Lossless Wavelet Based Image Compression with Adaptive 2D Decomposition

Manfred Kopp
Technical University of Vienna
Institute of Computer Graphics
Karlsplatz 13/186-2, A-1040 Wien, Austria
email: m.kopp@ieee.org
WWW: http://www.cg.tuwien.ac.at/~kopp/

Abstract

2D wavelets are usually generated from 1D wavelets through the rectangular or through the square decomposition scheme. In this paper a new adaptive 2D decomposition scheme for compression related applications is presented. The adaptive 2D decomposition selects 2D wavelet functions based on the compression of the coefficients, but needs only the same number of 1D filter operations as the rectangular decomposition for the compression and even less for the decompression. Results for lossless image compression have shown improvements in the compression rate between 1% and 10% compared to the square decomposition. Only in the case of very small images (below 50x50) the adaptive decomposition was outperformed by the square decomposition because of the overhead to store the selection, which 2D wavelets should be used.

Keywords: image compression, wavelets, adaptive 2D wavelet decomposition

1. Introduction

Wavelets and wavelet transformations have a wide variety of different applications in computer graphics including radiosity [1], multiresolution painting [2], curve design [3], mesh optimization [4], volume visualization [5], image searching [6], animation control [7], BRDF representation [8], and, one of the first applications in computer graphics, image compression [9][10][11][12]. As it is noticed in the preamble of [13], wavelets and wavelet transforms can become as important and ubiquitous in computer graphics as spline based techniques are now.

In this paper a new decomposition scheme for 2D wavelets based on adaptive selection of 2D wavelets based on the compression of their coefficients is presented and the application of this scheme in the field of lossless image compression is discussed. Chapter 2 gives a short introduction in the theory of orthonormal wavelet transformations and multiresolution analysis in one dimension. Chapter 3 gives an overview of extensions to generate multidimensional wavelets. In Chapter 4, the proposed method, the adaptive 2D decomposition is presented. Chapter 5 deals with some special aspects of wavelet based lossless image compression. Experimental results with some standard test images of photographs and also with some computer generated images are analysed in Chapter 6, and conclusions with some future research directions are outlined in Chapter 7.

2. Orthonormal Wavelets in 1D

This short introduction deals only with a subset of wavelets. A more detailed overview of the different kinds of wavelet transforms including continuous wavelet transform, frames, and biorthogonal wavelets can be found in [14] and [15].

The orthonormal wavelet transform is based on two functions $\psi(x)$ and $\phi(x)$, which have the properties:

$$\int \phi(x)dx = 1; \qquad \int \psi(x)dx = 0 \tag{1}$$

These functions with their translations and dilatations $\psi_{j,k}(x)$ and $\phi_{j,k}(x)$ build an orthonormal basis and therefore any function in $L^2(R)$ can be reconstructed with these basis functions. $\phi(x)$ is called scaling- or smooth-function, and $\psi(x)$ wavelet or detail function. $\psi_{j,k}(x)$ and $\phi_{j,k}(x)$ can be constructed from their mother functions $\psi(x)$ and $\phi(x)$ in the following manner:

$$\phi_{j,k}(x) = \sqrt{2^j}\phi(2^j x - k), \qquad \psi_{j,k}(x) = \sqrt{2^j}\psi(2^j x - k), \quad j,k \in Z \tag{2}$$

$\{\phi_{j,k} \mid k \in Z\}$ form an orthonormal basis of functions in vector space V^j. These vector spaces are nested, that is, $V^0 \subset V^1 \subset V^2 \subset V^3 \subset \ldots$. Given a function f(x) over [0,1], this function can be aproximated in V^j with 2^j scaling coefficients s_k^j:

$$f^j(x) = \sum_{k=0}^{2^j-1} s_k^j \phi_{j,k}(x) \quad \text{with} \quad s_k^j = \left\langle f(x), \phi_{j,k}(x) \right\rangle = \int f(x)\phi_{j,k}(x)\,dx \tag{3}$$

