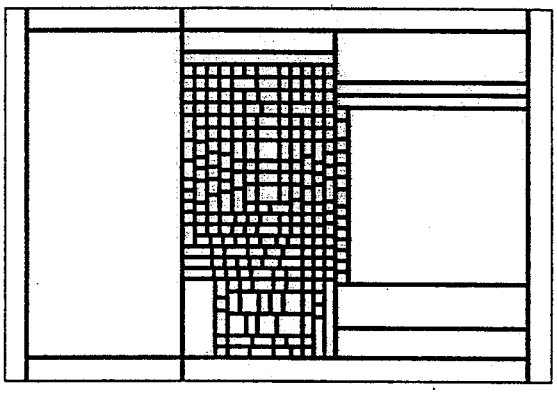
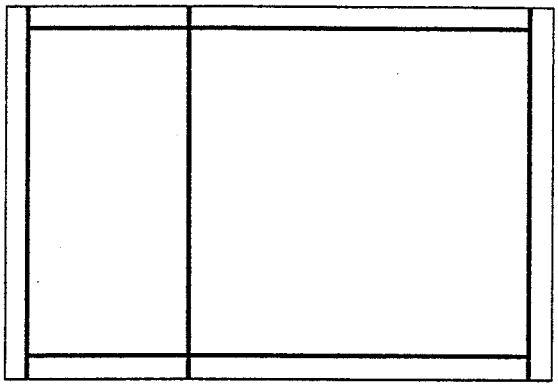


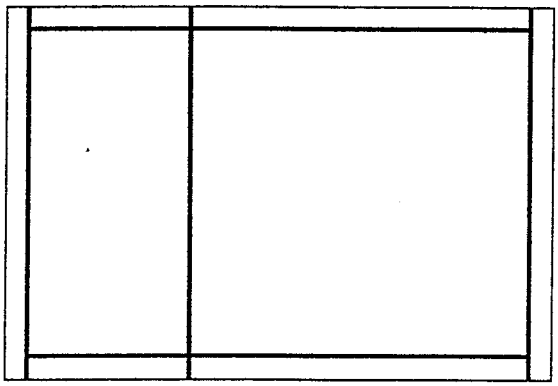
Fig. 4 (a) – (c) The feature map representing the fractal dimension of the interframe difference for each color coordinate in Fig. 1 (a) – (c), respectively, after normalizing to 255 for ease of observation.



(a)



(b)



(c)

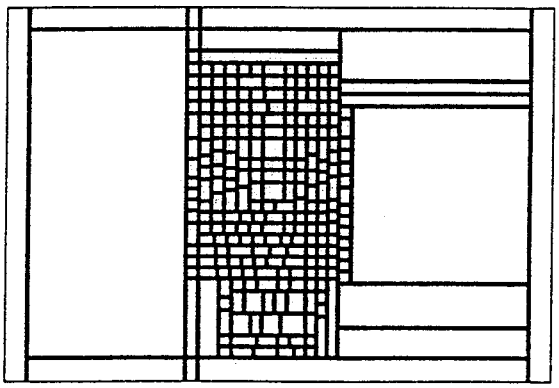
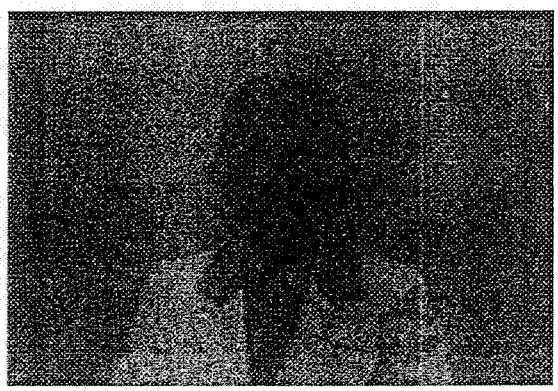
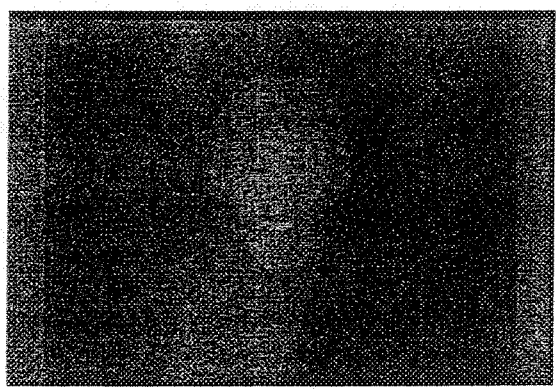


Fig. 5 (a) – (c) The partitioning mask corresponding to the feature map of the Claire sequence in Fig.3 (a) – (c), respectively. (d) The final partitioning mask obtained after grouping the three Y-C_b-C_r masks in (a) – (c).

(a)



(b)

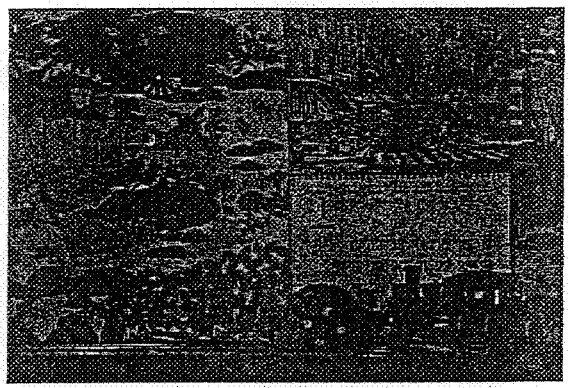


(c)

Fig. 6 (a) – (c) The decompressed Y, C_b and C_r color components of the first frame in the color Claire sequence.

