

MULTIRESOLUTION DECOMPOSITION BASED DATA COMPRESSION WITH B-SPLINES

Ruşen Öktem¹, Moncef Gabbouj²

Signal Processing Laboratory, Tampere University of Technology

P.O. Box 553, FIN-33101 Tampere, Finland

¹email: rusen@cs.tut.fi ² email: moncef@cs.tut.fi

ABSTRACT

A new multiresolution decomposition scheme based on constrained B-Spline decimation/interpolation is proposed. The proposed decomposition algorithm is tested on greyscale and color image compression, and the results are compared with DCT based compression. Dithering and postprocessing are applied to the compressed images to deal with artifacts at low bit rate compression.

1. INTRODUCTION

Multiresolution decomposition refers to analyzing an image at different resolution levels. Development of Laplacian Pyramidal image compression algorithm by Burt and Anderson [1], accelerated the improvement of multiresolution analysis based image compression techniques. Unser *et al* compared cubic spline pyramid with Laplacian pyramid in [2], and found out that cubic spline pyramid constructed by successive least squares spline decimations leads to a better preserving of details in each multiresolution level. In this work, a novel multiresolution decomposition structure is developed by a set of constrained least squares spline approximations. Each resolution level is represented by B-Spline functions and corresponding coefficients which are used as transform coefficients. The number of B-Spline coefficients representing each decomposition level are reduced by introducing orthogonality constraints. The proposed decomposition structure is used in a coding algorithm which is applied to low bit rate greyscale and color image compression. In case of greyscale image compression, the proposed algorithm is combined with transform domain dithering for removing artifacts caused by the lossy compression around edges.

2. LEAST SQUARES B-SPLINE DECIMATION

Consider a space \mathbf{S} spanned by cubic cardinal basis-spline functions. Any function $f(x)$ of \mathbf{S} can be written as

$$f(x) = \sum_{j=-\infty}^{\infty} \beta(x-j)p_j, \quad (1)$$

where p_j are the spline coefficients and $\beta(x)$ is the cubic B-Spline function which has a finite support from -2 to 2 [3]. $f(x)$ is defined to be the best approximation from \mathbf{S} to a function $z(x)$ in L_2 sense if,

$$\forall f(x) \in \mathbf{S}, \quad \langle f(x), z(x) - f(x) \rangle = 0, \quad (2)$$

where $\langle \cdot \rangle$ is the inner product operator.

Let $\mathbf{z}_0 = [z_1, z_2, \dots, z_N]^T$ be an N -long data vector corresponding to data at (x_1, \dots, x_N) . Best approximation of \mathbf{z}_0 from \mathbf{S} with M spline coefficients $\mathbf{p}_1 = [p_1, \dots, p_M]^T$ can be obtained by using equations (1-2),

$$\mathbf{p}_1 = (\mathbf{B}_1^T \mathbf{B}_1)^{-1} \mathbf{B}_1^T \mathbf{z}_0, \quad \hat{\mathbf{z}}_0 = \mathbf{B}_1 \mathbf{p}_1, \quad (3)$$

where

$$\hat{\mathbf{z}}_0 = [\hat{z}(x_1) \dots \hat{z}(x_N)]^T, \quad \hat{z}(x_i) \in \mathbf{S}, \quad (4)$$

$$\mathbf{B}_1 = \begin{bmatrix} \beta(x_1 - l + 1) & \dots & \beta(x_1 - l + M) \\ \vdots & & \vdots \\ \beta(x_N - l + 1) & \dots & \beta(x_N - l + M) \end{bmatrix}, \quad (5)$$