Also the detail functions $\{\psi_{j,k} \mid k \in Z\}$ form an orthonormal basis of functions in the detail vector space W^j, which is the orthogonal complement of V^j in V^{j+1}. W^j can be thought of as containing the detail in V^{j+1}, which can not be represented in V^j. The vector space V^{j+1} can be decomposed in the following manner:

$$V^{j+1} = V^j \oplus W^j = V^{j-1} \oplus W^{j-1} \oplus W^j = \ldots = V^0 \oplus W^0 \oplus W^1 \oplus \ldots \oplus W^j \tag{4}$$

Let d_k^j be the detail coefficients, given through:

$$d_k^j = \left\langle f(x), \psi_{j,k}(x) \right\rangle = \int f(x)\psi_{j,k}(x)\,dx, \tag{5}$$

then $f^j(x)$ can be calculated from the detail coefficients $\{d_k^i \mid i<j\}$ and the scaling coefficient s_0^0 as follows:

$$f^j(x) = f^{j-1}(x) + \sum_{k=0}^{2^{j-1}-1} d_k^{j-1} \psi_{j-1,k}(x) = s_0^0 \phi_{0,0}(x) + \sum_{m=0}^{j-1} \sum_{k=0}^{2^m-1} d_k^m \psi_{m,k}(x) \tag{6}$$

The calculation of the coefficients $\{ s_0^0, d_k^i \mid 0 \le i < j; 0 \le k < 2^i \}$ from the scaling coefficients $\{ s_k^j \mid 0 \le k < 2^j \}$ is called wavelet transformation. The fast wavelet transformation uses a pyramid scheme with two subband filters, the smoothing or scaling filter (h_m), and the detail or wavelet filter (g_m). In one transformation step the 2^i scaling coefficients s_k^i are replaced by 2^{i-1} scaling coefficients s_k^{i-1} and 2^{i-1} detail coefficients d_k^{i-1} :

$$s_k^{i-1} = \sum_m h_{2k-m} s_m^i; \qquad d_k^{i-1} = \sum_m g_{2k-m} s_m^i \tag{7}$$

This step is repeated on the remaining scaling coefficients, until s_0^0 is computed. The reconstruction step can be performed using the adjoint filtering operation:

$$s_k^i = \sum_m h_{k-2m} s_m^{i-1} + g_{k-2m} d_m^{i-1} \tag{8}$$

3. Wavelets in Higher Dimensions

The definitions in Chapter 2 deal with wavelets in 1D space, but for image compression wavelet transformations in 2D are needed. One way to extend the formulas to 2D is to use a dilatation matrix D in (2) as described in [16], instead of a simple dilatation factor, e.g. the quincunx scheme uses the dilatation matrix $D = \left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$.

More popular methods extend the one dimensional wavelets to higher dimensions with tensor products of 1D wavelets and scaling functions, i.e. the rectangular and the square wavelet basis functions.

