論文 95-4-4-8

Performance analysis and implementation issues of segmentation image coder using the fractals and the human visual system

Jong Whan Jang*, Jae Gil Jeong*, Doo Yeong Park*, Woo Suk Yang** Regular Members

Fractal과 Human Visual System을 이용한 세그멘테이션 영상코더의 성능분석과 구현문제에 관한 연구

正會員 張 鍾 煥*, 鄭 在 吉*, 朴 斗 暎*, 梁 雨 錫**

이 논문은 1993년도 한국학술진흥재단의 공모과제 연구비에 의하여 연구되었음

ABSTRACT

A new texture segmentation-based image coding technique which performs segmentation based on roughness of textural regions and properties of the human visual system (HVS) is presented. The segmentation is accomplished by thresholding the fractal dimension so that textural regions are classified into three texture classes: perceived constant intensity, smooth texture, and rough texture. An image coding system with high compression and good image quality is achieved by developing an efficient coding technique for each segment boundary and each texture class. We compare the coding efficiency of this technique with that of a well established technique (discrete cosine transform (DCT) image coding) for two different types of imagery; a head and shoulder image with little texture variation and a complex and natural outdoor image with highly textured areas. Also the implementation issues of the proposed image coder are discussed.

要 約

인간 시각 시스템의 특성과 텍스처 영역의 거친 정도에 따라 영상을 분할하는 새로운 텍스처 분할 영상코딩 기술 (new texture segmentation-based image coding technique)을 제안한다. 영상분할은 프랙탈 차원을 임계하여 얻는다. 따라서 텍스처 영역이 3 가지의 다른 텍스처 클래스 (인간이 인지한 상 인텐서티, 부드러운 텍스처 및 거친 텍스처)로 구분된다. 고압축 및 고화질의 영상코딩 시스템을 위해 각각의 세그멘트 경계면과 텍스처 클래스에 효과적인 영상코딩 기술을 개발한다. 제안된 기술과 널리 사용하고 있는 DCT 영상코딩 기술의 효율을 2 가지의 다른 종류의 영상 (텍스처 변동이 거의 없는 head and shoulder 영상 및 복잡하고 거친 텍스처가 있는 natural outdoor 영상)에 대하여 비교 분석하고 제안된 영상코더의 구현문제에 대하여 논의한다.

*배재대학교
**홍익대학교 전기공학과
論文番號 : 94203-0729
接受日字 : 1994年 7月 29日

Ⅰ. Introduction

The digital representation of an image requires a very large number of bits. For example, a 512× 512 pixel, 256 gray level digital image requires over two million bits. This large number of bits is a substantial drawback when it is necessary to store or transmit a digital image. Prior to transmission or storage, one would like to have a system that reduces this number as much as possible, while keeping the degradation in the decoded image to a minimum. This is the goal of image compression, often referred to as image coding.

Classical image coding technique [9,11] solely guided by information theory, led to a plethora of methods. The compression ratio, starting at one with the first digital picture in the early 1960's, appeared to have reached a saturation level around 10:1 in the early 1980's. This, however, did not mean that the upper bound given by the entropy of the source had also been reached. First, this entropy is not known and depends heavily on the model used for the source, i.e., the digital image. Second, information theory does not take into account what the human eye sees and how it sees. Recently, techniques attempting to overcome these limitations are incorporating properties of the fractals and the human visual system (HVS) and tools of image analysis into image compression to achieve high compression ratios with small loss in visual quality. This reasoning follows from the fact that in many compression applications, a human is the final observer of the image operated upon. Applications of various models of the fractals and the HVS have in fact been empirically found to improve compression performance [2,5,6,12].

One such technique is segmentation-based image compression [2,6,7]. In segmentation-based image compression, the image to be compressed is segmented, i.e., the pixels in the image are separated into mutually exclusive spatial regions based on some criteria. Once the image has been segmented, information is extracted describing the boundaries (shapes) and textures (interiors) of the image segments, and compression is achieved by efficiently encoding this information. Unfortunately, there are limitations with segmentation-based image compression. The main limitation is due to the fact that the image data have been segmented into regions of constant intensity. In complicated texture areas, a good representation of the texture requires many small segments. However, in order to get low bit rates, the number of segments must be limited and thus the quality is degraded.

A new texture segmentation-based image coding technique proposed by Jang and Rajala [6] solves the texture representation problem by proposing a methodology for segmenting an image into texturally homogeneous regions with respect to the degree of roughness as perceived by the HVS using the properties of the fractals and the HVS. The main three measures to characterize the texture information are the fractal dimension, the expected value, and the just noticeable difference (JND). The segmented image information is then encoded for transmission.

