A SIMPLE THREE–PARAMETER SURFACE FITTING SCHEME FOR IMAGE COMPRESSION

Salah Ameer [1]

Otman Basir [2]
Department of Electrical and Computer Engineering
University of Waterloo
Waterloo, ON Canada

Keywords Image Compression, Surface Fitting, Blocking Effects, Progressive Transmission, and Multiplication – and Division – Free Implementations.

Abstract This paper describes a simple scheme to compress images through surface fitting. The scheme can achieve better than 60:1 compression ratio with acceptable image quality degradation. The results are superior to those of JPEG at comparable ratios. Another advantage is that no multiplications or divisions are required, making the implementation suitable for online or progressive compression. Blocking effects were reduced (up to 0.5dB of PSNR improvement) through simple line fitting on block boundaries.

1 INTRODUCTION

With the increasing demand on data transfer and storage, data compression has become a necessity. Generally, compression falls in two categories: lossless (exact reconstruction) (Berg and Mikhael, 1994) and lossy. The former has a low compression rate while the latter has a higher one.

Image compression has been widely investigated, and many algorithms have been proposed (Egger et al., 1999). The human visual system is nonlinear; hence, a compromise (to a certain extent) between perceptual quality and high compression ratios can be reached.

In DPCM (Differential Pulse Code Modulation) coding (Habibi, 1977), a pixel is predicted from its causal neighborhood, and the prediction error is quantized and coded. High compression is difficult to attain due to accumulated errors and the need for multi-model prediction.

To overcome these limitations, block-based compression techniques (dividing the image into nonoverlapping blocks) were suggested (Egger *et al.*, 1999). At higher rates, these techniques suffer from visually annoying artifacts on block boundaries. Sub-band coding (wavelets) (Lin and Vaidyanathan, 1996) is free of such artifacts; however, the reconstructed image tends to be blurry.

Block-based techniques can be categorized into training-type and non-training type techniques. Training-type techniques include vector quantization (Li and Zhang, 1995), neural networks (Jiang, 1999), and iterated functions or fractals (Wohlberg and de Jager, 1999). In this category, compression performance is dependent on how similar is the image to the training set. Non-training type techniques include block truncation (Delp and Mitchell, 1979), transform coding (including Discrete Cosine Transform used in JPEG (Furht, 1995), and surface fitting (Eden *et al.*, 1986).

Polynomial fitting was employed in various compression techniques, including, block-based compression through splines (Watanabe, 1997), prediction of motion compensation vectors in video coding (Karczewics *et al.*, 1997), and block-based image compression through variable size triangular blocks (Lu *et al.*, 2000). Segmentation-based compression (Biswas, 2003) also uses 1D and 2D polynomial fitting. Zigzag scanning was used in (Nguyen and Oommen, 1997) to convert the block to 1D and then to perform linear approximations between selected knots.

Ameer S. and Basir O. (2006).
A SIMPLE THREE–PARAMETER SURFACE FITTING SCHEME FOR IMAGE COMPRESSION.
In *Proceedings of the First International Conference on Computer Vision Theory and Applications*, pages 101-106
DOI: 10.5220/0001360501010106

Surface fitting has been used in image segmentation (Lim and Park, 1988), in image noise reduction (Sinha and Schunck, 1992), and for quality improvement of block-based compression (Laha et al., 2004). It was used in (Baseri and Modestino, 1994) (using splines) to encode the lowest frequency band in subband coding. Fitting a surface to known samples can help to reconstruct the lost sub-band coefficients (Hemami and Gray, 1997). In (Kim and Lee, 2002), surface fitting was combined with RBF networks to perform compression using a predefined set of patterns for the RBF centers.

A simplified derivation for first order (plane) fitting was proposed in (Strobach, 1991). The coefficients (assumed to be uniformly distributed) of a 2Nx2N block are computed from their NxN counterparts. A PSNR of 32 dB was obtained for 16:1 compression with high complexity of building the quadtree. A related quadtree approach was proposed by (Hasegawa and Yamasaki, 2002) to predict block corners from the upper left one. These four corners are used in the decoder to find the coefficients of (dxy + ax + by + c).

This paper exploits the implementation of surface fitting techniques in image compression. No edge detection or error calculations are performed to eliminate the need for image-dependent thresholds and/or multipass operations. Section 2 introduces plane fitting. To maintain comparable complexity, only three parameters are used in higher order implementations described in Section 3. Results and comparisons are presented in Section 4, followed by conclusions and suggestions in Section 5.

