The Journal of Natural Science,
Pai Chai University, Korea
Volume 7: 67~73, 1995

프락탈 디멘션을 근사하기 위한 적당한 브록 크기 결정에 관한 연구

장종환
배재대학교 이공대학 정보통신공학과

Determination of the Proper Block Size for Estimating the Fractal Dimension

Jongwhan Jang
Dept. of Information and Communication Engineering, PaiChai University

본 논문에서는 인간시각 시스템의 특성을 이용하여 세그멘테이션을 행하는 새로운 텍스처 세그멘테이션 영상코딩 기술을 제안한다. 제안된 방법은 영상을 HVS가 인지한 러프니스 정도에 관하여 영상을 텍스처의 같은 성질의 영역으로 세그멘테이션하는 방법론을 제안하여 상 세그멘트를 갖는 세그멘테이션 영상코딩기술의 문제점을 해결한다. 세그멘테이션은 텍스처 영역을 3 가지의 다른 텍스처 크래스(인간이 인지한 상 인테셔티, 부드러운 텍스처 및 거칠은 텍스처)로 구분하기 위해서 프락탈 디멘션을 임계하여 얻는다. 프락탈 디멘션을 근사하기 위한 적당한 브락 크기를 결정하는 것이 중요하다. 좋은 화질을 갖는 재생 영상은 여러 종류의 영상에 대해서 약 0.1에서 0.25 비트/픽셀에서 얻는다.

In this paper, a new texture segmentation-based image coding technique which performs segmentation based on properties of the human visual system (HVS) is presented. This method solves the problems of a segmentation-based image coding technique with constant segments by proposing a methodology for segmenting an image into texturally homogeneous regions with respect to the degree of roughness as perceived by the HVS. 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. It is very important to determine the proper block size for estimating the fractal dimension. Good quality reconstructed images are obtained with about 0.1 to 0.25 bit per pixel (bpp) for many different types of imagery.

Keywords: block, segmentation, fractal dimension, decoded, compression, quality.

1. Introduction

Many data processing applications require storage of large volume of data. The number of data processing applications such as in the areas of electronic publishing, video conferencing etc is increasing rapidly. At the same time, there has been a proliferation of computer communication networks and teleprocessing applications.

Any image compression method can be broadly classified as being either statistically -based (algebraic) or symbolically-based (structural). The statistically-based image compression techniques address the image compression problem from an information theory viewpoint, with the focus on eliminating the statistical redundancy among the pixels in the image. Symbolically-based image compression techniques employ properties of the HVS and tools of image analysis to achieve good image quality at very low data rates. Applications of various models of the HVS have in fact been empirically found to improve compression performance.[5,6,8]

One approach to symbolically-based image compression techniques is segmentation-based image compression.[2,5,7] 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. Properties of the HVS can also be incorporated into the criteria to obtain a reconstructed image with a small visual loss. 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. Segmentation is an important step in image coding. It is important to design the segmentation algorithm so that image segments are allocated in a way that achieve high compression with small visual quality loss. In the proposed new technique, this is accomplished by incorporating properties of the HVS at various stages in the segmentation algorithm. The segmentation technique we present segments an image into texturally homogeneous regions with respect to the degree of rouhgness as perceived by the HVS. The segmentation algorithm uses a variation of centroid-linkage region growing. The region growing is directed by the texture distance measure between image blocks. The meaure of the roughness of the textural regions is represented by the fractal dimension. In the actual segmentation, the fractal dimension is thresholded so that the textural regions are classified into three textural classes; perceived constant intensity, smooth texture, and rough texture. Analysis of the sensivity of the calculation of fractal dimension to block size is described.

We overcome the texture representation problem and present the determination of the proper block size in this paper by proposing a methodology for segmenting an image into texturally homogeneous regions with respect to the degree of roughness as perceived by the HVS. After segmentation, the image can be viewed as being composed of region boundaries and texturally homogeneous regions. As image coding system with high compression and good image quality is achieved by developing an efficient coding technique for the region boundaries and the three textural classes. The proposed algorithm is applied to different types of imagery.