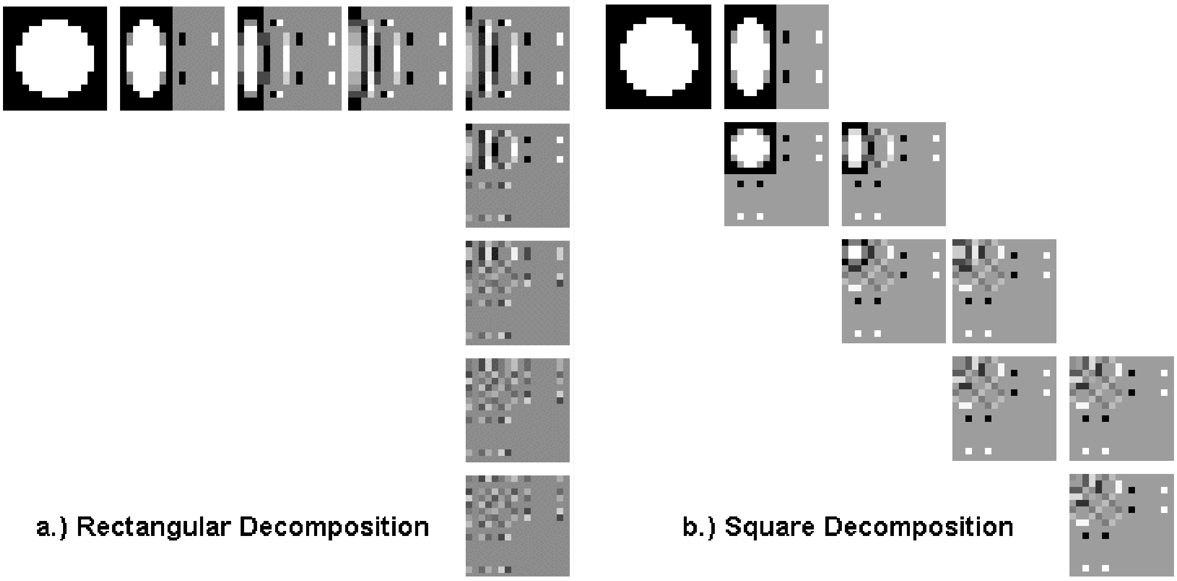
3.1 The Rectangular Decomposition

The rectangular or standard wavelet basis functions are generated through the carthesian product of the 1D wavelet basis functions in every dimension. In the 2D case, the rectangular wavelet basis functions are:

$$\phi(x,y)=\phi_{0,0}(x)\phi_{0,0}(y) \qquad \ldots\text{2D scaling function}$$

$$\left.\begin{aligned} {}^{R1}\psi_{j,k}(x,y) &= \psi_{j,k}(x)\phi_{0,0}(y) \\ {}^{R2}\psi_{j,k}(x,y) &= \phi_{0,0}(x)\psi_{j,k}(y) \\ {}^{R3}\psi_{j,m,k.n}(x,y) &= \psi_{j,k}(x)\psi_{m,n}(y) \end{aligned}\right\} \qquad \begin{matrix} \ldots\text{2D wavelet functions} \\ 0\le j,m<L; \\ 0\le k<2^j;\,0\le n<2^m \end{matrix} \tag{9}$$

The fast wavelet transformation with the rectangular basis wavelets, also known as rectangular decomposition, is computed by successively applying the 1D wavelet transformation to the data in every dimension. In the 2D case, all the rows are transformed first, then a 1D wavelet transformation is applied on all columns of the intermediate result. Figure 1a illustrates the rectangular decomposition. The wavelet coefficients of the 1D transformation steps are stored in the right (row transform) or lower (column transform) part, the scaling coefficients in the left or upper part, respectively.



a.) Rectangular Decomposition

b.) Square Decomposition

figure 1

3.2 The Square Decomposition

The square or nonstandard wavelet basis functions are also generated through cartesian product of 1D wavelets and 1D scaling functions. In contrast to the rectangular basis functions, the square basis functions always use tensor products of wavelet and/or scaling functions of the same resolution level. In the 2D case, the square wavelet basis functions are:

$$\phi(x,y) = \phi_{0,0}(x)\phi_{0,0}(y) \qquad \ldots \text{2D scaling function}$$