2 ALGORITHM DESCRIPTION

2.1 Mathematical Formulation

The image is divided into nonoverlapping blocks, each considered as a 3D surface. The z-axis is the pixel gray value (i.e., intensity g). The simplest case is a plane, i.e., ***z = ax + by + c***. To reduce computations, the block center is selected as the origin. Formulating as an MSE problem, we have

$$\underset{a,b,c}{\text{Minimize}} \sum_{x=-(N-1)/2}^{(N-1)/2} \sum_{y=-(N-1)/2}^{(N-1)/2} \left(ax+by+c-g(x,y)\right)^2 \qquad (1)$$

where N is the block dimension. Setting the derivatives with respect to ***a, b***, and ***c*** to zero results in,

$$a = Z_{10}, \quad b = Z_{01}, \text{and} \quad c = Z_{00} \qquad (2)$$

where,

$$Z_{ij} = \frac{\sum_{x=-(N-1)/2}^{(N-1)/2} \sum_{y=-(N-1)/2}^{(N-1)/2} x^i y^j g(x,\ y)}{\sum_{x=-(N-1)/2}^{(N-1)/2} \sum_{y=-(N-1)/2}^{(N-1)/2} x^{2i} y^{2j}} \qquad (3)$$

To reduce the number of additions, we sum row–wise (or column–wise) and use the partial sums in finding more than one parameter. Simple manipulations can be performed to convert each multiplication to two shifts or fewer.

2.2 Quantization

Experiments on many pictures showed that quantization should be uniform for c and nonuniform for a and b (uniform was assumed for all in (Strobach, 1991)). When the origin is selected as the upper left corner, the range of c increases by more than 20% compared to the case of selecting the block center.

The distributions for a and b are very similar and can be well approximated (for N>3) by zero mean Laplacian random variables. Quantization thresholds follow the pattern $\pm(P^{1/Q}-1)$, …, $\pm(P-1)$ where Q is the number of intervals. The value of P was set to 32, though it is not critical. Consequently, the levels are 0, $\pm\ (P^{3/2Q}-1)$, … $\pm\ (P^{(2Q-1)/2Q}-1)$. Stretching the levels and thresholds by $(1+e^{-\ |N-4|/2})$ was found useful experimentally. These pre-saved levels are of great help in eliminating the division in (3).

2.3 Encoding

To eliminate the need for sending coding tables, comma coding (followed by a sign bit) was implemented for ***a*** and ***b*** and binary coding for ***c***. Compression ratio CR is defined as

$$CR = \frac{No.\ of\ bits\ in\ original\ file}{No.\ of\ bits\ in\ compressed\ file} \qquad (4)$$

2.4 Post Processing

At the decoder, block boundaries are linearly interpolated (both horizontally and vertically) to reduce blocking effects. The procedure ignores pixel values at the boundaries and replaces them with those obtained from the nearest two points. Mathematically,

$$\begin{aligned} \hat{g}(Nx, y) &= \left(2\hat{g}(Nx-1, y) + \hat{g}(Nx+2, y)\right)/3 \\ \hat{g}(Nx+1, y) &= \left(\hat{g}(Nx-1, y) + 2\hat{g}(Nx+2, y)\right)/3 \end{aligned} \qquad (5)$$

where $\hat{g}$ is the reconstructed image, x = 1, … X/N, y = 1, … Y, and X and Y are the image dimensions. A similar argument can be applied to the vertical direction. The division in (5) can be eliminated through the following modification

$$\begin{aligned}\hat{g}(Nx,y) &= \left(3\hat{g}(Nx-1,y)+\hat{g}(Nx+2,y)\right)/4\\ \hat{g}(Nx+1,y) &= \left(\hat{g}(Nx-1,y)+3\hat{g}(Nx+2,y)\right)/4\end{aligned} \tag{6}$$

These simple procedures improve both visual quality and PSNR given by

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{255^2\,\mathrm{XY}}{\sum_x\sum_y\left(\mathrm{g(x,y)}-\hat{\mathrm{g}}\mathrm{(x,y)}\right)^2}\right) \tag{7}$$

