JOURNAL OF VISUAL COMMUNICATION AND IMAGE REPRESENTATION
Vol. 8, No. 3, June, pp. 317–324, 1997
ARTICLE NO. VC970353

SHORT COMMUNICATION

Polynomial Approximation Coding for Progressive Image Transmission

Chung-Yen Lu,[1] An-Yi Chen, and Kuei-Ann Wen

National Chiao-Tung University, Hsinchu, Taiwan, Republic of China

Received March 13, 1996

This paper presents a progressive image transmission scheme in which polynomial approximation coding is effectively applied to encoding residual images at each stage. This polynomial approximation coding is derived from a regressive model based on the relation between a set of data and their positions. Because the close relation is embedded in a certain polynomial function, the highly efficient compression can be implemented and the decimation and interpolation are also easily realized at the receiving side. Simulations and experimental results are discussed in the later part of this paper. © 1997 Academic Press

I. INTRODUCTION

Progressive image transmission (PIT) has been widely developed to serve many applications, such as remote image data-base access, and telebrowsing and teleconferencing over relatively low bit-rate channels. Generally, transmission is usually divided into stages and a coarse low resolution image is transmitted as an initial approximation. Resolution of the image is refined by sending more stages, and the image is transmitted progressively in order to give a better approximation of the original image [1–13].

Many PIT techniques have been proposed [3–13], and they can be roughly classified into three categories: pyramidal, transform-based, and iterative encoding. In the pyramidal approach [3–6], the levels of the pyramid correspond to the successive approximations of the original image. The image can be progressively reconstructed by adding levels of the pyramid to the top level. In the transform-based approach [6–9], the image first undergoes a block transform and the transformed coefficients are transmitted progressively in a certain order, usually from low to high order. In this manner, successive approximations with progressively higher resolution are obtained by inverse transforming the coefficients. The third approach is to iteratively encode the residue or difference image, either in the spatial domain or in the transform domain [9–13]. At each stage, an error or difference image is generated and then encoded at the next stage.

In this paper, a PIT system is proposed with an iterating encoding technique. Residual images of variable block size are used at different stages, and they are down-sampled to a fixed size for polynomial approximation coding (PAC). The PAC is derived from a regressive model, which is a statistical tool for data analysis. A two-dimensional polynomial equation is properly suited to approximating the relation between data and their positions. Many research projects concerned with 2D polynomial equations for 2D signals have been proposed [14–18]. The PAC provides a simplified architecture for a PIT system with efficient compression.

II. POLYNOMIAL APPROXIMATION CODING

Mathematically, with the techniques of polynomial approximation coding, the discrete value sets can be embedded in a continuous function $f(x, y)$ defined on the unit square surrounding every pixel of the image. Furthermore, another two-dimensional polynomial $p(x, y)$ is employed to approximate the continuous function $f(x, y)$. That is, $f(x, y) = p(x, y) + e(x, y)$, where $e(x, y)$ is the error term.

A general type of 2D polynomial equation is $p(x, y) = \sum_i \sum_j \beta_{i,j} x^i y^j$, where β_{ij} are polynomial coefficients. The polynomial approximation coding is based on the estimation of polynomial coefficients, followed by quantization and variable-length coding (VLC). At the decoder, the received-bits stream is converted to a series of quantized polynomial coefficients. These polynomial coefficients are utilized to recover the original image data by the polynomial equation with adequate position indication (x, y). Certainly, the recovered data is an approximation result.

[1] E-mail: d8211828@lyle.ee.nctu.edu.tw.

1047-3203/97 $25.00

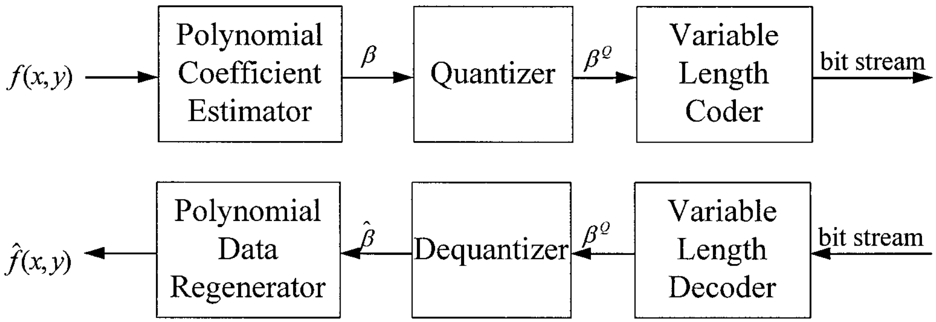

FIG. 1. The block diagram of polynomial approximation coding.

