

ABSTRACT

Title of dissertation: The Multiplicative Zak Transform,
Dimension Reduction, and
Wavelet Analysis of LIDAR Data

J. Christopher Flake, Doctor of Philosophy, 2010

Dissertation directed by: Professor John J. Benedetto
Professor Wojciech Czaja
Department of Mathematics

This thesis broadly introduces several techniques within the context of time-scale analysis. The representation, compression and reconstruction of DEM and LIDAR data types is studied with directional wavelet methods and the wedgelet decomposition. The optimality of the contourlet transform, and then the wedgelet transform is evaluated with a valuable new structural similarity index. Dimension reduction for material classification is conducted with a frame-based kernel pipeline and a spectral-spatial method using wavelet packets. It is shown that these techniques can improve on baseline material classification methods while significantly reducing the amount of data. Finally, the multiplicative Zak transform is modified to allow the study and partial characterization of wavelet frames.

The Multiplicative Zak Transform,
Dimension Reduction, and
Wavelet Analysis of LIDAR Data

by

Justin Christopher Flake

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2010

Advisory Committee:

Professor John J. Benedetto, Chair/Advisor

Professor Wojciech Czaja, Co-Chair/Co-Advisor

Professor Radu Balan

Professor Manoussos Grillakis

Professor Larry Davis

© Copyright by
Justin Christopher Flake
2010

Acknowledgments

Unlike many of the self-sufficient students that have blazed trails through the University of Maryland, I am significantly indebted to a large collection of individuals, and I am partially a product of their work through me.

To my advisor, John J. Benedetto, I have the most profound of thanks. John turned much of my listlessness into production, and he crafted within me his particular brand of mathematical curiosity. Without his presence, encouragement, conversation and comical interludes, not one word of this document would have been possible. I am honored to be his student, and his friend.

I humbly thank my co-advisor, Wojciech Czaja, the mechanic of my doctoral work, for his tireless efforts to give shape to my silly ideas and to put the truly bad ones to rest. Wojtek has the amazing ability to be both supportive and demanding, with the perfect amount of charm to balance the two. No one else has been quite as interested in “how I am doing.”

I would like to extend my deepest thanks to my doctoral dissertation committee. Drs. Radu Balan, Manoussos Grillakis and Dean’s representative Larry Davis, for setting aside the time and energy to take part in this process. I appreciate your involvement, above and beyond what is required of you at the university.

I would like to thank my parents, John H. Flake and Karen Joyce Newell, for their support over the years. Their sight has not been limited to here-and-now, and their ability to envision that faraway goal embodied in this document has helped keep me on track, even when I myself could not see it. To my father, thanks for

the many discussions and talks, about the technical and the tangential. You have always been my greatest teacher. To my mother, thanks for giving me the gift of empathy and respect. You have shaped me into a kind person.

To my step-father, Lanny R. Newell, I have the thanks and gratitude of a son. I can honestly say that had Lanny not come into my life, my academic journey would have ended with an undergraduate degree. Lanny taught me the value of integrity, discipline and intelligence and helped me to understand myself better through those facets.

I would like to recognize my brothers Jordan Flake and John H. Flake III, along with my step-brothers Ryan Newell, Alec Newell and Nathaniel Newell, for helping to shape who I am and supporting me through this ordeal.

To Regina M. Conrad, I would like to express an unwavering thanks, for her understanding and support throughout the final portion of my graduate student career. Regina has experienced the beastly moods and the frequent mismanagement of priorities that I have displayed and she has powered through them still providing a beacon of hope to me. I truly can't thank her enough for seeing us through this process.

I would like to thank a few of my colleagues, Kasso Okoudjou, Nate Strawn, Avner Halevy, Chris Miller, Chris Manon, Dave Shoup, Avanti Athreya Janicki, Ryan Janicki, Matt Konicki, Dave Widemann, Kevin Duke, Matt Hirn, Chris Shaw and Joseph Woodworth, for an enormity of conversations, amusement and in general, creating an atmosphere that at times was conducive to mathematics and at other times a complete riot. These are the people that make day-to-day life bearable at

the university.

A special thanks to a few special ladies that made my lack of planning and foresight bearable. Celeste Regalado, Sharon Welton, Liz Wincek and Ronnie Brown have all had a hand in making this long graduate career finally come to a close. Thank you from the bottom of my heart for your selflessness.

Finally, I would like to thank the department of defense, the National Geospatial Intelligence Agency, the Topographic Engineering Center and the University of Maryland at College Park for their support throughout all of the projects contained within.

Table of Contents

List of Figures	vii
List of Abbreviations	x
1 Introduction	1
1.1 Original Elements of the Thesis	3
1.1.1 Chapter Two	3
1.1.2 Chapter Three	4
1.1.3 Chapter Four	4
2 Compression of LIDAR Data Types	6
2.1 Introduction	6
2.2 DEM and LIDAR Data Types	11
2.2.1 Urban DEM, Quality and the Expert	16
2.2.2 Terrain Structural Similarity Index (TSSIM)	18
2.3 Time-Scale Compression Methods	22
2.4 Geometric Representation: Wedgelets	30
2.4.1 Wedgelets	31
2.4.2 Piece-Wise Constant Wedgelets	33
2.4.3 The Wedgelet Transform	42
2.4.4 Pruning the Quad-Tree	46
2.4.5 Implementation and a Discrete Green's Theorem	50
2.4.6 Linear Wedgelets	55
2.4.7 LIDAR Wedgelets	59
2.5 Experiments and Performance	61
3 Dimension Reduction and Material Classification	74
3.1 Introduction	74
3.2 Hyperspectral Data	80
3.3 Kernel-Based Frame DR Methods: The Pipeline	84
3.3.1 The Laplacian-Eigenmaps Kernel	88
3.3.2 Locally Linear Embedding	89
3.3.3 Dimension Reduction by Change of Coordinates	91
3.3.4 Landmarking	92
3.3.5 Endmember Selection	96
3.3.5.1 Frame Coefficients	96
3.3.6 Frame Based Kernel Method Results	97
3.4 Wavelet Packets and Spectral/Spatial Representation	100
3.4.1 Wavelet Packets	104
3.4.2 Best Basis Algorithm	107
3.4.3 Inverting the Wavelet Packet Transform	110
3.4.4 Combining LLE and Wavelet Packets	111
3.4.5 Numerical Results	114

4	The Multiplicative Zak Transform	119
4.1	Introduction	119
4.2	Notation	121
4.3	Gabor systems and the Zak transform	121
4.3.1	Properties of the Zak transform	122
4.4	The Multiplicative Zak transform	126
4.4.1	Superframes and Extension	135
4.4.2	The Multiplicative Zak Transform on Wavelet Sets	140
4.4.3	Inversion Formula	150
	Bibliography	154

List of Figures

2.1	An example of raw LIDAR.	7
2.2	Two canonical examples of DEM.	7
2.3	Tilted urban DEM displaying surface features.	8
2.4	Can you tell if there is an extreme edge on this roof?	9
2.5	New Orleans S-tile and the sub-tiles S_1 through S_{15}	12
2.6	The standard urban tile, S_{10}	13
2.7	Alexandria Virginia T-tile and the sub-tiles T_1 through T_{25}	14
2.8	This Map Shows the Density of the Point Cloud Samples.	15
2.9	Standard Gridding Techniques on the LIDAR Contained in T_{26}	16
2.10	Standard gridding techniques on the LIDAR contained in T_{27}	17
2.11	Simple Example of a Cartoon Model.	23
2.12	Modeling an Edge in a Cartoon Model: Do and Vetterli.	24
2.13	Frequency partition for Contourlets.	24
2.14	S-tiles Avg.: $\log(1 + _2)$	27
2.15	S-tiles Avg.: $\log(1 + _\infty)$	27
2.16	S-tiles Avg.: total variation	27
2.17	S-tiles Avg.: TSSIM	27
2.18	T-tiles Avg.: $\log(1 + _2)$	27
2.19	T-tiles Avg.: $\log(1 + _\infty)$	27
2.20	T-tiles Avg.: total variation	28
2.21	T-tiles Avg.: TSSIM	28
2.22	S_{10} : $\log(1 + _2)$	28
2.23	S_{10} : $\log(1 + _\infty)$	28
2.24	S_{10} : total variation	28
2.25	S_{10} : TSSIM	28
2.26	S_{10}	30
2.27	Contourlets defects.	30
2.28	Aligned edges, polygonal edges and flat surfaces.	31
2.29	Examples of Wedgelet Functions.	32
2.30	The Basic Wedgelet Set-Up.	33
2.31	First Layer Dyadic Decomposition.	33
2.32	Illustration of ordering in Layer 2.	34
2.33	An Edgelet on the sub-square S_l^j	34
2.34	An Edgelet on the sub-square S_l^j and its global coordinates.	36
2.35	Examples of steep and flat digital lines.	37
2.36	The four angular regions.	38
2.37	Correct Flat Increment.	38
2.38	Incorrect Flat Increment.	38
2.39	Correct Step Increment.	39
2.40	Incorrect Step Increment.	39
2.41	R1 Increment Values.	39
2.42	R2 Increment Values.	40

