

# Still Image Compression by Nonseparable Wavelets on the Quincunx Lattice

by

*Richard Julian*

Richard Andrews (B.E.hons)

Submitted in fulfilment of the requirements for

the Degree of

Doctor of Philosophy.

*Engineering  
(Electrical)*

University of Tasmania (June 2002)

## ABSTRACT

The recent unification of wavelet and subband theories has allowed the creation of a new field of investigation for the efficient compression of digital images: wavelet compression. It has seen remarkable improvements in compression results over the previous generation of DCT-based image compression schemes. The focus of research in this field has, however, been almost exclusively in the separable domain, which uses one-dimensional transforms.

The use of truly nonseparable wavelet transforms on two-dimensional image signals has been largely ignored. The purpose of this Thesis is to investigate more thoroughly this largely untouched field of wavelet-based image compression.

In this Thesis we discuss in depth the techniques for using multidimensional wavelet transforms and subband coding for image compression and provide results for extending existing compression techniques to the quincunx domain. Various results covering a number of coding methodologies are presented using the quincunx wavelet transform to demonstrate its advantages and disadvantages when compared to the separable decomposition method. Novel techniques are developed for the representation and storage of quincunx sampled images allowing in-place wavelet transforms to be performed in real-time. A novel extension to the Shapiro zero-tree compression method is developed which predicts and exploits, during coding, visually unimportant areas without the need for transmitting side-information. Results are presented which show that this process leads to significantly higher perceived image quality without increasing the bit-rate.

Several advantageous psychovisual properties of the quincunx resampling lattice are exploited in the creation of various extensions to simple compression methods. Results isolating the effects of utilizing these properties are presented.

We find that in general separable wavelet transforms perform better than their quincunx counterparts for bit-rate versus perceived quality of reconstruction, despite the quincunx resampling structure possessing inherent advantages over rectangular resampling. This is mainly attributed to the state of non-separable subband theory and filter design which has not progressed to a state where it is possible to achieve the same quality of filter design as exists in the one dimensional case.

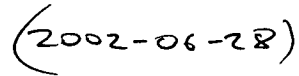


## Declaration of Originality

I declare that this Thesis embodies my own research work, and that it was composed by myself and has not been accepted wholly or partially for the award of any degree or diploma. Where appropriate the work of others is given due acknowledgement.

A handwritten signature in black ink that reads "RJ Andrews". The signature is written in a cursive style with a horizontal line underneath the name.

Richard Andrews

A handwritten date in black ink that reads "(2002-06-28)". The date is enclosed in parentheses and written in a cursive style.

©Richard Andrews 2002

The copyright of this Thesis belongs to the author.

This Thesis may be made available for loan and limited copying in accordance with the Australian *Copyright Act* 1968.

RJ Andrews (2002-06-28)

## Acknowledgements

The production of this Thesis is the culmination of what feels like a life's work. Many people have provided me with inspiration and encouragement during the course of my research without which I would not have been able to finish.

- Romeny for endless love, friendship, patience and gin.
- Emily and Jennifer for being the best mates I could have hoped for.
- The clan McMahon for endless hours of coffee and music.
- Thong Nguyen for the obvious and not so obvious.
- Jelena Kovacevic for sending the material that originally got me going.
- Ingrid Daubechies for hard-copy of a hard to find paper.
- Everyone who contributes to free software which has allowed me to perform my work with confidence.
- All my friends, who whether they know it or not have contributed in some way to making this happen.

The Arithmetic Coding used in this work was provided by John Danskin's implementation contained in the Dartmouth College "Baseline Wavelet Image Coder" package[19].

## About the Experimental Platform

The gathering of experimental data has been a significant part of the research for this Thesis. Working in the nonseparable domain was found to be a difficult and time consuming exercise compared to the separable domain.

Experimentation began using MATLAB with the excellent wavelet tool-box provided. This sufficed for early separable results such as those described in [4]. When the move was made to nonseparable experimentation it was found that MATLAB was too slow. This was due to the resampling necessary at each level of decomposition. The decision was made to move to writing code in C as this would result in faster execution and more rapid debugging iterations for experimental correctness. The disadvantage was that no code examples were available for nonseparable decompositions.

The development platforms used were GNU/Linux systems (various flavours) with GNU gcc and Sun UltraSPARC systems also running GNU gcc. Many thanks go to the volunteer developers of these tools who made this possible.

The C experimental platform was superseded with C++ code which was found to be more reliable and easier to maintain and extend. The experimental platform produced does indeed run much faster than MATLAB allowing nonseparable results to be gathered in reasonable time.

About a 18 months after moving away MATLAB, the “upsampled” representation (Chapter 5) was discovered which eliminates the need for the processor intensive resampling procedure; the reason for abandoning MATLAB in the first place. Nonseparable experimentation is probably feasible using MATLAB and the upsampled representation - this has not been verified as the author has not had access to MATLAB since the discovery of the upsampled representation. However the use of C and C++ code has allowed for the continuation of this research on a part-time basis after the author relocated away from the University to take up full-time employment. The upsampled representation has since been included as an integral part of the C++ experimental code libraries used for the majority of the results presented in this work.

## Advice to the New Researcher

The author strongly advises those researching in the nonseparable field to experiment with the upsampled representation before switching away from established tools such as MATLAB due to speed issues with resampling. Much time

was spent reinventing existing tools for custom code which are already available in established packages.

The upsampled representation is applicable to any resampling lattice (including separable) and so may find use in separable, quincunx, hexagonal or other sampling domains.

### **About the Source Code**

The source code for the experimental platform is available from the author, licensed under the conditions of the GNU General Public License (GPL) from the Free Software Foundation and is freely usable and redistributable under the conditions of that license. The copyright of the code belongs to the author, except for the Arithmetic Coding implementation as outlined in the Acknowledgements.

## About the Production of the Thesis

This Thesis was produced using only free<sup>1</sup> software - exceptions being the early results produced for [4]. Although this was never an aim of the Thesis it is an achievement worthy of note as a testament to the quality and availability of free software. The stability and reliability of GNU/Linux systems provided a motivating factor for the use of free software in the development of the experimental platform and later in the production of this Thesis. The philosophy of the free software movement also aligns very closely with that of cooperative scientific research, where geographically isolated participants can engage in peer-review and collaboration to produce results of the highest calibre.

Operating systems were provided by Slackware, Debian, Mandrake and Red-Hat Linux distributions running on Intel-based hardware. Wordprocessing and mathematical notation processing was performed using the L<sup>A</sup>T<sub>E</sub>X program which provides a graphical interface to the excellent L<sup>A</sup>T<sub>E</sub>X typesetting software which was used to provided typesetting, layout and cross-reference automation (including bibliography and reference management). Experimental results were produced using C and C++ programs written using mainly the KDevelop programming environment and compiled using the GNU compiler collection (gcc) for execution on GNU/Linux operating systems. Graphs and plots were provided by the GNUplot program and all figures were drawn using the XFig program. Image format conversions were performed using the GNU Image Manipulation Program (GIMP). Results were proofed using the xdvi, gv and ggv postscript interfaces to Aladdin Ghostscript which provided postscript rendering on many devices. Web-browsing and file-transfer capabilities were provided by Netscape, Mozilla/Galeon, Konqueror and SAMBA to name just a few.

---

<sup>1</sup>“Free” refers to the definition as adopted by the Free Software Foundation (<http://gnu.org>) meaning free as in “free speech”.

# Contents

ABSTRACT	i
Declaration of Originality	ii
	iii
Acknowledgements	iv
About the Experimental Platform	v
Advice to the New Researcher	v
About the Source Code	vi
About the Production of the Thesis	vii
List of Figures	xi
Reproduced Publications	xxi
Chapter 1. Introduction	1
1.1. Need for image compression	1
1.2. Digital Image Basics	2
1.3. Lossless and Lossy Data Reduction	5
1.4. Statistics of Images	8
1.5. Pixel Coding	10
1.6. Transform Coding	11
1.7. Human Vision	14
1.8. Thesis Aims and Structure	14
1.9. Original Content	16
Chapter 2. Wavelets and Subband Coding	18
2.1. A Brief History of Wavelets	18
2.2. Fourier Transform and Fourier Series	19
2.3. Discrete Fourier Transform	22
2.4. Discrete Cosine Transform	23
2.5. Two-dimensional Fourier Transforms	24
2.6. Inner Products, Correlations and Convolutions	27
2.7. Wavelet Basics	29
2.8. Scaling Functions and Wavelets	31
2.9. Continuous Wavelet Transform	32
2.10. Multiresolution Analysis	34

2.11. Discrete Wavelet Transform	36
2.12. Filters and Subbands	38
2.13. Wavelet Packet Transform	43
2.14. Tensor Product: Separable Wavelets in Multiple Dimensions	44
2.15. Nonseparable Wavelets: True Multidimensional Transforms	46
Chapter 3. The Quincunx Lattice	47
3.1. Lattices and Sub-lattices	48
3.2. Cosets	50
3.3. Unit Cells	52
3.4. Frequency-Space	53
3.5. Nonseparable Sampling	57
3.6. Resampling on the Quincunx Lattice	61
Chapter 4. Nonseparable Wavelet Design	62
4.1. Nonseparable Design Basics	62
4.2. General Results	68
4.3. Lifting	75
Chapter 5. Upsampled Representations and Cross-scale Equivalence	77
5.1. Quincunx Wavelet Transform	77
5.2. Equivalence Across Scales	79
5.3. Upsampled Representation	80
Chapter 6. Coding Frameworks	86
6.1. Quantization	86
6.2. Entropy Coding	88
6.3. Basic Coding Results	90
Chapter 7. Perceptually Efficient Coding and Psychovisual Models	103
7.1. HVS Redundancies	103
7.2. Psychovisual Coding	106
7.3. Results Using HVS Tuning	110
7.4. Distortion Due to Choice of Wavelets	122
7.5. Results Using Various Quincunx Filters	124
Chapter 8. Zero-tree Coding	128
8.1. Introduction To Zero-tree Coding	128
8.2. Zero-tree Coding Results	133
8.3. Zero-tree Coding Methods for Network Transport	140
Chapter 9. Cross-scale Wavelet Image Coding	143
9.1. Introduction To Cross-Scale Coding	143
9.2. Spatial Masking Shadows and Cross-Scale Coding	144
Chapter 10. Masking Shadows in Zero Trees	147



10.1. Utilizing Masking Shadows	147
10.2. Determining Masking Levels	151
10.3. Masking Shadow Prediction Results	151
10.4. Conclusion	161
Chapter 11. Conclusions and Future Work	162
11.1. Summary of Contributions	162
11.2. Further Research	164
Near Zero-trees	165
Appendix A - Original Images	171
REFERENCES	177

## List of Figures

1.2.1	The RGB colour cube visualized as a three-dimensional co-ordinate system.	4
1.2.2	Colour wheel visualization of 3-D colour-space using the HSV model.	5
1.3.1	Common viewing requirements for human vision experiments and models. Viewing distance of full screen images is often assumed to be at a distance of four times the screen height.	7
1.3.2	Comparison of angle and solid-angle. Angle is a 2D measure which defines an arc at some extent. Solid angle defines a spherical surface at some extent.	8
1.5.1	A predictive coding arrangement. The output from the predictor is generally close to the actual input so the difference is small. The error values are output to the symbol stream.	11
2.2.1	Sinusoidal transform kernel for the Fourier Transform. This function extends indefinitely.	20
2.2.2	Arbitrary signal to be analyzed by the Fourier Transform. Since the signal is not periodic in nature it is not suitable for Fourier analysis.	21
2.2.3	The artifact caused by periodization of the arbitrary signal above, for Fourier series analysis. This figure shows one period of the signal being analyzed. Here the origin where $t = k \cdot T$ is in the centre of the figure. The sharp jump in signal value shown is due to the artificial process of making the signal periodic.	21
2.3.1	A sampled sinusoidal kernel function for the DFT. The maximum frequency which can be reproduced from the sampled data is determined by the Nyquist criterion as $f_{max} = \frac{f_s}{2}$ .	22
2.4.1	The lowest frequency DCT kernel function is only half a period long, making it an odd function. The basis functions for the Cosine Transform are spaced at multiples of half a period of the sample period.	24
2.7.1	Tiling of the space-frequency domain by (a) Fourier analysis and (b) Wavelet analysis. Fourier analysis has a fixed resolution	

	in both space and frequency - global in space and singular valued in frequency. Wavelets are localized in both space and frequency.	30
2.8.1	Example of approximation of a signal and associated detail from a scaling-function/wavelet pair. The scaling function captures the low-frequency components, including average, and the wavelet signal captures the high-pass components.	32
2.9.1	Band-splitting property of the dyadic wavelet transform. The approximation (H) retains the low-frequency components while the detail signal (G) contains the rest of the bandwidth of the original signal. Because of previous band-limiting approximations, the high-frequency component is actually a narrow band.	34
2.10.1	Conceptual diagram of complimentary orthogonal function spaces. With DC at the centre, the function space on the right represents an approximation sub space of the original function space. The complimentary space represents those functional capabilities not present in the approximation subspace.	34
2.10.2	Conceptual diagram of nested subspaces show a Multiresolution Analysis. Here the many detail subspaces $W_n$ and the single low-pass subspace $V_N$ sum together to produce the original.	35
2.10.3	Power spectra showing band-splitting nature of the wavelet transform. The original bandwidth of a signal is split into two equal octaves; one low-pass, the other high-pass.	36
2.10.4	Power spectra in multiresolution analysis. This shows how the low-pass band of each level of band-splitting is split further. Each level of decomposition reduces the bandwidth by a factor of 2 (octaves).	36
2.12.1	Lenna image reconstructed from the lowest resolution details of a 6-level separable wavelet decomposition using Daubechies $D_4$ [17] filters. (a) Only the low-pass at level 6, (b) Low-pass plus detail at level 6. (c) Low-pass plus two details - levels 6 and 5. (d) Three details. (e) Four details. (f) Missing only the highest resolution detail.	39
2.12.2	Block diagram of a general filterbank system. There are $n$ channels in each filterbank and each channel performs shifting, filtering and resampling as required. Analysis involves shifting followed by filtering then downsampling. Synthesis involves upsampling, followed by filtering then reverting the shift.	40

2.12.3	Block diagram of the biorthogonal wavelet transform. Perfect reconstruction is possible without the constraints of orthogonality. Here the columns of the matrix $\tilde{H}$ and $H$ are orthogonal to one another.	42
2.13.1	The Wickerhauser Wavelet-Packet Decomposition in one dimension. In this decomposition, both the low and high-pass components are decomposed at each level, resulting in a set of components of equal bandwidth and equal spacing across the spatial frequency domain. Here $D$ is used to represent generalized dyadic downsampling in $n$ -dimensions.	44
2.14.1	The Mallat pyramid structure for 2-D separable wavelet decompositions. $A$ is the approximation of the original signal, $V$ contains vertical edges, $H$ contains horizontal edges and $D$ contains diagonal edges. All subspaces contain $\frac{N}{4}$ samples if $X$ contains $N$	45
3.1.1	Example quincunx lattice definition. The two basis vectors are $v_1$ and $v_2$ . The dots indicate the sampling points of (the first coset of) this lattice. Any lattice point is an integer linear combination of $v_1$ and $v_2$ , ie. $x \in \Lambda$ if $x = a_1 \cdot v_1 + a_2 \cdot v_2$ where $a_1, a_2 \in \mathbb{Z}$ .	49
3.2.1	The two cosets of the quincunx sublattice on the Cartesian lattice. There are no shared points between the cosets of a lattice and the sum of all cosets is the original lattice.	50
3.2.2	Sublattice described in example 3.2.3. There are three cosets and hence the subsampling matrix has a determinant of magnitude 3.	51
3.3.1	Two alternative unit cells for the resampling lattice of figure 3.2.2. The unit cell includes exactly one point from each coset of the sampling lattice. It has $volume =  det(D) $ in the original lattice and $volume = 1$ in the sublattice.	52
3.3.2	(a) The unit cell of the quincunx sublattice on the Cartesian lattice with its defining vectors. (b) The Voronoi cell of the same lattice.	53
3.4.1	The Voronoi cell $\mathcal{U}_c$ for the integer Cartesian lattice. This defines a periodic frequency response which spans the region $-\omega_s/2$ to $\omega_s/2$ . Each of the sampling points shown represents a point of repeat spectra ie. divisions by $\omega_s$ . The Voronoi cells tile to cover the frequency domain.	55

3.4.2	The Voronoi cells of integer sampling and Quincunx sampling shown on the same scale. The sampling points represent repeat spectra spacing in terms of integer sampling. The quincunx resampling has diagonal cutoff with respect to the previous sampling.	55
3.5.1	The cosets of (a) the Quincunx lattice and (b) Separable sampling by two in the horizontal direction. As can be seen, both sampling schemes have two cosets – one shown in white and one grey.	58
3.5.2	Diagram showing the spatial-frequency cut-off of Quincunx resampling. The retained spectrum is shaded grey. The full range of vertical and horizontal frequencies is retained, thus preserving vertical and horizontal edges. The most truncated frequencies are in the diagonal direction which has the lowest psychovisual importance to the human observer.	59
3.5.3	Frequency domain tiling of the quincunx lattice. Dark grey indicates the low-pass regions and light grey is the high-pass regions. The solid box shows the frequency range of the original sampling mechanism. The dashed boxes indicate the continuation of the high-pass region outside the original sampling region due to repeat spectra.	59
3.5.4	How vertical and diagonal edges of minimum spacing (maximum frequency) are retained in the quincunx lattice. The vertical edges are retained with the spacing between the edges intact, while the diagonal edges, corresponding to the cutoff frequency are blurred into a single feature and the high-frequency detail is completely lost.	60
3.6.1	Downsample and upsample process on the quincunx lattice showing intermediate stages. The letters represent particular signal elements and show how their location is altered by the resampling process.	61
4.1.1	General two-channel filterbank used in wavelet transforms with nonseparable sampling. The sampling within the filter framework leads to a shift variant system.	63
4.1.2	Two alternative ways to filter and downsample a signal. The upper version is the familiar filter-downsample method. The lower method is a polyphase version of the same process.	64
4.1.3	Upsampled representation of a filterbank in the quincunx domain. This is a simple extension of the forward version shown in Figure 4.1.2.	65

4.1.4	Impulse response at various points in a polyphase representation. There are two cosets and the impulses inside the structure show the retained signal after retaining only one coset in each case. The lower branch contains a shift operation which changes the coset at the origin.	65
4.1.5	General n-channel upsampled polyphase filterbank. This is a generalization of the case shown in Figure 4.1.3 so that it has n channels. The shift operators are used to change cosets.	66
4.2.1	Configuration of simple decomposition and reconstruction using normal (not-polyphase) filters.	68
5.1.1	Quincunx subband decomposition block diagram showing two levels of decomposition. The H and G operators include all resampling necessary to complete the step.	78
5.1.2	Image coordinate transform during Wavelet decomposition using the quincunx lattice. Unlike separable resampling, Quincunx resampling does more than scale the signal. The signal is rotated $-45^\circ$ and flipped about the horizontal axis.	78
5.1.3	Quincunx and separable sampling lattices. $d_1$ and $d_2$ are the directional vectors which make up the subsampling lattice. In the separable case $d_1$ and $d_2$ have the same direction as the directional vectors of the original Cartesian lattice. The small arrows indicate the shifts to the missing cosets, ie. the displacement vectors to the cosets not shown.	78
5.3.1	Two levels of pixel overlap with quincunx downsampling, in the upsampled domain. In this case the second resampling step elects to down-sample separably.	81
5.3.2	Normal nonseparable wavelet transform low-pass branch showing two levels of decomposition. The input is filtered then resampled then filtered then resampled.	82
5.3.3	Low-pass branch using upsampled filter operator $\uparrow H$ operator.	82
5.3.4	An arbitrary filter with 8 taps shown in (a) its normal state and (b) its (first) upsampled state.	82
5.3.5	Organization of cosets at different resolution levels within the upsampled structure. The transform progress from left to right. The white squares represent locations which have been used to store details of the higher-resolution levels.	83
5.3.6	Low-pass with upsampled operator and separable downsampling replacing two consecutive quincunx down-sample operations.	83

5.3.7	One level of quincunx decomposition performed in the upsampled domain. The orientation is the same in the downsampled and original signals. Each downsampled signal contains half as many coefficients as the original allowing both components to be stored in the space occupied by the original. The low and high-pass results can be stored in the original image space.	84
5.3.8	Detailed view of Lenna's eye from figure 5.3.7. This clearly shows how all the approximation values exist on a single quincunx coset.	84
6.3.1	Plot of CDF and PDF of wavelet transform coefficients for the Lenna, Goldhill and Barbara images with a 12 level quincunx wavelet transform based on the 5/3 Kovacevic and Vetterli filter set.	93
6.3.2	CDF and PDF plots of wavelet transform coefficients for the Lenna, Goldhill and Barbara images with a 6 level separable wavelet decomposition based on the Daubechies $D_4$ orthogonal filter bank.	94
6.3.3	Comparative plots of the probability densities of wavelet coefficient values for separable and quincunx grouped by image. Scales have been adjusted to enhance exaggerate differences.	95
6.3.4	Results for uniform quantization of separable wavelet decomposition of Lenna image using 6 levels of Daubechies $D_4$ decomposition. Quantization to 4, 8, 16, and 32 bins is done.	96
6.3.5	Results of uniform quantization of quincunx wavelet decomposition of the Lenna image. The same quantization interval is used for all coefficients in all scales.	97
6.3.6	Uniform scalar quantization of the Goldhill image with 6 levels of $D_4$ decomposition.	98
6.3.7	Uniform quantization of the Goldhill image performed using the same method as in Figure 6.3.5.	99
6.3.8	Scalar quantization of the Barbara image using 6 levels of $D_4$ decomposition.	100
6.3.9	Uniform quantization of the Barbara image performed using the same method as in Figure 6.3.5.	101
7.1.1	HVS spatial-frequency sensitivity curve with approximations to quincunx decomposition levels. The scale is logarithmic in frequency. Level 1 is the highest resolution detail and in this case there are 10 decomposition levels. Clearly levels 6 through	

	8 occupy the most sensitive areas of the curve and should be treated with the most lossless coding. This is a demonstration only and does not accurately reflect actual data.	104
7.1.2	Plot of the spatial frequency sensitivity function of the HVS given in Equation 7.1.1. Note the peak in relative sensitivity at around 3 c / deg (2.9845).	105
7.1.3	Function curve showing the change in visual contrast sensitivity near a significant stimulus. The dashed line represents the position of an abrupt change in image intensity which is the significant stimulus.	105
7.2.1	To find the relative sensitivity of the Human Visual System to a wavelet subband we need to find the mean of the sensitivity curve over the subband.	107
7.3.1	HVS tuned scalar quantization of 6 level $D_4$ separable decomposition of image Lenna.	111
7.3.2	HVS tuned scalar quantization of 6 level $D_4$ separable decomposition of image Goldhill.	112
7.3.3	HVS tuned scalar quantization of 6 level $D_4$ separable decomposition of image Barbara	113
7.3.4	Lenna image coded with HVS weighted scalar quantized wavelet transform data using 12 levels of quincunx decomposition with KV5/3 filters.	114
7.3.5	Goldhill image coded with HVS weighted scalar quantized wavelet transform data using 12 levels of quincunx decomposition with KV5/3 filters.	115
7.3.6	Barbara image coded with HVS weighted scalar quantized wavelet transform data using 12 levels of quincunx decomposition with KV5/3 filters.	116
7.3.7	Comparison of HVS weighted and uniform quantized images. Both images have very similar perceived quality while having disparate PSNR values.	117
7.3.8	Lenna image reconstructed from the lowest resolution details of a 6-level separable wavelet decomposition using Daubechies $D_4$ filters.	118
7.3.9	Lenna image reconstructed from a 12 level quincunx decomposition using the Kovačević and Vetterli 5/3 biorthogonal filter pair [101], [56].	119
7.3.10	The effects of selectively removing some of the of the three orientations of detail signals in a 6 level separable decomposition	



	of Lenna using $D_4$ wavelets. In each case details were removed from every level of the decomposition or not removed at all.	120
7.5.1	Results of using the Kovačević and Vetterli $D_4$ wavelet for quincunx with uniform scalar quantization on Lenna image.	125
7.5.2	Results of using the Kovačević and Vetterli $D_4$ wavelet for quincunx with uniform scalar quantization on Goldhill image.	125
7.5.3	Results of using the Kovačević and Vetterli $D_4$ wavelet for quincunx with uniform scalar quantization on Barbara image.	126
7.5.4	Lenna, Goldhill and Barbara decomposed and reconstructed using a modified KV5/3 filterbank with detail signals removed in the same fashion as 7.3.9. The modified filterbank is designed to reduce the central tap magnitude with respect to the neighbouring taps.	127
8.1.1	A tree of separable decomposition coefficients. Coefficients at coarse levels occupy a greater area in the image than those at finer levels. Several fine coefficients can reside within the same spatial region spanned by one coarse level coefficient.	129
8.1.2	A binary tree as present in the Quincunx zero-tree representation. This shows a 4 level tree spanning a $(2\sqrt{2} \times 2\sqrt{2})$ coefficient area at the finest level.	129
8.1.3	A quad-tree as present in the separable zero-tree representation. A 3 level tree is shown spanning a $4 \times 4$ coefficient area at the finest level	130
8.1.4	Comparison of binary and quad-trees covering the same area of coefficients at the finest resolution level. Both cover a $4 \times 4$ coefficient area.	130
8.1.5	(a) Original Lenna image. (b) Reconstructed image from the information in one pass of a zero-tree coder, using a 12 level quincunx decomposition and the Kovacevic and Vetterli 5/3 filter set. Original size 262144 bytes. First-pass zero-tree 607 bytes. Rate 0.0023 bpp.	131
8.1.6	The reducing level of the coding thresholds in the zero-tree coding scheme. As the threshold drops the uncertainty about the value of a coefficient becomes less. Each halving of the threshold adds one bit of precision to the information known about the coefficient value.	132
8.2.1	Plot of the total number of symbols coded with each zero-tree coding pass. The plot is not cumulative.	135

8.2.2	Cumulative plot of the total number of symbols coded for the first 5 passes of zero-tree coding. This is a cumulative plot of the data in 8.2.1.	136
8.2.3	Projection of the comparative coding costs for a bit-plane coder versus zero-tree coding.	136
8.2.4	Projection of the use of zero-tree hybrid coder where the naive coder replaces the zero-tree coder when the rate of data for the zero-tree coder exceeds that of the naive coder.	137
8.2.5	Orientations of successive subbands in a quincunx decomposition. Subspaces are of a different orientation than their immediate parents and children.	140
10.1.1	Normal zero-tree coding assumes the same threshold should be applied to all coefficient regardless of location or resolution. The masking shadow adjustment modifies the threshold associated with coefficients affected by contrast masking.	149
10.1.2	Flow-diagrams for the normal zero-tree algorithm and the masking adjustment zero-tree algorithm. The algorithm is applicable to coding and decoding processes. For the masking shadow version, the modification of significance threshold is done individually for each coefficient in the decomposition.	150
10.3.1	Lenna image (a) reconstructed from 3 complete passes of a zero-tree codec with no masking prediction (10kB at 0.30 bpp) and (b) with 4-neighbour masking prediction at the same bit-rate.	152
10.3.2	Goldhill image (a) reconstructed from 3 complete passes of the zero-tree codec with no masking prediction (19 kB at 0.59 bpp) (b) with 4 neighbourhood masking prediction at the same bit-rate.	153
10.3.3	Barbara image (a) reconstructed from 3 complete passes of the zero-tree codec with no masking prediction (28 kB at 0.89 bpp) (b) with 4 neighbourhood masking prediction at the same bit-rate.	153
10.3.4	Airplane image (a) reconstructed from 3 complete passes of the zero-tree codec with no masking prediction (11.2 kB at 0.34 bpp) (b) with 4 neighbourhood masking prediction at the same bit-rate.	154
10.3.5	Baboon image (a) reconstructed from 3 complete passes of the zero-tree codec with no masking prediction (30 kB at 0.94	

	bpp) (b) with 4 neighbourhood masking prediction at the same bit-rate.	154
10.3.6	Boat image (a) reconstructed from 3 complete passes of the zero-tree codec with no masking prediction (11 kB at 0.35 bpp) (b) with 4 neighbourhood masking prediction at the same bit-rate.	155
10.3.7	Fruits image (a) reconstructed from 3 complete passes of the zero-tree codec with no masking prediction (6.5 kB at 0.20 bpp) (b) with 4 neighbourhood masking prediction at the same bit-rate.	156
10.3.8	Girl image (a) reconstructed from 3 complete passes of the zero-tree codec with no masking prediction (7.0 kB at 0.22 bpp) (b) with 4 neighbourhood masking prediction at the same bit-rate.	156
10.3.9	Peppers image (a) reconstructed from 3 complete passes of the zero-tree codec with no masking prediction (10 kB at 0.31 bpp) (b) with 4 neighbourhood masking prediction at the same bit-rate.	157
11.2.1	A near zero-tree. One coefficient is significant (grey), but it can be advantageous to force it to be zero as this would create a complete zero-tree represented by a single symbol in the data stream. In this case 9 symbols are required to code the tree.	166
11.2.2	A separable near-zero-tree with a single non-zero coefficient (grey).	167
11.2.3	Near-zero quad-tree equivalent of Figure 11.2.2.	167
11.2.4	Binary zero-tree equivalent of Figure 11.2.1.	168

## Reproduced Publications

[4] Andrews and Nguyen. Separable and Quincunx Wavelet Image Coding, In *Proc. 6<sup>th</sup> IEEE intl. Workshop on Intelligent Signal Processing and Communications Systems (ISPACS)*, November 1998 (Melbourne, Australia).

# SEPARABLE AND QUINCUNX WAVELET IMAGE CODING

R. Andrews

D.T. Nguyen

Department of Electrical Engineering and Computer Science  
University of Tasmania, Churchill Ave. Sandy Bay 7005  
AUSTRALIA

Fax : +61 3 6226 2136

Richard.Andrews@utas.edu.au D.T.Nguyen@utas.edu.au

**Abstract** – It has been demonstrated that wavelets compete well against DCT based image compression techniques [1]. However the advantages of nonseparable wavelet transforms for image and video coding have not yet been adequately explored.

In this paper we discuss nonseparable wavelet transforms on the quincunx lattice and show that they have certain properties which make them an attractive choice for image compression.

We compare images compressed using separable and nonseparable biorthogonal filters, where the two-dimensional filters are obtained from the one-dimensional filters [2] thus preserving many of the properties from one-dimension to two-dimensions.

We show that the performance of separable and nonseparable techniques is comparable under similar constraints even though the nonseparable filters have support areas less than the separable filters.

## Introduction

Wavelets have become a field of intense research in the image processing field due to their close relationship with the well understood field of subband coding. However the research to date has been almost exclusively concerned with separable wavelet transforms in the form of the Mallat algorithm [5]. There are, however, restrictions which occur when applying separable transforms to images. Some of these restrictions can be lifted by moving to truly two-dimensional wavelet transforms.

## 1 Non-separable Wavelets

Although separable wavelet transforms have a simple and well understood algorithm there are certain advantages to a non-separable transform. For example, images are two-dimensional and thus can be better handled by a truly two-dimensional transform which considers the image as areas rather than as rows and columns. The separable wavelet decomposition has vertical and horizontal cut-off while the non-separable decomposition has a diagonal cut-off. This is better psychovisually since the perceptually least valuable component of vision is quantized first. Non-separable filter-banks have a greater flexibility than separable filters and can be more appropriately tailored for particular purposes.

The application of a non-separable wavelet decomposition is similar to the separable case. Subsampling is not performed by retaining every second column and row, as it is in the separable case, but rather samples on a lattice [6] are retained. This paper focusses on wavelets subsampled on the quincunx lattice; the simplest non-separable lattice.

Subsampling on the quincunx lattice differs from separable sampling in the rate of data reduction. In the separable case four band-pass components are generated and each has one quarter the number of samples of the previous signal (half as many samples in each dimension). In the quincunx case, each component has half as many samples, a factor of  $\frac{1}{\sqrt{2}}$  in each dimension. Figure 2 shows one level of decomposition with the quincunx lattice. Note the rotation of the image which occurs due to a change

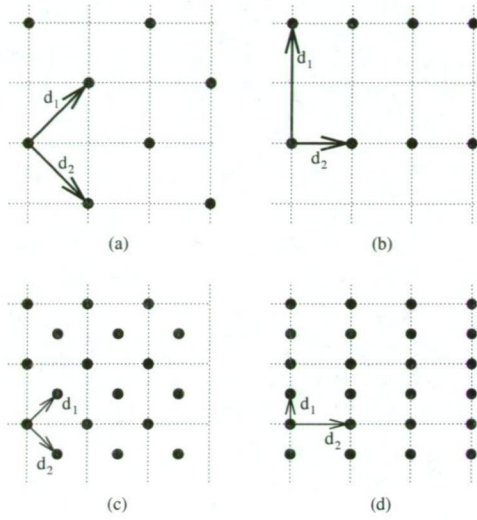


Figure 1: Downsampling lattices. (a) Quincunx, (b) vertical downsampling by 2, (c) reciprocal of (a), (d) reciprocal of (b).

of basis. This effect is reversed for each level of decomposition. Hence at odd depths the orientation is rotated but at even depths the orientation is normal.

A lattice is defined by a number of vectors  $d_0, d_1 \dots$ . For the quincunx lattice the two defining vectors used were  $d_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$  and  $d_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ . This however is usually abbreviated to a defining matrix

$$D_q = (d_0 \ d_1) = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (1.1)$$

Figure 1 shows the quincunx lattice and the lattice for downsampling by two in the vertical direction.

Downsampling can be characterized by

$$Y(\underline{n}) = X(D\underline{n}) \quad (1.2)$$

where  $Y(\underline{n})$  is the subsampled version of the signal  $X(\underline{n})$ , and  $D$  is the matrix characterizing the lattice. In the quincunx case this reduces to

$$Y(n_1, n_2) = X(n_1 + n_2, n_1 - n_2) \quad (1.3)$$

There are several issues which set this form of down-sampling apart from separable down-sampling. One of the most significant is the change

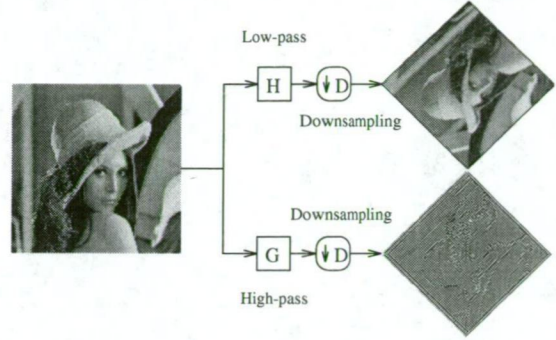


Figure 2: Quincunx downsampling of an image.

in basis which arises as shown in Figure 2. The effect is reversed at each levels and two applications is equivalent to separable down-sampling by two in both dimensions. If  $D_q$  is the quincunx down-sampling matrix then

$$D_q \cdot D_q = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (1.4)$$

$$= \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \quad (1.5)$$

If we consider separable downsampling

$$D_{horiz} = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \text{ and } D_{vert} = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$$

and the application of one followed by the other is

$$D_{sep} = D_{vert} \cdot D_{horiz} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \quad (1.6)$$

as above in (1.5).

One of the attractive features of the quincunx downsampling process is the retained spectrum after downsampling. When downsampling an image through separable means we must necessarily remove half the samples in each dimension as mentioned above. By halving the number of samples in a particular dimension we halve the bandwidth of the image in that direction. This is because each sample now effectively occupies twice as much space in the direction of downsampling. Now because we downsample on the cartesian grid (ie. horizontal and vertical coordinates) we halve the bandwidth

of the image in both horizontal and vertical directions. The problem associated with this scheme is related to the sensitivity of the human visual system (HVS). Our eyes are naturally more sensitive to vertical and horizontal edges than they are to diagonal edges. Separable downsampling thus removes perceptually valuable information. In contrast, the quincunx downsampling process has a diagonal cut-off and does not remove horizontal or vertical frequencies and so the resulting low-pass image will contain this data.

The frequency sensitivity function of the HVS has been known for some decades now. It has a peak sensitivity in the area of 3.5 cycles/degree and a general model can be found in [7]. Because the frequency characteristic of a separable wavelet transform is to decompose the signal into octave sub-bands, when attempting to apply the frequency sensitivity function for coding of wavelet transform coefficients, we must consider the sensitivity function in terms of octaves and determine a sensitivity value to apply to the entire octave. This reduces the resolution of the function to an approximation with just a few terms.

When applying quincunx downsampling the decimation process is more gradual in the frequency domain. This allows us to model the HVS frequency sensitivity function more closely. Now instead of having one term per octave we have two terms and thus a better approximation.

Although the frequency characteristics of the quincunx lattice appear very desirable at first there are also some considerations worth bearing in mind. Chiefly to achieve the same level of decimation from both the separable and quincunx methods we must apply the quincunx scheme twice as many times as the separable scheme. This is because the separable scheme retains only half as many samples as the quincunx scheme. Now, the effect of applying quincunx downsampling twice is equivalent to separable downsampling. This means that the retained spectra of both schemes are similar. However the filters used are different thus resulting in some possible variations in response.

An advantage which the separable scheme has over the quincunx scheme is that the band-pass signals which result have three distinct orientations: horizontal, vertical and diagonal. This allows the different orientations to be coded differently to match the sensitivity of the HVS to different ori-

entations. This process can also be used with the quincunx lattice but we do not have as fine control. In this case we must treat horizontal and vertical features together as they are retained in the same detail signal. Diagonal details are retained in the detail signal from the next (or previous) level of decomposition. This is not a disadvantage though, because it has been shown that the sensitivity of the HVS to errors in quantization of horizontal and vertical details is very similar [3] [4].

## 2 Wavelets Decomposition on the Quincunx Lattice

Wavelet decomposition has two basic steps: filtering and downsampling. The input image is filtered by a low-pass and a band-pass filter and both resulting signals are down-sampled. The filtering process is accomplished by two-dimensional convolution of the signal with the filter coefficients. In the separable case this convolution is done one dimension at a time. Thus the image is considered as a number of separate, unconnected signals; either rows or columns. Non-separable filtering convolves both dimensions simultaneously so the image is considered as areas.

Although there is a large number of wavelet filter sets available we are interested primarily in the differences brought about by the non-separable downsampling and convolution processes so we can make comparisons. We must also constrain ourselves to filters for which there are closely related one- and two-dimensional versions.

We concentrate on two wavelet filter sets here. We use a linear-phase (ie. symmetrical) bi-orthogonal set as proposed by Kovačević and Vetterli in [2] and its one-dimensional version (from which the two-dimensional filter is derived).

The one dimensional filter set derived from the generalized filter using  $a_1 = 2$  and  $a_2 = -6$  is:

$$h_0 = \begin{matrix} 1 & 2 & 1 \end{matrix} \quad (2.1)$$

$$g_0 = \begin{matrix} 1 & 2 & -6 & 2 & 1 \end{matrix} \quad (2.2)$$

The two dimensional filter set derived from the



above one-dimensional filter set is:

$$h_0 = \begin{matrix} & & 1 \\ & 1 & 4 & 1 \\ & & 1 \end{matrix} \quad (2.3)$$

$$g_0 = \begin{matrix} & & & & 1 \\ & & & 2 & 4 & 2 \\ & 1 & 4 & -28 & 4 & 1 \\ & & 2 & 4 & 2 \\ & & & & 1 \end{matrix} \quad (2.4)$$

Note that the bi-orthogonal filter set must obey the following conditions for alias cancellation:

$$\begin{aligned} h_1(z_1, z_2) &= g_0(-z_1, -z_2) \\ g_1(z_1, z_2) &= -h_0(-z_1, -z_2) \end{aligned}$$

where  $f(-z)$  is the Quadrature Mirror Filter (QMF) pair for  $f(z)$  and  $g_1$  and  $h_1$  are synthesis filters.

When we downsample the signal on the quincunx lattice we are retaining only every second sample. Thus the number of samples is reduced by a factor of two. Figure 1 shows the arrangement of the quincunx lattice and (1.1) gives one possible value for  $D_q$ , the downsampling matrix. However there are other possibilities, for example:

$$D_{q2} = \begin{pmatrix} 1 & 2 \\ 1 & 0 \end{pmatrix}, D_{q3} = \begin{pmatrix} 0 & 1 \\ 2 & -1 \end{pmatrix}$$

Although there are many possibilities for the downsampling matrix  $D_q$  we use  $D_q = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$  since this leads to separable downsampling at every other step in decomposition. This allows us to more readily make comparisons between separable and nonseparable methods.

From (1.3) we have  $Y(\underline{n}) = X(D\underline{n})$  for the downsampling process. The reverse is accomplished with the reciprocal sampling matrix

$$\begin{aligned} D_q^{-1} &= \frac{1}{2}D_q \\ &= \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{pmatrix} \end{aligned} \quad (2.5)$$

Figure 1 shows the reciprocal lattice generated by the inverse matrix  $D_q^{-1}$ . Note that the lattice points are now closer together than the cartesian grid points. The upsampling process inserts zeros at those lattice points which lie between pixels.

### 3 Coding and Results

Due to the difference in downsampling factor we apply two quincunx decomposition levels for every separable decomposition level. Thus the resulting approximations for both methods are the same size. Only the band-pass spaces are processed for compression. Due to the different frequency and orientation specific information contained within the various sub-bands, they are all encoded with the same algorithm. The hardest difference to account for is the different filter structures of separable and non-separable cases. We try to account for this by using two-dimensional filters obtained from one-dimensional filters.

We have - by the need for comparison - kept the coding algorithms as simple and 'naive' as possible. Coding is accomplished by two means, either hard thresholding of coefficients or uniform quantization of coefficients.

In general it was found that the separable method performed better than the non-separable method by a small degree. The results from the separable method were in general smoother than those from the non-separable method however the non-separable method tended to retain fine details better.

Figure 3 shows the effects of quantization, which has the more destructive effect on the quincunx scheme. The errors arising from this quincunx method tend to form patches of distortion whereas the errors from the conventional method tend to occur as ringing and blurring along one dimension. Figure 4 illustrates the effects of thresholding. In this case the quincunx method performs much better and is comparable, and in some cases better than the separable method. Figures 5 shows the situation when the same level of decomposition is applied to separable and quincunx methods. As can be seen, in this case the new quincunx method is far superior, but this situation is not fair for comparison purposes because the approximation signal for the quincunx case has many more pixels than the separable case.

The quincunx method showed errors in areas of low activity thus creating very noticeable errors in the image, whereas the separable method was found to blur or cause ringing in the image resulting in less noticeable errors but loss of fine detail.

It should be noted that the area of support of the



Figure 3: Reconstruction from (left) 6 non-separable levels and (right) 3 separable levels, with 8 quantization bins.



Figure 4: Reconstruction from (left) 6 non-separable levels and (right) 3 separable levels, with 90% of coefficients set to zero.

non-separable filters is not as large as that of the separable filters. For instance the 5x5 quincunx filter as shown in (2.4) has 12 coefficients which are zero. Thus the actual size of the filter is only 13 coefficients (ie. less than a 4x4 filter). Similarly the 3x3 filter only has 5 coefficient which are non-zero. The reduced support size adversely affects regularity and thus also reduces immunity to quantization and thresholding noise.

## References

- [1] Michael L. Hilton, Björn D. Jawerth and Ayan Sengupta. "Compressing Still and Moving Images with Wavelets." *Multimedia Systems*, Vol. 2, No. 3
- [2] Jelena Kovačević and Martin Vetterli "Nonseparable Multidimensional Perfect Reconstruction Filter Banks and Wavelet Bases for  $\mathcal{R}^n$ .", *IEEE Trans. Info. Theory*. Vol. 38, No. 2 March 1992



Figure 5: Reconstruction from (left) 3 non-separable levels and (right) 3 separable levels, with 90% of coefficients set to zero.

- [3] Watson, Yang, Solomon and Villasenor "Visual Thresholds for Wavelet Quantization Error." *SPIE Proceedings Vol 2657. Human Vision and Electronic Imaging*. 1996
- [4] Watson, Yang, Solomon and Villasenor "Visibility of Wavelet Quantization Noise." *IEEE Trans. Image Proc.* Vol. 6, No. 8, Nov 1997
- [5] Stephane G. Mallat "A Theory of Multiresolution Signal Decomposition: The Wavelet Representation." *IEEE Trans. Pattern Analysis and Machine Intelligence*. Vol. 11, No. 7, July 1989.
- [6] Dan E. Dudgeon and Russell M. Mersereau *Multidimensional Signal Processing*. Prentice Hall 1984.
- [7] King N. Ngan, Kin S. Leong and H. Singh "Adaptive Cosine Transform Coding of Images in Perceptual Domain." *IEEE Trans. ASSP* Vol. 37, No. 11, Nov 1989.

[5] Andrews and Nguyen. Separable versus Quincunx Wavelet Transforms for Image Compression. *In Proc. 6<sup>th</sup> IEEE intl. Conf. on Communications Systems* November 1998 (Singapore).

# SEPARABLE VERSUS QUINCUNX WAVELET TRANSFORMS FOR IMAGE COMPRESSION

R. Andrews

D.T. Nguyen

Department of Electrical Engineering and Computer Science  
University of Tasmania  
Churchill Ave. Sandy Bay 7005  
AUSTRALIA

**Abstract** – It has been demonstrated that wavelets compete well against DCT based image compression techniques [1]. However the advantages of nonseparable wavelet transforms for image and video coding have not yet been adequately explored.

In this paper we discuss nonseparable wavelet transforms on the quincunx lattice and compare results under simple data reduction methods using separable and nonseparable symmetrical (linear-phase) biorthogonal filters, where the two-dimensional filters for use on the quincunx lattice are obtained from the one-dimensional filters [3].

We attempt to use the most general coding schemes possible which do not disadvantage either separable or nonseparable techniques. For a comparison of separable wavelets, quantizers and coders the reader is directed to [2]. We attempt to use separable and quincunx wavelet filters of similar size and with similar properties. Even so it is difficult to determine the most appropriate filters for comparison.

We show that the performance of separable and nonseparable techniques is comparable under similar constraints.

Finally we investigate methods for performance enhancement through the use of psychovisual coding with the nonseparable transform.

## Introduction

Wavelets have become a field of intense research in the image processing field due to their close relationship with the well understood field of subband coding. However the research to date has been almost exclusively concerned with separable wavelet transforms, the natural extension into two dimensions of one-dimensional wavelet transforms - which are well understood and well documented.

There are, however, restrictions which occur when applying separable one dimensional wavelet transforms to images - which are two dimensional signals. Some of these restrictions can be lifted by moving from separable one dimensional wavelet transforms to truly two-dimensional wavelet transforms.

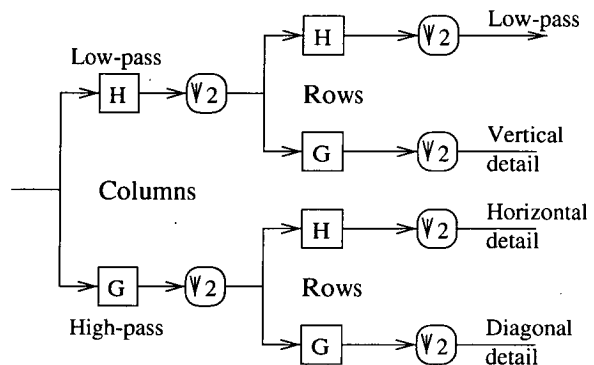


Figure 1: One level of decomposition by the separable Mallat Pyramid algorithm. Note the orientation specific outputs.

## 1 Wavelet Transforms

### 1.1 2D Separable Wavelets (Tensor Product)

The separable wavelet decomposition is known as the Mallat Pyramid scheme [4]. Figure 1 shows one level of decomposition. Filtering and subsampling is applied separably resulting in an approximation signal ( $low_{horiz}, low_{vert}$ ) and three detail spaces with distinct orientations of horizontal, vertical and diagonal.

Subsampling is done by a factor of two in each dimension, thus the resulting approximation and details each contain one quarter the number of samples as the previous signal.

### 1.2 2D Non-separable Wavelets

Although separable wavelet transforms have a simple and well understood algorithm there are certain advantages to a non-separable decomposition:

- Images are two-dimensional and thus can be better handled by a truly two-dimensional transform which considers the image as areas rather than as rows and columns.



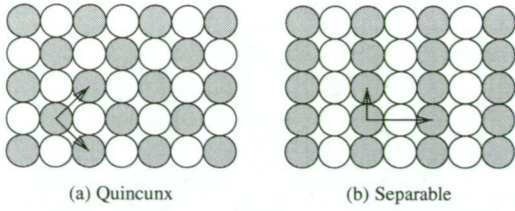


Figure 2: The two cosets of (a) the quincunx lattice, (b) horizontal subsampling by two.

- Separable wavelet decompositions have vertical and horizontal cut-off while the non-separable decomposition can have a cut-off at an angle. This is better psychovisually because it means that the perceptually least valuable component of vision is quantized first.
- Non-separable filter-banks have greater flexibility and can be more appropriately tailored for particular purposes. In particular orthogonal linear phase solutions are not possible with separable filters but can be created with non-separable filters.

The application of a non-separable wavelet decomposition is somewhat similar to the one-dimensional case. The low-pass component is repeatedly filtered and subsampled resulting in another low-pass and another detail signal.

However in this case samples on a lattice [5] are retained. This paper focusses on wavelets subsampled on the quincunx lattice, which is the simplest non-separable lattice.

In quincunx downsampling, each subsequent low-pass image has half as many samples as its parent, a factor of  $\frac{1}{\sqrt{2}}$  in each dimension. Figure 3 shows one level of decomposition with the quincunx lattice. The rotation of the image occurs due to a change of basis. This effect is reversed for each level of decomposition. Hence at odd depths (ie. after 1, 3, 5 etc decomposition levels) the orientation is rotated but at even depths the orientation is normal.

### 1.2.1 The Quincunx Lattice

A lattice is a discrete subset of the points in the original signal which may be used. A common lattice is that generated by interlacing in television pictures. In this case every second row of the picture is retained and displayed. This is called a coset of the image. In the next field the other coset is displayed.

In this paper we consider the quincunx lattice structure which is the simplest non-separable subsampling method. Figure 2 shows the two cosets from the quincunx lattice and from separable subsampling in the horizontal direction. Grey shows the pixels contained in one coset while white shows those contained in the other. Note that we can change from one coset to the other by shifting by one pixel. In the quincunx case this can be

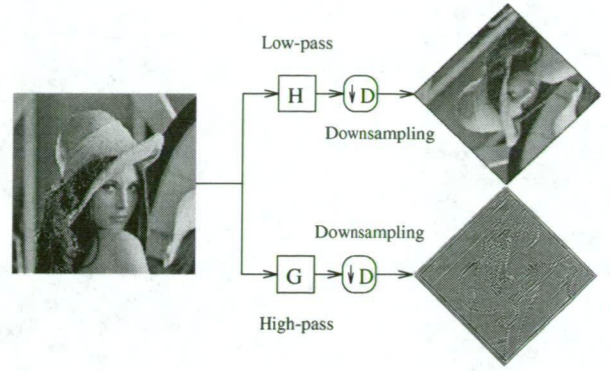


Figure 3: Subsampling on the quincunx lattice.

horizontally or vertically however in the separable case this must be in the direction of subsampling (horizontally in this case).

A lattice is defined by a number of vectors  $d_0, d_1 \dots d_n$ . For the quincunx lattice the two defining vectors used are

$$d_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \text{ and } d_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}.$$

The defining matrix is  $D = (d_0 \ d_1 \ \dots \ d_n)$  which in this case gives us

$$D_q = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (1.1)$$

Hexagonal down-sampling is another method which is frequently used in image processing. That generates four cosets [6, 5].

Subsampling can be characterized by the following equation

$$Y(\underline{n}) = X(D\underline{n}) \quad (1.2)$$

where  $Y(\underline{n})$  is the subsampled version of the signal  $X(\underline{n})$ , and  $D$  is the matrix characterizing the lattice.

In the quincunx case this reduces to

$$Y(n_1, n_2) = X(n_1 + n_2, n_1 - n_2) \quad (1.3)$$

There are several issues which set this form of down-sampling apart from separable subsampling. One of the most significant is the change in basis which arises. Figure 3 shows the effect of subsampling on the quincunx lattice. The subsampling process rotates the image by  $\frac{\pi}{4}$  and flips the image about the horizontal axis. Two applications is equivalent to separable down-sampling by 2 in both dimensions, ie. the image is returned to its original orientation and it is smaller by a factor of 2 in each dimension.

### 1.3 Frequency Band Considerations

One of the attractive features of the quincunx subsampling process is the retained spectrum after subsampling. When subsampling an image through separable

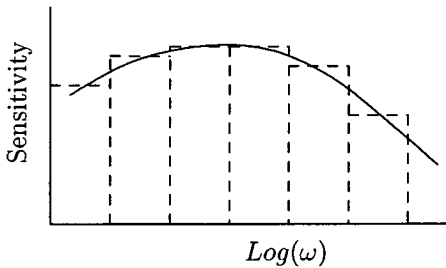


Figure 4: Frequency sensitivity of the human visual system with octave sub-band approximations shown.

able means we must remove half the samples in each dimension. By halving the number of samples in a particular dimension we are effectively halving the bandwidth of the image in that direction. This is because each pixel now occupies twice as much space in the direction of subsampling. So we halve the bandwidth of the image in both horizontal and vertical directions. The problem associated with this scheme is related to the sensitivity of the human visual system (HVS). Our eyes are naturally more sensitive to vertical and horizontal edges than they are to diagonal edges. When subsampling separably we remove the important information of horizontal and vertical features. In contrast, the quincunx subsampling process has a diagonal cut-off and does not remove horizontal or vertical features [7].

The frequency sensitivity function of the HVS has been known for some decades now. It has a peak sensitivity in the area of 3.5 cycles/degree and a general shape as shown in Figure 4 [8, 9, 12]. Because the frequency characteristic of a separable wavelet transform is to decompose the signal into octave sub-bands, when attempting to apply the frequency sensitivity function for coding of wavelet transform coefficients, we must consider the sensitivity function in terms of octaves and determine a sensitivity value to apply to the entire octave. This reduces the resolution of the function to just a few terms.

When applying quincunx subsampling the decimation process is more gradual in the frequency domain. This allows us to model the HVS frequency sensitivity function more closely. Now instead of having one term per octave we have two terms, thus greater accuracy.

Although the frequency characteristics of the quincunx lattice appear very desirable there are also some disadvantages to observe. Chiefly to achieve the same level of decimation from both the separable and quincunx methods we must apply the quincunx scheme twice as often as the separable scheme. This is because the quincunx scheme retains twice as many samples as the separable scheme. Now the effect of applying quincunx subsampling twice is equivalent to separable subsampling. This means that the retained spectral regions of both schemes are similar. However the filters used are different thus resulting in some possible variations in response.

In the separable scheme the high-pass signals which

result have three distinct orientations: horizontal, vertical and diagonal. This allows the different orientations to be coded differently to match the sensitivity of the HVS to different orientations. This process can also be used with the quincunx lattice but we do not have such fine control. In this case we must treat horizontal and vertical features together because they are retained in the same detail signal. Diagonal details are retained in the detail signal from the next level of decomposition. It has been shown that the sensitivity of the HVS to quantization of horizontal and vertical wavelet coefficients is about the same [9, 10] so this is not a problem.

## 2 Wavelet Decomposition on the Quincunx Lattice

Wavelet decomposition has two basic steps: filtering and subsampling. The input image is filtered by a low-pass and a high-pass filter and both resulting signals are down-sampled. The filtering process is accomplished by two-dimensional convolution of the signal with the filter coefficients. In the separable case this convolution is done in one dimension at a time, thus the image is considered as a number of separate, unconnected signals, either rows or columns. Non-separable filtering convolves both dimensions simultaneously. For this reason the image is considered as areas (under the support of the filter) rather than as rows and columns.

Although the number of wavelet filter sets available is large we are interested primarily in the differences brought about by the non-separable subsampling and convolution processes. We also constrain ourselves to filters for which there are well defined one- and two-dimensional versions.

We concentrate on two wavelet filter sets here. We use a linear-phase (ie. symmetrical) bi-orthogonal set as proposed by Kovačević and Vetterli [3] and its one-dimensional version (from which the two-dimensional filter is derived).

The one dimensional filter set derived from the generalized filter [3] using  $a_1 = 2$  and  $a_2 = -6$  is:

$$h_0 = \begin{matrix} 1 & 2 & 1 \end{matrix} \quad (2.1)$$

$$g_0 = \begin{matrix} 1 & 2 & -6 & 2 & 1 \end{matrix} \quad (2.2)$$

The two dimensional filter set derived from the above one-dimensional filter set is:

$$h_0 = \begin{matrix} & & 1 \\ 1 & 4 & 1 \\ & & 1 \end{matrix} \quad (2.3)$$

$$g_0 = \begin{matrix} & & & & 1 \\ & & 2 & 4 & 2 \\ 1 & 4 & -28 & 4 & 1 \\ & & 2 & 4 & 2 \\ & & & & 1 \end{matrix} \quad (2.4)$$



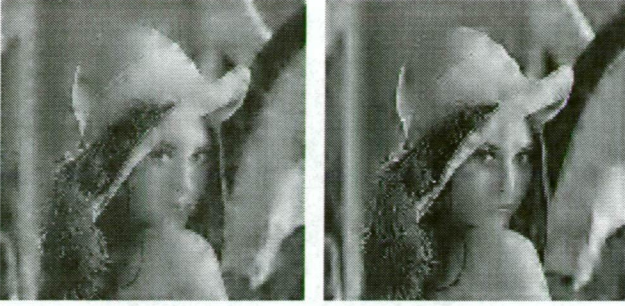


Figure 5: Reconstruction from (left) 6 non-separable levels and (right) 3 separable levels, with 8 quantization bins.



Figure 6: Reconstruction from (left) 6 non-separable levels and (right) 3 separable levels, with 90% of coefficients set to zero.

For alias cancellation we must have [11]:

$$\begin{aligned} g_0(z_1, z_2) &= h_1(-z_1, -z_2) \\ g_1(z_1, z_2) &= -h_0(-z_1, -z_2) \end{aligned}$$

where  $f(-z)$  is the Quadrature Mirror Filter (QMF) counterpart for  $f(z)$ .

### 3 Results

Due to the inherent difference in the algorithms used to decompose images with separable and non-separable processes it becomes very difficult to make fair comparisons between results obtained from these schemes.

Particular issues which cause difficulties in comparison are the decimation factor, retained spectra and filter differences.

Due to the difference in decimation factor we apply two quincunx decomposition levels for every separable decomposition level. This means that our resulting approximation spaces are the same size. All sub-bands are encoded with the same algorithm which ignores scale and orientation. Probably the hardest difference to account for is difference in filter structures for separable and non-separable cases which we have accounted for by using two-dimensional filters obtained from one-dimensional filters.

We have - by the need for comparison - kept the coding algorithms as simple and 'naive' as possible. Coding is accomplished by two means:

#### 1. Hard thresholding of coefficients

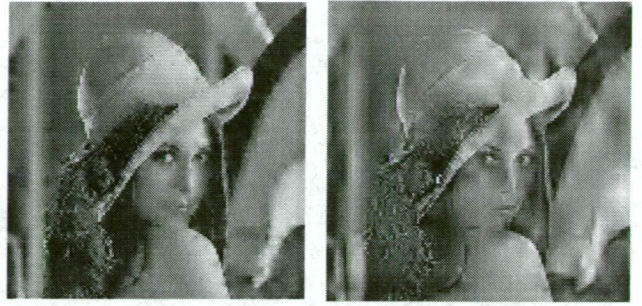


Figure 7: Reconstruction from (left) 4 non-separable levels and (right) 4 separable levels, with 90% of coefficients set to zero.

#### 2. Uniform quantization of coefficients

Hard thresholding is done on a percentage of coefficients basis. The absolute value of coefficients is used and the threshold value  $T$  is the value for which  $p\%$  of coefficients have absolute value less than  $T$ . Thus the same fraction of coefficients is removed (set to zero).

A uniform quantization scheme is used with a variable number of bins. Only results for 8 bins are shown here.

It was found that the artefacts and errors introduced by thresholding and quantization of the wavelet coefficients were of quite different natures but of a similar destructive magnitude for both quincunx and separable schemes.

In general it was found that the separable method performed better than the non-separable method by a small degree. The results from the separable method were in general smoother than those from the non-separable method however the non-separable method retained fine details better.

The images reproduced here are 128x128 pixel versions of the standard test image "lenna".

Figure 5 shows the effects of quantization, which has the more destructive effect on the non-separable scheme. As can be seen, the errors arising from the non-separable method tend to form noticeable patches of distortion whereas the errors from the conventional method tend to occur as ringing and blurring in one dimension.

Figure 6 shows the effects of thresholding. As can be seen, in this case the non-separable method performs much better and is comparable, and in some cases better than the performance of the separable method.

Figure 7 shows the situation when the same level of decomposition is applied to separable and non-separable methods. This situation is not for comparison purposes but to demonstrate the effects of having a larger approximation signal.

The non-separable method was found to corrupt areas of low activity thus creating very noticeable errors in the image, whereas the separable method was found to blur the image resulting in less noticeable errors but loss of fine detail.

A point to note is that the support size of the non-separable filters does not cover their full rectangular grid size. For instance the 5x5 filter as shown in (2.4)





Figure 8: Reconstruction from 6 quincunx levels with HVS tuned quantization.

has 12 coefficients which are zero. Similarly the  $3 \times 3$  filter only has 5 coefficient which are non-zero. This reduced support will adversely affect the regularity of the transform and could also reduce its immunity to quantization and thresholding noise.

## 4 HVS tuning

We present here a couple of images using a coding scheme based on the HVS sensitivity curve [8, 12]. A quantization scheme is used where the step size is adjusted so that levels which contain data to which the human eye is more sensitive are quantized with finer steps.

Figure 8 shows two images compressed to 15:1 and 30:1 respectively using the method. Bit rates are based on entropy of results only so run-length coding would almost certainly result in higher ratios. As can be seen the image quality is significantly better than for the previous cases in which HVS tuned quantization was not used.

To further improve the coding process some form of activity masking should also be taken into account. This method uses the fact that the human eye is less able to distinguish small details in areas of high activity, such as within textures and near significant edges. It is envisioned that this type of adjustment would have a marked advantage for the quincunx method since its most noticeable artefacts are due to errors in areas of low activity.

## References

- [1] Michael L. Hilton, Björn D. Jawerth and Ayan Sengupta. "Compressing Still and Moving Images with Wavelets." *Multimedia Systems*, Vol. 2, No. 3
- [2] Jian Lu, V. Ralph Algazi and Robert R. Estes, Jr. "A Comparative Study of Wavelet Image Coders." Centre for Image Processing and Integrated Computing (CIPIC). University of California, Davis, CA 95616.
- [3] Jelena Kovačević and Martin Vetterli "Nonseparable Multidimensional Perfect Reconstruction Fil-

ter Banks and Wavelet Bases for  $\mathcal{R}^n$ .", *IEEE Trans. Info. Theory*. Vol. 38, No. 2 March 1992

- [4] Stephane G. Mallat "A Theory of Multiresolution Signal Decomposition: The Wavelet Representation." *IEEE Trans. Pattern Analysis and Machine Intelligence*. Vol. 11, No. 7. July 1989.
- [5] Dan E. Dudgeon and Russell M. Mersereau *Multidimensional Signal Processing* Prentice Hall 1984
- [6] Edward H. Adelson, Eero Simoncelli "Orthogonal Pyramid Transforms for Image Coding." *SPIE Vol. 845 Visual Communications and Image Processing II*. 1987.
- [7] Jelena Kovačević and Martin Vetterli "FCO Sampling of Digital Video Using Perfect Reconstruction Filter Banks." *IEEE Trans. Image Processing*. Vol. 2, No. 1, January 1993
- [8] King N. Ngan, Kin S. Leong, H. Singh. "Adaptive Cosine Transform Coding in Perceptual Domain". *IEEE Trans. ASSP* Vol. 37, No. 11, November 1989.
- [9] Andrew B. Watson, Gloria Y Yang, Joshua A. Solomon and John Villasenor "Visual Thresholds for Wavelet Quantization Error." *SPIE Proceedings Vol 2657*, paper #44. Human Vision and Electronic Imaging, B. Rogowitz and J Allebach, Eds., The Society for Imaging Science and Technology. 1996
- [10] Andrew B. Watson, Gloria Y. Yang, Joshua A. Solomon, John Villasenor "Visibility of Wavelet Quantization Noise." *IEEE Trans. Image Processing* Vol. 6, No. 8, August 1997.
- [11] Martin Vetterli, Jelena Kovačević and Didier LeGall "Perfect Reconstruction Filter Banks for HDTV Representation and Coding." *Signal Processing: Image Communication 2* 1990. pp. 349-363.
- [12] Dadang Gunawan, Thong Nguyen "Psychovisual Image Coding Using Wavelet Transform." *Australian Journal of Intelligent Information Processing Systems*, Autumn Issue, Vol. 2, No. 1, March 1995.
- [13] Jerome M. Shapiro "Embedded Image Coding Using Zerotrees of Wavelet Coefficients." *IEEE Trans. Signal Processing*. Vol 41. No. 12. Dec. 1993.



[6] Andrews and Nguyen. Location Equivalence and Wavelet Image Transforms for the Quincunx Lattice. In *Proc. 5<sup>th</sup> IEEE intl. Symp. on Sig. Proc. and its Applications*, August 1999 (Brisbane, Australia).

# MULTISCALE LOCATION EQUIVALENCE AND WAVELET IMAGE TRANSFORMS ON THE QUINCUNX LATTICE

*Richard Andrews*

Electrical Engineering and Computer Science  
University of Tasmania, Hobart, Australia  
<Richard.Andrews@utas.edu.au>

## ABSTRACT

Work on Wavelet based coding of images [1] has relied almost completely on the use of the separable, ie. Tensor Wavelet Transform. This method treats images as one-dimensional rows and columns. Treating images in a truly multidimensional way allows for much greater flexibility in the manipulation of the information. The quincunx lattice is a natural choice for applying non-separable filtering because it is the simplest non-separable lattice. Its diagonal cut-off gives it advantageous psychovisual properties. This paper shows how to determine spatial location equivalence across different levels of a Wavelet decomposition on the quincunx lattice. This allows use of methods that use the continuity of features across scales such as embedded zero-tree coding and quad-tree coding. A novel method of decomposition is outlined which significantly reduces resampling computational complexity.

## 1. INTRODUCTION

Zero-tree [2] and quad-tree [3] coding of Wavelets has provided one of the most successful image coding approaches in recent times[4]. However it has been confined to the separable domain because of its dependence on multiscale inheritance. The most important features to human vision, edges, are multiresolution in nature, they occupy the same spatial location throughout several resolution levels. The utilization of this continuity of features across scales leads to efficient coding strategies.