The simple block diagram that roughly describes polynomial approximation coding is illustrated in Fig. 1. The polynomial-coefficient estimator (PCE) is used to estimate the polynomial coefficient. According to the diagram, after the stages of quantization and variable-length coding, $f(x, y)$ will be transmitted in the form of encoded bit streams. At the end of the encoder, the polynomial coefficients are obtained by regression techniques. The mathematical model will be presented shortly in the later part of this section.

Ideally speaking, the continuous function which describes the image would be

$$f(x, y) = \sum_{-\infty}^{\infty} \sum_{-\infty}^{\infty} \beta_{m,n} x^m y^n. \quad (1)$$

Taking a 4×4 case, for example, let

$$p(x, y) = \sum_i \sum_j \beta_{i,j} x^i y^j \cong \beta_0 + \beta_1 x + \beta_2 x + \beta_2 y + \beta_3 xy \quad (2)$$

and recall what was mentioned previously, that $f(x, y) = p(x, y) + e(x, y)$, where $e(x, y)$ is the error term. Now we define the matrix form of the polynomial approximation,

$$\mathbf{F} = \mathbf{X}\mathbf{b} + \mathbf{e}, \quad (3)$$

where

$$\mathbf{F}^{\mathrm{T}} = [f_1 \quad f_2 \quad \cdots \quad f_n]$$

is the original data matrix with n elements,

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 & y_1 & x_1 y_1 \\ 1 & x_2 & y_2 & x_2 y_2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & y_n & x_n y_n \end{bmatrix},$$

is the constant matrix formed by position variables,

$$\mathbf{b}^{\mathrm{T}} = [\beta_0 \quad \beta_1 \quad \beta_2 \quad \beta_3]$$

is the coefficient matrix, and

$$\mathbf{e}^{\mathrm{T}} = [e_0 \quad e_1 \cdots e_n]$$

is an error term matrix.

After some simple numerical procedures, the least-squares estimation of the polynomial coefficient vector **b** will be obtained with the equation

$$\mathbf{b} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{F}. \quad (4)$$

Since the elements of **X** are fixed, we may replace part of Eq. (4) with a generator matrix **G**:

$$\mathbf{G} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}} \quad (5)$$

The computation of **b** will finally be expressed as

$$\mathbf{b} = \mathbf{G}\mathbf{F}. \quad (6)$$

As for the decoding process, each point of the image data can be reconstructed according to Eq. (2). In other words, the reconstructed data matrix $\hat{\mathbf{F}}$ of interpolation or decimation is directly computed by

$$\hat{\mathbf{F}} = \tilde{\mathbf{X}}\mathbf{b}, \quad (7)$$

where

$$\tilde{\mathbf{X}} = \begin{bmatrix} 1 & \tilde{x}_1 & \tilde{y}_1 & \tilde{x}_1 \tilde{y}_1 \\ 1 & \tilde{x}_2 & \tilde{y}_2 & \tilde{x}_2 \tilde{y}_2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \tilde{x}_m & \tilde{y}_m & \tilde{x}_m \tilde{y}_m \end{bmatrix},$$