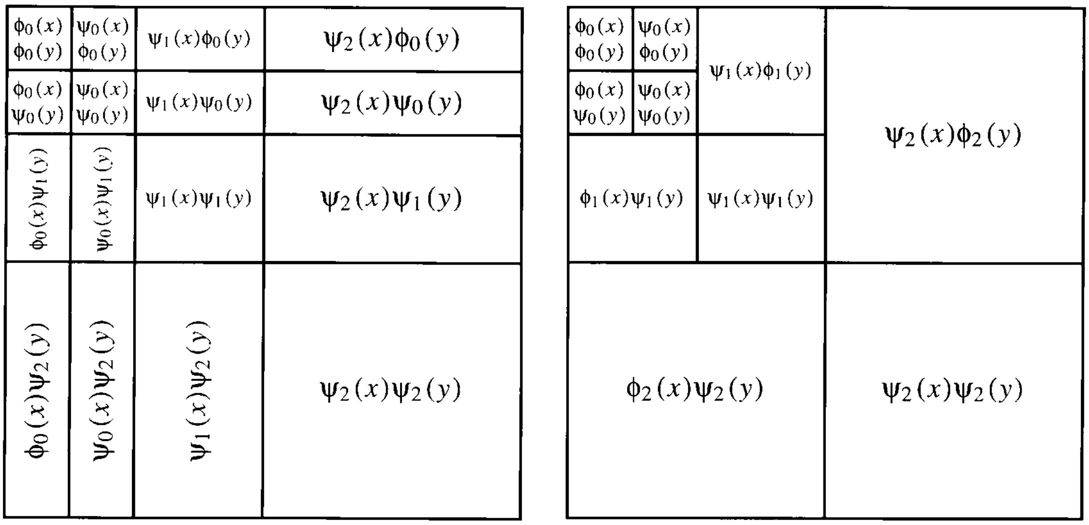
$$\left.\begin{aligned} {}^{h}\psi^{j}_{k,m}(x,y) &= \psi_{j,k}(x)\phi_{j,m}(y) \\ {}^{v}\psi^{j}_{k,m}(x,y) &= \phi_{j,k}(x)\psi_{j,m}(y) \\ {}^{d}\psi^{j}_{k,m}(x,y) &= \psi_{j,k}(x)\psi_{j,m}(y) \end{aligned}\right\} \quad \begin{aligned} &\ldots \text{2D wavelet functions} \\ &\quad 0 \le j < L;\ 0 \le k,m < 2^{j} \end{aligned} \qquad (10)$$

The square wavelet decomposition can be computed with a similar technique as the rectangular decomposition: A 1D wavelet transformation step, as described in (7) - not the whole wavelet transformation as in the rectangular decomposition - is applied in every dimension. This generates (2^{dim} -1) subbands with wavelet coefficients and one subband with scaling coefficients. This transformation scheme is applied recursively on the scaling coefficients until the lowest level is reached (figure 1b). The square decomposition is slightly more efficient to compute than the rectangular decomposition: For an $m \times m$ image only (8/3)(m^2-1) assignments are needed, compared to 4(m^2-m) in the rectangular decomposition. Also the compression ratios are usually better for the square decomposition, because the support of the wavelet functions are square and support width of the wavelet basis functions are lower or equal than their counterparts in the rectangular decomposition and therefore they exploit more locality.

4. The new method: Adaptive 2D Wavelet Decomposition

Let us take a closer look at the difference of the square and the rectangular decomposition in the 2D case: The first obvious similarity of the two schemes is the use of the same scaling function ϕ(x,y), whose coefficient is stored in the upper left corner of the transformed images.

In one decomposition step of the square decomposition three wavelet subbands and one scaling subband are generated. The wavelet subbands are not altered in the following decomposition steps. The first wavelet subband in the upper right part of the transformed image consists of the coefficients of the wavelet functions ${}^{h}\psi$ as defined in (10), the second subband in the lower left part has the coefficients of ${}^{v}\psi$, and the third subband in the lower right part holds the coefficients of ${}^{d}\psi$. All the wavelet functions in ${}^{d}\psi$ are also contained in ${}^{R3}\psi$ of the rectangular decomposition, therefore the coefficients in this part of the transformed image are the same for both decompositions. The upper right part of the square transformed image contains coefficients of the wavelet functions ${}^{h}\psi^{L}(x,y) = \psi^{L}(x)\phi^{L}(y)$, where L is the maximum resolution level. The corresponding rectangular decomposition holds the coefficients of the wavelet functions $\{{}^{R1}\psi_{L}(x,y);\ {}^{R3}\psi_{L,i}(x,y) \mid 0 \le i < L\ \}$ (figure 2). It can be seen from the definition of these wavelet functions in (9)(10), that this part of the transformed image in the rectangular decomposition can be generated from the square decomposition with a 1D wavelet transformation within every column of this part. In analogy, the lower left part of the rectangular decomposition can be generated from the square decomposition with a 1D wavelet transformation within every row.



a.) rectangular decomposition b.) square decomposition

figure 2

This relation between the square decomposition and the rectangular decomposition remains also in the following decomposition steps of the square decomposition with the resolution level L reduced by 1 from the above step.