This paper shows how to determine such multiresolution inheritance on the quincunx lattice. We also introduce a novel implementation for Wavelet decomposition and resampling using the quincunx lattice which can be extended to other sampling lattices. A comprehensive introduction to Wavelets in  $\mathcal{R}^n$  is given in [5] and comparisons between separable and quincunx decompositions are given in [6].

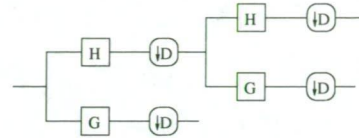


Figure 1: Quincunx subband decomposition block diagram showing two levels of decomposition.

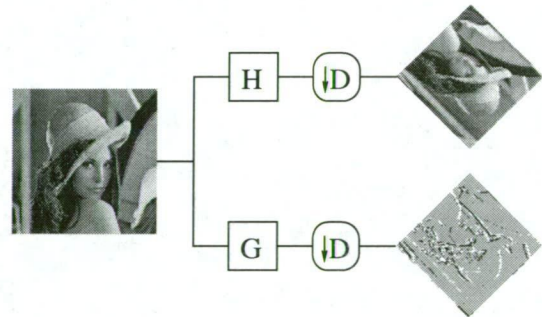


Figure 2: Image coordinate transform during Wavelet decomposition using the quincunx lattice.

## 2. QUINCUNX WAVELET TRANSFORM

The Fast Wavelet Transform is performed through a process of filtering and downsampling. Filtering is done by a low-pass filter  $H$  and a high-pass filter  $G$ . The downsampled high-pass signal contains the wavelet coefficients or details and the low-pass signal is further decomposed until a desired depth of decomposition is achieved.

It is the detail signals which are coded to achieve image compression. Wavelets are very effective for energy compaction and the detail signals are typically quite sparse leading to efficient compression.

Figure 1 shows two levels of Wavelet decomposition using the quincunx lattice and figure 2 shows the coordinate transform which occurs during quincunx sampling. A second transformation will return the image to its normal orientation.

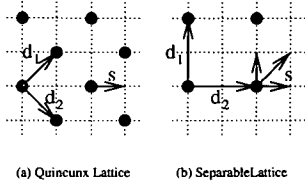


Figure 3: Quincunx and separable sampling lattices.

Whereas separable downsampling removes half the samples from one-dimension at a time, the quincunx lattice cannot be separated into such a scheme. Figure 3 shows the quincunx and separable sampling lattice.

### 3. EQUIVALENCE ACROSS SCALES

The Fast Wavelet Transform is a multiresolution process which decomposes a signal into components of various scales, starting from the finest detail and ending at the coarsest features. Each level of decomposition involves resampling a filtered signal to a lower resolution (decimation). This necessarily involves a transformation of the coordinates of samples, as the same sampling density must be retained. In the separable case this transformation is a trivial division by two in each dimension, but for the quincunx lattice the transform is more complicated. We describe the coordinate transform in terms of a matrix operation :

$$x_{j+n} = D^n x_j \quad (1)$$

where  $x_j$  is the location of the sample at resolution level  $j$  ( $j$  increasing for higher resolutions) and  $D$  is the sampling matrix which defines the lattice. In the case of the quincunx lattice the defining matrix<sup>1</sup> is  $D = \begin{pmatrix} d_1 & d_2 \\ 1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ . It should be noted that  $D^2 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$  which gives separable sampling by 2 in two-dimensions.

We assume that, within the current sampling lattice, a lattice point is the reference for a pixel whose area is the square immediately above and to the right of that lattice point, ie. the value of the sample on the lattice point defines the value over the entire area of the pixel.

To determine the child samples enclosed within a pixel boundary from a lower resolution level (parent pixel) we must first upsample the coordinates of its reference lattice point ie.  $x_{j+1} = Dx_j$ . In the upsampled domain this point will reside on the first coset of the lattice. Within the boundaries of the parent pixel, two pixels are contained at the current resolution level: the pixel referenced by the upsampled location of the parent lattice point (at  $Dx_j$ ), and its immediate neighbour, as determined by the sampling coset shift

vector  $s$ , (at  $Dx_j + s$ ). For  $D$  as defined above  $s = (1, 0)$ . See also figure 3.

When taken to the next resolution level, the problem reduces to separable downsampling. When covering multiple resolution levels this simple method no longer holds because we add extra cosets as we upsample. We can only predict the locations of the first coset without modifying our method (this is complicated and generally not necessary so it is not covered here). Instead we use the inverse result to solve the general problem in this case.

The inverse problem, ie. determining which pixel  $x_0$  in level 0, is the parent of  $x_n$  in level  $n$ , is much simpler. Because we are moving in the down-sampling direction we can simply use

$$x_j = \text{floor}(D^{-n}x_{j+n})$$

where  $\text{floor}(x)$  is the operator which returns the nearest integer value less than or equal to  $x$ .

This allows us to predict the parent pixel  $n$  resolution levels below the current level. It also gives an alternative for finding child pixels of a low resolution parent by performing a search of the high-resolution image for all pixels with the correct parent.

### 4. UPSAMPLED REPRESENTATION

Consider the process of downsampling a signal. Samples from one coset are retained while samples from the unwanted coset(s<sup>2</sup>) are discarded. We don't actually wish to move the pixels from their location within the image but it is apparent that the coordinate transform which occurs during downsampling (required to maintain sampling density) does move the pixels, thus an inverse relationship is necessary to determine their original position. This results in spaces left in the upsampled image where the discarded coset samples were previously located.

Nonseparable downsampling and further decomposition can be performed without this complicated coordinate transform. We concentrate on the quincunx example here. For the downsampling process we simply remove those pixels from the unwanted coset and leave the retained coset untouched. This has the same effect as downsampling and then upsampling again, hence we refer to this as the downsampled signal in the *upsampled* domain. This results in an image of the original size and orientation but where one coset is missing (the discarded coset samples). In this way the spatial relationship between samples at different decomposition levels is easily determined because their location is not altered from that of their parents and children.

Figure 4 shows the two child pixels per parent and the axes of the downsampled domain. We can see that the two

<sup>1</sup>This is not the only possibility for  $D$  but this one leads to separable sampling at every other step and retains spatial location significance.

<sup>2</sup>For quincunx there are only two cosets thus one is retained and one rejected, however separable has four cosets and three are rejected.



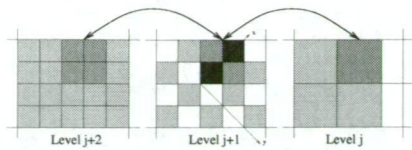


Figure 4: Two levels of pixel overlap with quincunx downsampling, in the upsampled domain.



Figure 5: Low-pass branch in detail.

children of a parent are (a) the child pixel with the same lattice point as the parent; and (b) the child pixel removed from the parent lattice point by the *upsampled* coset shift vector,  $\uparrow s = (1, 1)$ . This is consistent with our previous representation where in the downsampled domain the downsampled shift-vector applies.

#### 4.1. Cross-scale Coding

The spatial relationships between samples at different scales should now be apparent from figure 4. Although quad-tree coding is specifically designed for the four-coset separable decomposition it now becomes possible to design and implement similar coding schemes for non-separable sampling structures. In its present form zero-tree coding is also specific to separable sampling but this is an implementation issue and does not preclude a modified version being developed for other sampling structures using the same principles.

#### 4.2. In-Place Decomposition

Wavelet decomposition is achieved by filtering then downsampling then filtering then downsampling, etc. Thus the filters remain the same (one of the key properties of the Fast Wavelet Transform) but at each stage of decomposition the image has been resampled and thus has undergone a coordinate transformation.

We can describe this situation by

$$X_{low_1}(n) = D^{-1} \cdot H \cdot X(n)$$

where  $X_{low_1}(n)$  is the low-pass version of  $X(n)$  after one level of decomposition (filtering by  $H$  followed by downsampling by  $D$ ).

Examining the *low-pass* branch we can see that the second level low-pass signal is achieved by two sequential filter-subsample combinations as shown in figure 5.

Figure 6 shows an equivalent method of performing the same operations in the upsampled domain as described above. Here  $\uparrow H$  is a filtering operator derived from  $H$

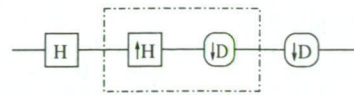


Figure 6: Low-pass branch using upsampled operator.

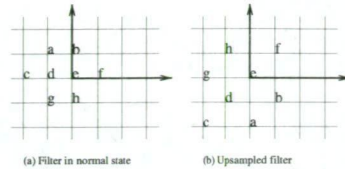


Figure 7: An arbitrary filter shown in (a) its normal state and (b) its first upsampled state.

by a process of upsampling. The  $\uparrow H$  operator performs the exact same filtering operation in the upsampled domain as  $H$  does in the downsampled domain. The resulting filtered sample values only exist on the retained coset.

In the downsampling operation only one coset is retained, thus the samples in the unwanted coset ceases to exist. Normally sampling density must be maintained and the image size shrinks and the sample location coordinates are transformed to account for this. In the upsampled representation we change the sampling density by removing a coset but avoid the need for transforming the coordinates. Because we have violated the sampling density conditions we must now also change the sampling density (ie. coordinate transformation) of the samples in the filter to account for this. This can be thought of as downsampling the signal then upsampling it again, and thus upsampling the filter to compensate.

During implementation we cannot make elements within a data array disappear, instead we mark them as non-existent for consideration in further filtering operations. Thus data locations of samples on the discarded coset are neither used in subsequent filtering operation nor are they changed in any way by future operations. This allows us to use these locations for other purposes such as storing detail signals.

After filtering for the second time (by  $\uparrow H$  in figure 6) the retained coset locations now contains the same samples as after the second filtering operation by  $H$  in figure 5 but they are still in their original positions within the original (ie. upsampled) lattice. Two quincunx downsampling operations are then required to return these samples to their usual downsampled coordinates. Using the quincunx downsampling matrix  $D$  from (1) results in separable downsampling at every second step. Hence two consecutive quincunx downsampling operations results in separable downsampling.

Figure 8 shows the resulting implementation using separable downsampling with quincunx filters. Note that we can continue to further upsample the upsampled operators and





Figure 8: Low-pass with upsampled operator and separable downsampling.

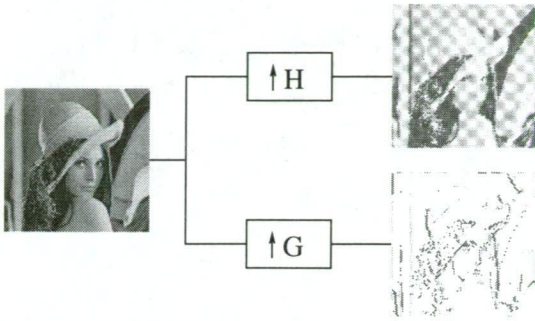


Figure 9: One level of quincunx decomposition performed in the upsampled domain.

keep the signal samples in their original locations, leaving all the downsampling operations until the very end (or not perform them at all). This way all filtering operations are carried out in-place within the original grid.

Figure 9 shows the resulting low-pass and detail spaces from one level of quincunx wavelet transform in the upsampled domain. Note that although the filtered images occupy the same space as the original, each has half the number of samples (critical sampling is preserved).

Figure 10 shows a detail which should clarify the existence of only one coset in the downsampled images.

#### 4.3. Computational Complexity

Quincunx downsampling requires every pixel to be transformed to a new coordinate position at each decomposition level, requiring 6 additions per pixel. Using the upsampled domain we can perform the process at a fraction of the computational cost. With the upsampled method we only transform the coordinates of the filter being used and if we store an upsampled version of the filter this cost can be removed as well. The downsampling process is reduced to the same computational cost as separable sampling. If all computations are carried out in place then all signal resampling cost can be removed, however we will still need to further up-sample the filtering operators for each level of decomposition unless they are precomputed.

### 5. CONCLUSIONS

By using an in-place representation, cross-scale coding methods, which have only been used with separable downsampling, become available to non-separable sampling systems without complicated and computationally expensive

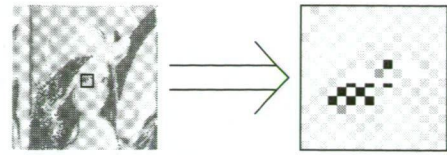


Figure 10: Detailed view of Lenna's eye from figure 9.

coordinate transforms. This opens the possibility of using these highly effective coding methods with the advantages of truly two-dimensional transforms.

The computational cost involved with non-separable sampling can be reduced or removed using upsampled filtering operators and performing all operations in-place. From a research perspective the in-place calculations also aid in the visualization of otherwise obscured multi-scale features.

### 6. FUTURE WORK

Future work in this area includes the development of the many cross-scale coding algorithms which have been developed solely for use with the separable sampling structure.

It is hoped that the ideas presented in this paper will encourage more investigation into the mostly neglected non-separable image coding domain.

### 7. REFERENCES

- [1] Mallat "A Theory of Multiresolution Signal Decomposition: The Wavelet Representation." IEEE Trans. Pattern Analysis and Machine Intelligence. Vol. 11, No. 7, July 1989.
- [2] Shapiro "Embedded Image Coding Using Zerotrees of Wavelet Coefficients." IEEE Trans. Signal Processing. Vol 41. No. 12. Dec. 1993.
- [3] Man, Kossentini and Smith "A Family of Efficient Channel Error Resilient Wavelet/Subband Image Coders." IEEE Trans. Circ. and Sys. for Video Tech. Vol. 9, No. 1, Feb. 1999. p.95
- [4] Lu, Algazi and Estes "A Comparative Study of Wavelet Image Coders." CIPIC. University of California, Davis, CA.
- [5] Kovačević and Vetterli "Nonseparable Multidimensional Perfect Reconstruction Filter Banks and Wavelet Bases for  $\mathcal{R}^n$ ." IEEE Trans. Info. Theory. Vol. 38, No. 2 March 1992
- [6] Andrews and Nguyen "Separable and Quincunx Image Coding." Proceedings of 6th IEEE Workshop on IS-PACS, Melbourne, Nov. 1998.

[7] Andrews and Nguyen. Predicting Masking Shadows in Zero-tree Wavelet Image Coding. In *Proc. IEE/IEAust intl. Conf. on Artificial Intelligence in Science and Technology*, December 2000 (Hobart, Australia).

# PREDICTING MASKING SHADOWS IN ZERO-TREE WAVELET IMAGE CODING

Richard Andrews

University of Tasmania, Hobart, Australia

<Richard.Andrews@utas.edu.au>

## ABSTRACT

A novel tool which exploits the phenomena of spatial contrast masking is presented as an enhancement to the Zero-tree coding method. It provides significantly decreased bit-rate while sustaining perceived visual quality. The enhancement is a logical and simple to implement modification of the zero-tree method and requires no side information to be coded.

## Introduction

The phenomenon of Spatial Contrast Masking causes minor image details to be obscured by significant edges in a region of an image. This phenomenon can be exploited to increase the efficiency of image coding where fine details are located in the vicinity of a major edge. Such areas affected by spatial masking are referred to as a Masking Shadow. Figure 1 shows an approximation to the variation in the perceptual contrast threshold for visibility of a stimulus in the vicinity of a significant edge.

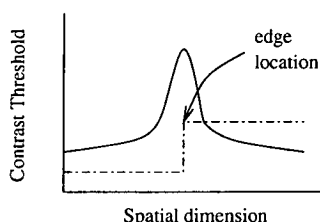


Figure 1: Visible contrast threshold of the human visual system in the vicinity of an edge. [1]

This paper presents a method of applying the masking phenomenon to zero-tree encoded wavelet transforms of images. The process predicts the presence of masking shadows through early embedded information in the zero-tree stream. Because it is predictive no side information needs to be added to take advantage of the predicted masking shadows, rather the method selectively removes information at an early stage of coding which is predicted to be obscured by masking shadows. The predictive nature differentiates

this process from other methods of utilizing spatial masking, which rely on sending side information to describe the degree of masking or *activity* detected in various regions of an image.

## 1. WAVELET CODING

Wavelet transforms have been found to be highly effective decorrelation transforms for image signals, and have been shown to have excellent properties for image compression in terms of both reconstructed PSNR and perceived image quality for a given bit-rate. Perceptually the wavelet transform is a near optimal representation of image data as it relates to the early Human Visual System, as it shares a similar subband, multiresolution nature.

In this paper the quincunx sampling lattice is used in the wavelet transform and thus the wavelet filters are designed for this lattice. This is a choice based on the continuation of previous research into the topic, and is aimed at increasing exposure of a mostly overlooked branch of wavelet coding. However, the methods and ideas outlined in this paper are equally applicable to any wavelet transform domain to which zero-tree coding of the coefficients can be applied. Appropriate modifications to the concepts of parent-child relationships in decompositions would need to be made for different resampling methods.

The quincunx lattice has been shown to have two properties which make it an attractive prospect for perceptual image coding. Firstly the quincunx lattice creates a sampling pattern with a frequency response cut-off which is diagonally oriented. Secondly it provides a more gradual decimation than separable sampling, which provides finer granularity for tuning spatial-frequency related phenomena.

## 2. ZERO-TREE CODING

The zero-tree coding process [3] is a process of detail refinement. An image which has been decomposed by a wavelet transform is coded starting at the lowest resolution levels and the coefficients of greatest magnitude. Each refinement first codes the position of coefficients which are discovered

to have a significant magnitude with respect to some threshold (which changes for each refinement operation). This is termed a *dominant* or *significance* pass. For all coefficients which are found to be significant with respect to the threshold, refinement information is also coded. This extra refinement is termed a *subordinate* pass and it is performed for *all* coefficients known to be significant with respect to the current magnitude threshold. Those found to be significant in a previous pass are also refined to have a greater precision in this pass. Thus for each reduction in the threshold, the number of subordinate bits required grows quickly.

Dominant and subordinate passes continue to alternate until the image is coded to some desired state, eg. bit-rate or number of refinements. Coding can terminate at any time.

### 3. PREDICTION OF MASKING SHADOWS

Spatial masking occurs in areas of high “activity” [1], which is generally measured in terms of some spatial derivative of pixel intensity local to the area of concern. The cause of spatial masking is significant and abrupt changes in image intensity resulting in very high contrast features, ie. edges. Mathematically these features are represented as regions with a large intensity gradient. When this occurs the human visual system becomes less sensitive to minor contrast changes in the vicinity of the edge. So in the presence of high contrast features, the visibility of minor features is reduced. The greater the change in intensity at the edge, the greater must be the contrast of a nearby feature to be visible. As distance from a significant feature increases, the effect of the contrast masking decreases. The area within which the effects of spatial masking are significant is the Masking Shadow caused by a significant feature.

In a subband context edges can be regarded as features which span multiple frequency bands while maintaining the same spatial location. In a wavelet decomposition this is marked by coefficients of similar magnitude and (typically) orientation in close spatial proximity at different levels of decomposition. Edges of high contrast produce wavelet coefficients of large magnitudes at many resolution levels; alternatively, many superimposed frequency band components are required to render a large gradient.

We can locate features which cause a large degree of masking by finding areas of an image’s wavelet decomposition which contain significant “overlapping” wavelet coefficients at several resolution levels.

If we consider the zero-tree coding of a wavelet decomposition we will notice that the first dominant (significance map) pass detects all coefficients within the decomposition which are very large in magnitude. This is exactly the information that we use to determine which areas of an image have a high degree of masking.

Because of the order in which a zero-tree method codes

a decomposition this significance information becomes available very early in the coding process; well before the vast majority of coefficients are processed. The process presented here extracts this information as it becomes available and uses it to better code further refinements based on the likelihood of spatial masking.

Because of the embedded nature of zero-tree coding, the exact same information used by the encoder to determine areas of masking is available to the decoder exactly when it is needed. For this reason, no side information needs to be sent.

### 4. MASKING-ADAPTED ZEROTREE CODING

This method adapts the coding process based on the masking predicted from the first dominant pass, and alters the zero-tree significance threshold on a coefficient by coefficient basis depending on the predicted effect of local Masking Shadows.

Those coefficients which have already been determined to be significant at the adaptation stage will not be affected. All coefficients which have not been found to be significant can be encoded with an arbitrary initial threshold - provided it is larger than the current threshold. It should be noted that for the first dominant pass, only a small number of coefficients will be found to be significant, those which have the very highest magnitudes. These define the most significant image features. Importantly these significant features cannot be modified by the coding process, only the lower magnitude features which contribute to fine details will have their coding altered.

Each pass through the zero-tree coding process reduces the significance threshold - typically a factor of  $\frac{1}{2}$  is used. This principle is fundamental to the behaviour of the zero-tree coder. For the modified coder the same principle applies, but each coefficient in the decomposition now has its own independent threshold. Each threshold is scaled by the same factor at each step in the coding process.

By raising the initial threshold of those coefficients found to be masked, we lower their priority in the coding process. This means that they are less likely to be coded at some particular refinement level than the coefficients with lower thresholds. This effectively means that these masked coefficients will be either (a) more coarsely coded than their unmasked counterparts, ie. fewer bits of precision, or (b) not coded at all if the threshold does not reach a sufficiently low level during coding.

The nature of the zero-tree coding algorithm is such that it most effectively codes areas of an image where there are no significant coefficients over multiple resolution levels. Thus areas of an image in a Masking Shadow are likely to have whole regions across many resolution levels where the coefficients are insignificant to a finer refinement level.



This decreases the bit-rate of the coded image, and generally without adversely affecting perceived image quality.

Figure 2 shows the result of coding the Lenna image with a single significance pass using zero-tree coding<sup>1</sup>. Comparing this with the original image shows how those features which are represented by the first pass data correspond to areas where significant masking would be present.



(a)

(b)

Figure 2: (a) Lenna image and (b) Lenna coded by a single pass of a zero-tree coder.

#### 4.1. Contribution versus Masking

It is important to be able to differentiate between those minor coefficients which contribute to the precision (sharpness of contrast) of a major edge and those which are in the vicinity of a major edge but which contribute to a minor feature in the Masking Shadow. The term “significant coefficient” is used here to describe a coefficient which is significant with respect to the first dominant pass of the zero-tree coding process.

The spatial proximity and resolution-level proximity of a coefficient to its significant neighbours is very important. If a minor coefficient is very close to a significant coefficient then it is likely to contribute to the feature, thus it should be considered as contributing to the masking shadow rather than being affected by it. If it is very far from a significant coefficient then the masking effects will be very small because the masking diminishes quickly with distance. Hence there is a narrow range of spatial distances from significant features in which Masking Shadows have noticeable effect. There are however several resolution levels for each spatial location so a masking shadow of small area in a low-resolution level can result in many masked coefficients in finer levels. This is particularly true in the higher resolution levels where most masking takes place and where the majority of the coefficients are contained but are of minimum

<sup>1</sup>This uses a flat weighting of all subbands. Better tuned coding would weight lower resolutions more significantly, thus decreasing bit-rate further. The bit-rate of this first pass image is 0.0138 bpp (580:1 compression ratio).

individual perceptual value (see Figure 4).

#### 4.2. Spatial Frequency Considerations

The human visual system has a sensitivity to features which varies across different spatial frequencies. The sensitivity drops off very quickly at higher frequencies, has a peak at around 3 to 5 cycles/degree and drops off slightly for lower frequencies. Figure 4 shows the approximate shape of the sensitivity curve.

This sensitivity variation provides us with another efficient means of coding a wavelet decomposition. We can more coarsely code coefficients from higher resolution levels which will contribute to features with higher spatial frequencies where the tolerance for quantization error is quite large. This is convenient because the very high resolution levels contain the majority of coefficients, and are of lowest perceptual importance.

When considering the interaction of spatial frequency sensitivity and proximity masking, it becomes apparent that the lower resolution levels of a subband decomposition are more important and should be coded more precisely since they contribute to highly visible features covering a large area in the image. Thus when considering coefficients for masking there must be a weighting which favours the masking of higher resolution coefficients over the masking of lower resolution ones depending on the sensitivity of the HVS to the band in question.

### 5. PREDICTION ALGORITHM

#### 5.1. Neighbours and Parents

The degree of masking which a coefficient is subjected to is determined by the significance of its neighbours and the neighbours of its *parent*<sup>2</sup> coefficients. It is unlikely that coefficients will mask those of a lower resolution.

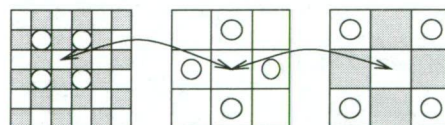


Figure 3: Three levels of parents and their 4-connected neighbours in a quincunx decomposition. Grey indicates pixel is not in the retained coset.

Figure 3 shows the four connected neighbours of a coefficient and two of its parent coefficients (through quincunx downsampling). The 4-connected neighbours are shown

<sup>2</sup>Any coefficient location produced by the downsampling transform is a parent of the original coefficient. A coefficient may have many children, and many coefficients may share the same parent.

with circles and the arrows show the parent-child relationships for a single coefficient location. These neighbours are in closest proximity to the coefficient in question and thus exert the greatest masking influence on it. For the purposes of the masking algorithm we will assume that features outside of one of these 4-connected neighbourhoods do not exert sufficient masking influence to be considered.

The discussion in this paper concentrates on the case where the quincunx sampling lattice is used. For this reason a 4-connected neighbourhood is convenient. If a separable decomposition was used then an 8-connected neighbourhood would be equally applicable.

Significance in any of the neighbour coefficients shown, indicates some degree of masking suffered by the (central) coefficient in question. While significance in any of the direct parents indicates that the child is most likely part of the feature and should not be masked.

## 5.2. Variation with Subband

The cause of masking is abrupt and significant changes in image intensity creating features of significant contrast. Coefficients at the lower resolution levels only contribute to the slowly varying components of an image which do not lead to high contrast features (except through superposition with higher resolution details). Hence the very lowest resolution coefficients should be considered to have little masking effect. A weighting factor is needed to account for this lowered contribution. It seems logical that coefficients in the subbands lying in the peak of HVS sensitivity will generate features of greatest perceived contrast and hence will contribute most to the masking of other coefficients. Hence a weighting based on the HVS spatial-frequency sensitivity to the masking coefficient can be used.

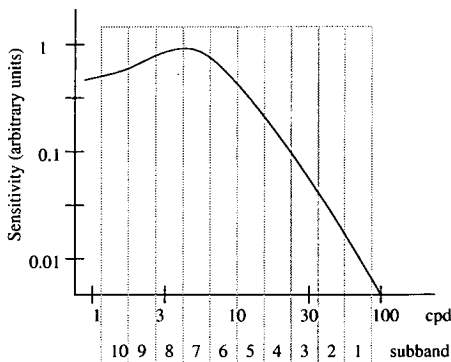


Figure 4: Sensitivity function for the HVS with approximate spans of quincunx subbands[2].

Figure 4 shows an approximation to the accepted model<sup>3</sup>

<sup>3</sup>The units used in the sensitivity functions found in literature have arbitrary units because they are a relative measure against similar experiments

[2][5] of the spatial frequency sensitivity function of the HVS. Shown also are the approximate spans of the quincunx decomposition levels<sup>4</sup>. It is apparent from this diagram that the greatest sensitivity is at subbands 6 through 10, so they should not be considered for spatial masking. Also the peak occurs in levels 6 and 7, thus these will have significant impact on the masking of higher resolution levels (ie. levels 1 through 5).

In subbands 1 through 5 there are 253952 coefficients (for a 512x512 pixel image) or 96.8% of the coefficients of the entire image. It can be appreciated then, that any method which reduces the data-rate per pixel in these higher resolution subbands will significantly reduce overall data rates. Also because these subbands lie outside the most sensitive areas of the HVS, quantization errors will have smaller perceived effects on image quality.

## 6. RESULTS

A 12 level quincunx decomposition of the 512x512 pixel Lenna image was used. Each level of the quincunx decomposition reduces overall number of pixels by a factor of 2 ( $\sqrt{2}$  in each dimension [4]), so 12 levels of decomposition results in an low-pass image equivalent to that resulting from 6 levels of separable wavelet decomposition. The resulting low-pass is of size 8x8 pixels.

The mean value was removed and coded separately so that the low-pass level could be treated as part of the zero-tree structure. Coding proceeded with 3 complete passes of the zero-tree [3] encoding method (significance followed by refinement).

The control example was coded as described in [3] (modified for quincunx sampling) for three complete passes. The comparison example uses the Masking Shadow prediction method outlined above to alter the zero-tree threshold on a per coefficient basis throughout the decomposition after the first pass. A masking factor weighted by spatial frequency sensitivity and the significance of 4-connected neighbours of parent coefficients was used for each coefficient not significant after the first pass of the zero-tree coder. The zero-tree threshold is only ever increased for any coefficient. This ensures that no extra coefficients can be brought into the significance map as a result of altering the threshold, it can only postpone the inclusion of coefficients if they are masked. After the thresholds were adjusted, the zero-tree coding was allowed to progress to completion.

under similar conditions. Their results are generally not portable to any degree of accuracy. They merely give a general representation of the changes which occur as we move through the spectrum without changing any other effects.

<sup>4</sup>Standard image size of 512x512 pixels and viewing distance of 15 times the image height. Levels are numbered starting with number 1 being the highest resolution detail possible from the image.





unadjusted



masking adjusted

Figure 5: Comparison of zero-tree encoded Lenna, without and with masking shadow adjustment.

On the decoding side, a similar process was used. The zero-tree encoded stream was read until the end of the first zero-tree pass. Thresholds were then altered based on the same significance information which was present on the encoding side, thus ensuring that the zero-tree thresholds determined were identical to those determined for the encoding stage. After the thresholds were adjusted up, the rest of the decoding process was allowed to proceed as normal.

Figure 5 shows the Lenna image encoded as described. The image on the left is encoded without any Masking Shadow adjustment, the image on the right is adjusted. Simple comparison of the zero-tree data shows that the unadjusted image produced a zero-tree stream with 55825 symbols (for three complete passes of the zero-tree coding process), while the adjusted image produced a stream of 39482 symbols - a 29% reduction. The encoder used a 4 symbol alphabet for all stages.

For comparison some indication of coding bit-rates is useful. Arithmetic coding<sup>5</sup> of the streams indicates an approximate bitrate of 0.15 bpp for the unadjusted image and 0.11 bpp for the adjusted image - a probable 25% decrease in bit-rate.

## 7. SUMMARY

Through the examples presented it has been shown that significant reduction in bit-rate can be achieved through the

use of Masking Shadow prediction in zero-tree encoding of wavelet transforms. The perceived image quality is shown to be preserved after the reduction of data, although some additional artefacts can be perceived.

## 8. REFERENCES

- [1] David McLaren. "Efficient DCT Video Coding for Broadband ISDN." PhD Thesis. University of Tasmania 1992
- [2] Ngan, Leong and Singh. "Adaptive Cosine Transform in Perceptual Domain." IEEE Trans. ASSP. vol. 37, no. 11, 1989
- [3] Shapiro. "Embedded Image Coding Using Zerotrees of Wavelet Coefficients." IEEE Trans. Sig Proc., v.41 no.11 1993
- [4] Andrews and Nguyen "Separable and Quincunx Wavelets Image Coding." 6th IEEE International Workshop on Signal Processing and Communication Systems. Melbourne, Australia. November 1998.
- [5] Gunawan and Nguyen. "Psychovisual Image Coding Using Wavelet Transform." Australian Journal of Intelligent Information Processing Systems, Autumn Issue, vol. 2, no. 1, 1995.

<sup>5</sup>The arithmetic coding method used was derived from a freeware implementation by John M. Danskin (based on a description from Bell, Cleary, and Witten Text Compression).

## CHAPTER 1

### Introduction

#### 1.1. Need for image compression

With the use of multimedia content becoming common there is a need for developing more efficient means of expressing that content. Uncompressed imagery, particularly photography, takes an enormous amount of data which makes its use impractical. It has been the development of efficient and fast image and video compression methods, particularly JPEG [33], [107] (Joint Photographic Experts Group) and MPEG [75] (Moving Picture Experts Group) standards, which have brought the world of popular multimedia into being. However as the amount of visual data being shared increases it puts greater strain on network and storage media. More efficient methods are needed.

When we discuss digital imagery and the compression of digital images, the fundamental aim is to gain the optimum image quality from the smallest amount of digital data. The terms file size and bit-rate will often be used in an abstract sense within this Thesis to describe the amount of data which is required to represent an image in its coded form; regardless of whether the image is actually stored in a file or in transit across a network. It is the concept of representing the number of bits necessary to store the information which is important. This data must be stored in some fashion, indeed to transmit an image across a network it must first be present in some file-like structure whether in computer memory or in some other storage medium.

Bit-rate is used in a similar manner but it will usually be used to provide a concept of the number of bits required for each pixel in the image. This is a better measure of the efficiency of image representation because it is independent of the size of the image being considered.

Compression ratio is a term closely related to bit-rate. By definition it is the ratio  $R = \frac{I_{uncompressed}}{I_{compressed}}$ . Here  $I_{uncompressed}$  is the amount of information required (in bits) to represent an image in its uncompressed form, typically 8-24 bits per pixel; and  $I_{compressed}$  is the amount of information required to represent the image in some coded form. The larger the compression ratio, the more image area can be expressed by a given file size.

Typical compression ratios which can be obtained are in the range of 15-50 for today's generation of image compression tools such as JPEG; and in the range of 30-100 for state of the art compression tools giving the same image quality.

## 1.2. Digital Image Basics

**1.2.1. Discretization into Pixels.** To store images in digital media it is necessary to have some means of expressing an image in numerical values. In common digital representations, the image becomes discrete in two fundamental ways;

- (1) spatial location and
- (2) image intensity.

A natural image can be magnified indefinitely (at least down to the quantum level) since the image is continuous. One can also measure the brightness or intensity of an image to an arbitrary precision limited only by the accuracy of the instruments used.

In contrast a digital image is typically represented as pixels. The term “pixel” being a concatenation of the words Picture Element. The term “pel” is interchangeable with “pixel”.

The concept of a pixel is of a small rectangular area of image over which the image has constant intensity and colour. Hence the image is sampled as a piecewise constant function. If the area of a pixel is small enough then the viewer will not notice that it is a square area, rather the smoothing of the Human Visual System (HVS) will give the appearance of a natural image. However when magnified, the true nature becomes apparent.

This term image intensity is used to mean the brightness of the image at a given point. However it is generalized to also refer to the properties of colour and brightness of the image. Image intensity is discretized by sampling the intensity and rounding it to a nearest value which is expressible.

With these two discrete ways of representing an image, we can completely define an image in terms of digital data. We have a finite number of discrete image pixels each of which can be indexed by a discrete coordinate set and we have a discrete value at each pixel.

Our image can then be represented by a discrete finite set of 3-tuples being (*co-ordinate 1*, *co-ordinate 2*, *intensity*) which is usually written as  $(x, y, I)$ . We require one intensity value for each pixel. It is usually possible to store these 3-tuples in a predictable manner such that the two spatial location indices can be inferred. Thus only the intensity values need to be stored. This is the principle of the Raster scan method. In this method, the width of the image must be known. All pixels intensities are stored sequentially starting at “top-left”<sup>1</sup> and scanning one row at a time. When the end of a row is reached scanning moves “down” a row and begins from “left” again on the new row. In this manner the position of the intensity value in the sequence completely determines its spatial position relative to the origin of the image.

---

<sup>1</sup>Note: this could mean any corner of the image depending on the definitions of up, down, left and right in effect.

One can see then that the amount of information required to express an image is equal to  $I = N_{pixels} \cdot I_{pixel}$  where  $N_{pixels}$  is the number of pixels in the image and  $I_{pixel}$  is the average information required per pixel. There are a few typical values for  $I_{pixel}$  in use:

- 8 bit per pixel is used for high quality black and white (more correctly known as greyscale) images.
- 15, 16 and 24 bits per pixel are in common use for full colour images.
- 32 bits per pixel exists in some specialized formats typically used in high-quality imaging such as digital photography.

### 1.2.2. Digital Colour Images.

1.2.2.1. *Tri-stimulus Principle.* Human vision has three colour receptors which respond to wavelengths within their respective sensitivity ranges [120, Chapter 3]. When we see the colour of an object our eyes receive many different wavelengths simultaneously, and it is the mixture of the responses of our three colour sensors which we interpret as the overall colour. There is no one wavelength that we are receiving, rather a continuous range of wavelengths at different strengths.

We can produce the same colour experience for the human observer by transmitting to the eye only three wavelengths, which correspond roughly to each of our colour receptors. A spectral power distribution of such a situation may bear little resemblance to the actual power spectrum received when looking at an object, but to the human observer the two colours will be indistinguishable.

This is referred to as the tri-stimulus principle and it is common to everyone<sup>2</sup>. Our three colour receptor cells have properties which give greatest response at colours we call red, green and blue. This principle is the basis of all digital colour representation.

1.2.2.2. *Colour Spaces.* Colour images are created by mixing the three primary colour elements together. This can be done in two general ways, either additive colour or “subtractive” colour. Additive colour increases the amount of the various colours (or primary phosphors as they are sometimes called) to produce a perceived colour. The starting point is black and adding colour elements increases brightness and adds hue. Subtractive colour starts with a given background colour (usually white) and absorbs particular frequency bands to produce a perceived colour. This is how paint works. Subtractive colour processing is much more complicated than additive colour processing and relies on highly non-linear mathematics and proprietary “black-magic” to achieve good results, but it is an essential part of the print industry.

The primary stimuli of the additive colour method are typically called called R, G and B (red, green and blue ). There are alternative colour space representations

---

<sup>2</sup>The tri-stimulus principle is no less valid for those with colour dysfunction (colour-blindness) as colour blindness only removes colour sensors. The only people for whom the tri-stimulus principle is not valid are a very rare set of females known as Tetrachromats. This genetic anomaly results in humans with four different colour receptors.

which transform this additive representation into some other form. Y, Cr and Cb [33] (luminance, chrominance red and chrominance blue) is used when the brightness of an image is important, HSI (hue, saturation and intensity) is used when subtle colour blending and shading is required.

The RGB [33], [1, chapter 55] representation uses a brightness value for each primary phosphor which gives an overall intensity or brightness, while the YCC and YIQ [44, III-6] (as used in NTSC television) methods use a brightness or luminance value (Y) and two *relative* colour components (which may be positive or negative in value). The actual amounts of RGB needed to result in the correct colour and brightness can be calculated from these values. The YCC methods have the advantage that they can easily be used to convert from colour to grey-scale (black and white) pictures by retaining only the luminance components - hence the ability for black and white television to display colour signals - it only uses the luminance component. YCC is typically the preferred colour space used in JPEG compression because the separation of colour from intensity allows for a sparse sampling of the colour data compared to the intensity data. Separating colour from brightness allows increased compression since the human visual system is less able to detect changes in colour over small distances compared to intensity.

A popular method of colour visualization is the colour cube. This is a representation of a three dimensional (R, G, B) colour value within a Cartesian space where the value of each primary colour determines a distance along a cardinal dimension of the colour space. Figure 1.2.1 shows how this works.

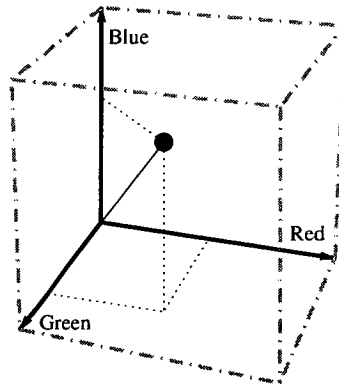


FIGURE 1.2.1. The RGB colour cube visualized as a three-dimensional co-ordinate system.

Colour cube and colour wheel representations are used by digital artists in the selection of colours as they allow for the representation a colour in terms of a Hue, Saturation and Intensity/Volume (HSV). The Hue and Saturation together determine the “colour” and are chosen from a colour wheel, where Hue is determined by the angle around the wheel from the origin and Saturation is determined by the distance from the centre of the wheel and represents the purity of the colour (phosphor). As Saturation approaches maximum, at least one primary phosphor



is driven at maximum providing a 'pure' colour output. The final value, Volume determines brightness. Figure 1.2.2 shows how this operates.

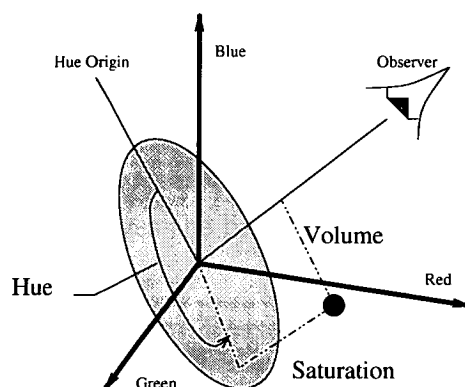


FIGURE 1.2.2. Colour wheel visualization of 3-D colour-space using the HSV model.

Looking down on the colour wheel the observer sees all possible colour hues. Due to the orientation of the colour wheel within RGB-space the three primary colours - red, green and blue - are separated by  $120^\circ$ . Every possible relative combination of magnitudes of the primary colours can theoretically be represented on the wheel by choosing a Hue (angle from the origin) and a Saturation (distance from centre). In practice not all possible RGB values are representable in this way without some non-linear trickery. This is because the HSI method can represent colours outside the possible range of RGB values (called the colour gamut) of the display hardware.

With the relative combination chosen the only component to determine is the total brightness or Volume (height above colour wheel). Increasing Volume proportionally increases the values of each of the primary colours and retains the same relative magnitudes within the perceptual colour model of the display device. Hence the brightness is changed without changing the perceived colour. This representation is popular among colour industry professionals because it provides an intuitive representation of colour-space and by altering only the Intensity value, the same colour can be faded from bright to black. With a Intensity value of 0 the resulting colour is black regardless of the relative combination of colours. The entire grey-scale is also represented on the Intensity axis. With Volume (distance from centre of colour wheel) set to 0, the R, G and B values will be identical and simultaneously vary with Intensity. Intensity of 0 gives black and Intensity of maximum (usually called 1.0) gives the brightest colour the display can represent.

### 1.3. Lossless and Lossy Data Reduction

Methods of coding data can be grouped into two broad categories: lossless and lossy. For digital data both categories are available and used in different circumstances.



The term “lossless” is used to describe coding methods where the original data can always be exactly recovered from the coded data. This can also be called reversible coding, since reversing the process recovers the original data. This type of coding is most useful in applications where all data must be preserved exactly, examples are in the compression of text or computer programs. Information which has a precise significance must be compressed in this way.

The term “lossy” is used to describe the converse case where the reconstructed data after decoding might be different from the original. This usually involves constraining the process so that the reconstructed signal has certain key properties which are retained while other less important properties may be lost or changed in the process. This can be described by the phrase “nearly the same”.

Lossy coding can produce much higher compression ratios than lossless coding but it is only suitable for data which has some perceived meaning which can be retained at the expense of exactness.

It is the lossy category into which the majority of image compression systems fall today. The key property which these systems attempt to preserve is the way the image is perceived by the intended viewer. Because of this it is possible to discard much information from a coded image and yet still retain the original meaning. When examined closely for differences, by a computer for instance, the subtle changes would become immediately apparent as numerical errors in brightness or colour, indeed they may be visible with the naked eye.

**1.3.1. Reconstruction Quality.** When we compress a signal through a lossy method, the reconstructed signal will usually be different from the original. We need some way to determine how closely the reconstructed signal resembles the original signal. The degree to which the two are similar (or different) from some particular perspective is referred to as the quality of the reconstructed signal.

When dealing with digital images there are two broad categories into which measures of reconstructed signal quality fall: statistical and perceptual. Statistical quality measures are used to describe, in a mathematical manner, how closely the sampled numerical values of the reconstructed and original signals correlate. The most commonly used measures of this type are the Mean Square Error and the Signal to Noise Ratio (SNR) and there are various modifications to these methods which are used to apply them to different signal contexts - Peak-SNR (PSNR) is common in image quality assessment. The common feature they have is that they compare the numerical value of a signal at each sample and create a single number which describes some function of the errors at all of the sampling points. An interesting point to notice about these types of measures is that if we shift the image by just one pixel and not change any values, we will get a significant error measure, yet the meaning of the image has not changed in any significant way from the point of view of the observer. This is a simple example showing the limited usefulness of such measures of quality in a perceptual context. It should be noted

that image coding methods attempt to return the image to its original location, however, sampling rate changes can lead to sub-pixel movement of an image which causes reasonable errors to be returned from these methods but does not affect the appearance of the image. A simple smoothing of the signal before determining error can be useful in overcoming this limitation, but this is only possible if the signal is of sufficiently high resolution that the process of smoothing does not noticeably (in terms of the observer) change the image.

Perceptual quality measures are more difficult to create because they rely on modeling human perception mathematically in a way which allows perceived errors to be assigned a numerical severity. Another factor which makes this task difficult is that no two people have exactly the same perceptual response and so some mean measure must be found which adequately covers everyone. This necessarily leads to a certain degree of statistical modeling of human response. When dealing with a complex perceptual system such as human vision it is often necessary to deal with only those primitive response models which can be determined. In particular the responses of the very early parts of the human visual system can be modeled as simple electrical circuit components. One reasonably successful method of determining reconstructed image quality on a perceptual basis is presented in [93].

The psychological components of human visual perception are impossible to measure directly and so we rely on what response we can ascertain from experimental subjects using primitive stimuli. The unfortunate part in this is that it is difficult to determine what the response of the HVS will be to a combination of such primitive stimuli because the HVS is extremely non-linear in its response.

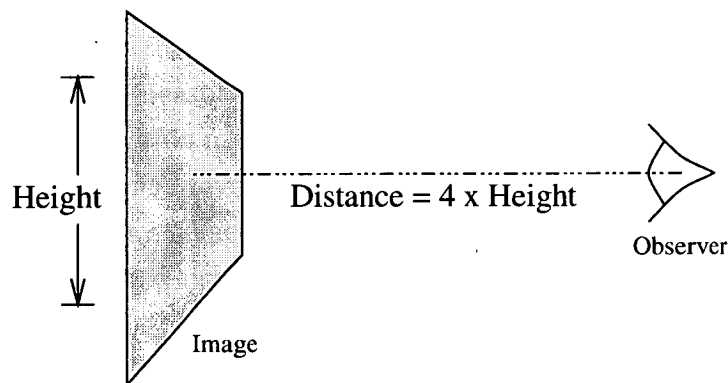


FIGURE 1.3.1. Common viewing requirements for human vision experiments and models. Viewing distance of full screen images is often assumed to be at a distance of four times the screen height.

Another problem with constructing models for human perception of digital images is that we have no way of knowing at what resolution the image will be displayed when viewed by the human viewer. The only true measure of resolution is in terms of solid angle; the angle covered by an object when viewed from the

focal point of our eye. Solid angle is a measure which defines a conical field through 3D space (see figure 1.3.2).

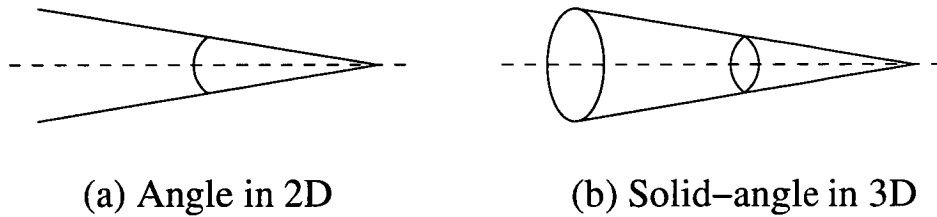


FIGURE 1.3.2. Comparison of angle and solid-angle. Angle is a 2D measure which defines an arc at some extent. Solid angle defines a spherical surface at some extent.

Any procedure we develop for optimizing the viewed quality of an image is intrinsically linked to the resolution (in terms of pixels per unit solid angle) at which it is displayed. A common assumption used when conducting experiments and designing coding methods is that the image will be of a size such that its distance from the viewer is four times its height (see Figure 1.3.1). Thus an image displayed on a large screen is expected to be viewed at a correspondingly greater distance. Once this condition is broken, which would happen if the viewer was too close or too far from the image, many of the principle factors used to optimally encode the signal become invalid and perceived quality drop.

## 1.4. Statistics of Images

There is a large amount of information contained in natural images. As the adage says “a picture tells a thousand words”. In a typical digital image there is a significant amount of redundant information and removing such redundancy can significantly reduce the amount of information necessary to express a digital image.

There are many ways of looking at the statistics of images and each way sheds light on different properties and redundancies within an image.

Considering how the image intensity varies from one pixel to the next is a first logical step. If we consider this then we see that in general there is quite a high statistical correlation in intensities from one pixel to its neighbour. This tells us that a large amount of information is wasted in describing each pixel intensity individually and it could be much more efficiently done using, for instance, the difference in intensity from one pixel to its neighbour.

This brings us to one of the key underlying concepts of digital imagery which is quantization. Although in nature the light intensity we might see can take any value and can vary over time, to represent images on computers we must make approximations to the actual level, in order to be able to store the intensity. In general 8 bits of data is sufficient to represent the intensity of an image and the

human viewer is unable to notice that any quantization has occurred. This is because, typically, the human visual system can only distinguish around 40 grey levels [72]. From this assessment we would conclude that 6 bits of information could adequately express all the intensity changes which the human visual system could perceive. 8 bits per pixel has been most uniformly adopted because it corresponds exactly to one byte of storage, which is a convenient unit to deal with on most computing platforms.

**1.4.1. Entropy.** Getting back to statistics, we can see that to express each pixel as an individual unit requires 8 bits under normal conditions. However when we consider groups of pixels, ie. image areas together then we can reduce this considerably. Using the example of the difference coding method, we would express the intensity of a pixel as the difference between the previous pixel and the current one. Thus, in general a much smaller number, ie intensity difference would need to be coded. If the image changed little from pixel to pixel then only a few possible states would exist for the pixel difference. If only a few values are to be expressed then a much smaller number of bits is required to uniquely define each value.

Entropy is a measure of the information required to uniquely represent a decision within an information framework. Typically it is expressed in bits per symbol, and in image coding work the symbols are most often pixels. The entropy (or self information) of a set of values is defined as

$$H = - \sum_i P(s_i) \cdot \log_D (P(s_i))$$

where  $s_i$  is the  $i^{th}$  unique symbol in the list of symbols and  $P(s_i)$  is the probability of occurrence of  $s_i$ .  $D$  is the number of possible values that each unit of information can take. Typically this is 2 and so the unit of information is the bit (two possible states). Hence this can be re-written as

$$H_{bits} = - \sum_i P_i \cdot \log_2 (P_i)$$

The entropy of a set such as this defines the theoretical minimum amount of information required to represent the set if each symbol is represented with a different number of bits dependent on its rate of occurrence. If a symbol occurs very frequently then it makes sense that it should require only a small number of bits to represent it. Thus the amount of information required to represent it many times can be greatly reduced. The cost of this is that to represent all symbols we must then allocate longer codes to the less frequently occurring symbols. This increases the amount of information required to represent one of the rare symbols but this is consistent with the idea that an uncommon event carries more information than a common event.

The worst case scenario for entropy coding is when all possible symbols are identically probable. In this case we cannot reduce the length of any code because no symbols are more common than any other symbols.

**1.4.2. Fourier Analysis.** Since the advent of affordable personal computing power, image transforms have become a common method of analysis. In particular transforms into the frequency domain, also known as Fourier analysis, are very common. Studying the spatial frequency statistics of image signals sheds more light on image properties and redundancies.

One of the key findings to arise out of Fourier analysis of images is that the majority of the information in an image is stored in the low-frequency section of the spectrum and only a small amount is stored in the high-frequency regions. The lower frequency areas correspond to coarse features within an image and thus have a more profound impact across an image, while high-frequency components correspond to finer details, which although significant to the viewer, have much less impact on the overall statistical construction of the image. Hence by discarding higher frequency components we find that we can create a good approximation to an image using only a fraction of the information required to store the original. Of course any fine details will be lost in this process, but from a statistical standpoint, the error will be relatively small.

## 1.5. Pixel Coding

Pixel coding refers to a family of coding methods which operate on the values of the pixels in an image and the inter-relationships between them. A simple example of this is a difference encoding method where an image is encoded using the difference from one pixel to the next. This method can be quite successful when the image properties change little from neighbour to neighbour.

**1.5.1. Predictive Coding and DPCM.** One signal compression method which has been adapted from one-dimensional signals into the realm of images is Differential Pulse Coded Modulation. Although the name stems mainly from the medium for which it was originally developed, ie. communication of digital data via radio or telecommunications system, the method remains an important one.

The basic principle is to develop a signal predictor system which can reasonably accurately predict the next signal sample value from the available context of “previous” samples. Using such a predictor, only the error between the predicted signal value and the actual value at the predicted sample is coded. If the predictor is good then the error signal values will be small and clustered around a zero mean, leading to a small number of bits required to represent the data. If there is no limit imposed on the error values which can be encoded, then this method of coding is guaranteed to be lossless.

Typically a predictor has either a linear or logarithmic rate of change and at times dramatic changes in the input signal lead to changes in signal value that

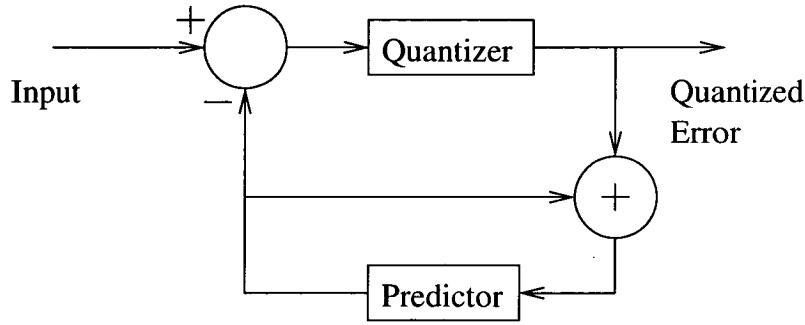


FIGURE 1.5.1. A predictive coding arrangement. The output from the predictor is generally close to the actual input so the difference is small. The error values are output to the symbol stream.

cannot be expressed with the available range of error values, in this case it can take some time for the output to track the input again. One such situation where this effect can occur is with  $\mu$ -law and A-law codecs for telephony.

## 1.6. Transform Coding

Transform coding is a method which represents a signal as a series of numerical values called transform coefficients which represent the content of particular primitive structures within the original signal. By analyzing these transform coefficients we can gain insight into deeper structures and patterns within the signals. Transform coding is effectively a basis transformation, where the transform coefficients represent the value of the signal on the new coordinate framework.

Since we are dealing with functions in general, the coordinate frameworks have sets of functions as the bases on which the signal is being projected. In a similar manner to the way a Cartesian coordinate can be projected onto another set of basis vectors for  $\mathbf{R}^n$ , so too can a sampled signal be projected onto a set of basis functions that span the function-space of the original signal.

The process of convolution can be used to project a signal onto a basis function. If the original signal is to be reconstructed then the set of basis functions which a signal is projected onto must completely span the function space of the original signal (ie. they must be able to reproduce by superposition, all the characteristics possible in the function space).

Sinusoids are a common basis set used in transform coding. In this case the set of basis functions is the set of sinusoids with all possible frequencies and phases within the signal domain. For sampled signals there is a maximum frequency representable as limited by the sampling density. Transform coding onto a basis of Sinusoids is called Fourier Analysis or Fourier Transform.

Other transform methods use basis functions which have a finite region of support. Among these are the Gabor and Wavelet transforms. In this case the set of basis functions used in the analysis needs to include shifted versions of a

mother function so that the function set can cover all points in the signal domain while using a basis with finite (compact) support.

**1.6.1. DCT (and JPEG).** One of the effects of using Fourier analysis is that at each frequency there are two complimentary sinusoids which must be accounted for to describe a signal in a manner that will allow it to be exactly recalculated (reconstructed) from the transform coefficients. These two parts can be thought of as either

- (1) Amplitude of Sine and Cosine functions with the same frequency;
- (2) Amplitude and Phase of a single sinusoid;
- (3) Real and Imaginary components of a complex phaser.

In each case, the result is that there are two parts to each coefficient.

To simplify this analysis method when dealing with signals with purely real values the Cosine Transform was developed. This method uses only cosine functions as its basis set and each coefficient has only a single real part to it. The side effect is that there are twice as many bases to account for, thus the information content remains constant.

For the analysis of discretely sampled digital data the Discrete Cosine Transform (DCT) [3] was developed to compliment the Continuous Cosine Transform for continuous signals. The DCT was found to be quite well suited to the analysis of image data because the majority of the energy contained within a natural image is contained within the low frequency components of the image. This means that an image can generally be well approximated by using only a small number of low-frequency DCT coefficients.

One of the disadvantages which the DCT inherits from Fourier analysis is that its basis functions are infinite in extent so they cannot describe the location at which particular features occur and also they are bad at representing localized features because of the large number of components needed to represent a singularity.

When sinusoids are the basis functions a change at one point in a signal will cause changes in many if not all transform coefficients. To combat this phenomenon the blocked DCT method was developed for image compression. This method breaks an image up into small blocks of pixels (typically 8x8) and performs a DCT on each of these areas separately. The advantage being that severe disturbances which occur in one block do not affect the transform coefficients in any of the other blocks at all thus localizing signal energy. However these blocks are located at predefined locations which do not necessarily completely contain a localized disturbance.

The method of Blocked DCT is used in the JPEG still image compression standard [33]. To compress the images this method examines the transformed DCT coefficients and performs a quantization process on the values where the quantization steps are based on the known average responses of the human visual

system to the primitive basis functions used in the transforms. The harshness of the quantization can be varied but in general many coefficients, particularly those representing high-frequency components, are quantized to zero. This allows for a very effective compression of the image data through run-length coding. The perceptually most-significant coefficients are coded first for each block. These are also the most finely quantized and are least likely to be zero. The coefficients are coded in a specific order with the most important first. This way it is typically the case that there will be many coefficients at the end (the least important ones) that have been quantized to zero. These can be very efficiently run-length coded.

For compression ratios of up to about 30:1 JPEG is an outstanding method [61]. Above that ratio, however, the loss of coefficient information leads to whole DCT blocks being harshly coded and badly represented which leads to highly visible errors where two DCT blocks of different intensities touch. These visible errors are known as blocking artifacts and they are a major limiting factor on compression achievable with the JPEG method. Since there is a fixed information cost associated with each block there is a lower bound to the size a JPEG stream can be for a given image size. Hence there is effectively an upper limit on the compression ratios which can be achieved.

**1.6.2. Wavelet Transform and Multiresolution Analysis.** Wavelets have found countless applications since their discovery - the solution of differential and integral equations [80], parametric curve approximation [20], signal de-noising and signal compression.

The Wavelet transform as used in signal compression is a multiresolution method which, as the name implies, analyzes signals using different levels of approximation to the original. The core idea behind multiresolution analysis (MRA) is that a signal is successively filtered or blurred to give an approximation to the original signal, and the differences between the approximated signal and the signal from which it was derived (which might also have been an approximation) are calculated. Only this difference information is required to reconstruct a higher resolution signal from its immediate approximation, so only the difference information needs to be stored.

In this way, when reconstructing the original signal, at each step in the process a better approximation to the original signal can be achieved by recombining the approximated signal (often call the low-pass signal) and the lost details (called the high-pass signal or the wavelet coefficients). In the limit the original signal can be represented by a single, very blurred approximation signal (requiring little data to represent) and a series of detail signals required to progressively reconstruct the signal one level at a time.

Compression of the signal data can be achieved because typically the detail signals contain only a small number of scattered coefficient which can be efficiently represented using a much smaller amount of data than for the original image.



Using intelligent quantization, many of the detail signal values can be set to zero, thus leading to good compression.

Intelligent coding schemes which make use of the inherent structure of a multiresolution decomposition can very effectively compress the signal. Multiresolution methods tend to start performing better than JPEG at higher resolution levels. At 30:1 compression ratio both are comparable [91] - with JPEG superior at lower compression ratios - however for higher compression ratios multiresolution methods are much more effective. This is because the signal quality degrades much more gradually than for the JPEG method so higher compression ratios can be achieved without significant perceivable errors.

Compression of up to 100:1 with good picture quality is achievable with multiresolution coding methods.

### 1.7. Human Vision

For the purpose of multimedia, the ultimate target for compressed images is the human viewer. It is the viewer who must be satisfied with the quality of the reconstructed image from a compressed data stream. Thus any quality measure must aim at the most useful image from a human perspective.

To do this we rely on modeling the behaviour of the human visual system so that we can predict the best optimization of image quality against required data size. As discussed above the modeling of human perception is very difficult, but due to many studies on the biological and psychophysical responses of the HVS it is now possible to model at least some basic responses. We have some understanding of the response to simple stimuli corresponding to different (spatial) frequency regions and we have some understanding of how some features can interact to obscure and enhance particular elements in an image.

What we do not yet understand is the way in which the human brain transforms the information presented in an image into recognizable elements which are remembered. With this type of information the best image compression method would be able to be created.

### 1.8. Thesis Aims and Structure

The chief aim of this Thesis is to enhance the body of knowledge associated with nonseparable wavelet-based image compression and perceptual coding in ways that will benefit these fields as topics of future research and practical development, and to present sufficient evidence of the benefits of the nonseparable paradigm that new research work will be inspired.

To this aim, results are presented which compare separable and nonseparable processes from a psychovisual perspective. We present a coherent introduction to nonseparable wavelets which reduces the knowledge barrier against new researchers entering the field and provides a comprehensive learning tool on the subject.

Results of a comparative study of separable and nonseparable methods are presented which show the strengths of each. Deficiencies in current knowledge of the nonseparable paradigm are highlighted and the potential benefits of research in those areas is discussed.

Several novel processes are presented for nonseparable transforms which enhance its appeal as a platform for future image compression research and remove practical barriers to its practical adoption.

The state of research in the field is advanced by a coding process based on psychovisual principles and nonseparable wavelet transforms which is shown to provide better perceived image quality than the Shapiro Zero-tree method.

With this Thesis the author aims to create a comprehensive learning tool that will reduce the barriers of entry into the nonseparable domain for interested researchers thus leading to accelerated development in the field. To further this aim, the full source code for the experimental platform used in the research is released by the author to the general community.

**1.8.1. Structure.** Chapter 2 develops the theory of wavelets and covers some brief history on the early pioneering methods. Once an understanding of simple one-dimensional wavelets is developed we move onto two-dimensional systems, through the use of the Tensor product to generate separable transforms.

In Chapter 3 we cover the lattice theory which is necessary for the truly two-dimensional transforms to be developed. Then multidimensional non-separable wavelets are developed. A comparison of separable and non-separable methods is made from a psychovisual image coding perspective which will demonstrate the motivation for moving from one-dimensional separable processing to nonseparable methods.

A review of knowledge in the field of filter design is presented in Chapter 4. In this chapter we present the design of the filters which were used to produce the results presented in later chapters. While essentially a literature review, expanded explanations are presented for several areas central to nonseparable processing. Examples are used extensively to illustrate the concepts.

In Chapter 5 we derive new upsampled representation and coding methods which provide significant speed and storage improvements for nonseparable transforms. The upsampled representation was used in the implementation for much of the experimental platform.

In Chapters 6 and 7 existing coding methods are reviewed which make use of psychovisual models of human vision. Coding results are presented which show the effectiveness of basic psychovisual tuning when used in conjunction with wavelet transforms and the quincunx lattice in particular.

Zero-tree coding which forms an integral part of much of the research work is introduced in Chapter 8. Extensions are derived which allow zero-tree coding to be used in the nonseparable domain and results are presented showing the power

of the coding method in nonseparable processing. Issues associated with zero-tree coding on the quincunx lattice are covered in detail and results are presented from the use of zero-tree coding on the quincunx lattice. The use of arithmetic coding with zero-tree data streams is discussed and the advantages of the combination are outlined.

Cross-scale coding is discussed in detail in Chapter 9 and we introduce the concepts of masking shadow prediction. We outline how masking shadow prediction interfaces extremely well with zero-tree coding providing a framework for conveying activity measures without side-information.

Chapter 10 covers in detail zero-tree coding and the novel masking shadow prediction methods. Results are presented for zero-tree coding with arithmetic encoding. Masking shadow prediction is developed and results are presented showing that it is a powerful tool which can significantly increase the perceived image quality of zero-tree based streams with without increased bit-rate.

Finally we summarize the contributions and results presented and provide pointers to further directions of research related to topics covered in this Thesis. Some of the ideas which could not be fully investigated are outlined and discussed in some depth.

### 1.9. Original Content

This section briefly outlines the original contributions made by this Thesis.

This Thesis presents the first known work on the use of the quincunx resampling scheme in a coding system with the aim of providing efficient image compression<sup>3</sup>. It also describes the first integration of psychovisual coding methods with the quincunx lattice.

The first comparison of the performance of separable and nonseparable wavelet transforms under various conditions is presented. In general this comparison is a difficult task as the properties of the filterbanks for separable and nonseparable transforms are significantly different. Here the two-dimensional nonseparable filters are derived from the one-dimensional separable filters.

Cross-scale equivalence frameworks are developed for the quincunx lattice which allow for the use of such cross-scale coding mechanisms as quad-trees and zero-trees. The use of these frameworks on the quincunx lattice was not possible without these developments.

The novel upsampled wavelet transform method is introduced. This method is crucial to any practical application of nonseparable wavelet transforms. It removes significant computation burden associated with upsampling and downsampling required during wavelet transforms. The upsampled transform method shifts the resampling burden from the signal to the filters, where it can be precomputed and

---

<sup>3</sup>Previous work done in this area has centred on the mathematical framework and has only mentioned image compression as one possible application. Previous results presented have been for the purpose of demonstrating the mathematical theory.

stored for rapid use. Speed increases are significant. This method also allows for more accurate examination of processing phenomena by researchers and developers as the image can be stored in memory in its original orientation.

A technique for compact storage of wavelet transform coefficient is presented which allows in-place wavelet transforms to be conducted with quincunx resampling. This technique has application in embedded technologies where memory minimization is a priority due to size and power dissipation requirements.

An implementation of zero-tree coding on the quincunx lattice is presented. This is the first published results which show the applicability of the technique to nonseparable wavelet decompositions.

A new technique is presented as an extension of the zero-tree coding method which exploits the perceptual phenomena of spatial contrast masking and frequency selectivity to produce significantly increased image quality without an increase in bit-rate. The theory of masking shadow prediction is introduced in the paradigm of quincunx resampling but is also applicable under any resampling mechanism.

## CHAPTER 2

### Wavelets and Subband Coding

This chapter reviews the theory of wavelets and examines their behaviour contrasting them with the Fourier Transform. We first provide a historical overview of the preceding research and areas of inspiration for the work presented in this thesis. Next we cover wavelets in one dimension and then move on to two-dimensions and finally we briefly examine nonseparable wavelets in two dimensions. Aspects of design and application are covered focusing on the Discrete Wavelet Transform (DWT).

#### 2.1. A Brief History of Wavelets

Wavelets as we know them today are the result of an iterative process of discovery and generalization. Wavelets have developed in parallel in the fields of mathematical analysis and subband coding. They have been known for some year in the field of mathematical analysis [32], having found application in analyzing and representing general functions in a particular function-space. In this form wavelets have been used in the solution of integral and differential equations[80] in a variety of situations.