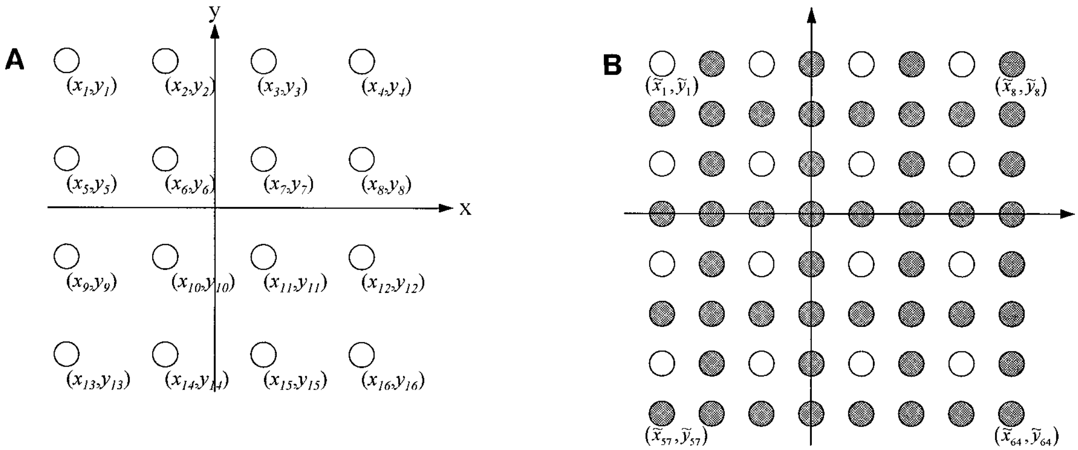

FIG. 2. (a) The set of (x, y) for X, (b) the set $(\tilde{x}, \tilde{y})$ for $\tilde{X}$.

and $\tilde{x}$ and $\tilde{y}$ are the position mapping variables of extension decoding for interpolation or decimation. This means, for a set of fixed polynomial coefficients, that the interpolation of data is implemented by a polynomial equation with the interpolation of the set of (x, y). An interpolation example of (x, y) is illustrated in Figs. 2a and 2b.

III. PROGRESSIVE IMAGE TRANSMISSION ALGORITHM

The progressive image transmission system transmits a set of residual image frames $\{\hat{f}^k, k > 0\}$. Let f^1 represent the original image and $\hat{f}^k$ represent the transmission image at stage k. In the ideal case, $f^1 = \sum_{k>0} f^k$, but actually the transmission will be limited in a certain number of stages. that is, $f^1 = \sum_{k=1}^{m} \hat{f}^k + f^R$, where f^R is the error image in this m-stages system.

Let f^k be the input frame at the kth stage; then the input frame at the next stage is generated by $f^{k+1} = f^k - \hat{f}^k$. The progressively reconstructed image is expressed as $\hat{F}^k = \hat{F}^{k-1} + \hat{f}^k$, where $\hat{F}^1 = \hat{f}^1$.

Each residual frame f^k consists of a series of nonoverlapped blocks $\{X_l^k, 1 \leq l \leq N_k\}$, where N_k is the number of blocks at frame k. The block size of each frame is varied at different stages in order to refine the process. For the sake of convenience, we set $N_{k+1} = 4N_k$. That is, if the size of X^k is $(2M) \times (2M)$ at stage k, the size of X^{k+1} is $M \times M$.

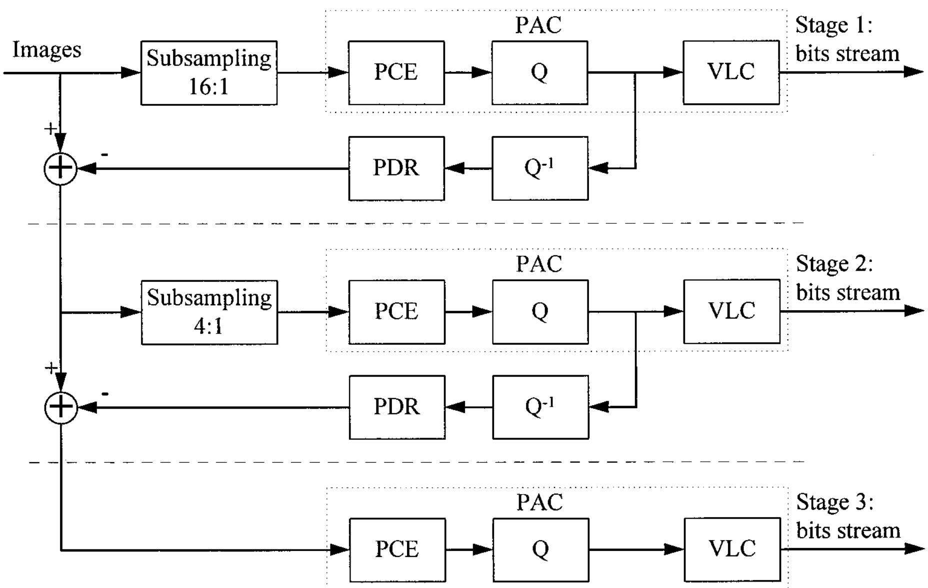

FIG. 3. The encoder of the three-stage PIT based on the PAC system.