3 EXTENSION TO HIGHER ORDERS

3.1 Adding the Term xy

Here we have ***z=(ax+c) (by+c)***. Minimizing MSE, we get

$$\mathrm{a/c} = \mathrm{k} + \sqrt{\mathrm{k}^2+\mathrm{r}},\ \ \mathrm{c}^2 = \frac{\mathrm{Z}_{10}\,\mathrm{a/c}+\mathrm{Z}_{00}}{(\mathrm{a/c})^2+1} \tag{8}$$

$$\text{and}\qquad \mathrm{bc} = \frac{\mathrm{Z}_{11}\,\mathrm{a/c}+\mathrm{Z}_{01}}{(\mathrm{a/c})^2+1}$$

where

$$\mathrm{k} = \frac{\mathrm{Z}_{11}^2-\mathrm{Z}_{01}^2-\mathrm{Z}_{00}}{2\mathrm{Z}_{11}\mathrm{Z}_{01}},\ \text{and}\quad \mathrm{r} = \frac{Z_{10}+Z_{11}Z_{01}}{Z_{11}Z_{01}} \tag{9}$$

a and b follow their plane counterparts with P=4.

3.2 Separable Monotonics

In this case,

$$\mathrm{z = a\ sign(x)\ |x|^m + b\ sign(y)\ |y|^n + c} \tag{10}$$

Minimizing MSE, we have

$$\begin{aligned}\mathrm{a} &= \sum_{x=-(N-1)/2}^{(N-1)/2}\ \sum_{y=-(N-1)/2}^{(N-1)/2} \frac{\mathrm{x|x|^{m-1}\,g(x,\ y)}}{\mathrm{N\ S_x}}\\ \mathrm{b} &= \sum_{x=-(N-1)/2}^{(N-1)/2}\ \sum_{y=-(N-1)/2}^{(N-1)/2} \frac{\mathrm{y|y|^{n-1}\,g(x,\ y)}}{\mathrm{N\ S_y}}\\ \mathrm{c} &= \mathrm{Z}_{00}\end{aligned} \tag{11}$$

where $S_x = \sum |x|^{2m}$ and $S_y = \sum |y|^{2n}$. The best MSE performance was the plane case, i.e., m = n =1.

3.3 Quadratic Surfaces

Different combinations of three unknowns were tried, e.g., ***z=(ax+by+c)²*** and ***z=(ax+c)²+(by+c)²***. The solutions are obtained through nonlinear equations. However, the performance was poor and hence was not considered.

3.4 Higher Orders

Many surfaces can be fitted (using three unknowns) by gray scale transformations of the form ***f(g(x,y))***, i.e.,

$$\mathrm{Minimize}\sum_{x=-(N-1)/2}^{(N-1)/2}\ \sum_{y=-(N-1)/2}^{(N-1)/2}\left[\mathrm{ax+by+c-f(g(xy))}\right]^2 \tag{12}$$

The Quality is inferior to that of the plane case. When ***f(x) = x^r***, the performance improves and reaches its optimum at *r=1*.

4 EXPERIMENTAL RESULTS

The proposed algorithm was tested on the standard image PEPPER (512x512 with 8 bits per pixel) shown in Fig(1). Fig(2) shows the reconstructed (before and after linear interpolation at block boundaries) images (using 8x8 blocks) for the plane case Q=4 and 5 bits for ***c*** (CR=57.92:1). It is clear that constant regions are well described with tolerable edge degradations. In comparison, a JPEG image, taken from MatLab, is shown in Fig(3) (CR=47.04:1). Although the PSNRs are comparable (around 27.5dB), visual quality of the reconstructed image is more pleasing than that of JPEG. Fig(4) shows zooming of the original and reconstructed images.

Table I: Performance for different block sizes (Q=max(2,N/2)).

N	PSNR (dB)	CR
4	29.52	17.79
5	28.66	26.45
6	28.61	36.46
7	27.75	46.22
8	27.57	58.50
9	26.74	69.74
10	26.41	84.85
11	25.76	96.97
12	25.43	114.66

Fig(1): Original image PEPPER.

Additional 4 bits are needed per image to send the number of intervals, Q, (2–5) and the number of quantization bits for ***c*** (3–6). The proposed scheme was implemented with different block sizes, as shown in Table I. The results are comparable (superior in many cases) to those listed in (Biswas, 2003). It is interesting to note that even values of N have higher CR than N–1 with slight reduction in PSNR.

Table II: Performance for different overlapped block sizes.