In this paper, we compare the coding efficiency of this technique with that of DCT image coding technique for two different types of imagery. The first is a head and shoulder image with little texture variation. This image is typical of video teleconferencing applications. The second is a complex and natural outdoor image with highly textured areas. Also the implementation issues of this technique are discussed.

We describe a new texture segmentation-based image codec and its major components and a discrete cosine transform (DCT) image codec in sections Ⅱ and Ⅲ respectively. In section Ⅳ, performance comparison is evaluated using computer simulated data of actual images. In section Ⅴ,

implementation issues of the texture segmentation-based image coding technique is discussed in terms of parallel processing. Finally, conclusions are provided in section Ⅵ.

Ⅱ. A texture segmentation-based image codec incorporating the fractals and the HVS

A block diagram of the proposed texture segmentation-based image coding system is presented in figure 1. Figure 1(a) shows a block of the transmitter including three main stages; the preprocessor, the segmenter, and the encoder.

The main purpose of the preprocessor which is the first stage of the proposed transmitter in figure 1(a) is to alter the image in such a way that fewer segments and textures are produced by the segmenter, but without degrading the visual quality of the segmented image. Thus, the number of image segments and the number of bits representing the textures of the segments are directly proportional to the bit rate of the coded image. Because of this, a minimum number of segments and an efficient representation of the textures are critical.

After preprocessing, the next step in the compression algorithm is the segmentation of the image. In this work, centroid-linkage region growing [4] is used. An important attribute of region growing segmentation is the production of disjoint segments with closed boundaries. This is important because segmentation-based compression requires a description of the boundary and texture of each

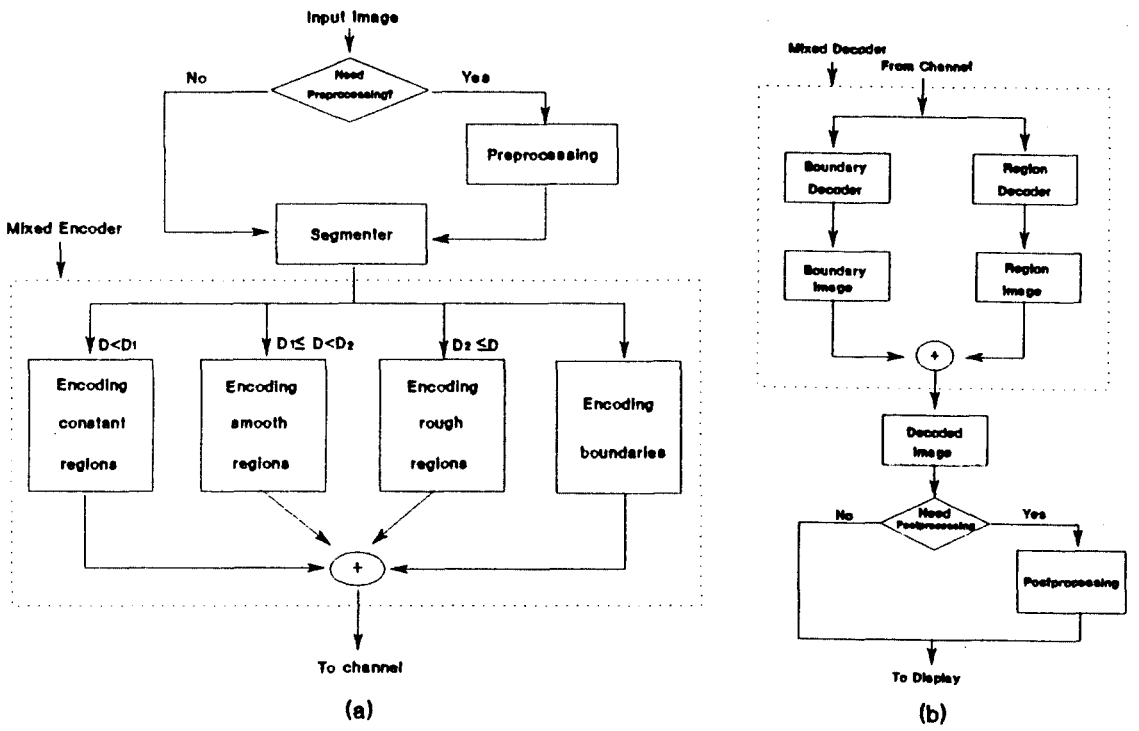


Fig. 1. The texture segmentation-based image coding system
(a) The block diagram of the transmitter
(b) The block diagram of the receiver

image segment. Such a description would be impossible if the segments overlapped or did not have closed boundaries. The texture features used to characterize the texture information are the mean, the JND (just noticeable difference), and the fractal dimension of an image block. The proposed texture segmentation algorithm is given in the table 1. In the segmentation algorithm, one property of the HVS we use is contrast sensitivity. Contrast sensitivity is a measure of the HVS's ability to distinguish two adjacent intensity patches. To obtain the contrast sensitivity, a subject is shown a test pattern, for example, two adjacent

Table 1. Texture-Based Image Segmentation Algorithm

Incorporating the HVS and the fractal model, the proposed texture-based image segmentation algorithm for image coder is defined as follows.