In section 2, we describe the image segmentation. In section 3, determination of the proper block size for estimating the fractal dimension is presented. Finally, conclusions are provided in section 4.

2. Image segmentation

The goal of the image segmentation process is to decompose an image into texturally homogeneous regions with respect to the degree of roughness as perceived by the HVS. Textural regions are classified into three classes; perceived constant intensity, smooth texture, and rough texture. For example, the background in a head and shoulder image or the sky in a natural image is considered as perceived constant intensity, the face or the shoulder is considered as smooth texture, and the trees and the bushes in a natural image are considered as rough texture. To extract texture information for accomplishing textural-based image segmentation, the fractal dimension, mean, and just noticealbe difference are used in the segmentation algorithm. The segmentation algorithm is based on a region growing technique. A unique of feature of the region growing process used in this research is that it is directed by the texture feature distance between image blocks. The region growing is achieved through a merging test condition between texturally homogeneous neighboring blocks. If the condition for merging is satisfied, an observing block can be merged into a neighbor block. Otherwise, a new region is declared.

For our segmentation, we have used a centroid linkage region growing method because it is gauranteed to produce disjoint segments with close boundaries and provides a sequential algorithm for growing region. The controid linkage region growing method is illustrated in paper.[4] The texture features are used the mean, JND, and the class type based on the fractal dimension of the image block.

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.

3. Determination of the proper block size for estimating the fractal dimension

When we compute the fractal dimension in an image, the pixel intensity in an image is considered as a surface above a plane. All points in the three-dimensional space as distance ε from the surface were considered, covering the surface with a blanket of thickness 2 ε. The algorithm for computing the fractal dimension was given in paper. [9]

In the proposed compression algorithm, an image is divided into blocks and the fractal dimension of each block is computed. It is very important to choose the proper block size in the image so that good estimates of the fractal dimensions are obtained and good image quality can be maintained. When the block size is small, there may not be enough pixels to describe the texture within the block. For example, if the blocks is 1×1, there is only one pixel in the blocks and it is not possible to characterize its texture. When the block size is large, several different textures may be present within the block and the estimated fractal dimension will not accurately represent the characteristics of the multiple textures. Another issue to be considered when choosing the block size is the computation requirements. The computation requirement for the large block is more expensive than the one for the small block. For example, consider the block sizes 8×8 and 16×16. The number of pixels in the block size 16×16 is four$\times$larger than the number in the block size 8×8. Therefore, the computation time to compute the fractal dimension in the larger block is more expensive. Considering the issues discussed above, we conclude that the smallest feasible block size is the best block alternative.

3.1 Fractal dimension

The definition of the fractal dimension is a set for which the Hausdorff-Besicovich dimension is strictly greater than the topological dimension. We consider object X in an E-dimensional space. N(E) is the number of E-dimensional sphere of diameter needed to cover X, where E is an integer and the E-dimensional space is the minimum integer dimensional space among all possible integer dimensional spaces which can envelop X. Thus, if N(E) is given by

$$N(\varepsilon) = K\ (1/\varepsilon)^{D},\ \text{as}\ \varepsilon \rightarrow 0, \qquad (1)$$

where K is a constant and X has Hausdorff dimension D. If D is fractional, D is also called the fractal dimension. For fractal objects, D is independent of ε.

If the fractal dimension is to be used to characterize the texture in an image, we need a method for estimating the fractal dimension from the given dataset. Many different estimators have been proposed; box counting,[1] yardstick,[3] blanket,[9] and power spectrum.[10] In our case, a blanket method is adopted since it is computationally efficient. The blanket method is described in detail in paper.[18] 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 $A(\varepsilon)$ is then the volume $V(\varepsilon)$ occupied by the blanket divided by 2ε. The area $A(\varepsilon)$ is given by

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

where K is a constant.