N	**PSNR (dB)**	**CR**
4	30.97	9.02
6	29.36	18.80
8	28.01	30.23
10	26.77	43.22
12	25.79	57.49

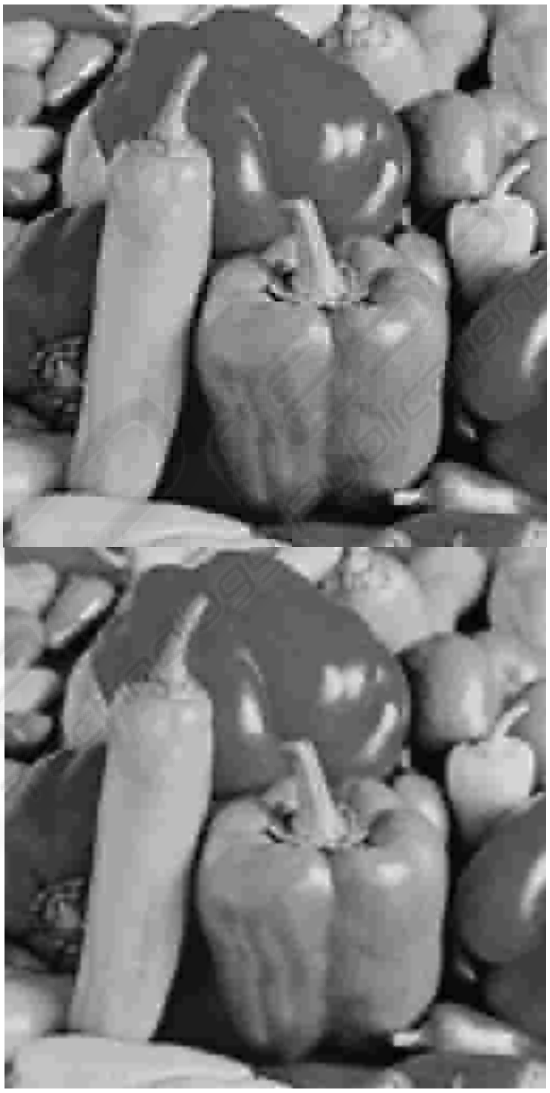
Table III: Performance for different sizes overlapped by N–8 pixels.

N	**PSNR (dB)**	**CR**
9	24.79	54.29
10	24.79	53.63
11	24.40	51.08

Table IV: Performance on different images using 8x8 blocks.

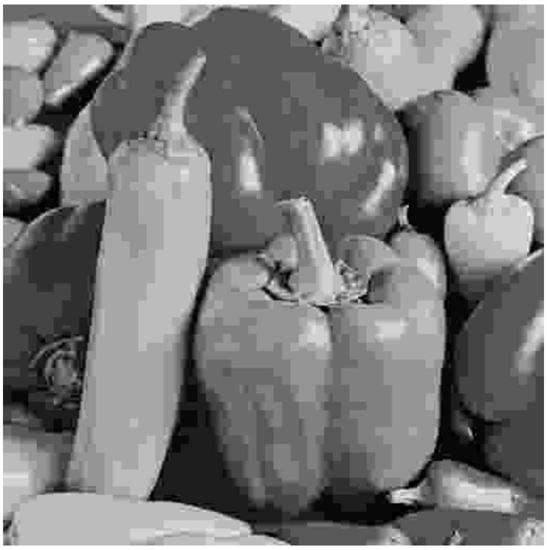
Image	**PSNR (dB)**	**CR**
Balloon	23.98	59.13
Boat	26.26	59.18
Chimp	26.32	57.09
Fern	28.86	62.51
Mandrill	20.62	54.58
Nature	24.66	57.18
Temple	23.51	55.78

For completeness, Table II lists results for similar size overlapped blocks, while Table III shows results for different block sizes overlapped by N–8 pixels. These two tables demonstrate the marginal PSNR gain obtained at the expense of CR reduction. Implementations on different images are given in Table IV.

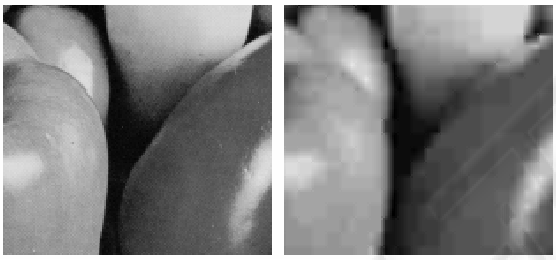


Fig(2): Reconstructed images (8x8 blocks) at 57.92:1 compression (upper) before linear interpolation 26.96dB and (lower) after linear interpolation 27.57dB.