This observation leads to an alternative construction scheme for the rectangular decomposition:

```
• apply a square decomposition step
• for every column in the upper right part:
        apply a 1D wavelet transform in the y-dimension
• for every row in the lower left part:
        apply a 1D wavelet transform in the x-dimension
• apply this scheme recursively on the upper left part of the transformed image
```

The 1D wavelet transformation consists of an iteration of transformation steps. The idea of the adaptive 2D decomposition is to replace the 1D transformations in the alternative construction scheme of the rectangular decomposition with an optimal number of transformation steps in respect to the compression rate of the coefficients. The pseudo-code of the adaptive 2D decomposition can be written as:

```
• apply a square decomposition step
• for every column in the upper right part:
        apply all 1D wavelet decomposition steps in the y-dimension
        calculate the compression rates for all steps
        select the number of steps with optimal compression rate
• for every row in the lower left part:
        apply all 1D wavelet decomposition steps in the x-dimension
        calculate the compression rates for all steps
        select the number of steps with optimal compression rate
• apply this scheme recursively on the upper left part of the transformed image
```

Note that this adaptive 2D decomposition also includes the rectangular and the square decomposition: If the square decomposition has the best compression of the coefficients, the adaptive 2D decomposition selects the square decomposition wavelet functions. The same is true for the rectangular decomposition. In the general case, the adaptive 2D decomposition selects some wavelet functions from the square decomposition, some from the rectangular decomposition and some "between" the square and the rectangular decomposition.

There are only $4(m^2-m)$ coefficient assignments needed to do the transformation for an m x m image, the same number as for the rectangular decomposition. The invers transform even needs less coefficient assignments, since the number of transformation steps is usually lower than the maximum.

There is a slight overhead for storing the number of 1D transformation steps: For an $m \times n$ image less than $m \cdot (ld(n)-1) + n \cdot (ld(m)-1)$ bits are needed for storing the adaptive decomposition. For example, an 1024x768 image with 8 bit graylevels needs 768 kBytes for the uncompressed image and about 16 kBytes for the additional data, only about 2% of the original.

5. Wavelets for Lossless Image Compression

Even though there are many papers about wavelet based image compressions, only few deal with the lossless case [11][12].

Lossless wavelet image compressions use a 2D wavelet transform to improve the compression rate of conventional compression algorithms like Huffman or arithmetic coding [17]. Since the pixels have to be reconstructed exactly, some special properties for the wavelet transformations are required.

Let us first consider graylevel images: There is one color channel with a finite number of possible values, it usually has 8 bit depth or 256 shades of gray. For lossless compression it must be guaranteed, that the invers transformation of the transformation does not change the pixels. This can be achieved for all compact wavelets, if the precision of the transformed image is high enough. But for high compression of the coefficients, the needed precision should be as low as possible.

Bekaert et al. [11] used unnormalized Haar wavelets for lossless image compression. Since the least significant bit of the scaling coefficient is redundant, all wavelet coefficients can be stored in 9 bit, in the case of a carthesian product of a 1D scaling function with a 1D wavelet, i.e. coefficients of ${}^{R1}\psi$,${}^{R2}\psi$, ${}^{h}\psi$ and ${}^{v}\psi$ from (9) and (10), or 10 bit, if the 2D wavelet is a product of two 1D wavelets, i.e. coefficients of ${}^{R3}\psi$ and ${}^{d}\psi$.

Zandi et al. [12] proposed the Reversible Two Six (RTS)-wavelet, which is more efficient than the Haar wavelets. Comparisons with state of the art lossless image compression algorithms, including DPCM, JBIG and JPEG, showed better results for the wavelet based method.

Images with colormaps can be dealt with the same algorithm efficiently, if the colormap is (re-)arranged in a way that neighbouring color indices represent colors with slight differences.