In unrelated developments wavelets were developed in the field of signal processing where they take the form of subband transforms and (critically sampled) filter banks [97], [100] which have been used successfully since the early 1980s for the compression and coding of speech.

It was the work of Mallat [63] in the field of image processing and Meyer in operator theory [69] that drew the connection between the mathematical definition of wavelets and their construction at a discrete level by the application of subband transforms. Multiresolution analysis combines both subband transform theory and operator analysis to provide a fundamental building block for the creation of useful wavelets. Of particular note in this field is the pioneering work of Daubechies in creating the first compactly supported (finite length), smooth, orthonormal wavelet families [17], [18].

Until the mid 1990s wavelet construction relied on frequency-domain design methods where translation and dilation become algebraic processes. The notable exception to this being the work of Donoho [21] using interpolating functions. In the mid 1990s the lifting scheme [90] was developed by Sweldens which uses simple predict and update operators in the time domain to allow construction of new wavelet transforms from existing transforms. It was then shown [54] that this process could be generalized such that any wavelets could be constructed from the

correct choice of predict and update operators in a ladder construction starting with a simple box function.

The work on nonseparable subband transforms in multiple dimensions by Vetterli, Karlsson, Kovacevic, Simoncelli and Adelson [48], [56], [83], [82], [2], [85] have provided the theory necessary for the development of the nonseparable wavelet compression methods developed in this thesis.

In the separable domain the work of Shapiro [81] has provided the inspiration for the majority of the principles used to create the efficient coders used to produce the experimental results.

## 2.2. Fourier Transform and Fourier Series

In some respects the Wavelet Transform is quite closely related to the windowed or short-time Fourier Transform (STFT), of which the Gabor Transform<sup>1</sup> is an example. Both transform the signal using finite length basis functions which are translated to account for position and scaled in some way to account for frequency bands. It is useful to briefly cover Fourier analysis and how it relates to Wavelet analysis.

Fourier series analysis is a mathematical transform which decomposes a signal into its separate frequency components. It is a particularly powerful method for the analysis of periodic signals due to the periodic nature of the basis functions which are used, ie. sinusoids. It is of limited use for the analysis of non-periodic (ie. finite) signals as the signal must first be made periodic before analysis can be performed.

The Fourier Transform is a related analysis tool which can be used on non-periodic signals. We can write the Fourier transform for a function  $f(t)$  as

$$(2.2.1) \quad F(\omega) = \mathcal{F}_\omega(f(t)) = \int_{t=-\infty}^{t=+\infty} f(t).e^{-j\omega t} dt$$

where  $\omega$  is the frequency of the basis function of interest. The integral spans  $-\infty < t < \infty$  which can be abbreviated to the region of support for finite length signals.

The inverse transform takes a function in the frequency domain and reproduces the signal in the time domain from it, thus making the Fourier Transform reversible or invertible. The inverse Transform complimentary to Equation 2.2.1 is given by

$$(2.2.2) \quad f(t) = \mathcal{F}_t^{-1}(F(\omega)) = \int_{\omega=-\infty}^{\omega=+\infty} F(\omega).e^{+j\omega t} d\omega$$

transforming a signal in  $\omega$  back to a signal in  $t$ .

One of the phenomena frequently experienced with the Fourier Transform relates to abrupt edges. Because the maximum rate of change which can be expressed

---

<sup>1</sup>Gabor Transforms use a Gaussian function as the windowing function to restrict the region of support of the sinusoids.

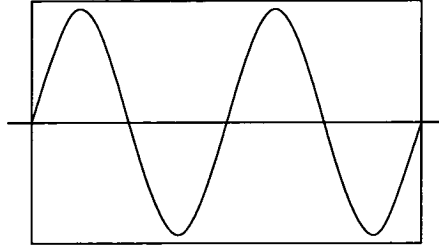


FIGURE 2.2.1. Sinusoidal transform kernel for the Fourier Transform. This function extends indefinitely.

by a sum of sinusoids is the sum of the rates of changes of the individual components, it is impossible to express a discontinuity as a finite set of sinusoids. A discontinuity results in an infinite rate of change of signal value, which can only be synthesized with Fourier components of infinite frequency.

For this reason the transmission of discontinuities in any media is impossible as every medium has a finite bandwidth.

The Fourier Transform  $F(\omega)$  is a function in  $\omega$  and its (usually graphical) representation over a range of frequencies is often referred to as a signal power spectrum. It is worth noting that  $F(\omega)$  is a complex function

$$\begin{aligned} F(\omega) &= R(\omega) + jI(\omega) \\ &= A(\omega)e^{j\phi(\omega)} \end{aligned}$$

it has both a magnitude,  $A(\omega) = |F(\omega)| = \sqrt{R^2(\omega) + I^2(\omega)}$ , and a phase angle,  $\phi(\omega) = \arctan\left(\frac{I(\omega)}{R(\omega)}\right)$ , at every value of  $\omega$ . The magnitude,  $A$ , represents the relative size of the component of frequency  $\omega$  in  $f(t)$ , while the phase angle  $\phi$  indicates how each sinusoidal basis function is aligned (delayed) with respect to a common reference. Two time-varying signals that have the same magnitude function (spectrum) will look very different if the phase angles of their components are different.

One often sought after quality in the Fourier Transform of a function in signal processing is linear-phase. This refers to the variation of  $\phi(\omega)$  with  $\omega$ . If the phase angle of a Fourier transform varies linearly with frequency, this translates to a constant delay in the time domain regardless of frequency. This property is important from a signal processing perspective because processing a signal with a linear phase filter does not spread the signal energy of significant features, eg. rapid changes in signal value (edges). Filters are often designed to have linear phase over a particular operating range of frequencies for this reason. In terms of finite impulse response filters, it can be shown that linear phase can only be achieved with symmetric filters - a result which generalizes to n-dimensions.

One of the limitations which is faced by the Fourier Transform is that its basis functions are infinite in extent, ie.  $\sin(\omega t)$  does not have a finite length over which it is non-zero. Thus the Fourier Transform integral must always be over all time, ie. from  $t = -\infty$  to  $t = +\infty$ . This is quite a limiting factor for practical application



because real life signals will not have infinite extent. Only those signals which are periodic can be properly analyzed by the Fourier Transform.

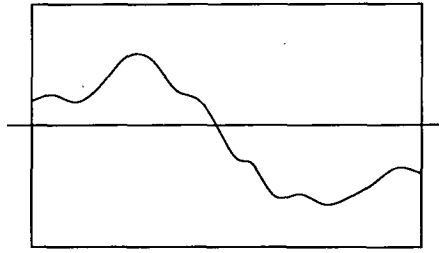


FIGURE 2.2.2. Arbitrary signal to be analyzed by the Fourier Transform. Since the signal is not periodic in nature it is not suitable for Fourier analysis.

One typical solution to this problem has been to simply force all functions to be periodic by assumption, thus for a function  $f(t)$  of finite length  $T$ , the function is periodized such that  $f(t + T) = f(t) : \forall t$ . This creates a guaranteed periodic signal of infinite extent which is then suitable for analysis by the Fourier Transform. Because this situation is artificial and the function is not normally periodic, the analysis of the signal will be somewhat corrupted by the periodizing process. One of the most common effects is that artificial sharp jumps in the signal function are created where  $f(T)$  cycles back to  $f(0)$ . This happens if the signal does not naturally return to some steady state value at the beginning and end of the supported time of length  $T$ , ie. if  $f(t) \neq f(T)$  which is the case for most short-time signals.

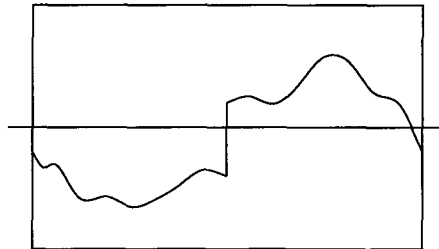


FIGURE 2.2.3. The artifact caused by periodization of the arbitrary signal above, for Fourier series analysis. This figure shows one period of the signal being analyzed. Here the origin where  $t = k \cdot T$  is in the centre of the figure. The sharp jump in signal value shown is due to the artificial process of making the signal periodic.

The power spectrum of such a corrupted signal will be spread out over a wide range of frequencies from lowest to highest, because all the frequency components are required to create the sharp edge in the periodic signal. Because of the spreading of the power spectrum this phenomenon is referred to as spectral leakage.

Continuous wavelet analysis uses basis functions of (generally) finite support length. A signal is transformed into a superposition of basis functions which are each different in scale and location. Because of the finite nature of the basis functions they are better able to represent localized variations in a signal than

Fourier analysis. For the analysis of periodic signals wavelets are unsuitable since their finite support means that an infinite set of wavelets at different locations from  $-\infty$  to  $\infty$  would be required.

### 2.3. Discrete Fourier Transform

With the advent of sampled signals and digital data it became necessary to develop a Discrete version of the Fourier transform for use on discrete data. For a discrete signal  $f(n)$  taken over a period of  $N$  samples

$$(2.3.1) \quad F(m) = DFT(f, m) = \sum_{k=0}^{N-1} f(k) \cdot e^{-j2\pi \cdot k \cdot m/N}, 0 \leq m < N$$

and the frequency measure  $m$  this time is also taken with respect to the sampling period. Equation 2.3.1 is independent of time and thus can be translated to any scale. If we know the period of time  $T$ , over which the  $N$  samples are taken then we can re-scale  $F(m)$  into  $F(\omega)$  by noting that  $\omega = m \cdot \frac{1}{T_s} = m \cdot f_s$  where  $T_s$  is the sampling period and  $f_s = \frac{1}{T_s}$  is the sampling frequency.

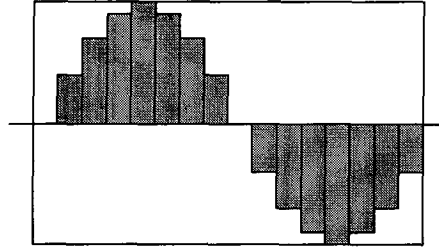


FIGURE 2.3.1. A sampled sinusoidal kernel function for the DFT. The maximum frequency which can be reproduced from the sampled data is determined by the Nyquist criterion as  $f_{max} = \frac{f_s}{2}$ .

It should be noted that a signal which is sampled of a period of  $N$  samples will in turn produce a DFT which is periodic over  $N$  samples and has a frequency resolution of  $\omega_{res} = 2\pi \cdot f_s = \frac{2\pi}{T_s}$  rad/sec. This power spectrum produced is periodic and infinite in extent. Due to the Nyquist sampling criterion the highest frequency which can be reproduced with a DFT is half that of the sampling frequency, ie.  $\frac{\omega_s}{2} = \frac{\pi}{T_s}$  rad/sec. Attempting to sample a signal with frequency components higher than the Nyquist frequency results in aliasing, due to the periodic nature of the power spectrum from the DFT. Signals must be passed through an analog low-pass filter prior to sampling to remove frequency components above the Nyquist frequency otherwise distortions occur.

The complimentary transform for the DFT, is the Inverse Discrete Fourier Transform (IDFT) and it is written

$$(2.3.2) \quad f(n) = IDFT(F(m)) = \sum_{m=0}^{N-1} F(m) \cdot e^{j2\pi \cdot n \cdot m/N}$$

and as with the forward DFT a DFT with a period of  $N$  points will create an  $N$  point discrete signal. The periodic nature of the DFT also means that the

reproduced signal is also technically infinite and periodic with a period of  $N$  samples. In this case the last sample in the set becomes the sample before the first sample. Because the forward DFT is taken as a finite sum over the sampling period it must assume that it samples one complete period of a periodic signal. For this reason the spectral leakage artifacts identified above for Fourier series again exist in this situation.

To keep within the scope of this thesis we will not further discuss problems of aliasing in this context as we will assume that the signals which we deal with have been successfully sampled. Indeed the topic of this thesis is digital images which have already been successfully acquired by some external method.

The term aliasing reappears later in the context of multichannel signal decompositions and multiplexing (particularly QMF structures) and has a different but related meaning.

**2.3.1. Fast Fourier Transform.** The process of performing a DFT on a signal is quite a computationally expensive exercise and because of this the procedure is seldom performed in this manner. In digital signal processing applications a much faster version of the DFT is used, called the Fast Fourier Transform (FFT)[13].<sup>2</sup>

The FFT gains its speed advantage by recursively splitting the signal into two equal sections and performing the DFT on each, then combining the results. This therefore limits the size of sampled signals on which the FFT can work to signals with  $2^j$  samples, where  $j \in \mathbb{Z} > 0$ . The cost of this small inconvenience provides a significant speed improvement over the normal DFT. The normal DFT is performed in order  $\mathcal{O}(N^2)$  operations, where  $N$  is the number of samples, whereas the FFT is performed in  $\mathcal{O}(N \cdot \log(N))$  operations. This becomes a significant gain as  $N$  becomes large.

Although a discussion of the method by which the FFT is performed is quite informative and can be useful in some situations, it is beyond the scope of this work to cover the topic in detail. This is left to the reader.

## 2.4. Discrete Cosine Transform

The Discrete Cosine Transform (DCT) is very similar to the DFT but it uses only cosine functions as transform kernels. Because of this, to preserve phase information, the kernel functions are multiples of half the fundamental frequency, thus creating both odd and even functions.

The DCT is defined as [28, chapter 3], [107]

$$(2.4.1) \quad C(u) = \alpha(u) \cdot \sum_{x=0}^{N-1} f(x) \cdot \cos\left(\frac{(2x+1)u\pi}{2N}\right)$$

---

<sup>2</sup>The FFT is generally credited to Cooley and Tukey [13] but traces of the algorithm extend back as far as 1924 [28, chapter 3, references]. A history of the FFT is set out in [12].

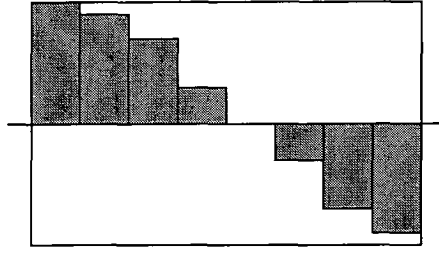


FIGURE 2.4.1. The lowest frequency DCT kernel function is only half a period long, making it an odd function. The basis functions for the Cosine Transform are spaced at multiples of half a period of the sample period.

for  $u = 0, 1, 2, \dots, N - 1$  and the inverse DCT is defined as

$$(2.4.2) \quad f(x) = \sum_{u=0}^{N-1} \alpha(u) C(u) \cos \left( \frac{(2x+1)u\pi}{2N} \right)$$

for  $x = 0, 1, 2, \dots, N - 1$ . In both Equations 2.4.1 and 2.4.2  $\alpha$  is a power normalization factor and

$$(2.4.3) \quad \alpha = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } u = 0 \\ \sqrt{\frac{2}{N}} & \text{otherwise} \end{cases}$$

The DCT has exceptional performance for image energy compaction. The main disadvantage of the DCT, like the Fourier transform is that it is a global transform. This means that every signal sample contributes to every transform coefficient and conversely every transform coefficient contributes to every signal sample. This causes problem when quantization of transform coefficients occurs because errors introduced are not localized within the signal rather they propagate throughout every pixel.

To avoid this globalization problem the JPEG standard [33] for image compression localizes the DCT to image blocks of 8x8 pixels. In this way, the coefficients of each block only affect that particular block and quantization does not affect other areas of the image. The blocking of the image, does create other problems for high compression ratios where heavy quantization is performed as noticeable discontinuities in signal value occur at the transition from one block to its neighbour.

## 2.5. Two-dimensional Fourier Transforms

So far we have only discussed the Fourier Transform of one-dimensional signals. To apply this method to images we need to generalize the method so that it can be applied to two-dimensional signals.

To move to a multidimensional context we must expand our definition of the Fourier Transform kernel function. In one dimension this is given by  $k(\omega, t) = e^{-j\omega t}$ . If we have a two-dimensional signal in  $x$  and  $y$  then we generalize the kernel

as  $k(\omega_x, \omega_y, x, y) = e^{-j2\pi(\omega_x \cdot x + \omega_y \cdot y)}$  which gives a sinusoid in  $x$  and  $y$  with different spatial frequencies in the  $x$  and  $y$  directions, ie.  $\omega_x$  and  $\omega_y$ .

The 2-D FT of  $f(x, y)$  is thus given by [28]

$$(2.5.1) \quad F(\omega_x, \omega_y) = \mathcal{F}(f(x, y), \omega_x, \omega_y)$$

$$(2.5.2) \quad = \int \int_{-\infty}^{+\infty} f(x, y) \cdot k(\omega_x, \omega_y, x, y) dx dy$$

$$(2.5.3) \quad = \int \int_{-\infty}^{+\infty} f(x, y) \cdot e^{-j2\pi(x \cdot \omega_x + y \cdot \omega_y)} dx dy$$

and thus  $F(\omega_x, \omega_y)$  is a function of two spatial frequencies, one in the  $x$  direction and one in the  $y$  direction. The complimentary transform returns  $f(x, y)$  by taking the inverse transform of  $F(\omega_x, \omega_y)$  as given by

$$(2.5.4) \quad \begin{aligned} f(x, y) &= \mathcal{F}^{-1}(F(\omega_x, \omega_y), x, y) \\ &= \int \int_{-\infty}^{+\infty} F(\omega_x, \omega_y) \cdot e^{+j2\pi(\omega_x \cdot x + \omega_y \cdot y)} d\omega_x d\omega_y \end{aligned}$$

in a similar manner to the one-dimensional case.

**2.5.1. DFT in 2D.** As with the one-dimensional case, we can discretize the Fourier Transform. In this case the DFT  $F(m, n)$  of a two-dimensional signal  $f(k, l)$  is given by

$$(2.5.5) \quad F(m, n) = \frac{1}{M \cdot N} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} f(k, l) \cdot e^{-j \cdot 2\pi \left( \frac{m \cdot k}{M} + \frac{n \cdot l}{N} \right)}$$

where the sampled signal and the resultant DFT is of size  $(M \times N)$  samples. The inverse transform which is the compliment of (2.5.5) is given by

$$(2.5.6) \quad f(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} F(m, n) \cdot e^{+j \cdot 2\pi \left( \frac{m \cdot x}{M} + \frac{n \cdot y}{N} \right)}$$

which generates the signal of size  $(M \times N)$  from the  $(M \times N)$  DFT.

As with the one-dimensional DFT, the 2-D DFT relies on artificially periodizing the sampled signal so as to perform analysis. In two-dimensions this is done by tiling the signal, ie. periodizing it in both dimensions separably. Hence  $f(x + i \cdot M, y + j \cdot N) = f(x, y) : \forall i, j \in \mathbb{Z}$ .

The periodization also occurs in the transform domain, so the DFT is periodic over a tile of size  $(M \times N)$ . For the usual case of rectangular sampling of the image data, the DFT periodic tiling repeats in the horizontal and vertical directions. Alternative sampling lattices lead to alternative periodization of the transform coefficients. This is covered in detail in [22] and will be revisited in Chapter 3.

**2.5.2. Separability and Tensor Products.** A property of the 2D DFT which has lead to it becoming popular in digital signal processing applications is its separable nature. Separability means that the signal can be split into components along each dimension which can be considered as separate components of

the overall signal. The concept of frequency in multiple dimensions is a vector describing the separable frequency components along each cardinal axis.

This allows a signal to be processed one dimension at a time using conventional one-dimensional DFT algorithms and kernels.

Consider (2.5.5). This can be re-written without loss of generality as

$$F(m, n) = \frac{1}{M} \sum_{x=0}^{M-1} e^{-j \cdot 2\pi \left(\frac{m \cdot x}{M}\right)} \left( \frac{1}{N} \sum_{y=0}^{N-1} f(x, y) \cdot e^{-j \cdot 2\pi \left(\frac{n \cdot y}{N}\right)} \right)$$

which puts no constraints on the order of operation. To process a signal in this way we first carry out the operation to produce the section in parenthesis ie.

$$F_{intermediate}(x, n) = \left( \frac{1}{N} \sum_{y=0}^{N-1} f(x, y) \cdot e^{-j \cdot 2\pi \left(\frac{n \cdot y}{N}\right)} \right)$$

which is a DFT of the  $x^{th}$  signal along one of the dimensions.

Note that the DFT kernel function used is a sinusoid dependent only on  $y$  and thus varies only along that dimension. This is equivalent to taking the  $x^{th}$  column of the image signal and producing a DFT of that column. Also note that the result of this operation is a columns of complex numbers, in the same way that the DFT of any signal results in a new sequence of coefficients of the same size.

To complete the operation we work with the other dimension

$$F(m, n) = \frac{1}{M} \sum_{x=0}^{M-1} e^{-j \cdot 2\pi \left(\frac{m \cdot x}{M}\right)} \cdot F_{intermediate}(x, n)$$

which then uses the numerical DFT values of all the columns and calculates the DFT of each row of the intermediate result. So the overall procedure consists of calculating the DFT in one dimension then calculating the DFT in the remaining dimension, of the DFT results from the first dimension.

The overall result is an  $M \times N$  'image' of DFT coefficients. An image which plots magnitude of the DFT as image intensity with coordinates derived from  $(m, n)$  forms a 2D power spectrum. The 2D analog to the Bode plot consists of two images, one which indicates spectral power by brightness at each frequency position and one which indicates phase angle by brightness at each frequency position. A 3D perspective plot can be more informative than an intensity plot.

The DCT can also be generalized to 2 dimensions in a similar manner as the DFT. It is again a separable transform given by

$$C(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cdot \cos\left(\frac{(2x+1)u\pi}{2N}\right) \cdot \cos\left(\frac{(2y+1)v\pi}{2N}\right)$$

and the inverse is

$$f(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \alpha(u)\alpha(v) \cdot C(u, v) \cdot \cos\left(\frac{(2x+1)u\pi}{2N}\right) \cdot \cos\left(\frac{(2y+1)v\pi}{2N}\right)$$

for  $x, y = 0, 1, 2, \dots, N-1$  and  $\alpha$  as in Equation 2.4.3.



## 2.6. Inner Products, Correlations and Convolutions

The correlation or inner product between two signals gives a numerical value describing how similar the two signals are. For this reason correlation can be used to determine how similar a signal is to some particular basis function of interest which allows for certain features of a signal to be easily identified and assigned a numerical value.

Correlation integrals, or inner products have the general form

$$(2.6.1) \quad \langle f(x), g(x) \rangle (n) = \int f(x) \cdot g(n+x) dx$$

for two continuous signals. In this case  $n$  provides a means of describing the value of the correlation integral as the origins of the two signals are moved apart. This generates a new function  $\langle f, g \rangle (n)$  which describes the degree of correlation of the two signals and varies with  $n$ . This function can be thought of as a projection operation for expressing one function as a projection onto the other.

Closely related to inner products are convolution integrals. A convolution of two continuous signals  $f(x)$  and  $g(x)$  is given by

$$(2.6.2) \quad [f(x) * g(x)](n) = \int f(x) \cdot g(n-x) dx$$

and again  $n$  gives a means of describing the effect of moving the origins of the signals with respect to each other. As can be seen from Equations 2.6.1 and 2.6.2 the two equations are almost exactly the same, the only difference being that for the convolution integral, the signal  $g(x)$  is reversed before the inner product is calculated. We can thus write

$$[f(x) * g(x)](\cdot) = \langle f(x), g(-x) \rangle (\cdot)$$

with rearrangements for the value of  $n$  - the origin displacement - to account for the reversal of direction.

The process of convolution is used to great extent for digital signal processing, indeed it is the single most important process. The process of convolution allows for real-time signal processing to occur. When a signal is passed through a filter the output from the filter is the convolution of the signal with the impulse response of the filter. There is a very strong connection between convolution and the Fourier transform. It can be shown that the process of convolution in the time domain is equivalent to multiplication in the Fourier domain, ie.

$$[f(x) * g(x)](\cdot) \iff F(\omega) \cdot G(\omega)$$

This applies in both the continuous and discrete domains and provides an essential tool for signal analysis and processing.

To use convolution with discrete data we must first discretize the convolution process. This is a simple matter of replacing the integral with a finite sum. Unlike

the Fourier Transform which relies on the signals being infinite in extent, convolution requires no such restriction and can be applied easily to signals of any finite length in both the continuous and discrete forms. A simple consideration of the process of integration using digital signals which are piece-wise constant gives us

$$[f(x) * g(x)](k) = \sum_i f(i) \cdot g(k - i)$$

for discrete convolution.

In this case the sum over  $i$  is over the range of values of  $i$  for which  $f(i)$  and  $g(k - i)$  are non-zero and overlap, termed the region of support. If the signals are infinite in length as can occur with infinite impulse response filters, then the resulting signal is also infinite in length. In general the result will have a maximum length of  $l_{result} = l_f + l_g - 1$  ie. the sum of the lengths of the two signals less one sample. This is easily verified if we note that there must be at least one sample of overlap of the non-zero signal values for the convolution sum to be non-zero. For the continuous case the length would be  $l_f + l_g$  as the minimum required overlap interval becomes zero size.

**2.6.1. Filtering.** Convolution is used in signal processing because it can be used to shape the frequency characteristics of a signal. Consider two functions  $f(x)$  and  $g(x)$ , we shall call  $f(x)$  the signal and  $g(x)$  the filter. Taking the convolution of the two signals gives us a function  $[f * g](n)$  as given by Equation 2.6.2.

Consider now the Fourier Transform of  $f(x)$ , written as  $F(\omega)$ , and of  $g(x)$  written as  $G(\omega)$ . These are given by

$$\begin{aligned} F(\omega) &= \int_{x=-\infty}^{+\infty} f(x) \cdot e^{-j\omega x} dx \\ G(\omega) &= \int_{x=-\infty}^{+\infty} g(x) \cdot e^{-j\omega x} dx \end{aligned}$$

and each gives the frequency characteristics of its signal in  $x$ . If we wish to augment or suppress certain frequencies within our signal  $F(\omega)$  then we create a new function with the desired characteristics, ie. a function in  $\omega$  with large value at frequencies which are to be augmented and a value close to zero for those frequencies to suppress. We can multiply (in a vector sense) the two spectra, ie. Fourier Transforms, together to obtain the required frequency response. Multiplication here means  $(F \cdot G)(\omega) = F(\omega)G(\omega)$ .

Let  $G(\omega)$  be such a function which shapes the spectrum of  $F(\omega)$ . The spectrum of the result, which we shall call  $Y(\omega)$  will have a value at frequency  $\omega$  which is the product of the Fourier Transform of its parent signal,  $F(\omega)$ , at  $\omega$  and the filter Fourier Transform,  $G(\omega)$  at  $\omega$ , ie.  $Y(\omega) = F(\omega)G(\omega)$ .

We now have the Fourier Transform of our filtered signal and if we determine the corresponding actual signal from the Fourier Transform using Equation 2.2.2

we have

$$\begin{aligned} y(x) &= \int_{\omega=-\infty}^{\omega=+\infty} Y(\omega) \cdot e^{j\omega x} d\omega \\ &= \int_{\omega=-\infty}^{\omega=+\infty} F(\omega)G(\omega) \cdot e^{j\omega x} d\omega \end{aligned}$$

which is the result given by the convolution integral of  $f(x)$  and  $g(x)$ .

$$\begin{aligned} y(\cdot) &= \int_{\omega=-\infty}^{\omega=+\infty} F(\omega)G(\omega) \cdot e^{j\omega x} d\omega \\ &= \int f(x)g(n-x) dx \end{aligned}$$

Thus if the Fourier Transform (spectrum) of  $g(x)$  shapes the spectrum of  $f(x)$  in the desired manner then the process can be performed by a process of convolution, thus eliminating the need to use the Fourier Transform altogether and allowing real-time processing to occur. The function for the filter  $g(x)$  can be calculated in the Fourier domain so that its spectrum  $G(\omega)$  has the required properties for shaping  $F(\omega)$ .

For discrete versions of the Fourier Transform and convolution the same results hold. In this case both the signal  $f(n)$  and the filter  $g(n)$  must have the same sampling density, ie. they must be sampled at the same spacing so that the samples line up.

**2.6.2. Spatial Domain Filtering (mask operators).** Another common means of manipulating the properties of an image is through the use of spatial masks. This process involves convolving the image with a small (usually 5x5 samples or smaller) filter image which is designed to perform a very simple task. One of the most common uses of this type of method is edge detection. Masks can be designed to be derivative operators and thus generate large values when the area of the image that they are convolved with has a significant rate of change of intensity with spatial distance.

The process of masking relates back to the Fourier Transform again since differentiation operators are high-pass filters.

## 2.7. Wavelet Basics

The term wavelets has come to be used to mean a set of functions derived by processes of dilation and scaling from a single mother wavelet function. Unlike Fourier analysis, where the bases are sinusoids, wavelets are mostly of finite length (compact support), and as a result they can show information about both frequency and location of features. The mother wavelet determines the overall shape of all those wavelets derived from it. Because of this all wavelets derived from a common mother wavelet will have similar properties but at different scales of application. They therefore allow us to analyze signals in a way which identifies traits which repeat in a self-similar manner at different scale levels. This is a

similar concept to fractals where the underlying self similarity of nature can be used to express the complexity of systems or images in terms of simple rules.

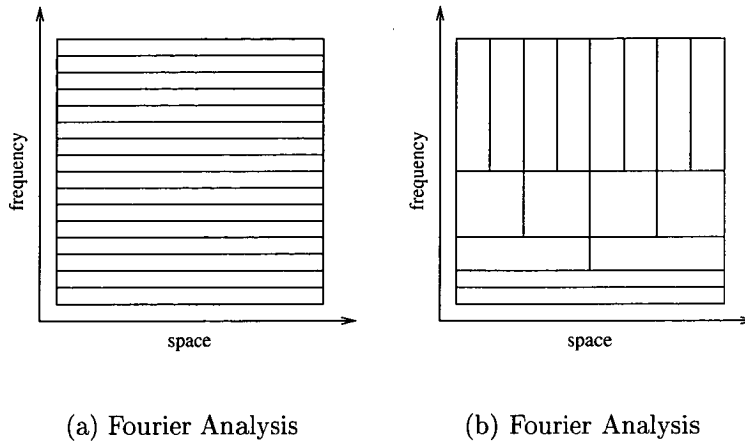


FIGURE 2.7.1. Tiling of the space-frequency domain by (a) Fourier analysis and (b) Wavelet analysis. Fourier analysis has a fixed resolution in both space and frequency - global in space and singular valued in frequency. Wavelets are localized in both space and frequency.

Figure 2.7.1 shows the tiling of the space-frequency domain by both Fourier and wavelet analysis. Fourier analysis uses basis functions which are singular in frequency and infinite in extent, hence they can only analyze one frequency. Often this is useful when exact frequency components are of concern. Wavelet bases are localized in space (compact support) and frequency and the two are related by an inverse relationship. The act of scaling a wavelet function to make it smaller in space increases its bandwidth and minimum frequency.

There are many different tilings of the space-frequency plane possible, some others are explored in [36], [37], [35] and [11].

Wavelets are an efficient means of expressing signals because of their multiresolution nature. A signal can be expressed first as a very coarse approximation at the lowest scale which provides an overall description and then at successively finer scales more details can be added to create increasingly better quality. This process can be stopped at any time when the quality of the signal is good enough for some particular purpose.

One reason that wavelets are good at expressing general signals is that the majority of the information that is needed to describe a signal goes into creating the broad overall description, ie. the lower frequency features. Once an overall shape of a signal is determined the rest of the information added is fine detail. Depending on how good the approximation is at coarse scales, the fine details can contain surprisingly little information. Indeed the wavelet coefficients for finer scales contain mostly sparsely spaced coefficients of small value which contribute little to the signal. Generally a good approximation to the original signal can be

obtained without using the higher resolution details at all, or in the case when there are significant fine details, only the most significant of the wavelet coefficients are needed.

To perform a single level wavelet transform, two parameters must be specified, one for scale and one for location. Since wavelets are finite in length, as the relative position between the origin of the wavelet basis changes, so too will the value of the wavelet transform. The scale parameter varies the shape of the wavelet transform kernel (basis function) so that it is conditioned for a particular feature size or frequency range.

For typical digital applications of wavelets, the actual wavelet filter itself is unchanged but the signal is resampled to different sampling densities which has the effect of applying the wavelet with a different scale. The dyadic wavelet is the simplest and most popular; it increases the length of support of the wavelet in the original sampling units by a factor of 2 for each decomposition level. Other resampling methods are possible for example rational sampling rates are explored in [57], [52], [51]

## 2.8. Scaling Functions and Wavelets

The wavelet transform is based on a system of successive approximations. Each successive level of analysis uses the approximation of the level before and takes a further approximation of that, and also produces a detail signal to reconstruct the previous signal.

We follow here the analysis given in [63].

If the signal is originally  $F$  and the approximation operator is  $H$  then the first approximation to  $F$  is given by  $A_1 = H \cdot F$ . We can write  $A_0 = F$  to demonstrate that  $F$  is the original approximation of the signal. Indeed if the signal was continuous then there would be no highest resolution level at which we could express it. The term  $A_0$  can be interpreted as the approximation due to sampling of a continuous signal.

The complimentary detail  $W_1$  is given by  $W_1 = G \cdot F$  where  $G$  is the detail operator which extracts a detail signal from  $A_{n-1}$  which is complimentary to  $A_n$  and allows  $A_{n-1}$  to be reconstructed.

Applying the approximation operation again, the second level approximation is given by

$$\begin{aligned} A_2 &= H \cdot A_1 \\ &= H \cdot H \cdot F \end{aligned}$$

and in general for the  $n^{th}$  level approximation  $A_n$  is given by

$$(2.8.1) \quad A_n = \left( \prod_{i=1}^n H \right) \cdot F$$

and thus  $H_n$  the operator to find the  $n^{th}$  level approximation by  $A_n = H_n \cdot F$  is given by  $H_n = (\prod_{i=1}^n H) = H^n$ .

To find the detail which compliments  $A_n$  we apply the detail operator  $G$  to  $A_{n-1}$  and thus

$$(2.8.2) \quad \begin{aligned} W_n &= G \cdot A_{n-1} \\ &= G \cdot \left( \prod_{i=1}^{n-1} H \right) \cdot F \end{aligned}$$

so the operator which extracts  $W_n$  from  $F$  is given by

$$G_n = G \cdot \left( \prod_{i=1}^{n-1} H \right)$$

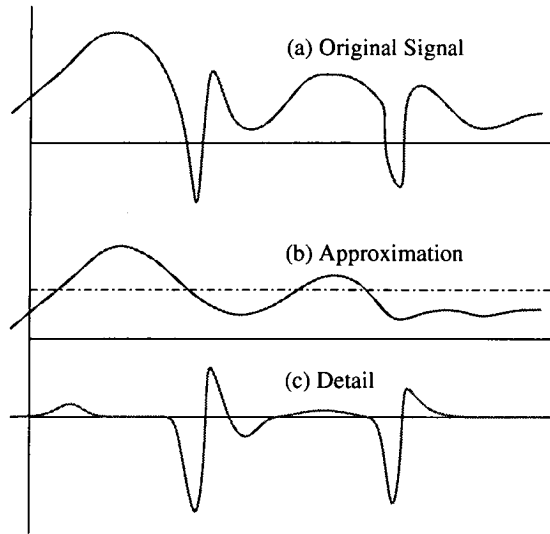


FIGURE 2.8.1. Example of approximation of a signal and associated detail from a scaling-function/wavelet pair. The scaling function captures the low-frequency components, including average, and the wavelet signal captures the high-pass components.

Figure 2.8.1 shows an example of the first level approximation and detail signal of some arbitrary signal. The approximation preserves the low-pass energy but loses all fine detail. The fine detail is preserved in the detail which is a high-pass signal (ie. average = 0).

## 2.9. Continuous Wavelet Transform

A Wavelet Transform is achieved by taking the inner product (and thus the convolution) of some signal of interest with the appropriate wavelet basis function or transform kernel (with a particular scale and location) [17],[63]. For example

$$(2.9.1) \quad \mathcal{W}(f(x)) = \langle \psi_{a,b}(x), f(x) \rangle$$

defines a wavelet transform where  $\psi_{a,b}(x)$  is the wavelet transform kernel function which has the correct scale and position.



The Wavelet functions themselves are (in general) derived from a single mother function by a combination of *dilation* (change of scale) and *translation* (displacement in space).

Let the mother function for a wavelet transform be  $\psi(x)$  then we derive the associated function which is scaled by factor  $a$  and translated by  $b$ , as [31], [18]

$$(2.9.2) \quad \psi_{a,b}(x) = \frac{1}{\sqrt{a}} \psi\left(\frac{x-b}{a}\right)$$

where  $a, b \in R, a \neq 0$ .

The factor of  $\frac{1}{\sqrt{a}}$  normalizes the functions to have a constant  $L_2$  norm which prevents rescaling of signal energy.

Now using Equations 2.9.1 and 2.9.2 we get

$$\begin{aligned} \mathcal{W}(f(x), a, b) &= \langle \psi_{a,b}(x), f(x) \rangle \\ &= \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(x) \psi^*\left(\frac{x-b}{a}\right) dx \end{aligned}$$

which gives us the Wavelet Transform of  $f(x)$  with the wavelet  $\psi_{a,b}$  of scale  $a$  and translation  $b$ .

For a transform to be useful for image coding it must be invertible so that the original signal, or one like it, can be reconstructed from the transform coefficients. Although Equations 2.9.1 and 2.9.2 will give results for any wavelet kernel  $\psi$ , for the transform to be invertible the kernel must satisfy certain constraints as given by Equation 2.9.5.

The inverse transform pair for Equation 2.9.2 that recreates the original signal from the wavelet transform can be expressed as

$$(2.9.3) \quad f(x) = \frac{1}{C} \int_0^{+\infty} \int_{-\infty}^{+\infty} \frac{1}{\sqrt{a}} \mathcal{W}(f(x), a, b) \psi\left(\frac{x-b}{a}\right) \frac{1}{a^2} \cdot da \cdot db$$

where

$$(2.9.4) \quad C = \int_{-\infty}^{+\infty} \frac{\hat{\psi}^*(\omega) \hat{\psi}(\omega)}{\omega} d\omega$$

obviously the inverse transform exists only if

$$(2.9.5) \quad 0 \leq C < \infty$$

hence this is termed the *admissibility criteria* for the wavelet  $\psi(x)$ . An immediate and obvious implication of this is that the wavelet  $\psi(\cdot)$  must be a high-pass function ie.  $\hat{\psi}(0) = 0$ . Here  $\hat{\psi}(\omega)$  is the Fourier Transform of  $\psi(x)$ .

Figure 2.9.1 shows the band-splitting nature of the wavelet transform. In this case the dyadic wavelet is used and the original signal is split into two octaves with the cutoff frequency at the half of the previous bandwidth.

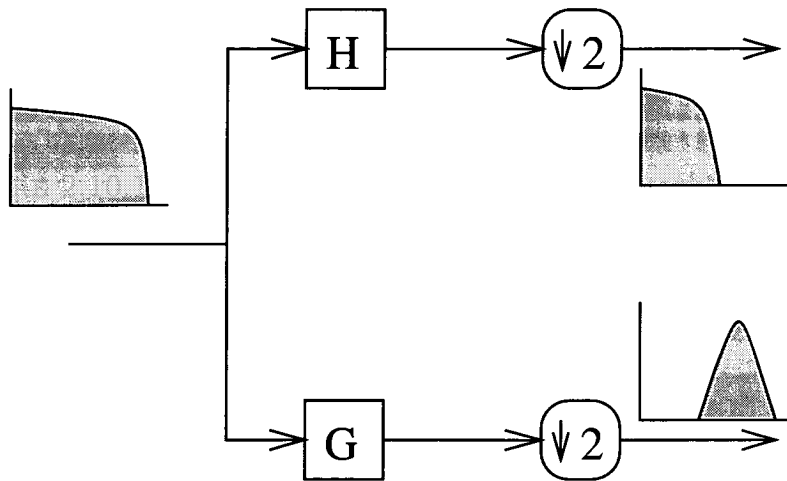


FIGURE 2.9.1. Band-splitting property of the dyadic wavelet transform. The approximation (H) retains the low-frequency components while the detail signal (G) contains the rest of the bandwidth of the original signal. Because of previous band-limiting approximations, the high-frequency component is actually a narrow band.

## 2.10. Multiresolution Analysis

**2.10.1. Function Spaces.** Some knowledge of function spaces is necessary to gain an understanding of the way in which multiresolution analysis and the discrete wavelet transform operates.

The underlying concept behind function spaces is that in general a function can be expressed as the sum of a set of other basis functions. Any function which can be represented in such a way is said to reside in a the function space for which the simple functions are the basis functions.

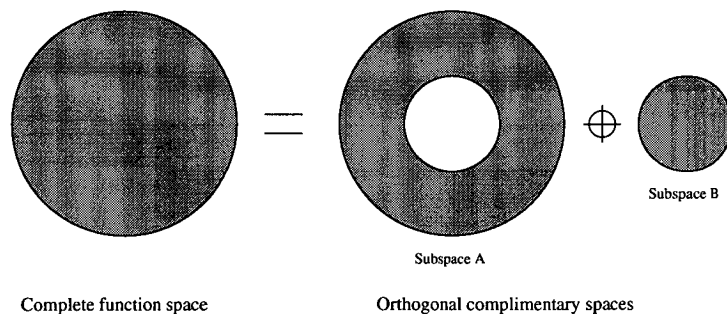


FIGURE 2.10.1. Conceptual diagram of complimentary orthogonal function spaces. With DC at the centre, the function space on the right represents an approximation sub space of the original function space. The complimentary space represents those functional capabilities not present in the approximation subspace.

### 2.10.1.1. *span of a basis function set.*

The span of a function set represents all possible functions which can exist in the function space. A basis for a function space is a set of functions which, when used in linear combinations, provide all the possible characteristics which can occur within the function

space. Any function which is a member of the function space can be represented as a linear combination of the basis functions translated as necessary.

2.10.1.2. *order of continuity and smoothness :  $C^n$ .*

A function is said to be  $C^n$  smooth if all derivatives after the  $n^{th}$  are 0. Obviously the first derivative cannot be 0 as this would mean the function was a constant value everywhere. Being  $C^n$  means that a function can be represented by a polynomial of degree  $n$ . Thus all functions  $f(\cdot) \in C^n$  are in the function space spanned by  $\{1, x, x^2, \dots, x^n\}$ .

**2.10.2. Multiresolution Analysis.** Multiresolution analysis was first described by Mallat in [63]. A multiresolution analysis (MRA) is a series of approximation subspaces

$$(2.10.1) \quad \dots \subset V_{n+2} \subset V_{n+1} \subset V_n \subset V_{n-1} \subset \dots$$

where each next subspace in the hierarchy is at a higher resolution level than the previous. If the difference in resolution between levels is a factor of 2 then  $f(x) \in V_n \iff f(2x) \in V_{n+1}$ . This implies that  $f(x)$  is band-limited to  $V_n$  to start with. If  $f(x)$  is not band-limited then this decomposition breaks down because  $f(x)$  cannot be confined to any function-space. Obviously for sampled signals we have an upper limit to the signal spectrum as defined by its sampling.

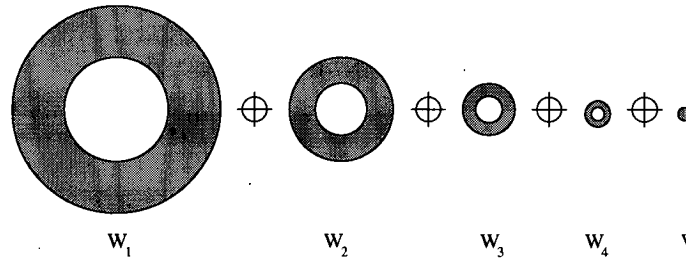


FIGURE 2.10.2. Conceptual diagram of nested subspaces show a Multiresolution Analysis. Here the many detail subspaces  $W_n$  and the single low-pass subspace  $V_N$  sum together to produce the original.

If  $f(x) \in V_n$  then the projection of  $f(x)$  onto a subspace is obtained by applying a suitable projection operator  $P$  to  $f(x)$ . The projection operator reduces the information contained in the result  $P \cdot f(x) \in V_{n+1}$  when compared to  $f(x)$  and so  $f(x) \in V_n$  cannot be recovered from it. The information which is necessary to reconstruct  $f(x)$  from  $P \cdot f(x)$  is contained in the orthogonal complement of  $V_{n+1}$  which we shall call  $W_{n+1}$ . Thus in terms of subspaces  $V_{n+1} + W_{n+1} = V_n$ .

The projection of  $f(x)$  onto  $W_{n+1}$  can be obtained from  $f(x) \in V_n$  by the use of another suitable projection operator  $Q$ .

So we have that  $P \cdot f(x) + Q \cdot f(x) \implies f(x)$ .

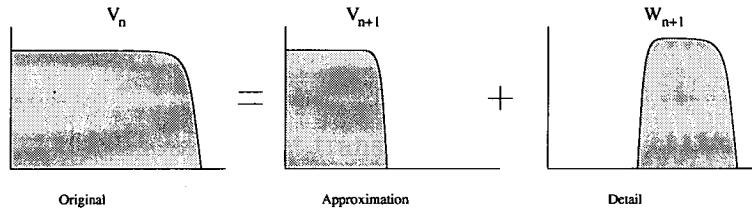


FIGURE 2.10.3. Power spectra showing band-splitting nature of the wavelet transform. The original bandwidth of a signal is split into two equal octaves; one low-pass, the other high-pass.

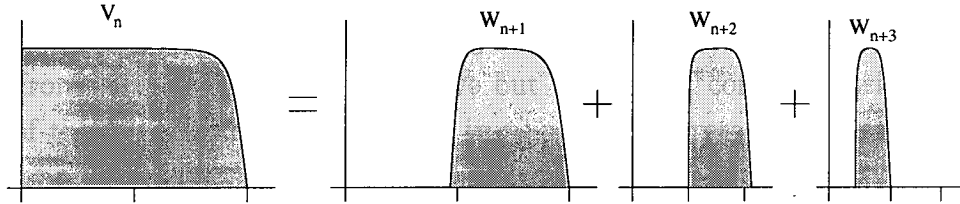


FIGURE 2.10.4. Power spectra in multiresolution analysis. This shows how the low-pass band of each level of band-splitting is split further. Each level of decomposition reduces the bandwidth by a factor of 2 (octaves).

Multiresolution analysis is a term used to describe the analysis of a signal into components at different levels of scale or resolution. In terms of a digital signal multiresolution analysis is a process of performing a transform and then resampling to a lower level - repeating for each change in resolution level. The essential property being that each transform step results in a low-pass version of the signal which can then be further decomposed. The loss of information which occurs as a result the signal being sampled to a lower resolution is retained in other subbands. Thus the process is reversible: the original signal can be reconstructed from the end low-pass signal and the detail signals via a process of upsampling and an inverse transform.

## 2.11. Discrete Wavelet Transform

Whereas the continuous wavelet transform is implemented as a continuous convolution/correlation integral, there is no need to impose such an implementation on the discrete wavelet transform. We can define the discrete wavelet transform in a manner analogous to 2.9.1 as

$$(2.11.1) \quad \mathcal{W}(f(n)) = \sum_{k \in -\infty}^{\infty} \psi_{a,b}(k) \cdot f(n+k)$$

This will provide us with the detail signal from the wavelet  $\psi_{a,b}$ . This can be useful from a signal analysis perspective but we are not concerned with it in the scope of this thesis. We are interested in using the discrete wavelet transform as a perfect reconstruction transform with which we can efficiently code image data.

From this perspective it is the multiresolution interpretation of the DWT that interests us.

The wavelet transform for data compression is implemented as a multiresolution analysis. We begin with the signal at its highest resolution  $A_0$  and decompose it into an approximation  $W_1$  and a detail signal  $V_1$  at the next lower resolution level.  $W_1$  corresponds to  $A_1$ . We apply the same procedure again this time starting with the approximation from the previous decomposition  $A_1 = W_1$ . The process of decomposition involves filtering the approximation signal  $A_n$  with a low-pass and high-pass filter and then downsampling the results. The downsampled low-pass signal is  $W_{n+1}$  and the downsampled high-pass signal is  $V_{n+1}$ .

This process continues until some desired level of decomposition is achieved - usually denoted by the size of approximation signal; or the signal is too small to convolve with the filters.

The final result is a single low resolution approximation signal  $W_N$  and a set of detail signals  $\{V_N, V_{N-1}, \dots, V_1\}$  which can be used to reconstruct the original image from the approximation.

The most common form of the wavelet transform involves downsampling by a factor of 2 after filtering. This is the so called dyadic wavelet decomposition.

**2.11.1. Orthonormal compact wavelets.** We briefly mention probably the most famous family of wavelets; the Daubechies family as described in [17]. These were the first smooth wavelets discovered which were orthonormal and with compact support in the limit.

The low pass of the Daubechies  $D_4$  wavelet is given by [78]

$$h_k = \frac{1 + \sqrt{3}}{4\sqrt{2}}, \frac{3 + \sqrt{3}}{4\sqrt{2}}, \frac{3 - \sqrt{3}}{4\sqrt{2}}, \frac{1 - \sqrt{3}}{4\sqrt{2}}$$

with a high-pass determined by the normal rules of QMF, such that

$$g_k = \frac{1 - \sqrt{3}}{4\sqrt{2}}, -\left(\frac{3 - \sqrt{3}}{4\sqrt{2}}\right), \frac{3 + \sqrt{3}}{4\sqrt{2}}, -\left(\frac{1 + \sqrt{3}}{4\sqrt{2}}\right)$$

The  $D_4$  filters provide minimum support for 2 vanishing moments. The family contains filters of increasing size which correspond to solutions with 3, 4, etc. vanishing moments.

We use the Daubechies  $D_4$  wavelet for the purpose of comparison with non-separable transforms later as the complexity of the  $D_4$  filters is similar to those developed for nonseparable sampling.

One modification was required for the Daubechies filter for parts of the research. The modification refers to the normalization factor used in the Daubechies filter. Examining the filters we find that  $\sum h_k = \sqrt{2}$  so the filters are not completely normalized. The reason for this is that filters used in a subband decimation system must have a normalization to account for the act of upsampling. Downsampling removes coefficients completely and no values are altered, however

upsampling occurs before filtering during reconstruction. This upsampling introduces coefficients of value zero and the result when convolving a sequence where every other value is zero is a result of half the magnitude if all coefficients were retained. So the normalization of  $\sqrt{2}$  on the decomposition and reconstruction leads to a total normalization of 2 which counters the effects of upsampling.

This is a proper design criterion but it has a very serious disadvantage - each time the approximation signal is decomposed its mean value increases by a factor  $\sqrt{2}$ . This isn't a problem if all we want to do is decompose and reconstruct, however often we wish to examine the relationships between coefficient values at different resolution levels. To allow this to occur, the normalization of the filters is sometimes rearranged so that the decomposition filters are normalized to 1 but the reconstruction filters are normalized to 2. An alternative method is to normalize all filters to 1 and multiply the reconstruction result by a factor of 2.

## 2.12. Filters and Subbands

In this section we cover in more detail some of the methods used to implement the discrete wavelet transform in terms of critically sampled filterbanks and subband coding. Each step of the multiresolution decomposition is a matrix transform on a signal vector implemented as a filterbank as shown in Figure 2.12.2. The most common case is the two-channel case which corresponds to the dyadic discrete wavelet transform. The two-channel case involves two branches in the filterbank (low-pass and bandpass) and a resampling factor of 2.

We cover orthogonal, biorthogonal and quadrature-mirror filters (QMFs) in this section focusing on the properties and restriction of each design.

An excellent overview of the use of multiscale decomposition and subband coding for the analysis of images and other data can be found in [24].

Figure 2.12.1 dramatically shows the multiresolution concept at work. Each of the images is a projection of the original image onto a different low-pass space. They show the effect of adding the detail signals to the approximation signal which starts with very little definition and finishes with very good signal quality. This shows how progressive image transmission with multiresolution coding would occur. With an original image of size 512x512 pixels the first two detail levels image (c) could be reconstructed from only 256 coefficients - which could be in the vicinity of one byte per coefficient (1000:1 compression) for an integer to integer transform. All images are reconstructed to their original size through a complete inverse wavelet transform, but where the higher resolution detail signals are absent (zero contribution).

**2.12.1. Orthogonal Filters.** Orthogonality is a term borrowed from linear algebra. It takes meaning in the context of subband coding because the act of transforming a signal through a filter bank is equivalent to a matrix multiplication of the signal with a transform matrix.



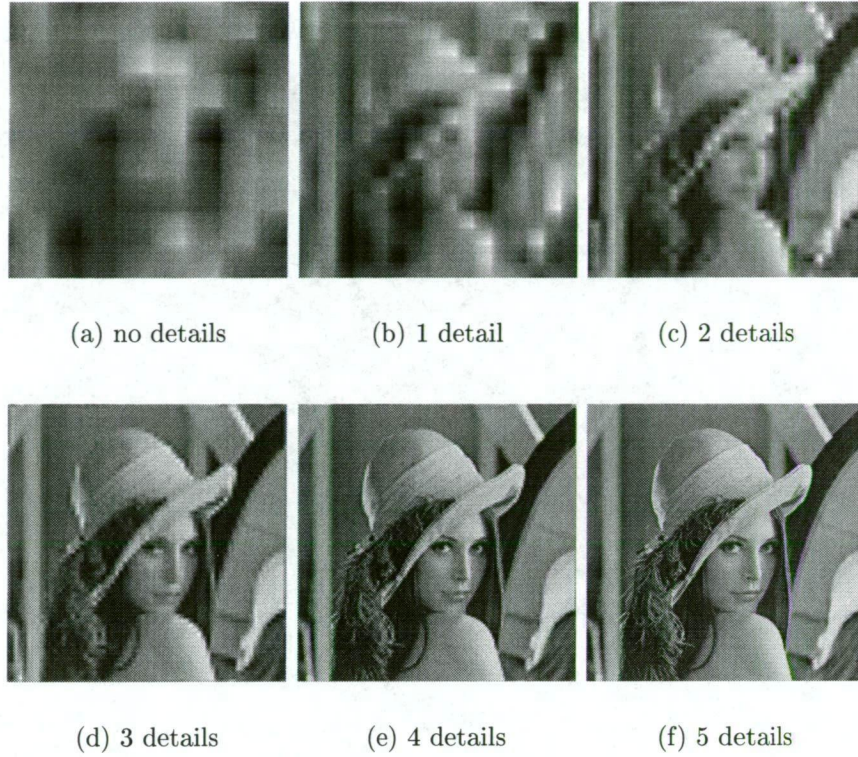


FIGURE 2.12.1. Lenna image reconstructed from the lowest resolution details of a 6-level separable wavelet decomposition using Daubechies  $D_4$  [17] filters. (a) Only the low-pass at level 6, (b) Low-pass plus detail at level 6. (c) Low-pass plus two details - levels 6 and 5. (d) Three details. (e) Four details. (f) Missing only the highest resolution detail.

$$y = M \cdot x$$

Here  $M$  is a square matrix of size  $l \times l$  and  $x$  and  $y$  are signal vectors of size  $l$ . For such a transform to be reversible, the transform matrix must be invertible, ie.  $M^{-1}$  must exist. If it does then the original signal can be reconstructed by

$$\hat{x} = M^t \cdot y$$

The transform matrix  $M$  is orthogonal if [84]

$$(2.12.1) \quad M \cdot M^t = M^t \cdot M = I$$

In terms of linear filtering each of the columns of the transform matrix represents a basis in the transform ie. a filter at a particular location with respect to the origin of the signal. More than one filter is included in the matrix and all filters occur in all possible positions to account for the action of convolution. Equations 2.12.5 and 2.12.6 give an example of this. If the transform matrix is orthogonal then the inner product of any column with any other is zero, and the inner product of a column with its self is unity.

A set of functions  $\{f_n(\cdot)\}$  is orthogonal if those functions represent the bases of an orthogonal matrix, ie.

$$(2.12.2) \quad \langle f_n(\cdot), f_m(\cdot) \rangle = \delta(m - n)$$

where  $\delta(\cdot)$  is a Dirac delta function, ie.  $\delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases}$ .

In this situation the function set  $\{f_n(\cdot)\}$  can be all the translates of a filter used in convolution, so not only are the filters required to be orthogonal to each other but also to the integer translates of themselves.

For example  $f_n(k) = g(k)$  and  $f_m(k) = g(k - 1)$  are both columns in a filterbank matrix containing the filter  $g(\cdot)$ . For the matrix to be orthogonal we must have  $\langle f_n(k), f_m(k) \rangle = 0$  ie.  $\langle g(k), g(k - 1) \rangle = 0$ , the two translates of  $g(k)$  are orthogonal to each other. This is one of the key restriction in the design of filterbanks for perfect reconstruction and hence wavelets.

Equation 2.12.2 leads to

$$\langle \langle f_m(\cdot), g(\cdot) \rangle, f_n(\cdot) \rangle = g(\cdot) \cdot \delta(m - n)$$

ie. any function  $g$  in the function-space  $F$  spanned by  $\{f_n\}$  can be decomposed and reconstructed using  $\{f_n(\cdot)\}$ .

Orthogonality is an important property for a signal transform to have. Although perfect reconstruction can be obtained without orthogonality, orthogonality minimizes the error introduced by quantization of coefficients. This is essentially because it minimizes the interaction between transform coefficients.

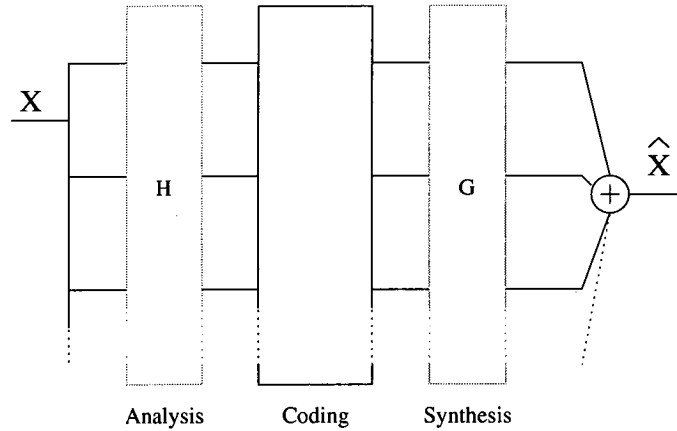


FIGURE 2.12.2. Block diagram of a general filterbank system. There are  $n$  channels in each filterbank and each channel performs shifting, filtering and resampling as required. Analysis involves shifting followed by filtering then downsampling. Synthesis involves upsampling, followed by filtering then reverting the shift.

If we consider the filterbank representation of a wavelet transform as in Figure 2.12.2, the forward transform is given by [84]

$$(2.12.3) \quad y = H^t \cdot x$$

and the inverse transform is given by

$$(2.12.4) \quad \hat{x} = G \cdot y$$

where  $H$  and  $G$  are filterbank matrices for the forward and inverse transforms. In the two channel one dimensional case we have for example

$$(2.12.5) \quad H = \begin{pmatrix} h_0(0) & h_0(k_0) & h_1(0) & h_1(k_1) \\ h_0(-1) & h_0(k_0 - 1) & h_1(-1) & h_1(k_1 - 1) \\ h_0(-2) & h_0(k_0 - 2) & h_1(-2) & h_1(k_1 - 2) \\ \vdots & h_0(k_0 - 3) & \ddots & h_1(k_1 - 3) & \cdots \\ & h_0(k_0 - 4) & h_1(2) & h_1(k_1 - 4) \\ h_0(2) & \vdots & h_1(1) & \vdots & \ddots \\ h_0(1) & & & & \end{pmatrix}$$

and

$$(2.12.6) \quad G = \begin{pmatrix} g_0(0) & g_0(k_0) & g_1(0) & g_1(k_1) \\ g_0(1) & g_0(k_0 + 1) & g_1(1) & g_1(k_1 + 1) \\ g_0(2) & g_0(k_0 + 2) & g_1(2) & g_1(k_1 + 2) \\ \vdots & g_0(k_0 + 3) & \ddots & g_1(k_1 + 3) & \cdots \\ & g_0(k_0 + 4) & g_1(-2) & g_1(k_1 + 4) \\ g_0(-2) & \vdots & g_1(-1) & \vdots & \ddots \\ g_0(-1) & & & & \end{pmatrix}$$

where the columns are all the possible translates of all the filters in the filterbank. In this case only two filters,  $h_0$  and  $h_1$  in the analysis bank and  $g_0$  and  $g_1$  in the synthesis bank, are explicitly shown.

Combining Equations 2.12.3 and 2.12.4 we see that for analysis followed by synthesis

$$(2.12.7) \quad \hat{x} = G \cdot H^t \cdot x$$

and for perfect reconstruction we require  $\hat{x} = x$  which leads to

$$(2.12.8) \quad G \cdot H^t = I$$

For critically sampled systems we have that  $G$  and  $H$  are square matrices of the same size and we can choose  $G = (H^{-1})^t$ . This particular solution allows us to reverse the order of transform, ie. let the analysis bank become the synthesis bank and vice verse.

For orthogonal systems in terms of transform matrices we now have (from Equation 2.12.1)

$$G = H$$

ie. the synthesis filters are inverted version of the analysis filters and the filters must satisfy

$$g_i(n) = h_i(-n)$$

ie. the synthesis filter is the same as the analysis filter, but reversed.

Orthogonal transforms are characterized by the fact that the same kernel function is used in both the forward and inverse transforms but with conjugation in the frequency domain which is equivalent to reversal in the spatial or time domain. The Fourier Transform is an example of an orthogonal transform kernel, and there are many orthogonal wavelets. Most famous among the orthogonal wavelets are the Daubechies series [17] of compactly supported orthonormal wavelets with  $n$  vanishing moments.

**2.12.2. Biorthogonal Filters.** Orthogonality is a key property in transform coding but it is also very restrictive. By moving to biorthogonality we can alleviate some of these restrictions while maintaining many of the essential advantages of orthogonality. One of the most important restrictions of orthogonality in terms of image processing is the mutual exclusivity of orthogonality and linear-phase. While orthogonality minimizes the interaction of coefficients and thus minimizes the error due to quantization, linear phase preserves the alignment of components in disparate frequency bands after quantization. This is essential for high-performance image compression because perceptually edges and lines are the most important features of an image. Although statistically orthogonal filters will result in a lower mean square error rate, the perceived quality will be less than for linear-phase filters.

Biorthogonality uses two different transform kernels for the forward and inverse transforms. If this pair of kernels is properly constructed then it is possible to have perfect reconstruction, ie.  $f(x) = \mathcal{F}^{-1}(\mathcal{F}(f(x)))$ , with  $\mathcal{F}()$  and  $\mathcal{F}^{-1}()$  using different transform kernels.

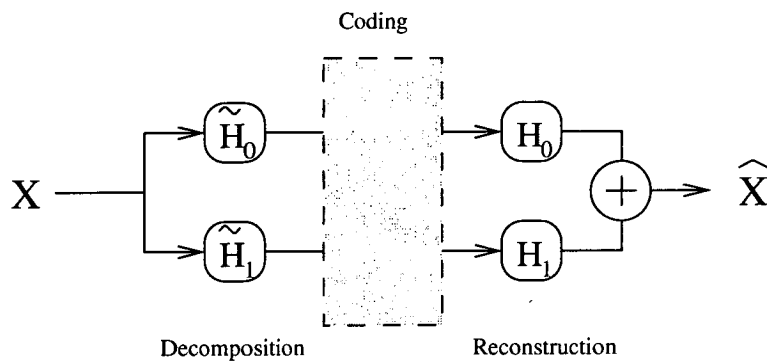


FIGURE 2.12.3. Block diagram of the biorthogonal wavelet transform. Perfect reconstruction is possible without the constraints of orthogonality. Here the columns of the matrix  $\tilde{H}$  and  $H$  are orthogonal to one another.

In terms of matrix operations we have

$$y = \tilde{H}^t \cdot x$$

and

$$\hat{x} = H \cdot y$$

For perfect reconstruction we require that  $\hat{x} = x$  or  $H \cdot \tilde{H}^t = I$ .

For orthogonality we would require that the columns of  $H$  be orthogonal to each other. For biorthogonality we require that the columns of  $\tilde{H}$  are orthogonal to the columns of  $H$ . So if  $\tilde{H}$  is made up of  $\{\tilde{f}_n\}$  and  $H$  is made up of  $\{f_n\}$  then biorthogonality is expressed as

$$(2.12.9) \quad \langle \tilde{f}_n, f_m \rangle = \delta(m - n)$$

We can also write

$$(2.12.10) \quad \langle \langle f_m(\cdot), g(\cdot) \rangle, \tilde{f}_n(\cdot) \rangle = g(\cdot) \cdot \delta(m - n)$$

In this case  $f_m(\cdot)$  and  $\tilde{f}_m(\cdot)$  are different function sets which span the same function space. Equation 2.12.2 reduces to 2.12.9 when we make  $g(\cdot) = \delta(\cdot)$  ie. Equation 2.12.10 becomes

$$\langle \langle f_m(\cdot), \delta(\cdot) \rangle, \tilde{f}_n(\cdot) \rangle = \delta(m - n)$$

or

$$\langle f_m(\cdot), \tilde{f}_n(\cdot) \rangle = \delta(m - n)$$

Biorthogonality reduces to orthogonality if  $\tilde{f}_n = f_n$ , ie when the same matrix is used in analysis and synthesis. For this reason biorthogonality can be thought of as a generalization of orthogonality.

Biorthogonality allows different analysis and synthesis filters to be used to obtain perfect reconstruction. Perfect reconstruction is obtained not with one filterbank but with a pair of filterbanks designed to work together.

### 2.13. Wavelet Packet Transform

The Wavelet Packet Transform (WPT) was first proposed by Wickerhauser [38], [115], [113]. At the heart of this transform is the same method as for the DWT, ie. discrete convolution of signal with subband filters. With the DWT only the low-pass subsampled signal is further decomposed into subsequent low and high-pass bands. With the WPT both low-pass and high-pass signals are further decomposed using the same subband filters. This results in a far larger number of subspaces all at one particular level of decomposition. The subspaces are no-longer octave subbands but all are of the same bandwidth, equally distributed over the original signal spectrum. Again critical sampling ensures that the total number of samples is retained so the information cost is the same as for the normal wavelet transform.