Step 1) Divide the image into NR×NC blocks (NR and NC are the numbers of row and column blocks, respectively).

Step 2) Calculate the feature set: the mean and the class type for each block and the JND lookup table.

Step 3) Calculate the distance between an observing block and its 4-connected neighboring blocks. The distance is given by

$$D(OB,NB)=\begin{cases} 0 & \text{if} \begin{cases} F(OB)<D_1,\ C(OB)=C(NB), \\ |M(OB)-M(NB)|<JND(OB,NB) \\ \text{or} \\ D_1\le F(OB)<D_2,\ C(OB)=C(NB) \\ \text{or} \\ F(OB)\ge D_2,\ C(OB)=C(NB) \end{cases} \\ 1 & \text{otherwise} \end{cases}$$

where F(OB) is the fractal dimension of an observing block. C(OB) and C(NB) are the class types of an observing block and its neighboring block respectively. M(OB) and M(NB) are the means for an observing block and its neighboring block respectively. JND(OB,NB) is the just noticeable difference between an observing block and its neighboring block.

Step 4) If there is a neighboring block with distance 0, then merge the observing block into it; else declare a new region. If there are more than two good neighboring blocks, merge the observing block into a neighboring block whose mean value is closest to the mean value of the observing block.

Step 5) Repeat step 3 to step 4 until all blocks are segmented and stop.

squares. The luminance of one square is varied until it is just noticeably different in luminance than the adjacent square. The JND between the two luminance values is used to calculate the contrast, see [3] for details. It has shown that the HVS has greatly reduced contrast sensitivity in very bright or very dark intensity regions of an image. Here the contrast sensitivity is used in defining the threshold for the split-merge condition for regions belonging to the perceived constant intensity in the proposed texture segmentation-based compression technique.

To determine the split-merge condition between the regions belonging to class I, we use a visual threshold based on the HVS properties in the segmentation algorithm. The HVS-based threshold is adapted to local intensity characteristics in the image by using the JND as the visual threshold.

A split-field method is used to measure the JND on SUN SPARC workstation with 1024×768 19″ color monitor. The image display device is divided down the middle into two equal-size fields. The left field is a constant reference intensity and increases linearly up to 40 steps above the constant reference intensity. To perform the tests, the viewer simply clicks the mouse at the point where the difference between the left and right fields is no longer discernible. This point is the JND between the reference intensity on the left and the

test intensity on the right.

Five test subjects were asked to take a measurement of JND. Each test subject sat at a distance of six times the image height away from the screen. The test subject was given approximately three minutes before the start of the experiments, to allow for adaptation to the laboratory's illumination. The test subject was asked to take five seconds in each click to allow for adaptation to the screen. The average of the results of five test subjects is shown in figure 2. It is seen that the experimental results agrees with the HVS contrast sensitivity properties [3]. The JND is largest in the lowest and highest intensity areas of the image. The JND is smallest and nearly constant in the middle intensity areas of the image. To determine threshold, the approximated JND curve is derived in figure 2. The bold line corresponds to the approximated JND curve. Based on this result, a larger threshold is chosen for the lowest (0 to 74) and highest intensity areas (241 to 255), while a smaller threshold (about 5) is used in the middle intensity area (75 to 240).

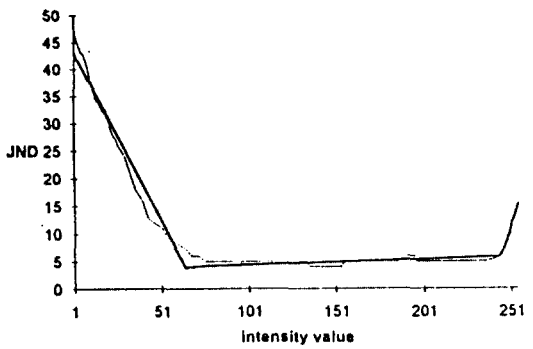


Fig. 2. The mean of five subjects' JND measurements on SUN SPARC workstation with 1024×768 19" color monitor. The bold line corresponds to the approximated JND curve.

An image is segmented into texturally homogeneous regions with respect to the degree of roughness as perceived by the HVS [1]. The segmentation is accomplished by the thresholding the fractal dimension so that membership is in one of three textural classes. The three classes are perceived constant intensity (class Ⅰ), smooth texture (class Ⅱ), and rough texture (class Ⅲ). Regions belonging to the perceived constant intensity have a fractal dimension less than D_1. The second class contains regions with the fractal dimension between D_1 and D_2. The third class contains regions with the fractal dimension greater than D_2.