RGB direct color images can be handled as three independent color channels, each encoded the same way as a graylevel image.

6. Experimental Results

My implementation of lossless wavelet compression used an adaptive arithmetic coder [17] for the entropy encoding. In the RGB pictures, every color component was handled independently. The proposed decomposition was tested with standard grayscale pictures, including "Lena" and "baboon", and some computer generated RGB images, generated with various programs developed at our Institute. Table 1 shows the compression rates of the 8bit grayscale images with the popular TIFF image format with LZW compression, square decomposition with Haar and RTS wavelets, and adaptive 2D decomposition, also with Haar and RTS-wavelets. In all cases, the adaptive 2D decomposition performed better than the square decomposition. In almost all cases, the RTS wavelet has better results than the Haar wavelet.

name	size	compr. rate TIFF LZW	compr. rate square dec. Haar wav.	compr. rate adapt. dec. Haar wav.	compr.rate square dec. RTS wav.	compr. rate adapt. dec. RTS wav.
airplane	512x512	1.23	1.71	1.74	1.80	1.84
baboon	512x512	0.86	1.25	1.26	1.27	1.29
boats	720x576	1.14	1.70	1.75	1.78	1.85
bridge	256x256	0.87	1.28	1.30	1.29	1.31
camera	256x256	1.20	1.58	1.60	1.54	1.57
couple	256x256	1.23	1.74	1.79	1.77	1.84
goldhill	720x576	1.05	1.58	1.62	1.65	1.68
lena	512x512	0.95	1.57	1.58	1.63	1.65
peppers	512x512	0.98	1.59	1.62	1.65	1.69

table 1

Since the adaptive 2D decomposition produces some overhead for storing the number of optimal transformation steps, which is proportional to m·ld(n)+n·ld(m) in an m×n image, a comparison between the adaptive 2D decomposition and the square decomposition with the same image in different resolutions was made. Table 2 compares the compression rates with RTS wavelets between adaptive 2D decomposition and square decomposition of the "Lena" image in different resolutions. Only in the resolutions below 64x64 pixels the square decomposition performed better, but in this case even the uncompressed image is more efficient.

decomposition	512x512	256x256	128x128	64x64	32x32
square	1.63	1.423	1.242	1.024	0.854
adaptive 2D	1.65	1.442	1.249	1.033	0.851

table 2

A comparison of compression rates of computer generated RGB images is shown in table 3. The wavelet compression uses RTS wavelets and encodes every color channel separately. The images "denker", "trees", and "xmas94" were generated with raytracing programs, "room1" with a radiosity package, and "roessler" with a visualization program.

name	size	compression rate TIFF LZW	compression rate square decomp.	compression rate adapt. decomp.
denker	263x381	2.18	2.09	2.24
roessler	410x343	8.91	3.09	3.42
room1	480x320	1.59	2.49	2.57
trees	800x500	2.30	1.96	2.01
xmas94	800x600	3.23	2.75	2.92

table 3

As expected, all images could be encoded more efficiently with the adaptive decomposition scheme than with the square decomposition. The results also showed, that the simple separate encoding of the color channels looses much coherence, so that the TIFF LZW compression gives better results sometimes. Therefore more sophisticated encoding models for the adaptive arithmetic coder which exploit the coherence between the color channels have to be considered to get better results.

7. Conclusion and Future Work

The new adaptive 2D decomposition scheme offers better compression rates than the square and the rectangular decomposition, if the images are above a threshold size. In the studied case of lossless image compression, this threshold size is about 50x50 pixels. Also a comparison between RTS and Haar wavelets was made, which showed better performance for the RTS wavelets. The experimental results also showed that for color encoding the coherence between the color channels should be exploited, because a simple separate color channel arithmetic encoding generates sometimes worse results than LZW based compression algorithms.

Future research will be made in the application of this adaptive decomposition in other computer graphics areas, including lossy compression, extensions to 3D for volume data compression, video compression, and wavelet radiosity.