The proposed quantization reduces PSNR by less than 0.3dB. A compensation of around 0.5dB was obtained with linear interpolation at block boundaries; however, the visual quality was not improved that much for N>8. No significant differences were noticed between the implementations of (5) and (6). Interpolation gains are higher for odd N than for even N. Diagonal interpolation (after horizontal and vertical interpolation) produces an insignificant degradation of 0.03dB. A negligible improvement of 0.02dB was obtained with a one-step extrapolation in the four directions of each block.



Fig(3): JPEG image at 47.04:1 compression and 27.48dB.



Fig(4): Zooming of images in Fig(1).

A better reduction of blocking effects (around 0.7dB) was obtained with a 10-point cubic fitting. This slight increase did not improve visual quality and was not favored against the linear one due to the added complexity.

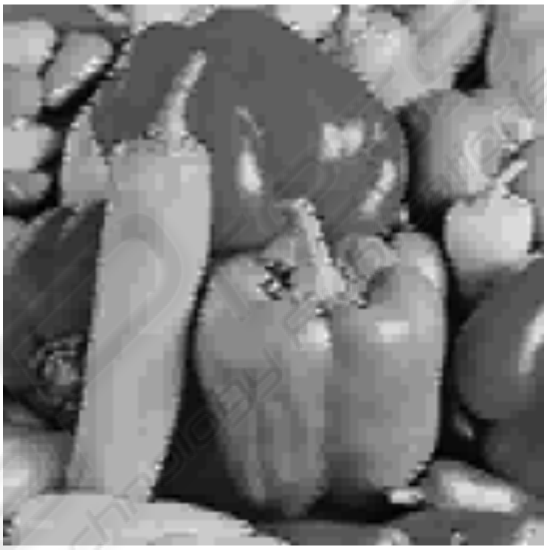
PSNR improvement of 0.1dB (at 51.72:1 compression) can be obtained when quantizing ***c*** to 6 bits. This slight increase is visually more pleasing in homogeneous regions. In fact, the reconstruction quality is sensitive to the quantization of c more than to that of a and b.

Table V lists some results for different values of Q when ***c*** is quantized to 5 bits. As for subjective quality, reconstructed images are visually acceptable; however, the cases Q=2 and Q=3 are slightly annoying because of blockiness. No significant differences were noticed between other values of Q. Though not implemented, ***c*** can be adaptively quantized in a similar fashion to that of the DC value in the JPEG compression scheme.

An increase of approximately 10% in CR was obtained (PSNR decreases by 0.2dB) with ***(a+b)/2*** and ***(a–b)/2*** instead of ***a*** and ***b***. However, the visual quality was similar as that of Q=3 (see Table V).

Table V: Performance for different Q on 8x8 blocks.

Q	PSNR (dB)	CR
2	26.51	69.81
3	27.24	63.74
4	27.57	58.50
5	27.68	54.51
6	27.76	51.12
7	27.81	48.41
8	27.85	45.92



Fig(5): Reconstruction (xy case): at 64.53:1 compression and 26.21dB with c quantized to 5 bits and Q=4.

Fig(5) shows the reconstructed image for the xy case. It has less quality and higher computational cost than the plane case. Blocking effects are more annoying at the region boundaries. Similar to the plane case, the quantization of ***c*** has more influence on the subjective quality of the reconstructed image. This finding is not surprising since c represents the average gray level of the block.

5 CONCLUSIONS AND SUGGESTIONS

An image compression scheme has been implemented via surface fitting. The performance is superior (both perceptually and in PSNR value) to that of JPEG at compression ratios >32:1.

Each block is represented by three quantized coefficients. To reduce quantization error (Strobach, 1991) and the number of bits allocated to the constant parameter ***c***, the block center was chosen as the origin. Simple quantization and coding schemes were used to reduce cost; however, there is still room for improvement in this aspect. In fact, sending functions of ***a*** and ***b*** can increase CR keeping PSNR almost unaffected. This observation needs further investigations together with a more perceptually correlated error measure.

Plane fitting implementation is multiplication- and division-free. The number of shifts can be drastically decreased at the decoder by adopting similar calculations to that of (Hasegawa and Yamasaki, 2002). This low computational cost makes the proposed algorithm suitable for real time applications. Embedded coding can be achieved by sending ***c*** on bit bases followed by ***a***(***b***) and ***b***(***a***).