To estimate the fractal dimension, we choose the blanket method [11] since it is computationally efficient. A brief explanation of the procedure for estimating the fractal dimension is given here. All points in the three-dimensional space at distance ϵ from the surface are considered, covering the surface with a blanket of thickness 2ϵ. The surface area $A(\epsilon)$ is then the volume $V(\epsilon)$ occupied by the blanket divided by 2ϵ. The area $A(\epsilon)$ is given by

$$A(\varepsilon) = \frac{V(\varepsilon)}{2\varepsilon} = \varepsilon^2 \cdot N(\varepsilon) = K \cdot \varepsilon^{2-D} \quad (1)$$

where K is a constant.

To compute the fractal dimension, we apply the log function to both sides of Eq. (1). The following equation is given by

$$\log A(\epsilon) = (2-D)\log(\epsilon) + K \quad (2)$$

From Eq. (2), we can deduce a procedure to estimate the fractal dimension of an image surface using a least-squares linear regression.

The output of the segmenter is a gray level image consisting of many segments. The images are partitioned so that each segment contains the same degree of roughness as perceived by the HVS. The objective now is to apply an efficient coding technique to the boundaries and each texture class to achieve high compression with small visual degradation.

The last stage in the transmitter is the mixed encoding of the segments of each class and their boundaries. During the segmentation, the segments are classified as one of the three texture

classes. The objective of the coding is to obtain an efficient representation of the segmented image data for transmission or storage. The image coder should use more bits to encode the information for which the HVS is more sensitive and use fewer bits to encode the information which the HVS is less sensitive. To accomplish this we propose a mixed encoder. It consists of four separate stages: the boundary encoding and three textural class encodings, see figure 1(a).

For boundary coding, accurate representation of the boundary is necessary to describe the location of the region boundary because of the HVS sensitivity of the edges. We choose an errorless coding scheme to represent the boundaries. A binary image representing the boundaries is created. Then, the binary data is encoded using an arithmetic code since it has been found to be superior to Huffman code, runlength code, and crack code [15].

For regions belonging to perceived constant intensity, only the mean intensity values need be transmitted to describe the textures of the regions. In this case, lossy compression has already taken place since we are approximating each region texture with a constant value. We do not wish to introduce any further compression, so that the lossless arithmetic code is again employed to achieve further compression. Since a mean intensity requires 8 bits, the mean values are converted into a 8×N binary array, where N is the number of segments belonging to perceived constant regions. The mean vector is then encoded using an arithmetic code.

Regions belonging to smooth and rough texture are not directly encoded. To get higher compression, their regions are modeled first using polynomial functions. The coefficients of the polynomial functions are encoded because the variance of the coefficients is less than that of the original data. An arithmetic code is used to encode the coefficients. A smaller amount of error between the original image data and the modeled image data is chosen for smooth texture than rough texture because of the sensitivity of the HVS. In general, modeling these regions by functions of higher order polynomials is computationally excessive. The first order polynomial functions are used because the sum squared error (SSE) for the first order polynomial functions as given in Eq. (3) is not much greater than that of the SSE for a second-order polynomial function, while the SSE for the first order polynomial function is much greater than the SSE for the zero order polynomial function [2].

$$\sum_i \sum_j (g(i,j) - z(i,j))^2. \tag{3}$$

where g(i, j) and z(i, j) are the intensity of the original image and the modeled image at index (i, j) respectively.

At the receiver given in figure 1(b), two types of coded information come into the mixed decoder, boundary information and region texture information. The boundary decoder must regenerate the boundaries for the decoded image. The regions decoder must fill in the missing texture information with each region.

The missing part of the decoded image after the boundaries are decoded is texture information within regions. Since regions belonging to class Ⅰ are perceived constant regions, their mean values are painted within the appropriate regions. Regions belonging to class Ⅱ and Ⅲ are reconstructed by reproducing polynomial functions. The information of the boundaries and the textures are combined to form the reconstructed image.

Ⅲ. A discrete cosine transform image codec.