8. Notes and Acknowledgements

The grayscale testimages including the famous "Lena" and "baboon" picture were downloaded from the URL `ftp://ftp.funet.fi/pub/graphics/misc/test-images` .

This paper is also available from the URL:

`http://www.cg.tuwien.ac.at/research/TR/95/TR-186-2-95-11Abstract.html` .

If you make a reference to this paper, please include also this URL, since it is easier to access for most people than the conference proceedings itself.

I want to thank Werner Purgathofer and Robert Tobler for their helpful comments in the preparation of this paper.

9. References

[1] Steven Gortler, Peter Schröder, Michael Cohen, Pat Hanrahan, "Wavelet Radiosity", Computer Graphics, Annual Conference Series (Siggraph'93 Proceedings), pp. 221-230, 1993

[2] Deborah Berman, Jason Bartell, David Salesin, "Multiresolution Painting and Compositing", Computer Graphics, Annual Conference Series (Siggraph'94 Proceedings), pp. 85-90, 1994

[3] Adam Finkelstein, David Salesin, "Multiresolution Curves", Computer Graphics, Annual Conference Series (Siggraph'94 Proceedings), pp. 261-268, 1994

[4] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsberry, Werner Stuetzle, "Multiresolution Analysis of Arbitrary Meshes", Computer Graphics, Annual Conference Series (Siggraph'95 Proceedings), pp. 173-182, 1995

[5] L. Lippert, M. Gross, "Fast Wavelet Based Volume Rendering by Accumulation of Transparent Texture Maps", Eurographics'95 Conference Proceedings, Computer Graphics Forum Vol. 14, Nr. 3, pp. 431-443, 1995

[6] Charles Jacobs, Adam Finkelstein, David Salesin, "Fast Multiresolution Image Querying", Computer Graphics, Annual Conference Series (Siggraph'95 Proceedings), pp. 277-286, 1995

[7] Zicheng Liu, Steven Gortler, Michael Cohen, "Hierarchical Spacetime Control", Computer Graphics, Annual Conference Series (Siggraph'94 Proceedings), pp. 35-42, 1994

[8] Peter Schröder, Wim Sweldens, "Spherical Wavelets: Efficiently Representing Functions on a Sphere", Computer Graphics, Annual Conference Series (Siggraph'95 Proceedings), pp. 161-172, 1995

[9] Ronald DeVore, Björn Jawerth, Bradley Lucier, "Image Compression Through Wavelet Transform Coding", IEEE Transactions on Information Theory, Vol. 38, Nr. 2, pp 719-746, March 1992

[10] J. M. Shapiro, "Embedded Image Coding using Zerotrees of Wavelet Coefficients",IEEE Transactions on Signal Processing, Vol. 41, Nr. 12, pp 3445-3462, 1993

[11] Philippe Bakaert, Geert Uytterhoeven, Yves Willems, "An Experiment with Wavelet Image Coding", Proceedings of 11th Spring Conference on Computer Graphics, Bratislava, pp. DM1-6, May 1995

[12] Ahmad Zandi, James Allen, Edward Schwartz, Martin Boliek, "CREW: Compression with Reversible Embedded Wavelets", Preprint from RICOH California Research Center, available from http://www.crc.ricoh.com/misc/crc-publications.html

[13] Alain Fournier, Michael Cohen, Tony DeRose, Michael Lounsbery, Leena-Maija Reissell, Wim Sweldens, "Wavelets and their Applications to Computer Graphics", Siggraph'95 Course Notes 26, Los Angeles, August 1995

[14] Charles K. Chui, "An Introduction to Wavelets", Academic Press Inc., Boston, MA 1992

[15] Ingrid Daubechies, "Ten Lectures on Wavelets", Vol. 61 of CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM, Philadelphia, PA 1992

[16] Jelena Kovacevic, Martin Vetterli, "Nonseparable Multidimensional Perfect Reconstruction Filter Banks and Wavelet Bases for R^n

[17] Ian H. Witten, Radford M. Neal, John G. Cleary, "Arithmetic Coding for Data Compression", CACM Vol. 30, No. 6, pp. 520-540, June 1987