2.43	R3 Increment Values.	40
2.44	R4 Increment Values.	40
2.45	R1 Wedgelet.	41
2.46	R2 Wedgelet.	41
2.47	R3 Wedgelet.	41
2.48	R4 Wedgelet.	41
2.49	Level 0 Wedge Partitions.	44
2.50	Level 1 Wedge Partitions.	44
2.51	Level 2 Wedge Partitions.	45
2.52	Level 3 Wedge Partitions.	45
2.53	Level 4 Wedge Partitions.	45
2.54	Level 5 Wedge Partitions.	45
2.55	Example of a Valid Partition.	46
2.56	Local Pruning Example.	47
2.57	Two Consecutive Wedgelets.	50
2.58	Expression Using $W_l^j(\theta, k)$ to Find $W_l^j(\theta, k + 1)$	51
2.59	An Unknown Wedgelet.	54
2.60	Step 2.	54
2.61	Step 3.	54
2.62	Step 4.	55
2.63	Step 5.	55
2.64	Step 6.	55
2.65	Step 7.	55
2.66	The Value of the Wedgelet is Calculated.	56
2.67	A Typical Urban Roof.	57
2.68	Urban Roof Approximation with Wedgelets.	57
2.69	Urban Roof Approximation with Linear Wedgelets.	59
2.70	S_{10} with $\lambda = 100$	62
2.71	S_{10} with $\lambda = 50$	62
2.72	S_{10} with $\lambda = 10$	62
2.73	S_{10} with $\lambda = 1$	63
2.74	S_9 λ vs. Coeffs. vs. TSSIM.	64
2.75	Flat Wedgelets vs. Linear Wedgelets.	65
2.76	S_{10} Varied Angles and Norms.	66
2.77	S_9 Varied Angles and Norms.	66
2.78	Contourlets vs. Wedgelets.	67
2.79	S_{10} PW Constant Wedgelets 4/40 Angles.	68
2.80	S_{10} Contourlets Level 1/2.	68
2.81	S_{10} Linear Wedgelets 4/40 Angles.	69
2.82	S_{10} Linear Wedgelets L1 Norm 4/40 Angles.	69
2.83	S_{10} Linear Wedgelets L1 Norm 40 Angles 15% Retained Coefficients.	69
2.84	S_{10} Original.	70
2.85	S_9 Extreme Angular Resolution Test.	70
2.86	S_{10} Extreme Angular Resolution Test.	70
2.87	S_{10} Angular Variation and Boundary TV-norm.	71

2.88	S_{10} Boundary TV-norm.	72
2.89	S_{10} ℓ^2 -norm.	72
2.90	S_{10} Contourlets vs. Wedgelets.	72
2.91	S_9 Contourlets vs. Wedgelets.	73
3.1	Article Data Projected onto \mathbb{R}^3 : Maggioni.	76
3.2	Hand-Written Numeral Data Projected onto \mathbb{R}^3 : Maggioni.	77
3.3	Frame-Based Kernel Dimension Reduction Pipeline.	78
3.4	The Prototypical Example of a Hypercube.	81
3.5	True Color Image of the Urban Data Set.	82
3.6	True Color Image of Smith Island.	82
3.7	Statistical Spectral Envelopes.	83
3.8	Urban Tile Spectral Band.	84
3.9	Urban Tile Spectral Band.	84
3.10	Typical Hyperspectral Data Cube Point $x_j \in \mathbb{R}^D$	85
3.11	Data Points $X \subset \mathbb{R}^2$	87
3.12	Manifold $X \subset M$	87
3.13	A Poorly Chosen Neighborhood.	87
3.14	A Well Chosen Neighborhood.	88
3.15	A Decent ϵ -Neighborhood.	89
3.16	A Decent k -Neighborhood.	89
3.17	Typical Landmarking Sample.	94
3.18	Dimension Reduction on the Sampled Set.	94
3.19	A Schematic of the Out-of-Sample Extension.	95
3.20	Class Map of the Urban Data Tile.	98
3.21	The Frame Flag.	99
3.22	Frame Coefficient Maps of Smith Island	101
3.23	Smith Trial Ground Truth Results.	102
3.24	Smith Competing Ground Truth Results (total only).	102
3.25	The Entropy Determines the Subtree that is Chosen for Reconstruction.	103
3.26	The Wavelet Coefficient/Filter Tree.	105
3.27	The Wavelet Packet Coefficient/Filter Tree.	106
3.28	An Example of a Wavelet Packet Coefficient Tree.	108
3.29	An Example of a Corresponding Entropy Function.	109
3.30	The Best Basis Algorithm.	110
3.31	The Shannon Entropy Function.	111
3.32	Spectral/Spatial Dimension Reduction Schematic.	112
3.33	Classification Results on Urban Data Set.	116
3.34	Selection of Classes of the Vector Angle Classification Scheme for the Urban Data Set.	117
3.35	The Combination of WPT with PCA Outperforms Simple PCA.	118
4.1	Estimate of the MZT of the MHF.	153

List of Abbreviations

DWT	Discrete Wavelet Transform
DEM	Digital Elevation Model/Matrix
LE	Laplacian Eigenmap
LIDAR	Light Detection and Ranging
LLE	Locally Linear Embedding
MSE	Mean Square Error
NGA	National Geospatial Intelligence Agency
NWC	Norbert Wiener Center for Harmonic Analysis and Applications
PCA	Principal Component Analysis
PSNR	Peak Signal to Noise Ratio
SSIM	Structural Similarity Index
SVDD	Support Vector Data Description
TEC	Topographic Engineering Center
TSSIM	Terrain Structural Similarity Index
WPT	Wavelet Packet Transform
\mathbb{R}	The set of real numbers
\mathbb{R}^d	The set of d -tuples of real numbers
\mathbb{Z}	The set of integers
\mathbb{N}	The set of natural numbers
\mathbb{C}	The set of complex numbers

Chapter 1

Introduction

Harmonic analysis is a field epic in both scope and breadth. Admitting techniques that abut on the edge of number theory and providing the vehicle for many sharp inequalities in PDEs are a couple of examples that illustrate this vast often disconnected portion of mathematics. With this being said, it is not surprising that connecting seemingly disparate subjects within the framework of harmonic analysis can be a feasible reality. This thesis represents three distinct projects within the realm of pure and applied harmonic analysis. Whether it be the representation, compression and reproduction of DEM and LIDAR data types with wavelet and wedgelet methods, dimension reduction of hyperspectral data by frame-based kernel methods or wavelet packet methods for the purposes of material classification or the analysis of wavelet systems with the multiplicative Zak transform, the clear and pertinent underlying theme is the analysis of objects by both time and scales.

DEM and LIDAR are related data types that differ from images by a great deal. Instead of representing color reflectance information about a scene, DEM represents the altitude of a surface in a raster form and LIDAR is essentially a point cloud sampling of such a surface. The goal of this portion of the thesis is to decompose these structures with an intrinsic transform, so that when compression is performed, the important underlying geometries are preserved. A natural place to

start with such a project is by parsing through standard image processing techniques that share similar goals. Taking into account the way in which the data differs from images, and improving on the techniques inherited by image processing, we can establish a couple of transform techniques which perform this process well. The resulting transforms, contourlets and wedgelets, are part of this larger view into the analysis of time and scale, and utilizing these structural components to understand these particular objects.

Hyperspectral data is a pure product of harmonic analysis. Nothing could be more classical than spectral information of this sort. While the methods for the reduction of dimension of general data sets has not always existed exclusively within the world of harmonic analysis, the reduction of dimension of this particular type of data can certainly be considered loosely part of the family. Frame based kernel methods for dimension reduction of hyperspectral data can be seen in the light of time-scale analysis through the diffusion wavelet viewpoint. The methods in this thesis do not cover this perspective, but the alternate analysis of a spatial-spectral decomposition of hyperspectral data through the use of wavelet packets clearly fulfills this condition. The end result in any case is a reduced dimension data cube for the express purpose of material classification. Both of the methods above are shown to be superior to classification of the raw data by a collection of standard techniques.

Wavelets themselves are a personification of time-scale analysis. Ever since Calderón showed the world that we could represent with scale via the affine group, we have been honing our ability to generate and utilize these wavelet systems. It is this

same scale property that makes the affine group such a confounding group to work with, so much so that replicating the actions of the Zak transform in the Heisenberg group case for Gabor systems is neither simple nor straightforward. However, an idea in the form of the multiplicative Zak transform yields a tantalizing clue to the reproduction of this process, and allows us the ability to work on familiar ground with the troublesome affine group.

1.1 Original Elements of the Thesis

1.1.1 Chapter Two

Chapter two concerns the evaluation of many standard and modern multi-scale wavelet-like representation schemes for DEM/LIDAR data. An innovation within this framework was the removal of luminance from the calculation of TSSIM, to more aptly describe the DEM data type.

Another innovation within this chapter concerns the structures used to implement the modified wedgelet algorithms. These were sped up many times from their basic Beamlab levels merely by using efficient structures tailor built for the process. The recursive decomposition scheme was created to minimize the number of redundant calculations.

The largest contribution in this chapter consists of extending the wedgelet transform to working on raw LIDAR data. This generalization of the standard linear wedgelet transform works on gridded information and non-uniform data points. This effectively allows for a multitude of new data types as well as providing a flexible

utility for gridding LIDAR data, and removing gaussian and salt-and-pepper noise, when that noise occurs on large structural elements, all at the same time.