Best-basis construction [113], [114], [77] uses the structure of the wavelet-packet decomposition to find the best basis for a particular signal. The general idea being that at each decomposition step the information cost for coding a



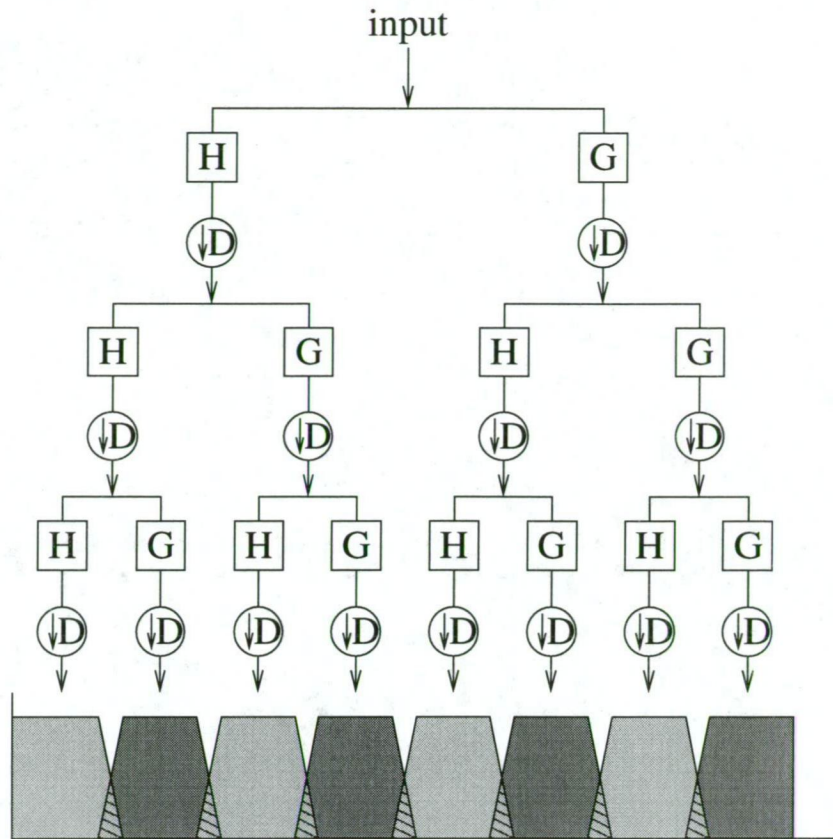


FIGURE 2.13.1. The Wickerhauser Wavelet-Packet Decomposition in one dimension. In this decomposition, both the low and high-pass components are decomposed at each level, resulting in a set of components of equal bandwidth and equal spacing across the spatial frequency domain. Here  $D$  is used to represent generalized dyadic downsampling in  $n$ -dimensions.

subband is compared to the information cost of coding the two smaller subbands at the next level of decomposition (and possibly combinations at deeper levels). The solution with the lowest rate versus distortion cost is chosen. This method results in a decomposition where the decomposition paths are truncated at various points to optimize the rate-distortion trade-off.

In [10] and related works, the idea is expanded to the use of Wavelets, Wavelet packets, Fractals and Fourier analysis to find a best possible dictionary of image primitive functions and decomposition methods.

## 2.14. Tensor Product: Separable Wavelets in Multiple Dimensions

In much the same way as we can determine the Fourier Transform of a multidimensional signal by separably considering one dimension at a time, we can determine the separable wavelet transform of an image. Considering each dimension separably reduces the problem to a one-dimensional subband filtering problem which we can solve quite easily with one-dimensional wavelets and filterbanks.



The separable wavelet transform is performed by considering first each column of the image as a separate signal. Each column is then filtered to produce a low-pass approximation and a detail signal. So the image signal now consists of two signals a vertical low-pass and a vertical high-pass, both resampled to a lower resolution level. If we consider the rows of these result signals we can then apply the wavelet transform to each row of each signal. So the vertical low-pass is passed through a horizontal low-pass and a horizontal high-pass. Obviously the low-pass low-pass combination will result in an overall approximation, which the low-pass high-pass combination will isolate a signal with horizontal high-pass and vertical low-pass, ie. vertical edges.

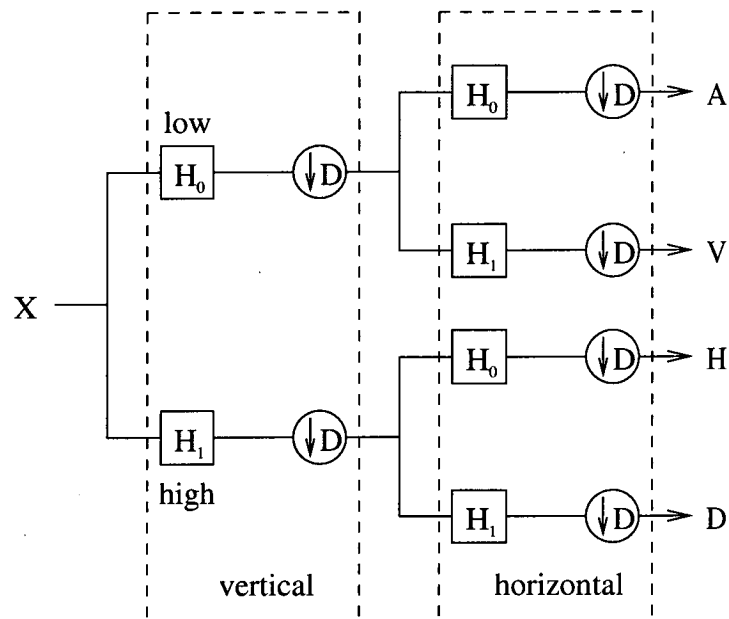


FIGURE 2.14.1. The Mallat pyramid structure for 2-D separable wavelet decompositions.  $A$  is the approximation of the original signal,  $V$  contains vertical edges,  $H$  contains horizontal edges and  $D$  contains diagonal edges. All subspaces contain  $\frac{N}{4}$  samples if  $X$  contains  $N$

Similarly the vertical high-pass signal is passed through horizontal low-pass and high-pass filters, resulting in two signal, one with horizontal edges (vertical high-pass and horizontal low-pass) and one with only diagonal edges (high-pass in both dimensions).

After each filtering process the signals are decimated and following the discrete wavelet pattern, the sampling is critical so the number of coefficients after sampling is the same as before. Each subband contains one quarter the number of the previous image signal. This technique is the famous Mallat Pyramid [63] for multiresolution analysis of images.

An excellent introduction to separable wavelet image coding can be found in [39] and a well-rounded text discussing the processing of image data in a subband context is [119]. A good tutorial on the subject is [29].

### 2.15. Nonseparable Wavelets: True Multidimensional Transforms

Because images are two-dimensional signals it makes sense that we should consider them with a transform that lies in their native domain, ie. a truly two-dimensional transform. To do this we need both 2-D sampling patterns and 2-D wavelets which are constructed to operate on the sampling pattern chosen.

The approximation and detail results given in equations 2.8.1 and 2.8.2 are general and are independent of the sampling method used to decimate the signal.

The wavelets transform as used in digital imaging consists of filtering and resampling. It is possible for the filtering and resampling to be done in a completely non-separable fashion.  $H$  and  $G$  from equations 2.8.1 and 2.8.2 are combined filter and re-sample operators. Thus  $A_1 = H \cdot F$  is the result of filtering by filter  $F(\omega)$  and resampling. We dedicate Chapter 3 to covering non-separable wavelets and resampling on non-separable lattices in detail.

## CHAPTER 3

### The Quincunx Lattice

To date the bulk of research into wavelet and subband coding of signals has been done in the one-dimensional domain. This builds on the knowledge of generations of research into analog and digital filtering and transmission. Digital image processing is a relatively new science and the facilitation of research has been largely through the ability to treat  $n$ -dimensional signals separably in the one-dimensional domain thus using the knowledge and results already obtained. This path leads to the separable results which have found a multitude of application in data processing. The proliferation of raster scan devices increases the application of such results as the scan order matches closely the separable model.

When dealing with image processing for human consumption we need to take into account the properties of the human visual system. One of the significant properties of the HVS is that the receptor cells in our retina are arranged in a close hexagonal packing structure similar to honeycomb and are not rectangularly packed. This leads to a disparity between the arrangement of pixels we typically see on display devices and how the data is received in our visual system.

Hexagonal sampling of data has been the topic of some strong research in the field [83], [111], [110], [62] which sheds light on the sensitivity of our HVS to particular visual primitives and presents interesting paths for investigation. Hexagonal sampling is a complicated nonseparable sampling arrangement but its similarity to the HVS leads to advantages in visual representation.

A trade-off between the ease of implementation of the separable arrangement and the perceptual advantages of hexagonal sampling is quincunx sampling. The quincunx arrangement leads to perceptually efficient representations due mainly to its diagonal frequency cutoff, and also to elegant implementations – it reduces to separable resampling every second operation.

This chapter introduces the quincunx lattice and explains the properties which make it attractive for image compression. First general lattices and then the quincunx lattice are described, then we derive some of the properties of the quincunx lattice and explain the significance of these in a psychovisually driven image processing context. Some of the mathematics concerned with the sampling process is examined and we make comparisons with the lattice for separable sampling (by two).

We find that the quincunx lattice has a diagonal cutoff in the frequency domain which is an attractive feature for psychovisual coding and that its rate of decimation is lower than that of separable sampling. We also find that repeat

applications of quincunx sampling can result in separable sampling at alternating steps and can also lead to convergence of signals to smooth scaling functions and wavelets.

### 3.1. Lattices and Sub-lattices

Conceptually a lattice is a set of discrete points in a coordinate system which are regularly spaced<sup>1</sup>. A common example is the pixels on a digital display which are arranged in a Cartesian grid forming a lattice with regular spacing in the horizontal and vertical directions.

If  $a_1, a_2, \dots, a_N$  are linearly independent real basis vectors in  $\mathbf{R}^N$  a lattice is defined as [22],[55]

$$(3.1.1) \quad \Lambda = \{\lambda_1 a_1 + \lambda_2 a_2 + \dots + \lambda_N a_N\}$$

where  $\lambda_1, \lambda_2, \dots, \lambda_N \in \mathbf{Z}$  are integral multipliers along the vectors  $\{a_n\}$ .

In the case of the Cartesian pixel grid example given above, the basis vectors are the horizontal and vertical cardinal unit vectors  $a_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$  and  $a_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$  and one unit is the distance between pixels.

When a lattice is defined it provides a coordinate framework defined by the directional vectors of the lattice. Hence any point on the lattice can be specified as an integer linear combination of the basis vectors for the lattice. On the normal Cartesian grid this is trivial and familiar as a location  $(x, y)$  is found as  $(x \cdot a_1, y \cdot a_2)$ .

A sublattice is again a lattice which is some subset of another (original) lattice where the basis vectors of the new lattice are integer linear combinations of the basis vectors of the original lattice. The set  $\{\underline{u} = D\underline{n}, \underline{n} \in \Lambda\}$  is a sub-lattice of  $\Lambda$  if  $\underline{n} \in \Lambda$ .  $D$  is a square matrix whose columns are the new directional vectors of the sublattice defined in terms of the basis vectors of the original lattice. In this case  $D$  is called a subsampling matrix and it is not unique because there will be other subsampling matrices that are capable of producing the same sub-lattice.

If there are two lattices  $\Lambda_1, \Lambda_2$  then  $\Lambda_2$  is a sublattice of  $\Lambda_1$  if every point in  $\Lambda_2$  is a point in  $\Lambda_1$ , ie  $\Lambda_2 \subset \Lambda_1$ . This is sometimes written  $\Lambda_2 \in \Lambda_1$  in the literature associated with lattice and number theory.

**EXAMPLE 3.1.1.** Lets consider the 2D quincunx sub-lattice on the Cartesian lattice. Our original lattice  $\Lambda = \{\underline{i} \cdot \lambda_1 + \underline{j} \cdot \lambda_2\}$  and  $\underline{i}$  and  $\underline{j}$  are the orthogonal unit directional vectors on the 2D Cartesian grid. This lattice is the set of all points with integer coordinates in 2D space. The quincunx<sup>2</sup> (or red-black [94]) sub-lattice is formed by taking every second sample in a chess-board arrangement. Figure 3.5.1 (a) shows the quincunx subsampling arrangement when sampling from the Cartesian grid. This subsampling arrangement can be defined by two vectors

<sup>1</sup>Note that a regularly spaced set is not necessarily a lattice.

<sup>2</sup>Quincunx means “an arrangement of four and one in the centre.” - the dictionary (Oxford?).

which make up the subsampling matrix.

Firstly the directional vectors  $v_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$  and  $v_2 = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$ , as shown in Figure 3.1.1, define quincunx subsampling since from any point on the quincunx sublattice, we can move to any other point on the sublattice using only an integer linear combination of  $v_1$  and  $v_2$ . Thus in this case the subsampling matrix would be  $D = \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix}$ .

The more usual form for the quincunx subsampling matrix used in this document is given by  $D = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ . The main reason for the use of this matrix is that repeated application of this sampling arrangement returns to separable sampling by 2 in each dimension after every second applications. That is

$$(3.1.2) \quad D \cdot D = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}.$$

This is not the case for other matrices. It will also be shown later that this matrix has an important property for wavelet design as it allows for convergence results to be developed leading to continuous wavelets on the quincunx lattice. In particular  $D = \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix}$  does not lead to convergence [56, 58].

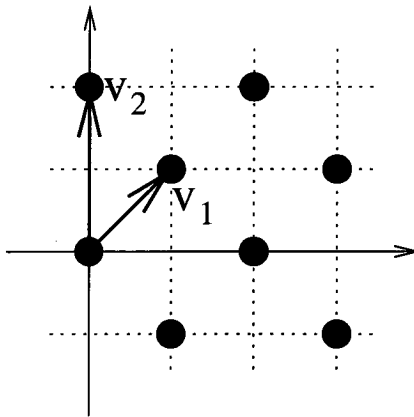


FIGURE 3.1.1. Example quincunx lattice definition. The two basis vectors are  $v_1$  and  $v_2$ . The dots indicate the sampling points of (the first coset of) this lattice. Any lattice point is an integer linear combination of  $v_1$  and  $v_2$ , ie.  $x \in \Lambda$  if  $x = a_1 \cdot v_1 + a_2 \cdot v_2$  where  $a_1, a_2 \in \mathbb{Z}$ .

The observant reader will at this point realize that the definition of a subsampling lattice is equivalent to a basis transformation where the new basis vectors correspond to integer linear combinations of the original basis vectors as described by the subsampling matrix.

### 3.2. Cosets

When a sublattice is defined on a lattice it will not include every point of the original lattice since this would result in the original lattice again. It is possible to define a lattice matrix which does include every point of the original lattice - in this case the matrix will have a determinant of magnitude 1. The points which are included in a sublattice are determined by the subsampling matrix and the point of origin of the new lattice with respect to the original lattice. A *coset* is one of the possible sets of points which can be included depending on the origin of the sublattice relative to the original lattice.

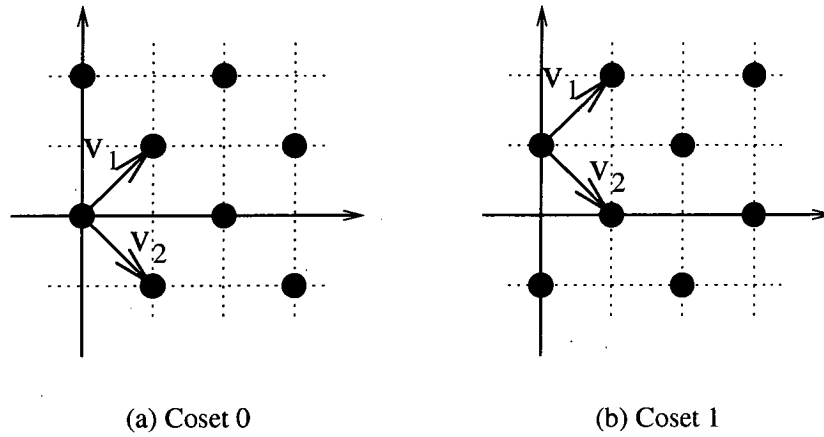


FIGURE 3.2.1. The two cosets of the quincunx sublattice on the Cartesian lattice. There are no shared points between the cosets of a lattice and the sum of all cosets is the original lattice.

Figure 3.2.1 shows the two possible cosets of the quincunx lattice. Note that both cosets have the same directional vectors (ie. subsampling matrix) and differ only by the offset of the coset origin with respect to the original lattice origin.

The number of cosets on a lattice is given by the absolute value of the determinant of the of the subsampling matrices derived from  $D$ , ie.  $\text{cosets} = |\det(D)|$ . Conversely the sampling density is given by  $\frac{1}{|\det D|}$  and a sampling density of 1 must include all points in the original lattice. All possible subsampling matrices for a given sublattice will have the same number of cosets and one can convert from one matrix to another matrix for the same lattice by multiplication with a *unimodular* matrix, ie a matrix with determinant equal to  $\pm 1$  [55]. Note, however, that the number of cosets does not uniquely define a sublattice.

EXAMPLE 3.2.1. For  $D_1 = \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix}$  we have  $\det(D) = (1 \cdot 2) - (1 \cdot 0) = 2$  as expected.

As expected we find that the determinant is equal to 2, the number of cosets of the sublattice which exist on the original lattice. So the sampling density of the resulting lattice is  $\frac{1}{2}$  with respect to the original lattice.

EXAMPLE 3.2.2. For  $D_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$  we have  $\det(D) = (1 \cdot -1) - (1 \cdot 1) = -2$

We can convert between the two matrices by  $D_1 = A \cdot D_2$  using the unimodular matrix  $A = \begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix}$ .

The intersection of any of the cosets of a sampling lattice will be empty (ie. no point can be in more than one coset) and the union of all cosets is the original lattice.

$$(3.2.1) \quad \bigcap_{i=1}^{|\det(D)|} \Lambda_i = \mathbf{0}$$

$$(3.2.2) \quad \bigcup_{i=1}^{|\det(D)|} \Lambda_i = \Lambda_{orig}$$

where  $\Lambda_i$  is the  $i^{th}$  coset of  $\Lambda$  on the original lattice  $\Lambda_{orig}$ .

With different subsampling arrangements the number of cosets will vary.

EXAMPLE 3.2.3. Consider the subsampling arrangement shown in Figure 3.2.2. The two directional vectors are  $v_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$  and  $v_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$  so  $D_3 = \begin{pmatrix} 2 & 1 \\ 1 & -1 \end{pmatrix}$ . It has a determinant of 3 and a subsampling factor of 3 (sampling density of  $\frac{1}{3}$ ) ie. one in three samples of the original lattice is retained in the sublattice.

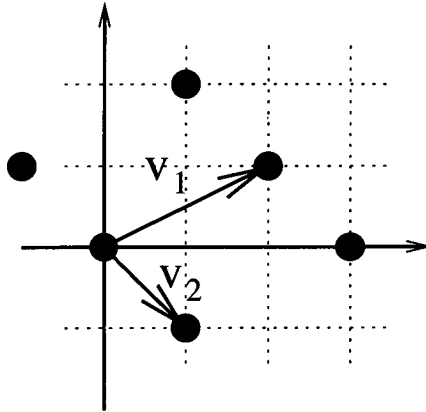


FIGURE 3.2.2. Sublattice described in example 3.2.3. There are three cosets and hence the subsampling matrix has a determinant of magnitude 3.

An alternate subsampling matrix for this example could be  $D_3 = \begin{pmatrix} 2 & 3 \\ 1 & 0 \end{pmatrix}$ .

In this case the determinant of the subsampling matrix is again  $\det(D_3) = 3$ .



It should be noted that the generation of the Cartesian lattice in  $\mathbf{R}^N$  can be accomplished with a sampling matrix  $D_{cart} = I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & 1 \end{pmatrix}$ , the identity matrix in  $\mathbf{R}^N$ . Since  $\det(I) = 1$  it follows that the orthonormal Cartesian sampling lattice can be generated by any unimodular matrix.

### 3.3. Unit Cells

The unit cell  $\mathcal{U}_c$  of a sampling lattice is defined as a parallelepiped on the parent lattice (or on  $\mathbf{R}^N$  if the lattice is not a sublattice) containing exactly one point from each coset resulting from a sublattice, on the original lattice. By definition the unit cell of a sublattice projected onto the parent lattice spans a unit area in the coordinate space of the sublattice - hence the name 'unit' cell.

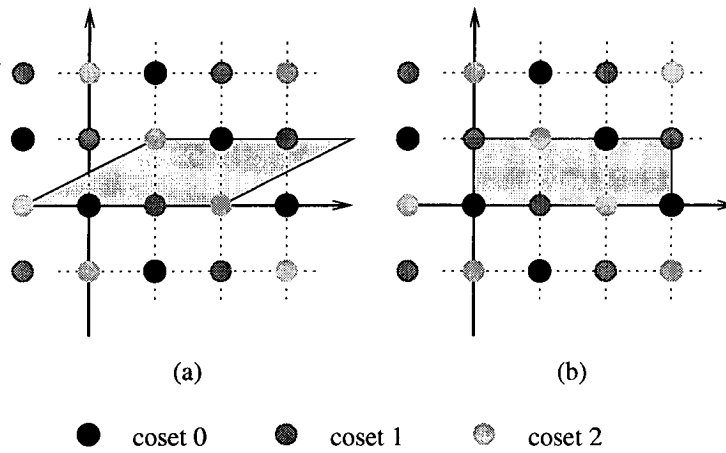


FIGURE 3.3.1. Two alternative unit cells for the resampling lattice of figure 3.2.2. The unit cell includes exactly one point from each coset of the sampling lattice. It has  $volume = |\det(D)|$  in the original lattice and  $volume = 1$  in the sublattice.

For simplicity we will refer to the unit cell as the parallelepiped formed by the basis vectors of the sublattice sampling matrix. Figure 3.3.1 shows two possible unit cells for the sampling matrix defined in Example 3.2.3.

EXAMPLE 3.3.1. The 2D quincunx lattice generated by  $D = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$  has a unit cell as shown in Figure 3.3.2. It can be seen that it is bounded on all vertices by members of the same coset, however only one of the points is included within the volume of the unit cell. One point from each of the two quincunx cosets is included within the cell - a fact which becomes more obvious in Figure 3.3.2 (b) where the coset is shifted from the position in Figure 3.3.2 (a).

The sum (union) of the unit cells with origins on each point of a lattice  $\Lambda \in \mathbf{R}^N$  tiles all of  $\mathbf{R}^N$  as is described in Equation 3.2.2.

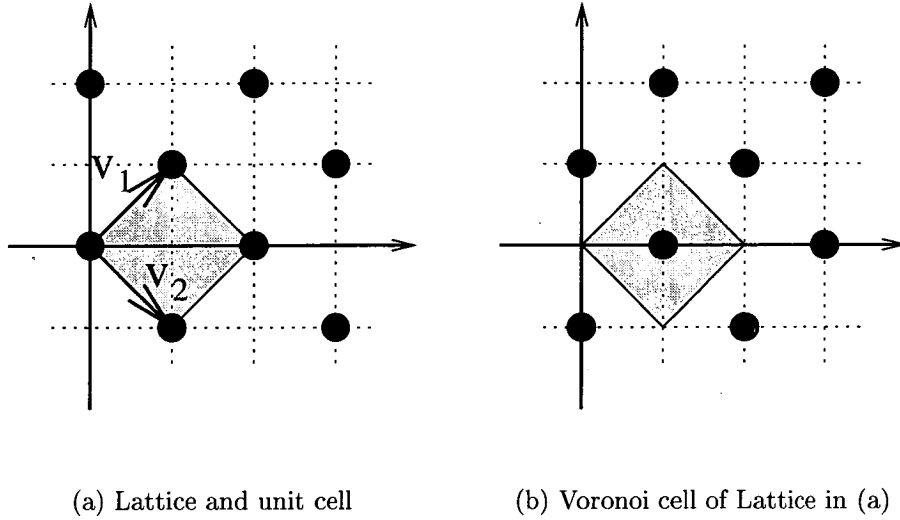


FIGURE 3.3.2. (a) The unit cell of the quincunx sublattice on the Cartesian lattice with its defining vectors. (b) The Voronoi cell of the same lattice.

Going back to Figure 3.2.1 we can write

$$\Lambda_i = \{n \cdot D + k_i\}$$

where  $k_i \in \mathcal{U}_c$  are the unique members of any unit cell of the lattice.

The Voronoi cell is a special version of the unit cell which is centred about the origin of the coordinate system. It has interesting properties when considering the Fourier transforms associated with a sublattice.

### 3.4. Frequency-Space

A sampling lattice defines a periodic sampling of the original lattice. The periodic nature of the sampling therefore defines a frequency response of the sampling system which has a periodic structure based on the sampling lattice. In the separable (Cartesian) case the sampling lattice defines a Fourier Transform with a periodicity which is rectangular and has periods defined by the sampling frequency  $\omega_s = (\omega_{sx}, \omega_{sy})$ .

**3.4.1. Downsampling.** If  $f(x)$  is a continuous function which is transformed into a new coordinate system determined by  $D$  then  $g(x)$ , which is  $f(x)$  in the new coordinate system, is given by

$$g(x) = f(Dx)$$

ie. the coordinate transform matrix defines  $x \rightarrow D^{-1}x$  or  $x \leftarrow Dx$ . Restating to avoid ambiguity, a coordinate  $x$  in the Cartesian coordinate system is transformed to a coordinate  $D^{-1}x$  in the new coordinate system.

If we discretize this we have

$$(3.4.1) \quad g(n) = f(Dn)$$

which we can use to define a sampling operation if  $D$  defines a sub-lattice of the original sampling lattice.

This can be expressed alternatively in the  $z$ -transform and frequency domains as [22]

$$(3.4.2) \quad G(\omega) = \frac{1}{N} \sum_{k \in \mathcal{U}_c^t} F\left((D^t)^{-1} \cdot (\omega - 2\pi k)\right)$$

and

$$G(z) = \frac{1}{N} \sum_{k \in \mathcal{U}_c^t} F\left(W_{D^{-1}}(2\pi k) \cdot z^{D^{-1}}\right)$$

where  $N = \det(D)$  and  $W(\cdot)$  is the modulation function ( $N^{\text{th}}$  root of unity) on the lattice defined as

$$W_M(\omega) = (e^{-j\langle\omega, m_1\rangle}, e^{-j\langle\omega, m_2\rangle}, \dots, e^{-j\langle\omega, m_n\rangle})$$

where  $m_i$  is the  $i^{\text{th}}$  column vector of  $M$ . The notation  $z^M$  is used as a shorthand notation to denote resampling onto  $M$  and expands as

$$\begin{aligned} z^M &= z^{\begin{pmatrix} a & c \\ b & d \end{pmatrix}} \\ &= \begin{pmatrix} z^{m_1} & z^{m_2} \end{pmatrix} = \begin{pmatrix} z^{\begin{pmatrix} a \\ b \end{pmatrix}} & z^{\begin{pmatrix} c \\ d \end{pmatrix}} \end{pmatrix} \\ &= (z_1^a z_2^b, z_1^c z_2^d) \end{aligned}$$

which specializes for the quincunx case as

$$z^D = (z_1 z_2, z_1 z_2^{-1})$$

So the transform  $z \rightarrow z^D$  is  $(z_1, z_2) \rightarrow (z_1 z_2, z_1 z_2^{-1})$ .

In a more conventional sense, in the frequency domain  $M \cdot \omega = (a\omega_1 + b\omega_2, c\omega_1 + d\omega_2)$  as expected.

The frequency response of a function sampled onto another lattice is determined by the Voronoi cell of  $D^{-1}$ . The lattice points in a Voronoi cell diagram represent repeat spectra points in the Fourier domain. As an example consider two lattices - lattice  $\Lambda_0$  is the normal Cartesian lattice and lattice  $\Lambda_1$  is the quincunx lattice (which is a sublattice of  $\Lambda_0$ ). The Voronoi cell of  $\Lambda_0$  as shown in Figure 3.4.1 spans a frequency range from  $-\omega_s/2$  to  $\omega_s/2$  in each dimension which is as we expect from our sampling theory and the repeated cells tile the frequency domain. The Voronoi cell for  $\Lambda_1$  as shown in Figure 3.3.2 (b) is for a lattice defined by  $D^{-1} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{pmatrix}$  so considering the two Voronoi cells together on the same scale as in Figure 3.4.2 (a) we find that the cutoff for resampling on the quincunx

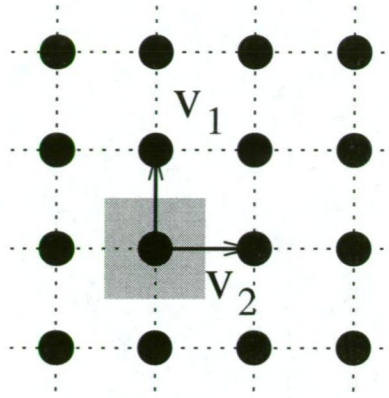


FIGURE 3.4.1. The Voronoi cell  $\mathcal{U}_c$  for the integer Cartesian lattice. This defines a periodic frequency response which spans the region  $-\omega_s/2$  to  $\omega_s/2$ . Each of the sampling points shown represents a point of repeat spectra ie. divisions by  $\omega_s$ . The Voronoi cells tile to cover the frequency domain.

is diagonal with respect to the previous sampling lattice. This is an advantageous property from a perceptual coding standpoint because the contrast discrimination of the HVS is minimum for diagonal orientations.

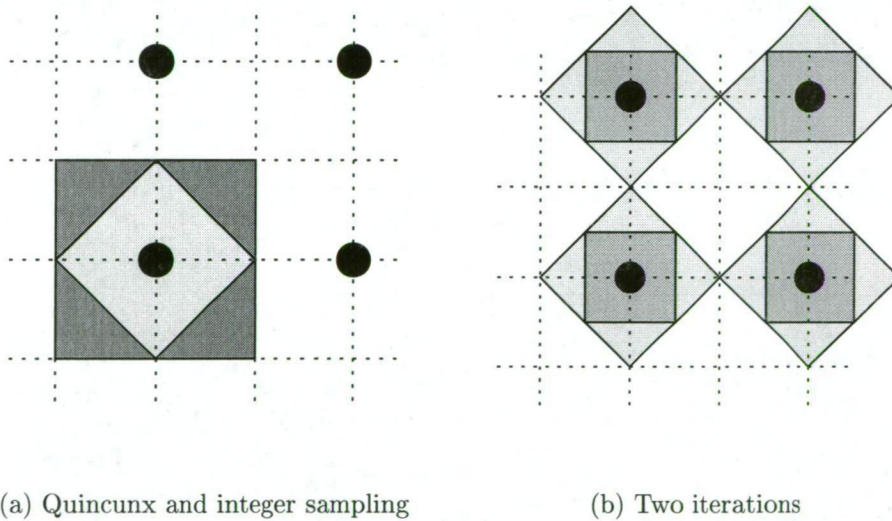


FIGURE 3.4.2. The Voronoi cells of integer sampling and Quincunx sampling shown on the same scale. The sampling points represent repeat spectra spacing in terms of integer sampling. The quincunx resampling has diagonal cutoff with respect to the previous sampling.

Figure 3.4.2 (b) shows two iterations of quincunx sampling on the same scale as the integer sampling. This shows how the retained portion of the spectra is narrowed by each operation. If a high-pass filter is first used then the repeat spectra are centred around points offset by  $\pi$  which tiles the missing sections in

the spectra. It should also be noted how the second iteration of quincunx downsampling has resulted in the same scenario as generated by one level of separable downsampling, ie. where  $D = 2I$  and  $D = \frac{1}{2}I$ .

**3.4.2. Upsampling.** The continuous transform process in the upsampling direction is quite straight forward, we simply have

$$(3.4.3) \quad f(x) = g(D^{-1}x)$$

where  $f(x)$  exists in the original coordinate system and  $g(x)$  exists in the transformed coordinate system. This is a related process to upsampling but with a critical difference; upsampling sets to zero all points that are not on the coset upsampled to. The process of downsampling followed by upsampling demonstrates that the values sampled at only one coset are retained and all others are set to zero. So discretizing Equation 3.4.3 we get

$$f(n) = \begin{cases} g(D^{-1}n) & \text{if } n = Dk \\ 0 & \text{otherwise} \end{cases}$$

and again we can express this in the z-transform and frequency domains as [22]

$$F(\omega) = X(D^t \omega)$$

and

$$F(z) = X(z^D)$$

**3.4.3. Coset removal (up/down sampling).** The overall operation of downsampling followed by upsampling can then be represented as

$$\hat{f}(n) = \begin{cases} f(n) & \text{if } n = Dk \\ 0 & \text{otherwise} \end{cases}$$

which is obvious from the necessary behaviour of removing all but one coset. We also have

$$(3.4.4) \quad \hat{F}(\omega) = \frac{1}{N} \sum_{k \in \mathcal{U}_c^t} X(\omega - 2\pi (D^t)^{-1} k)$$

$$(3.4.5) \quad \hat{F}(z) = \frac{1}{N} \sum_{k \in \mathcal{U}_c^t} X(W_{D^{-1}}(2\pi k) \cdot z)$$

If we now specialize Equations 3.4.4 and 3.4.5 for the quincunx case we have

$$\begin{aligned} F(\omega) &= \frac{1}{2}X(\omega) + \frac{1}{2}X\left(\omega - 2\pi (D^t)^{-1} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix}\right) \\ (3.4.6) \quad &= \frac{1}{2}(X(\omega_1, \omega_2) + X(\omega_1 + \pi, \omega_2 + \pi)) \end{aligned}$$

and

$$(3.4.7) \quad F(z) = \frac{1}{2}(X(z_1, z_2) + X(-z_1, -z_2))$$

Both Equations 3.4.6 and 3.4.7 have been arrived at for separable sampling via alternative methods.

### 3.5. Nonseparable Sampling

It has been argued for some time (for example [56]) that image processing should be handled in a truly two-dimensional fashion which would allow images to be treated as two-dimensional structures rather than as the rows and columns which we are familiar with. Unfortunately it has been easier and most effective in the past to adapt the great body of work done in one-dimensional signal processing to applications in image processing than to attack head-on the increased complexity of the multidimensional paradigm. This has led to great success but it has largely left unexplored the range of possibilities which nonseparable multidimensional signal processing potentially has to offer.

With computational power increasing at an explosive rate it seems appropriate that the issue of multidimensional signal processing be addressed at this point in time. The increased mathematical complexity and associated processing delays have been a hindrance to this field but this should no longer be a deciding factor since the processing power required is within the range of the average consumer and easily available to researchers.

For image processing, treating images in a nonseparable manner opens up many possibilities which are not possible in the separable case. The extra degrees of freedom introduced expand the possibilities of filter design significantly. We can now treat images as we perceive them - as two-dimensional regions and features - not necessarily as rows and columns.

For wavelet analysis in particular the nonseparable domain opens up many possibilities which aren't available in the separable domain, especially in the area of filter design. Many of the restrictions on 1-D wavelet filters can be removed when dealing with 2-D filters and the number of free variables increases dramatically making it possible to more finely tune the filters to a particular purpose. On the other hand the extra free variables makes the design process more lengthy and involved as there are more effects to be considered because of the greater degree of freedom.

**3.5.1. The Quincunx Lattice.** The term quincunx means *an arrangement of four in a square with one in the centre*. This exactly describes the quincunx lattice.

As shown above in Example 3.2.2 the sampling density of the quincunx lattice is  $\frac{1}{2}$  so the quincunx lattice describes a sampling process where the number of samples retained is half of the total number of samples in the original image. Thus we have downsampling by a factor of two in total and a factor of  $\frac{1}{\sqrt{2}}$  in each dimension. Thus the number of cosets generated by the quincunx lattice is two. Half the points reside in each coset, one coset is retained at sampling and one

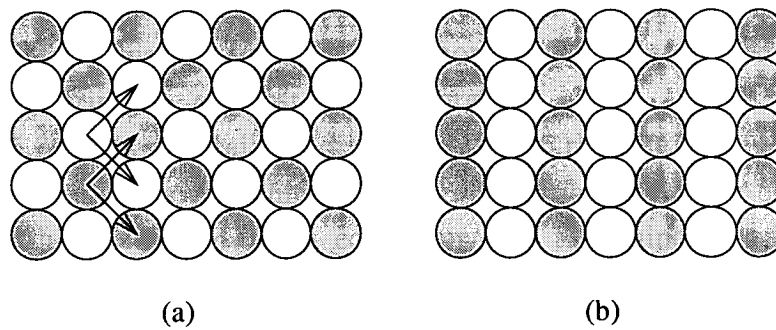


FIGURE 3.5.1. The cosets of (a) the Quincunx lattice and (b) Separable sampling by two in the horizontal direction. As can be seen, both sampling schemes have two cosets – one shown in white and one grey.

is discarded. Figure 3.5.1 shows the quincunx lattice and another lattice with a sampling density of  $\frac{1}{2}$  - separable downsampling by two in the horizontal direction - also known as column interlacing.

**3.5.2. Why Quincunx?** The arrangement of the points in the quincunx lattice produces some desirable properties when it is used as a subsampling lattice in image coding. Subsampling is an integral part of wavelet decomposition as it changes the resolution or scale of the image being considered. This allows for multiresolution decomposition or analysis to occur.

One of the desirable properties of the quincunx lattice is its diagonal cutoff in the frequency domain<sup>3</sup> as shown in Section 3.4.1. When a signal is sampled to a lower resolution it loses some of its high-frequency components. Those frequencies which are retained are those which are supported by the new sampling arrangement. The quincunx lattice has the desirable feature that the full range of vertical and horizontal frequencies are retained in the resampled image. This is in contrast to separable sampling which loses half the range of frequencies in either the horizontal or vertical dimension for each resampling application. Figure 3.5.2 shows the spectrum of quincunx resampling. As can be seen the full range of exactly vertical and horizontal frequency components is retained, with the diagonal components suffering the greatest loss of energy.

Figure 3.5.3 shows in detail the diagonal cutoff in the context of repeat spectra highlighting the low-pass and high-pass regions of the frequency domain tiling. There the high-pass and low-pass components of the quincunx sampling have been rearranged to emphasize their relationship to the normal frequency range of (ordinary) integer sampling.

As already stated the reason that this diagonal cutoff behaviour is important is that the most significant features to the HVS are vertical and horizontal edges; diagonal edges are of least importance (ie. the HVS has the least sensitivity to diagonal edges). Having the full range of horizontal frequencies means that sharp

<sup>3</sup>See also [50] for an example of this phenomenon in 3 dimensions, where it finds application in interlaced video transmission.



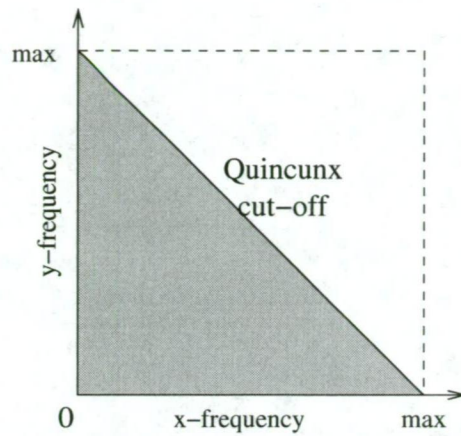


FIGURE 3.5.2. Diagram showing the spatial-frequency cut-off of Quincunx resampling. The retained spectrum is shaded grey. The full range of vertical and horizontal frequencies is retained, thus preserving vertical and horizontal edges. The most truncated frequencies are in the diagonal direction which has the lowest psychovisual importance to the human observer.

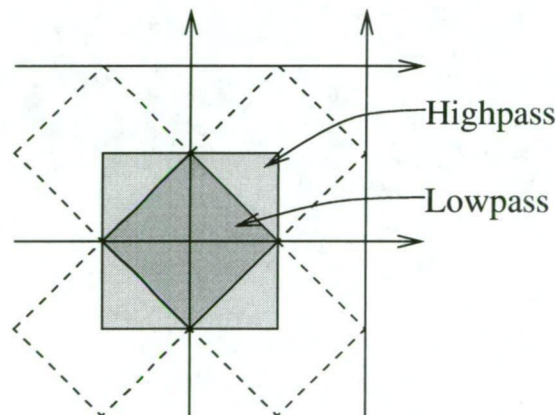


FIGURE 3.5.3. Frequency domain tiling of the quincunx lattice. Dark grey indicates the low-pass regions and light grey is the high-pass regions. The solid box shows the frequency range of the original sampling mechanism. The dashed boxes indicate the continuation of the high-pass region outside the original sampling region due to repeat spectra.

vertical edges are represented as accurately in the subsampled image as in the original and similarly with vertical frequencies and horizontal edges. As the orientation of the edges moves away from horizontal or vertical and towards diagonal the upper limit of retained frequencies drops until at  $45^\circ$  only half the original bandwidth is available, hence the quincunx lattice is said to have a diagonal cut-off.

Figure 3.5.4 shows an example of how sharp vertical and diagonal edges are handled by the subsampling scheme. It is important that this cut-off exists on

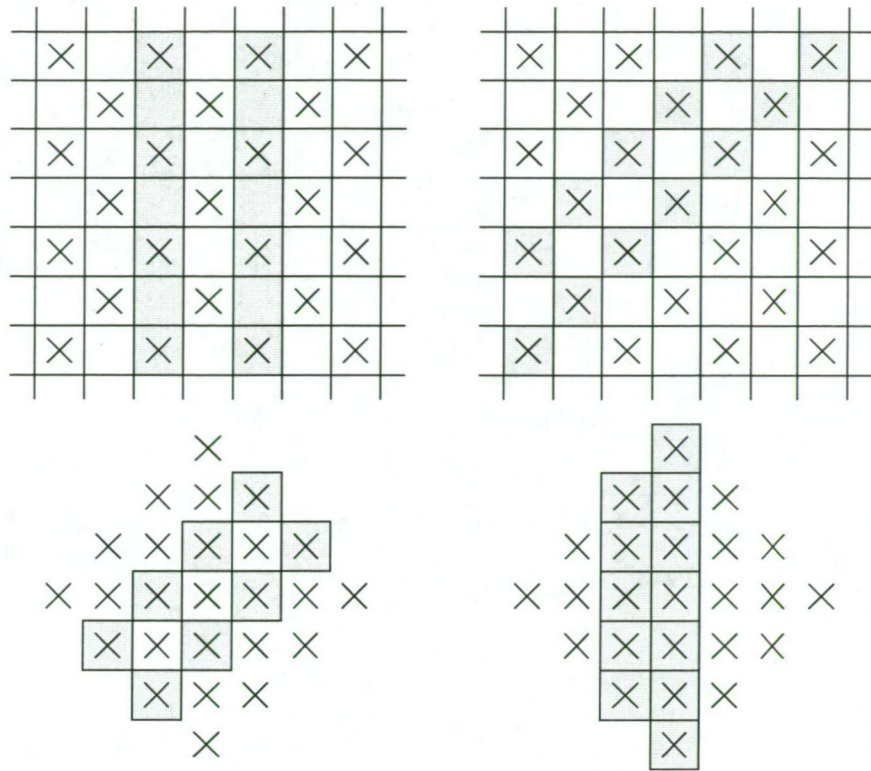


FIGURE 3.5.4. How vertical and diagonal edges of minimum spacing (maximum frequency) are retained in the quincunx lattice. The vertical edges are retained with the spacing between the edges intact, while the diagonal edges, corresponding to the cutoff frequency are blurred into a single feature and the high-frequency detail is completely lost.

the diagonal since these orientations are the least important to human vision. This means that the data which may be lost due to coding of the details in this orientation will not have as significant an effect on reconstructed image quality as it would if the details were in the vertical or horizontal direction.

The quincunx subsampling process has another benefit as well: the rate of decimation is slower than by the separable method. The sampling density of the quincunx lattice is  $\frac{1}{2}$  or  $\frac{1}{\sqrt{2}}$  in each direction meaning that half the samples are retained and half are discarded. For separable sampling the sampling ratio is  $\frac{1}{2}$  in each direction with an overall ratio of  $\frac{1}{4}$ . Retaining more samples after sampling can be an advantage when we are coding wavelet data.

It is important when we code the details or wavelet coefficients that we apply a quantization method appropriate to the relative sensitivity of the HVS to the details. It is known that the HVS has a sensitivity which varies with frequency (see Section 7.1.1 for more details on this). This allows us to more coarsely code those frequencies to which the HVS is insensitive. The DCT based coders employed in the JPEG and MPEG standards make good use of this property of the HVS. However wavelet decompositions divide the image spectrum into frequency bands which makes it more difficult to accurately consider the sensitivity of the HVS to those frequencies contained within each broad band. This is particularly difficult



when using separable sampling because it divides the image spectrum into octaves - which is quite severe decimation. When using the quincunx sampling process the spectrum is split up differently. In some ways the decimation is slower, ie. more areas of the image spectrum are retained in the low-pass subband. In fact two decompositions with the quincunx lattice are required to perform the same level of decimation as one level of decomposition using separable sampling. Thus the frequency band of the wavelet data from quincunx sampling is more compact than for separable allowing for better tuning of coding for HVS sensitivity.

### 3.6. Resampling on the Quincunx Lattice

The effects of resampling using the quincunx lattice are often not intuitive. Resampling of a signal performs a change of basis on the signal. The result is contained in a different coordinate system to the original signal. In the case of separable sampling, the cardinal vectors for both the original signal and the resulting resampled signal, point in the same directions, thus the only perceived change is a change of the scale of the image.

This is not the case with the quincunx lattice. In this case resampling has the effect of rotating the signal by  $-45^\circ$  and flipping it around the horizontal axis. Figure 5.1.2 shows the effects of quincunx downsampling on the Lenna image.

The commutativity of quincunx resampling was proven by Kovacevic and Vetterli [55]. This proved that the quincunx lattice was a candidate for use in wavelet transforms and subband decompositions. Figure 3.6.1 is provided as an aid to visualize the effects of quincunx sampling and to show the effects of downsampling and upsampling. A case of perfect reconstruction is demonstrated with the original signal decomposed and then reconstructed.

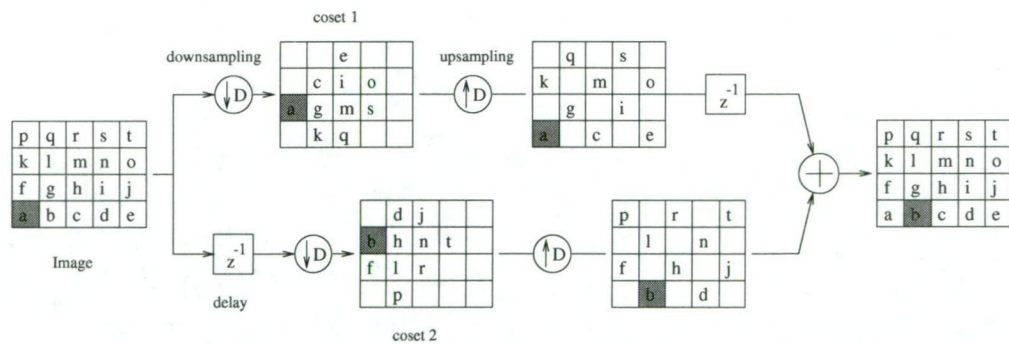


FIGURE 3.6.1. Downsample and upsample process on the quincunx lattice showing intermediate stages. The letters represent particular signal elements and show how their location is altered by the resampling process.

Although quincunx resampling appears to drastically alter the appearance of the image this change in basis is self-correcting. Applying quincunx downsampling a second time will return the signal to its original orientation as detailed in Section 3.4.1.

## CHAPTER 4

### Nonseparable Wavelet Design

This chapter describes the processes used to design the filters used in the research of this thesis. This chapter is largely background information as novel filter design is beyond the scope of this thesis. The design methods can be attributed to several different authors, chiefly Vaidyanathan, Kovačević, Vetterli and Sweldens (for example [98], [103], [55], [90]).

The design process used in the production of this thesis involved taking an existing framework and creating filters using that framework. Knowledge and methods for non-separable filter design are very limited and there are no known factorizations in two-dimensions to allow the filter design process to be modularized. The production of new methods of filter design for the nonseparable paradigm is worthy of a substantial written work on its own.

This Chapter covers in depth the design process – expanding and explaining the designs in greater detail. We cover all the nonseparable filters used in the course of the research of this thesis.

#### 4.1. Nonseparable Design Basics

We start with the basic background for nonseparable filter design.

There are two major directions of filter design for the nonseparable paradigm.

(1) Polyphase transforms[98], [96], [56]

This is a representation that simplifies filter design in some ways by considering the effect of each coset as a separate filter in a multi-input network.

(2) 1-D to 2-D transforms

This is a method which is already used in one-dimensional and separable designs, such as the McClellan transform. It allows many properties of 1D filters to be retained in 2D filters.

We will deal with each in turn.

**4.1.1. Polyphase Design.** Polyphase design is very useful when dealing with non-separable sampling lattices. It allows us to break filters down into cosets and consider the interactions between the different cosets and how each will affect overall filter behaviour. The design thus consists of designing several filters (one for each coset) in such a way that the overall polyphase filter produces the desired result. A polyphase transform takes a single-input, single-output, shift-variant system and turns it into a multi-input, multi-output, shift-invariant system. Since

resampling is performed outside the filterbank in this case the filterbank its self is a shift invariant operator.

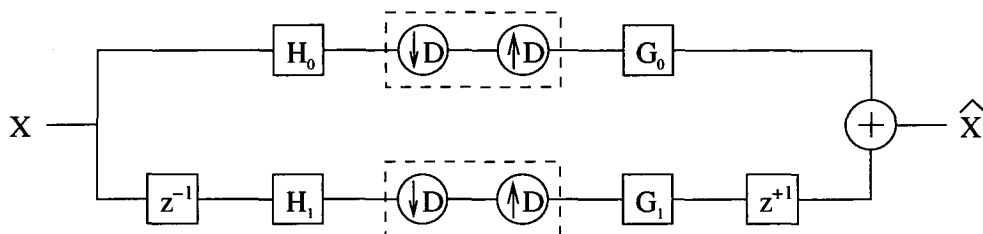


FIGURE 4.1.1. General two-channel filterbank used in wavelet transforms with nonseparable sampling. The sampling within the filter framework leads to a shift variant system.

Figure 4.1.1 shows a general non-polyphase subband coding scheme. The downsample and upsample operations performed after filtering make the analysis-synthesis operations in each branch shift variant, ie. shifting the input by one sample does not shift the output (of that branch) by one sample, but rather produces a different output shape altogether.

The heart of the polyphase design method is the polyphase representation [23], [98], [48]. If our sampling lattice has  $M$  cosets then we can write the  $z$ -domain polyphase representation of a filter operating on that lattice as

$$F(z) = F_0(z^D) + \underline{k}_1 F_1(z^D) + \cdots + \underline{k}_{M-1} F_{M-1}(z^D)$$

or

$$(4.1.1) \quad F(z) = F_0(z) + \underline{k}_1(z) F_1(z) + \cdots + \underline{k}_{M-1}(z) F_{M-1}(z)$$

where  $F_n(z^D)$  is the filter component associated with coset  $n$  and  $\underline{k}_n$  is the  $z$ -transform of the shift vector which moves from the origin of coset 0 to the origin of coset  $n$ . Obviously  $\underline{k}_0$  is omitted since it has no effect; its value is  $z^0 = 1$  by definition.

The various filter components written as  $F_n(z^D)$  are represented in the *upsampled domain*. The filters  $F_n(z)$  each represent a filter for a particular coset which can be applied to a coset after downsampling occurs on the original signal. The downsampling extracts one coset and performs the change of basis brought about by the downsample matrix.

The filters  $F_n(z^D)$  are these filters upsampled so that they operate on one coset *prior* to downsampling, ie. in the pre-filtered state. The filter taps are rearranged so as to coincide with the location of the correct signal components. The process which changes the tap locations is the upsampling process using the sampling matrix  $D$ .

Using  $z^D$  transforms coordinates via  $x \rightarrow D \cdot x$ . In the  $z$ -domain the result is a transform of  $\begin{pmatrix} z_0 & z_1 & z_M \end{pmatrix} \rightarrow \begin{pmatrix} z_0^{D_0} & z_1^{D_1} & z_M^{D_M} \end{pmatrix}$  which we denote  $z \rightarrow z^D$ .

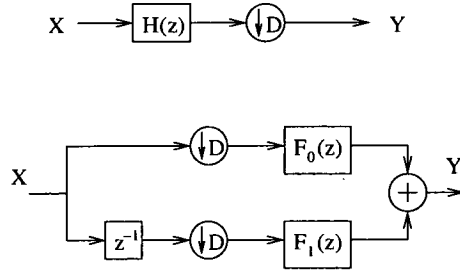


FIGURE 4.1.2. Two alternative ways to filter and downsample a signal. The upper version is the familiar filter-downsample method. The lower method is a polyphase version of the same process.

Because each filter component only operates on one coset, there must be one filter component for each coset.

For the quincunx case we have two cosets and the shift vector  $\underline{k}_1$  depends on our choice of lattice matrix  $D$ . For the situation where  $D = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$  then we use  $\underline{k}_1 = (1, 0)$  which has  $z$ -transform of  $\underline{k}_1(z) = z_1^{-1}$ . Now our polyphase quincunx filter can be written as

$$F(z^D) = F_0(z^D) + z_1^{-1} \cdot F_1(z^D)$$

and there are two polyphase components. This filter is shown in Figure 4.1.2. The filter  $F_0(z^D)$  operates on coset 0 while  $F_1(z^D)$  operates on coset 1.

Often the polyphase components are written as a polyphase matrix of filters [103], [47], [101], [99]

$$F_p(z) = \begin{pmatrix} F_{00}(z) & F_{01}(z) \\ F_{10}(z) & F_{11}(z) \end{pmatrix}$$

Each column is the components that operate on one coset, and each row is one complete filter in the filterbank.

Figure 4.1.3 shows the polyphase representation of a quincunx filterbank using  $\underline{k}_1 = \begin{pmatrix} 1 & 0 \end{pmatrix}$  as the displacement vector. Because of the commutativity of up-/down-sampling with filtering [55] we can use the upsampled representation of the filterbank within the upsampling/downsampling boundaries. This allows us to analyze the filter in terms of a filterbank without resampling.

We will define the polyphase filterbank as [47]

$$H_{p(i, \underline{k})} \begin{pmatrix} z_1 & z_2 \end{pmatrix} = \sum_{u_1=-\infty}^{\infty} \sum_{u_2=-\infty}^{\infty} h_i \left( \underline{k} + D \cdot \begin{pmatrix} u_1 & u_2 \end{pmatrix} \right) \cdot z_1^{-u_1} z_2^{-u_2}$$

where  $i$  represents the filter within the filterbank, and  $\underline{k} = \begin{pmatrix} k_1 & k_2 \end{pmatrix}$  spans a unit cell of the sampling lattice and thus determines which coset is involved. Since  $\underline{k}$  spans the unit cell of the sampling lattice, there are only a small number of discrete

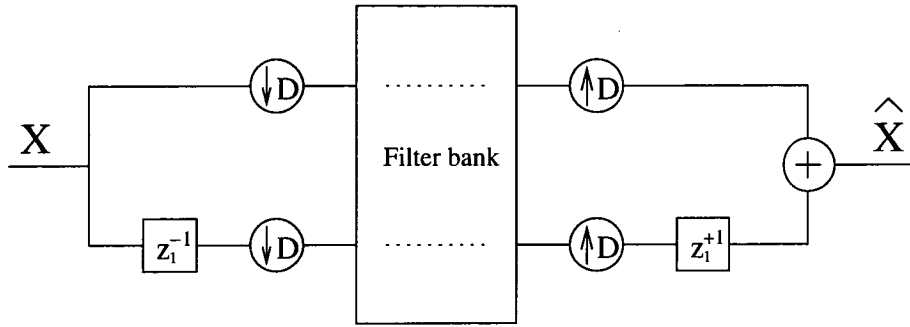


FIGURE 4.1.3. Upsampled representation of a filterbank in the quincunx domain. This is a simple extension of the forward version shown in Figure 4.1.2.

values which it can take on. In particular for the quincunx lattice and sampling lattice  $D$  as above this displacement vector can take on the values of  $\begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$  or  $\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$ .

**4.1.2. Impulse response.** Consider the 1-D impulse response of the filterbank in Figure 4.1.3. If we start with an impulse at origin or at  $2n \cdot k$  then only one coset (coset 0) can contain the impulse after downsampling as shown in Figure 4.1.4. Similarly if we have an impulse, offset from origin by  $(2n + 1) \cdot k$ , then we have only an impulse transmitted through coset 1.

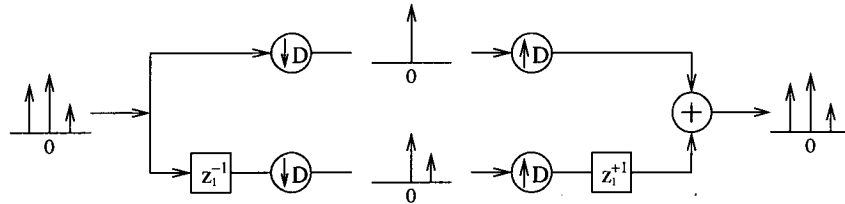


FIGURE 4.1.4. Impulse response at various points in a polyphase representation. There are two cosets and the impulses inside the structure show the retained signal after retaining only one coset in each case. The lower branch contains a shift operation which changes the coset at the origin.

It can be seen then that each branch of the system is periodically shift-variant, ie. the response varies depending on the location of features with respect to the coset origins. The use of the polyphase representation allows the system to be represented as a multi-input multi-output, shift-invariant system which allows investigation of linear filtering in this structure.

Figure 4.1.5 shows the generalization of Figure 4.1.3 to  $n$ -channels of the filterbank. In this case each of the shift operators shifts the origin of the sampling lattice to the origin of the coset of interest.



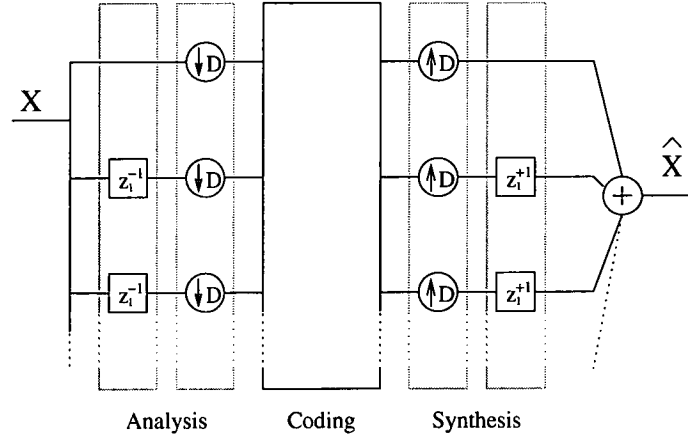


FIGURE 4.1.5. General n-channel upsampled polyphase filterbank. This is a generalization of the case shown in Figure 4.1.3 so that it has n channels. The shift operators are used to change cosets.

Following the analysis of [56, (section III)] we can define a polyphase representation of both the signal and the filters.

$$\begin{aligned} X(z) &= \sum_{\forall k \in \mathcal{U}_c} z^k \cdot X_k(z^D) \\ &= p_i^T(z) \cdot x_p(z^D) \end{aligned}$$

where

$$X_k(z) = \sum_{\forall i \in \mathbb{Z}^n} x(Di - k) \cdot z^{-i}$$

In a similar fashion, the filters for a filterbank can be defined as

$$\begin{aligned} H(z) &= \sum_{\forall k \in \mathcal{U}_c} z^{-k} H_k(z^D) \\ &= p_f^T(z) \cdot h_p(z^D) \end{aligned}$$

In these equations  $p_i(z)$  and  $p_f(z)$  are the inverse and forward polyphase transforms. They are vectors of matrices needed to shift to a particular coset for a particular element in the polyphase representation. For example, the quincunx case has two cosets; coset 0 is located at the origin - thus  $\underline{k}_0 = (0 \ 0)$ , and coset 1 is located at  $\underline{k}_1 = (1 \ 0)$ . So the forward polyphase transform  $p_f(z) = (z_1^{-1} z_2^0 \ z_1^0 z_2^0) = (z_1^{-1} \ 0)$ .

The entire filterbank system then can be described by

$$\begin{aligned} Y(z) &= p_i^T(z) \cdot G_p(z^D) \cdot H_p(z^D) \cdot x_p(z^D) \\ &= p_i^T(z) \cdot T_p(z^D) \cdot x_p(z^D) \end{aligned}$$

where  $T_p(z) = G_p(z^D) \cdot H_p(z^D)$  is referred to as the polyphase transfer matrix.

We apply a wavelet transform to an image by the use of filter banks; we have an analysis bank which transforms the image into wavelet coefficients and a synthesis bank which transforms wavelet coefficients into an image. Each filter

bank consists of a low-pass filter and a series of band-pass filters (depending on the sampling pattern and number of cosets these may simply be high-pass filters). We concentrate on the case where we have two bands per decomposition, ie. a low-pass and a high-pass. In this case we have two filters in each filter bank (a total of 4 polyphase components).

If our filters are orthogonal (paraunitary) then the synthesis filters in each channel are - by definition - the same as the corresponding analysis filters (subject to reversal in the spatial domain), however for non-orthogonal system eg. biorthogonal, the synthesis filters are - in general - different from their corresponding analysis filters.

In the case of the two-channel decomposition we can use the Quadrature Mirror relationship which allows us to derive one filter bank from the other and guarantee alias cancellation. This means we only need to design one filter bank that satisfies Quadrature Mirror Filter constraints in the design and we will have designed both analysis and synthesis filter banks simultaneously. This puts extra design restrictions on the filters, but in two dimensions there is generally a large scope in freedom of design because of the large number of coefficients which need to be designed - thus yielding a large number of degrees of freedom.

**4.1.2.1. Perfect Reconstruction Constraint.** The most important property for a filter bank in data compression is perfect reconstruction. Perfect reconstruction means that once a signal has been analyzed into its derived coefficients by the analysis filter bank it can be synthesized, by the synthesis filter bank, back into a signal identical to the original signal (possibly delayed some samples due to processing, and linearly scaled).

Once we know that the wavelet coefficients can completely reconstruct the signal we can begin the process of data compression by quantizing the coefficients. If we cannot guarantee perfect reconstruction with all of the information from the analysis then it is difficult to aim for good reconstruction when some of the information is removed by coding.

**4.1.2.2. Zero Aliasing Constraint.** Aliasing is a major problem in digital signal processing. In the case of the polyphase representation, aliasing occurs when a signal component from one coset affects another coset in the result signal. Alias cancellation is also referred to as cross-talk cancellation, which comes from the transmultiplexer applications in which multirate filterbanks have been used to combine and separate signal components.

If we consider the polyphase representation of a signal on a sampling lattice, there are effectively several signals - one per coset. A polyphase filterbank can be considered to be similar to a multiplexer-demultiplexer arrangement (in reverse) where the original signal is first broken into separate channels in the analysis process and then recombined into a single signal in the synthesis process. This is in contrast to telecommunications applications where several signals are combined

via a multiplexer into a single signal for transport via some channel, then demultiplexed into the original signals at the other end. In this example, polyphase aliasing manifests its self as cross-talk: the signal in one channel affects the signal in another channel. In the analysis/synthesis paradigm aliasing manifests as distortions in the reconstructed signal as the various cosets interfere with each other.

**4.1.3. Multidimensional Transforms [9].** This technique revolves around the McClellan transform – a method for taking one-dimensional filters and producing two-dimensional filters subject to certain constraints. This technique has the advantage that the filters which are derived can be made to have many of the properties of the original one-dimensional filter.

These transforms were not used to produce any filters used in the research of this thesis. They are noted for completeness.

## 4.2. General Results

First we introduce some results on 2-channel (coset) filtering which are general in all cases. The results can be applied to any wavelet transform operation which can be expressed in the  $z$ -domain, but we will pay particular attention to the case where quincunx wavelet filtering is applied.

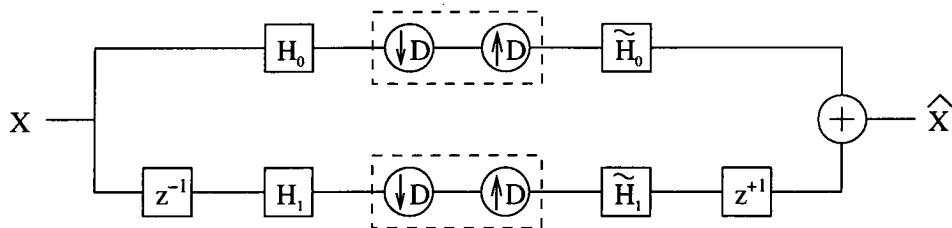


FIGURE 4.2.1. Configuration of simple decomposition and reconstruction using normal (not-polyphase) filters.

The process of decomposition and reconstruction by one level of wavelet decomposition is performed by this sequence of operations

- (1) Filtering by analysis filter - result is same dimension as original
- (2) Subsampling - reduces dimension and loses unwanted cosets. Information is now spread across multiple channels.
- (3) Upsampling - inserts zero at positions of removed cosets, several channels at same dimension as original signal.
- (4) Filtering by synthesis filter - create components of result from each channel.
- (5) Recombination - result components from several channels are summed to regain original signal.

We can combine the subsample/upsample operations into a single operation with a z-transform given in general for  $N$  channels by [103]

$$(4.2.1) \quad Y(z) = \frac{1}{N} \sum_{k=0}^{N-1} X(\mathcal{W}^k z)$$

where  $\mathcal{W}^k = e^{-j2\pi k/N}$  is called the  $N^{\text{th}}$  root of unity. This specializes our case (two cosets) to the famous result of

$$(4.2.2) \quad Y(z) = \frac{1}{2} (X(z) + X(-z))$$

The effect of filtering applies  $F(z)$  to the signal. So the net effect of applying all the steps to the signal  $X(z)$  in Figure 4.2.1 is

$$\hat{X}(z) = \frac{1}{2} \left( H_1(z) \frac{1}{2} (\tilde{H}_1(z) + \tilde{H}_1(-z)) + H_0(z) \frac{1}{2} (\tilde{H}_0(z) + \tilde{H}_0(-z)) \right) X(z)$$

In this case we consider the effects of sample shifting to be absorbed into the filter operators of  $H_1(z)$  and  $\tilde{H}_1(z)$  (or which ever operators are determined to absorb the shift, the effect is the same regardless).

Expanding this we get

$$(4.2.3) \quad \hat{X}(z) = \frac{1}{2} \left[ \begin{array}{l} \tilde{H}_0(z) H_0(z) X(z) + \tilde{H}_0(z) H_0(-z) X(-z) \\ + \tilde{H}_1(z) H_1(z) X(z) + \tilde{H}_1(z) H_1(-z) X(-z) \end{array} \right]$$

and we find that we can factor this as

$$(4.2.4) \quad \hat{X}(z) = \frac{1}{2} \left( \begin{array}{cc} \tilde{H}_0(z) & \tilde{H}_1(z) \end{array} \right) \left( \begin{array}{cc} H_0(z) & H_0(-z) \\ H_1(z) & H_1(-z) \end{array} \right) \left( \begin{array}{c} X(z) \\ X(-z) \end{array} \right)$$

This representation is quite useful to us because it allows us to consider the effects of perfect reconstruction and alias cancellation on the system. This result can be found in [104] and [102] among others.