\mathbf{B}_1 is the $N \times M$ matrix of basis spline functions and \mathbf{p}_1 is the $M \times 1$ vector of spline coefficients. A decimation matrix $\mathbf{B}_{1,s}$ can be obtained by taking every N/M 'th row of \mathbf{B}_1 and discarding the others. Hence, least squares decimation follows,

$$\mathbf{z}_1 = \mathbf{B}_{1,s} \mathbf{p}_1. \quad (6)$$

3. B-SPLINE MULTIREOLUTION DECOMPOSITION

Consider the problem of decomposing \mathbf{z}_0 into $m + 1$ different levels of resolution. Let $\hat{\mathbf{z}}_0$ given by equation (3) be the coarsest resolution level. It can be uniquely represented by a set of spline functions and M -long coefficient vector \mathbf{p}_1 (see equation (3)), where $M = N/2^m$. Let \mathbf{d}_1 be the 1'st scale difference vector which is expressed as,

$$\mathbf{d}_1 = \mathbf{z}_0 - \hat{\mathbf{z}}_0 = (\mathbf{I}_N - \mathbf{B}_1(\mathbf{B}_1^T \mathbf{B}_1)^{-1} \mathbf{B}_1^T) \mathbf{z}_0 \quad (7)$$

where \mathbf{I}_N is the identity matrix of size $N \times N$. A higher resolution level representation for \mathbf{z}_0 can be obtained by adding an approximation of \mathbf{d}_1 to $\hat{\mathbf{z}}_0$. Let $\hat{\mathbf{d}}_1$ be the best approximation (in L_2) to \mathbf{d}_1 obtained in the same way as $\hat{\mathbf{z}}_0$, from basis spline functions and corresponding $N/2^{m-1}$ parameters. From equation (7), it is seen that \mathbf{d}_1 is orthogonal to the space spanned by basis functions of $\hat{\mathbf{z}}_0$, that is

$$\underbrace{(\mathbf{B}_1^T \mathbf{B}_1)^{-1} \mathbf{B}_1^T}_{\mathbf{F}_1} \mathbf{d}_1 = \mathbf{0}. \quad (8)$$

Impose a constraint such that, the best approximation of \mathbf{d}_1 , which is $\hat{\mathbf{d}}_1$, is also orthogonal to the space spanned by the basis functions of $\hat{\mathbf{z}}_0$, i.e. $\mathbf{F}_1 \hat{\mathbf{d}}_1 = \mathbf{0}$. This constraint introduces $N/2^m$ new equations for constructing $\hat{\mathbf{d}}_1$. Hence, the number of coefficients needed to represent $\hat{\mathbf{d}}_1$ is reduced by half. The new $N/2^m$ -long coefficient vector representing $\hat{\mathbf{d}}_1$ is called \mathbf{p}_2 .

The 2'nd scale difference vector is expressed as,

$$\mathbf{d}_2 = \mathbf{d}_1 - \hat{\mathbf{d}}_1, \quad (9)$$

which is orthogonal to both spaces spanned by the basis functions of $\hat{\mathbf{z}}_0$ and $\hat{\mathbf{d}}_1$. The next higher resolution level data can be obtained by adding an approximation of \mathbf{d}_2 to $\hat{\mathbf{z}}_0 + \hat{\mathbf{d}}_1$. Let $\hat{\mathbf{d}}_2$ be the best approximation (in L_2) to \mathbf{d}_2 obtained in the same way as $\hat{\mathbf{z}}_0$, from basis spline functions and corresponding $N/2^{m-2}$ parameters. Impose a constraint such that, the best approximation of \mathbf{d}_2 , which is $\hat{\mathbf{d}}_2$, is also orthogonal to the spaces spanned by the basis functions of $\hat{\mathbf{z}}_0$ and $\hat{\mathbf{d}}_1$. This constraint introduces $N/2^{m-1}$ new equations for constructing $\hat{\mathbf{d}}_2$. Hence, the number of coefficients needed to represent $\hat{\mathbf{d}}_2$ is reduced by half. The new $N/2^{m-1}$ -long coefficient vector representing $\hat{\mathbf{d}}_2$ is called \mathbf{p}_3 .