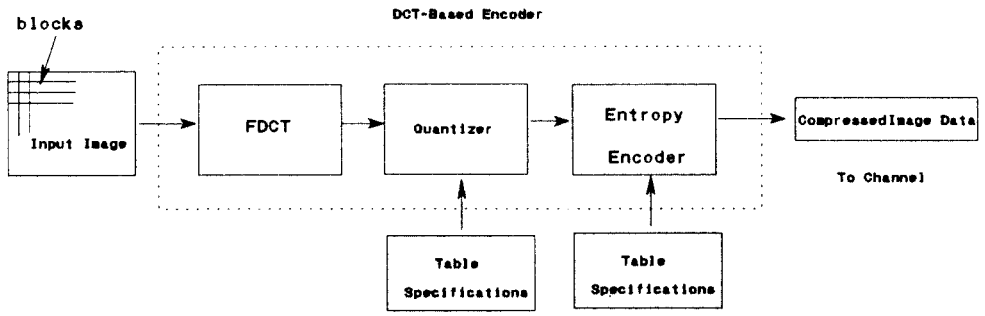
A block diagram of the DCT image coding system is given in figure 3 [14]. In the encoding and

decoding process the input image is processed in blocks. Figure 3(a) shows a block diagram of the transmitter including three main stages: F(forward)DCT, quantizer, and entropy encoder. The advantage of the DCT is that it is easy to evaluate because it does not use complex-number arithmetic; there are fast algorithm for calculating it, and it is more efficient at signal decorrelation than most other transforms.

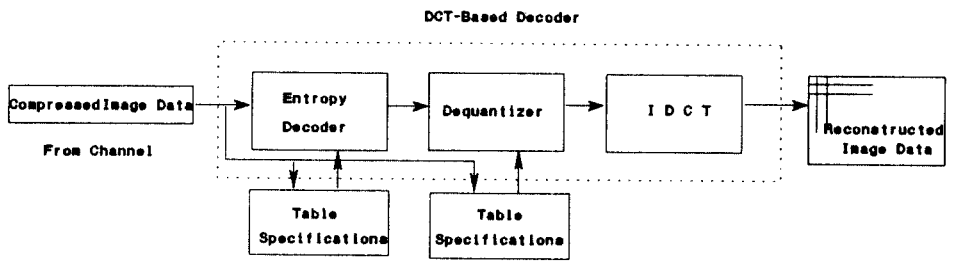
In the encoding process the input image data are grouped into blocks, and each block is transformed by the forward DCT (FDCT) into a set of values referred to as DCT coefficients. One of these values is referred to as the DC coefficient and the others as the AC coefficients. Each of the coefficients is then quantized using one of corresponding values from a quantization table.

After quantization, the DC coefficient and the AC coefficients are prepared for entropy encoding. The previous quantized DC coefficient is used to predict the current quantized DC coefficient, and the difference is encoded. The quantized AC coefficients undergo no such differential encoding, but are converted into a one-dimensional zig-zag sequence. The quantized coefficients are then passed to an entropy encoding procedure which compresses the data further. Arithmetic coding conditioning table specification is used.

Figure 3(b) shows the main procedures for the DCT decoding processes. Each step shown performs essentially the inverse of its corresponding main procedure within the encoder. The entropy decoder decodes the zig-zag sequence of quantized DCT coefficients. After dequantization the DCT coefficients are transformed to an block by the inverse DCT (IDCT).



(a) The block diagram of the transmitter



(b) The block diagram of the receiver

Fig. 3. The DCT image coding system

Ⅳ. Performance analysis

To compare this texture segmentation-based image coder with DCT image coder, two different types of imagery are coded. The two test images are shown in figure 4. The first is a head and shoulder image. This image is typical of teleconferencing applications. The second is a complex and natural outdoor image with many edges and highly textured areas. Each image consists of 256×256 pixel with 256 gray levels. The 8×8 block size is used for two techniques. D_1 and D_2 for the proposed segmentation-based image coding system are used as 2.042 and 2.411 respectively.

The decoded images of the two test images are shown in figure 5 and 6 using the proposed image coding algorithm and the DCT algorithm. Figure 5 shows the decoded images of Miss USA. The decoded images of the proposed image coding system and the DCT are on the left and right respectively. The compression rate (CR) is around 0.105 bpp for two techniques. The block effects of the decoded image of Miss USA based on the DCT are more prominent than that of the decoded image based on the texture segmentation-based technique. The areas of hairs, eyes, and noses of the decoded image based on the DCT have prominent blocking effects. Figure 6 shows the decoded images of House. The decoded images of the proposed image coding system and the DCT are on the left and right respectively. CR is around 0.16. The image quality of the decoded images is better for the proposed image coding system. Since the texture segmentation-based technique incorporates the texture characteristics based on the HVS, visual loss in highly textured areas such as trees on the left and right and roofs of the house seems to be small to a human. The block effects of the decoded image of the House based on the DCT are more prominent than that of the decoded image based on the proposed texture segmentation-based technique. Comparison of CR/SNR table for the two techniques is given in the table 2. It shows that the proposed texture segmentation-based image coder performs better the DCT image coder in terms of SNR for low and high bit rates.

Fig. 4. The two test images. Each image is 256×256 pixel with 256 levels. Miss USA and House are on the left and right respectively.