Blocking effects were reduced with simple 2-point linear interpolation. This reduction compares well to the reduction obtained with 10-point cubic fitting.

Work is in progress to incorporate better edge and/or texture descriptions to improve PSNR.

REFERENCES

Baseri, R., and Modestino, J., 1994. Region-based coding of images using a spline model. In *Proc. IEEE International Conference Image Processing*, Vol. 3, pp 866 – 870.

Berg, A., and Mikhael, W., 1994. A survey of techniques for lossless compression of signals. In *Proc. of the 37th Midwest Symposium on Circuits and Systems*, Vol. 2, pp 943 – 946.

Biswas, S., 2003. Segmentation based compression for gray level images. In *Pattern Recognition 36*, pp 1501 – 1517.

Delp, E., and Mitchell, O., 1979. Image Compression Using Block Truncation. In *IEEE Trans. Comm.*, Vol. Com-27, No. 9, pp 1335 – 1342.

Eden, M., Unser, M., and Leonardi, R., 1986. Polynomial representation of pictures. In *Signal Processing* 10, 385–393.

Egger, O., Fleury, P., Ebrahimi, T., and Kunt, M., 1999. High-performance compression of visual information—a tutorial review—Part I: still pictures. In *Proc. IEEE*, Vol. 87, No. 6, pp 976 – 1011.

Furht, B., 1995. A survey of multimedia compression techniques and standards Part I: JPEG Standard. In *Real-time Imaging*, pp 49–76.

Habibi, A., 1977. Survey of adaptive image coding technique. In *IEEE Trans. Comm.*, Vol. Com-25, No. 11, pp 1275 – 1284.

Hasegawa, M., and Yamasaki, I., 2002. Image data compression with nonuniform block segmentation and luminance approximation using bilinear curved surface patches. In *Systems and Computers in Japan*, Vol. 33, No. 10, pp 31 – 40.

Hemami, S., and Gray, R., 1997. Subband-coded image reconstruction for lossy packet networks. In *IEEE Trans. IP*, Vol. 6, No. 4, pp 523 – 539.

Jiang, J., 1999. Image compression with neural networks – A survey. In *Signal Processing: Image Communication* 14, pp 737 – 760.

Karczewics, M., Nieweglowski, J., and Haavisto, P., 1997. Video coding using motion compensation with polynomial motion vector fields. In *Signal Processing: Image Communication* 10, pp 63 – 91.

Kim, H., and Lee, J., 2002. Image coding by fitting RBF-surfaces to subimages. In *Pattern Recognition Letters* 23, pp 1239–1251.

Laha, A., Pal, N., and Chanda, B., 2004. Design of vector quantizer for image compression using self-organizing feature map and surface fitting. In *IEEE Trans. IP*, Vol. 13, No. 10, pp 1291 – 1303.

Li, W., and Zhang, Y., 1995. Vector–based signal processing and quantization for image and video compression. In *Proc. IEEE*, Vol. 83, No. 2, pp 317 – 335.

Lim, Y. and Park, K., 1988. Image segmentation and approximation through surface type labeling and region merging. In *Elect. Lett.*, Vol. 24 No. 22, pp 1380 – 1381.

Lin, Y. and Vaidyanathan, P., 1996. Theory and design of two-dimensional filter banks: a review. In *Multidimensional Systems and Signal Processing* 7, pp 263 – 330.

Lu, T., Le, Z., and Yun, D., 2000. Piecewise linear image coding using surface triangulation and geometric compression. In *Proc. Data Compression Conference*, pp 410 – 419.

Nguyen, T., and Oommen, B., 1997. Moment-preserving piecewise linear approximations of signals and images. In *IEEE Trans. PAMI*, Vol. 19, No. 1, pp 84 – 91.

Sinha, S., and Schunck, B., 1992. A two stage algorithm for discontinuity preserving surface reconstruction. In *IEEE Trans. PAMI*, Vol. 14, No. 1, pp 36 – 55.

Strobach, P., 1991. Quadtree-structured recursive plane decomposition coding of images. In *IEEE Trans. SP*, Vol. 39, No. 6, pp 1380 – 1397.

Watanabe, T., 1997. Picture coding employing B-spline surfaces with multiple vertices. In *Elect. & Comm. in Japan Part I*, Vol. 80, No. 2, pp 55 – 65.

Wohlberg, B. and de Jager, G., 1999. A review of the fractal image coding literature. In *IEEE Trans. IP*, Vol. 8, No. 12, pp 1716 – 1729.