As in perfect reconstruction based multiresolution decomposition, similar steps can be performed until $\hat{\mathbf{d}}_m$ is represented by $N/2$ -long vector of coefficients. Those steps lead to a set of recursive equations,

$$\mathbf{p}_i = (\mathbf{E}_i^T \mathbf{E}_i)^{-1} \mathbf{E}_i^T \mathbf{d}_{i-1}, \quad (10)$$

$$\mathbf{d}_{i-1} = (\mathbf{I} - \mathbf{E}_{i-1}(\mathbf{E}_{i-1}^T \mathbf{E}_{i-1})^{-1} \mathbf{E}_{i-1}^T) \mathbf{d}_{i-2}, \quad (11)$$

$$\mathbf{E}_i = \mathbf{B}_i \mathbf{P}_i \begin{bmatrix} \mathbf{I} \\ \vdots \\ -(\mathbf{B}^{(i,r)})^{-1} \mathbf{B}^{(i,p)} \end{bmatrix}, \quad (12)$$

$$\mathbf{F}_{i-1} = \begin{bmatrix} \mathbf{F}_{i-2} \\ (\mathbf{E}_{i-1}^T \mathbf{E}_{i-1})^{-1} \mathbf{E}_{i-1}^T \end{bmatrix} \quad (13)$$

with the initial conditions,

$$\mathbf{E}_1 = \mathbf{B}_1, \quad \mathbf{d}_0 = \mathbf{z}_0, \quad \mathbf{F}_1 = (\mathbf{E}_1^T \mathbf{E}_1)^{-1} \mathbf{E}_1^T, \quad (14)$$

where \mathbf{B}_i is the $N \times N/2^i$ B-Spline decimation matrix, \mathbf{d}_i is the difference data representing high frequency components, \mathbf{P}_i is some permutation matrix, and $\mathbf{B}^{(i,p)}$ and $\mathbf{B}^{(i,r)}$ are formed by the first $N/2^{i+1}$ and the last $N/2^{i+1}$ columns of $(\mathbf{F}_{i-1} \mathbf{B}_i \mathbf{P}_i)$, respectively. The recursive formulations (10-14) lead to a transformation equation,

$$\mathbf{p} = \mathbf{A} \mathbf{z}_0, \quad (15)$$

where

$$\mathbf{p} = [\mathbf{p}_1 \ \mathbf{p}_2 \ \dots \ \mathbf{p}_{m+1}]^T, \quad \mathbf{A} = [\mathbf{A}_1^T \ \mathbf{A}_2^T \ \dots \ \mathbf{A}_{m+1}^T]^T, \quad (16)$$

$$\mathbf{A}_i = (\mathbf{E}_i^T \mathbf{E}_i)^{-1} \mathbf{E}_i^T \prod_{j=1}^{i-1} (\mathbf{I} - \mathbf{E}_{j-1}(\mathbf{E}_{j-1}^T \mathbf{E}_{j-1})^{-1} \mathbf{E}_{j-1}^T). \quad (17)$$

\mathbf{A} is called B-Spline Transform matrix. It is composed of $m + 1$ block matrices, where the size of \mathbf{A}_i is $N/2^{m-i+2} \times N$, for $i = 2, \dots, m + 1$, and that of \mathbf{A}_1 is $N/2^m \times N$.

Matrix \mathbf{A} has the following important properties:

- It provides a multiscale resolution decomposition for the input data. $\mathbf{A}_1 \mathbf{z}_0$ represents the low pass characteristics of \mathbf{z}_0 , and $\mathbf{A}_i \mathbf{z}_0$ for $i = 2, \dots, m + 1$ represents the high pass characteristics of \mathbf{z}_0 at m different scales of resolution.
- It is completely data independent, constructed from the permutation and B-spline basis functions matrices.
- The small support of B-spline functions provides that \mathbf{A}_i has few nonzero elements.
- Although \mathbf{A} is not an orthogonal matrix,

$$\mathbf{A}_i \mathbf{A}_j^T = \mathbf{0}, \quad \forall i \neq j, \quad \mathbf{A}_i \mathbf{A}_i^T \neq \mathbf{I}. \quad (18)$$

\mathbf{A} is real with full rank. That is, an orthogonal basis for the span of \mathbf{A} can be constructed via *Gram Schmidt Orthogonalization* procedure, applying it to each block separately so that

$$\tilde{\mathbf{A}} \tilde{\mathbf{A}}^T = \mathbf{I}. \quad (19)$$

- Both the orthogonalized and nonorthogonalized \mathbf{A}_i 's have one principal basis function, where each row is a shifted version of that principal basis function.
- $\tilde{\mathbf{A}}$ has a high energy compaction efficiency.