FIG. 4. The test image Lena of size 512 × 512 (8 bpp) pixels.

The m-stage PIT algorithm is described by the following procedures:

Step 0. Set stage variable $k = 1$, block size N $= 2^{(m+1)} \times 2^{(m+1)}$.

Step 1. Divide the frame f^k into regular blocks, whose size is $N \times N$.

Step 2. Each block image is subsampled $4^{k-1}:1$ to the size of 4×4.

TABLE 1
PAC Coefficient Coding Categories

Range	DC Difference category	AC category
0	0	
−1, 1	1	1
−3, −2, 2, 3	2	2
−7, . . . , −4, 4, . . . , 7	3	3
−15, . . . , −8, 8, . . . , 15	4	4
−31, . . . , −16, 16, . . . , 31	5	5
−63, . . . , −32, 32, . . . , 63	6	6
−127, . . . , −64, 64, . . . , 127	7	7
−255, . . . , −127, 127, . . . , 255	8	8

TABLE 2
PAC Default DC Code

Category	Base code	Length
0	00	2
1	01	3
2	10	4
3	110	6
4	1110	8
5	11110	10
6	111110	12
7	1111110	14
8	1111111	15

TABLE 3
PAC–PIT Default AC Code for First Stage

Run/ category	Base code	Length	Run/ category	Base code	Length
EOB	00	2			
0/1	100	4	1/5	11111111010	16
0/2	01	4	1/6	111111111001	18
0/3	101	6	1/7	111111111010	19
0/4	1100	8	1/8	111111111011	20
0/5	11111100	13	2/1	11101	6
0/6	11111111000	17	2/2	11110	7
0/7	11111111001	18	2/3	11111110	11
0/8	111111111000	20	2/4	11111111011	15
1/1	1101	5	2/5	111111111100	17
1/2	11100	7	2/6	111111111101	18
1/3	111110	9	2/7	111111111110	19
1/4	11111101	12	2/8	111111111111	20

Step 3. The subsampled image is encoded by PAC.
Step 4. If $k = m$, stop the procedure.
Step 5. Reconstruct the block image $\hat{f}^k$ from PAC decoder.
Step 6. The residual image f^{k+1} is the difference between f^k and $\hat{f}^k$.
Step 7. Set $k = k + 1$, $N = N/2$, go back to Step 1.

An encoder of the three-stage PIT system is illustrated in Fig. 3. Since the original image is divided into a series of nonoverlapped blocks of size 16×16 at the first stage, $m = 3$ is revealed. These blocks are downsampled to be blocks of size 4×4 as the input of the polynomial coefficients estimator; somehow the downsampling process just picks up one out of every 4×4 pixels. After the quantization of polynomial coefficients, there are two data paths: one is variable length coding which is used to reduce the bit rates; the other is inverse quantization and polynomial data reconstruction (PDR). PDR reconstructs the block of size 16×16 by directly implementing Eq. (2). At the

TABLE 4
PAC–PIT Default AC Code for Second and Third Stages

Run/ category	Base code	Length	Run/ category	Base code	Length
EOB	0	1			
0/1	100	4	2/1	11101	6
0/2	1100	6	2/2	1111100	9
0/3	1101	7	2/3	111111001	12
0/4	1111010	11	2/4	11111110110	15
0/5	1111110110	15	2/5	11111110111	16
0/6	11111110000	17	2/6	11111111000	17
0/7	11111110001	18	2/7	11111111001	18
0/8	11111110010	19	2/8	11111111010	19
1/1	101	4	3/1	111100	7
1/2	11100	7	3/2	1111101	9
1/3	1111011	10	3/3	111111010	12
1/4	111111000	13	3/4	11111111011	15
1/5	1111110111	15	3/5	11111111100	16
1/6	11111110011	17	3/6	11111111101	17
1/7	11111110100	18	3/7	11111111110	18
1/8	11111110101	19	3/8	11111111111	19