All of these methods are novel when used on DEM data, as many people have not utilized the fact that DEM and image data behave very distinctly from one another.

1.1.2 Chapter Three

This chapter is interesting for a number of reasons. The pipeline used to implement frame based kernel methods for dimension reduction for the purpose of material classification are all elements that have been used apart from one another for a very long time. There are two shared contributions within this context. The first is that by ordering the flow of the pipeline in a particular directions, we have in effect produced a new kernel for dimension reduction.

The second novel process is the application of frames to replace the aging concept of endmembers. The conceptual difference allows for a great deal of flexibility when it comes time to classify the data, as frames are essentially designable.

1.1.3 Chapter Four

The multiplicative Zak transform has been around now for over a decade. However, it hasn't really progressed from the original result found by Gertner et al. [35]. Part of the problem with wavelets is that they aren't much of a problem at all. Wavelets tend to want to span $L^2(\mathbb{R})$, much more often than an average

function g could create a spanning Gabor system.

One addition to this outlook is the introduction of superframes as an attempt to glue the two Hardy spaces back together so that actual wavelet sets may be analyzed. Unfortunately this result has only been shown to be true in one direction, namely, that of all wavelet frames consisting of a two-part wavelet-superframe. This result is proven in Theorem **7**.

The idea of analyzing the multiplicative Zak transform on wavelet sets is also a new one. The important structure in use is the dilation-partition of the real-line by a wavelet set. This preserves all of the important properties of the transform itself, and allows us to shift our view to various sets, where functions have the splitting property with their wavelet generators. This leads to a slightly more general classification of wavelet frames, one that allows support across zero in $\hat{\mathbb{R}}$, allowing for genuine wavelet frames. This occurs in Theorem **9**.

Theorem **11** illustrates a very provocative relationship between the multiplicative Zak transform, and the alternate version of that transform, in relation to the sum of all dilates of the generating function.

The final innovation of this thesis is the inversion formula for the multiplicative Zak transform given in theorem **12**. Once the other elements from the paper (such as the inverse of Theorem **7**) have been worked out, this will provide a simple mechanism to create families of wavelet frames with designed properties.

Chapter 2

Compression of LIDAR Data Types

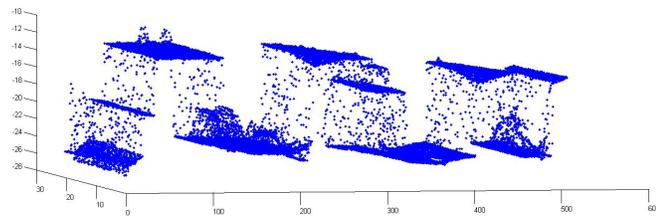
2.1 Introduction

The use of Digital Elevation Matrices (DEM) has seen a rapid expansion of activity in many diverse fields in the last decade. Among these fields are geography, geology and urban planning. Interest in using DEM as a tool spiked with hurricane Katrina, since all aspects of government and civilian intervention required extensive and accurate knowledge of surface details along the gulf coast. Given the increasing prevalence of inexpensive massive data storage, DEM is finding a strong following in fields where images used to suffice.

Digital Elevation Matrices are processed from raw Light Detection And Ranging (LIDAR) data. LIDAR is essentially a point-cloud of three-dimensional data taken by attaching a light emitter and collector to the underside of an airplane and flying the plane over the region of interest. A combination of GPS information for the plane's flight path and collected light returns are converted into a point cloud that is a rough approximation of the surface that has been sampled. Specialty technicians and software then convert this raw data into a gridded raster form that we know as DEM. This is a painstaking process, often involving human manipulation of altitude values and positional shifts based on ground truth and known artifacts. Whenever DEM is mentioned in the following pages, it is assumed to implicitly

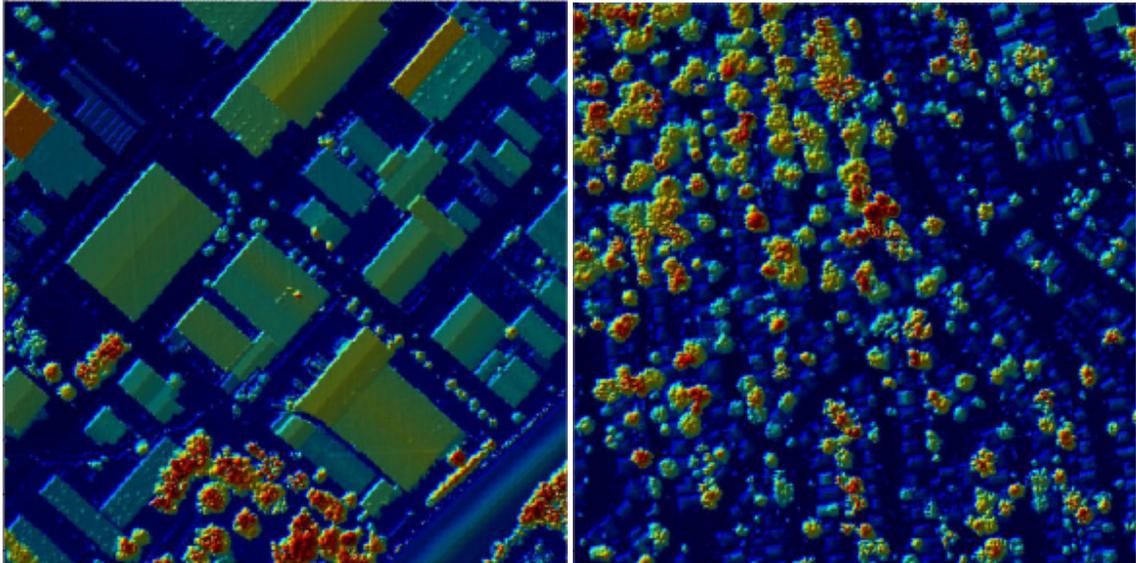
depend on the underlying LIDAR structure.

Figure 2.1: An example of raw LIDAR.



Note that in figure 2.1, the data is truly three dimensional, and non-unique height values can be explained by multiple passes, multi-path propagation or even the result of first and second pass LIDAR penetrating through different materials. The process of gridding LIDAR into DEM unfortunately removes much of the rich structure that LIDAR data sets possess. However, the DEM is a more efficient structure, containing a great deal of surface information inherited from the LIDAR.

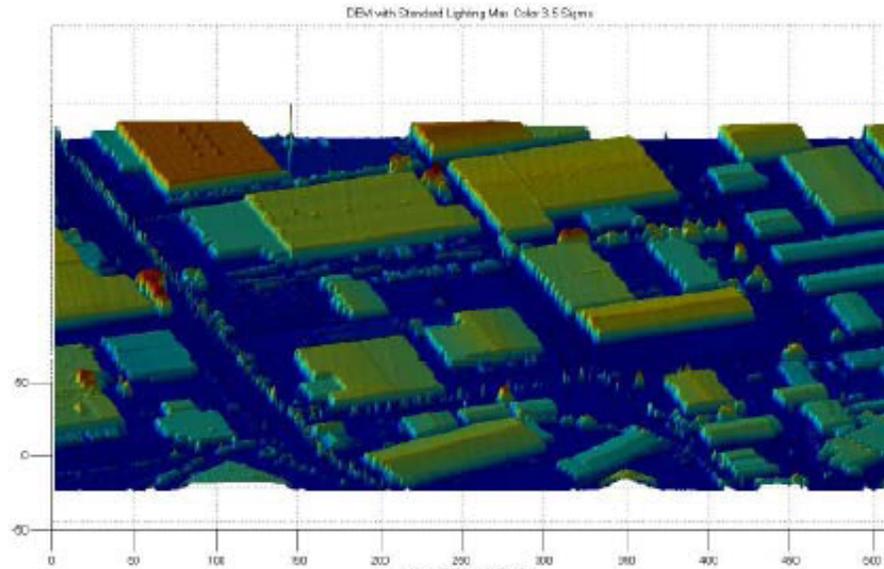
Figure 2.2: Two canonical examples of DEM.



In figure 2.2, and for the remainder of this chapter, the z -value, or altitude of the

surface at a pixel, is represented by a color between blue (low altitude) and red (high altitude).

Figure 2.3: Tilted urban DEM displaying surface features.