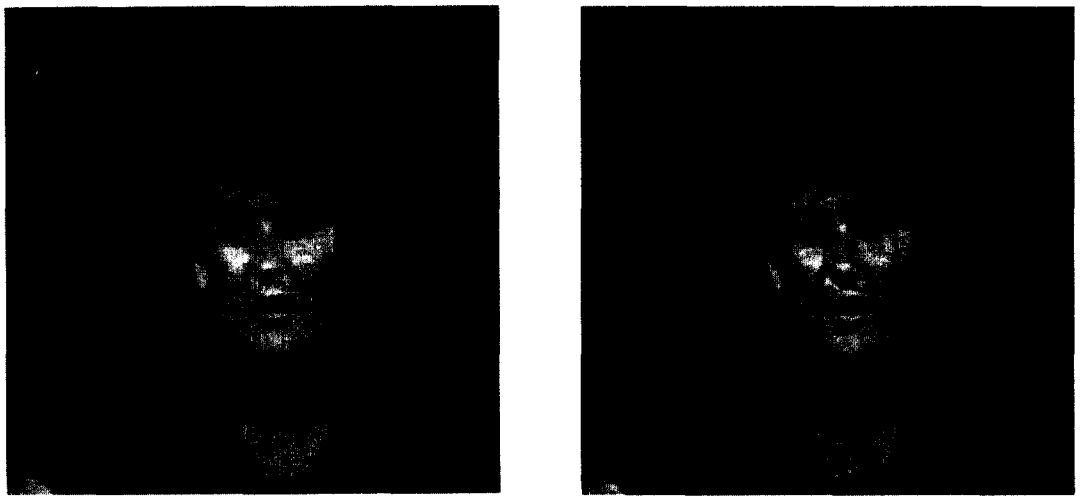


Fig. 5. The decoded images of Miss USA. The decoded images of the proposed image coding system and the DCT are on the left and right respectively.

Fig. 6. The decoded images of House. The decoded images of the proposed image coding system and the DCT are on the left and right respectively.

Table 2. Comparison of CR/SNR for the proposed and the DCT image coding systems. CR stands for compression ratio.

CR	Miss USA		House	
	Proposed	DCT	Proposed	DCT
8	27.3	26.5	15.5	14.0
10	26.4	25.6	14.8	13.2
20	25.9	24.5	12.0	10.8
40	21.2	19.4	9.7	8.5
80	18.1	15.2	8.5	6.3
100	15.2	13.1	6.9	5.8

V. The implementation issues

Even though the proposed coding algorithm shows better performance, we have to develop the advanced implementation techniques to utilize the advantages in real-time image and video coding system. There are three major parts that we have to consider for the efficient implementation of real-time video coding systems.

The first part is the preprocessing. Preprocessing part includes the preconditioning of input signal such as linear filtering and/or nonlinear filtering. Also, some limited form of the data reduction techniques such as subsampling considering the characteristics of human visual system (for example, human visual system is less sensitive to the high frequency signal and color signals) can be used. Many parallel algorithms and parallel processing techniques are available for the purpose of preprocessing such as lowpass filtering, median filtering, and so on.

The second part is the image segmentation. This is the critical part for the presented algorithm to be useful for the real-time video and image processing system. The presented algorithm has many parallelism. However it is difficult to utilize these parallelism for the high throughput because of the characteristics of the algorithm such as global data communication requirements, shared memory accesses, irregular computing structures and so on.

Since the presented algorithm is actually based upon the fixed size rectangular block given in figure 7., we can arrange the operations for each block to be regular, and we can design a processor which can perform these operations efficiently. If we arrange these processors properly and use interprocessor communication links properly between processors, several blocks can be processed concurrently with several processors.

We suggest the parallel structure shown in figure 8 which utilizing that all data transfers are only occurring between neighboring blocks (see table 1, M(NB), C(NB), and JND(OB,NB) are required to compute the distances of neighboring blocks), and these transfers are occurring with a fixed amount and fixed directions. It is the desirable characteristics for the parallel architecture, especially for the VLSI implementation.