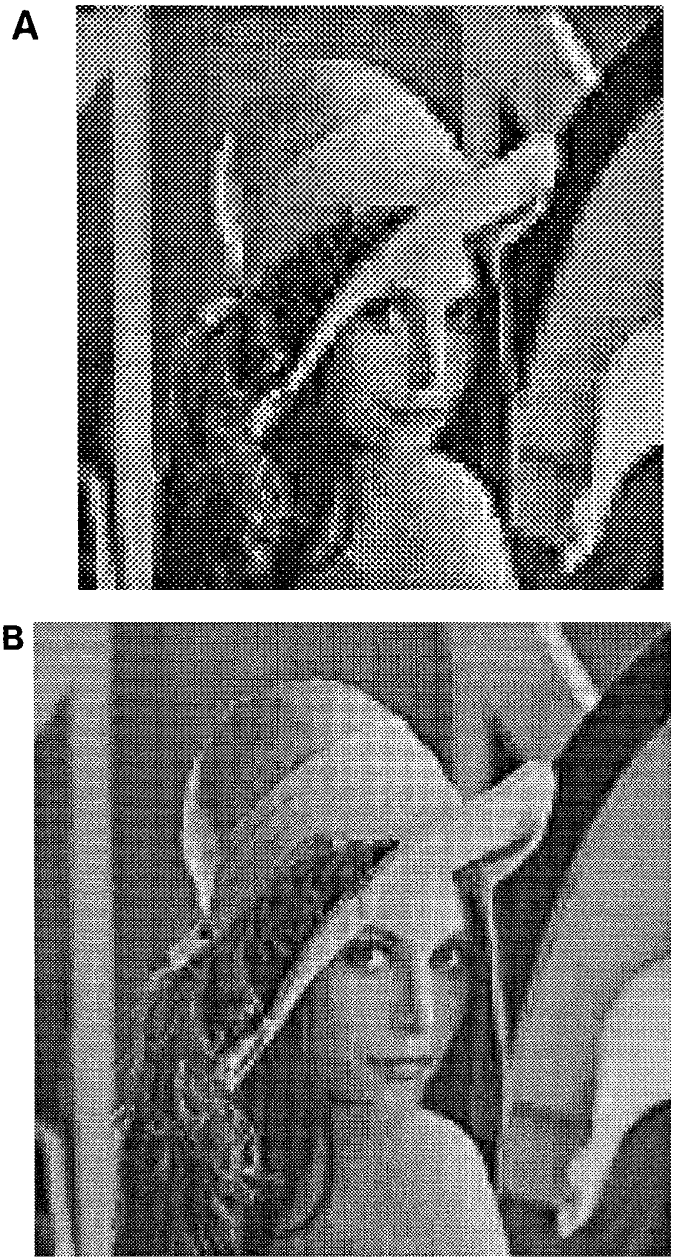

FIG. 5. (a) Reconstructed image of size 128×128 pixels at the first stage and (b) reconstructed image of size 256×256 at the second stage.

FIG. 6. (a) Reconstructed image of size 512 × 512 pixels at the first stage, (b) reconstructed image of size 512 × 512 at the second stage, and (c) reconstructed image of size 512 × 512 at the third stage.

second stage, the residual image is the difference between the original image and the reconstructed image at the first stage. The residual image is divided into nonoverlapping blocks of size 8 × 8. With downsampling at the ratio of 4 : 1, PCE, quantization, and VLC are used again to encode the residual data. The residual image for the input at the third stage is constructed by inverse quantization and PDR at the second stage. At the third stage, the residual image from the previous stage is divided into nonoverlapping blocks of size 4 × 4 and coded by PAC.

IV. EXPERIMENTAL RESULTS

High-level language simulation using the proposed progressive image transmission scheme on the test image Lena is presented. The test image is of size 512 × 512 and quantized to 256 levels, as shown in Fig. 4.

The peak signal-to-noise ratio (PSNR) is used to determine image reconstruction fidelity. The PSNR is defined as

$$\mathrm{PSNR} = 10 \log_{10}\left(\frac{255^2}{\mathrm{MSE}}\right),$$

where MSE is the mean-square error given by

$$\mathrm{MSE} = \frac{1}{N^2}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}[\hat{F}(u,v) - F(u,v)]^2,$$

with F and $\hat{F}$ denoting the original and reconstructed images, respectively, and N the image size.