Aliasing results from the aliased input  $X(-z)$ . For alias cancellation, we require that the all terms which involve the aliased input  $X(-z)$  be removed - ie. canceled from the output expression. That is, we require

$$(4.2.5) \quad \tilde{H}_0(z) H_0(-z) + \tilde{H}_1(z) H_1(-z) = 0$$

as a condition for alias cancellation.

Independent of this result we can look at perfect reconstruction. In this case we require that the output signal is an exact replica of the input signal, ie.  $\hat{X}(z) = X(z)$ . From Equation 4.2.4 we require that

$$(4.2.6) \quad \left( \begin{array}{cc} \tilde{H}_0(z) & \tilde{H}_1(z) \end{array} \right) \left( \begin{array}{cc} H_0(z) & H_0(-z) \\ H_1(z) & H_1(-z) \end{array} \right) = \left( \begin{array}{cc} 1 & 0 \end{array} \right)$$

This effectively gives us two conditions to satisfy

$$(4.2.7) \quad \tilde{H}_0(z) H_0(z) + \tilde{H}_1(z) H_0(-z) = 1$$

$$(4.2.8) \quad \tilde{H}_0(z) \tilde{G}(z) + \tilde{H}_1(z) \tilde{G}(-z) = 0$$

Note that the results obtained so far in this section make no assumption about the nature of the filters used. Indeed  $H_0, H_1, \tilde{H}_0$  and  $\tilde{H}_1$  may be four totally unrelated filters, yet if they satisfy the conditions in Equations 4.2.5, 4.2.7 and 4.2.8 they will produce a perfect reconstruction filterbank with alias cancellation.

There are things we can do to make our lives easier so we don't have to design all four filters. We will now briefly cover some of the design methods which can aid in the production of the filters.

**4.2.1. Quadrature Mirror Filter (QMF) relationship.** The QMF relationship is an important one for alias cancellation and wavelet filter bank design in general. The QMF relationship can be stated as

$$(4.2.9) \quad H_1(\omega_1, \omega_2) = H_0(\omega_1 + \pi, \omega_2 + \pi)$$

This makes the filter frequency responses mirror images about the cut-off frequency. We can design filters in the frequency domain which have particular properties using this technique. One property which is often desired is smoothness of the low-pass filter at the frequency origin. In the QMF paradigm this means that the high-pass filter will have a similar property as  $\omega \rightarrow \pi$ .

Looking at Equation 4.2.9, in the z-domain we then have (with  $z = e^{j\omega}$ ):

$$H \begin{pmatrix} z_1 & z_2 \end{pmatrix} = H \begin{pmatrix} e^{j\omega_1} & e^{j\omega_2} \end{pmatrix}$$

Equation 4.2.9 then becomes

$$(4.2.10) \quad \begin{aligned} H \begin{pmatrix} z_1 & z_2 \end{pmatrix} &= H_1(e^{j\omega_1}, e^{j\omega_2}) = H_0(e^{j(\omega_1+\pi)}, e^{j(\omega_2+\pi)}) \\ &= H_0(-e^{j\omega_1}, -e^{j\omega_2}) \\ &= H_0(-z_1, -z_2) \end{aligned}$$

Thus we see that the QMF relationship in the z-domain can be written

$$(4.2.11) \quad H_1 \begin{pmatrix} z_1 & z_2 \end{pmatrix} = H_0 \begin{pmatrix} -z_1 & -z_2 \end{pmatrix}$$

When considering the spatial filter configuration this equates to

$$(4.2.12) \quad h_1(n_1 + n_2) = (-1)^{n_1+n_2} h_0(n_1 + n_2)$$

for the two-channel case. This closely resembles the one-dimensional version written as

$$(4.2.13) \quad h_1(n) = (-1)^n h_0(n)$$

If we look carefully at this relationship we see that the quadrature mirror property equates to reversal in  $n$  and modulation of the filter taps, regardless of the number of dimensions.

**4.2.2. Unitary case.** Orthogonality of basis functions equates to unitary structure in the transfer matrix. In this situation we refer to the transfer function of  $\hat{X}(z) = T(z) \cdot X(z)$ . For perfect reconstruction the transfer function  $T(z)$  must equate to a simple scaling and shifting of the signal, ie.  $T(z) = k \cdot z^{-n}$ .

Strictly speaking, perfect reconstruction requires  $k = 1$  and  $n = 0$  but in the general case where signals are causal, we can relax the constraints to allow shifted versions of the original. In any event the original signal can be exactly reproduced to such a point that  $\hat{X}(z) = k \cdot X(z^{-n})$ . For image processing we require that  $n = 0$ , but if this property is not achieved directly through  $T(z)$  it can certainly be obtained with minimal post processing.

We start with a discussion of matrix properties related to the unitary case.

A square matrix  $H$  is paraunitary if it satisfies [104]

$$(4.2.14) \quad \tilde{H}(z) \cdot H(z) = H(z) \cdot \tilde{H}(z) = c \cdot I$$

where  $\tilde{H}(z) = H_*^T(z^{-1})$  or in the complex plane  $\tilde{H}(e^{j\omega}) = (H(e^{j\omega}))^*$ . Compare this also with Equation 2.12.1. In the first expression  $z$  becomes  $z^{-1}$  because the adjoint of advance ( $e^{j\omega} = z$ ) is delay ( $e^{-j\omega} = z^{-1}$ ).

$X_*(z)$  means conjugation of the coefficients of  $X(\cdot)$  but not of  $z$  its self, ie.  $X_*(z) = x_0^* + x_1^*z + x_2^*z^2 + \dots$  if  $X(z) = x_0 + x_1z + x_2z^2 + \dots$  and  $x_i$  are complex number coefficients.

$(X(e^{j\omega}))^*$  represents the Hermitian transpose of  $X(e^{j\omega})$  in the complex plane.

A rational filter is an all-pass filter if it satisfies

$$(4.2.15) \quad \tilde{H}(z) \cdot H(z) = 1$$

In the domain of FIR filters the only filters which satisfy 4.2.15 are pure delays, ie. where  $H(z) = z^{-n}$ .

From Figure 4.2.1 the decomposition filters are  $H_0(z)$  and  $H_1(z)$  and so the matrix in 4.2.4 completely defines the decomposition properties of the filterbank.

To satisfy Equation 4.2.5 we can choose  $\tilde{H}_0(z) = H_1(z)$  and  $\tilde{H}_1(z) = -H_0(z)$  which satisfies the requirements for alias cancellation. Equation 4.2.3 becomes

$$(4.2.16) \quad \begin{aligned} \hat{X}(z) &= \frac{1}{2} (H_1(-z) H_0(z) X(z) - H_0(-z) H_1(z) X(z)) \\ &= \frac{1}{2} X(z) (H_1(-z) H_0(z) - H_0(-z) H_1(z)) \end{aligned}$$

**4.2.2.1. Polyphase Representation.** If we consider the polyphase representation then we can express the filters in the filterbank in the downsampled domain as

$$H_i(z) = H_{i0}(z^D) + z_1^{-1} \cdot H_{i1}(z^D)$$

and the polyphase matrix associated with the decomposition filter bank is given by

$$H_p(z) = \begin{pmatrix} H_{00}(z) & H_{01}(z) \\ H_{10}(z) & H_{11}(z) \end{pmatrix}$$

Here  $z^D = \begin{pmatrix} z_1 & z_2 \end{pmatrix}^D = \begin{pmatrix} z_1^{D_{00}} z_2^{D_{10}} & z_1^{D_{01}} z_2^{D_{11}} \end{pmatrix}$  and with  $D = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$  we get  $z^D = \begin{pmatrix} z_1 z_2 & z_1 z_2^{-1} \end{pmatrix}$ .

When the polyphase matrix is paraunitary the filterbank provides perfect reconstruction with identical analysis and (reversed) synthesis filters. From Equation 4.2.14 there are some restrictions on the filterbank [101].

- The reconstruction filters are derived from the decomposition filters by reversal and modulation

$$H_1(n_1, n_2) = (-1)^{n_1+n_2} \cdot H_0(n_1, n_2)$$

so the polyphase matrix completely defines or is defined by the filters used in the entire filterbank.

- Both polyphase components have to be of the same size; therefore perfect symmetry in up/down, left/right and diagonal directions simultaneously isn't possible. For proof of this see [101, appendix A].
- The polyphase components have to satisfy the orthogonality condition  $H_{00}(z) H_{00}(z^{-1}) + H_{01}(z) H_{01}(z^{-1}) = 1$

**4.2.3. Design of KV 5/3 filters.** Here we first replicate the process of design for the Kovacevic and Vetterli 5x5/3x3 linear-phase filterbank used in [101] and [56]. We then note some deficiencies of this initial design and present others which may be of value.

We start with the filterbank shape given by

$$(4.2.17) \quad h_0 = \begin{pmatrix} a & & \\ 1 & -a_1 & 1 \\ & a & \end{pmatrix}$$

$$(4.2.18) \quad h_1 = \begin{pmatrix} & \frac{ad}{a_1} & & & \\ & a + \frac{d}{a_1} & d & a + \frac{d}{a_1} & \\ 1 & a_1 & a_2 & a_1 & 1 \\ & a + \frac{d}{a_1} & d & a + \frac{d}{a_1} & \\ & & \frac{ad}{a_1} & & \end{pmatrix}$$

We begin the design with  $h_0$ , the analysis high-pass filter. For  $h_0$  to be a high-pass filter we require that if  $h_0(z) = \sum h_0(k_1, k_2) \cdot z_1^{k_1} z_2^{k_2}$  then

$$(4.2.19) \quad \sum_{k_1, k_2} h_0(k_1, k_2) = 0$$



We aim for additional circular symmetry in the filters to make them as isotropic as possible. Following this goal we then set  $a = 1$  in  $h_0$  in Equation 4.2.17. Satisfying Equation 4.2.19 requires that  $a_1 = 4$  which completely defines  $h_0$ .

We now turn our attention to  $h_1$  - the synthesis high-pass filter. As with  $h_0$  we aim for circular symmetry so we set  $d = a_1$  and  $\frac{ad}{a_1} = 1 \Leftrightarrow d = \frac{a_1}{a}$ . We find then that Equation 4.2.18 becomes

$$(4.2.20) \quad h_1 = \begin{pmatrix} & & 1 & & \\ & a + \frac{a_1}{a_1} & a_1 & a + \frac{a_1}{a_1} & \\ 1 & a_1 & a_2 & a_1 & 1 \\ & a + \frac{a_1}{a_1} & a_1 & a + \frac{a_1}{a_1} & \\ & & 1 & & \end{pmatrix}$$

and by noting that  $a = 1, a_1 = 4$  from the design of  $h_0$  we arrive at  $a_2 = -28$  and the complete filter without normalization is

$$(4.2.21) \quad h_1 = \begin{pmatrix} & & 1 & & \\ & 2 & 4 & 2 & \\ 1 & 4 & -28 & 4 & 1 \\ & 2 & 4 & 2 & \\ & & 1 & & \end{pmatrix}$$

This almost completes the design of the filterbank. We can now create the low-pass filters  $g_0$  and  $g_1$  from the high-pass bank by using the QMF relationship for biorthogonal filters. This then gives us

$$g_0 = \begin{pmatrix} & & 1 & & \\ & 2 & -4 & 2 & \\ 1 & -4 & -28 & -4 & 1 \\ & 2 & -4 & 2 & \\ & & 1 & & \end{pmatrix}$$

and

$$g_1 = \begin{pmatrix} & -1 & & & \\ -1 & -4 & -1 & & \\ & -1 & & & \end{pmatrix}$$

It is now that we must note that  $\sum_{k_1, k_2} g_0(k_1, k_2) = -32$ . This gives us our normalization factor for the 5x5 filters. Similarly  $\sum_{k_1, k_2} g_1(k_1, k_2) = -8$  gives our normalization factor for the 3x3 filters. So our final filter-sets are

$$(4.2.22) \quad g_0 = \frac{1}{32} \begin{pmatrix} & & -1 & & \\ & -2 & 4 & -2 & \\ -1 & 4 & 28 & 4 & -1 \\ & -2 & 4 & -2 & \\ & & -1 & & \end{pmatrix}, \quad g_1 = \frac{1}{8} \begin{pmatrix} & 1 & & \\ 1 & 4 & 1 & \\ & 1 & & \end{pmatrix}$$

$$(4.2.23) \quad h_0 = \frac{1}{8} \begin{pmatrix} & & -1 \\ & -1 & & & \\ -1 & 4 & -1 & & \\ & & & -1 & \\ & & -1 & & \end{pmatrix}, \quad h_1 = \begin{pmatrix} & & & -1 & \\ & -2 & -4 & -2 & \\ -1 & -4 & 28 & -4 & -1 \\ & -2 & -4 & -2 & \\ & & & -1 & \end{pmatrix}$$

Looking at the shape of the 5x5 filters we note that the central tap of magnitude 28 is 7 times the magnitude of any of its neighbours – or indeed any other taps in the filter. This is a common occurrence in the design of filters in multiple dimensions. The effect of this massive central tap is that it reduces the smoothness of the filter response. This becomes a problem when we begin to integrate coding frameworks with these filterbanks. The 3x3 filters are reasonably well behaved and result in some energy compaction. The main problem that occurs is due to the 5x5 filter which is the low-pass synthesis filter  $h_1$ . When a wavelet decomposition is performed and the detail signals are harshly quantized, often there is little information left in the decomposition apart from the approximation signal. When this is passed back through the synthesis low-pass filter the massive central tap of  $h_1$  dominates the response. The reconstruction is not smooth and without the balancing effect of other detail signals the reconstruction is quite distorted. The effects of this are demonstrated in Figure 7.3.9. In the next section we produce designs which reduce the size of the central tap with respect to surrounding taps in an attempt to smooth the synthesis high-pass response.

**4.2.4. Modified KV5/3 filterbank.** The filters shown above in Equations 4.2.22 and 4.2.23 are the only solution possible which have circular symmetry. In order to pursue other filter designs we have to abandon this property.

We begin with the analysis high-pass filter  $h_0$  again. This time we will pay attention to the effects on the synthesis high-pass  $h_1$  at the same time. Looking at  $h_1$ , it is high-pass so the sum of all taps must be zero. We will allow the central tap  $a_2$  to be the dependent value. The value of  $a_2$  will be the sum of all the other tap values in the filter so if we reduce the other taps then we reduce  $a_2$ . This is of little use however since we are not concerned with the magnitude of the central tap but rather its magnitude when compared to the other taps, ie. the ratio of  $a_2$  to its neighbour taps.

The structure of the filters is such that it is difficult to obtain a good solution. If we consider the distance between the central tap and its immediate neighbours after normalization then we find that by reducing the difference in one dimension we increase the difference in the other. The process can result in wildly non-isotropic filters.

4.2.4.1. *Design 1.* Circular symmetry is not possible so make the analysis high-pass

$$h_0 = -\frac{1}{6} \begin{pmatrix} & \frac{1}{2} & \\ 1 & -3 & 1 \\ & \frac{1}{2} & \end{pmatrix}$$

From this point designs can vary wildly. We sacrifice good high-pass in the vertical direction and set  $d = 1$  resulting in

$$h_1 = -\frac{1}{8} \begin{pmatrix} & -\frac{1}{6} & & \\ & \frac{1}{6} & -1 & \frac{1}{6} \\ 1 & 3 & -6\frac{1}{3} & 3 & 1 \\ & \frac{1}{6} & -1 & \frac{1}{6} \\ & & -\frac{1}{6} & \end{pmatrix}$$

For the analysis low-pass (not shown) in the horizontal dimension this filter behaves quite well as the central tap is not much greater than its neighbours. However in the vertical dimension we see that the central tap is actually of opposite sign to its neighbours.

Results obtained with this design showed some small improvement over the circular design under appropriate conditions. In particular the dotting effect which results from the large central tap of the synthesis low-pass tempered slightly. However as more detail becomes available the artifact due to the orientation specific nature of the filter can manifest as noticeable distortion.

### 4.3. Lifting

The lifting scheme emerged originally as a means of creating new wavelets from existing wavelets [88], [90] - providing a much needed design tool. It was then discovered that lifting was a general tool which could be applied to the entire process of designing biorthogonal filterbanks, not just in one-dimension but in arbitrary dimensions.

For a quick introduction to the lifting scheme see [89], [96], [95]. More in depth discussion can be found in [54], [53], [88], [90] and [16].

The lifting scheme stems from a simple observation that a wavelet transform as applied with a filtering operation performs two basic steps.

- An approximation to the original signal is determined at a lower resolution than the original.
- An error signal is determined which reflects those details in the original signal which cannot be recovered from the approximation signal.

The lifting scheme approaches this problem from a different perspective. First the signal is split into its two cosets. Then one coset is used to predict the other, thus the first coset is the approximation. The difference between the predicted second coset and the actual second coset becomes the details signal. The first

coset is then updated from the error signal such that it preserves the mean value of original signal.

The basic principle is that one signal is used to predict the other and then the error is used to adjust the first signal. This can be applied to any situation where there are two signals of this nature. The process is also always reversible as long as the predict and update operations can be reversed.

The process can be applied to the operators as well as the signal which allows us to produce new wavelet filters from existing wavelets. This is an important property which could lead to important developments in the nonseparable domain where design by Fourier transform is difficult. Indeed the lifting scheme would be an excellent method by which to build upon the limit library of existing biorthogonal filters.

## Upsampled Representations and Cross-scale Equivalence

Zero-tree [81] and quad-tree [64] coding of Wavelets has provided one of the most successful image coding approaches in recent times[61]. However it has been confined to the separable domain because of its dependence on multiscale inheritance. The most important features to human vision, edges, are multiresolution in nature, they occupy the same spatial location throughout several resolution levels. The utilization of this continuity of features across scales leads to efficient coding strategies.

This Chapter shows how to determine such multiresolution inheritance on the quincunx lattice. We introduce a novel implementation for wavelet decomposition and resampling using the quincunx lattice which can be extended to other sampling lattices and reduces and in some cases completely removes the computational cost of resampling signals at each step in a multiresolution decomposition. This removes a significant obstruction to the use of nonseparable decompositions in real-time processing. A comprehensive introduction to Wavelets in  $\mathcal{R}^n$  is given in [56] and coding comparisons between separable and quincunx decompositions are given in [4], [5].

### 5.1. Quincunx Wavelet Transform

The Fast Wavelet Transform is performed through a process of filtering and downsampling. Filtering is done by a low-pass filter  $H$  and a high-pass filter  $G$ . The downsampled high-pass signal contains the wavelet coefficients or details and the low-pass signal is further decomposed until a desired depth of decomposition is achieved.

It is the detail signals which are coded to achieve image compression. Wavelets are very effective for energy compaction and the detail signals are typically quite sparse leading to efficient compression.

Figure 5.1.1 shows two levels of Wavelet decomposition using the some general resampling and Figure 5.1.2 shows the coordinate transform which occurs during quincunx sampling. A second quincunx transformation will return the image to its normal orientation, but at half its original size.

Whereas separable downsampling removes half the samples from one-dimension at a time, the quincunx lattice cannot be separated into such a scheme. Figure 5.1.3 shows the quincunx and separable sampling lattice. The small arrows indicate the offset vectors of the various cosets from the origin. The Quincunx lattice has only two cosets, but the separable lattice has four cosets.

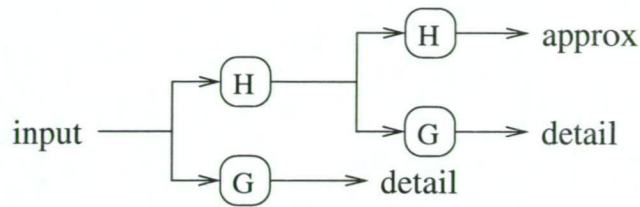


FIGURE 5.1.1. Quincunx subband decomposition block diagram showing two levels of decomposition. The H and G operators include all resampling necessary to complete the step.

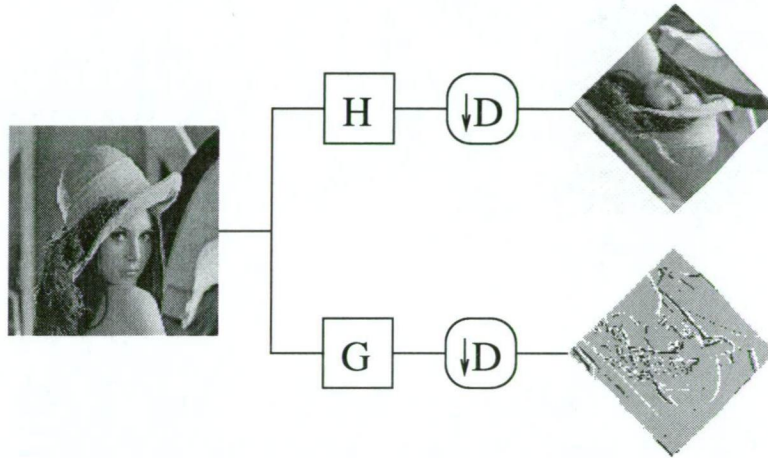


FIGURE 5.1.2. Image coordinate transform during Wavelet decomposition using the quincunx lattice. Unlike separable resampling, Quincunx resampling does more than scale the signal. The signal is rotated  $-45^\circ$  and flipped about the horizontal axis.

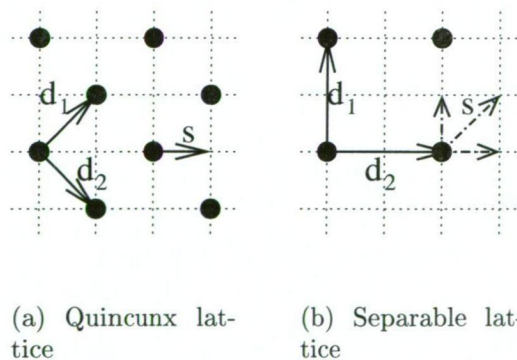


FIGURE 5.1.3. Quincunx and separable sampling lattices.  $d_1$  and  $d_2$  are the directional vectors which make up the subsampling lattice. In the separable case  $d_1$  and  $d_2$  have the same direction as the directional vectors of the original Cartesian lattice. The small arrows indicate the shifts to the missing cosets, ie. the displacement vectors to the cosets not shown.

## 5.2. Equivalence Across Scales

The Fast Wavelet Transform is a multiresolution process which decomposes a signal into components of various scales, starting from the finest detail and ending at the coarsest features. Each level of decomposition involves resampling a filtered signal to a lower resolution (decimation). This necessarily involves a transformation of the coordinates of the samples, as the same sampling density must be retained. In the separable case this transformation is a trivial division by two in each dimension, but for the quincunx lattice the transform is more complicated. We describe the coordinate transform in terms of a matrix operation discussed in Equation 3.4.1:

$$(5.2.1) \quad x_{j+n} = D^n \cdot x_j$$

where  $x_j$  is the location of the sample at resolution level  $j$  ( $j$  increasing for higher resolutions) and  $D$  is the sampling matrix which defines the lattice. In the case of the quincunx lattice the defining matrix<sup>1</sup> is  $D = \begin{pmatrix} d_1 & d_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ . It should be noted (from Section 3.1) that  $D^2 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$  which gives separable sampling by 2 in two-dimensions.

We assume that, within the current sampling lattice, a lattice point is the reference for a pixel whose area is the unit cell with an origin at that lattice point, ie. the value of the sample on the lattice point defines the value over the entire area of the pixel. This is similar to a piecewise continuous definition of a function for discretization.

To determine the child samples enclosed within a pixel boundary from a lower resolution level (parent pixel) we must first upsample the coordinates of its reference lattice point ie.  $x_{j+1} = D \cdot x_j$ . In the upsampled domain this point will reside on the first coset of the lattice. Within the boundaries of the parent pixel, two pixels are contained at the current resolution level: the pixel referenced by the upsampled location of the parent lattice point (at  $D \cdot x_j$ ), and its immediate neighbour, as determined by the sampling coset shift vector  $s$ , (at  $D \cdot (x_j + s)$ ). For  $D$  as defined above  $s = (1, 0)$ . See also Figure 5.1.3.

When taken to the next resolution level, the problem reduces to separable downsampling because of the chosen definition of  $D$ . When covering multiple resolution levels this simple method no longer holds because we add extra cosets as we upsample. We can only predict the locations of the first coset without modifying our method. This is complicated and generally not necessary so it is not covered here. Instead we use the inverse result to solve the general problem in this case.

---

<sup>1</sup>This is not the only possibility for  $D$  but this one leads to separable sampling at every other step and retains spatial location significance.



The inverse problem, ie. determining which pixel  $x_0$  in level 0, is the parent of  $x_n$  in level  $n$ , is much simpler. Because we are moving in the downsampling direction we can simply use

$$x_j = \text{floor}(D^{-n} \cdot x_{j+n})$$

where  $\text{floor}(x)$  is the operator which returns the nearest integer value less than or equal to  $x$ .

This allows us to predict the parent pixel  $n$  resolution levels below the current level. It also gives an alternative for finding child pixels of a low resolution parent by performing a search of the high-resolution image for all pixels with the correct parent.

### 5.3. Upsampled Representation

In this section we introduce the concept of the upsampled representation and show the advantages it has over traditional methods of resampling.

Consider the process of downsampling a signal. Samples from one coset are retained while samples from the unwanted coset(s)<sup>2</sup> are discarded, then the coordinates are transformed to maintain sampling density. Often we don't wish to move the pixels from their location within the image but it is apparent that the coordinate transform which occurs during downsampling (required to maintain sampling density) forces this to occur. An inverse relationship is then necessary to determine their original position in the image, ie. to upsample them back to their original position.

From Figure 5.1.2 we can see that the location of pixels within the downsampled image is different from those in the original. The calculation of the coordinate transform responsible for this change of basis is computationally expensive. However, nonseparable downsampling and further decomposition can be performed without any complicated coordinate transform by using the upsampled representation presented here.

We concentrate on the quincunx example. For the downsampling process we can simply remove those pixels from the unwanted coset and leave the retained coset untouched, this is the central idea behind the process of downsampling followed by upsampling as discussed in Section 3.4.3. The author borrows the term "upsampled" from the Kovacevic and Vetterli papers on polyphase filter representations.

We refer to a signal which has had one coset removed but has not undergone coordinate transformation as a downsampled signal in the *upsampled* domain. This results in an image of the original size and orientation but where one coset is missing (the discarded coset samples). In this way the spatial relationship

---

<sup>2</sup>For quincunx there are only two cosets thus one is retained and one rejected, however separable has four cosets and three are rejected.

between samples at different decomposition levels is easily determined because their location is not altered from that of their parents and children.

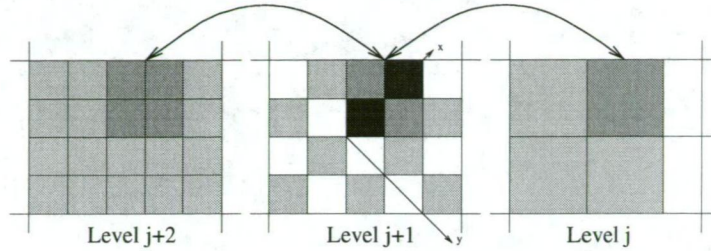


FIGURE 5.3.1. Two levels of pixel overlap with quincunx downsampling, in the upsampled domain. In this case the second resampling step elects to down-sample separably.

Figure 5.3.1 shows the two child pixels per parent and the axes of the downsampled domain. We can see that the two children of a parent are

- (1) the child pixel with the same lattice point as the parent; and
- (2) the child pixel removed from the parent lattice point by the *upsampled* coset shift vector, which for the quincunx case is given by  $\uparrow s = D \cdot s = (1, 1)$ .

This is consistent with our previous representation where in the downsampled domain the downsampled shift-vector applies.

The spatial relationships between samples at different scales should now be apparent from Figure 5.3.1. Although quad-tree coding is specifically designed for the four-coset separable decomposition it now becomes possible to design and implement similar coding schemes for non-separable sampling structures.

In its original form [81] zero-tree coding is also specific to separable sampling but this is an implementation issue and does not preclude a modified version being developed for other sampling structures using the same principles. In Chapter 8 we develop a version of zero-tree coding for the quincunx lattice and demonstrate it as a powerful coding method.

**5.3.1. In-Place Decomposition.** Wavelet decomposition is achieved by filtering then downsampling then filtering then downsampling, etc. Thus the filters remain the same (one of the key properties of the Fast Wavelet Transform) but at each stage of decomposition the image has been resampled and thus has undergone a coordinate transformation. We can describe this by

$$X_{low_1}(n) = D^{-1} \cdot H \cdot X(n)$$

where  $X_{low_1}(n)$  is the low-pass version of  $X(n)$  after one level of decomposition (filtering by  $H$  followed by downsampling by  $D$ ).

Examining the low-pass branch we can see that the second level low-pass signal is achieved by two sequential filter-subsample combinations as shown in Figure 5.3.2.

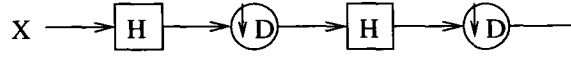


FIGURE 5.3.2. Normal nonseparable wavelet transform low-pass branch showing two levels of decomposition. The input is filtered then resampled then filtered then resampled.

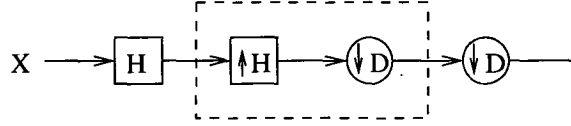


FIGURE 5.3.3. Low-pass branch using upsampled filter operator  $\uparrow H$  operator.

Figure 5.3.3 shows an equivalent method of performing the same operations in the upsampled domain. Here  $\uparrow H = D \cdot H$  is a filtering operator derived from  $H$  by a process of upsampling. The  $\uparrow H$  operator performs the exact same filtering operation in the upsampled domain as  $H$  does in the downsampled domain. The resulting filtered sample values only exist on the retained coset.

The upsampled filters are described by Kovacevic and Vetterli in their work on polyphase transforms such as [47]. We write

$$F(z) = F_0(z^D) + \underline{k}_1 F_1(z^D) + \cdots + \underline{k}_{M-1} F_{M-1}(z^D)$$

where  $F(z)$  is the overall filter in the downsampled domain.

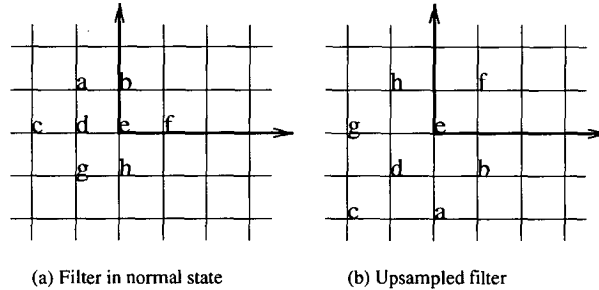


FIGURE 5.3.4. An arbitrary filter with 8 taps shown in (a) its normal state and (b) its (first) upsampled state.

In the downsampling operation only one coset is retained, thus the samples in the unwanted coset cease to exist. Normally sampling density must be maintained and the image size shrinks and the sample location coordinates are transformed to account for this. In the upsampled representation we change the sampling density by removing a coset but avoid the need for transforming the coordinates. Because we have violated the sampling density conditions we must now accordingly change the sampling density (ie. coordinate transformation) of the samples in the filter used in subsequent operations. This is equivalent to downsampling the signal then upsampling it again, and also upsampling the future filter to compensate.



The next convolution operation which is performed in the upsampled domain will result in the same values as in the downsampled case - the only difference is that the source and destination locations have undergone an upsampling coordinate transform.

Data locations of samples on the discarded coset are neither used in subsequent filtering operation nor are they changed in any way by future operations. This allows us to use these locations for other purposes such as storing detail signals from a wavelet transform step. As a result of critical sampling we have exactly the number of locations to store the entire decomposition.

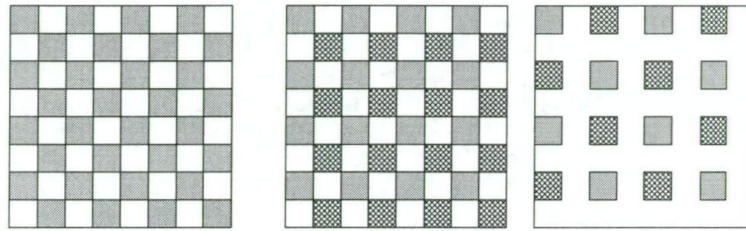


FIGURE 5.3.5. Organization of cosets at different resolution levels within the upsampled structure. The transform progress from left to right. The white squares represent locations which have been used to store details of the higher-resolution levels.

This representation is very efficient in the amount of memory required to perform a wavelet transform. Under normal circumstances with non-separable resampling the amount of storage required increases due to the lost areas of storage around an image as seen in the downsampled signals in Figure 5.1.2. The upsampled representation eliminates this wastage.

After filtering for the second time (by  $\uparrow H$  in Figure 5.3.3) the retained coset locations now contain the same samples as after the second filtering operation by  $H$  in Figure 5.3.2 but they are still in their original positions within the upsampled lattice. Two quincunx downsampling operations are then required to return these samples to their usual downsampled coordinates. Using the quincunx downsampling matrix  $D$  from Equation 5.2.1 results in separable downsampling at every second step. Hence two consecutive quincunx downsampling operations results in separable downsampling.

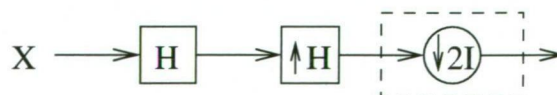


FIGURE 5.3.6. Low-pass with upsampled operator and separable downsampling replacing two consecutive quincunx down-sample operations.

Figure 5.3.6 shows the resulting implementation using separable downsampling with quincunx filters. Note that we can continue to further upsample the upsampled operators and keep the signal samples in their original locations, leaving all the downsampling operations until the very end, or not perform them at all and just use upsampled storage for the entire decomposition.

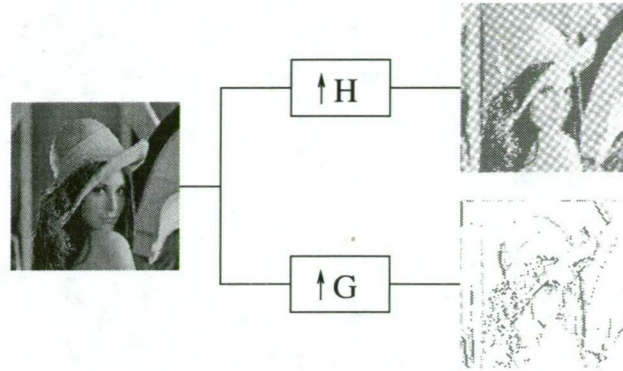


FIGURE 5.3.7. One level of quincunx decomposition performed in the upsampled domain. The orientation is the same in the downsampled and original signals. Each downsampled signal contains half as many coefficients as the original allowing both components to be stored in the space occupied by the original. The low and high-pass results can be stored in the original image space.

Figure 5.3.7 shows the resulting low-pass and detail spaces from one level of quincunx wavelet transform in the upsampled domain. Note that although the filtered images occupy the same space as the original, each has half the number of samples (critical sampling is preserved). Due to the sample shift operator incorporated into the high-pass filtering operation, the high-pass signal occupies a different coset than the low-pass. Thus the low-pass and detail can be interleaved within the original image storage with the low-pass occupying coset 0 and detail in coset 1 as outlined in Figure 5.3.5. For subsequent operations, more points originally occupied by the approximation are converted for use by detail signals.

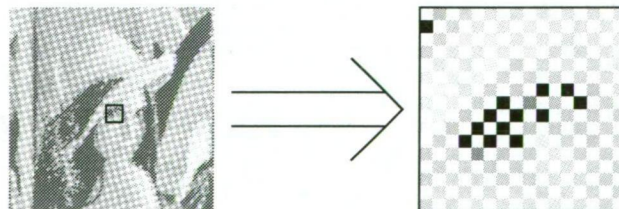


FIGURE 5.3.8. Detailed view of Lenna's eye from figure 5.3.7. This clearly shows how all the approximation values exist on a single quincunx coset.

Figure 5.3.8 shows a detail which should exemplify the existence of only one coset in the downsampled images. In this case it is the approximation signal - we store the detail signal in the coset which is not occupied by the approximation signal.

An effect which occurs when a quincunx filter is upsampled is that many zero coefficients are inserted into the filter matrix. This is analogous to the conventional upsampling of a signal where zero-valued coefficients are used to fill in the missing coefficients prior to filtering. With quincunx upsampling the size of the filter matrix may grow by a factor of 2 in each direction to accommodate the position of the upsampled coefficients. As a result the upsampled filters become very sparse with the majority of coefficients being zero. Under these conditions conventional convolution methods are not efficient as they involve many wasted calculations which sum to zero simply because of the sparseness of the filters. In these situations it is advantageous to adopt a convolution method which addresses only those coefficients which are non-zero. This allows for convolution to be calculated in the same order of time as for the non-upsampled filters.

**5.3.2. Computational demand.** Normal quincunx downsampling requires every pixel to be transformed to a new coordinate position at each decomposition level, requiring 6 additions per pixel (for a quincunx specific algorithm). Using the upsampled domain we can perform the process at a fraction of the computational cost. With the upsampled method we only transform the coordinates of the filter being used and if we store an upsampled version of the filter this cost can be removed as well. The downsampling process is reduced to the same computational cost as separable sampling. If all computations are carried out in place then all signal resampling cost can be removed. In this way it is possible to significantly reduce the computational burden of using nonseparable sampling schemes in wavelet transforms which removes one of the significant barriers to nonseparable transforms in real-time processing.

**5.3.3. Note on other lattices.** Although the upsampled representation was presented for the quincunx lattice, the concept is equally applicable to any resampling lattice, including separable. The transform is required to be critically sampled however so that the total number of samples in the decomposition is constant regardless of the state of the decomposition.

The upsampling process on the separable lattices is particularly easy, it is merely a matter of expanding the distance between filter coefficients by a factor of two.



## CHAPTER 6

### Coding Frameworks

In this chapter we review the coding methods which exist for compression of digital data. We concentrate on those methods which have been used in the production of results in the course of this research, ie. for lossy image compression.

We also review psychovisual phenomena of the HVS and explain how they can be exploited to achieved better perceived image quality from lossy compression. We derive some results for the quincunx decomposition relating to subband selectivity and explain how these values can be used in coding frameworks to increase perceived image quality. Basic statistical results of quincunx and separable transform data is presented and analysed.

#### 6.1. Quantization

Quantization is central to lossy coding methods. It has many varied forms but the central purpose remains the same, to reduce the digital data requirement for representing data by reducing the precision at which numerical values are expressed. Numerical values can be treated as scalar or vector depending on how values relate to each other. If some numerical values are closely related then it makes sense to consider the combination of values as a single conceptual value which can then be quantized in some manner which is meaningful to the object. This is the basis of vector quantization.

When numerical values are discrete and largely unassociated then scalar quantization is more sensible. An excellent review of quantization procedures with a more rigorous analysis of individual methods can be found in [30]. This section provides an overview of the types of Quantization used in the production of this work and how they relate to the coding of wavelet transform data for image compression.

**6.1.1. Scalar Quantization.** For a long time scalar quantization has been the backbone of compression techniques. Scalar quantization takes a single numerical value and approximates it by using one of a set of discrete intervals, the reconstructed value is some value in the interval representative of all possible values contained. The number of allowed quantized values is generally small and so the number of bits required per value can be reduced significantly. This process is especially effective when used in conjunction with entropy based coding such as Huffman coding [40], [71]. In this situation, generally a significantly large proportion of coefficients will be quantized to zero, thus giving that value a significantly



higher probability than the other quantized values. Hence allocation of a smaller (in number of bits) codeword will lead to significant data saving.

Uniform scalar quantization is very common in signal coding. It is the easiest quantization mechanism to implement. Uniform quantization simply divides the valid range of values into equally spaced regions of equal size (except possibly the centre area where values are approximately zero). Every original numerical value is represented by the index of the region within which the value falls. Because in many situations it is overwhelmingly common for values to be close to zero, the region surrounding the zero value within which all values are quantized to zero, is twice the size of all the other regions. The region which quantizes values to zero is often referred to as the dead zone.

Increasing the commonness of one particular value in a coding mechanism is an effective way of reducing the information cost of encoding the data because if any value is more common than any other value then this fact can be used to create statistically-based coding methods which will further reduce the bit-rate. The entropy of a quantized data set determines the minimum information required to code the data without a-priori knowledge. This will be discussed further in Section 6.2.

**6.1.2. Vector Quantization.** Vector quantization [26], [27], [14], [15] differs from scalar quantization in that several numerical values are considered simultaneously as a single *vector*. A vector in this case is a conceptual term for a set of values which should be considered together to contribute to some meaning. For instance when considering 3-axis colour representations such as RGB on a digital display device, we can effectively quantize the colour by considering how the red, green and blue values all contribute to a single colour vector. In this way we can apply information about our understanding of what information is most important in a colour value to better quantize the overall value (eg. [59]).

Vector quantization is the generalization of this idea. We can often more effectively quantize a signal by applying knowledge about how several components interact as a single vector.

A common method of vector quantization which is analogous to uniform scalar quantization is uniform volume vector quantization. In this method the hyper-volume which describes the valid range of vectors is divided up into equally spaced quanta of equal value. Each region has its own index and any vector value which falls within a particular region is represented by the index representative of that region of the valid range. The various regions of the valid range is broken up in some way and the relationship between the regions of quantization and the quantized values is called a codebook.

Although uniform volume is a simple, easy to implement method of vector quantization it often does not offer any advantages over scalar quantization because it uses an ad-hoc quantization scheme. Another method of performing vector

quantization is by error minimization. A codebook of vectors is somehow assembled and quantization proceeds by approximating the value vectors to the vectors in the codebook by finding the result which minimizes some error metric.

Given a large sample population it is possible to derived a codebook which minimizes the overall error metric in the general situation.

## 6.2. Entropy Coding

Quantization is only part of an efficient coding scheme. Entropy coding takes advantage of the relative probabilities of different symbols occurring and assigns codewords of variable length to each symbol to minimize the total number of bits necessary to represent the quantized data. Obviously before entropy coding can take place we must have quantized data, otherwise there is an infinite number of possible states which a coefficient can take, and hence an infinite number of unique codewords would be needed. For a continuous range the information contained in any values is effectively infinite since it has an implicit infinite precision.

The entropy of a sequence of sample values is given by

$$H = - \sum_{\forall n} p_n \cdot \log_b(p_n)$$

where  $p_n$  is the probability of value  $n$  and  $b$  is the number of possible values in the base of units of entropy. For example if  $b = 2$  then the units of entropy is bits per symbol where a bit can have 2 possible values.  $b = 3$  means the unit of information is the tri-bit and  $b = 10$  means the unit of information is the numeral (ten possible values). Of all the possible bases available, base 2 is the only frequently used base as bits per symbol is easily transferable to required storage in digital media.

The entropy of a sequence of symbols tells us the average information cost per symbol of the sequence, so

$$total\ data = N \cdot H$$

where  $N$  is the number of symbols in the sequence and  $H$  is the average information cost (Entropy) of the symbol set. The worst case scenario is when all symbol values possible are equally probable.

**6.2.1. Huffman Coding.** Huffman coding[40] operates on a tree based system for determining codewords. It requires knowledge of the complete sequence of coefficient values which are to be coded from which to determines relative probabilities and optimized codewords. The information is then coded as

- a header specifying the table of codewords and symbols that they match;
- the data encoded in sequence using the optimum codewords determined.

Huffman codes are created in such a way that regardless of the length of a codeword the end of one codeword and the beginning of the next can always be uniquely identified.

The overhead imposed by the codeword table can reduce the efficiency of this coding method significantly. Because of this the efficiency of Huffman coding is only seen when the data set is large so that the overhead of the codeword table becomes small in comparison to the size of the data compressed. Huffman coding is the basis of many common statistical lossless coding methods such as pack, pkzip (now WinZip) and gzip used for both text and binary executable compression.

In some practical situations the Huffman codebook is agreed upon by encoder and decoder and is included in the implementation. This leads to quite successful coding methods, but it relies upon a-priori knowledge of the statistics of the information being coded. As such it is not dynamically adaptable to sources with significantly different statistical profiles.

**6.2.2. Arithmetic Coding.** Similarly to Huffman Coding, Arithmetic Coding [118], [71] assigns codewords based on the probability of occurrence of symbols. Codewords are assigned as a binary number which uniquely defines a numeric precision interval of the size of the probability of occurrence of each symbol. Very low probability means a small interval which requires more bits to uniquely define.

Again, there is an overhead which is the required table of codewords.

**6.2.3. Adaptive arithmetic Coding.** Adaptive Arithmetic Coding [46] works in a similar manner to Arithmetic Coding in that codewords are assigned based on probability intervals of the symbols. In this case, however, the probabilities of the different symbols is updated after each symbol is coded and codeword sizes can be adaptively changed to match the recent probability distribution of the symbols. Any changes in symbols used must obviously be predictable so that a decoder can correctly reconstruct the correct symbol sequence from the bit-stream. This technique can be very effective if the symbol sequence encoded has distinct sections involved which each have significantly different probability distributions. As the distribution changes the coding algorithm can adapt to maintain a close to ideal codebook without the need for encoding symbol tables.

**6.2.4. Run-Length Coding.** Run-length coding is one of the most popular techniques for lossless data reduction due to its simplicity and low computational cost. The concept behind run-length coding is to identify sections of a data stream where the same symbol is represented many times consecutively. In this case it becomes efficient to code the value of the symbol and the number of repetitions. The number of repetitions is referred to as the length of the “run”, hence the name of the coding method.

The disadvantage of this coding method is that it requires an overhead for discriminating between ordinary symbols and symbol runs. This adds to the size of the data stream for bad-case scenarios where runs of symbols are rare. One common trade-off is to only allow runs of only one particular symbol - usually

the zero value. This reduces the overhead in discriminating runs from symbols because only one more coding symbol needs to be added to the original set.

The zero value is chosen because after quantization it is typically the most common symbol in the quantized data stream. The JPEG image coding scheme makes very effective use of run-length coding in exactly this fashion. The symbol data from DCT blocks is read from the processed data in a particular order which maximizes the probability of long runs of zero-symbols occurring at a particular point in the stream. Run-length coding can then efficiently compress the consecutive zero values into just a few bits.

**6.2.5. Lempel-Ziv (Welch) Sequence Coding.** The LZ coding mechanism[121] is very efficient in cases where sequences of symbols recur with regularity. A finite codebook of sequences is stored in the codec and as each previously unseen sequence is encountered it is stored in the codebook. When a previously encountered sequence is encountered, only the reference into the codebook is coded which is typically much smaller than the cost of the entire sequence of symbols.

The Compuserve GIF image format [43] uses LZW coding [112], [70] as the basis for image data compression. It is very effective in the compression of images which are mostly constant in colour but with recurring structure such as advertising material which is text on a constant background. It is much less effective with natural images as the probability of encountering repetitive structure is very small. In this case the overhead of discriminating sequences from non-sequences requires an enormous amount of information.

### 6.3. Basic Coding Results

We present here some basic results from the quantization of quincunx wavelet decomposition data. We begin with some statistical analysis of the transform data itself and describe properties we expect to see expressed by various coding strategies based on the statistics. The three test images used were Lenna, Goldhill and Barbara. They are significantly different images and produce a good range of results for comparison of the coding methods used.

The decorrelating effect of the wavelet transform results in a large number of very small coefficients - often referred to as insignificant coefficients. These coefficients represent the failure of the low-pass projection operation to accurately predict signal values. If the wavelet shape is suitable for the signal being processed then the prediction will be good and the wavelet coefficients will be very small in magnitude. Large wavelet coefficient values are the result of sharp transitions in intensity value - such as at edges - which have a rate of change beyond that which can be represented by the projection operator. Since these cannot be predicted they create significant wavelet coefficients. Typically much of an image is slowly varying in intensity across space and only a small percentage of the image area contains significant activity at any given scale. For this reason the majority of the

image information is contained in the information of the coarser scale subbands. Since the majority of coefficients are contained in finer resolution levels, we expect that the majority of these coefficients, and thus the majority of all coefficients, will be small in magnitude. This lead to good results after quantization and is one of the key reasons for the success of wavelet encoding of images.

We use a cumulative density function (*cdf*) and a probability density function (*pdf*) to represent the distribution of wavelet coefficient values. The function  $cdf(x)$  represent the fraction of all the coefficient values in a decomposition which have a (signed) value less than  $x$ . This can be expressed as

$$cdf(x) = \frac{num(x)}{N}$$

where  $N$  is the total number of coefficients in the decomposition and  $num(x)$  is the number of coefficients in the decomposition with a value  $< x$ .

As  $x \rightarrow \infty$ ,  $num(x) \rightarrow N$  (all value must be less than  $x$ ) and  $cdf(x) \rightarrow 1.0$

As  $x \rightarrow -\infty$ ,  $num(x) \rightarrow 0$  and  $cdf(x) \rightarrow 0$ .

Cumulative densities are always monotonic in nature.

The rate of change of the cumulative density function gives the probability density of coefficients existing at some value of  $x$  (ie.  $pdf(x)$ ). The integral of the  $pdf(x)$  over some range gives the probability of any random coefficient value lying in the range, ie.

$$prob_{a,b} = \int_{x=a}^{x=b} pdf(x) \cdot dx$$

is the probability of a coefficient lying in the range  $(a, b)$ .

A good decorrelating transform will result in a very small probability of coefficients with large magnitude (regardless of the sign of the coefficient value). This leads to a lower overall entropy of the decomposition after quantization and makes scalar quantization an effective means of data reduction.

**6.3.1. Quincunx statistics.** We first examine the statistics of quincunx wavelet transforms of the test images. The wavelet transform used is 12 levels deep which leads to an approximation signal of size 8x8 pixels. This is the smallest approximation size which can be used with the filters which can be up to 5 taps wide. Most research to date has concentrated on decompositions with a larger approximation image; 32x32 and 64x64 seem to be popular numbers. We use a smaller approximation signal because it allows us to better isolate artifacts and visual effects produced by the quantization of wavelet coefficients which might be overlooked in a partial decomposition. The scope of this thesis focuses more on the differences between coding separable and quincunx decompositions rather than producing a complete image coder with the best possible quality. The choice of smallest approximation size fits well with the aims of the thesis.

The KV5/3<sup>1</sup> filter-set was used because it provides a reasonable quality decomposition with complexity similar to that of the  $D_4$  filter. Kovacevic and Vetterli presented an orthogonal filter for the quincunx lattice which was the compliment of the  $D_4$  filter. This filter was used for some early work but it was found that its highly irregular shape lead to extremely noticeable and unattractive artifacts after quantization. The KV5/3 filter set however has good image processing qualities and is derived from a separable filter set leading to good comparisons with one-dimensional filter banks.

Looking at the results we find that as expected the majority of the coefficients are of very small magnitude. The PDF shows us that the coefficient values are clustered around zero as expected from the decorrelating effect of the wavelet transform.

By examining Figure 6.3.1 we find that Lenna is likely to be the easiest image to successfully code with the quincunx transform. It has a compact coefficient distribution with most coefficients packed close to the zero value and almost none outside 20% of maximum value. This leads to effective information removal through quantization.

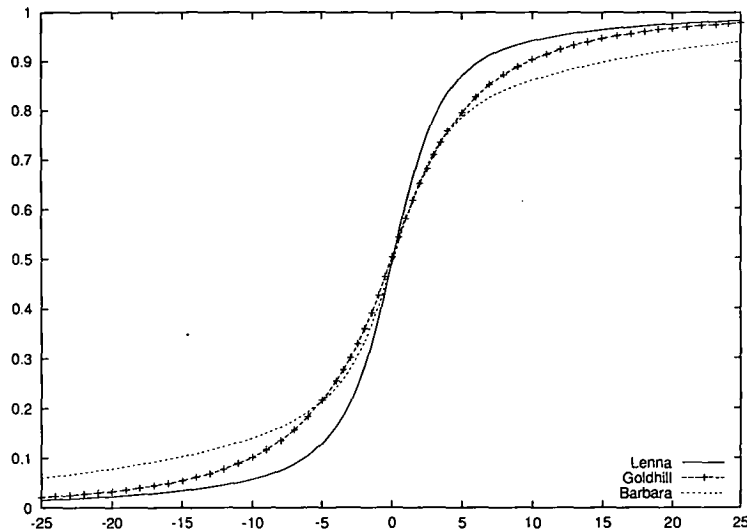
Barbara is likely to be the most difficult to code. Although it has a higher PDF peak at zero than the Goldhill image it has a coefficient distribution with a significant proportion of coefficients of large value: 15% of coefficients above 20% of maximum magnitude. The source of these significant coefficients is the sharp stripe and check patterns which dominates the image. This type of distribution is not common among natural images and the Barbara image can be considered a bad case for subband coding.

The Goldhill image contains a large proportion of vertical and horizontal features which will not be well predicted in diagonally oriented subbands of the quincunx decomposition. The low PDF of the Goldhill image at zero is a symptom of this problem. It is expected that the quincunx transform will not work well with this image because of this fact. Another feature of the Goldhill image is the disparity in image intensity of the very top and very bottom of the image. At the top is light sky and at the bottom is a dark street. This leads to problems with the circular convolution method used to process the images in that the edge of the image represents a large discontinuity which will require a considerable amount of information to code.

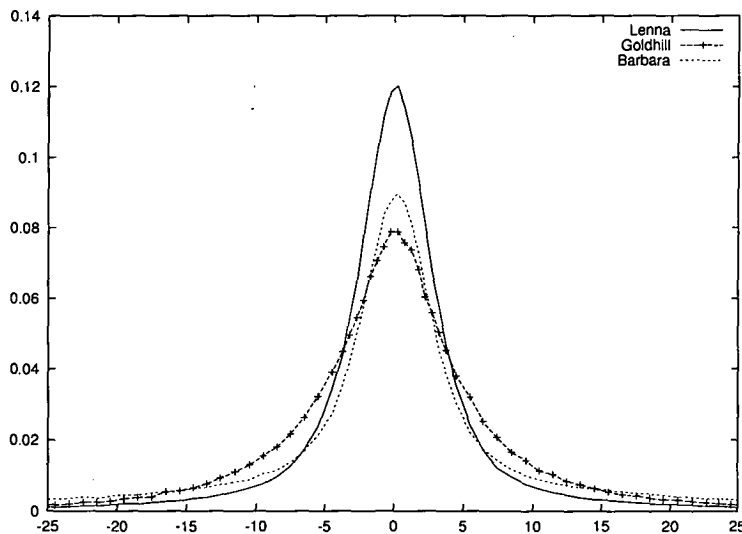
**6.3.2. Separable statistics.** The separable transform uses a 6 level decomposition with the Daubechies  $D_4$  orthogonal wavelet filters. This results in an approximation image of size 8x8; the same size as the approximation from the quincunx case. This allows for comparison of the two decompositions based as much as possible on the effects of quantizing the detail signals. These effects are

---

<sup>1</sup>The abbreviation KV5/3 is used to designate the 5x5,3x3 biorthogonal filters designed by Kovacevic and Vetterli as presented in [56].



(a) CDF of quincunx transform



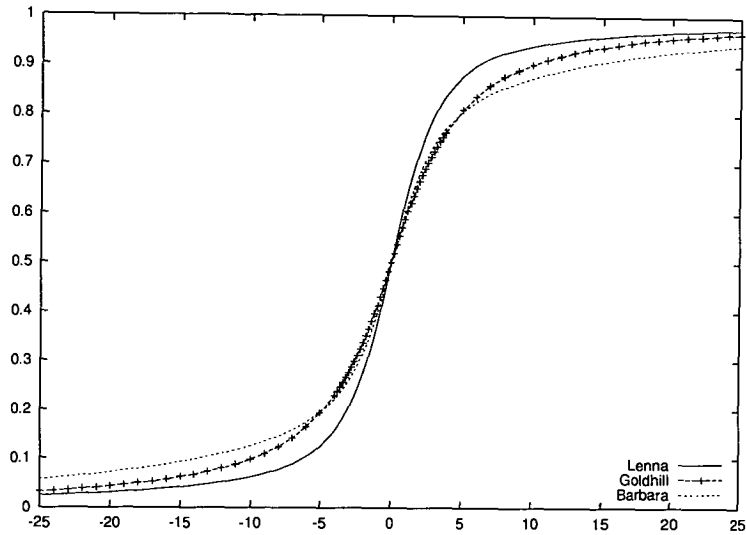
(b) PDF of quincunx transform

FIGURE 6.3.1. Plot of CDF and PDF of wavelet transform coefficients for the Lenna, Goldhill and Barbara images with a 12 level quincunx wavelet transform based on the 5/3 Kovacevic and Vetterli filter set.

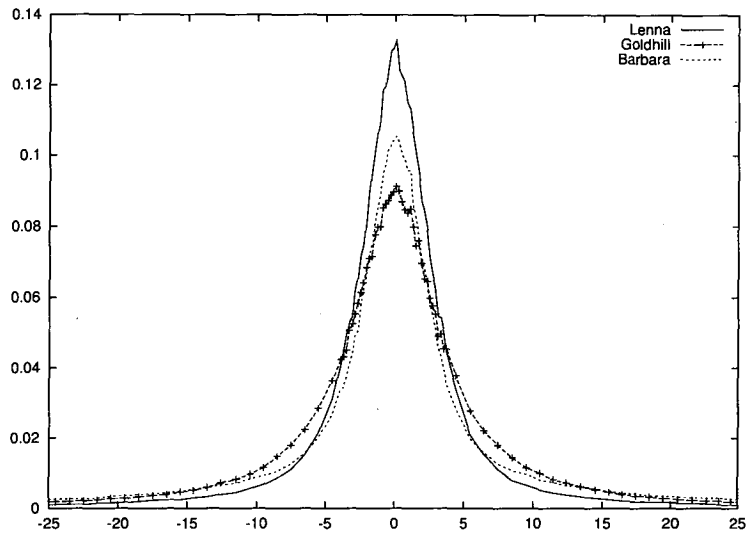
less pronounced for more shallow decompositions as more result information exists in the approximation which doesn't have to be recovered from the detail signals.

As with the quincunx results we find that the transform decorrelates the signal well. The PDF density is clustered tightly around the zero coefficient value indicating that the vast majority of coefficients are unimportant.





(a) CDF of separable transform



(b) PDF of separable transform

FIGURE 6.3.2. CDF and PDF plots of wavelet transform coefficients for the Lenna, Goldhill and Barbara images with a 6 level separable wavelet decomposition based on the Daubechies  $D_4$  orthogonal filter bank.

From Figure 6.3.2 we can see that as with the quincunx case, the Lenna image has the most compact coefficient distribution making it the most likely candidate for successful coding. As with Figure 6.3.1 the Barbara image has the least compact distribution of coefficient values.

**6.3.3. Comparison of results.** Figure 6.3.3 shows the PDFs from separable and quincunx decompositions of the three image. The trend we find is that the PDFs for the separable images are more concentrated at zero. The separable

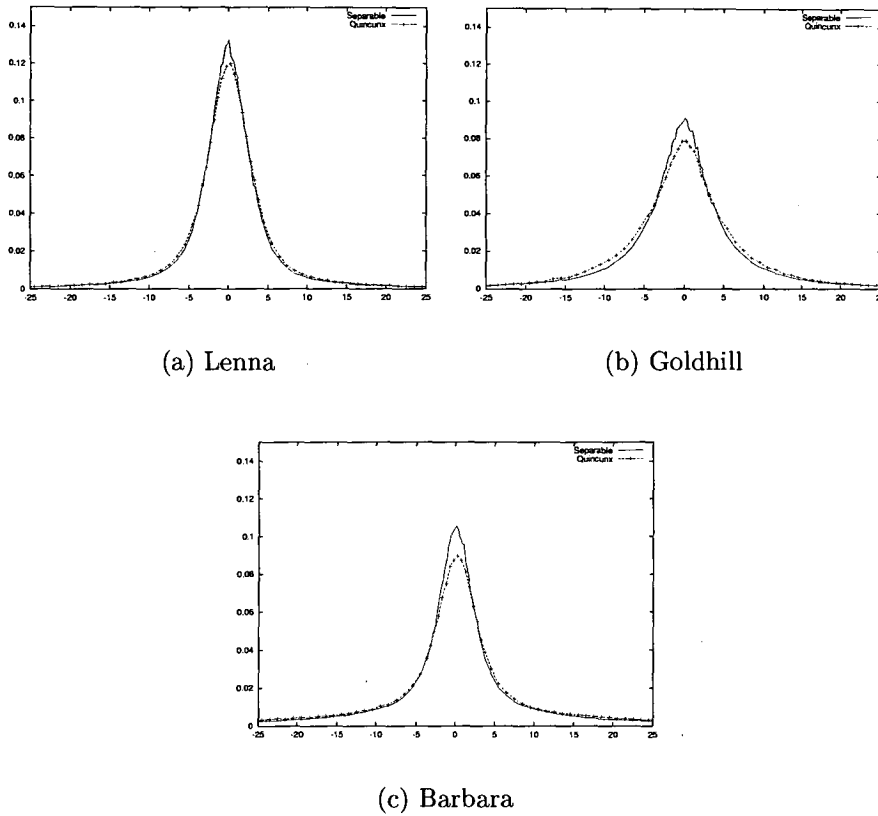


FIGURE 6.3.3. Comparative plots of the probability densities of wavelet coefficient values for separable and quincunx grouped by image. Scales have been adjusted to enhance exaggerate differences.

Lenna image reaches a peak PDF of 0.135 compared to a peak of 0.120 for the quincunx case. Similar results occur for the other test images. This indicates that the separable transform is performing a better job of decorrelating the image data than the quincunx decomposition. We can expect that the scalar quantization of coefficients will have less detrimental effect on the separable decomposition than on the quincunx decomposition.

**6.3.4. Basic Quantization Comparisons.** In this section we examine the results of using scalar quantization on wavelet decompositions in both separable and non-separable domains. We use uniform scalar quantization to code the detail signals from the wavelet decompositions.

Values of the entropy of the quantized decomposition are given as a comparative measure of the approximate coding cost of the image. Peak-Signal to Noise Ratio (PSNR) values are included for comparison of the statistical errors introduced by the various decompositions.

PSNR is calculated as

$$(6.3.1) \quad PSNR = 10 \cdot \log \left( \frac{\max \text{ value}^2}{MSE} \right)$$

where

$$(6.3.2) \quad MSE = \frac{1}{N} \sum_{n_1, n_2}^N (orig(n_1, n_2) - reco(n_1, n_2))^2$$

is the Mean Square Error of the reconstructed image (*reco*) with respect to the original image (*orig*). The max value used in Equation 6.3.1 depends on the dynamic range of the image intensities. It is typically  $2^8 - 1 = 255$  for 8-bit images.

As discussed in the Introduction Chapter, PSNR is not a particularly valid measure for perceptual quality assessment, so in general we have ignored the analysis of the PSNR values. They are included for completeness and as a general guide to the degree of distortion present in the reconstructed images.



(a) 4 bins

(b) 8 bins



(c) 16 bins

(d) 32 bins

FIGURE 6.3.4. Results for uniform quantization of separable wavelet decomposition of Lenna image using 6 levels of Daubechies  $D_4$  decomposition. Quantization to 4, 8, 16, and 32 bins is done.





FIGURE 6.3.5. Results of uniform quantization of quincunx wavelet decomposition of the Lenna image. The same quantization interval is used for all coefficients in all scales.

**6.3.5. Uniform Scalar Quantization.** Uniform scalar quantization is simple to implement and analyze. We present results here which demonstrate the different effects of uniform scalar quantization on the decomposition detail signals for separable and quincunx.

Quantization was applied over the dynamic range of the entire decomposition. By this we mean that the uniform quantizer was scaled such that the extent of the largest magnitude quantization bins was exactly sufficient to encompass the largest coefficient in the decomposition. Quantization bin size is then determined as

$$\text{bin size} = \frac{\text{range of values}}{\text{number of bins}}$$

We use a dead-zone of one bin size either side of zero. Quantized values are reconstructed to the mean value of the bin into which they were quantized. The number of bits desired for each symbol is used to determine the number of





FIGURE 6.3.6. Uniform scalar quantization of the Goldhill image with 6 levels of  $D_4$  decomposition.

possible bins. For example if we desire two bits per pixel we have four possible reconstruction values and so four bins.

We find that the entropy of the quantized image signals is considerably lower than the maximum possible value allowed by the number of bits per pixel desired. This is due to the decorrelating effect of the wavelet transform and the probability distribution of the wavelet transform coefficients as outlined in Section 6.3 above.

Figures 6.3.5, 6.3.7 and 6.3.9 shows the effects of uniform scalar quantization on quincunx wavelet decompositions. In this case a 12 level decomposition is applied based on the 5/3 Kovacevic and Vetterli filters. Quantization is uniform with the same quantization intervals used across all detail signals. The example shows 4, 8, 16 and 32 quantization bins available.

Figures 6.3.4, 6.3.5, 6.3.6, 6.3.7, 6.3.8, 6.3.9 compare the effects of scalar quantization on both quincunx and separable wavelet transformed images. Naturally as the amount of information contained in the images decreases, the quality of





FIGURE 6.3.7. Uniform quantization of the Goldhill image performed using the same method as in Figure 6.3.5.

the image decreases. The range of values was chosen to demonstrate the rapid deterioration of both decomposition techniques as the data-rate is reduced. No file sizes are quoted as there is little point considering such an end-product in this situation. The purpose of the results is to compare the change in perceived quality with size of uniform quantization bin for both separable and quincunx decompositions. Doing so we see that there is no clear inherent benefit to using either decomposition. We choose the number of quantization bins as the measure of data rate because this applies uniform scalar quantization across the dynamic range of transform coefficients without the bias which can be introduced by filter normalization factors or other scaling effects.