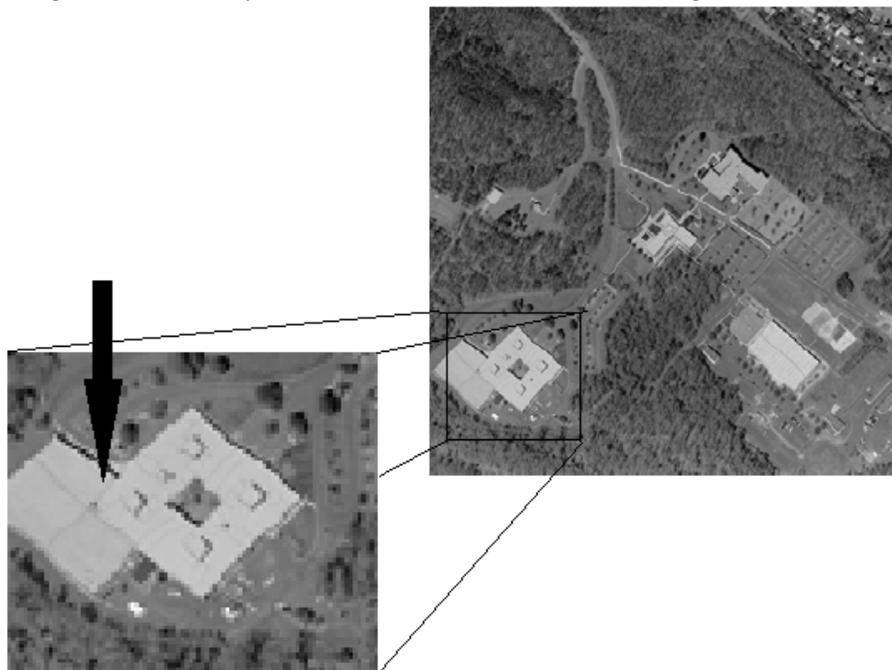


The fact that DEM is really a gridded approximation to a surface allows us to utilize the data much differently than you would an image. In figure 2.3 we can rotate the camera looking down onto the DEM from 25° , changing the light source and shading to show that the surface lives in three dimensions. This kind of information is vital for understanding line of sight for mobile phone communications in a densely populated city, or even to understand large-scale drainage paths in flood-prone areas.

Two facts separate DEM data from traditional image data in important ways. First, DEM can be an excessively large way to store three dimensional data, and confounds efficient methods of storage and transmission. It isn't that DEM is an obnoxious data structure, but that there is simply too much of it, and increasingly at higher and higher resolutions. Only a very small portion of the globe has been mapped via LIDAR/DEM, and even with that supposed dearth, there is simply too

much data to process. Some groups have been known to deposit their DEM data into large storehouses often many miles away from the analysis center. In these cases, in order for data to be processed, first they must requisition the data, and then have it delivered. A very time consuming and tedious prospect. Secondly, DEM data is very distinct from image data in that many techniques that derive from image processing fail to work correctly or are inefficient when used with DEM data, treated as if it were a black and white image. One very simple way to visualize this principle is to notice that in an overhead image of a building and a parking lot, if the two are of the same color and there is no shadow, it may appear that there is in fact no edge separating the two entities. In this case, DEM would very clearly show that an edge exists and how extreme it is.

Figure 2.4: Can you tell if there is an extreme edge on this roof?



One solution to the problem of large data sizes for DEM is through some

method of compression. This is where the second fact emerges to confound the first: traditional image processing compression techniques do not preserve the important geometries in DEM that make them useful to an analyst. In a heavily urban DEM, and this will be the primary focus of this chapter, the data is essentially composed of many flat areas hemmed in by heavy edge detail. Many traditional wavelet compression techniques compromise the edges and add Gibbs-like noise where there was none. For many technicians that work closely with DEM, this is an unacceptable loss of integrity. The edges are what makes the DEM special!

A straightforward approach to the solution of these problems is by using modified wavelet-like systems that are amenable to edges and directional details. Specifically, curvelets, contourlets and ridgelets [23] [24] [25] [13] are relatively modern multi-scale methods that emphasize intelligent tiling of the frequency domain and are thus directionally oriented wavelet-like systems that can be applied to the representation and compression of DEM data. The first portion of this chapter will detail experiments that conclude the superiority of contourlets over other standard and modified methods in representing urban DEM. The idea of what it means for one method to be superior over another will also be discussed since we have established that DEM and image data are distinct. Given this, it will be useful to evaluate our quantitative methods of establishing how closely a represented/compressed DEM resembles the original by going through various quality indices and metrics that attempt to accomplish this goal. We eventually settle on the Terrain Structural Similarity index (TSSIM), a modified version of the structural similarity index (SSIM) [51] and use this as a quality comparison between any two DEM tiles.

Since geometry plays a vital role in the underlying structure of urban DEM, it makes sense that techniques based on scale-space geometric representation may perform better than multi-scale methods. The second section of this chapter deals with studying and modifying wedgelets [26], a geometrical representation method, to work specifically with urban DEM. Wedgelets end up vastly outperforming contourlets in the representation, compression and quality reconstruction of heavily urban DEM data tiles. We will introduce several modifications to the basic wedgelet scheme that enhance the ability of the transform to optimally decompose urban structures in DEM. These increase the potency of the compression while maintaining strong reconstruction in terms of TSSIM.

2.2 DEM and LIDAR Data Types

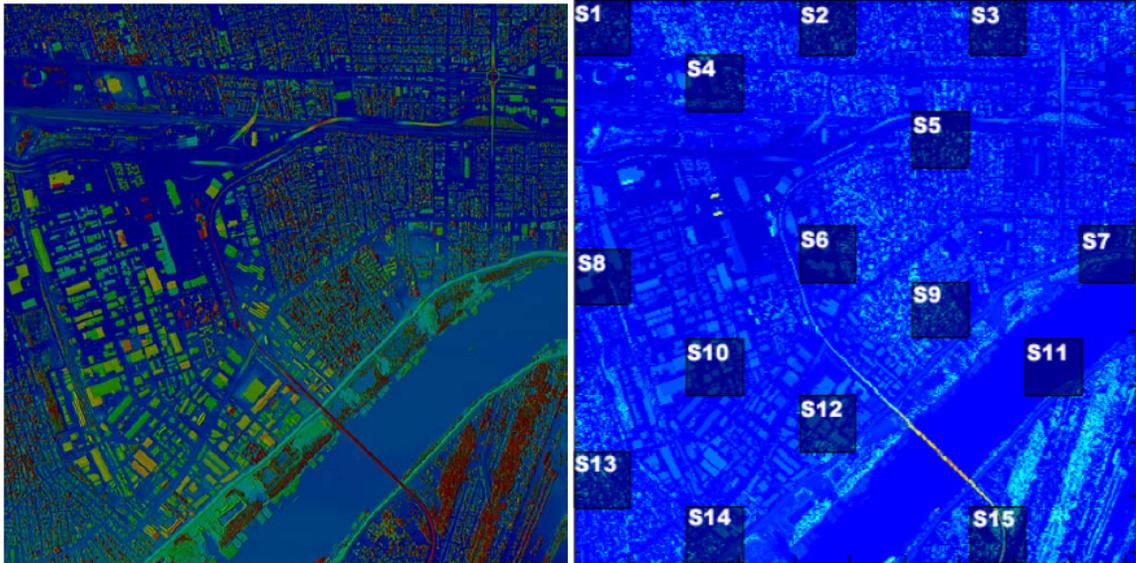
Digital Elevation Models/Matrices are real-valued matrices for which the rows and columns represent some geographic latitude and longitude and the value of the pixel corresponds to an approximation of the surface altitude at that position. Most users of this data type either pay to have raw LIDAR data converted to its more usable form, or have in-house processes and technicians that do the same. Either way, raw LIDAR point-clouds are the underlying data type that give DEM their important structure.

DEM has a few very distinct uses. Some very coarse forms (where the grid points are separated by 50 meters or more) are used to map vast regions, mountainous and sub-sea level alike. These can be used to predict ice/water flow and

collection since they contain information about the gradient of elevation. Another use for such regions is the analysis of straight line paths connecting two entities on separate parts of the surface. This can be helpful in planning systems of communication that rely on the propagation of EM signals. Finer versions of DEM (where the grid points are separated by 1 meter or less) are very useful for analyzing various aspects of urban importance. Cell-phone tower location, line-of-sight analysis and noise reduction are a few such uses of the fine-scale urban DEM.

This research project dealt with a few specific data sets that will arise again and again throughout this chapter. We have what is known as the S data set, a large portion of New Orleans (note the colors with respect to sea level), which we have carved into a number of manageable 512×512 sized data tiles. We refer to these as S_1 through S_{15} , with the standard urban example being S_{10} .

Figure 2.5: New Orleans S-tile and the sub-tiles S_1 through S_{15} .



The second important data set that is used throughout this chapter is the

so-called T -tile. This is a large portion of Alexandria Virginia, with Telegraph road running right through the middle. This tile happened to contain the headquarters of the Topographic Engineering Center (TEC). This is the organization that financed this research project. Clearly an important tile. Similar to S , this is a very large high resolution DEM, that we broke up into a number of manageable 512×512 sub-tiles. We also refer to these as T_1 through T_{25} .

The T tile has a lot going for it aside from its artful color scheme. This represents an area roughly six thousand meters square, with 1 meter resolution. In the left most image in figure 2.7, there is a large shaded region that actually sits above the TEC headquarters. This is one of the only regions for which we had both LIDAR and DEM data representing the same geographic region. The shaded region

Figure 2.6: The standard urban tile, S_{10} .