From a theoretical viewpoint, if a surface is a perfect fractal surface, then the fractal dimension will remain constant over all ranges of scale. In practice, there are scale range limitations of fractal dimensions due to limitations in textural images. For example, the resolution limit of the image system sets a lower limit on the fractal scaling behavior. An upper limit may be set by the structure being examined. Thus, a real surface will be fractal over some range of scales rather than over all scales. These limiting scales can be expressed as upper (ϵ_{max}) and lower cutoff (ϵ_{min}) scales.

To compute the fractal dimension, we apply the log function to both sides of Eq. (2). A least square linear regression is applied to fit a straight line to the plot of $\log A(\epsilon)$ vs. $\log(\epsilon)$ for ϵ_{min} and ϵ_{max}. The fractal dimension D is equal to 2 minus the slope of the straight line.

3.2 Experimental results for determining the proper block size

To determine the best block size in terms of the fractal dimension, three 30 × 30 subimages in the test image in fig. 1 are taken. Each subimage is assumed to belong to one of the texture classes. We investigated the variation of the fractal dimension versus the block size for each class. Block size is varied from 2 to 30 and is increased from the left, top corner. Curves of fractal dimension versus block size are given in fig. 2. In each plot, the x-axis represents the block size and the y-axis the fractal dimension. The curve with a diamond symbol correponds to the perceived constant intensity, the curve with a cross symbol to the smooth texture, and the curve with the square symbol to the rough texture.

Examining the curves in fig. 2, the fractal dimension corresponding to perceived constant intensity (◇) has almost a constant value, 2.0, for blocksize 2, ... , 30. That is, there exists only a single texture of perceived constant intensity in each block. The shape of the curve corresponding to smooth texture (+) are quite variable for blocks between (2, ... , 7) and (19, ... , 30) but are nearly constant for the middle block size (8, ... , 18). The reason for variability in the smaller 2, ... , 7 blocks is the small number of pixels to characterize the texture. In larger 19, ... , 30 blocks more than one texture is present. The middle 8, ... , 18 blocks have only one texture and provide the least variable estimate of the fractal dimension. Note, the 30 × 30 subimage at the bottom of fig. 1 has three different textures; the neck and two sweaters. In general, when the block size is large, there is more likelihood that several textures will be in the block. and the value of the fractal dimension will not remain constant. The shape of the curve corresponding to the rough texture (□) looks similar to that of the smooth texture. At the small block sizes, there are too few pixels to estimate the texture and at the large block size, multiple textures are present in the block. The curve is nearly constant for middle (8, ... , 18) block sizes. In summary, the larger block sizes may not give good estimates of the fractal

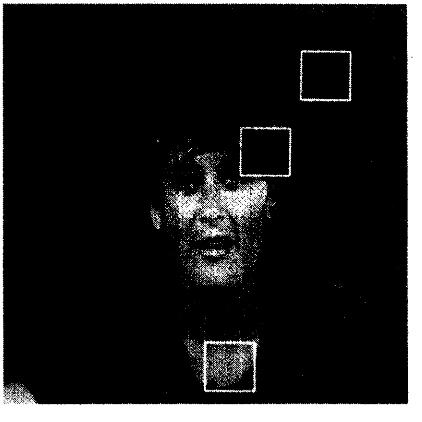

Fig. 1. Test image of Miss USA

dimension because they contain several textures and the smaller block may not contain enough pixels to characterize the texture.

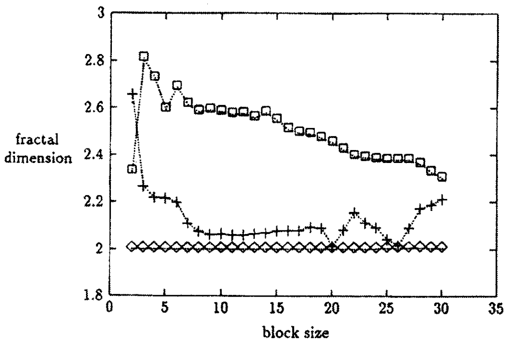

Fig. 2. Plot of fractal dimension versus block size in Miss USA. Miss USA with three subimages is given on the top. Three subimages on the top, middle, and bottom belong to perceived constant intensity, rough texture, and smooth texture respectively. A plot of fractal dimension versus block size is given on the bottom. The curves with a diamond symbol, a cross symbol, and a square symbol correspond to perceived constant intensity, rough texture, and smooth texture respectively.