If we consider those images which use 8 bins we can gain an insight into the distortion which is generated by both types of decompositions. The separable decomposition results in a very “blocky” distortion, large square areas of distortion occur where there are few or no discernible features of the image. The quincunx



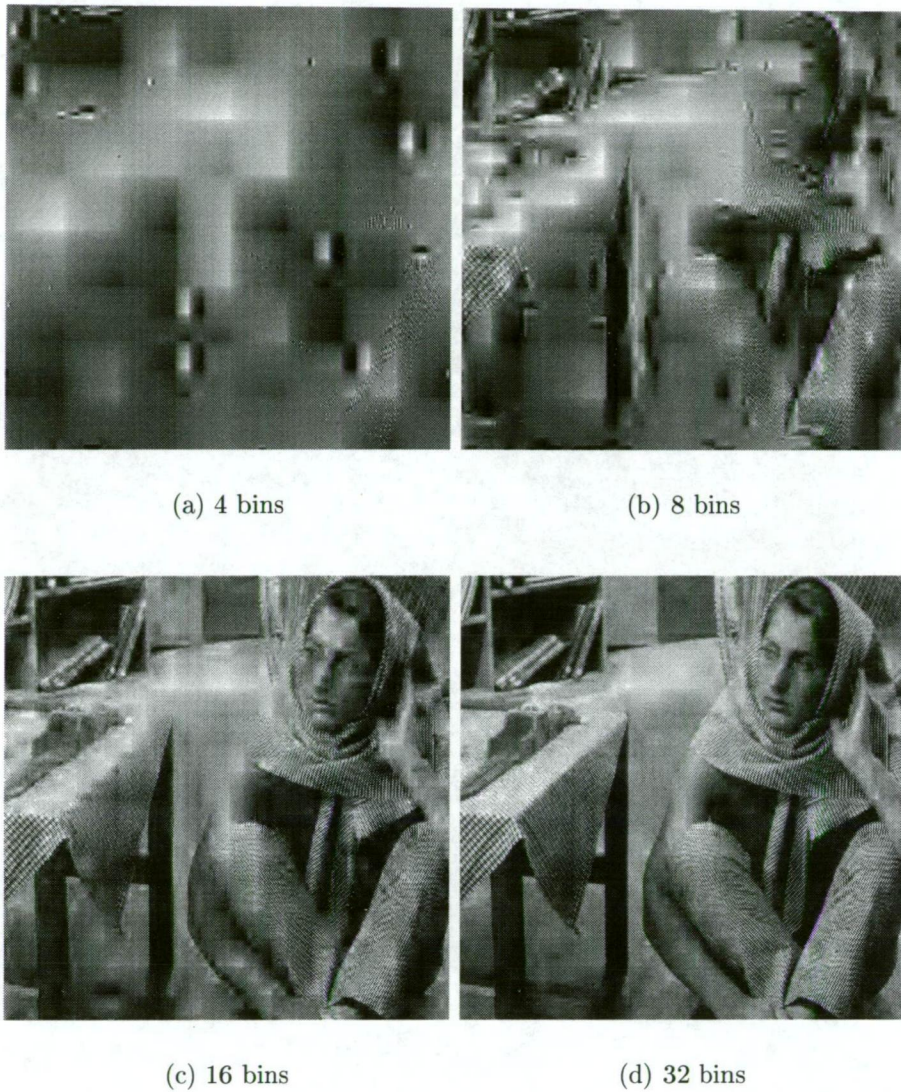


FIGURE 6.3.8. Scalar quantization of the Barbara image using 6 levels of  $D_4$  decomposition.

decomposition results in a similar state of distortion but the form of the distortion is quite different; in this case we have areas with few features, but unlike the separable case, the areas are not square but have a more rounded, smoothed appearance. This is due to the two-dimensional region of support of the filters used in the quincunx transform. Each coefficient affects an almost circular region (see the shape of the KV5/3 filters in Section 4.2.3) and the lack of a significant coefficient at low resolution can lead to large regions without signal data. For the Lenna image, the amount of discernible detail in both reconstructions is about the same. Each decomposition has particular features which it represents well. The Goldhill image with its dominant vertical and horizontal features is reproduced better under the separable decomposition, while the Barbara image appears marginally better under the quincunx decomposition with more precise and pleasing reconstructions of some significant image structures.





FIGURE 6.3.9. Uniform quantization of the Barbara image performed using the same method as in Figure 6.3.5.

bins	Lenna	Goldhill	Barbara
4	1.015	0.955	1.013
8	1.562	1.499	1.364
16	2.184	2.014	2.257
32	2.774	2.564	3.173

TABLE 6.1. Entropies of quincunx decompositions of test images using KV5/3 filters with uniform scalar quantization. Unit of entropy is bits/symbol; bins column represents number of possible states of each pixel.

Moving on to the 16 bin images we see that the Goldhill image is markedly better in the separable case than in the quincunx case, however much of the distortion is due to edge effects, a topic which has been ignored in the scope of this thesis. It is expected that the severe distortion in the sky visible in the quincunx reconstruction would largely disappear if mirrored convolution were applied. Since

bins	Lenna	Goldhill	Barbara
4	18.31	17.21	19.13
8	23.48	21.67	23.14
16	29.09	27.65	28.46
32	34.24	33.56	33.80

TABLE 6.2. Table of PSNR (dB) values for reconstructed images after uniform scalar quantization of 12 level quincunx decompositions using KV5/3 wavelets.

bins	Lenna	Goldhill	Barbara
4	17.83	19.49	16.28
8	21.73	22.49	18.90
16	26.62	27.06	23.52
32	31.74	32.38	29.09

TABLE 6.3. Table of PSNR (dB) values for reconstructed images after uniform scalar quantization of 6 level separable decompositions using  $D_4$  wavelets.

bins	Lenna	Goldhill	Barbara
4			
8	1.602	1.388	1.252
16	2.402	1.828	2.102
32	3.179	2.565	3.061

TABLE 6.4. Entropies of quincunx decompositions of test images using the quincunx KV- $D_4$  orthogonal filters with uniform scalar quantization. Entropy values are in bits/symbol.

the quincunx filters are symmetrical but the comparison filter in the separable domain (the  $D_4$  filter) is orthogonal, mirrored convolution would cause distortion in the orthogonal filter but not the quincunx one thus skewing the results further. The Barbara image has a significantly better reconstruction in the quincunx case for the 16 bin case and the Lenna image is reconstructed well from both decompositions. For this last case, the separable case represents a visually more pleasing reconstruction although it doesn't contain the same level of feature content as the quincunx case. This is due to the good reconstruction of the vertical features in the background structures. For skin tones, the circular symmetry of the quincunx filters is especially good as distortions appear more natural, whereas the blocky distortions in the separable reconstruction are more obvious and disturbing.

## CHAPTER 7

### Perceptually Efficient Coding and Psychovisual Models

This chapter focuses on the ways in which certain perceptual redundancies within an image can be exploited to construct efficient coding methods. A brief explanation of the HVS model and some significant phenomena associated with it are first discussed, then we move onto ways in which some of these phenomena lead to lower bit-rate coding without sacrificing image quality.

Results are presented for applying psychovisual tuning to coding of quincunx data. These results illustrate the concepts covered and provide information on the importance of each of the redundancies. We also cover the psychovisual effects associated with using different filters. Results are presented to illustrate this.

The term psychovisual is used to refer to properties of human visual perception. The term encompasses factors resulting from the early image capture system through to the human perception of significant visual features. When light falls on our eyes we see an image but what information we perceive from that image is subject to many factors. The HVS is tuned to detect certain types of visual information and discard other superfluous information. It is tuned for detecting features which are characterized by abrupt changes in some image characteristic (such as colour and intensity) within a small spatial distance.

Being able to ignore or remove data which is essentially invisible to the human observer can significantly reduce the amount of data required to represent an image without a noticeable change in the perceived quality of the image.

#### 7.1. HVS Redundancies

The following sections briefly introduce various types of psychovisual redundancies which can be exploited to aid in coding. Further elaboration of the phenomena and how to exploit them in coding quincunx wavelet data is covered below in Section 7.2.

**7.1.1. Spatial Frequency Sensitivity.** The sensitivity of the human eye to stimuli of different spatial frequencies is a non-constant function of frequency. There is little sensitivity for very high frequencies because the details are too fine for our retina to discern and the mucus of our eyes acts as a low-pass filter; and there is little sensitivity for very low frequencies which are perceived as background. The eye is tuned to a narrow band of spatial frequencies with a peak response centred in the vicinity of 3-5 cycles per degree of solid angle. Above and below this spatial frequency range the response drops off. Figure 7.1.1 shows the

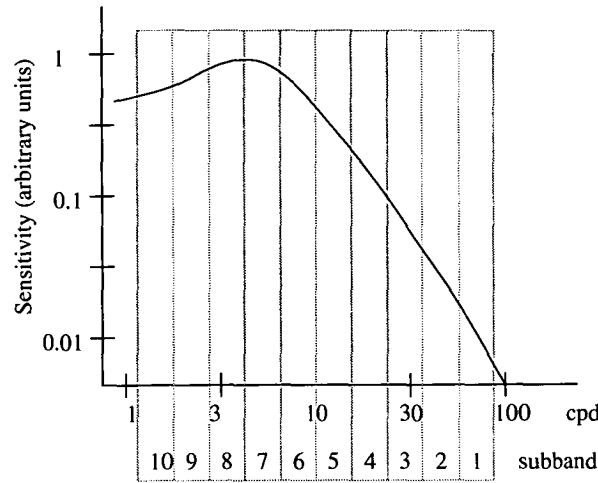


FIGURE 7.1.1. HVS spatial-frequency sensitivity curve with approximations to quincunx decomposition levels. The scale is logarithmic in frequency. Level 1 is the highest resolution detail and in this case there are 10 decomposition levels. Clearly levels 6 through 8 occupy the most sensitive areas of the curve and should be treated with the most lossless coding. This is a demonstration only and does not accurately reflect actual data.

approximate shape of the spatial sensitivity response function of the HVS. For specific measurement results and applications see [49], [116], [25], [76], [117], [93], [60], [109], [74], [34].

Because each person is different the exact form of the sensitivity function will vary. A general form for the sensitivity function is given by Ngan et al.[74] as

$$(7.1.1) \quad S(\omega) = (0.32 + 0.69 \cdot \omega) e^{-(0.29 \cdot \omega)}$$

Although not perfect it provides a mathematical form which is quite close to the majority of results in the field and importantly, it is mathematically simple and continuously integrable over the active range of human vision (from about 0.1 to 30 cycles / degree), a feature we make use of in Section 7.2.1.

The frequency selective behaviour of the HVS allows us to more harshly threshold and quantize image features which lie outside this sensitive band. In particular this is useful for subband coding since the image is conveniently broken down into frequency bands which can be then treated differently depending on the sensitivity of the HVS to that band.

The JPEG image compression standard makes efficient use of this principle by allocating a specific quantization level to each frequency of the DCT depending on its relative importance to human perception. JPEG uses a quantization matrix which determines the quantization scaling factor to be used based on spatial frequency and orientation of the DCT basis function which lead to the coordinate.

**7.1.2. Contrast Masking.** Contrast Masking is also referred to as Spatial Masking. It occurs when there is a significant feature such as an edge and also fine details in close proximity to the edge. Because the HVS is tuned to extract

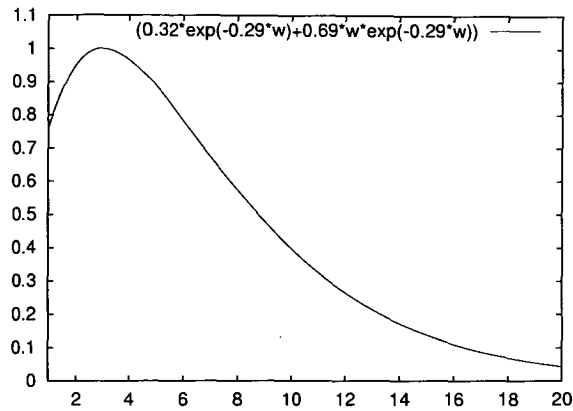


FIGURE 7.1.2. Plot of the spatial frequency sensitivity function of the HVS given in Equation 7.1.1. Note the peak in relative sensitivity at around 3 c / deg (2.9845).

the significant features from any given context, fine details tend to blend into the background. The closer to the edge the less sensitive the HVS is to the fine details because they are obscured by the lateral inhibition<sup>1</sup> from the edge.

In areas where there is a significant feature, other details at finer scales can be more highly quantized without a perceivable difference in the image. Figure 7.1.3 shows the approximate shape of the curve showing variation of contrast sensitivity of the HVS with distance from a significant stimulus [60], [73], [72, chapter 4], [108].

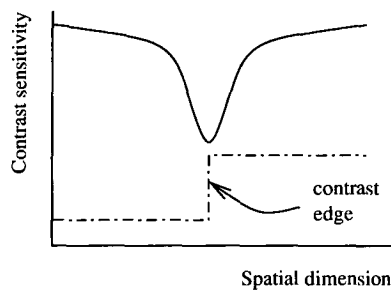


FIGURE 7.1.3. Function curve showing the change in visual contrast sensitivity near a significant stimulus. The dashed line represents the position of an abrupt change in image intensity which is the significant stimulus.

**7.1.3. Orientation Selectivity.** As discussed in Section 7.1.1, the HVS has a sensitivity function for various spatial frequencies. In addition to this the orientation of different spatial frequencies also affects the sensitivity of the HVS to the signal.

<sup>1</sup>Lateral inhibition is a phenomena of the very early HVS. The response of the HVS to point stimuli has a reinforcing affect at close proximity and then an inhibitory effect as the distance between stimuli increases. The difference of two Gaussians or  $DoG(x)$  function is a famous model of this phenomenon.

Of greatest sensitivity to the HVS are vertical edges, that is signals with a horizontally directed spatial frequency vector. As the signal orientation moves towards diagonal, the sensitivity drops significantly, typically by 3dB [72]. This allows us to bias our coding towards providing finer quality for those coefficients which contribute to vertical or horizontal edges, and coarser quality for those which contribute to diagonal edges.

## 7.2. Psychovisual Coding

In this section we outline how the phenomena outlined in the previous sections can be exploited with respect to the quincunx wavelet decomposition method to create efficient image coding methods. Of most importance are spatial contrast masking, orientation selectivity and spatial frequency selectivity.

**7.2.1. Using Spatial Frequency Sensitivity.** Wavelet transforms decompose the image into hierarchical subbands. Each successively coarser band has a smaller bandwidth (usually a constant ratio of the previous bandwidth). The division of the image data into such frequency bands allows a spatial-frequency sensitivity function to be applied to the quantization process and coding process. Wavelet coefficients (as opposed to those contained in an approximation signal) represent details which could not be predicted from the low-pass version of the image. For successively coarser bands the central frequency becomes lower. The first detail signal spans up to the Nyquist frequency of the image. From a perceptual standpoint the maximum frequency for a given sampled image is limited by the sampling density of the display, which is typically around 75-100 dots per inch (30-40 dots per cm).

Typical viewing distance is about 0.5m which puts the upper frequency of the display as seen by the observer at:

$$\begin{aligned}\omega_{cutoff} &= \frac{2\pi}{360} \cdot \frac{1}{\arctan\left(\frac{2}{4000 \times 0.5}\right)} \text{ cycles/deg} \\ &= 17.5 \text{ cycles/deg}\end{aligned}$$

and the peak sensitivity frequency of 5 cycles/deg corresponds to features with *period* = 7 dots. If we take a typical image size of 512x512 pixels and we decompose it to a depth of 8 levels (4 dyadic levels) then our low-pass has a maximum frequency of

$$\omega_{low} = \frac{17.5}{2^4} = 1.09 \text{ cycles/deg}$$

thus the detail spaces span the range from 17.5 down to 1.09 cycles/deg and so the peak sensitivity of the HVS lies in the middle. For this reason we can harshly quantize the finest and coarsest detail signals, but there will also be a detail image with peak sensitivity.



As we can see the low-pass band is well below the peak of the HVS sensitivity and thus it can be quantized reasonably harshly without significant perceptual degradation.

If we decompose the image further so there are more detail spaces then there will be a greater amount of detail at lower frequencies which can be severely quantized. The upper frequency zone is independent of the number of decompositions as it is defined by the physical viewing environment.

Throughout this work we have based perceptual coding on a model viewing environment where the monitor has an upper frequency of 4000 cycles/metre (90 dpi) and the viewer is 0.5 metres from the monitor resulting in an upper limit of 17.5 cycles/deg of solid angle at the eye of the observer.

It should be noted that the HVS sensitivity drops off more substantially with high-frequency than with low-frequency thus low-frequency components will generally be psychovisually more important than the high-frequency components of which only the most significant details are actually noticed by the HVS.

Applying the HVS sensitivity curve as shown in Figure 7.1.1 to wavelet decompositions is not a trivial matter. The measurements of sensitivity were performed for visual gratings of constant frequency and varying magnitude. This is easily transposed to the Fourier (and hence JPEG) domain where individual frequency components are considered. However in the wavelet domain, each signal spans a range of frequencies as defined by the bandwidth of the wavelet or scaling-function associated with the signal.

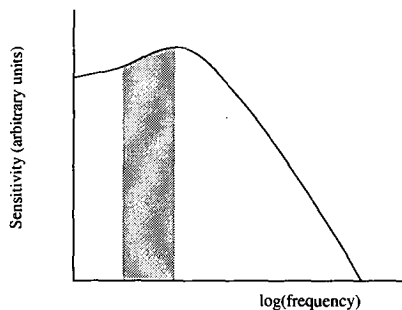


FIGURE 7.2.1. To find the relative sensitivity of the Human Visual System to a wavelet subband we need to find the mean of the sensitivity curve over the subband.

Consider the wavelet subband as shown in Figure 7.2.1. We can determine the mean sensitivity value of the subband as follows

$$(7.2.1) \quad S_{mean} = \frac{\int_{\omega_0}^{\omega_1} S(\omega) \cdot d\omega}{\omega_1 - \omega_0}$$

where  $\omega_0$  is the lowest frequency in the band and  $\omega_1$  is the highest frequency in the band.

This measure assumes that the contribution of all components in the band is equal. A more general form might be

$$S_{mean} = \frac{\int_{\omega_0}^{\omega_1} S(\omega)W(\omega) \cdot d\omega}{\omega_1 - \omega_0}$$

where  $W\{\omega\}$  is a weighting function in  $\omega$  which modifies the importance of parts of the band. One logical value for  $W(\omega)$  is  $W(\omega) = \Psi(\omega)$ , where  $\Psi(\omega)$  is the magnitude of the Fourier transform of the wavelet function at frequency  $\omega$ . Although this can possibly scale the values of  $S_{mean}$  the values have arbitrary units so it doesn't have any significance.

For the purposes of our experimentation we assume that the contribution is constant over the support of each subband.

Using the representation of the sensitivity factor from [74] Equation 7.1.1 we have

$$S(\omega) = (0.32 + 0.69 \cdot \omega) e^{-(0.29 \cdot \omega)}$$

hence

$$\begin{aligned} & \int_{\omega_0}^{\omega_1} S(\omega) \cdot d\omega \\ &= \int_{\omega_0}^{\omega_1} (0.32 \cdot e^{-0.29 \cdot \omega} + 0.69 \cdot \omega \cdot e^{-0.29 \cdot \omega}) \cdot d\omega \\ (7.2.2) \quad &= \frac{0.32}{-0.29} e^{-0.29 \cdot \omega} + \frac{0.69}{(-0.29)^2} (-0.29 \cdot \omega - 1) e^{-0.29 \cdot \omega} \Bigg|_{\omega_0}^{\omega_1} \\ &= - (9.3079 \cdot e^{-0.29 \cdot \omega} + 2.3793 \cdot \omega \cdot e^{-0.29 \cdot \omega}) \Bigg|_{\omega_0}^{\omega_1} \end{aligned}$$

The calculation of the frequency range of the subband we are concerned with is not automatic. With separable sampling the decomposition levels results in an octave subband structure. In this case the highest frequency subband would span from  $\frac{1}{2}\omega_{max}$  to  $\omega_{max}$ . With quincunx sampling this is not quite the case as there are two possible orientations. Firstly note that two quincunx decompositions are equivalent to one separable decomposition so the bandwidth covered by two iterations is again  $\frac{1}{2}\omega_{max}$  to  $\omega_{max}$ .

We can approximate both quincunx bands to the separable band under which they fall and this makes for a simple solution. However this removes one of the strengths of the quincunx decomposition which is its ability to tune coding based on frequency sensitivity values in a more finely grained manner than separable decompositions. To thoroughly solve this problem we need to consider the sensitivity factor separately for two different orientations. When the subband has horizontal-vertical orientation then its maximum frequency is as expected. For diagonally oriented subbands however, the distance between immediately adjacent pixels has increased by a factor of  $\sqrt{2}$  hence the maximum frequency is now

$\omega_{max_{diag}} = \frac{1}{\sqrt{2}}\omega_{max_{HV}}$ . Now the lower frequency bound of the subband is the upper bound of the next lower resolution subband.

If the maximum screen resolution is  $\omega_{screen}$  then the highest resolution subband will span the frequency range  $\frac{1}{\sqrt{2}}\omega_{screen}$  to  $\omega_{screen}$ . The next lower subband will span from  $\frac{1}{2}\omega_{screen}$  to  $\frac{1}{\sqrt{2}}\omega_{screen}$  etc. through all the subbands to the lowest resolution band. The  $n^{th}$  highest subband ( $n \geq 1$  with  $n = 1$  being the highest resolution) spans the frequency range  $(\omega_{screen} \cdot (\sqrt{2})^{-n}, \omega_{screen} \cdot (\sqrt{2})^{1-n})$ .

As an example we calculate the mean sensitivity of the HVS to the 5<sup>th</sup> decomposition detail signal.

From above,  $\omega_{low} = 17.5 \cdot (\sqrt{2})^{-5} = 3.09$  c/deg and  $\omega_{high} = \sqrt{2} \cdot 3.09 = 4.4$  c/deg.

So using 7.2.2 we get an overall relative sensitivity factor of

$$S_{mean}(5) = \frac{1.200 + 0.0782}{4.4 - 3.09} = 0.9756$$

As a quick check, looking at the sensitivity curve plot in Figure 7.1.2 we see that the sensitivity result appears approximately correct.

**7.2.2. Coding using Sensitivity Data.** For each subband in a decomposition we can determine the mean sensitivity of the HVS based on the expected viewing distance. Table 7.1 shows the relative sensitivities for 20 levels of quincunx downsampling. The normal application of this information is to adjust the threshold levels and step size for quantization. The lower the sensitivity value, the larger the threshold and step size.

Subband	$\omega_{low}$	$\omega_{high}$	$S_{mean}$
1	12.37	17.5	0.1468
2	8.75	12.37	0.3617
3	6.19	8.75	0.6293
4	4.375	6.19	0.8550
5	3.09	4.375	0.9772
6	2.188	3.09	0.9925
7	1.547	2.188	0.9332
8	1.094	1.547	0.8377
9	0.773	1.094	0.7344
10	0.547	0.773	0.6398
11	0.387	0.547	0.5605
12	0.273	0.387	0.4976

TABLE 7.1. Values of mean relative sensitivity of the HVS for subbands in the quincunx decomposition. Values are for viewing a 90 dpi monitor at 40cm.  $\omega_{low}$  and  $\omega_{high}$  are in cycles per degree of solid angle.

A logical method would be to set  $thresh \propto \frac{1}{S_{mean}}$  and  $step \propto \frac{1}{S_{mean}}$ . For uniform quantization often the threshold and the step size are the same.

Applying this quantization scheme results in immediate gains in perceived quality when the bit-rate is held constant.

By applying a scaling to all wavelet coefficients based on the sensitivity of the HVS to the subband in which the coefficients occur, allows us to bias the quantization process. If we scale all wavelet coefficients in a subband by the sensitivity factor for that subband then the zero-tree coding process will favour coding those coefficients in the more sensitive subbands.

So we make

$$\text{value}_{\text{new}} = \text{value}_{\text{orig}} \cdot S_{\text{mean}}$$

We find that applying this simple scaling results in dramatic improvements in image quality for unchanged bit-rate. For the Lenna image, the quality change is not particularly great but for the Goldhill and Barbara images we see a significant improvement in the image quality.

### 7.3. Results Using HVS Tuning

We present here some results showing the significant difference which simple tuning of the quantization process can make. We make use of the perceptual phenomena discussed above to bias the quantization process more towards perceptually important components.

**7.3.1. Frequency Selectivity Results.** In this section the quantization of the decomposed signals is tuned to use the spatial frequency sensitivity model of the human visual system. We apply scalar quantization to the decomposition detail signals with a varying number of possible states, however the signals are weighted so that those which are more important to the HVS (on a frequency sensitivity basis) are magnified and thus more likely to be finely quantized.

Section 7.2 outlines how HVS frequency sensitivity operates and how we can arrive at a weighting function for wavelet subbands.

To perform HVS weighting in this section, coefficients for each subband were weighted before quantization and corrected to their original scale after quantization. This results in a higher probability that coefficients in perceptually more important subbands will be coded to a higher precision than those in unimportant subbands.

If  $S(n)$  is the sensitivity weighting factor for subband  $n$  then all coefficient values in that subband are scaled by  $S(n)$  so  $y' = y \cdot S(n)$ . After scaling, the dynamic range of the modified decomposition is determined and scalar quantization is performed identically across all scales. After quantization is finished coefficient values are scaled back to their original dynamic range ie.  $\hat{y} = \hat{y}' \cdot \frac{1}{S(n)}$ , where  $\hat{y}$  is the quantized version of  $y$ .

The entropy values of HVS and uniform quantized images are quite similar. In general for the same number of quantization bins the PSNR values are higher for uniform quantization as can be seen by the comparison in Table 7.5. This is to



FIGURE 7.3.1. HVS tuned scalar quantization of 6 level  $D_4$  separable decomposition of image Lenna.

be expected as it is generally known that uniform quantization leads to the lowest mean square error of all non-adaptive quantization schemes. However contrary to the PSNR results the perceived image quality goes against these results when viewed under the designed conditions as outlined in section 7.2.1 (in this case 0.5m from the monitor). This is because although uniform quantization produces a result with a lower statistical error, it does not weigh the relative importance of the errors to the human visual system. This can be demonstrated by comparison of the images in Figure 6.3.5 (c) and Figure 7.3.4 (c). For convenience these images are reproduced below in Figure 7.3.7. The Goldhill images in Figure 7.3.7 (c) and (d) have a difference in PSNR of  $3.82dB$  yet there is no perceivable difference in quality. Statistics based thought would equate a difference of nearly  $4dB$  with a significant change in image quality, but this is shown to not be the case.

As can be seen from Figure 7.3.7 the use of HVS weighting of scalar quantization can make a marked difference in the perceived quality of an image for the



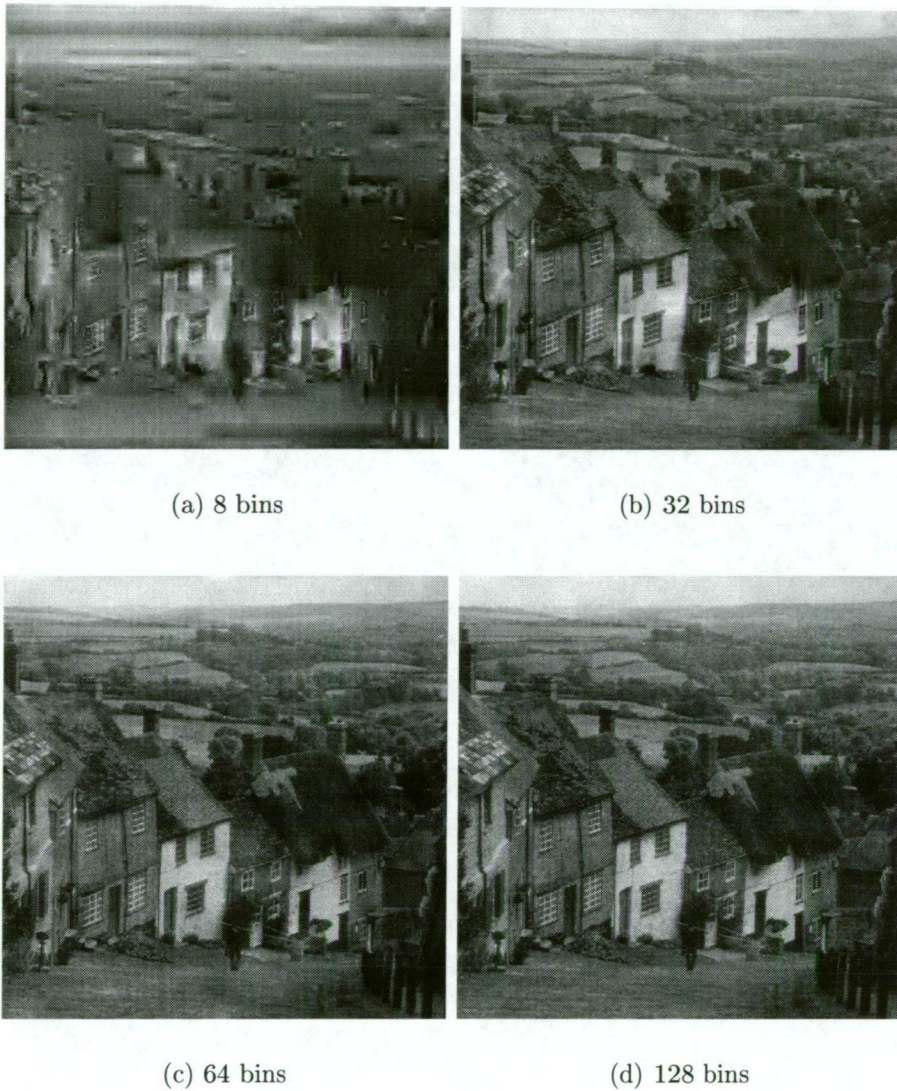


FIGURE 7.3.2. HVS tuned scalar quantization of 6 level  $D_4$  separable decomposition of image Goldhill.

same PSNR, or lower the PSNR while retaining the same perceived image quality. With HVS weighting of coefficients, the details reconstructed are those to which the human eye is most tuned and so the images have a higher perceived quality than achieved with uniform quantization and the same error. This apparent contradiction between PSNR value and perceived quality is a reasonably common occurrence, one which is encountered in other areas of this Thesis as well - in particular Section 10.3.

**7.3.2. Orientation Selectivity Results.** The structure of a quincunx decomposition is such that the subbands alternate between horizontal-vertical information and diagonal information. The first level of decomposition results in subbands where the full gamut of horizontal and vertical information is contained in the low-pass signal, but only part of the diagonal information is contained in



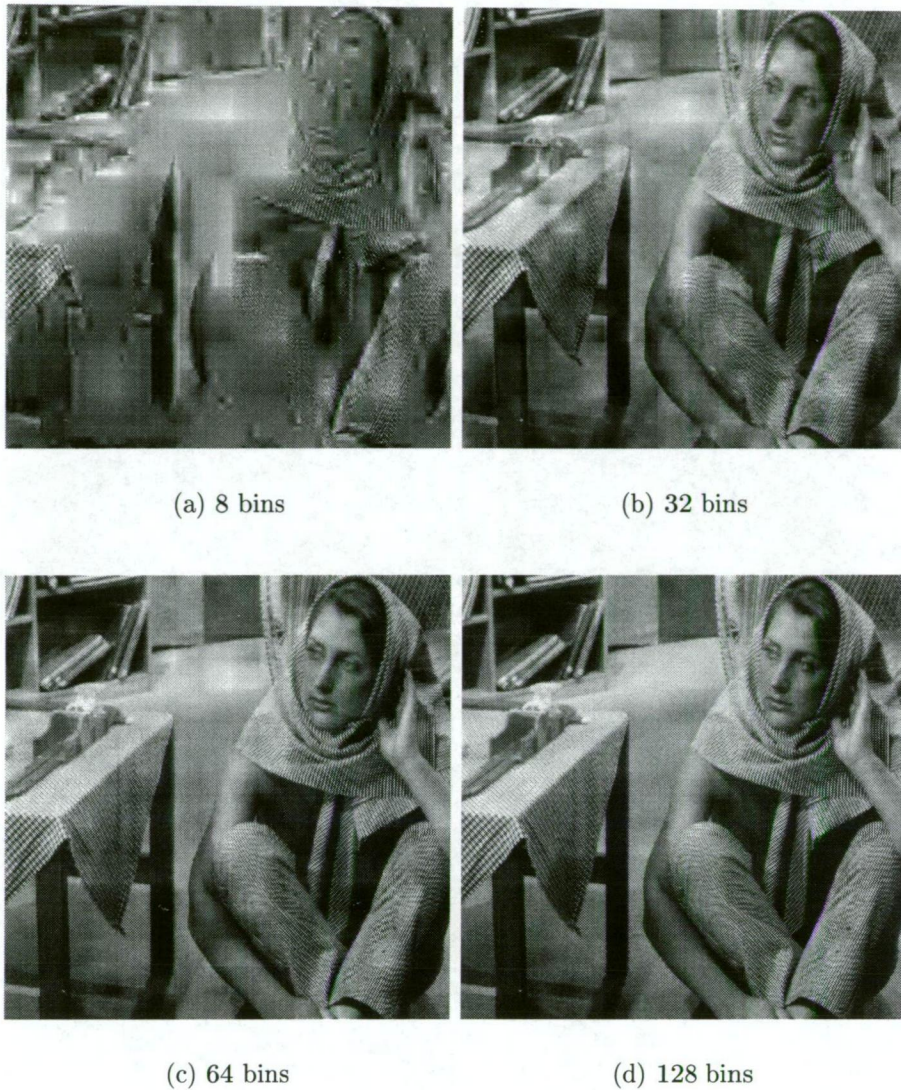


FIGURE 7.3.3. HVS tuned scalar quantization of 6 level  $D_4$  separable decomposition of image Barbara

bins	Lenna	Goldhill	Barbara
4	0.0637	0.0956	0.161
8	0.212	0.331	0.486
16	0.495	0.811	0.991
32	0.985	1.584	1.656
64	1.782	2.522	2.499

TABLE 7.2. Entropy values of reconstructed images from HVS weighted scalar quantization of 6 level separable decompositions with the  $D_4$  wavelet.

the low-pass. The detail signal contains the missing predominantly diagonal information. The second decomposition step results in a detail with horizontal and vertical details.

Using the fact that the HVS is less sensitive to diagonal details than to vertical or horizontal details, we can refine our coding. For subbands which result in a



bins	Lenna	Goldhill	Barbara
4	18.79	19.26	17.90
8	20.45	22.71	19.53
16	22.87	25.17	22.27
32	28.48	29.98	27.58
64	34.61	35.98	33.78

TABLE 7.3. PSNR values of reconstructed images from HVS weighted scalar quantization of 6 level separable decompositions with the  $D_4$  wavelet.



FIGURE 7.3.4. Lenna image coded with HVS weighted scalar quantized wavelet transform data using 12 levels of quincunx decomposition with KV5/3 filters.

detail with predominantly diagonal information, we can more coarsely code the data.

We can modify the mean relative sensitivity data from Table 7.1 above to account for the reduced sensitivity of the HVS to diagonal features. Netravali



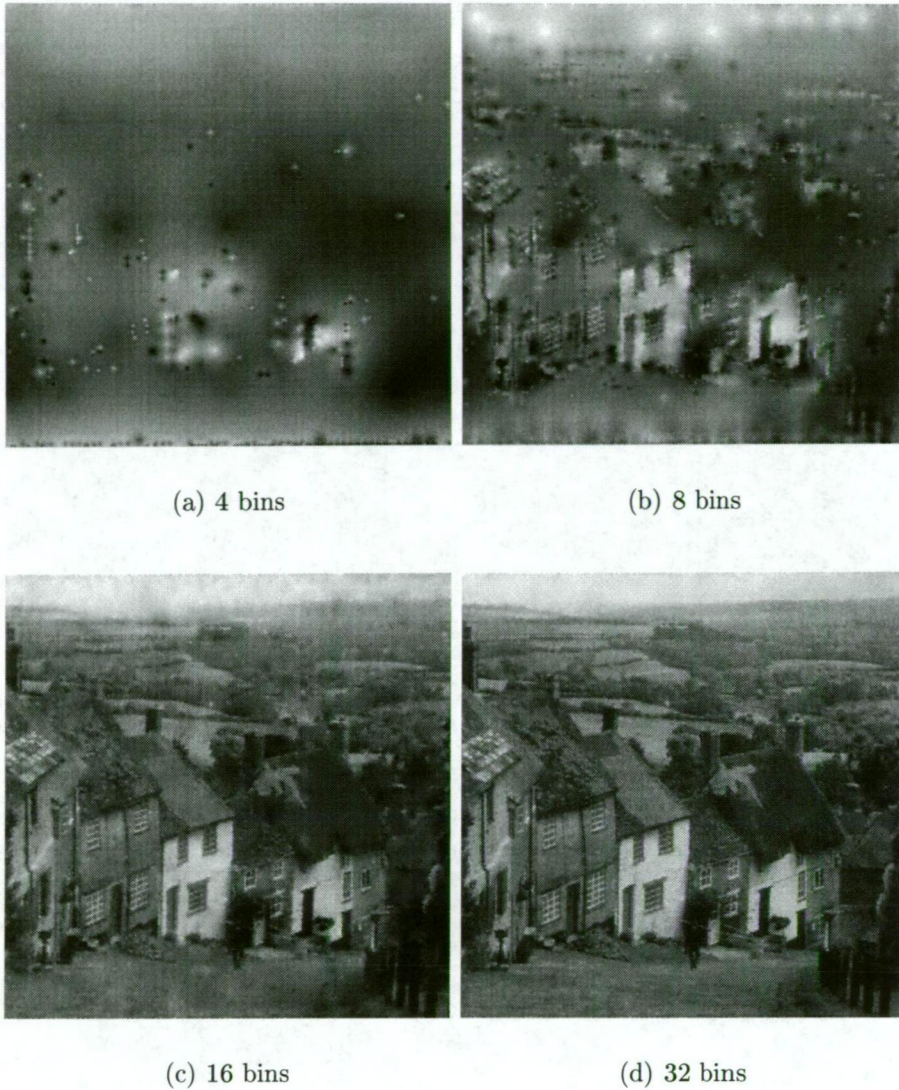


FIGURE 7.3.5. Goldhill image coded with HVS weighted scalar quantized wavelet transform data using 12 levels of quincunx decomposition with KV5/3 filters.

and Haskell [72] found that the sensitivity of the HVS to diagonal features closely follows that of vertical and horizontal features but with a contrast gain 3dB lower. There is no relationship between the arbitrary relative sensitivity values of Ngan et al. [74] and the quantitative values referred to by Netravali and Haskell, so we make an assumption here.

We assume that a 3dB reduction in contrast gain translates to a sensitivity reduction by a factor of 2. The 3dB reduction in sensitivity only occurs for features exactly at the diagonal but the diagonally oriented subbands of the quincunx decomposition contain details from all orientations but weighted more towards diagonal content. To account for this we apply a correcting factor to bring the change in sensitivity to a factor of 1.5 which we find works well.

To illustrate the sensitivity of our eyes to the various subbands we examine the distortion resulting from the selective removal of signal energy from particular



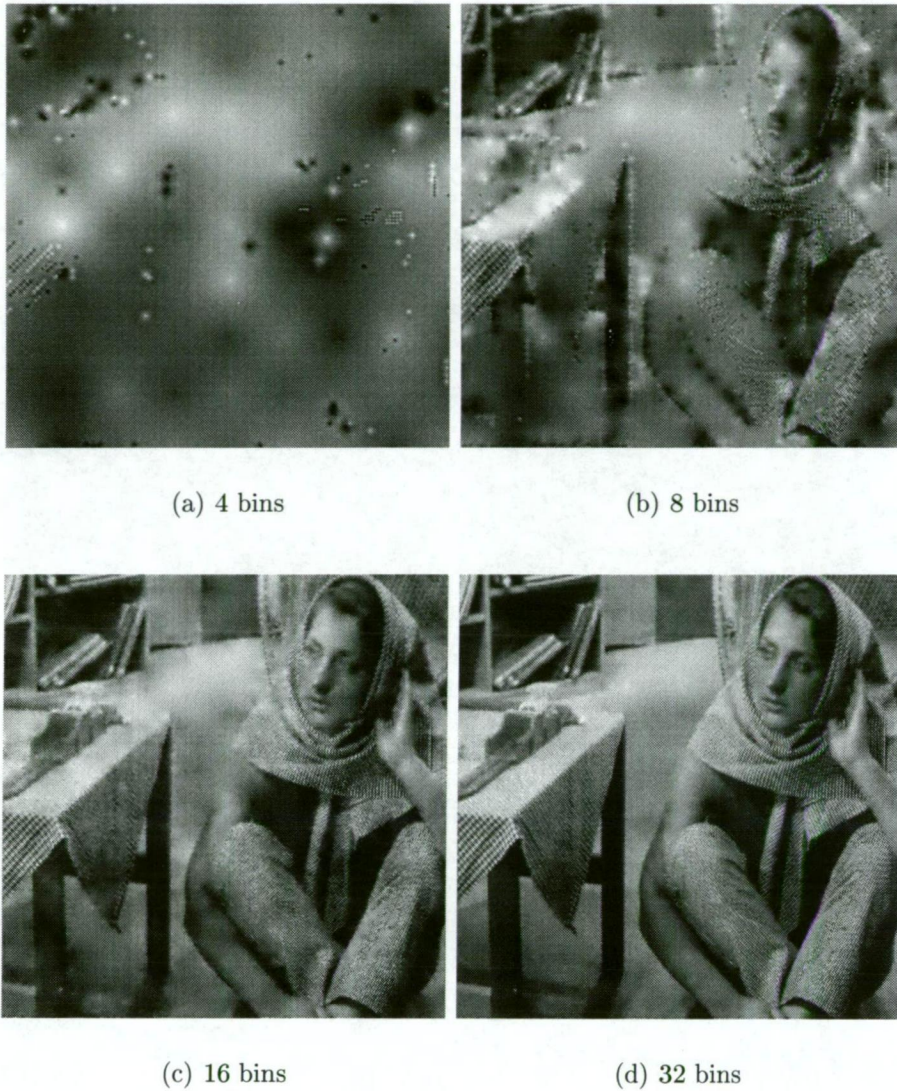


FIGURE 7.3.6. Barbara image coded with HVS weighted scalar quantized wavelet transform data using 12 levels of quincunx decomposition with KV5/3 filters.

subbands in a wavelet decomposition. It is also useful to investigate this because lossy coding of wavelet decompositions can result in severe removal of energy from many of the subbands of a decomposition. A desirable property for a wavelet transform to have is that the reconstruction from just the approximation data be of good quality without the addition of energy from higher frequency subbands. If the reconstruction from the approximation data is good then more of the detail signal energy can be removed without affecting image quality too severely.

We find that the reconstruction from the separable and quincunx transforms take on different forms of distortion as significant amount of detail energy are removed suggesting that the choice of transform to use is dependent on the image being processed.





(a) Uniform quantization.  
PSNR=29.09dB.



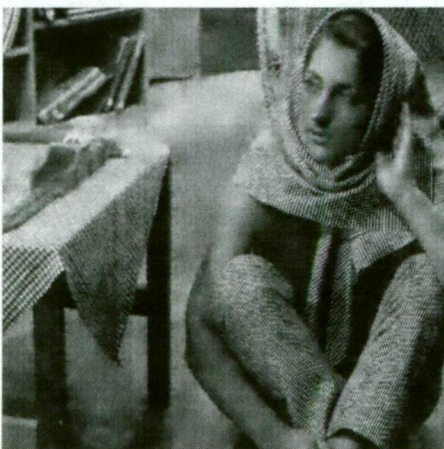
(b) HVS quantization.  
PSNR=26.79dB.



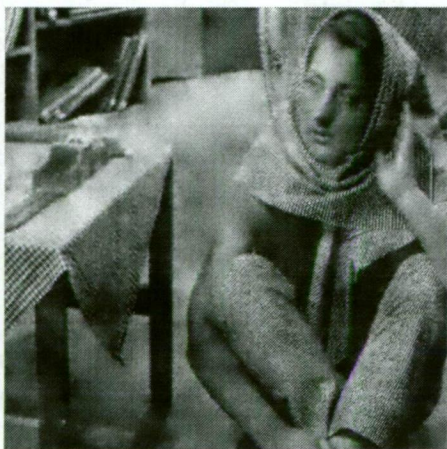
(c) Uniform quantization.  
PSNR=27.06dB.



(d) HVS quantization.  
PSNR=23.24dB.



(e) Uniform quantization.  
PSNR=28.46dB.



(f) HVS quantization.  
PSNR=26.04dB.

FIGURE 7.3.7. Comparison of HVS weighted and uniform quantized images. Both images have very similar perceived quality while having disparate PSNR values.



bins	Lenna	Goldhill	Barbara
4	1.021	0.972	1.035
8	1.692	1.536	1.392
16	2.436	2.160	2.071
32	3.095	2.803	2.870

TABLE 7.4. Entropies of quincunx decompositions of test images with HVS tuned quantization. Unit of entropy is bits/symbol.

bins	Lenna	Goldhill	Barbara
4	16.95 (18.31)	16.14 (17.21)	17.88 (19.13)
8	21.21 (23.48)	18.86 (21.67)	21.45 (23.14)
16	26.79 (29.09)	23.24 (27.65)	26.04 (28.46)
32	31.82 (34.24)	26.91 (33.56)	30.17 (33.80)

TABLE 7.5. PSNR values of reconstructed images from HVS weighted (and uniform) scalar quantization of 12 level quincunx decompositions with the KV5/3 wavelet. Uniform quantization results reproduced from Table 6.2 in parentheses.

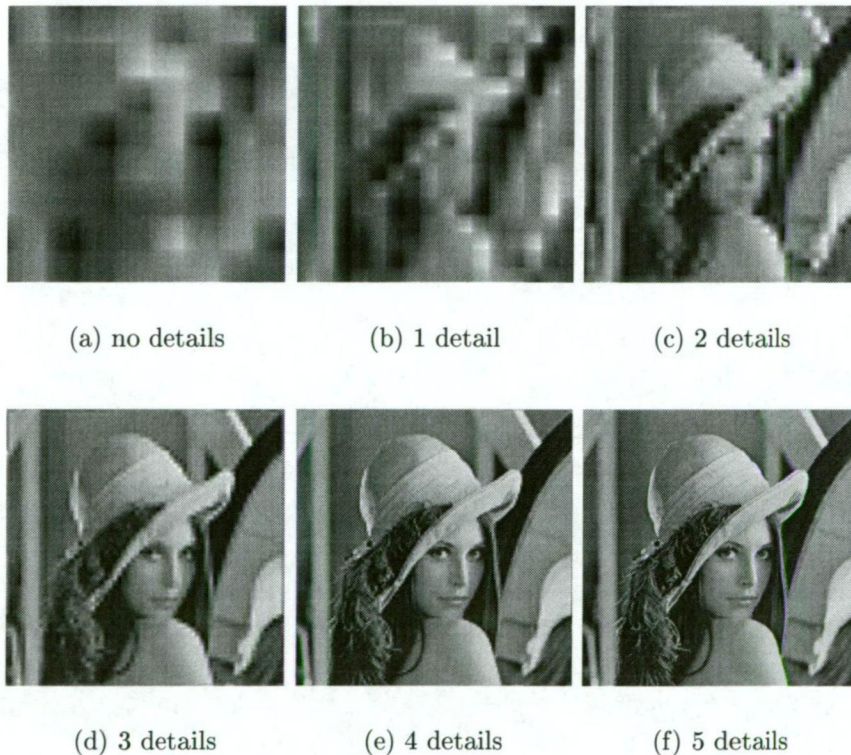


FIGURE 7.3.8. Lenna image reconstructed from the lowest resolution details of a 6-level separable wavelet decomposition using Daubechies  $D_4$  filters.

Figure 7.3.8 is reproduced from Figure 2.12.1. It shows how the quality of the reconstructed image varies as detail energy is removed from a separable decomposition.



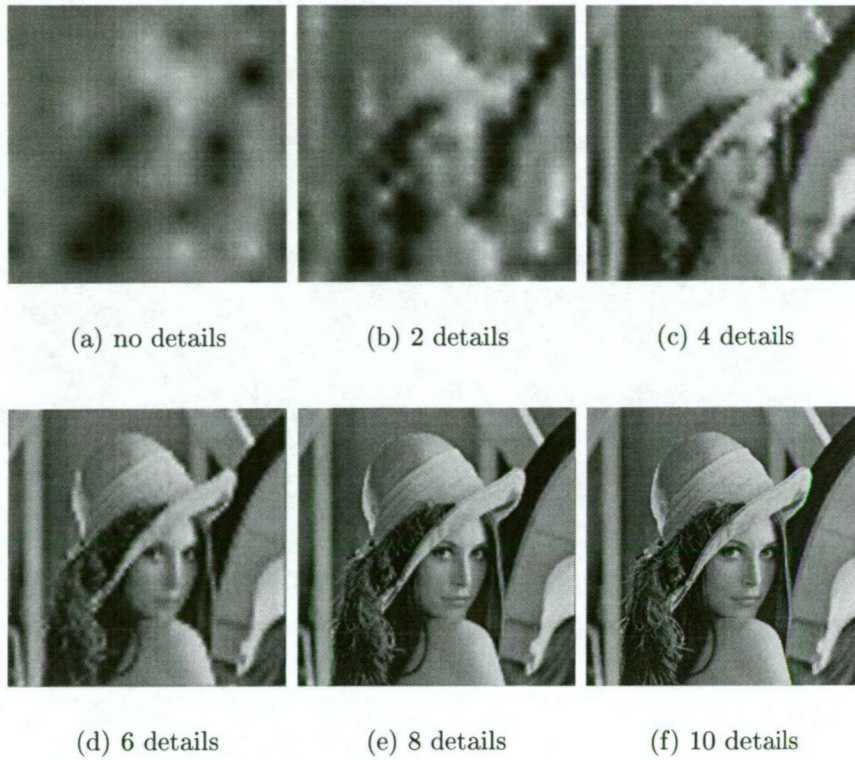


FIGURE 7.3.9. Lenna image reconstructed from a 12 level quincunx decomposition using the Kovačević and Vetterli 5/3 biorthogonal filter pair [101], [56].

Figure 7.3.9 is a similar situation for the quincunx case. In this case there are 12 levels of decomposition to ensure that the approximation signals of both the separable and quincunx decompositions are the same size.

One of the most noticeable effects which can be seen is that the quincunx reconstruction from just the low-pass data is of significantly lower quality than the  $D_4$  version. In particular, the image is not smooth, but rather contains strong dots - bright areas centred on the positions of the low-pass values. The reason for these dots is the shape of the KV5/3 filters. They have a very large central tap with respect to the size of the surrounding taps. This causes the central tap to dominate the shape of the filter. In this case where there is no additional detail to counter-balance this effect, the artifacts are quite strong.

It can be seen that both separable and quincunx methods perform with a similar degree of reconstructed quality when at least 2 (equivalently 4 for quincunx) details signals are used.

Figure 7.3.10 shows the effects of removing selected detail signals from a separable decomposition of Lenna. The orientation specificity of the separable transform is dramatically demonstrated by the complete lack of features of a particular orientation when its wavelet coefficients are absent. This is strikingly evident in Figures 7.3.10 (a), (c) and (e) where the strong vertical feature on the left-hand side of



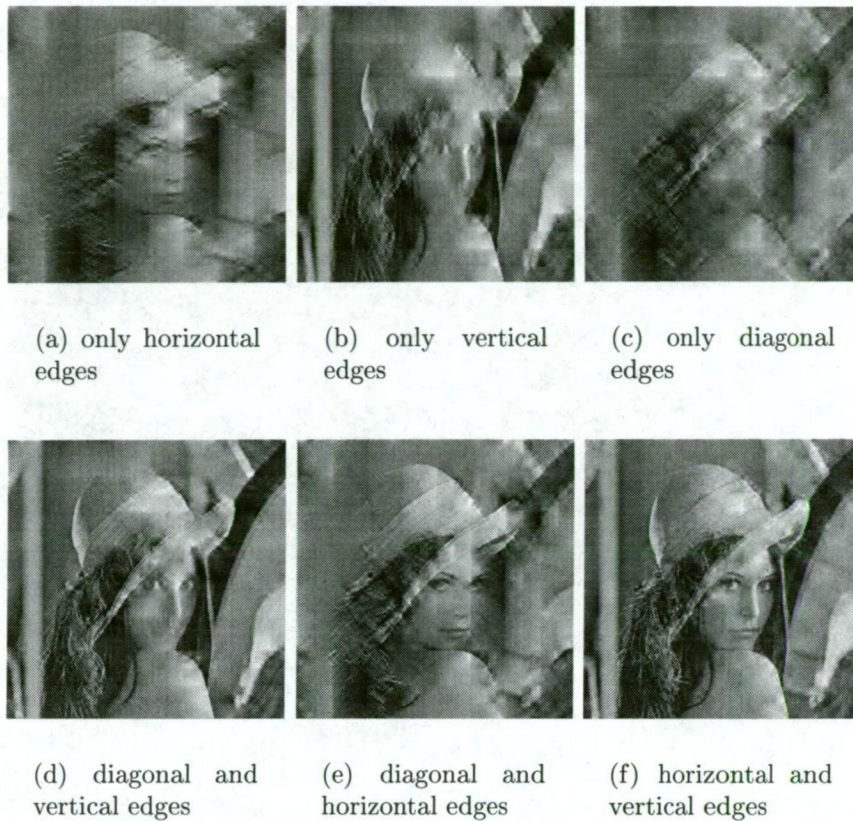


FIGURE 7.3.10. The effects of selectively removing some of the of the three orientations of detail signals in a 6 level separable decomposition of Lenna using  $D_4$  wavelets. In each case details were removed from every level of the decomposition or not removed at all.

the image is completely removed when vertical edge data (low-pass vertically and high-pass horizontally) is removed.

Comparing the visual quality of the images in Figures 7.3.10 (d), (e) and (f) we find that (f) is the most pleasing, followed by (d) and lastly (e). This was predicted from the theory of orientation specificity because the HVS is most sensitive to vertical edges, and least sensitive to diagonal images. Figure 7.3.10(f) contains vertical and horizontal edges, the two most important, and so is the most pleasing. The least pleasing is Figure 7.3.10(e) which does not contain any vertical edges - hence only the low importance components are left.

Comparing the top line of Figure 7.3.10, it is obvious that (b) is the best of the bunch. Again it corresponds well with what is predicted since it contains the vertical edges which are most important. The least pleasing of the top row is Figure 7.3.10(c) which contains only diagonal components.

At this point it is also useful to demonstrate one final experiment for orientation selectivity. The reader is again directed to Figure 7.3.10 and is asked to tilt his/her head so that edges which were horizontal are now viewed as vertical. Doing this it can be seen that a dramatic improvement is observed in Figure 7.3.10(a) which

now has only vertical features, and a dramatic decline is noticed for (b) which now has horizontal features.

**7.3.3. Spectral Masking.** Because of the lateral inhibition function of the HVS there are certain spatial frequencies or spacings of regular features at which the HVS displays resonant/anti-resonant tendencies. It is possible for features of a particular spacing to interfere with other features so as to obscure some features. Unfortunately this type of phenomenon isn't generally usable for image coding because it relies on the interaction of particular ranges of spatial frequencies and because we can't control the viewing distance the viewer chooses we can't guarantee that certain features will have a particular spatial frequency on the retina.

Spectral masking was not used in the production of results for this work but it is included here for completeness.

**7.3.4. Spatial Masking.** Spatial masking occurs when fine details (ie. minor features located in high-resolution levels) are masked by significant features (usually of larger scale, ie. lower resolution). If significant features at coarse detail levels mask details at finer detail levels then we can adaptively vary the quantization threshold for the wavelet coefficients at different resolution levels and locations depending on the behaviour of parent<sup>2</sup> pixels and neighbours to parent pixels.

If we code our details in order from coarsest to finest we can predict the quantization levels at both the encoder and the decoder since we must already have knowledge of the coarse details in both cases. This coding method lends itself well to the zero-tree coding process (as will be discussed in detail below) and forms the basis of one of the major contributions of this work.

This type of masking can extend to several decomposition levels. The stronger the feature is the greater the number of levels it will have a masking effect over. The effects can also be compounded by overlapping features at different scales. This is a common occurrence since many image features (ie. edges) span several scales.

The locality of the significant features also determines the quantization of the finer details. The closer - in spatial terms - the significant feature is to the detail the greater the masking effect and thus the greater the quantization threshold can be. The neighbourhood of masking is typically about 3 pixels at the same detail level as the feature which will expand as the levels become finer.

Additionally the presence of significant features also obscures other weaker features in the same detail level. This generalized case is known as Activity Masking or Contrast Masking and it takes into account the overall activity within an image region. The greater the activity the greater the amount of information which is obscured from the perception of the observer.

---

<sup>2</sup>Parent-child relationships are covered in detail in Chapter 5.

Later chapters, in particular Chapter 10, of this Thesis are dedicated to further developing these concepts and the implementation of a coder on the quincunx lattice for this purpose.

#### 7.4. Distortion Due to Choice of Wavelets

This section examines the effect that the choice of wavelet has on the reconstructed image quality. A more rigorous analysis of this topic can be found in [105] and [65]. We restrict ourselves to considering the intensity of the images and ignore the effects of colour. The quantization of colour has been thoroughly investigated and it is widely recognized that the intensity information is much more important. A good example of this is the colour sampling methods of JPEG [33] which samples colour data in a more sparse sampling grid than for intensity data.

It has been shown [86],[87] that the quality of a reconstructed image in a wavelet subband coding system is highly dependent on the choice of wavelet. For the separable case it now appears to be generally accepted that the 9/7 biorthogonal pair of Antonini, Barlaud and Daubechies [8],[61] gives the best all round performance for psychovisual assessment.

For the quincunx case no such wide-ranging assessments have been made because it is not possible to make use of the body of filter design work which has been carried out in the one-dimensional domain. The lack of available filters for use with the quincunx decomposition makes comparisons between separable and non-separable decompositions very difficult. However this remains one of the aims of this Thesis. The design of the required filters is outside the scope of this Thesis but sufficient work has been presented to provide some rough comparisons.

Those filters which have been considered in the research leading to this Thesis have been found to have some problems which make them generally unsuitable for image processing. This is to be expected since they were not designed with such a purpose in mind. In fact all available quincunx filter families produced to date are, to the best knowledge of the author, for demonstration purposes only, ie. merely to demonstrate that a quincunx decomposition is possible. Their designs are thus low in complexity and lacking many of the important properties (with respect to image compression) of some of their well designed one-dimensional counterparts. Smoothness of response is a particular point of concern in some cases and attempts were made to correct this in a case outlined in Chapter 4.

The para-unitary filters as proposed by Kovačević and Vetterli in [101] are very irregular and in general not suited to images with any smooth areas as synthesis of smooth areas from such jagged functions requires many coefficients. The irregularity in the reconstruction (synthesis) wavelets leads to very visible errors when even a small number of coefficients are lost in coding. The loss in SNR may not be great - this filter set was found to have better PSNR than others tested

- but the visibility of the coding errors is very high and thus from a perceptual point of view the reconstructed quality is not acceptable. Results showing this distortion and PSNR values are shown in Section 7.5.

The zero-phase filters of Kovačević and Vetterli in [101] and [56] are much smoother than the Daubechies  $D_4$  based para-unitary filters and their performance is in general much better from a perceptual stand-point. This is in part due to the symmetry which these filters possess. However they are still not smooth enough for the applications for which they are being used in the context of this work. The large relative magnitude of the central tap of the filters leads to very obvious errors when only a small number of coefficients is available. The quality degradation is far better than for the para-unitary case due to the more regular nature of the filters and their symmetry but it is still unsuitable for useful image compression.

Symmetric filters are highly sought-after for image processing work. One of the key properties of symmetric filters is their linear-phase nature which leads to a constant delay on all frequency components. Thus the alignment of the many components required to synthesize an edge is not disrupted by the action of the filter.

With the KV5/3 linear phase filters the distortion that is caused by the quantization (and eventual removal) of the coefficients appears in the form of patches of distortion which are a result of the isolated (in space) large magnitude coefficients which remain. The inverse transform results in primal wavelets being constructed around these centres as successive inverse transforms act on the signal. Thus we find throughout the reconstructed image, areas of error where the image contains patterns resembling the shape of the primal wavelet<sup>3</sup> derived from the filters used. If this wavelet is smooth and local in nature then its effects are not very noticeable.

Another problem which is faced by truly 2-D wavelet compression is that the filters which are used *are* actually two-dimensional. This means that their area of support covers not just a single column or row but an area of the image at every resolution level. Thus a single coefficient (in a low resolution detail signal) can lead to reconstructed visual effects spanning a very large area of the image. Currently this is a disadvantage because the filters used are not well suited to the task of image compression; generally because their shape does not form a good match for common image structures/features. Thus major features are synthesized which do not accurately represent the image content and more coefficients are required to refine the representation. It is the belief of the author that as the design of suitable 2-D filters improves, the ability of truly 2-D wavelets to efficiently represent images will improve dramatically. This is because of the fact that a single coefficient *can* synthesize features over such a large area of support.

---

<sup>3</sup>A primal wavelet is the waveform which result from the inverse wavelet transform of a delta function. The single coefficient results in a waveform which is the basic shape of all wavelets derived from the filters.

The choice in determining significant coefficients becomes very difficult when the wavelet basis function is not well suited to the signal which is being compressed. Although principles of orthogonality remain, the interaction between coefficients in the support areas in two dimensions is significantly greater than in the one-dimensional case. We wish to determine which coefficients contribute the most significant information to the human viewer and retain as much of that information as possible. When our basis is not well suited to our signal then we are often faced with the situation where several weak coefficients are needed to construct a significant feature (which could be expressed by a single large coefficient), and yet the magnitude of neither one of the coefficients alone is enough to allow it to be preserved through quantization. Hence all information associated with the feature is lost. If the same feature could be expressed by a smaller number of larger coefficients, compression would likely result in better reconstructed image quality.

In general the wavelet decomposition represents an efficient decorrelation and compaction of the data present in a signal. Statistically the reconstructed quality is expected to be good from a particular amount of retained information energy. For consumption by the human viewer the best results will be obtained when the major details contained in the transform coefficients represent major features as perceived by the human viewer. In this case the basis functions will closely coincide with image primitives of the early HVS. The more closely we can tune the basis functions to those primitives the better will our representation be for a small number of coefficients.

It was with this in mind that non-separable sampling was chosen as the basis of the majority of the work presented in this Thesis. The response of the primitive areas of the HVS to stimuli is two-dimensional in nature. A point stimulus elicits an excitation/inhibition response from the HVS which extends into an area surrounding the location of the stimulus on the retina. In an image processing context the response is often approximated by a "Difference of two Gaussians (DoG)" function ie.  $DoG(x) = G_1(x) - G_2(x)$  where  $G_1(x)$  and  $G_2(x)$  are Gaussian functions with differing values of standard deviation. It is this inhibition behaviour which leads to the phenomenon of spatial activity masking where an intense feature can obscure the presence of finer details if they occur within close spatial proximity to the intense stimulus. It also leads to the varying spatial frequency sensitivity of the HVS. It is desirable then that a basis function for image coding reproduce this inhibition property.

### 7.5. Results Using Various Quincunx Filters

In this section we compile some results of using various filters developed for the quincunx lattice. The results presented here are presented for completeness



and to exemplify statements made above about the suitability or lack thereof of the various filters to image processing applications.

**7.5.1. The Quincunx  $D_4$  Wavelet.** As stated above, Kovačević and Vetterli presented a quincunx complement to the orthogonal Daubechies  $D_4$  filter[101]. We present here some examples of coding using this filter set.

Figures 7.5.1 - 7.5.3 show the results of scalar quantization of images transformed using the quincunx  $D_4$  filter-set. The quantization is performed using the same method outlined above for the separable  $D_4$  and quincunx KV5/3 filters. As can be seen there is significant, highly visible distortion in the image even with a high precision of coefficient values.

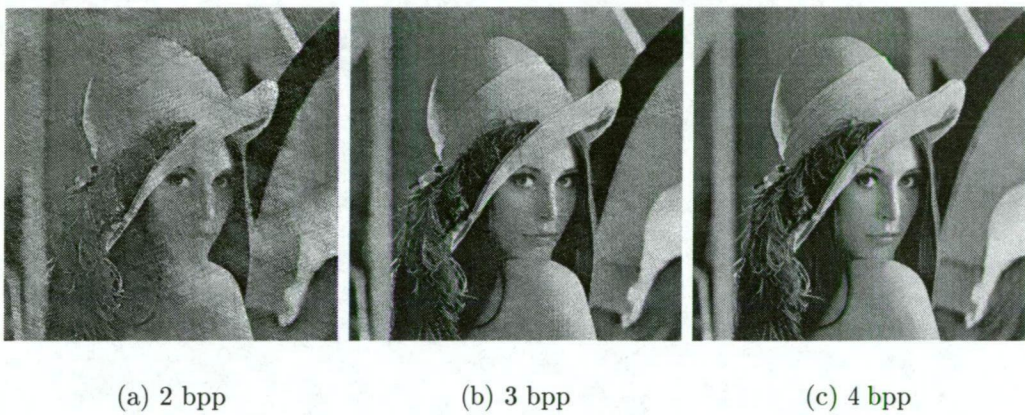


FIGURE 7.5.1. Results of using the Kovačević and Vetterli  $D_4$  wavelet for quincunx with uniform scalar quantization on Lenna image.

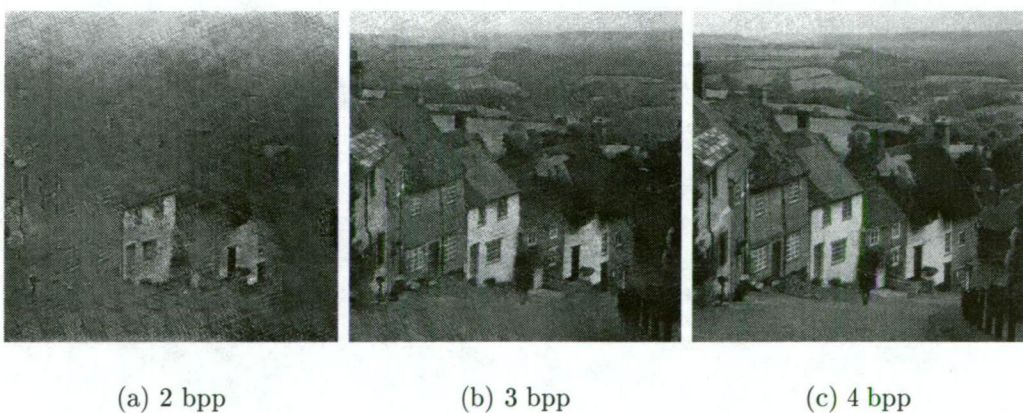


FIGURE 7.5.2. Results of using the Kovačević and Vetterli  $D_4$  wavelet for quincunx with uniform scalar quantization on Goldhill image.

The cross-hatch distortion proves to be exceptionally irritating to the eye and persists through all forms of coding tested. It is worth noting the PSNR values



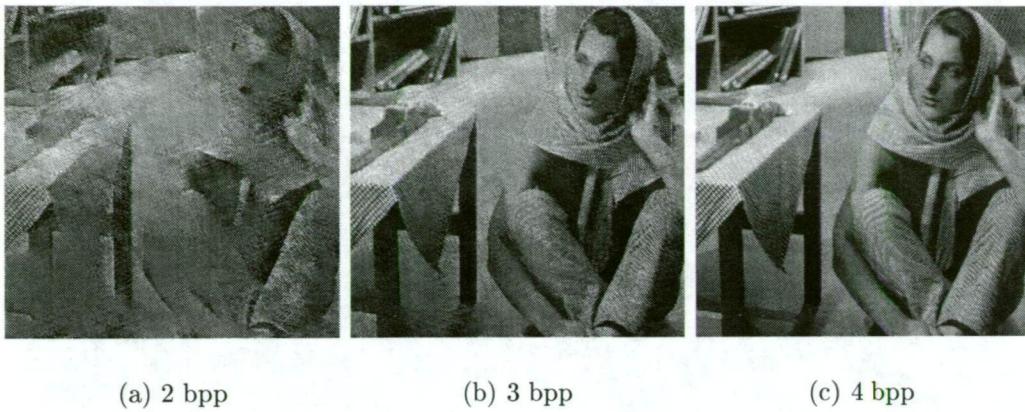


FIGURE 7.5.3. Results of using the Kovačević and Vetterli  $D_4$  wavelet for quincunx with uniform scalar quantization on Barbara image.

for the quincunx orthogonal filter because they emphasise a previous statement about the suitability of PSNR as a perceived quality measure.

bpp	Lenna	Goldhill	Barbara
2	18.44	17.56	16.13
3	24.41	21.79	21.19
4	30.50	27.31	27.50
5	36.75	33.71	34.04

TABLE 7.6. PSNR values from using orthogonal  $D_4$  filters for the quincunx lattice. These results show that the quincunx  $D_4$  wavelet produces good PSNR values.