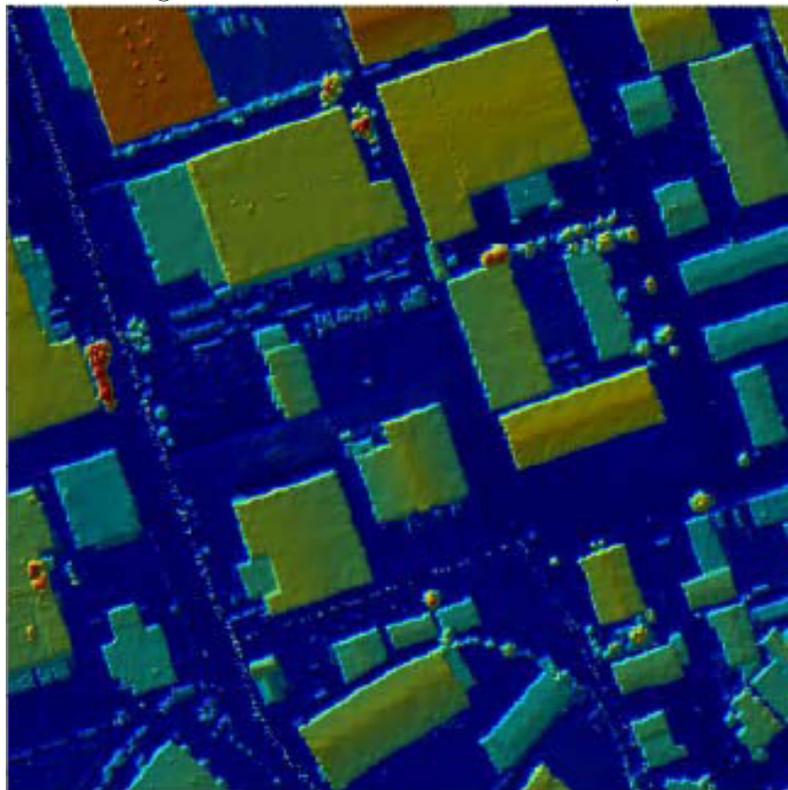
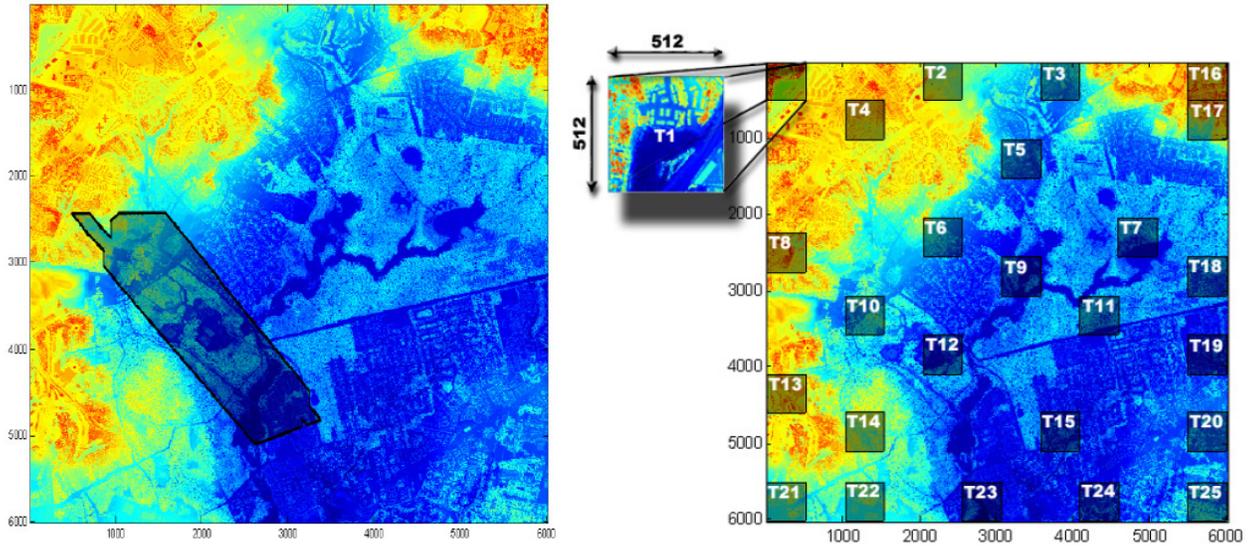


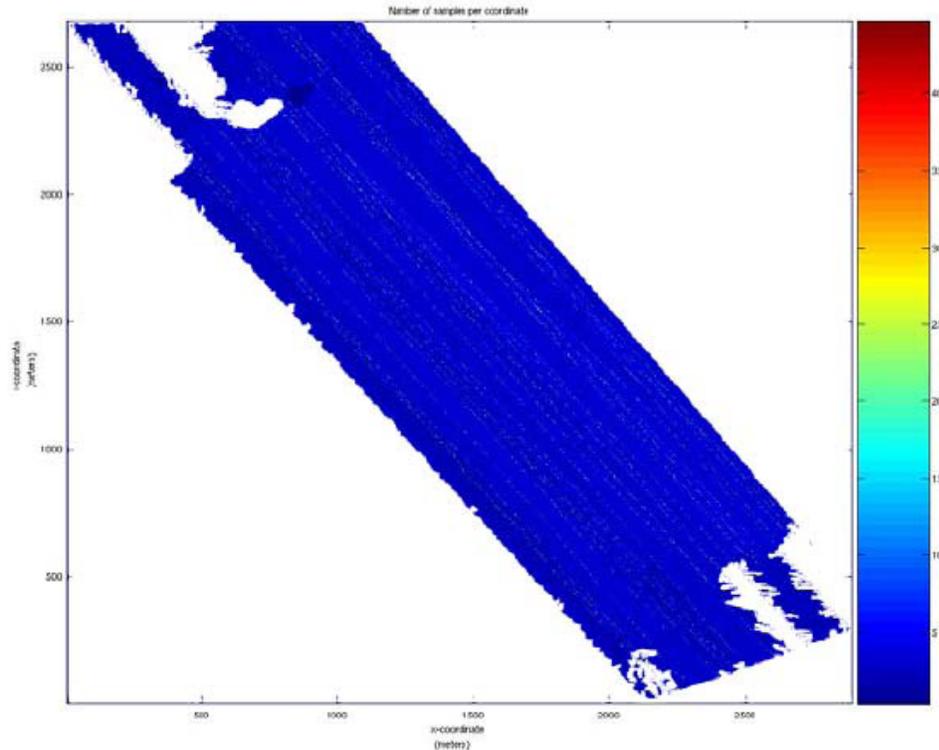
Figure 2.7: Alexandria Virginia T-tile and the sub-tiles T_1 through T_{25} .



represents the boundary of the LIDAR data that we received, and this fortunate concurrence allowed us to try and understand the gridding process in more depth. The amount data contained within the point cloud is very large, containing roughly 4.9 million three dimensional points. The density of the LIDAR point cloud is similar to the uniform density of the DEM. The point cloud comprises about 13.6 percent of the points in the entire large T tile.

Since LIDAR comes before DEM in the processing chain, it is likely that the LIDAR retains more intrinsic geometric information from the scene. Because of this reason, it seems prudent to try and gain an automatic advantage by working directly with the LIDAR and avoiding the loss of information that occurs from gridding it. This seemed like a very good idea, until it became clear that the process was both intensive, proprietary and not reproducible without other resources. The process of gridding LIDAR is for all intents and purposes, a black-box. The following is a

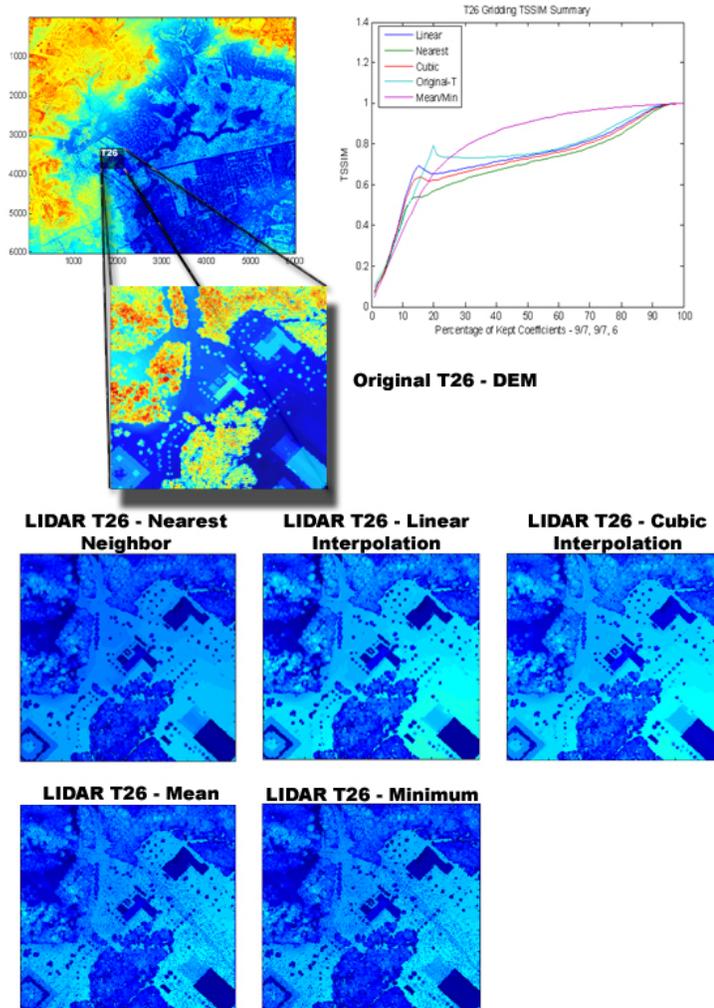
Figure 2.8: This Map Shows the Density of the Point Cloud Samples.



brief summary of our attempts at studying the gridding process, how sensitive our techniques were to the gridding mechanism, and ultimately how the conversion to DEM is quite opaque. For this we create a 26'th T tile, and pull it right from the region above the TEC headquarters.