4. B-SPLINE TRANSFORM APPLIED TO IMAGE COMPRESSION

B-Spline transform can be generalized for 2-D data as,

$$\mathbf{Z}_t = \tilde{\mathbf{A}}\mathbf{Z}_0\tilde{\mathbf{A}}^T, \quad (20)$$

where \mathbf{Z}_t is the transformed 2-D data, analyzed at different scales of resolution, and $\tilde{\mathbf{A}}$ is the orthogonalized B-Spline Transform matrix. \mathbf{Z}_t has the same total energy as \mathbf{Z}_0 . To obtain an efficient compression of input image data, the transform coefficients matrix \mathbf{Z}_t is quantized with a uniform quantizer. After quantization, the high frequency coefficients which consist of few nonzero elements, are generally collected in some regions. A quadtree algorithm [Chapter 7, 4], provides an efficient way of decomposing blocks of high frequency coefficients into smaller blocks of only zeros or nonzeros. Hence, only nonzero coefficients and their locations which are output of the quadtree algorithm are needed to be stored or transmitted after Huffman encoding.

5. SIMULATION RESULTS WITH IMAGE COMPRESSION

B-Spline Image Compression is tested for a set of 512×512 greyscale and color images. The results are compared with Discrete Cosine Transform used in the same compression algorithm. Quantitative results of greyscale image compression are presented in Tables I-II. Figure 1 presents Lenna image at four different resolution levels and Figure 2 presents the difference images. In Figures 3-5, compression results for Lenna and Harbour images are displayed. The quantitative results present better performance for B-Spline Transform than DCT. However, B-Spline Transform results in ringing artifacts around the edges of greyscale images, whereas DCT results in blockiness. To remove artifacts caused by B-Spline image compression, transform domain dithering followed by spline smoothing [Chapter 14, 3] is applied. Transform domain dithering is the operation of adding zero-mean pseudorandom white noise to the transform coefficients before quantization. The same noise is subtracted from the decoded coefficients before inverse transform. The results of dithering and smoothing are displayed in Figure 6 and Table III.

Table 1: Energy Packing Efficiency of the first 64×64 (lowest resolution) coefficients for 512×512 images.

Transform type	Image			
	Lenna	Harbour	Bridge	Cablecar
2D (8×8) block DCT	0.9917	0.9744	0.9657	0.9825
B-Spline Transform	0.9934	0.9753	0.9682	0.9845

In color image applications, YUV color model is used where U and V components are sampled at half the sampling rate. Visual artifacts which are seen at Y,U and V components separately, are not so visible in RGB image. Hence, color image compression with B-Spline Transform performs better both visually and quantitatively than DCT. The results are presented in Figures 7-9 and Table IV.

6. CONCLUSIONS

A new transformation based on cubic B-Spline modeling is proposed and its application to image compression is presented. The new transformation provides multiresolution decomposition of input data, while collecting the most of the input energy in the lowest resolution level coefficients. Test results show better performance than DCT based compression especially in the smooth areas. Applying transform domain dithering and postprocessing with smoothing result in better performance near the edges as well.

7. REFERENCES

- [1] P. J. Burt and E. H. Adelson, "The Laplacian Pyramid as a Compact Image Code", *IEEE Transactions On Communications*, Vol. COM-31, No. 4, pp. 532-679, April 1983.
- [2] M. Unser, A. Aldroubi and Murray Eden, "The L_2 Polynomial Spline Pyramid", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 15, No. 4, pp. 364-379, April 1993.
- [3] C. De Boor, "A Practical Guide to Splines", *Applied Mathematical Sciences*, Vol. 27, Springer-Verlag, 1978.
- [4] R. J. Clarke, *Digital Compression of Still Images and Video*, Academic Press, 1995.