Table 7.6 shows these PSNR results which are generally fractionally higher than the equivalent results for the linear phase KV5/3 filter set. This is a result which could be predicted since orthogonal filters are generally found to produce better SNR results than linear-phase filters - in fact the accuracy of orthogonal filters is the key reason for their use.

This situation is a good example of why the PSNR value was not used as a quality metric for this Thesis. Although in this case the orthogonal filters produce a higher PSNR and therefore a statistically more accurate reproduction of the original image, the result is of much lower perceptual quality because of the noticeability of the coding artifacts introduced by the filters.

**7.5.2. Modified Kovačević and Vetterli Filterbanks.** To have a successful coding method it is important that the filters have a smoother response than the KV5/3 filters. One such design was attempted in Section 4.2.4 which reduces the central tap value of the low-pass reconstruction filter. Results of using this filter are shown in Figure 7.5.4. As can be seen there is some change in the quality of the images reconstructed from only the coarsest subbands which encourages

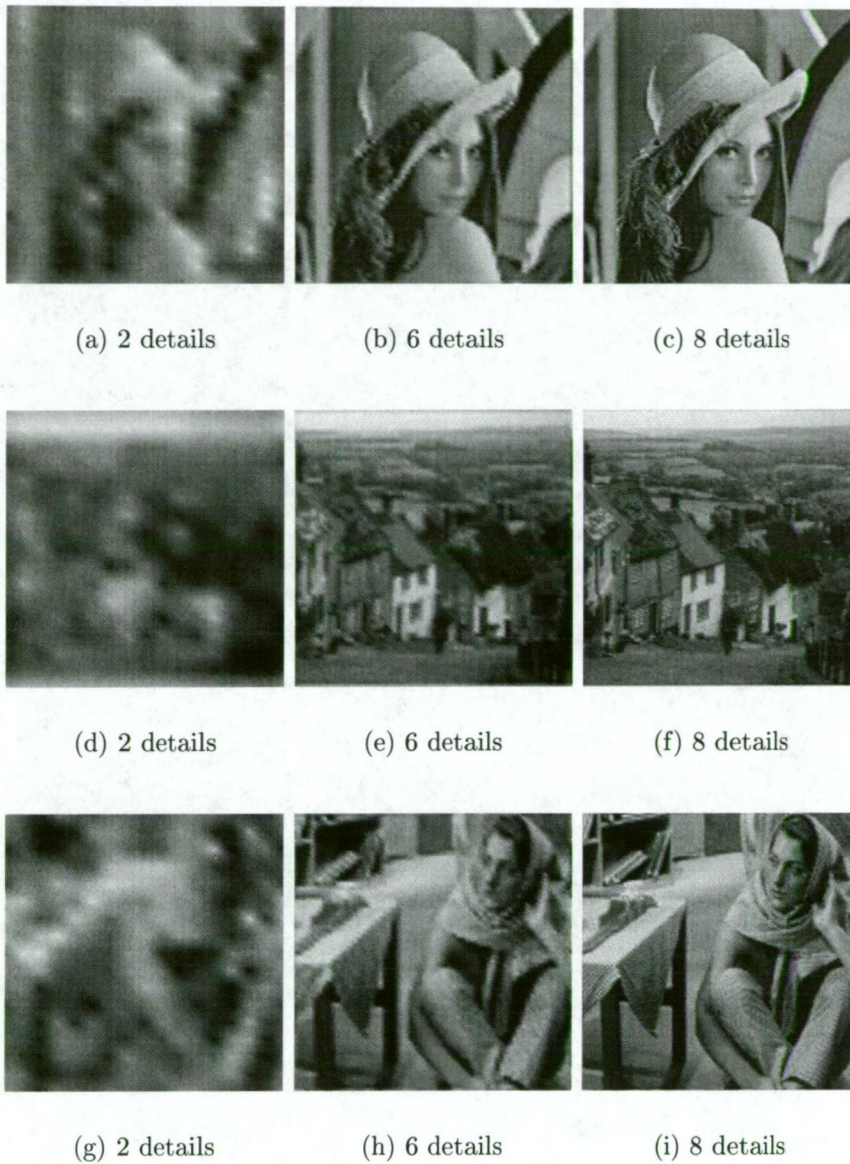


FIGURE 7.5.4. Lenna, Goldhill and Barbara decomposed and reconstructed using a modified KV5/3 filterbank with detail signals removed in the same fashion as 7.3.9. The modified filterbank is designed to reduce the central tap magnitude with respect to the neighbouring taps.

further investigation but the design of such filterbanks is well beyond the scope of this Thesis.



## CHAPTER 8

### Zero-tree Coding

In this chapter we review zero-tree coding and expand upon the ideas of Shapiro for the purpose of coding quincunx wavelet decompositions. We show that orientation specificity of the quincunx subsampling structure offers two alternative zero-tree structures but we find from results that there is no major advantage in either design.

Many of the ideas presented here are equally applicable to the separable domain as to the quincunx domain. Quincunx resampling was used as the test platform for much of the experimentation and separable results are incomplete due to time restrictions.

#### 8.1. Introduction To Zero-tree Coding

Zero-tree coding is a coding method which exploits the structure of a wavelet decomposition of an image. It is very efficient when the image has well separated areas of activity because the method efficiently codes regions with an absence of activity rather than the presence of activity. This property becomes very important for very low bit-rate compression because as threshold levels for coding become larger, areas of inactivity (approximately zero value) in the transform coefficients become the dominant feature. Only a small number of coefficients are needed to express the information of the image and they are generally closely clustered around major image features such as thick edges and lines.

**8.1.1. The zero-tree algorithm.** The zero-tree method looks at a wavelet transform by visual area, and in a hierarchical manner. It retains the concept of hierarchical parents and children in a multiresolution context. The key concept in zero-tree coding is the tree: this is a set of coefficients beginning at a particular spatial and resolution location and spanning all coefficients which are children of the tree root. We can consider coefficients as residing at particular locations within a 3-space defined by 2-D location and resolution depth. A child of a coefficient is defined as another coefficient at finer detail level, and thus lower in the decomposition hierarchy, contained within the spatial region covered by its parent. Every coefficient has one and only one immediate parent at the immediate next coarse level but one parent may have several immediate children. The number of children per parent depends on the decomposition tree type (eg. separable, quincunx, hexagonal, etc.). Separable, ie. Mallat pyramid decompositions

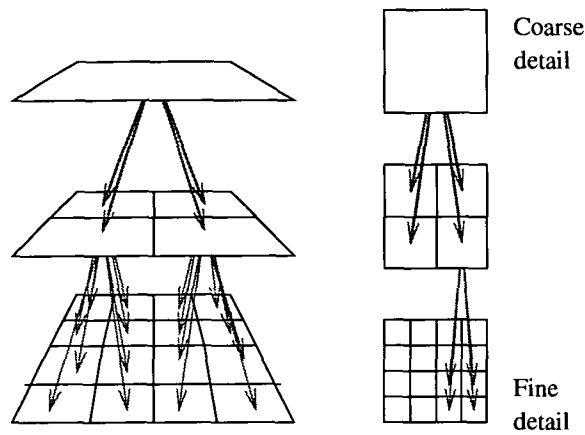


FIGURE 8.1.1. A tree of separable decomposition coefficients. Coefficients at coarse levels occupy a greater area in the image than those at finer levels. Several fine coefficients can reside within the same spatial region spanned by one coarse level coefficient.

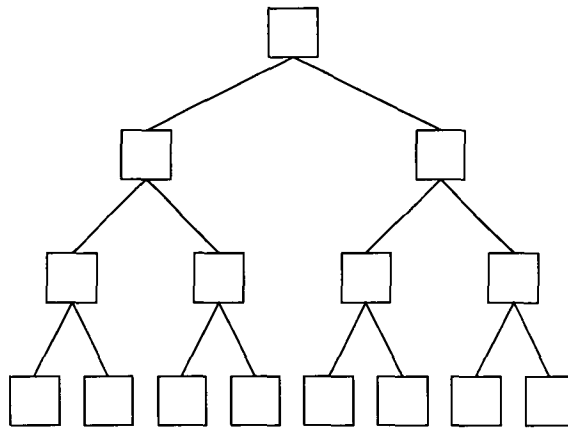


FIGURE 8.1.2. A binary tree as present in the Quincunx zero-tree representation. This shows a 4 level tree spanning a  $(2\sqrt{2} \times 2\sqrt{2})$  coefficient area at the finest level.

result in a decomposition tree with four children per parent, whereas the quincunx decomposition results in a tree with only two children per parent.

A zero-tree is a tree of coefficients in which the parent and all children, as far down the tree as it extends, are *zero*. In this case *zero* is used to mean that the coefficient is below the current threshold of significance. Zero-trees become common under very low bit-rate conditions when the majority of coefficients are below the threshold. In this case it is possible to reduce the majority of the area of a decomposition to just one or two symbols.

Due to the way in which image decompositions are structured, zero-trees occur frequently over many coefficients particularly when the coding threshold is high such as at the beginning of the coding process. This means that a rough reconstruction of the original image can be obtained with very little information by setting the threshold to a high level.

Figure 8.1.4 shows a wavelet coefficient tree for separable and quincunx decompositions. It can be seen that the separable method - which has more cosets

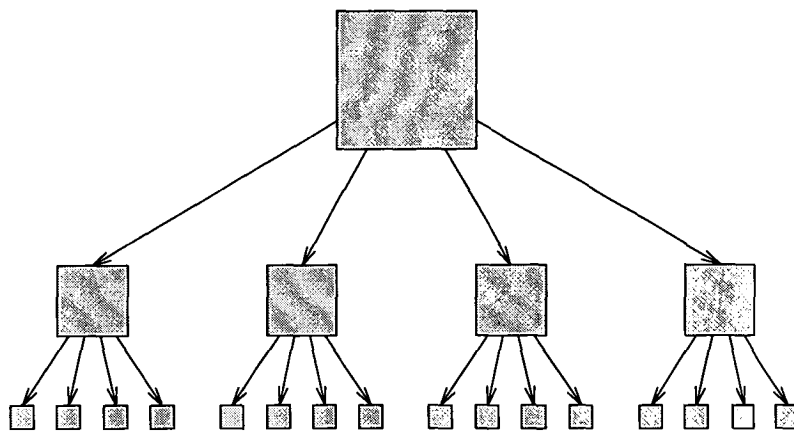


FIGURE 8.1.3. A quad-tree as present in the separable zero-tree representation. A 3 level tree is shown spanning a 4x4 coefficient area at the finest level

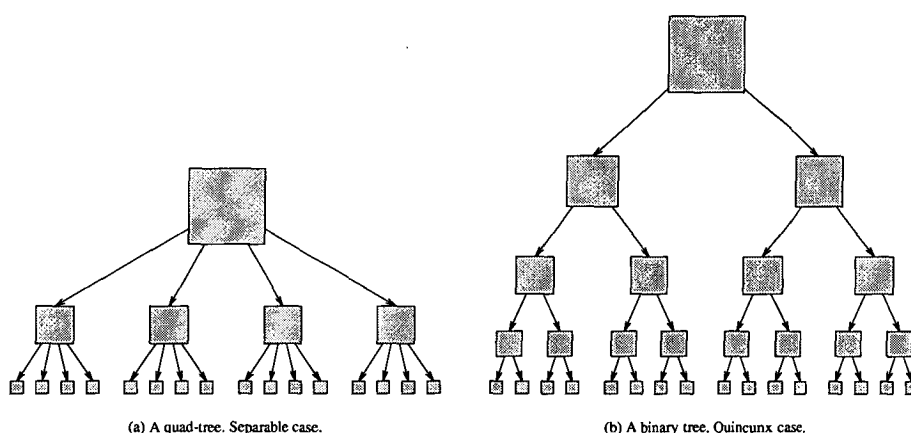


FIGURE 8.1.4. Comparison of binary and quad-trees covering the same area of coefficients at the finest resolution level. Both cover a 4x4 coefficient area.

- creates a flatter tree structure. This has the advantage of requiring fewer coefficients to represent the tree but it has the disadvantage that significant coefficients are more detrimental to the coding efficiency.

The coding process begins with an initial coding threshold to which coefficients are to be compared for significance. All coefficients are coded as one of four possible symbols:

- (1) (+) - significant and positive in sign
- (2) (-) - significant and negative in sign
- (3) (IZR) - insignificant (but with significant children)
- (4) (ZTR) - the root of a zero-tree (insignificant and all children are insignificant)

The zero-tree root (ZTR) symbol is the most powerful because it tells the coder that there is a tree rooted at its location and all coefficients which are children of the location in question are insignificant as well. Zero-trees are powerful because they remove the need to code (to any precision) a large number of values in the



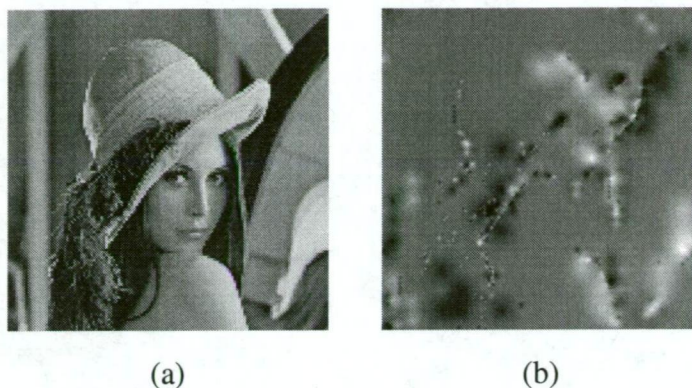


FIGURE 8.1.5. (a) Original Lenna image. (b) Reconstructed image from the information in one pass of a zero-tree coder, using a 12 level quincunx decomposition and the Kovacevic and Vetterli 5/3 filter set. Original size 262144 bytes. First-pass zero-tree 607 bytes. Rate 0.0023 bpp.

decomposition structure. Their value is inferred to be below the current threshold of significance and so not bits of precision are coded for them.

If the root of the zero tree is high in the decomposition hierarchy (ie. at a coarse resolution level) then the number of coefficients which are children can be very large resulting in an efficient method of expressing a large number of coefficients. As the coding threshold drops, the probability of a coefficient being insignificant becomes smaller and zero-trees become uncommon. Under these conditions the coding overheads of this method tend to outweigh the benefits gained from zero-tree coding. Other more uniform coding methods are better employed in this case. This is discussed further in Section 8.2.

Since the coding threshold only ever drops throughout the coding process, once a coefficient is known to be significant with respect to a threshold it will always be significant with respect to any future thresholds. This fact is important to keep in mind when explaining the coding process.

There are two sections to the coding process: often called the dominant and subordinate passes in the literature. Both passes scan through all coefficients in all levels of the decomposition. The purpose of a dominant pass is to determine which coefficients, that were not significant in the previous significance pass, have become significant due to a reduction in the coding threshold. It also codes whether a newly significant coefficient is positive (+) or negative (-) in amplitude and if an insignificant coefficient is isolated (IZR), or the root of a zero-tree (ZTR). A dominant pass always occurs immediately after the coding threshold is lowered.

Once the dominant pass is completed, the subordinate (or refinement) pass is performed. The purpose of the subordinate pass is to refine the precision of the magnitude of any coefficients which are currently significant, including those which have just been added by the dominant pass. For each coefficient in the map of significant values, one extra bit of precision is added to the value coded. There are only two possible symbols in the subordinate pass

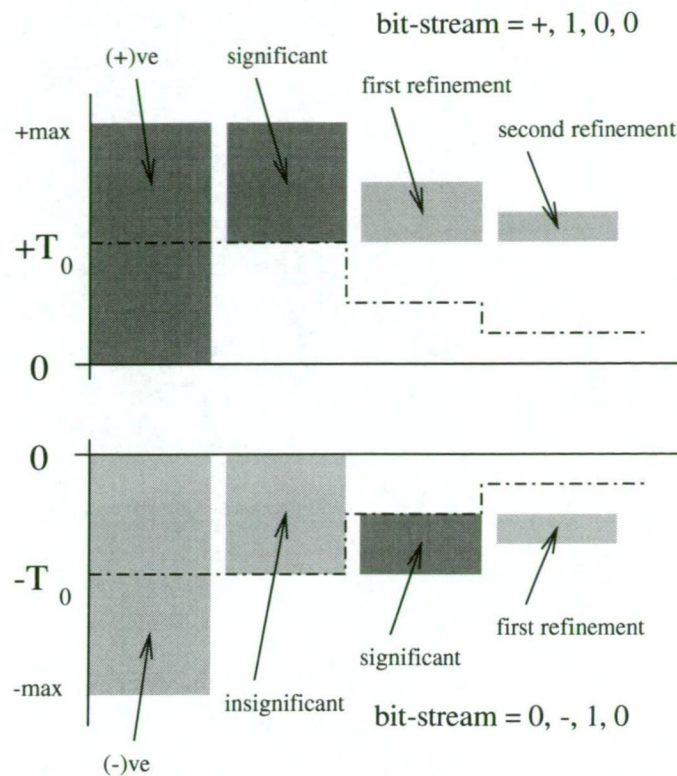


FIGURE 8.1.6. The reducing level of the coding thresholds in the zero-tree coding scheme. As the threshold drops the uncertainty about the value of a coefficient becomes less. Each halving of the threshold adds one bit of precision to the information known about the coefficient value.

- (1) (+) - Refine the value to the upper half of the uncertainty range
- (2) (-) - Refine the value to the lower half of the uncertainty range

The order in which coefficients are scanned is important because it defines the parent-child relationship between pixels. The parent-child relationship defines the order in which trees are created. It is important that the coefficients of greater importance are higher in the tree structure as this allows zero-trees to be constructed at lower levels while the important coefficients remain significant in magnitude.

**8.1.2. Break-even Considerations.** Zero-tree coding is very efficient at coding wavelet decompositions when the coefficient trees are very sparse. This typically occurs at the start of coding when the zero-tree threshold is high and so most coefficients are insignificant with respect to the threshold. With each pass, the threshold is lowered and more coefficients are likely to become significant. As the complexity of the significant coefficient tree becomes greater, the cost of coding the dominant and significant passes becomes greater.

For latter stages of the coding process we propose that it is more appropriate to use a more traditional bit-plane method for coding the remaining bits of the coefficients. The use of such methods can be justified for the least significant bits of the coefficients because the values of these coefficients are close to being random. This cannot be said for the most significant bits though, as Shapiro showed there



is a strong correlation between the position of low-valued coefficients in the tree, which all share the same first few most significant bits.

Obviously an intelligent image coder should recognize when the coding method it uses is no longer optimal. It may be more efficient to use zero-tree coding for the first few bits of precision of the coefficients and then use another method for the remaining bits of precision up to the desired depth.

Preliminary results obtained on this topic indicate that the zero-tree coding process is only useful up to about 3 complete passes. After this, the size of the significance map overhead counteracts any advantage gained by coding the remaining zero-trees.

Section 8.2 covers this in detail and provides projections of data rates for hybrid zero-tree coders.

## 8.2. Zero-tree Coding Results

In this section we present results on the zero-tree coding of quincunx wavelet coded image. All results refer to a 12 level decomposition using the KV5/3 biorthogonal wavelet filters of Kovacevic and Vetterli.

First we begin by examining some statistics of the zero-tree coding of the images and highlight the strengths and weaknesses of the zero-tree coding method. We then move onto combining the zero-tree coding method with adaptive arithmetic coding.

We find that the zero-tree coding method is extremely effective at very low bit-rates where the probability of zero-trees is high. We also show that the use of adaptive arithmetic coding in combination with zero-tree coding makes for an especially powerful coding system which delivers good image quality at very-low bit rates.

**8.2.1. Stand-alone Zero-tree Coding.** To gain an understanding of the strength of the zero-tree coding method we have produced results which isolate the number of symbols required to code particular images. Zero-tree coding proceeds in a two-step process which repeats. The first step is the significance pass which encodes the information about the location of newly significant coefficients within a wavelet decomposition. The second part is the subordinate pass which adds an extra bit of precision to the values coded for all significant coefficients in the entire decomposition.

The power of the zero-tree coding method comes from the zero-tree symbol (ZTR) which indicates that all children of a coefficient are ignored. This can efficiently code information about the insignificance of a large number of coefficients. The greater the number of zero-trees in a decomposition the more efficient the method becomes, and with greater depth to each zero-tree the efficiency is increased again.

pass	Lenna	Goldhill	Barbara
1	2082	3482	6131
2	6110	13166	21340
3	23610	51308	64716
4	51933	117168	82504
5	106300	159083	102232

TABLE 8.1. Table of significance pass symbols coded for the first 5 passes of zero-tree coding of KV5/3 wavelet decompositions. Subordinate pass numbers are given in Table 8.2.

pass	Lenna	Goldhill	Barbara
1	343	628	872
2	1554	3066	8998
3	7831	16046	21852
4	23698	52669	63194
5	55613	113062	100684

TABLE 8.2. Table of subordinate pass symbols coded for the first 5 passes of a zero-tree coding of KV5/3 wavelet decomposition. The corresponding significance values are given in Table 8.1.

pass	Lenna	Goldhill	Barbara
1	16.25	16.60	51.21
2	21.38	21.36	19.75
3	24.53	24.68	23.65
4	26.36	26.69	25.49
5	27.15	27.44	26.09

TABLE 8.3. Table of PSNR results for the zero-tree coding described in Tables 8.1 and 8.1.

It becomes apparent that the zero-tree coding method then is most effective for very harsh coding where the number of significant coefficients is very small and the probability of zero-trees of large depth occurring is very high. The zero-tree method can code an entire decomposition to one bit of precision very efficiently.

As the precision of the coding increases, the number of significant coefficients in the decomposition becomes larger. This reduces the effectiveness of zero-tree coding in two ways, firstly it increases the overhead of the significance and subordinate maps but more importantly it reduces the probability of zero-trees of significant depth occurring. Without a high probability of zero-trees the coding method becomes inefficient.

The data in Tables 8.1, 8.2 and 8.3 provides basic data on the behaviour of the zero-tree coder as the level of precision changes.

For the first pass of the zero-tree coder the significance threshold is set to be half the maximum value which exists in the decomposition. From the probability

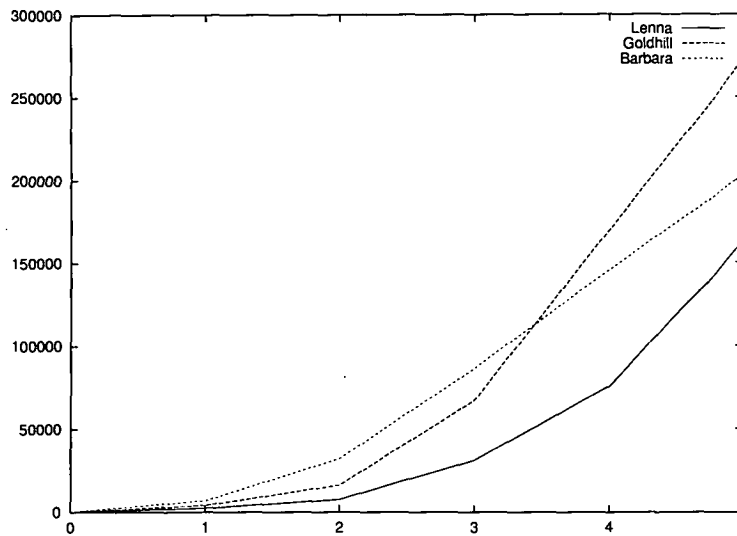


FIGURE 8.2.1. Plot of the total number of symbols coded with each zero-tree coding pass. The plot is not cumulative.

distribution of the wavelet coefficients discussed above in Section 6.3, the probability density drops to a negligible value above about 20% of maximum. Thus in the first pass we expect a very small number of significant coefficients.

Referring to Table 8.2 we see that the number of significant coefficients in the first pass ranges from 343 to 872. Note the subordinate numbers in this case give the number of significant coefficients – the significance map size gives the total number of all symbols (ZTR, IZR, +, -) in the significance map.

Referring back to the probability density plots in Section 6.3 we see that Lenna has the most concentrated PDF and this, as expected, is reflected in a low number of significant coefficients at the first level. Barbara which has the least compact PDF generates the highest number of significant coefficients at the first level.

As we move to the second pass of the coder, the threshold is now 25% of the maximum coefficient value possible. The effect on the symbol requirements is obvious and dramatic. The subordinate numbers indicate that between 1211 and 8126 new significant coefficients are generated. This increase in the number of significant features is reflected also in the significance pass numbers where the size of the significance map has increased dramatically. This increase is due to the drop in probability of zero-trees. When a zero-tree disappears, it is necessary to code all of its nodes in the significance map, when previously only the zero-tree root node needed to be coded.

We find that the zero-tree coding method is very effective for very-low bit-rate coding but becomes increasingly less effective as the precision of the coding increases. This indicates that after some number of passes of a zero-tree coder it becomes more economical to switch coding methods and use another method that copes better with more randomly distributed data.

We can easily translate the number of symbols coded into the data rate by noting that the zero-tree alphabet consists of four possible symbols (ZTR), (IZR),



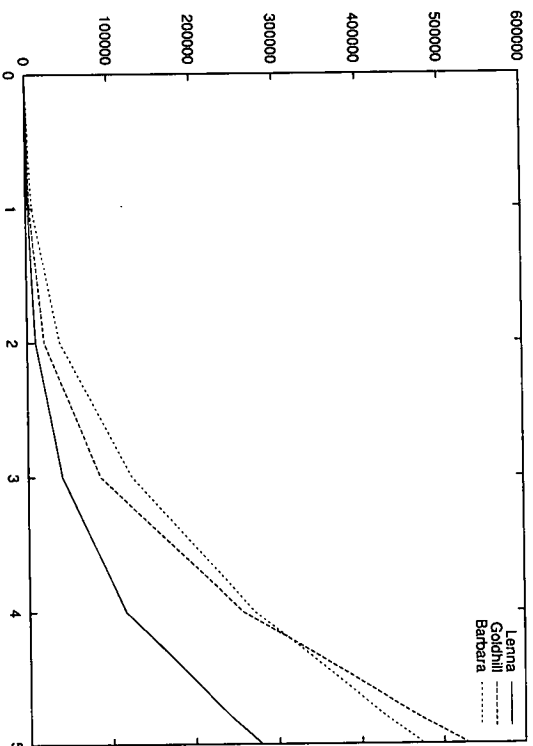


FIGURE 8.2.2. Cumulative plot of the total number of symbols coded for the first 5 passes of zero-tree coding. This is a cumulative plot of the data in 8.2.1.

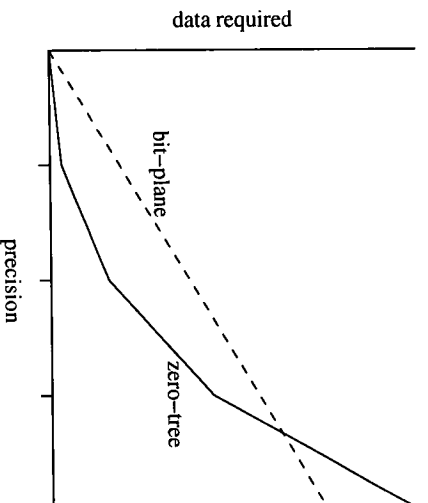


FIGURE 8.2.3. Projection of the comparative coding costs for a bit-plane coder versus zero-tree coding.

(+) and (-). This requires 2 bits per symbol and thus the number of bytes required is  $bytes = \frac{symbols}{4}$ .

Consider now naive bit-plane coding which codes all coefficients one bit at a time starting at the most significant bit. For the image size used which is  $512 \times 512$  there are 262144 coefficients in a critically sampled decomposition which requires 32768 bytes to encode. Thus we get one bit of precision over all coefficients for the cost of 32768 bytes.

In comparison the zero-tree coder has provided us with the first bit of precision of all coefficients in the decomposition for the price of between 606 and 1750 bytes. For the next level of precision the coding cost of the naive bit-plane coder is fixed at 32768 bytes for the pass, but the rate of the zero-tree coder increases as the probability of zero-trees decreases. At some point the rate of the zero-tree coder will exceed that of the naive bit-plane coder. Figure 8.2.3 shows a this concept.

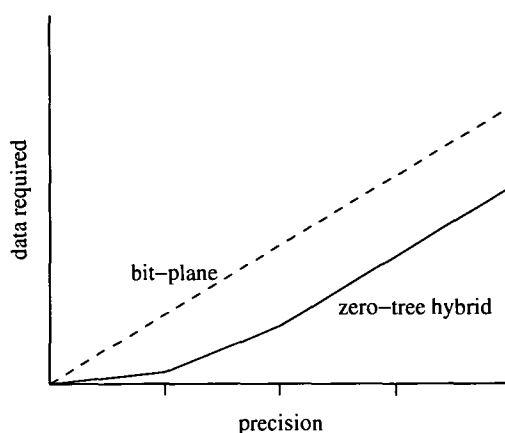


FIGURE 8.2.4. Projection of the use of zero-tree hybrid coder where the naive coder replaces the zero-tree coder when the rate of data for the zero-tree coder exceeds that of the naive coder.

Obviously an intelligent bit-plane coder will make better use of the probability distributions of the bit-planes it is coding and will have a better coding rate than the naive coder, but the naive coder remains as an upper limit to the data-rate which would be generated. So the fact remains that after some precision, the zero-tree coders will perform worse than the general case for coders.

The solution to this problem would be to replace the zero-tree coder with another bit-plane coder when the data-rate per bit of precision of the zero-tree coder exceeds the rate for the other coder, thus creating some zero-tree hybrid coder. Diagrammatic projections of the data rates in this situation are shown in Figure 8.2.4. This structure would allow us to take advantage of the zero-tree coder's capacity to efficiently code the first few bits of each coefficient, but we also never exceed the cost per bit of the other coder. In practice we often can't afford the time required to code the data with both methods which would allow comparison, such a system would need to predict the change in advance.

**8.2.2. Arithmetic Coding.** If we consider the way in which zero-tree coding progresses we find that an efficient complement is adaptive coding of the symbols. When processing the early stages of the bit-stream the significance threshold will be large and so many zero-trees will result while significant and isolated insignificant points will be considerably rarer. Thus in early stages of coding the ZTR symbol needs to have a very low information cost.

The next step after the significance coding is subordinate coding. At this point the alphabet changes from (ZTR, IZR, +, -) to (+, -) and so the coding must be changed to reflect the fact. Here both (+) and (-) symbols are expected to be of the same probability and so they should have the same cost, and the probability of ZTR or IZR must be zero since these symbols cannot occur in the subordinate stages of coding. So we have two unrelated data sets with significantly different probability distributions for symbols.

pass	Lenna	Goldhill	Barbara
1	611	1032	1755
2	2527	5090	9815
3	10337	21928	31457
4	29245	64388	67881
5	69723	132424	118610

TABLE 8.4. Table of file-size (in bytes) required for zero-tree coded images without Arithmetic Coding.

pass	Lenna	Goldhill	Barbara
1	443	692	1086
2	1758	3212	7365
3	7369	14570	24257
4	20914	43765	49392
5	48792	89273	83217

TABLE 8.5. Table of file-size (in bytes) required for zero-tree coded images with Arithmetic Coding.

As we progress to lower thresholds the probability of zero-tree roots becomes less and so our coding scheme must possess a means to reduce the information cost of (+) and (-) at the expense of (ZTR). Adaptive coding (eg. [46],[19], [92]) does this; it monitors the recent probability of symbols which have been coded and modifies its codebook appropriately. It does this in a predictable manner such that at the decoder the changes in codewords can be predicted without any side information. The efficiency of the adaptive coding can make significant advances on the coding performance of the system.

The ability to adapt quickly and effectively to changes in alphabet and symbol probability are important. Adaptive arithmetic coding has been shown [46] to perform well with small alphabets as the coding histogram can quickly establish symbol probability distributions when the probability of all symbols is relatively large. Also the need for escape sequences the first time a symbol is encountered means that with fewer symbols there is greater efficiency as the escape symbols are only encountered a small number of times.

All remaining zero-tree coded results generated for the production of this thesis make use of the Adaptive Arithmetic Coder implementation by John Danskin found in the Baseline Wavelet Construction Kit [19].

As can be seen from the results in Tables 8.4 and 8.5, the Arithmetic Coding produces a significant gain over the same stream without Arithmetic Coding. Unlike the zero-tree coding method its performance doesn't vary with data-passed to it. The output from the arithmetic coder is about 70% of the unprocessed data rate regardless of the properties of the original data.

Further improvements can be made from this situation since the Arithmetic Coder has not been optimized to recognize when a change of alphabet occurs,

rather it is left to adapt to the changing histogram of the data. By notifying the coder as the alphabet changes, the coding performance can be further increased.

**8.2.3. Cross-scale Considerations and Quincunx Trees.** When we look at the separable zero-tree coding algorithm we find that the choice of parent for any given point in a decomposition is easy. The decomposition occurs in a very regular fashion. Each decomposition level produces three oriented sub-spaces. Thus at any decomposition level and orientation we obviously choose the parent to be at the point with the same spatial position and same orientation at the next coarsest resolution level. This leads to three trees of zero-tree coefficients which all converge to a common parent at the approximation.

With the quincunx decomposition our choice of a parent becomes more complicated. Each level of decomposition produces only a single detail subspace. Immediately one would assume that this makes the choice of parent even easier than for the separable case since here there is only one choice with the same spatial position at the next lowest resolution.

The problem is that the quincunx decomposition has two orientations in its subspaces which alternate, so the next finer or coarser detail signal is of a different orientation than the the previous one. Levels  $2i + 1, i \in \mathbb{Z}$  generate diagonal details while levels  $2i$  generate horizontal-vertical details. In the separable case there are three independent trees - one for each orientation - and the next finer or coarser level is the same orientation.

As we can see now for the quincunx case, choosing the parent point from the next lowest resolution level will mean that the parent is chosen from a sub-space with a different orientation - which can impact on the coherence of zero-trees. If we want to choose the parent so it has the same orientation we must go to the subband two resolution levels removed, ie. from level  $n$  to level  $n + 2$ . It now becomes clear that the choice of a parent means making the choice of either resolution level similarity or orientation consistency. Figure 8.2.5 shows the parent-child relationship in the quincunx decomposition and the orientations of the subspaces involved.

Let us consider what we are attempting to encode when we use zero-tree coding. We are looking for areas in consecutive subbands where the magnitudes of the wavelet coefficients are all negligible which allows us to encode the entire block as one symbol representing a tree of zeros starting at a very low resolution.

Consider now the information contained in subsequent subbands as orientations alternate between diagonal and horizontal-vertical details. It is reasonable to expect then that features which generate details of negligible magnitude in one orientation may generate significant details in the other orientation; and then negligible in the following subband again.

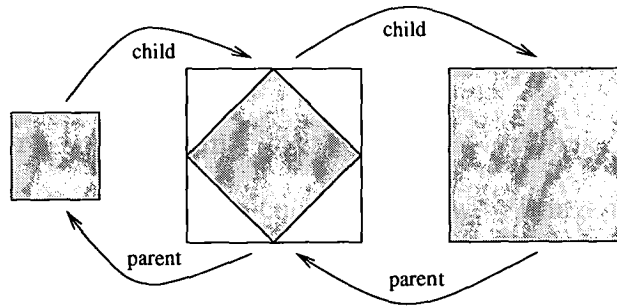


FIGURE 8.2.5. Orientations of successive subbands in a quincunx decomposition. Subspaces are of a different orientation than their immediate parents and children.

For example, consider a diagonal edge feature. In level  $j + 1$  we would have significant diagonal details along the edge due to the diagonal cutoff of the low-pass at this level. At level  $j + 2$  we have only horizontal-vertical details and thus the edge data will not be contained in this subband but at levels  $j + 3$  and  $j + 5$  etc. we again have significant details but at  $j + 4$  and  $j + 6$  we have nothing. From a zero-tree coding perspective this is inefficient since we can establish a tree of insignificant coefficients if we consider only similar orientations in the tree.

On the other hand if we do have areas where there is negligible activity in both orientations it would be more efficient to have the parents be direct that way a single zero-tree would be able to account for all the coefficients which would need two zero-tree roots if we used parents based on orientation. We expect that this cost is smaller than that of requiring isolated zeros at each alternate level, especially since the ZTR symbol will typically have much less information cost than the IZR symbol since it will be a dominant symbol in the early stages of coding.

Preliminary results obtained indicated that there was no advantage to using orientation based child-parent relationships. This was an unexpected result, but time limits have prevented further investigation.

### 8.3. Zero-tree Coding Methods for Network Transport

One of the important issues for consideration with image transport across digital networks is the speed of availability of information to faithfully reproduce images. Users become impatient waiting for an indication of the overall outcome. For this reason progressive coding is very important and highly desirable. There are various types of progressive coding available for use with image coding for network transport which have found implementation in JPEG[33], GIF[43], TIFF[42], PNG[106] and other image formats.

Normal quantization of wavelet decompositions codes coefficients in subbands in order, starting with coarsest resolution and ending with finest resolution. This



method allows for the image to be accurately rendered as an approximation at each resolution level in succession - an ideal situation for progressive transmission.

Zero-tree coding makes successively finer approximations to the image quality. Reconstruction starts at the coarsest resolution level but all subbands are reconstructed at some approximation level and then the next finer level of reconstruction begins at the lowest resolution.

This allows for a very low-quality approximation of the most significant features of the image to be obtained very quickly - more quickly than with the ordinary subband coding order. It also allows for the fast display of the most significant features throughout the image at all resolution levels. It may take longer to build up a high quality image which displays all necessary information to recognize what is needed in an image.

A disadvantage of the zero-tree coding method is that all the coefficients to build up a full resolution level image are coded before any refinement is made to the lower resolution levels. It might be more sensible to encode the lower resolution levels to a finer precision before moving onto the higher resolution levels. This could be done by processing and buffering the entire zero-tree stream, but transmitting the significance maps and subordinate passes of coarser resolutions first. This adds considerable complexity to the codec as the significance and subordinate streams need to be broken into segments and resynchronized.

A similar effect could be achieved by adding perceptual weighting to lower resolution subbands. By biasing these coefficients they appear greater in magnitude and are coded to a higher precision early in the encoding process and thus decoded with similar speed. However, the refinement then continues to a higher precision than is required which consumes more data than is actually needed to represent the image the desired quality.

The correct balance will depend greatly on the depth of the decomposition. If there are many levels of decomposition then a large amount of data must be processed to generate even a low resolution approximation of the image when using standard zero-tree coding. In this case it is sensible to break the coding process into segments, each constructing an approximation of increasing size for assessment.

Typically a thumbnail image needs to be in the vicinity of 40 to 80 pixels in at least one dimension<sup>1</sup> to distinguish major features. Depending on the size of the final lowest level approximation, this size image may be achievable with very little decoding. The next step in image detail should be in the area of 300 pixels in at least one dimension. This gives an intermediate resolution preview of the image, typically fine enough to ascertain if the image contains some particular feature required.

---

<sup>1</sup>The Exif digital photograph transfer standard [45] defines a standard embedded thumbnail size of 160x120 pixels.

Wavelet coding is also ideally suited to the encoding of extremely large images as has been demonstrated in recent times by MrSID codec from LizardTech[41] and the ECW engine from ERM[66] for Gigabyte and larger images. These codecs are capable of rendering sections of very large images with a particular bound on the most fine resolution which is considered. This can produce a zoomed section of the image which removes detail too fine to be displayed by the hardware raster sampling rate.

## CHAPTER 9

### Cross-scale Wavelet Image Coding

This chapter focuses on cross-scale coding methods such as zero-tree coding [81], [6], [79] which can be applied to the quincunx sampling lattice. Comparisons with the separable lattice will also be made. The techniques developed in this chapter are not restricted to the quincunx lattice but quincunx is used as the primary resampling method. Comparisons with separable methods are made where time permitted but these comparisons are not complete.

#### 9.1. Introduction To Cross-Scale Coding

The term “cross-scale” is used to describe a situation where information from several scales or resolution levels of a wavelet decomposition is used to code the image information as a whole. Most wavelet coding schemes to date have treated each individual detail signal as a separate entity which is to be coded in some way to maximize its reconstructed quality or minimize some error metric - a method which stems from orthogonal data processing but is not necessarily appropriate for perceptual coding. By considering the contribution of each detail as contributing towards a whole within a particular context then we can construct very efficient coding methods as will be demonstrated in the remainder of this work.

Each resolution level has particular significance to the HVS because of spatial-frequency sensitivity and orientation selectivity (see Sections 7.1.1 and 7.1.3 ) and so each level needs to be treated in a way which recognizes this fact. More important though is to consider the interactions between different scale levels. We must remember that it takes information from many scale levels to synthesize the features which we perceive in the reconstructed image. Using a deeper understanding of the way the components interact between all the scale levels leads to better efficiency than considering each detail signal in isolation.

One of the key features exploited by cross-scale coding methods is that image features are created by details from several scales which all reside within the same spatial region of the image (superposition principle). Because of the fact that wavelet detail signals retain spatial location information, we can easily determine which areas of an image any particular wavelet coefficient may contribute to and thus which features within the image are affected. Perceptually significant features such as edges and lines span a range of spatial frequencies, this is because of the abrupt change in signal value at an edge, ie. a high rate of change in image intensity. Frequency analysis tells us that to produce a large rate of change requires high-frequency components, and to maintain that change requires a large range

of frequency components from low to high with significant magnitudes. Because each wavelet detail signal spans only a narrow range of frequencies, significant sharp features require wavelet coefficients in several detail spaces yet at the same location within each detail.

**9.1.1. Zero-tree Coding.** Zero-tree coding is a very efficient method for encoding images at low bit rates where the majority of the information cost is associated with determining the positions of significant coefficients. It achieves success by using the common property of natural images which is that if there is no activity at low resolution levels in a certain region of the image then it is likely that there will not be any activity in higher resolution levels in the same area of the image. When such a situation occurs we can simply encode all levels of zero activity with a single symbol which is called a zero-tree root. A detailed examination of Zero-tree coding is presented in Chapter 8.

Zero-tree coding becomes less efficient as the bit-rate increases due to the increased number of significant coefficients. This reduces the likelihood of the large trees of zero values in the details from which zero-tree coding gains its efficiency.

**9.1.2. Quad-tree Coding.** Quad-tree [64] coding again uses an image hierarchy to efficiently predict behaviour across scales. The image is broken down into blocks which are each broken down into sub-blocks, each covering approximately the same area of image space at different resolutions. The key feature exploited here is that the statistical properties of the detail signals is similar at each of the levels due to the similarity in the underlying structure in the image which is represented. Hence analysis of a small number of coefficients allows for accurate prediction of many more coefficients within the same image area at various scales.

Quad-tree coding results are not presented here. The topic is discussed for completeness and to direct the reader to an additional related topic of research.

## 9.2. Spatial Masking Shadows and Cross-Scale Coding

Spatial contrast masking [72, chapter 4] (Section 7.1.2) is a phenomenon observed in the human perception of images. It has been noted that in the vicinity of dramatic stimuli, the presence of less significant fine features is obscured. This means that in areas where there is significant activity particularly at low resolution levels we can more coarsely code coefficients at higher (finer) resolution levels without any perceivable change in the image. We refer to an area where features are obscured by activity masking as a “Masking Shadow” cast by some large feature.

As the spatial distance from the stimulus becomes greater, the masking effect decreases. The compound effect of many significant stimuli in one area can lead to large sections of high resolution coefficients having little or no perceptual value, thereby allowing the coding process to discard a large number of coefficients and the information cost of coding them.

The use of spatial masking is not new; it has been used before, typically in the form of an “activity function”[67] ie. some measure of the degree by which activity in an area of an image is likely to obscure fine features. Activity functions are calculated from the original image (usually as some gradient of image intensity) and thus represent quite accurately the likelihood of masking occurring within a particular region of an image.

However to make use of an activity function in coding subband data the activity function data for all regions must be coded as side information. It is in this process where much of the information about the activity function is lost as the side information must be minimized. To code the activity function to any degree of accuracy would require significant information cost, which counters the gain made from discarding masked coefficients.

A novel approach is presented here and expanded in Chapter 10 for considering spatial masking in the wavelet domain. It requires no coding of side information and can be used to predict accurately the likelihood of masking occurring in the vicinity of a particular wavelet coefficient.

**9.2.1. Conditions for Masking.** Let us consider the conditions required to create a significant feature in an image which would lead to spatial masking in its vicinity. The feature must be highly visible; high visibility is attained by having significant contrast edges. Such high contrast edges require a large band of spatial frequencies to generate the sharp transitions in signal value. The wavelet representation of such a feature must therefore have significant coefficients at the spatial location of the feature and at many scale levels from medium-low resolution (basic feature outline) to high resolution (fine details).

Using this knowledge we have a simple method for determining the location of significant features in the reconstructed image which are likely to cause spatial masking and we require only the lower-frequency (larger scale) wavelet coefficients from the transformed image.

To make use of the information in a way which benefits the coding of images we need to be able to determine which coefficients can be discarded because of contrast masking. We have a way of determining significant coefficients which would lead to masking in their vicinity, so any coefficients residing in the vicinity of these features could be subject to masking. Since only subtle features will be masked, a requirement for masking is that coefficients should not themselves be contributors to a highly visible feature. Using the same argument as used above, we can see that a coefficient is part of a significant feature if there are other coefficients of similar large magnitude in the same spatial vicinity and in close proximity to the scale of the coefficient. If two coefficients are separated by many scale levels in which all the coefficients at the same location are very small then they are not likely to be part of the same feature.



Obviously for the coefficients to be in a situation where their presence would be greatly masked, they must be of a small magnitude. Large magnitude coefficients will have a significant visual presence whether they are combined with other coefficients or not.

By applying this method of masking shadow prediction we find that we can achieve significant increases in the perceived image quality with the same data rate. Chapter 10 is dedicated to expanding these concepts and results are presented in Section 10.3.

**9.2.2. Importance of the Image Average.** When performing a wavelet decomposition it is often advantageous to remove the DC value beforehand. This reduces the magnitude of the low-pass coefficients substantially. Now, rather than having many large positive coefficients in the low-pass with values close to the image average, those coefficients will have values close to zero, but with both positive and negative amplitude. This reduces many of the coefficients to zero during quantization coding, which increases the likelihood of information cost reduction particularly with zero-tree coding. Once the image has been fully reconstructed, the value of the image average can be again added back in. It makes little difference if the average is removed before any wavelet decompositions or only for the last level, however, there is less computational effort involved in calculating the average of the small sized low-pass image, when compared to the same calculation for the full sized image. Since the average is preserved by low-pass processing, there is no requirement for it to be calculated before or after wavelet decomposition - the value should be identical in both cases.

For a coding method such as zero-tree coding the removal of the image average is very significant. If the lowest resolution image has a DC component then many of the values will be truncated to a low precision during coding to meet the information cost constraints. Low precision in the low pass approximation leads to highly visible distortions in the reconstructed image. However if the approximation has no DC component then the same information cost can code the approximation to a greater precision. The reason for this is that the zero-tree method works on a four symbol alphabet (+, -, I, Z). If we changed the alphabet to have only three symbols (+, I, Z) where all coefficients are non-negative, then the information costs would be comparable for both cases.

It should be noted that these arguments do not hold in the case where quantization rather than zero-tree coding is used as the first bit of information in a large positive coefficient is equivalent to the sign bit of a zero-based signed data value.

### Masking Shadows in Zero Trees

In this chapter the concept of masking shadows, presented in Chapter 9, is treated in greater detail. The concept is built upon to create a complete coding framework based on the ideas of the zero-tree coding of Shapiro. We show through perceptual testing and statistical results that this coding framework provides us with a means to further reduce bit-rate and improve perceived image quality.

Many of the ideas presented here are equally applicable to the separable domain as to the quincunx domain. Quincunx resampling was used as the test platform for much of the experimentation and results are based on this.

#### 10.1. Utilizing Masking Shadows

In this section we show how to exploit the psychovisual phenomena discussed in Chapter 6 to produce a novel, efficient image coder based on zero-trees of quincunx wavelet coefficients.

Spatial masking occurs in the vicinity of significant high-contrast image features, notably sharp strong edges. The presence of these strong features causes areas in the image where less prominent details are easily lost, being perceived as noise. We refer to these areas as “masking shadows” because the presence of a significant feature produces an area of the image where it is difficult to discern fine detail - analogous to the shadow cast over small objects by larger objects.

The zero-tree coding method provides a near ideal environment in which to predict and utilize these masking shadows in a coder. To understand how we can do this we need to start by considering - from a different perspective - what wavelet transforms are and how they fit together in a zero-tree hierarchy.

See Section 9.2 for an introduction to the principles of masking shadows. We concentrate here on the algorithm to implement a codec.

To construct a large magnitude, high-contrast feature requires the superposition of components from different frequency bands, from the mid to highest frequencies. If we consider what this means in terms of a wavelet transform, such a feature when decomposed would create significant coefficients in many detail signals of a decomposition, all overlapping in their area of support at the point corresponding to the location of the feature in the original image.

From a synthesis or reconstruction perspective, this then provides us with a means of predicting the location of large magnitude, high-contrast features from a wavelet decomposition. The decomposition must first be considered as a layering of signals, where signals of many different resolution levels have coefficients

at common spatial locations within the image signal. Where there is an alignment/superposition of the detail signals over several resolution levels, there is likely to be a significant feature; and from what we've already seen (Section 7.1.2), the presence of such a significant feature induces a masking shadow causing small details to become imperceptible to the HVS. It is comparatively easy to determine if the intensity of an image is changing rapidly by simply examining the image itself, in particular the gradient of intensity over space, and this is indeed the method adopted in some previous investigations into this topic ([68], [67], [34]). This method has a fundamental disadvantage which is the side information which is required.

To effectively make use of the principles of contrast-masking in a coding context we can more coarsely code those areas of an image which fall in a masking shadow. To be able to decode the information we must also have a means of determining at the decoding stage, which areas were encoded with the coarser granularity. In previous methods this has required the coding of side-information with the image data which describes the level of masking within various parts of the image (sometimes referred to as the activity factor [67]). However there is a way, using zero-tree coding, to take advantage of the benefits of coarsely coding masked areas without the need for sending any side information. It is this method which makes for particularly successful coding.

**10.1.1. Implied Activity Measures.** Consider the first pass of a zero-tree coder on an image, as in Figure 8.1.5. The information which is contained in the first pass through all detail levels is those few high-frequency coefficients which are significant with respect to the highest threshold level of the coding process. There are only a few coefficients (across all resolution levels) and they correspond to the location of the highest magnitude variations in image intensity as detected by the high-pass filter branch of the wavelet decomposition.

Considering the information in terms of a decomposition layering, the superposition of significant coefficients indicates the location of those very large, high-contrast features which induce masking. It is possible to use the first small section of an embedded zero-tree code to predict those areas of an image which would likely be affected by contrast masking shadows.

It is important to note that the masking shadows are predicted from information which is available in identical form at both the decoder and the encoder.

We are concerned with altering the coding of those parts of the image which are obscured by masking shadows and we do not want to interfere with the coding of the significant features as are contained in the first pass of the zero-tree code, only those features which are contained in subsequent passes will be of sufficiently small magnitude to be obscured by masking shadows. This gives us a natural means of discriminating those coefficients which can be considered as being affected by masking shadows.

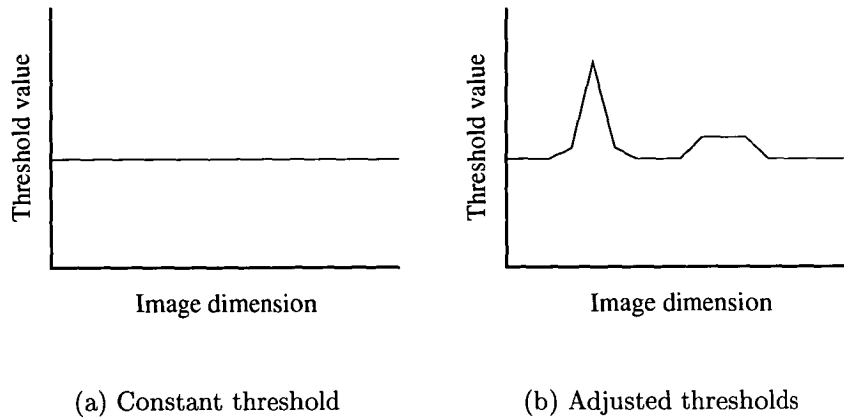


FIGURE 10.1.1. Normal zero-tree coding assumes the same threshold should be applied to all coefficient regardless of location or resolution. The masking shadow adjustment modifies the threshold associated with coefficients affected by contrast masking.

**10.1.2. Modifying the zero-tree algorithm.** We do not wish to alter the coding of the significant features so anything which is significant in the first pass of the zero-tree coder will not be altered. One way to increase the coarseness of coding in the zero-tree domain is to raise the threshold which determines if a coefficient is considered significant. Only significant coefficients are coded with any precision. In normal zero-tree coding the threshold used is global across all coefficients in an entire decomposition. This is not a requirement for zero-tree coding to work, what is necessary is that it is possible to predict in the decoding stage what the threshold used in the encoding stage was. It is also necessary that the threshold doesn't fall at a rate greater than the data precision increase provided by a single pass of the subordinate coding stage.

*PROPOSITION. We propose that the threshold used to determine if a coefficient is significant or not, be unique to a particular location (including resolution level) in a decomposition rather than global across all coefficients in a decomposition.*

If we raise the threshold of any location then we decrease the probability that the coefficient at that location will be significant when compare to its individual threshold. Thus we can delay the inclusion of any coefficient into the significance map by raising the zero-tree threshold associated with it.

It is important that a threshold never be lowered for any coefficient; this would cause such a coefficient to be included into the significance map at an earlier stage, possibly a stage that has already been completely coded, thus corrupting the coding. Similarly it is not possible to alter the threshold of any coefficient which has already been coded as significant since it has already been coded to 1 bit of precision. What can be done is to selectively raise the threshold of certain coefficients which correspond to minor features within the masking shadow generated by significant features.

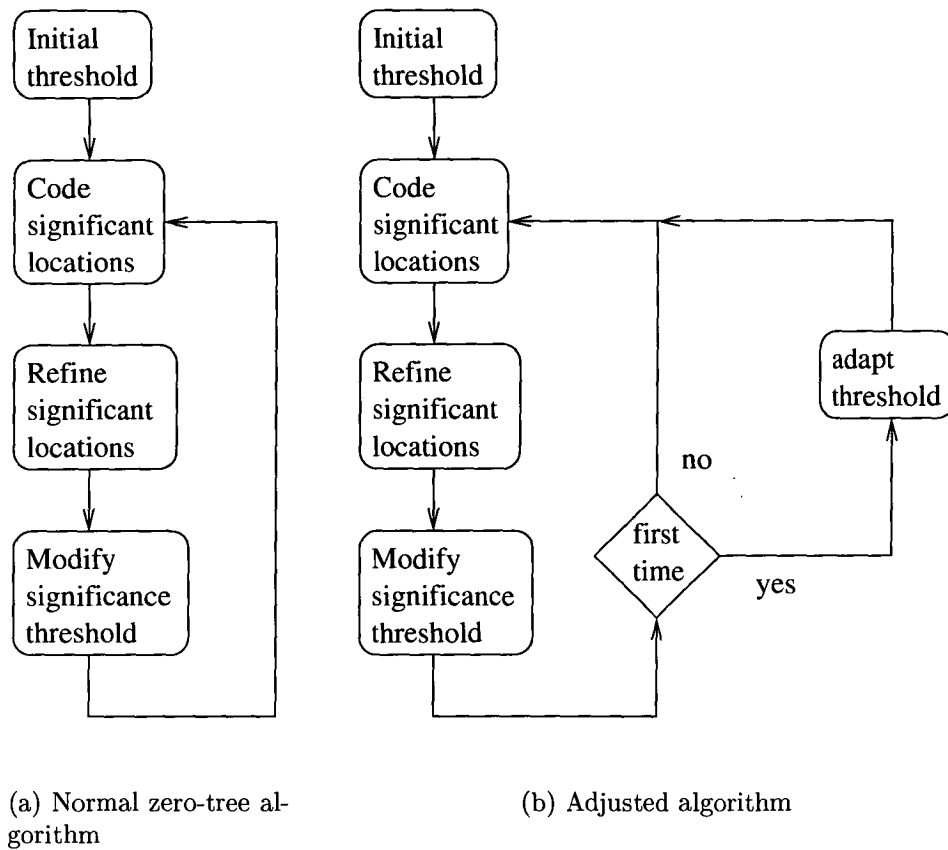


FIGURE 10.1.2. Flow-diagrams for the normal zero-tree algorithm and the masking adjustment zero-tree algorithm. The algorithm is applicable to coding and decoding processes. For the masking shadow version, the modification of significance threshold is done individually for each coefficient in the decomposition.

**10.1.3. Putting It All Together.** We now have all the tools we need to put together the coding scheme. Firstly we can predict the location of masking shadows by detecting the high magnitude coefficients responsible for such shadows in the very first significance pass of the zero-tree coding scheme. We can then determine which areas of the image are affected and raise the zero-tree significance threshold for these areas in accordance with the degree of masking which is predicted. The zero-tree coding process then continues to completion, and those coefficients which were subject to contrast masking will be coded with a higher granularity than those unaffected.

Figure 10.1.2 shows how the adapted zero-tree algorithm differs from the Shapiro method. When decoding, we start with the first significance pass of the zero-tree stream which is all we have. After decoding the significance pass we have exactly the same information as was used in the encoding stage of the codec to modify the significance thresholds of masked locations. By applying the same algorithm used in the coder we will arrive at exactly the same individual threshold values as were determined by the encoder. The zero-tree decoding can then

progress to completion with the modified thresholds. Those image areas subject to masking shadows will have been coded with less precision because of their delayed inclusion into the significance map.

It should be noted that the reduction in threshold value that occurs at the end of each zero-tree pass must now be calculated individually for each coefficient. The reduction is performed by dividing all threshold values by the same constant value, so the ratio of one threshold to any other remains constant.

## 10.2. Determining Masking Levels

That part of the codec which has been ignored so far is the part responsible for determining which threshold values to change after the masking shadows are predicted and by how much to vary them. This is the part of the coder which can be tuned and is open to most variation. The general principles of the coder are laid down. Although these cannot be fixed, for the coder to be effective it must accurately predict which coefficients are going to be masked and what level of increase in granularity the coding process can suffer without the reduction in precision becoming noticeable.

For the purposes of experimentation performed in the production of this work and [7] the masking level was determined from a set of parents at the same location as any prospective insignificant coefficient. The masking contribution of significant coefficients is broken up into two influencing parts, immediate neighbours and parents. Immediate neighbours can have an affect but it will generally be less than that of significant parents of neighbours. Each significant neighbour of a parent adds a weighting to the threshold modification based on the HVS sensitivity measure for the subband in which the significant value occurs. If there are many significant neighbours of parents then the masking weightings are cumulative to a degree.

Numbers which are found to produce good results, are a bias of 25% immediate masking and 75% parental masking. For each parent considered, its parents and immediate neighbours are again weighted in the same fashion. The maximum increase in the threshold of a coefficient is limited to 100% which guarantees delaying the inclusion of the coefficient into the significance map by at least one pass (but no more than one pass) of the zero-tree coder.

Results of the coding process are presented below in section 10.3.

## 10.3. Masking Shadow Prediction Results

In this section we present results which show the effectiveness of the masking shadow prediction algorithm with zero-tree coding. We contrast reconstructed image quality and file-size for coding that includes masking shadow prediction (referred to as masking shadow versions) with coding that uses only the zero-tree coding (referred to as ordinary versions).





(a) No masking prediction

(b) 4-neighbourhood masking prediction

FIGURE 10.3.1. Lenna image (a) reconstructed from 3 complete passes of a zero-tree codec with no masking prediction (10kB at 0.30 bpp) and (b) with 4-neighbour masking prediction at the same bit-rate.

All results make use of the Arithmetic Coder outlined above to produce the output streams, thus the numbers quoted are actual stored file sizes.

We find that the use of masking shadow prediction leads to a decrease in the bit-rate of all images processed with it when processed to the same Zero-tree end-point. Typical bit-rate reduction is in the area of 18% of total stream size with reductions of up to 27% observed. Subjective quality assessment of the images shows the two versions to be of very similar quality.

Processing to a Zero-tree end-point is an unsuitable means of performing comparative assessment since there are two non-constant components in each comparison – bit-rate and image quality. By holding the bit-rate constant the comparisons are much easier to make and more reliable. We find that the image quality using masking shadow prediction is noticeably better when compared to the ordinary processing. For some images the improvements are quite striking, and for others very subtle.

After presenting the result image pairs to a set of disinterested viewers the images processed with masking shadow prediction were found to have better image quality. In some cases the differences occur in areas of low importance to the image – notably Boat and Airplane images and in these cases the viewers noted that the images were very difficult to distinguish.

Referring to Figure 10.3.1 both Lenna images are of very similar quality. Areas to note are the vertical frame at the left of the image. In the ordinary version (a) there is definite corruption of the vertical edge, but in (b) this has been severely diminished. There is also improvement in the area of the lips and the hair strand at the shoulder.



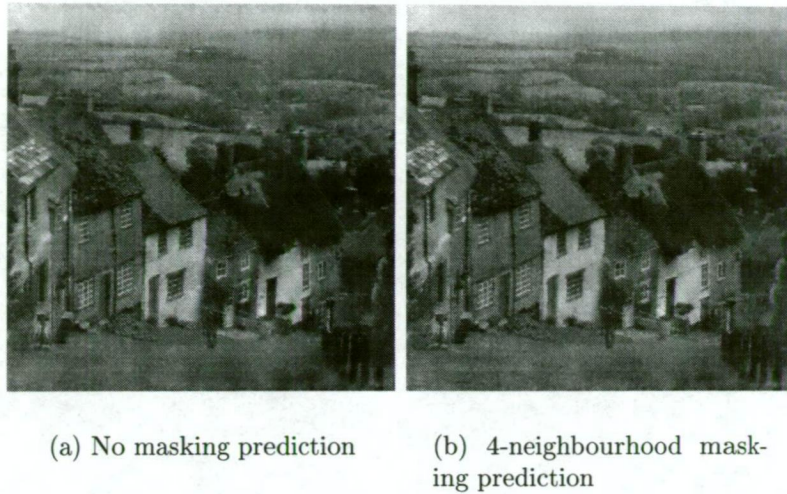


FIGURE 10.3.2. Goldhill image (a) reconstructed from 3 complete passes of the zero-tree codec with no masking prediction (19 kB at 0.59 bpp) (b) with 4 neighbourhood masking prediction at the same bit-rate.

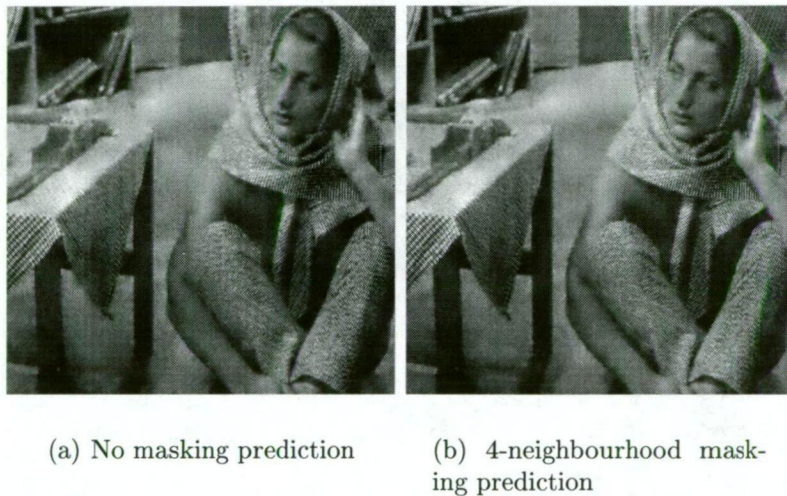
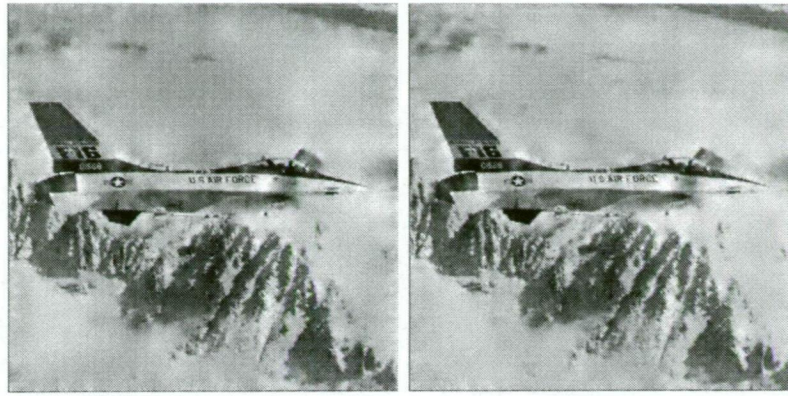


FIGURE 10.3.3. Barbara image (a) reconstructed from 3 complete passes of the zero-tree codec with no masking prediction (28 kB at 0.89 bpp) (b) with 4 neighbourhood masking prediction at the same bit-rate.

Both Goldhill images in Figure 10.3.2 were of very similar quality. In this case some viewers indicated that they could not see a noticeable improvement from one to the other. There are slight improvements which can be seen in the sky and horizon.

The Barbara images in Figure 10.3.3 are of similar quality but the masking shadow version (b) demonstrates improvements in quality in the aliasing effect on the pants leg near the centre of the image. In (a) there is significant strobing of the lines on the pants, but in (b) the lines are clearly separated.