Note that our gridded versions of T_{26} are not nearly as exciting as the DEM version. We speculate that the data was collected at different times in the year, accounting for several foliage differences. Furthermore, trees can be a tricky issue to accurately model with LIDAR, as the first-pass does not often penetrate the canopy of the trees, while the second-pass can. This can create a multi-valued surface (which it is) near trees, and is often corrected by hand. Similar issues arise over water. We specifically repeated this experiment with another tile, now T_{27} , trying to see the

Figure 2.9: Standard Gridding Techniques on the LIDAR Contained in T_{26} .

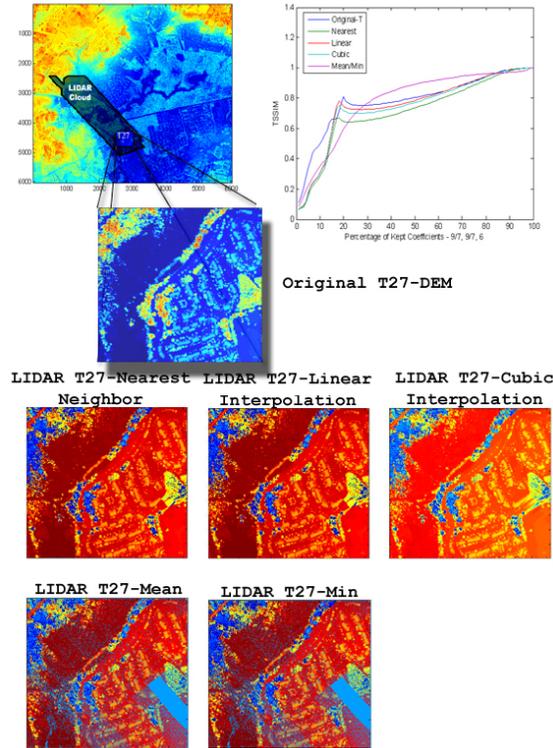


effects of water on LIDAR.

2.2.1 Urban DEM, Quality and the Expert

It is always an interesting endeavor to apply mathematics to questions of subjectivity. How can you quantify attractiveness? Is Monet ϵ more creative than Manet? This is embellishing for the actual question of how do you tailor quality results to your benefactors when the paragon of excellence is to please an expert? In this project, our indicated goal was to represent, compress and reconstruct urban

Figure 2.10: Standard gridding techniques on the LIDAR contained in T_{27} .



DEM in such a way that a trained expert would consider the image representation to be reasonable. This kind of tricky response to the question, ‘what is quality?’, forced us to change our perspective on how to measure visual quality in a quantitative way. There are various schools of thought on this issue, and the norms that first get used to verify the quality of an image are the ℓ^1 norm or the total-variation norm. It has been suggested that there are physiological explanations for why these norms generally behave more like the eye’s response.

Following this line of reasoning, we wished to find a measure of quality, that reflected the human eye’s ability to distinguish structures and differentiate similar images based on changes that mean-squared-error, ℓ^1 and total-variation do not always pick up on. If we can measure images against each other, especially in a

way that is amenable to the human eye, then we are one step closer to pleasing the experts. Since we are not working with images, but scenes in which structure and detail are the prominent features, we can narrow our scope to finding a measure that approximates the human eye's ability to distinguish minute structural differences in similar scenes.

We will restrict the field of applicable DEM data by insisting that we only are concerned with urban DEM. This has its own inherent problems with lack of a rigorous definition. Are the suburbs urban? Is the Guggenheim a structure that we think of as being in an urban environment? This issue was avoided ahead of time by quizzing the experts and asking them to sort a collection of tiles into urban, rural and suburban. In general, we will just think of the distinction as man-made forms versus natural scenery. For us, urban DEM contains heavy edge discontinuities that usually lie along polygons. Natural terrain tends to vary continuously along irregular boundaries.

2.2.2 Terrain Structural Similarity Index (TSSIM)

The problem of automatically measuring image quality in a way that agrees with human inspection is not new. In 1995, Eskicioglu and Fisher [30] studied a massive collection of common image quality assessment methods (ℓ^p , MSE, ℓ^∞ and many more), degraded a collection of black and white images, and then had human subjects rank degraded images against the original. They then took the measurements from the collection of quality measures and looked at the correlation

between human and the measures concept of quality.

Their findings were fairly inconclusive, showing some of the measures to be essentially worthless for predicting a human's response to the quality, but they did make an important point. It is not difficult to create an example of a slightly degraded image, that is quite strikingly bad to the human eye, for which the MSE is quite low. It really shouldn't be that easy to fool a standard measure of image quality. Really what needs to be understood is how the eye perceives changes and to model that more efficiently.

In 2004, Wang, Bovik, Sheikh and Simoncelli [51] introduced a new quality index, known as SSIM, and with it improved upon this type of comparison between automatic image quality assessment as compared to human subjective tests. With this experiment, they have shown their index, SSIM, to be far and away better at predicting human quality judgements, over peak signal to noise ratio (PSNR), which is related to MSE, Sarnoff and UQI methods.

Part of their approach is to really pay attention to the underlying physiology of the human eye, and to model their system in a similar manner. They remark that the perceivability of image details depends on the sampling density of the image signal, the distance from the image plane to the observer, and the perceptual capability of the observers visual system. In practice, the subjective evaluation of a given image varies when these factors vary.

They first act to separate the following three layers from an image: luminance, contrast and structure. These will be operated on independently.

The SSIM is a statistical measure. It is computed based on the moments of

the pixel value distribution. Fix a window size N . Let $a = \{a_i\}_{i=1}^N$ and $b = \{b_i\}_{i=1}^N$ be the pixels corresponding to this window, centered on the target pixels in the two images, respectively. We first compute the weighted mean for each window:

$$\mu_a = \sum_{i=1}^N w_i a_i \text{ and } \mu_b = \sum_{i=1}^N w_i b_i.$$

where the positive weights w sum up to 1, thus providing a weighted average. The case when all the w are equal corresponds to the statistical mean estimator, but inverse distance weights can also be used. We performed our experiments with an 11×11 Gaussian window.

The luminance comparison between the two target pixels is given by:

$$l(a, b) = \left(\frac{2\mu_a \mu_b}{\mu_a^2 + \mu_b^2} \right)$$

This index ranges from 0 to 1, with 1 being the case for $a=b$.

Next, the variances (second moments) are computed:

$$\sigma_a^2 = \sum_{i=1}^N w_i a_i^2 - \mu_a^2 \text{ and } \sigma_b^2 = \sum_{i=1}^N w_i b_i^2 - \mu_b^2$$

and the second component is defined to be

$$c(a, b) = \left(\frac{2\sigma_a \sigma_b}{\sigma_a^2 + \sigma_b^2} \right).$$

The third component is based on the crosscorrelation:

$$\sigma_{a,b} = \sum_{i=1}^N w_i a_i b_i - \mu_a \mu_b$$

and defined as:

$$s(a, b) = \left(\frac{\sigma_{a,b}}{\sigma_a \sigma_b} \right).$$

Combining the three components multiplicatively, we obtain:

$$SSIM(a, b) = \left(\frac{2\mu_a\mu_b + C_1}{\mu_a^2 + \mu_b^2 + C_1} \right) \left(\frac{2\sigma_a\sigma_b + C_2}{\sigma_a^2 + \sigma_b^2 + C_2} \right) \left(\frac{\sigma_{ab} + C_3}{\sigma_a\sigma_b + C_3} \right)$$

where the constants C_i are included for numerical stability in the computation.

A MATLAB implementation of the SSIM index is available at

<http://www.cns.nyu.edu/~zwang/files/research/ssim/> .