Figure 1: Lenna image decomposed into four resolution levels by B-Spline Transform. a.) Lowest b.) second lowest c.) second highest d.) highest resolution levels.

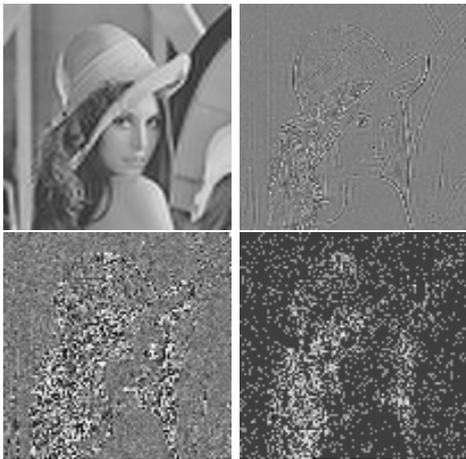


Figure 2: a.) Lenna image at the lowest resolution scale. b.) 2 times magnified first, c.) 10 times magnified second d.) 10 times magnified third scale difference images.



Figure 3: *Left* : Original Lenna image. *Right* : Original Harbour image .



Figure 4: Reconstructed from DCT coded *Left* : Lenna *Right* : Harbour images.



Figure 5: Reconstructed from B-Spline Transform coded *Left* :Lenna *Right* : Harbour images.



Figure 6: *Left* : Dithering applied to Lenna image. *Right* : Dithering followed by smoothing.



Figure 7: *Left* : Original Lenna image. *Right* : Original Sailboat image.



Figure 8: Reconstructed from DCT coded *Left* : Lenna *Right* : Sailboat images.



Figure 9: Reconstructed from B-Spline Transform coded *Left* : Lenna image. *Right* : Sailboat images.

Table 2: PSNR and MAE vs bitrate results with quantizer step size = 35.

Image	DCT		
	PSNR (dB)	MAE	bitrate (bpp)
Lenna	32.57	3.50	0.23
Harbour	28.62	6.04	0.48
Bridge	26.85	8.48	0.56
Cablecar	31.54	3.56	0.32
Cornfield	30.50	4.20	0.41
Ball	38.51	0.69	0.11
Image	B-Spline Transform		
	PSNR (dB)	MAE	bitrate (bpp)
Lenna	32.87	3.12	0.19
Harbour	28.82	5.90	0.43
Bridge	26.96	8.35	0.51
Cablecar	31.70	3.53	0.28
Cornfield	30.71	4.20	0.35
Ball	40.22	0.59	0.11

Table 3: PSNR and MAE vs bitrate results with dithering applied in transform domain.

Image	dithering		
	PSNR (dB)	MAE	bitrate (bpp)
Lenna	29.94	6.31	0.19
Harbour	27.56	8.06	0.44
Cablecar	29.16	6.81	0.29
Image	dithering+smoothing		
	PSNR (dB)	MAE	bitrate (bpp)
Lenna	32.57	4.11	0.19
Harbour	28.61	6.61	0.44
Cablecar	31.21	5.05	0.29

Table 4: PSNR and MAE vs bitrate results with quantizer step size = 21, 17, 17 for Y, U, and V components, respectively.

Image	DCT		
	PSNR (dB)	MAE	bitrate (bpp)
Lenna	27.13	14.12	0.70
Sailboat	26.44	15.72	1.07
Fruits	29.82	5.64	0.58
House	30.03	10.23	0.49
Peppers	29.63	10.77	0.70
Image	B-Spline Transform		
	PSNR (dB)	MAE	bitrate (bpp)
Lenna	27.13	14.13	0.60
Sailboat	27.66	14.00	0.99
Fruits	31.63	4.95	0.51
House	30.91	9.42	0.41
Peppers	30.71	9.81	0.55