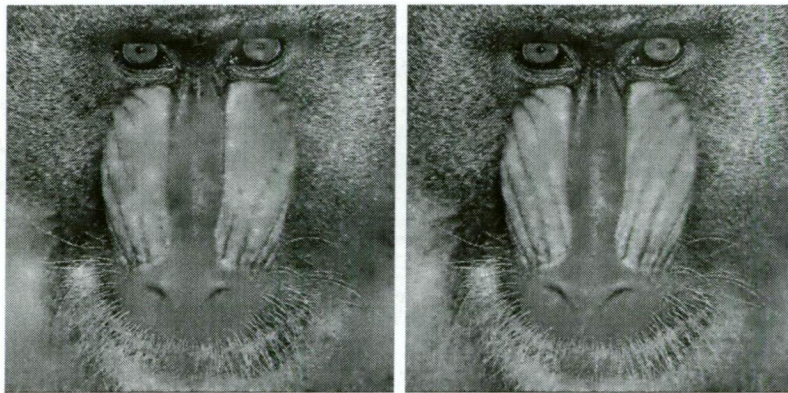




(a) No masking prediction

(b) 4-neighbourhood masking prediction

FIGURE 10.3.4. Airplane image (a) reconstructed from 3 complete passes of the zero-tree codec with no masking prediction (11.2 kB at 0.34 bpp) (b) with 4 neighbourhood masking prediction at the same bit-rate.



(a) No masking prediction

(b) 4-neighbourhood masking prediction

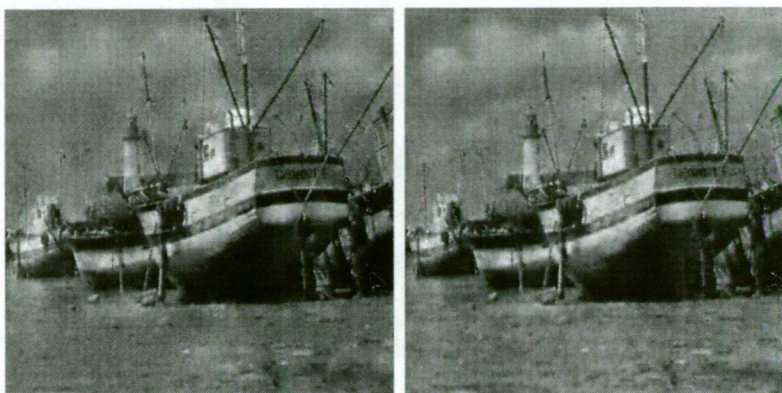
FIGURE 10.3.5. Baboon image (a) reconstructed from 3 complete passes of the zero-tree codec with no masking prediction (30 kB at 0.94 bpp) (b) with 4 neighbourhood masking prediction at the same bit-rate.

Other minor improvements were noted in the book spines and the nose and lips area of the face.

For the Airplane images shown in Figure 10.3.4 there were areas in the masking shadow version (b) which exhibited reduced quality; the front of the tail in particular. One viewer noted the pilot was more clearly defined in (a). There are subtle but major (in spatial area) image features which are defined in (b) but not in (a). In particular the clouds in the upper left are more defined in (b).

Baboon as shown in Figure 10.3.5 is one of the most striking examples of improvements due to the use of masking shadow prediction. There are significant





(a) No masking prediction

(b) 4-neighbourhood masking prediction

FIGURE 10.3.6. Boat image (a) reconstructed from 3 complete passes of the zero-tree codec with no masking prediction (11 kB at 0.35 bpp) (b) with 4 neighbourhood masking prediction at the same bit-rate.

facial features visible in the masking shadow version (b) which are blurred out in the ordinary version. Of particular importance are the nostrils and the lines on the side of the nose - particularly on the right side of the image.

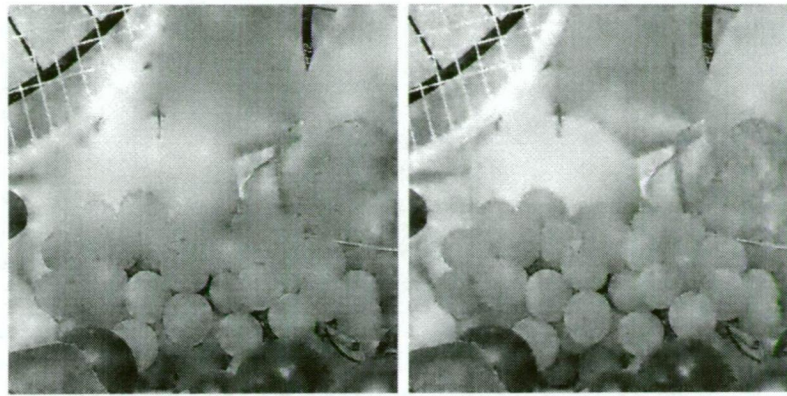
The Boat image shown in Figure 10.3.6 exhibits non-obvious gains in quality in the masking shadow version (b) over the non-masked version. In particular the texture of the ground and the edges of the clouds are more clearly defined in (b). The low perceptual importance of these areas to the overall image was demonstrated by several viewers who could not immediately determine the image which they believed had better quality. Some viewers indicated that there were features which were better defined in each of the images - in particular the name on the back of the boat is more clearly defined in (a).

The fruits image as shown in 10.3.7 has possibly the lowest quality reconstruction of all the images processed. This is due to the large areas of almost constant intensity broken only by fine lines. However all viewers noted that the version processed with masking shadow prediction (b) had significantly higher quality. Of particular note were the grapes at the back of the bunch and the edges of the artichoke on the right.

The Girl image (Figure 10.3.8) demonstrated another striking improvement when processed with masking shadow prediction (b). The quality improvement is quite dramatic especially in the areas of the mouth, nose, wool clothing and face-paint. All viewers immediately indicated (b) as the higher quality image without any hesitation.

The Peppers image as shown in Figure 10.3.9 has a high level of activity and exhibits clear gains with masking shadow prediction processing (b). Of particular interest is the texture on the left of the front pepper which is blurred out in (a)





(a) No masking prediction

(b) 4-neighbourhood masking prediction

FIGURE 10.3.7. Fruits image (a) reconstructed from 3 complete passes of the zero-tree codec with no masking prediction (6.5 kB at 0.20 bpp) (b) with 4 neighbourhood masking prediction at the same bit-rate.



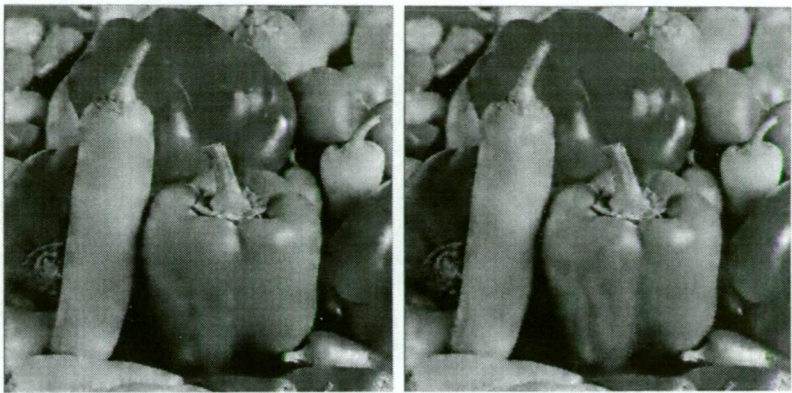
(a) No masking prediction

(b) 4-neighbourhood masking prediction

FIGURE 10.3.8. Girl image (a) reconstructed from 3 complete passes of the zero-tree codec with no masking prediction (7.0 kB at 0.22 bpp) (b) with 4 neighbourhood masking prediction at the same bit-rate.

but quite visible in (b). The gains in overall perceived quality are comparatively small for this image with some viewers requiring an extended period of comparison to determine which was of better quality.

**10.3.1. Perceptual Rating Results.** The observers of the reconstructed images were asked to categorize the relative quality of two compressed images they were presented with, taking into account the original image. The images were shown in a randomized order to remove bias based on previous ratings. Observers were asked to create their own viewing conditions of ambient lighting, monitor



(a) No masking prediction                      (b) 4-neighbourhood masking prediction

FIGURE 10.3.9. Peppers image (a) reconstructed from 3 complete passes of the zero-tree codec with no masking prediction (10 kB at 0.31 bpp) (b) with 4 neighbourhood masking prediction at the same bit-rate.

Image	Score
Airplane	-0.12
Baboon	1.06
Barbara	0.65
Boat	0.41
Fruits	1.00
Girl	1.47
Goldhill	1.06
Lenna	1.00
Peppers	0.82

TABLE 10.1. Average scores rating relative subjective quality of masking shadow prediction against ordinary zero-tree coding for image compression. Positive scores favour of masking shadow prediction, negative scores favour ordinary zero-tree coding.

resolution, brightness and gamma etc., the only restriction placed on this was that both images be observed under the same conditions.

The observers were asked to categorise the differences between images as one of:

- One image is clearly superior
- One image is slightly superior
- Neither image is superior to the other

Numerical weightings were applied to these categorisations as follows

- Clear difference scores  $\pm 2$
- Slight difference scores  $\pm 1$
- No superior image, both score 0.



- If the superior image uses masking shadow prediction, the score is (+)ve.  
If the superior image uses ordinary coding, the score is (-)ve.

This method is not statistically rigorous by any means but it allows for a clear analysis of trends observed in the subjects' judgment of image quality. Positive scores favour masking shadow prediction, negative scores favour ordinary coding. The average scores of the various images are given in Table 10.1. The number of observers tested was 17.

The scores presented reflect the items of discussion mentioned above regarding each image. Clearly the most dramatic effect is seen in the "Girl" image which most observers (59%) classified as clearly superior to the ordinary zero-tree version. All observers classified it as somewhat better quality. The other images to score very highly in favour of masking shadow prediction were "Baboon" and "Goldhill".

The average score of most of the images presented is positive in favour of the masking shadow prediction coding method which shows that the gains received from it apply to general image processing. The only image to receive a result against masking shadow prediction was "Airplane". "Airplane" and "Boat" were the images which scored the closest to the neutral zero point. The reasons for this are discussed above in the discussions for Figures 10.3.4 and 10.3.6. When examining the images for a second time the observer realises that there are features visible in the masking shadow versions of both "Boats" and "Airplane" which are not visible in the original. However these areas are of low importance as observers tend to focus on the high-contrast areas immediately. Indeed this is the key reason that masking shadow prediction works.

Clearly the amount of perceived quality gain from masking shadow prediction is dependent on image content. That only one of nine images was judged in favour of ordinary coding, and then only by the smallest of margins, is a ringing endorsement of masking shadow prediction and shows its suitability for general image compression.

**10.3.2. PSNR Results.** If we consider the PSNR values for the reconstructed images, as shown in Table 10.2, we see that the "masking (br)" column has the highest values of the three processing methods for all images. Next is the "ordinary" column and last is the "masking (ep)" column.

These results indicate that masking shadow prediction is good at decreasing the statistical error when compared to normal zero-tree coding. This is an unpredicted result considering that the masking shadow prediction method is based on principles of human perception rather than statistical analysis. The results can be interpreted as being consistent with the higher perceived quality of the masking shadow processed images, however several of the PSNR results are in direct contradiction to the subjective test results. The "Airplane" image is the most obvious example of this as the PSNR values indicate that the masking shadow version

Image	ordinary	masking (br)	masking (ep)
Airplane	24.20	25.65	23.56
Baboon	22.43	23.77	21.55
Barbara	24.30	25.22	23.65
Boat	23.63	25.57	22.99
Fruits	23.98	26.32	23.51
Girl	23.53	25.34	23.15
Goldhill	25.23	25.85	24.68
Lenna	25.04	26.08	24.53
Peppers	24.87	25.46	24.11

TABLE 10.2. PSNR values of reconstructed images. The “ordinary” column refers to processing without masking shadow prediction. The “masking (br)” column refers to processing with masking shadow prediction resulting in a bit-rate equal to that of the “ordinary” data stream. The “masking (ep)” column refers to processing with masking shadow prediction to the same zero-tree processing end-point as the “ordinary” column.

Image	ordinary	masking (ep)	$\Delta$ rate
Airplane	11240	8438	24.9%
Baboon	30724	24916	18.9%
Barbara	29093	24257	16.6%
Boat	11384	8374	26.4%
Fruits	6673	5016	24.8%
Girl	7184	5586	22.2%
Goldhill	19208	14570	24.1%
Lenna	9933	7369	25.8%
Peppers	9996	7917	20.8%

TABLE 10.3. Bit-rate values for the images given in Table 10.2. The “ordinary” and “masking (br)” images were processed to have the same bit-rate. The “masking (ep)” images were processed to the same stage of zero-tree coding as the “ordinary” images.

is significantly superior to the ordinary version, yet subjective testing yielded a significantly different result.

The lowest of the PSNR values is consistently the “masking (ep)” column which represents the situation where masking shadow prediction is applied to coding of an image where the coding proceeds to the same zero-tree end-point as the ordinary data stream (not the same bit-rate or file-size). In this case the masking shadow data stream is of significantly smaller size than the ordinary stream - typically around 20% smaller. The lower PSNR value is a consequence of the lowered amount of information used to code the image.

Table 10.3 gives a comparison of the bit-rates of the various coding methods. The “ $\Delta$  rate” column gives the percentage decrease in bit-rate of the masking shadow processed images over their ordinary zero-tree coded counterparts. As can be seen, the bit-rate savings when processed to the same zero-tree end-point

are typically around 20% - quite a significant saving. The corresponding decrease in image quality is around 0.6 dB PSNR.

**10.3.3. Summary.** Examining the image results shown in Figures 10.3.1 - 10.3.9 we can see that the images that use masking shadow prediction are in general of significantly higher quality. Of particular note are the Girl and Baboon images. In the Girl image, the reconstructed image quality using masking shadow prediction is quite obviously better with significantly more noticeable features visible which are completely blurred out in the version that does not use masking shadow prediction. A similar situation exists in the case of Baboon, where there are significant facial features which are visible in the masking shadow version but not in the version that does not use masking shadow prediction.

If we consider the comparison between reconstructed images processed to the same Zero-tree end-point, we find that the masking shadow versions are reconstructed with a considerably lower bit-rate (18-28% lower) when compared to the ordinary process to the same zero-tree end-point. This reduction of information occurs in the areas that are specifically targeted by the masking shadow prediction algorithm so there is a slight reduction in the image quality when compared to ordinary zero-tree processing however the reduction in quality is not commensurate with the reduction in bit-rate compared to ordinary processing. Tables 10.2 and 10.3 show PSNR values and associated bit-rates.

A more useful comparison when evaluating the perceived image quality gains from a coding process is to keep the bit-rate constant for all processing methods. Because masking shadow prediction allocates less information cost to areas which are predicted to be perceptually unimportant it also frees up that information for areas that are important. When processing to the same bit-rate the coder then has the ability to allocate extra information to perceptually important areas which would normally have to compete with insignificant details.

The result is that the most significant features are coded with a greater precision in the masking shadow version than the ordinary version, but insignificant features are sometimes lost completely. This is borne out in the results.

The "Peppers" image is a good one to analyze. In this image the masking version shows the details of the texture of the front pepper with significantly better precision than the ordinary version - in fact in the ordinary version the texture is blurred to such a degree that it does not exist. The reason the masking shadow version is better is that less information has been allocated to fine details in preceding zero-tree passes and so there is an information budget available to code the wavelet coefficients for those features (front pepper texture) to a higher level of precision than in the ordinary case.

### 10.4. Conclusion

In this chapter we have presented results which clearly demonstrate the usefulness of masking shadow prediction as a compliment to zero-tree coding. We have shown that for equal bit-rates, images processed using masking shadow prediction have a better reconstructed subjective image quality than those processed through the ordinary zero-tree coding process. This was demonstrated by the results of subjective viewing tests carried out under a wide range of normal viewing conditions. All images processed but one showed noticeable improvement in perceived quality when processed using masking shadow prediction.

In addition to this, the use of masking shadow prediction also resulted in higher PSNR values than ordinary zero-tree coding.

The images processed cover a wide range of content and composition showing that the masking shadow prediction method is a powerful tool which is applicable to general image compression.

## Conclusions and Future Work

This Thesis accomplishes all its aims as outlined in Section 1.8. It represents a significant contribution to the field of image coding in general and to wavelets in a nonseparable context in particular. Results have been presented throughout this Thesis which demonstrate the effectiveness of the new coding methods introduced. In addition the Thesis represents an effective learning tool which lowers the barrier of entry for new researchers into the field of nonseparable wavelet coding.

The results presented in this Thesis constitute significant evidence that the quincunx lattice in particular and nonseparable lattices in general can produce viable coding frameworks for high quality image compression, while the Thesis itself represents a distillation of the essential knowledge needed by a new researchers in the field. All these accomplishments have significantly furthered the body of knowledge and advanced the general state of research in the field.

### 11.1. Summary of Contributions

We briefly re-cap here the major contributions which this Thesis has made to the field which will emphasize the success of the Thesis in achieving its aims.

We have successfully shown how wavelet transforms on nonseparable resampling lattices can be used as a basis for effective image compression methods. In general we showed that the current state of wavelet filter design in the quincunx domain is not of the standard that exists in the separable domain. This is to be expected as all filters used were designed merely to demonstrate that wavelet transforms can exist on nonseparable sampling lattices. The results indicate that there are advantages to the separable decomposition over the quincunx decomposition - more evidence of the viability of nonseparable platforms. Results of uniform scalar quantization indicate that the quincunx decomposition can perform at the same level as the separable decomposition but this is not the general case and is dependent on the signal data being processed.

**11.1.1. Comparisons.** First comparisons between separable and quincunx wavelet decompositions were presented showing those deficiencies in current nonseparable coding theory which have significant negative impact on its widespread adoption. This provides a valuable direction for future researchers in the field as contributions in these areas would have immediate practical benefits.

Comparisons of statistical properties of transforms were presented which compare the ability of the two decompositions to decorrelate image data. It was found



that the separable decomposition with  $D_4$  wavelet decorrelates data slightly better than the quincunx decomposition with a KV5/3 wavelet.

In terms of reconstruction after quantization it was shown that the performance of the quincunx decomposition under uniform scalar quantization was comparable to that of the separable decomposition.

Coding artifacts from separable and quincunx decompositions were contrasted. Of particular interest was the “dotting” effect noted for the diamond shape KV5/3 filters when only approximation data remains. These artifacts indicate certain deficiencies in the KV5/3 decomposition which are not a problem in the separable  $D_4$  decomposition, namely the large relative magnitude of the central tap in the quincunx filters.

Results were presented for an orthonormal quincunx wavelet which is the complement of the Daubechies  $D_4$  wavelet. These showed that the wavelet produced better PSNR results than the linear-phase versions but with a lower perceptual quality due to the irregular shape of the filters.

HVS weighted scalar quantization results were presented which showed that the quincunx wavelet transform is an effective image compression method.

**11.1.2. Upsampled Processing.** The upsampled representation of signals resampled on a lattice is an original contribution that was shown to be an essential tool for practical applications using nonseparable wavelets. This method allows for the computationally expensive resampling process to be avoided when dealing with nonseparable wavelet transforms and filters. A simple extension allows for an entire decomposition to be stored and extracted from the memory originally occupied by image data, which has application in embedded systems where memory constraints are important.

The upsampled method allows nonseparable signal processing to be used in real-time applications, which would not be feasible without it due to the time consumed by resampling. Although presented in the quincunx context upsampled processing can be performed using any sampling lattice compatible with wavelet transforms making it a valuable tool for other work in the nonseparable domain. It also provides a useful visualization tool for researchers and developers, allowing detail signal components to be visualized in the context of the original signal.

**11.1.3. Zero-tree Coding in Quincunx Domain.** The first application of zero-tree coding using a nonseparable resampling mechanism was presented and shown to be a very efficient coding method, as with the separable case. This is an important contribution as it exposes the nonseparable field to the most successful family of wavelet coders, an essential ingredient for forming successful coders with nonseparable filtering.

Factors affecting the choice of parent-child relationships on the quincunx lattice were discussed leading to two different implementations of zero-tree coding for the quincunx lattice. Coding results using the quincunx zero-tree method were

presented showing the effectiveness of the compression method especially when combined with Adaptive Arithmetic Coding.

**11.1.4. Masking Shadow Prediction.** The masking shadow prediction coding method represents the single most important contribution made by this thesis. It provides a mechanism for dramatically increasing perceived image quality without any corresponding increase in bit-rate.

A theory was presented for predicting the presence of perceptually unimportant areas of an image from the earliest information available from zero-tree coding. The exploitation of these predicted masking shadows was shown through coding results to reduce the bit-rate of Arithmetic Coded, zero-tree coded images by a further 20% without significantly affecting the perceived quality of the reconstructed image. In a constant bit-rate context, the images processed with masking shadow prediction showed noticeable increases in the perceived quality and PSNR values.

As with other methods presented in this Thesis, this method was designed and implemented on the quincunx domain but is equally applicable to the separable domain (and others) where similar coding results are expected. This makes it a particularly powerful tool for image compression researchers.

**11.1.5. General Contributions.** This Thesis represents a solid introduction to the theory and practical considerations of using nonseparable wavelet transforms for image compression - particularly when used in conjunction with psychovisual models of the Human Visual System. Much of the work presented is in the quincunx domain but is equally applicable to other resampling lattices such as hexagonal and separable. The psychovisual properties of the hexagonal lattice make it an attractive prospect for future research into perceptually based image compression techniques. The tools presented in this Thesis form a foundation on which to build such work.

The distilled knowledge presented in the theory and review sections of the Thesis as well as the original contributions presented represent a useful and comprehensive learning tool for new researchers interested in the field. It is hoped that this will lead to an increase in the rate of uptake of nonseparable technology by researchers and developers.

## 11.2. Further Research

This Thesis has achieved its aims of providing a strong foundation on which further research into nonseparable wavelet image compression techniques can be built. The results obtained indicate particular areas worthy of further study which are beyond the scope and time constraints of this Thesis.

**Research into the effects of orientation-based child-parent relationships in zero-tree coding.** It was found that there was little advantage to

using an orientation-based child-parent relationship over direct child-parent relationships in the quincunx case. This is an unexpected result and further research may shed more light on the reasons for this.

**Masking shadow prediction for other lattices.** The masking shadow prediction method for quincunx zero-tree coding which was presented can also be applied to zero-tree coding of separable and other decompositions where it is expected to achieve similar coding improvements to those results presented for the quincunx case.

**Hybrid zero-tree coding.** The zero-tree coder was found to be highly efficient for very low bit-rate compression but as the precision increased the effectiveness of the coder declines. A hybrid coder using zero-tree coding for the early stages and another method for latter, more random stages would most likely provide good results for higher bit-rate/quality settings.

**Design of non-separable filters for use on the quincunx lattice.** Results presented in this Thesis indicated that significant increases in perceived reconstructed image quality could be achieved by more appropriate design of wavelet filters for the quincunx lattice. During the course of the research the author noted that the lifting scheme of Sweldens would be an excellent method by which to develop filters for this purpose.

**Upsampled Processing on Other Lattices.** The upsampled processing method presented in this Thesis is not restricted to the quincunx lattice and may provide a useful tool for other nonseparable and separable domains, either for performance enhancement of wavelet transforms or as a visualization tool for nonseparable wavelet detail signals.

### Near Zero-trees

This section proposes another novel extension to the zero-tree coding mechanism which would improve its efficiency.

Image compression is inherently a process of optimization where the quality of the reconstructed images is sacrificed for a reduction in the amount of data. One way in which zero-tree coding can be improved is by the use of “near-zero-trees”. A near-zero-tree is a subband decomposition tree in which the majority but not all of coefficients are insignificant. In some instances like this it is sensible to artificially force those significant coefficients to be insignificant so that a complete zero-tree structure can be formed. This leads to a reduction in the amount of data which is needed, but it also increases the error in the reconstructed image. In some cases however it has a net positive impact from a rate versus distortion perspective since the increase in distortion is less than would occur for the sacrifice of any one symbol.

A decision must be made as to whether it is appropriate to sacrifice the precision of certain coefficients so that the overall goal of a reduced amount of data can be furthered. There are certain coefficients which have greater visible significance than others which should not be distorted in this way.

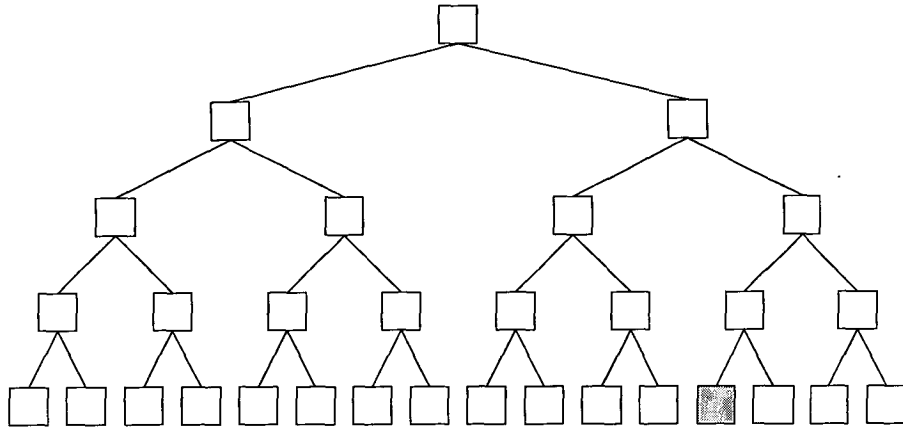


FIGURE 11.2.1. A near zero-tree. One coefficient is significant (grey), but it can be advantageous to force it to be zero as this would create a complete zero-tree represented by a single symbol in the data stream. In this case 9 symbols are required to code the tree.

The decision of which coefficients can be distorted and at what cost is at the heart of image compression algorithms. Typically compression uses immutable rules which must be obeyed, and this results in a homogeneity throughout the compression process which makes for easy analysis. However at the centre of image compression is the Human Visual System which is highly context sensitive and unlikely to notice a deviation from any homogeneous methodology.

A bad case scenario for zero-tree coding is when there is a large tree of coefficients in which all are insignificant but one, and the significant coefficient is at a fine detail level in the tree hierarchy. Figure 11.2.1 shows an example of such a situation on the quincunx tree. In this case none of the parents of the significant coefficient can be zero-tree roots because they have a significant child. The amount of data required to represent this situation is nearly maximal whereas the amount of image information expressed is very small. Since the significant coefficient is deep in the resolution hierarchy it corresponds to fine-detail in the high-frequency region of the decomposition. Thus reducing the magnitude of the coefficient would have only a small localized effect on the reconstructed image, but it could lead to a noticeable reduction in the data required.

If we consider the quincunx and separable zero-tree structures we find that the quincunx structure initially appears more conducive to having near zero trees which could be successfully changed to zero-trees. Figure 11.2.2 shows two hierarchical level steps for a separable zero-tree. There are 21 coefficients in the tree covering an area of  $4 \times 4$  coefficients at the least significant (finest resolution) level. In contrast Figure 11.2.1 shows the same situation for the quincunx tree structure.

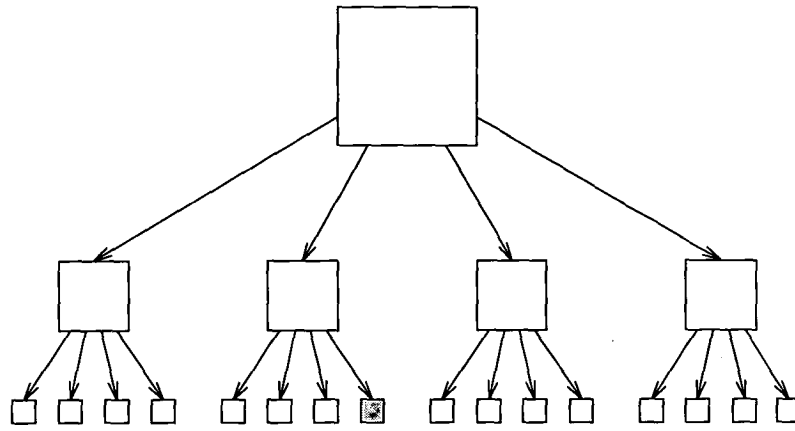


FIGURE 11.2.2. A separable near-zero-tree with a single non-zero coefficient (grey).

In this case there are 31 coefficients in the tree covering an area of 4x4 coefficients in the least significant level. It can be seen that with the quincunx tree more coefficients are required for a tree representation covering the same image area.

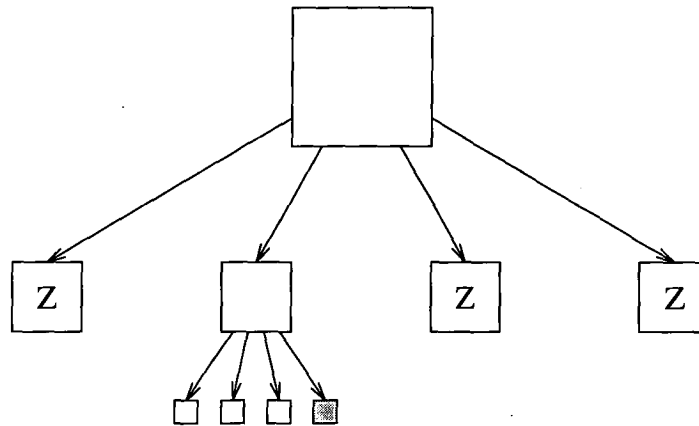


FIGURE 11.2.3. Near-zero quad-tree equivalent of Figure 11.2.2.

For both Figure 11.2.2 and Figure 11.2.1 the entire structure can be collapsed to a single zero-tree by forcing a single coefficient to be zero. In the case of the separable tree, this is a reduction of 20 coefficients, and for the quincunx case the reduction is 30 coefficients.

This is the case for full trees. For zero-trees the situation is quite different. Figures 11.2.3 and 11.2.4 show the the zero-tree structures which result from the full-trees in Figures 11.2.2 and 11.2.1. As can be seen, both trees compress to zero-tree structures of 9 necessary nodes. Both can be compressed to a single coefficient by setting the greyed insignificant nodes to insignificant. This is the case for a tree covering 4x4 nodes at the least significant level, however for a full image it is likely that a tree would cover more than twice this many decomposition levels.

Consider an example. For the separable case, 5 or 6 levels of decomposition can be expected for large images and for the equivalent quincunx decomposition



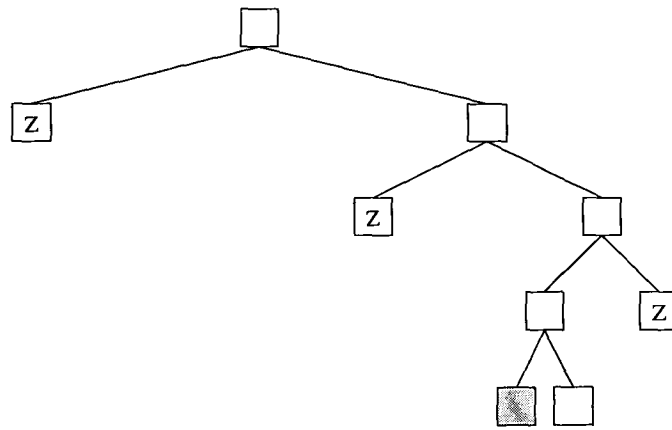


FIGURE 11.2.4. Binary zero-tree equivalent of Figure 11.2.1.

this is 10 or 12 levels. A tree of this size would span an image area of  $32 \times 32$  coefficients at the least significant level. For both separable and quincunx cases, a near zero-tree which represents this structure would contain 21 coefficients, which could be compressed to a single coefficient by forcing the single significant node to zero. This example highlights both the compression which can be achieved using the zero-tree structure and the reduction in data which can occur when collapsing near zero-trees to single nodes. Initially we have a decomposition of  $32 \times 32$  coefficients at the least significant detail level which gives a total of 1024 coefficients, which can be represented by a single zero-tree symbol (2 bits in a four-symbol alphabet dominant pass).

**Implementation Details.** A suggested implementation is presented here, although no experimental evidence is provided due to time constraints.

Near-zero-tree coding is difficult to put into a usable algorithm. The problem we face is that we must somehow transfer the knowledge to the decoder about nodes which do not obey the rules of zero-tree coding.

One way to do this builds upon the method outlined in this Thesis for use in masking shadow prediction coding. We create a means by which near-zero-trees can be predicted from lower resolution components already coded. This means that our decision about which nodes to make insignificant can no longer be based on the actual value of the nodes but must be based on the value of preceding nodes which are known.

We work with a dynamic thresholding system again as used in the masking shadow prediction coding. We increase the threshold as the number of levels of a zero tree increases. In this way the deeper a zero-tree gets, the more likely it is that a coefficient will be forced to be part of the tree - and since the depth increases as we move towards finer resolutions errors will be less pronounced. The difficult part is determining a rate of increase for the significance threshold. This factor needs to be determined before any useful method can be created from it.

When scanning for zero-trees, if we start at coarse resolutions and move down through the children we can adaptively set thresholds for these coefficients. Thresholds are unaffected until an insignificant coefficient is found. The threshold of significance values applied to its children are then increased slightly and checking proceeds into the children. If all are insignificant then the threshold applied at the next finer level is increased again. Once a coefficient is found which is significant with respect to the modified threshold the threshold bias returns to zero for its children. The process then begins again. It is possible for a zero-tree to begin at this level so if insignificant coefficients are found then their children will be treated with a biased threshold.

The key point of this method is that the exact threshold used for every coefficient can be determined at the decoding stage from those coefficients which have come before. In particular coefficients which have previously been part of a zero-tree will have had their threshold artificially increased by the encoding stage and this increase in threshold then allows us to calculate the correct dynamic range for the coefficient when reconstructed.

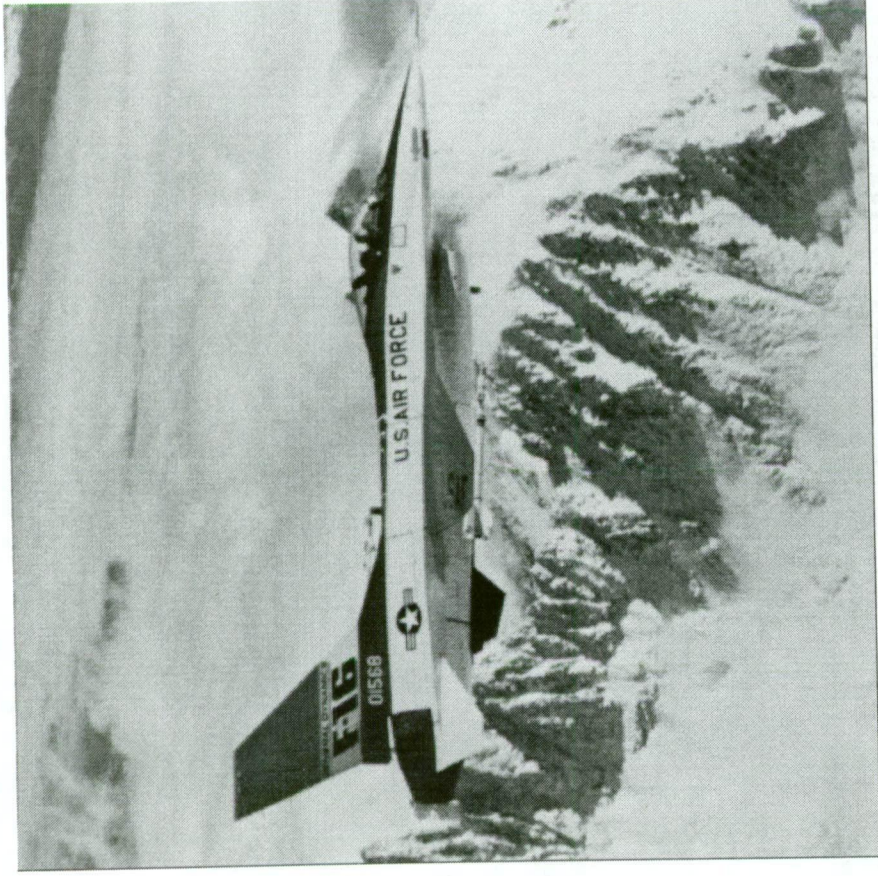
(blank)

## **Appendix A - Original Images**

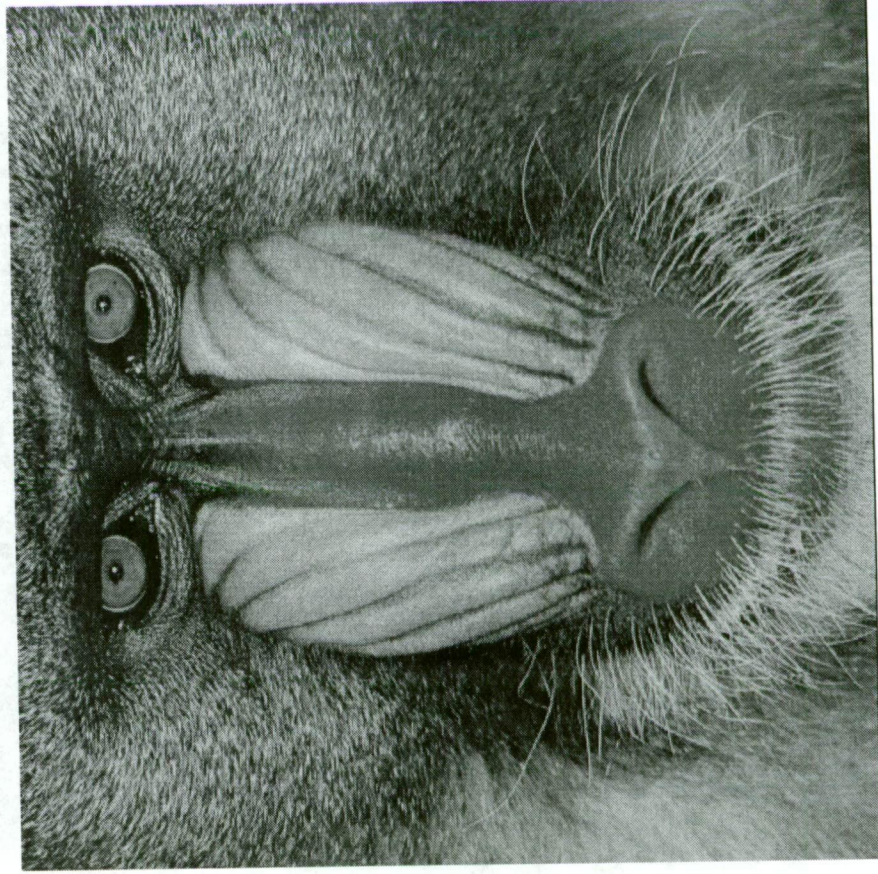
The following are the original images used in the course of the research of this Thesis. They represent a standard image set which is used in image coding globally. All images are 512x512 pixels in size at 8 bpp greyscale continuous-tone derived from conventional photographs.

The images are

- Airplane - an F-16 fighter aircraft with a mountains and clouds
- Baboon - close-up of the face of a baboon
- Barbara - indoor scene with a woman sitting on the floor
- Boat - boats on stands at a dry-dock
- Fruits - fruit-bowl with various items of fruit
- Girl - face of a girl with facepaint
- Goldhill - landscape scene of village and fields to the horizon
- Lenna - indoor scene of a woman in a feathered hat
- Peppers - still-life composition of various pepper fruit

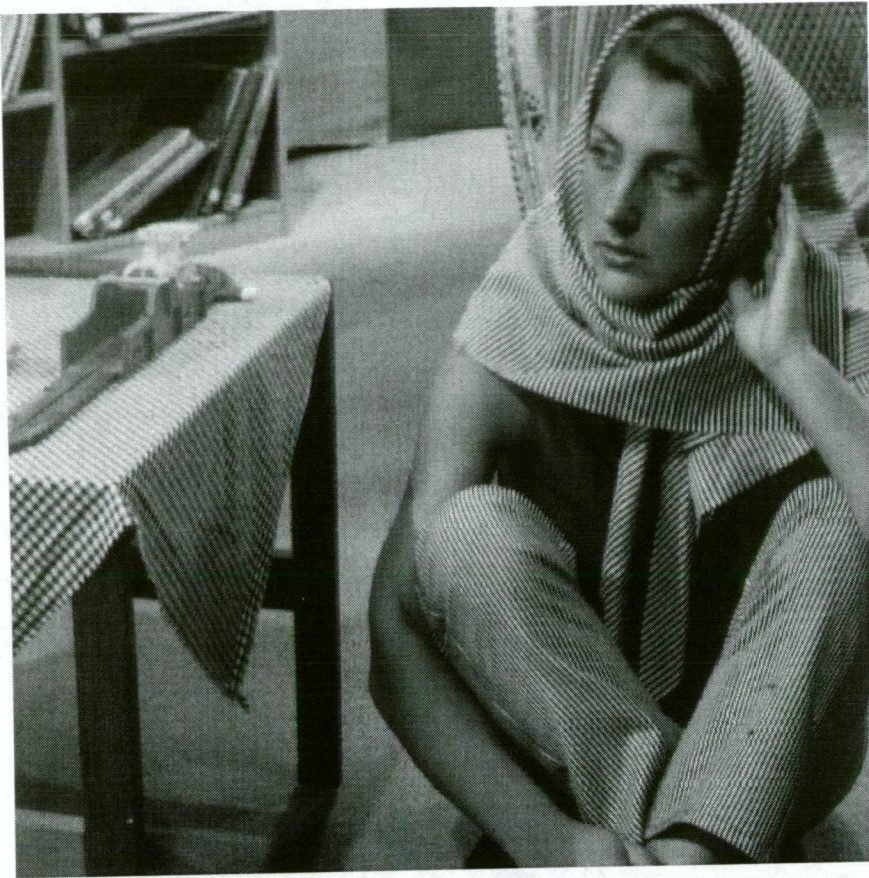


Airplane



Baboon





Barbara



Boat



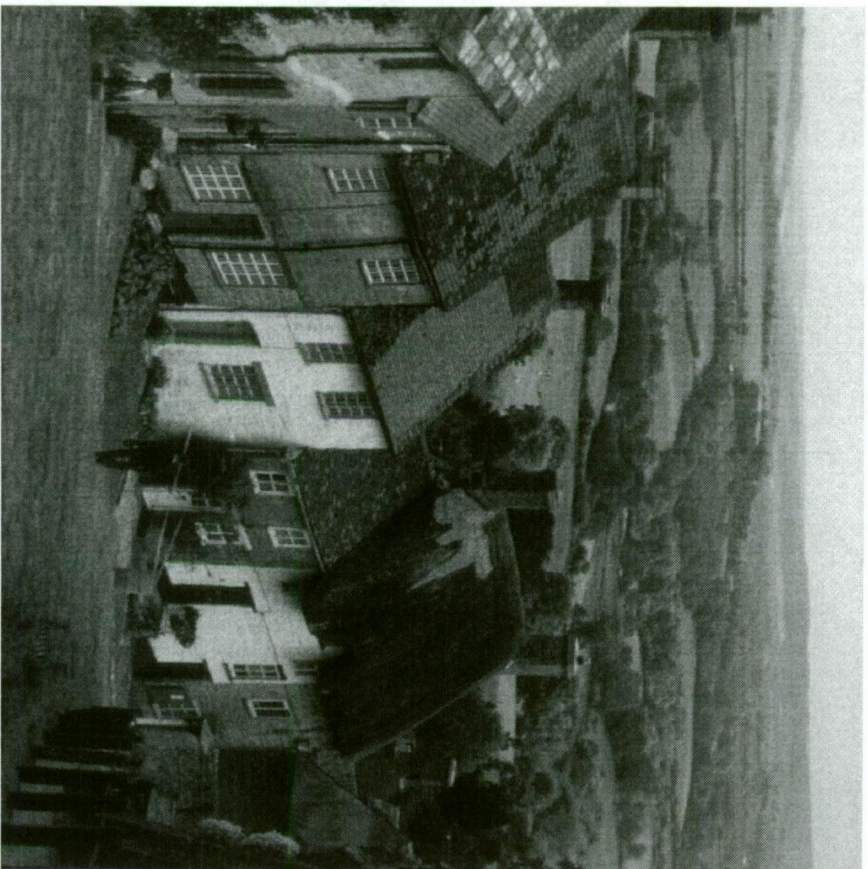


Fruits



Girl



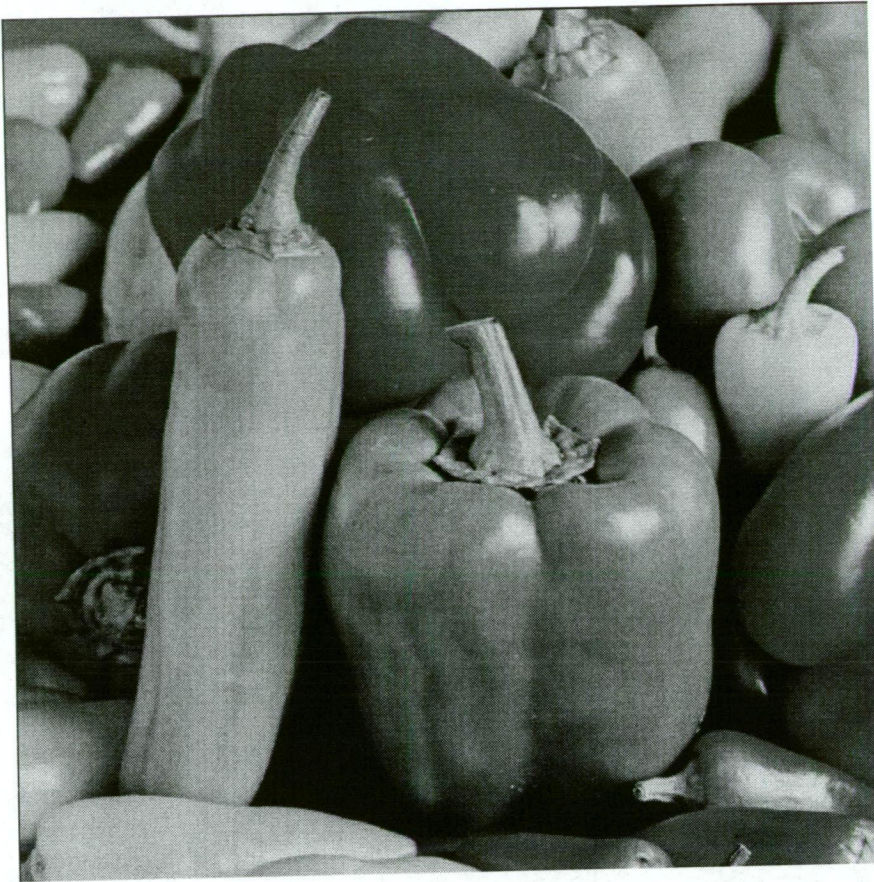


Goldhill



Lenna





Peppers

## REFERENCES

- [1] Michael Abrash. *Michael Abrash's Graphics Programming Black Book*. Coriolis, 1997.
- [2] Adelson and Simoncelli. Orthogonal pyramid transforms for image coding. *SPIE, Visual Communications and Image Processing II*, 845, 1987.
- [3] N. Ahmed, T. Natarajan, and K. R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, C-23(1):90–93, 1974.
- [4] Andrews and Nguyen. Separable and quincunx wavelet image coding. In *Proc. 6th IEEE Intl. Workshop on ISPACS*, November 1998. (Melbourne, Australia).
- [5] Andrews and Nguyen. Separable versus quincunx wavelet transforms for image compression. In *Proc. 6th IEEE Intl Conf. on Communications Systems*, November 1998. (Singapore).
- [6] Andrews and Nguyen. Multiscale location equivalence and wavelet image transforms for the quincunx lattice. In *Proc. of 5th IEEE Intl. Symp. on Sig. Proc. and its Applications*, August 1999. (Brisbane, Australia).
- [7] Andrews and Nguyen. Predicting masking shadows in zero-tree wavelet image coding. In *Proc. IEE/IEAust Intl. Conf. on Artificial Intelligence in Science and Technology*, December 2000. (Hobart, Australia).
- [8] Antonini, Barlaud, and Daubechies. Image coding using wavelet transform. *IEEE Trans. Image Proc.*, 1(2):205, April 1992.
- [9] Chen and Vaidyanathan. Multidimensional multirate filters derived from one dimensional filters. *Electronics Letters*, 27(3):225–227, January 1991.
- [10] S. Chen and D. Donoho. Basis pursuit. Technical Report. Department of Statistics, Stanford University.
- [11] A. Cohen and J. Kovačević. Wavelets: The mathematical background. *Proc. IEEE*, 84(4):514–522, April 1996. Invited paper.
- [12] J. W. Cooley, P. A. W. Lewis, and P. D. Welch. Historical notes on the fast fourier transform. *Proc. IEEE*, 55(10):1675–1678, October 1967.
- [13] J. W. Cooley and J. W. Tukey. An algorithm for the machine computation of complex fourier series. *Mathematics of Computation*, 19, 1965.
- [14] P. Cosman, K. Oehler, E. Riskin, and R. Gray. Using vector quantization for image processing. *Proc. of the IEEE*, 81(9):1326–1341, September 1993.
- [15] Pamela C. Cosman, Robert M. Gray, and Martin Vetterli. Vector quantization of image subbands: A survey. *IEEE Transactions on Image Processing*, 5(2):202–225, 1996.
- [16] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. In *J. Fourier Anal. Appl.*, volume 4. 1998. pp. 247–269.
- [17] Ingrid Daubechies. Orthonormal bases of compactly supported wavelets. In *Comm. Pure. Appl. Math.*, volume 41, pages 909–996. 1988.
- [18] Ingrid Daubechies. Ten lectures on wavelets. In *SIAM*, volume 61. 1992.
- [19] G. Davis and John Danskin. Baseline wavelet transform coder construction kit version 0.3, January 1997. <http://www.cs.dartmouth.edu/~gdavis/wavelet/wavelet.html>.



- [20] Tony D. DeRose, Michael Lounsbery, and Leena-Maija Reissell. *SIGGRAPH '95 Course Notes : Wavelets and their applications in computer graphics*, chapter Curves and Surfaces. University of British Columbia, 1995.
- [21] David L. Donoho. Interpolating wavelet transforms. Technical report, 1992.
- [22] E. Dubois. The sampling and reconstruction of time-varying imagery with applications in video systems. *Proc. IEEE*, 73(4):502–522, April 1985.
- [23] Dudgeon and Mersereau. *Multidimensional Signal Processing*. Prentice-Hall, 1984.
- [24] J.-L. Starck F. Murtagh. Multiscale image and data analysis. Technical Report. School of Computer Science, The Queen's University of Belfast.
- [25] Ove Franzen and Mark Berkley. Apparant contrast as a function of modulation depth and spatial frequency: a comparison between perceptual and electrophysiological measures. *Vision Res.*, 15:655–660, 1975.
- [26] Gaidon, Barlaud, and Mathieu. Image sequence coding using quincunx wavelet transform, motion compensation and lattice vector quantization. *SPIE, Visual Communications and Image Processing*, 1818:241–252, 1992.
- [27] A. Gersho and R. Gray. *Vector Quantization and signal compression*. Kluwer, 1992.
- [28] Raphael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley, 1982.
- [29] R. A. Gopinath and C. S. Burrus. *Wavelets: A Tutorial in Theory and Applications*, chapter Wavelet transforms and filter banks. Academic Press, San Diego, 1992.
- [30] R. M. Gray and D. L. Neuhoff. Quantization. *IEEE Trans. Inform. Theory*, 44(6), 1998.
- [31] A. Grossman, R. Kronland-Martinet, and J. Morlet. Reading and understanding continuous wavelet transform. In *Wavelet, Time-frequency Methods and Phase-Space*. Springer-Verlag, 1989.
- [32] A. Grossman and J. Morlet. Decomposition of Hardy functions into square integrable wavelets of constant shape. *SIAM Journal on Mathematical Analysis*, 15(4):723–736, 1984.
- [33] Joint Picture Experts Group. Digital compression and coding of continuous-tone still images. ISO/IEC Standard ISO 10918-1, ITU-T T.81. ISO/IEC JTC1 SC29 WG1.
- [34] Gunawan and Nguyen. Psychovisual image coding using wavelet transform. *Australian Journal of Intelligent Information Processing Systems*, 2, March 1995.
- [35] C. Herley, J. Kovačević, and M. Vetterli. *Linear Algebra for Signal Processing*, volume 69, chapter Wavelets, Filter Banks, and Arbitrary Tilings of the Time-Frequency Plane. Springer-Verlag, 1995. A. Bojanczyk and G. Cybenko, eds.
- [36] C. Herley, J. Kovačević, K. Ramchandran, and M. Vetterli. Arbitrary orthogonal tilings of the time-frequency plane. In *Proc. IEEE-SP Int. Symp. on Time-Frequency and Time-Scale Analysis*, pages 11–14, Victoria, British Columbia, Canada, October 1992.
- [37] C. Herley, J. Kovačević, K. Ramchandran, and M. Vetterli. Tilings of the time-frequency plane: Construction of arbitrary orthogonal bases and fast tiling algorithms. *IEEE Trans. Signal Proc., special issue on Wavelets and Signal Processing*, 41(12):3341–3359, December 1993.
- [38] Nikolaj Hess-Nielsen and Mladen Victor Wickerhauser. Wavelets and time-frequency analysis. *Proceedings of the IEEE*, 84(4):523–540, April 1996. Special issue on wavelet applications.
- [39] Hilton, Jawerth, and Sengupta. Compressing still and moving images with wavelets. *Multimedia Systems*, 2(3), 1993.
- [40] D. A. Huffman. A method for the construction of minimum redundancy codes. In *Proc. Institute of Electrical and Radio Engineers*, volume 40. September 1952.
- [41] LizardTech Inc. Mpsid compression engine. [www.lizardtech.com/products/mpsid](http://www.lizardtech.com/products/mpsid).

- [42] Adobe Systems Incorporated. Tagged image file format (TIFF) standard version 6.0. File format specification, 1992.
- [43] Compuserver Incorporated. Graphics interchange format (sm) standard version 89a. File format standard. Compuserve Incorporated, Columbus Ohio.
- [44] ed. James Arvo. *Graphics Gems II*. Academic Press, 1991.
- [45] Japanese Electronics Industry Development Association (JEIDA). Exif digital picture standard. Industry standard, 1998.
- [46] Joshi, Crump, and Fischer. Image subband coding using arithmetic coded trellis coded quantization. *IEEE Trans. Circ. and Sys. for Video Tech.*, 5(6):515–523, Demember 1995.
- [47] Karlsson and Vetterli. Theory of two-dimensional multirate filter banks. *IEEE Trans. Acoust., Speech, and Signal Proc.*, 38(6):925–937, June 1990.
- [48] G. Karlsson, M. Vetterli, and J. Kovačević. Nonseparable two-dimensional perfect reconstruction filter banks. In *Proc. SPIE Conf. on Vis. Commun. and Image Proc.*, pages 187–199, Cambridge, MA, November 1988.
- [49] D. H. Kelly. Spatial frequency selectivity in the retina. *Vision Res.*, 15:665–672, 1975.
- [50] Kovačević and Vetterli. FCO sampling of digital video using perfect reconstruction filter banks. *IEEE Trans. Image Proc.*, 2(1), January 1993.
- [51] J. Kovačević. Perfect reconstruction filter banks with rational sampling rates in one and two dimensions. SPIE Conf. on Vis. Commun. and Image Proc., November 1989. (Philadelphia, PA).
- [52] J. Kovačević. Perfect reconstruction filter banks with rational sampling rate changes. IEEE Int. Conf. Acoust., Speech, and Signal Proc., May 1991. (Toronto, Canada).
- [53] J. Kovačević. Wavelet families of increasing order in arbitrary dimensions. Proc. SPIE Conf. on Vis. Commun. and Image Proc., August 1997. (San Diego, CA).
- [54] J. Kovačević and W. Sweldens. Wavelet families of increasing order in arbitrary dimensions. *IEEE Trans. Image Proc.*, January 1998. Submitted.
- [55] J. Kovačević and M. Vetterli. The commutativity of up/downsampling in two dimensions. *IEEE Trans. Inform. Th.*, 37(3):695–698, May 1991.
- [56] J. Kovačević and M. Vetterli. Nonseparable multidimensional perfect reconstruction filter banks and wavelet bases for  $R^n$ . *IEEE Trans. Inform. Th., special issue on Wavelet Transforms and Multiresolution Signal Analysis*, 38(2):533–555, March 1992.
- [57] J. Kovačević and M. Vetterli. Perfect reconstruction filter banks with rational sampling factors. *IEEE Trans. Signal Proc.*, 41(6):2047–2066, June 1993.
- [58] J. Kovačević and M. Vetterli. Nonseparable two- and three-dimensional wavelets. *IEEE Trans. Signal Proc.*, 43(5):1269–1273, May 1995.
- [59] A. Kruger. Median-cut colour quantization. In *Dr. Dobbs Journal*. September 1994. number 219.
- [60] Gordon E. Legge and John M. Foley. Contrast masking in human vision. *J. Optical Society of America*, 70(12):1458–1471, December 1980.
- [61] Lu, Algazi, and Estes. A comparative study of wavelet image coders. *SPIE, Applications of Digital Image Processing XVII*, 2564, 1995.
- [62] Astrid Lundmark, Niclas Wadstromer, and Haibo Li. Recursive subdivisions of the plane yielding nearly hexagonal regions. Technical Report. Image Coding Group, Dept. Elec. Eng., Linkopings universitet.
- [63] Stephane Mallat. A theory of multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Patt. Recog. and Mach. Intell.*, 11(7), July 1989.

- [64] Man, Kossentini, and Smith. A family of efficient channel error resilient wavelet/subband image coders. In *IEEE Trans. Circ. and Sys. for Video Tech.*, volume 9, page 95, February 1999.
- [65] Mandal, Panchanathan, and Aboulnasr. Choice of wavelets for image compression. Lecture Notes in Computer Science, 1995. Dept. Electrical Engineering. University of Ottawa, Canada.
- [66] Earth Resources Mapping. Erw wavelet compression algorithm. <http://www.ermapper.com>.
- [67] David McLaren. *Efficient DCT Video Coding for Broadband ISDN*. PhD thesis, University of Tasmania., 1992.
- [68] David McLaren and D. T. Nguyen. Removal of subjective redundancy from DCT-coded images. *IEE Proceedings-I*, 138(5):345–350, October 1991.
- [69] Y. Meyer and R. Coifman. *Wavelets and Operators*. Cambridge University Press, 1993. English translation.
- [70] M. R. Nelson. LZW data compression. *Dr. Dobbs's Journal*, October 1989.
- [71] M. R. Nelson. Arithmetic coding and statistical modelling. In *Dr. Dobbs Journal*. February 1991. number 173.
- [72] Netravali and Haskell. *Digital Pictures: Representation and Compression*. Plenum Pres, 1988.
- [73] Arun N. Netravali and Birendra Prasada. Adaptive quantization of picture signals using spatial masking. *Proc. IEEE*, 65(4):536–548, April 1977.
- [74] Ngan, Leong, and Singh. Adaptive cosine transform coding in perceptual domain. *IEEE Trans. Acoust., Speech, and Signal Proc.*, 37(11), November 1989.
- [75] Moving picture experts group. Coding of moving pictures and associated audio for digital storage media at up to about 1,5 mbit/s. ISO/IEC Standard ISO 11172-1. ISO/IEC JTC1 SC29 WG11.
- [76] R. Frank Quick and Thomas A. Reichert. Spatial-frequency sensitivity in contrast detection. *Vision Res.*, 15:637–643, 1975.
- [77] K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate-distortion sense. *IEEE Trans. on Image Process.*, 2(2):160–175, 1993.
- [78] Leena-Maija Reissell. *SIGGRAPH '95 Course Notes : Wavelets and their applications in computer graphics*, chapter Multiresolution and wavelets. University of British Columbia, 1995.
- [79] Amir Said and William A. Pearlman. A new fast and efficient image codec based on set partitioning in hierarchical trees. *IEEE Transactions on Circuits and Systems for Video Technology*, 6:243–250, 1996.
- [80] Peter Schoder. *SIGGRAPH '95 Course Notes : Wavelets and their applications in computer graphics*, chapter Wavelet radiosity : wavelet methods for integral equations. University of British Columbia, 1995.
- [81] Jerome Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. Signal Proc.*, 41(12), December 1993.
- [82] Simoncelli and Adelson. Non-separable extensions to quadrature mirror filters for multiple dimensions. *Proc. IEEE*, 78(4):652–663, April 1990.
- [83] Simoncelli and Adelson. Subband image coding with hexagonal quadrature mirror filter. Picture Coding Symposium, 1990. MIT Media Laboratory, Cambridge, MA.
- [84] E P Simoncelli and E H Adelson. Subband transforms. In John W Woods, editor, *Subband Image Coding*, chapter 4, pages 143–192. Kluwer Academic Publishers, Norwell, MA, 1990.

- [85] E P Simoncelli, W T Freeman, E H Adelson, and D J Heeger. Shiftable multi-scale transforms. *IEEE Trans. Information Theory*, 38(2):587–607, March 1992. Special Issue on Wavelets.
- [86] G. Strang. Creating and comparing wavelets. *Numerical Analysis: A. R. Mitchell Anniversary Volume*, 1996.
- [87] Oyvind Stromme. *On the applicability of wavelet transforms to image and video compression*. PhD thesis, Dept. Computer Science, University of Strathclyde, 1999.
- [88] Wim Sweldens. The lifting scheme: New philosophy in biorthogonal wavelet constructions. *SPIE, Wavelet Applications in Signal and Image Processing III*, 2569, 1995.
- [89] Wim Sweldens. Wavelets and the lifting scheme: A 5 minute tour. *Zeitschrift fur Angewandte Mathematik und Mechanik*, 76 (suppl.2), 1996.
- [90] Wim Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM Journal of Math. Anal.*, 29(2), 1997.
- [91] Wim Sweldens and Peter Schroder. *SIGGRAPH '95 Course Notes : Wavelets and their applications in computer graphics*, chapter Building your own wavelets at home. University of British Columbia, 1995.
- [92] D. Taubman and A. Zakhor. Multirate 3-d subband coding of video. *IEEE Trans. Image Proc.*, 3(5), Sept 1994.
- [93] Christopher C. Taylor, Zygmunt Pizlo, Jan P. Allebach, and Charles A Bouman. Image quality assessment with a gabor pyramid model of human visual system. In *Proc. IS&T/SPIE Intl. Symposium on Electronic Imaging Science and Technology*, volume 3016, pages 493–496, February 1987.
- [94] Uytterhoeven and Bultheel. The red-black wavelet transform. Report TW 271, December 1997. Dept. Computer Science, Katholieke Universiteit Leuven, Heverlee, Belgium.
- [95] G. Uytterhoeven, D. Roose, and A. Bultheel. Wavelet transforms using lifting scheme. Technical Report ITA-Wavelets Report WP 1.1, April 1997. Department of Computer Science, Katholieke Universiteit Leuven, Belgium.
- [96] G. Uytterhoeven, F. Van Wulpen, M. Jansen, D. Roose, and A. Bultheel. WAILI: Wavelets with integer lifting. TW Report 262, July. Department of Computer Science, Katholieke Universiteit Leuven, Belgium.
- [97] P. P. Vaidyanathan. Theory and design of m-channel maximally decimated quadrature mirror filters with arbitrary m, having perfect reconstruction property. *IEEE Trans. Acoust. Speech and Sig. Proc.*, 35(2):476.
- [98] P.P. Vaidyanathan. Multirate digital filters, filter banks, polyphase networks and applications: A tutorial. *Proc. IEEE*, 78(1):56–93, January 1990.
- [99] Vetterli and Le Gall. Perfect reconstruction FIR filter banks: Some properties and factorizations. *IEEE Trans. Acoust., Speech, and Signal Proc.*, (7):1057–1071, July 1989.
- [100] M. Vetterli. Filter banks allowing perfect reconstruction. *Signal Processing*, 10(3):219–244, April 1986.
- [101] M. Vetterli, J. Kovačević, and D. J. LeGall. Perfect reconstruction filter banks for HDTV representation and coding. *Image Communication, special issue on HDTV*, 2(3):349–364, October 1990.
- [102] Martin Vetterli. Multi-dimensional sub-band coding: Some theory and algorithms. *Signal Processing*, 6:97–112, 1984.
- [103] Martin Vetterli. A theory of multirate filter banks. *IEEE Trans. Acoust., Speech, and Signal Proc.*, 35(3):356–372, March 1987.
- [104] Martin Vetterli. *Wavelets and Their Applications*, chapter Wavelets and Filter Banks for Discrete-Time Signal Processing, pages 17–66. Jones and Bartlett, Boston, 1992.

- [105] Villasenor, Belzer, and Liao. Wavelet filter evaluation for image compression. *IEEE Trans. Image Proc.*, 4(8):1053–1060, August 1995.
- [106] World Wide Web Consortium (W3C). PNG (portable network graphics) specification version 1.0. <http://w3c.org/Graphics/PNG>, 1996.
- [107] G.K. Wallace. The JPEG still picture compression standard. *Commun. of the ACM*, 34(4):30–44, April 1991.
- [108] Watson, Borthwick, and Taylor. Image quality and entropy masking. *SPIE, Proceedings*, 3016, 1997.
- [109] Watson and Solomon. A model of visual contrast gain control and pattern masking. *Journal of the Optical Society of America A*, 14, 1997.
- [110] Andrew B. Watson and Albert J. Ahumada. A hexagonal orthogonal-oriented pyramid as a model of image representation in visual cortex. *IEEE Trans. biomedical eng.*, 36(1), January 1989.
- [111] Andrew B. Watson and Albert J. Ahumada. An orthonogonal oriented quadrature hexagonal image pyramid. *IEEE Trans. Biomedical Engineering*, 36:97–106, 1989.
- [112] T. Welch. Technique for high-performance data compression. *Computer*, June 1984.
- [113] Mladen Victor Wickerhauser. Best-adapted wavelet packet bases. In Ingrid Daubechies, editor, *Different Perspectives on Wavelets*, number 47 in Proceedings of Symposia in Applied Mathematics, pages 155–171. American Mathematical Society, San Antonio, Texas, 1993. Minicourse lecture notes.
- [114] Mladen Victor Wickerhauser. *Adapted Wavelet Analysis from Theory to Software*. A. K. Peters, 1994.
- [115] Mladen Victor Wickerhauser. Custom wavelet packet image compression design. In Todor Cooklev, editor, *Proceedings of the 3rd International Workshop on Image and Signal Processing, Manchester, UK, 4–7 November 1996*, page 7, Manchester, UK, 1996. UMIST, UMIST.
- [116] Hugh R. Wilson. Quantitative prediction of line spread function measurements: implications for channel bandwidths. *Vision Res.*, 18:493–496, 1978.
- [117] Hugh R. Wilson and Stephen C. Giese. Threshold visibility of frequency gradient patterns. *Vision Res.*, 17:1177–1190, 1977.
- [118] I. H. Witten, R. M. Neal, and J. G. Cleary. Arithmetic coding for data compression. In *Communications of the ACM*. June 1987.
- [119] Woods and O’Neil. Subband coding of images. *IEEE Trans. Acoust., Speech, and Signal Proc.*, 34(5):1278–1288, October 1986.
- [120] Gunter Wyszecki and W. S. Stiles. *Color Science : Concepts and Methods, Quantitative Data and Formulae*. John Wiley and Sons, 1982.
- [121] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Trans. Info. Theory*, May 1977.