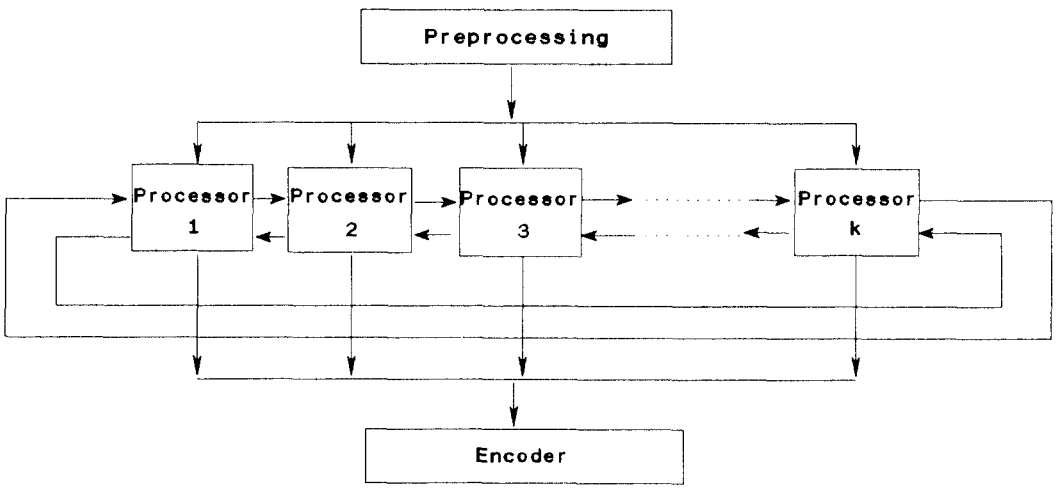


Fig. 6. The block partition of an image

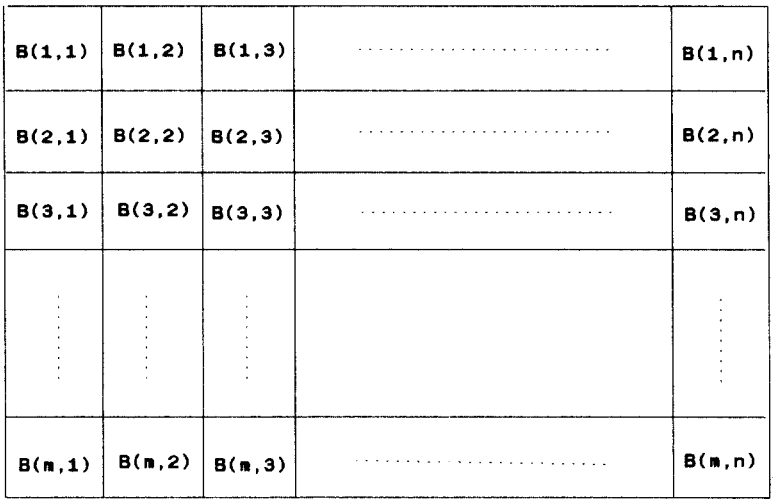


Fig. 7. The parallel architecture for the proposed segmentation algorithm

This system also has scalability, i.e., if we need more speed we can add more processors for higher throughput. It is possible because there is no bottleneck for the interprocessor communications and generally the number of blocks are larger than the number of processors.

The architecture shown in figure 8 works as follows. If k=4, processor 1 processes first row of blocks, processor 2 processes the second row of blocks, and so on. And 5th row is processed by the processor 1, and so on. In this scheme, each processor computes fractal dimension, class, mean, JND, and other required operations for the assigned block, stores the results for the processing of next row, and transfers the results to the two neighboring processors (left and right).

We can also partition the computations of each block into several computations such as computation of fractal dimension, computation of classes of blocks, computation of mean of the block, and computation of JND. With these, each processor can be built with several pipelines to obtain high throughput.

The third part is encoder, we can utilize 4 parallel paths shown in figure 1(a), and parallelism inside of each region. For this part, sophisticated bit-allocation and rate control schemes are needed to obtain the best results for the fixed bit rate. This part is directly related to the bit-allocation scheme, bit rate, whether VBR (variable bit rate) or CBR (constant bit rate) scheme is used [10].

If we study details of implementation issues presented here, we can implement real-time high quality video compression systems utilizing segmentation-based compression algorithm, which can be used in many applications such as video phone, video conference, medical imaging system, facsimille, machine vision, image classification and so on [8].

Ⅵ. Conclusions

In this paper, we propose a new texture segmentation-based image coding technique using the fractals and the HVS, which achieves compression in the neighborhood of 0.1 to 0.2 bpp. The seg-

mentation is good and conforms to the human perception of roughness. Through our experiment, the proposed method works well for a variety of images, and performs better the DCT images coder in terms of visual image quality and SNR. We discuss the implementation issues of the proposed texture segmentation-based image coding system in terms of parallel processing for the real-time system.