Through extensive experimentation, we have found that block sizes of 8 × 8 up to 14 × 14 have almost a constant value. Thus these blocks meaningfully represent the textural characteristics of a region. An estimate of the means and the standard deviations of the fractal dimensions for the blocks in each class for the test image as a result of these simulations are given in table 1.

We chose an 8 × 8 block size for the block-by-block segmentation algorithm since the smaller block size reduces the computation and storage requirement and as will be seen later is consistant with giving the best image quality. Furthermore, by comparing curves in the plot, curves on square, cross, and diamond symbols are the top, middle, and bottom respectively for the mid sized blocks from 8 × 8 to 14 × 14, rougher texture produces higher fractal dimension.

Table 1. Statistics of fractal dimension in Miss USA

class	blocks size	mean	standard deviation
Perceived constant intensity	2X2 to 28X28	2.000705	0.000002
smooth texture	8X8 to 19X19	2.072152	0.000110
rough texture	8X8 to 14X14	2.576650	0.000086

4. Conclusions

Segmented images were obtained with D_1=2.033, D_2=2.371, and block size 8 × 8. Information about the number of total segments and the boundary points and the number of the bits to represent the boundaries is given using an arithmetic code. Number of segments, number of boundary points, and number of bits for arithmetic in Miss USA are 231, 1910, and 1344 respectively. The compression ratio for the test image is 0.13 bpp. The decoded image for test image is given in fig. 3.

These results indicate that, using the texture-based segmentation image compression system, compression ratios in the neighborhood of 0.01 to 0.25 bpp are attainable with good image quality.

In addition to working well, there are also other advantages to using the method. One advantage is that the block-by-block method for segmentation-based compression is a more parallel approach than the pixel-by-pixel one. This allows for a fast implementation for the coding algorithm. The algorithm based on the pixel-by-pixel method is not conductive to being done in a parallel fashion.

Another advantage of our block-by-block method is that it allows more readily for compression ratio and image quality trade-offs. By varying parameter, the compression ratios can be easily controlled.

Fig 3. The decoded image of test image.

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, 135(2): 121-132, April 1988.
3. T. J. Dennis and N. G. Dessipris, "Fractal modeling in image texture analysis," IEE Proceedings, F(5):227-235, Oct. 1989.
4. R. M. Haralick and L. G. Shaprio, "Image segmentation techniques," Comput. Graphics Image Processing, 29:100-132, 1985.
5. A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," IEEE Trans. on Image Processing, Vol. 1, No. 1, Jan. 1992.
6. A. K. Jain, Fundamentals of Digital Image Processing, Prentice Hall. 1989.
7. J. W. Jang and S. A. Rajala, "Texture segmentation-based image coder incorporoating properties of the human visual system," In Proc. ICASSP'91, May 1991.
8. J. W. Jang and S. A. Rajala, "Performance comparison of texture segmentation-based image coding technique with DCT image coding technique," IEEE Workshop on Visual Signal Processing and Communications, pp. 200-205, Sep. 1992.
9. M. Kunt, M. Benard, and R. Leonardi, "Recent results in high compression image coding," IEEE Trans. Circuit Syst., Vol. CAS-34, No. 11, pp. 1306-1336, Nov. 1987.
10. N. B. Nill, "A visual model weighted cosine transform for image compression and quality assessment," IEEE Trans. Commun. 33(6):551-557, 1985.
11. S. Peleg, J. Naor, R. Hartley, and D. Avnir, "Multiple resolution texture analysis and classification," IEEE Trans. Pattern Anal. Machine Intell., PAMI-6(4):518-523, July 1984.
12. A. P. Pentland, "Fractal-based description of natural scenes," IEEE Trans. Pattern Anal. Machine Intell., PAMI-6(6):664-674, Nov. 1984.