(a)

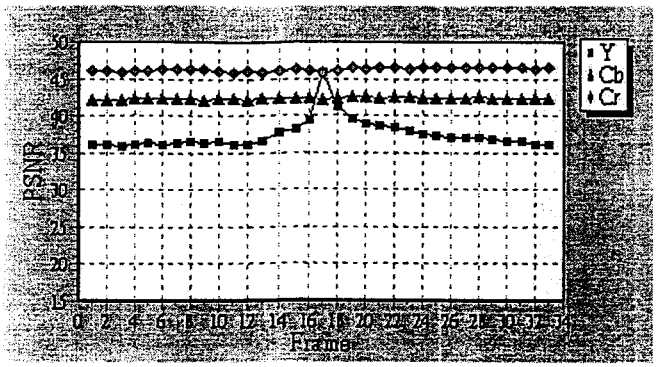
(b)



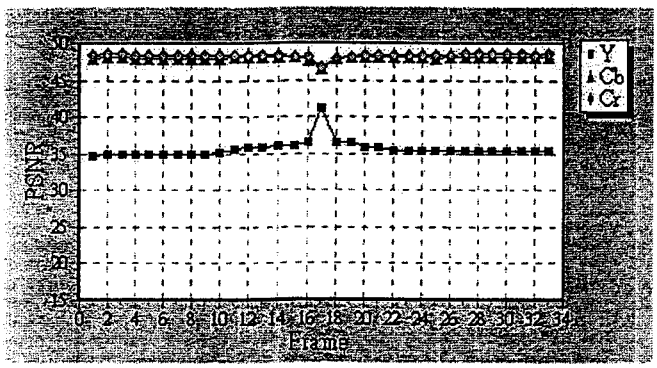
(c)

(d)

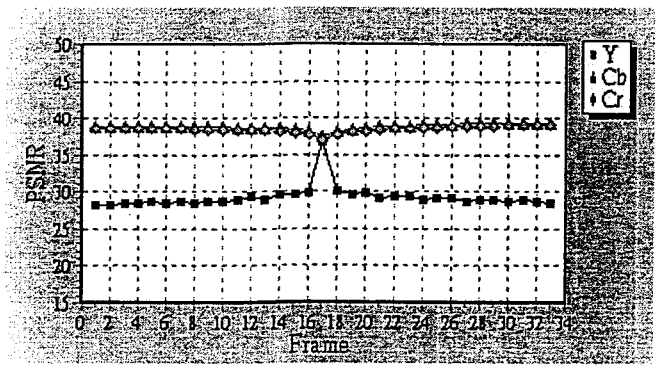
Fig. 7 The reference frame, also the 17th frame, in a GOP consisting of 33 consecutive frames, for the test sequence: (a) Claire, (b) Susie, (c) Train, and (d) Football.



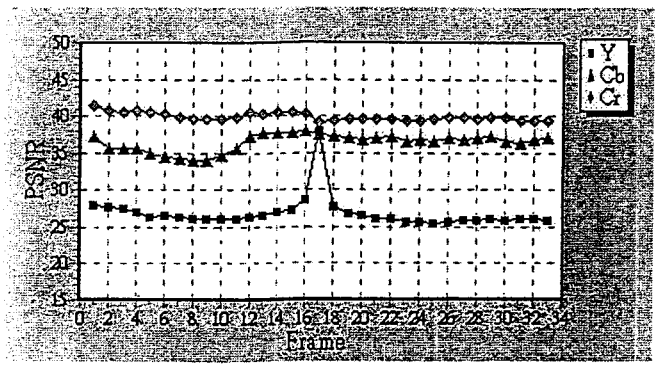
(a)



(b)



(c)



(d)

Fig. 8 The *PSNR* of each color component for every frame in the test sequence: (a) Claire, (b) Susie, (c) Train, and (d) Football.

REFERENCES

[1] M. Goldberg and H. Sun, "Image Sequence Coding Using Vector Quantization," *IEEE Trans. Commun.*, vol. COM-34, pp. 703-710, 1986.

[2] B. G. Haskell, F. W. Mounts, and J. C. Candy, "Interframe coding of videotelephone pictures," *Proc. IEEE*, vol. 60, pp. 792-800, July 1972.

[3] J. L. Mitchell, *MPEG video: compression standard*, New York: Chapman & Hall, 1997.

[4] W. F. Mount, "A video encoding system using conditional picture elements replenishment," *Bell Syst. Tech. J.*, vol. 48, pp. 2545-2554, Sept. 1969.

[5] H. H. Nagel, "Displacement vectors derived from second order intensity variations in image sequences," *Computer Vision, Graphics, and Image Processing*, pp. 85-117, 1983.

[6] H. Schweitzer, "Occam Algorithms for Computing Visual Motion," *IEEE Trans. Pattern Anal. And Machine Intell.*, vol. 17, No. 11, pp. 1013-1042, 1995.

[7] R. F. Chang and W. M. Chen, "Interframe Difference Quadtree Edge-Based Side-Match Finite-State Classified Vector Quantization for Image Sequence Coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol.6, No. 1, pp. 32-41, 1996.

[8] A. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Trans. Image Processing*, vol. 1, no. 1, pp. 18-30, Jan. 1992.

[9] M. S. Lazar, L. T. Bruton, "Fractal Block Coding of Digital Video," *IEEE Trans. Circuits and Systems for Video Technology*, vol.4, no.3, pp. 297-308, June 1994.

[10] Falconer, K. J., *Fractal Geometry: Mathematical Foundations and Applications*, Chichester ; New York : Wiley, 1990.

[11] B. B. Mandelbrot, *Fractal Geometry of Nature*, New York: W.H. Freeman, 1983.

[12] Alex P. Pentland, "Fractal-Based Description of Natural Scenes, " *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, No.6, pp. 661-674, Nov. 1984.