The PIT system transmits images in three stages. The image is first subdivided into pixel blocks of 32 × 32, which are processed in left-to-right, top-to-bottom directions. As each 32 × 32 block or subimage is encountered, its 1024 pixels are downsampled into a block of 4 × 4 pixels. Then the polynomial coefficients of this block are computed and quantized. In particular, the nonzero AC coefficients are encoded using a variable-length code that defines the coefficient's value and the number of preceding zeros. The DC coefficient is differentially coded relative to the DC coefficient of the previous subimage (Table 1). Tables 2 and 3 provide the default PAC Huffman codes at the first stage. At the second and third stages, all the polynomial coefficients are coded by the same PAC Huffman code listed in Table 4.

We present two sets of decoded images of different sizes. The first set of decoded images are of sizes 128 × 128, 256 × 256, and 512 × 512 at three different stages. The first two images are illustrated in Fig. 5, while the other set of decoded images are of the same size, 512 × 512, as shown in Fig. 6. The bit rate and the PSNR of the coded images are listed in Table 5.

V. CONCLUSION

By utilizing a customized method of incorporating the polynomial regression coding for the residual images, an effective and efficient progressive image transmission scheme has been developed. The polynomial approximation coding, derived from a regressive model, obtains a set of optimum coefficients of a polynomial expression for a square image. Since the residual images are approximated by continuous functions, the scaling property of the reconstructed images is achieved.

TABLE 5
PAC–PIT and Hierarchical Mode JPEG Bit-Rate vs PSNR for the Lena Image

	PAC–PIT		HM–JPEG [9]	
Transmission stage	Bit-rate (bpp)	PSNR (dB)	Bit-rate (bpp)	PSNR (dB)
First stage	0.046	23.12	0.146	23.80
Second stage	0.142	26.57	0.403	28.46
Third stage	0.345	30.13	—	—

APPENDIX: THE COMPUTATION COMPLEXITY OF PAC

In order to design a row–column computation architecture, we redefine the matrix form in Eq. (3) to the form in Eq. (8):

$$\mathbf{F} = \mathbf{X}_{\mathrm{row}}\mathbf{B}\mathbf{X}_{\mathrm{col}} \tag{8}$$

where data matrix

$$\mathbf{F} = \begin{bmatrix} f_{00} & f_{01} & f_{02} & f_{03} \\ f_{10} & f_{11} & f_{12} & f_{13} \\ f_{20} & f_{21} & f_{22} & f_{23} \\ f_{30} & f_{31} & f_{32} & f_{33} \end{bmatrix},$$

beta matrix

$$\mathbf{B} = \begin{bmatrix} \beta_{00} & \beta_{01} \\ \beta_{10} & \beta_{11} \end{bmatrix},$$

and constant matrices

$$\mathbf{X} = \mathbf{X}_{\mathrm{row}} = X_{\mathrm{col}}^{\mathrm{t}} = \begin{bmatrix} 1 & -1.5 \\ 1 & -0.5 \\ 1 & 0.5 \\ 1 & 1.5 \end{bmatrix}.$$

The computation of the beta matrix could be obtained with the following equation:

$$\begin{aligned}\mathbf{B} &= [(\mathbf{X}^{\mathrm{t}}\mathbf{X})^{-1}\mathbf{X}^{t}]\mathbf{F}[\mathbf{X}(\mathbf{X}^{t}\mathbf{X})^{-1}] \\ &= \mathbf{G}\mathbf{F}\mathbf{G}^{t}\end{aligned} \tag{9}$$

where

$$\mathbf{G} = \mathbf{S}\mathbf{W} = \begin{bmatrix} \frac{1}{4} & 0 \\ 0 & \frac{1}{10} \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 & 1 \\ -3 & -1 & 1 & 3 \end{bmatrix}. \tag{10}$$

Due to the scaling operation **S** could be merged with the quantization process, only the computation of **W** needs to be implemented in PCE. A multiplier-free PCE architecture could be easily designed. Therefore, the architecture complexity of PAC is much less than that of the DCT-based coding algorithm.