For the purposes of studying terrain elevation data, we chose to discard the luminance component, considering that absolute mean elevation differences play little role in evaluating the similarity of urban DEMs. We thus defined the Terrain Structural similarity index as:

$$TSSIM(a, b) = \left(\frac{2\sigma_a\sigma_b + C_2}{\sigma_a^2 + \sigma_b^2 + C_2} \right) \left(\frac{\sigma_{ab} + C_3}{\sigma_a\sigma_b + C_3} \right).$$

With the appropriate choice of constants, this expression simplifies to:

$$TSSIM(a, b) = \frac{\sigma_{ab} + C}{\sigma_a^2 + \sigma_b^2 + C},$$

It should be noted that while TSSIM is a remarkable tool for detecting visual differences in DEM, it is extremely sensitive to translation, rotation and scaling. It would clearly not be a strong choice if we were comparing to sets of data in which

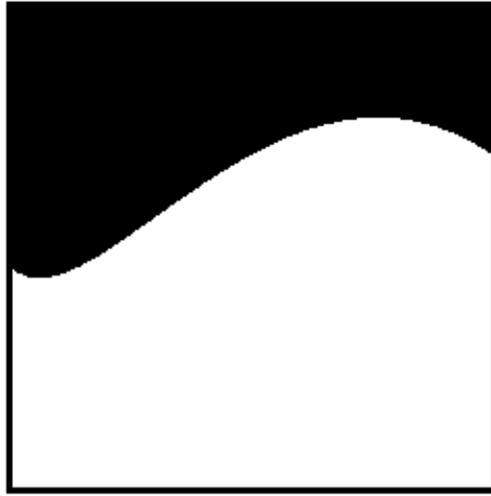
the movement of a sensor provided error within one of these fields. However, this is not the case for us, and in fact, since DEM has so much more underlying structure than do black and white images, we would be hard pressed to find a better measure of quality that conforms to the experts as well. Note that TSSIM is not a metric, and two images I_1 and I_2 are equal if and only if both have the same size and $TSSIM(I_1, I_2) = 1$. Furthermore, note that $TSSIM(a, b)$ is actually a matrix of local TSSIM values. When we talk about the value of TSSIM between two images, we are referring to the average across this entire TSSIM map.

2.3 Time-Scale Compression Methods

Computer vision and image processing have long known that edges are the predominant feature in images: they govern the structure and are boundaries for remaining smooth portions. The multi-scale property of wavelets have brought new tools into the realm of dealing with edges within images, but most often these tools are clunky and difficult to work with for the kinds of edges typically found in images. In [43] the authors described the deficiency of standard wavelet technique to model the edges within images as a problem with the underlying description of these edges. The issue apparently is due to the behavior of Besov models with relation to so-called cartoon-like models. These cartoon-like models showed how sub-optimally standard wavelet techniques dealt with these edges. The standard cartoon model, as seen below, is composed of two distinct constant valued regions separated by a curve that is C^2 , and is more generally in the literature, piece-wise

C^2 .

Figure 2.11: Simple Example of a Cartoon Model.



To combat these problems, several new techniques were established to more optimally decompose the cartoon-like model of edges within images. One way to think about this optimal decomposition is in expressing an edge along a nice discontinuity in some manner that is intrinsic to the image, and therefore requires very few coefficients to store. Vetterli and Do have a fantastic image [23] which expresses this well.

In 2004 Candes and Donoho introduced curvelets [13]. This was the first attempt at sparsely approximating the cartoon model, and doing so in an optimal manner. Their idea is novel and powerful, and has led to a variety of important offshoot results. Contourlets were introduced by Vetterli and Do [23] [24] in 2005 and sought to perform a similar function as curvelets, with a new partition of the frequency domain, directional filter-bank analysis and user-friendly discrete framework for an algorithmic implementation.

Figure 2.12: Modeling an Edge in a Cartoon Model: Do and Vetterli.

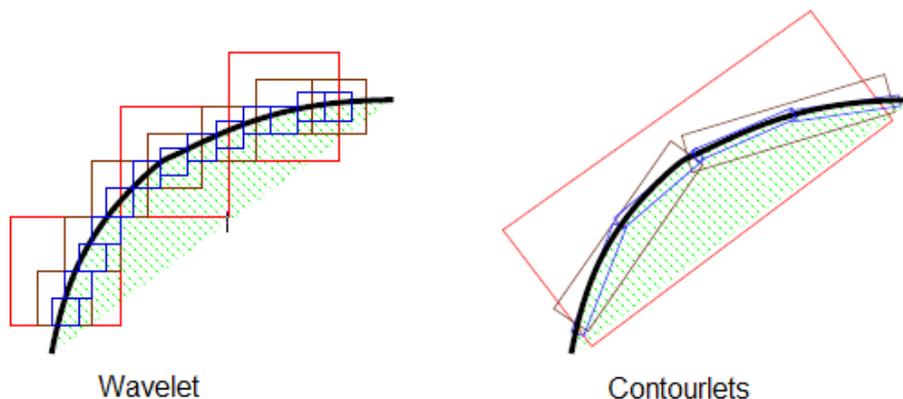
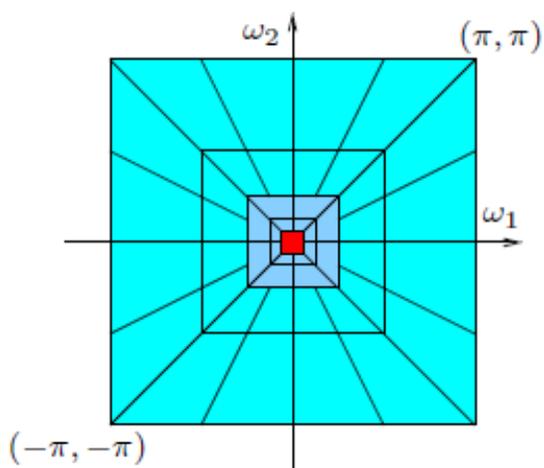


Figure 2.13: Frequency partition for Contourlets.



Ridgelets were a precursor to Contourlets, and was created by Vetterli and Do in 2003 [25] to enhance the ability of wavelet systems to be directionally sensitive. Many of the features of ridgelets become an important part of the wedgelet representation scheme, which will be detailed later in this chapter. In 2005, Labate, Lim, Kutyniok and Weiss [44] furthered each of these methods by creating the shearlet transform. Unfortunately this transform was not included in the analysis of DEM

data types, as the transform was still quite new at the time of this projects inception. It would be interesting to measure how well that transform would perform against the others gathered already.

This short section shall compare a few standard wavelet compression methods to those new methods which attempt to encode edge data in an optimal way for the cartoon model, and will identify the method best able to represent, compress and accurately reconstruct urban DEM data types. To provide a baseline level of comparison to more traditional image compression techniques, we have included the discrete cosine transform (DCT) [54] and a couple of standard (5/7 and 9/7) wavelet filter banks [46] into our list of techniques.

The experiment consists of the following steps:

1. Select a DEM dataset I , $(S_1 - S_{15}, T_1 - T_{25})$,
2. Decompose the DEM data set I into
 - 5/3 wavelet filter-bank,
 - 9/7 wavelet filter-bank,
 - the discrete cosine transform,
 - the curvelet transform,
 - the contourlets transform (1 level),
 - the contourlets transform (2 levels),
 - the contourlets transform (3 levels),
 - the ridgelet transform.

3. Stepping p back from 100% to 0%, threshold the representations from 2. retaining p percent of the coefficients. Reconstruct each threshed representation and save the reconstructed image, I_p ,
4. Generate curves comparing the original DEM data set I to each of the reconstructed images varying by p percentage, I_p , using the following metrics and quality index,
 - $\log(1 + \|I - I_p\|_{\ell^2})$,
 - $\log(1 + \|I - I_p\|_{\ell^\infty})$,
 - $\|I - I_p\|_{tv}$, where tv is the total-variation norm,
 - $TSSIM(I, I_p)$.

We have used this experiment flow-chart with three sets of data. First we have processed the fifteen S tiles using the above steps. For each norm-type, we have averaged across the fifteen S tiles, creating four graphs, one for each of the used norms/quality index. Each curve represents a single multi-scale decomposition method, with the results being averaged across each of the S tiles.

Similarly, the T tiles have been processed using the experiment guidelines and have been averaged across the 25 different T tiles.

The next group of results deals solely with the tile S_{10} . Instead of thresholding from 100 percent to 0, we have only run through the important visual range of 0 to 30 percent of retained coefficients.