References

1. M. Barnsley, *Fractal everywhere*, Academic Press, Inc. 1988.
2. M. J. Biggar, O. J. Morris, and A. G. Constantinides, "Segmented-image coding: performance comparison with the discrete cosine transform," *Proc. IEE*, Part F, Vol. 135(2), pp.121-132, April. 1988.
3. T. N. Cornsweet, Visual Perception, New York: Academic Press, 1771.
4. R. M. Haralick and L. G. Shaprio, "Image segmentation techniques," *Compt. Graphics Image Processing*, Vol. 29, pp.100-132, 1985.
5. A. E. Jacquin, "A novel fractal block-coding technique for digital images," *In Proc. ICASSP'90*, Vol. 4, pp.2225-2228, April. 1990.
6. J. W. Jang and S. A. Rajala, "Texture segmentation-based image coder incorporating properties of the human visual system," *In Proc.* ICASSP'91, May. 1991.
7. J. W. Jang, "Performance evaluation of the texture segmentation-based image coding techniques using the fractals," *IEEE Trans. on Image Processing*, Submitted, July. 1994.
8. K. Dezhgosha, M. M. Jamali, and S. C. Kwatra, "A VLSI architecture for real-time image coding using a vector quantization based algorithm," *IEEE Trans. on Signal Processing*, Vol. 40, No. 1, pp.181-189, Jan. 1992.
9. M. Kunt, M. Benard, and R. Leonardi, "Recent results in high compression image coding," *IEEE Trans. on Circuit Syst.*, Vol. CAS-34, No. 11, pp.1306-1336, Nov. 1987.
10. N. M. Marafih, Y. Q. Zhang, and R. L. Pickholtz, "Modeling and queueing analysis of variable-bit-rate coded video sources in ATM networks," *IEEE Trans. on Circuit and Systems for Video Technology*, Vol. 4, No. 2, pp.121-128, April. 1994.
11. H. G. Musmann, P. Pirsch, and H. J. Grallert, "Advances in picture coding," *Proc. IEEE*, Vol. 73, pp.523-548, April. 1985.
12. K. N. Ngan, K. S. Leong, and H. Singh, "Adaptive cosine transform coding of images in perceptual domain," *IEEE Trans. on ASSP*, Vol. 37, No. 11, pp.1743-1750, 1989.
13. S. Peleg, J. Naor, and R. Hartley, and D. Avnir, "Multiple resolution texture analysis and classification," *IEEE Trans. on Pattern Anal. Machine Intell.*, Vol. 6, No. 4, pp.518-523, July. 1984.
14. K. R. Rao, *Discrete cosine transform algorithm, advantages, applications*, Academic Press, 1990.
15. J. Rissanen, "Generalized Kraft-inequality and arithmetic coding," *IBM J. Res. Devel.*, Vol. 20, pp.198-203, 1976.

張鍾煥 (Jong Whan Jang)정회원

1979년 2월 : 한양대학교 전자통신공학과 졸업(공학사)

1986년 5월 : 미국 North Carolina State University 전기 및 컴퓨터공학과 졸업(공학석사)

1990년 12월 : 미국 North Carolina State University 전기 및 컴퓨터공학과 졸업(공학박사)

1990년 5월~현재 : 배재대학교 정보통신공학과 조교수

1992년 3월~현재 : 배재대학교 전자계산소장

※ 주관심분야 : 영상통신 및 영상처리, 마이크로프로세서응용

鄭在吉 (Gae Gil Jeong) 정회원

1980년 2월 : 한양대학교 전자공학과 졸업(공학사)

1987년 5월 : 미국 North Carolina State University 전기 및 컴퓨터공학과 졸업(공학석사)

1991년 8월 : 미국 North Carolina State University 전기 및 컴퓨터공학과 졸업(공학박사)

1979년 5월~1985년 6월 : 국방과학연구소 연구원

1991년 8월~1992년 8월 : 한국전자통신연구소 선임연구원

1992년 9월~현재 : 배재대학교 전자공학과 조교수

※ 주관심분야 : 병렬처리구조, 영상통신 및 영상처리, 마이크로프로세서응용, 디지탈신호처리

朴斗暎 (Doo Yeong Park)정회원

1981년 2월 : 한양대학교 전자공학과 졸업(공학사)

1987년 5월 : 미국 North Carolina State University 전기 및 컴퓨터공학과 졸업(공학석사)

1993년 8월 : 미국 North Carolina State University 전기 및 컴퓨터공학과 졸업(공학박사)

1993년~1994년 : 한국전자통신연구소 선임연구원

1994년 3월~현재 : 배재대학교 정보통신공학과 조교수

※ 주관심분야 : 초고속광대역통신망, ATM스위치성능평가, 데이타통신, 컴퓨터네트워크

梁雨錫 (Woo Suk Yang) 정회원

1979년 2월 : 서울대학교 전기공학과 졸업(공학사)

1986년 6월 : 미국 University of Toledo 전기공학과 졸업(공학석사)

1991년 5월 : 미국 North Carolina State University 전기 및 컴퓨터공학과 졸업(공학박사)

1978년 11월~1984년 8월 : (주)대우

1990년 12월~1991년 8월 : 금성사 책임연구원

1991년 8월~현재 : 홍익대학교 전기공학과 조교수

※ 주관심분야 : 로보틱스, 컴퓨터비젼, FA