REFERENCES

1. K. R. Sloan, Jr., and S. L. Tanimoto, Progressive refinement of Raster images, *IEEE Trans. Comput.* **C-28** (11), Nov. 1979, 871–874.
2. P. J. Burt and E. H. Adelson, The Laplacian pyramid as a compact image code, *IEEE Trans. Comm.* **COM-31,** Apr. 1983, 532–540.
3. K. H. Tzou, Progressive image transmission: A review and comparison of techniques, *Opt. Eng.* **26,** July 1987, 581–589.
4. L. Wang and M. Goldberg, Progressive image transmission using vector quantization on images in pyramid form, *IEEE Trans. Comm.* **37**(12), Dec. 1989.
5. G. Candotti and S. Carrato, Pyramidal multiresolution source coding for progressive sequences, *IEEE Trans. Consumer Electron.* **40**(4), Nov. 1994, 789–795.
6. K. H. Tan and M. Ghanbari, Layered image coding using the DCT pyramid, *IEEE Trans. Image Process.* **4**(4), Apr. 1995, 512–516.
7. K. N. Ngan, Image display techniques using the cosine transform, *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-32,** Feb. 1984, 173–177.
8. E. Dubois and J. L. Moncet, Encoding and progressive transmission of still pictures in NTSC composite format using transform domain methods, *IEEE Trans. Comm.* **COM-34,** Mar. 1986, 310–319.
9. A. Jain and S. Panchanathan, Scalable compression for image browsing, *IEEE Trans. Consumer Electron.* **40**(3), Aug. 1994, 394–404.
10. L. Wang and M. Goldberg, Progressive image transmission by multistage transform coefficient quantization, *in "IEEE International Conference on Communications,* Toronto, Ontario, June 1986," pp. 419–423.
11. W. D. Hofmann and D. E. Troxel, Making progressive transmission adaptive, *IEEE Trans. Comm.* **COM-34,** Aug. 1986, 806–813.
12. L. Wang and M. Goldberg, Progressive image transmission by residual error vector quantization in transform domain, *in "Proc. IEEE International Phoenix Conference on Computer Communication,* Scottsdale, AZ, Feb. 1987," pp. 178–182.
13. L. Wang and M. Goldberg, Lossless progressive image transmission by residual error vector quantization, *IEEE Proc.* **135**(Pt. F5), Oct. 1988, 421–430.
14. R. M. Haralick and L. Watson, A facet model for image data, *Comput. Graphics Image Process.* **15,** 1981, 113–129.
15. M. Eden, M. Unser, and R. Leonardi, Polynomial representation of pictures, *Signal Process.* **10,** 1986, 385–393.
16. M. Kocher and R. Leonardi, Adaptive region growing technique using polynomial functions for image approximation, *Signal Process.* **11,** 1986, 47–60.
17. M. Biggar, O. J. Morris, and A. G. Constantinides, Segmented-image Coding: performance comparison with the discrete cosine transform, *IEEE Proc.* **135**(Pt. F2), Apr. 1988.
18. E. Karabassis and M. E. Spetsakis, An analysis of image interpolation, differentiation, and reduction using local polynomial fits, *Graph. Models Image Process.* **57**(3), May 1995, 183–196.

CHUNG-YEN LU was born in Taipei, Taiwan, R.O.C., in 1969. He received the B.E.E., M.E.E., and Ph.D. degrees from the Department of Electronics Engineering and the Institute of Electronics Engineering at National Chiao-Tung University, Taiwan, R.O.C., in 1991, 1993, and 1997, respectively. Mr. Chung-Yen Lu is currently working on his post-doctoral research in the area of wireless media transmission and video processing with the Spring Foundation of National Chiao-Tung University, Taiwan, R.O.C.

AN-YI CHEN was born in Tainan, Taiwan, R.O.C., in 1971. He received his B.E. degree from the Department of Electrical Communication Engineering at National Chiao-Tung University in 1993, and his M.S. degree from the Department of Electrical and Computer Engineering at the State University of New York at Buffalo in 1995. Presently Mr. Chen is pursuing his Ph.D. degree at the Institute of Electronics Engineering at National Chiao-Tung University. His current research interests are image and video processing, especially compression and transmission techniques.

KUEI-ANN WEN was born in Keelung, Taiwan, Republic of China, in 1961. She received the B.E.E., M.E.E., and Ph.D. degrees from the Department of Electrical and Computer Engineering at National Cheng Kung University, Taiwan, R.O.C., in 1983, 1985, and 1988, respectively. She is presently a professor in the Department of Electronics Engineering, National Chiao-Tung University, Hsinchu, Taiwan, R.O.C., where she has joined the Center for Telecommunications Research. Her current research interests are in the area of high-speed digital signal processing, parallel processing, and VLSI circuit design, and error correcting coding.