These metrics/quality index exhibit very different behavior as the DEM data

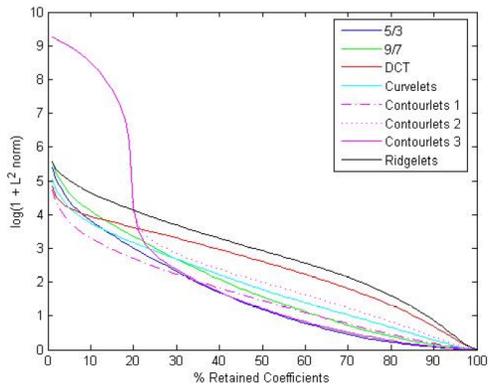


Figure 2.14: S-tiles Avg.: $\log(1 + || ||_2)$

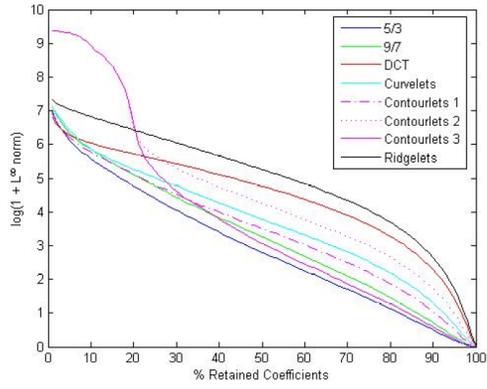


Figure 2.15: S-tiles Avg.: $\log(1 + || ||_\infty)$

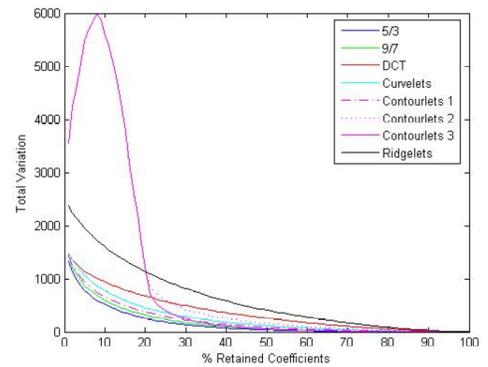


Figure 2.16: S-tiles Avg.: total variation

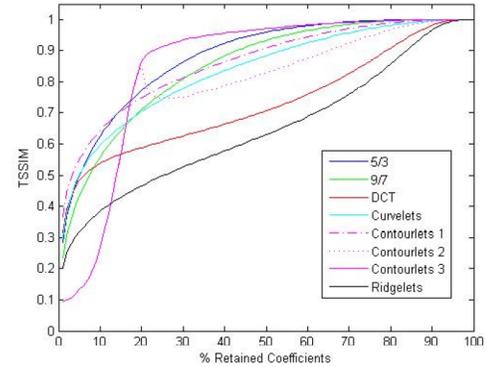


Figure 2.17: S-tiles Avg.: TSSIM

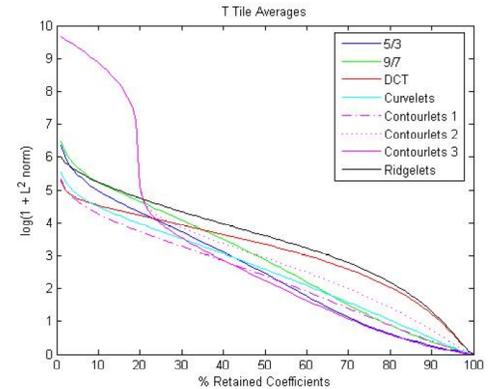


Figure 2.18: T-tiles Avg.: $\log(1 + || ||_2)$

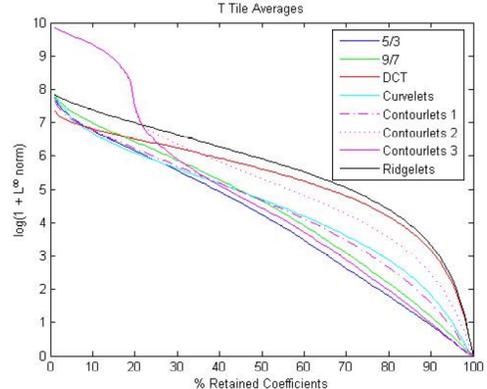


Figure 2.19: T-tiles Avg.: $\log(1 + || ||_\infty)$

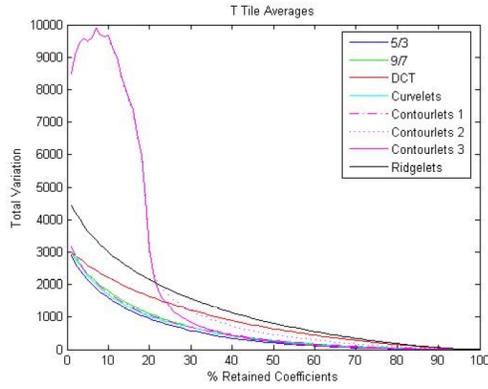


Figure 2.20: T-tiles Avg.: total variation

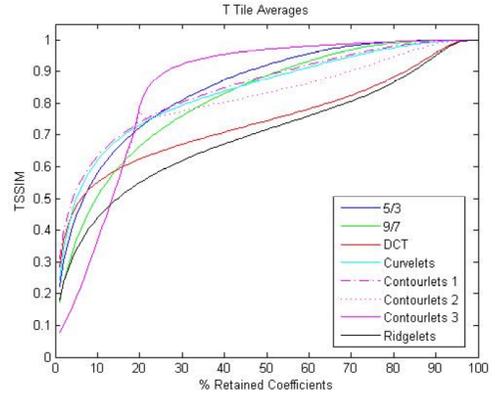


Figure 2.21: T-tiles Avg.: TSSIM

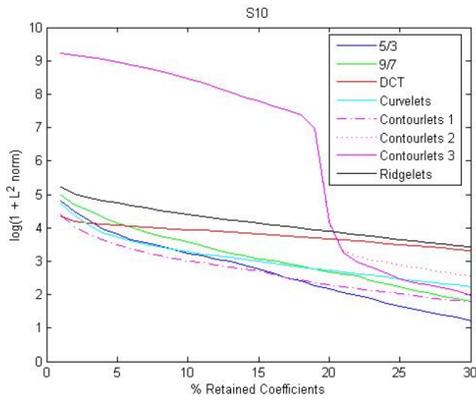


Figure 2.22: S_{10} : $\log(1 + || ||_2)$

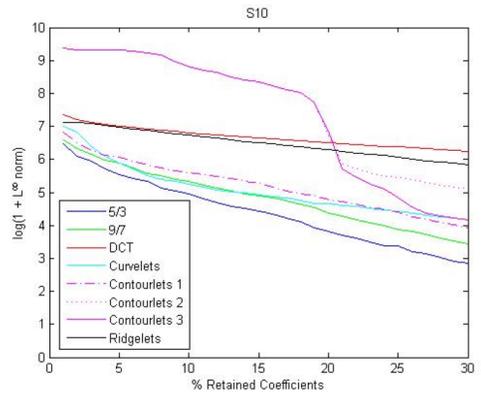


Figure 2.23: S_{10} : $\log(1 + || ||_\infty)$

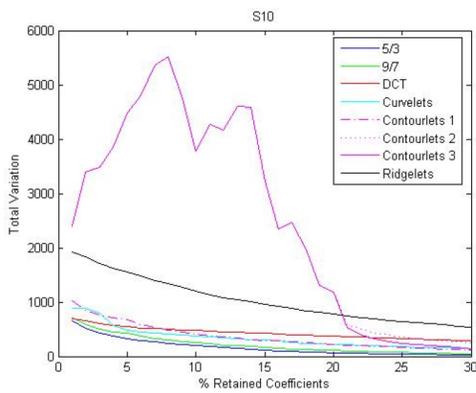


Figure 2.24: S_{10} : total variation

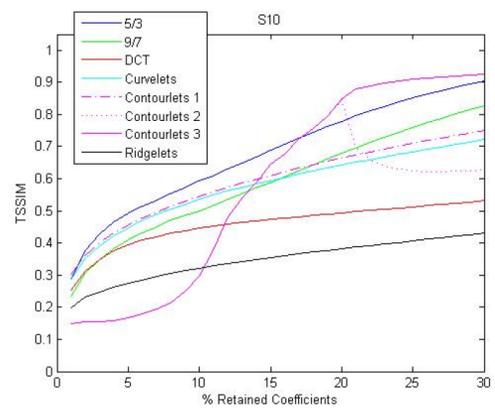


Figure 2.25: S_{10} : TSSIM

sets are degraded more and more. It became clear during this experiment that the TSSIM index became a more reliable measurement device as compared to the expert opinions at the Topographic Engineering Center. In fact, we established that a TSSIM of 0.8 between two DEM tiles (especially when those tiles were urban) represented a ‘good-enough representation’ for the experts. Interestingly enough, the Contourlets level 3 decomposition does essentially as well as the other representations until approximately 20 percent retained coefficients. At this point the Contourlets level 3 method jumps far ahead of the other methods, for the combined T and S tile averages. Note that around this 20 percent jump, the TSSIM values hover at or above 0.8. The blow-up of this process for S_{10} help to illustrate that for urban DEM tiles, this effect is even sharper. Given that TSSIM is the index for which we want to optimize, it became clear that contourlets were the optimal representation to represent urban DEM.

This experiment used brute force to come to a quick solution of how to represent urban DEM in such a way that the experts find the representation satisfactory. We have avoided a more intricate study of these methods and how various thresholding techniques, or even small changes in underlying data may effect the results. The most important information that comes from this data, are the baseline results that we can use to improve on the technique. The next section of this chapter will deal with a multi-scale geometric representation which exceeds the ability of Contourlets to represent DEM data types.

2.4 Geometric Representation: Wedgelets

Wavelets have certainly enjoyed a strong following in the past 25 years, but they are not always the most useful or intuitive technique for certain problems. One of the characteristics of urban DEM is the proliferation of heavy edges throughout the scene. While directional wavelet methods help to capture these edges, they still add undesirable noise, by adding Gibbs-like ringing.

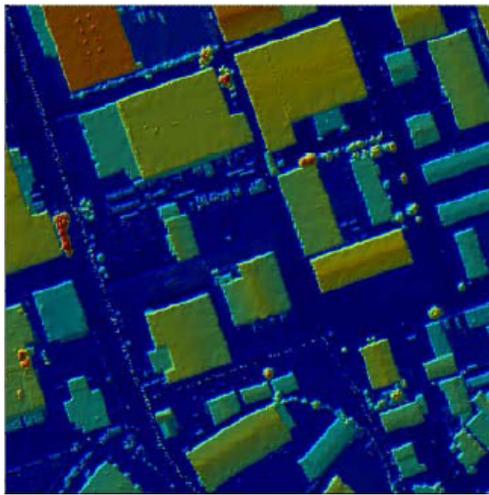


Figure 2.26: S_{10} .

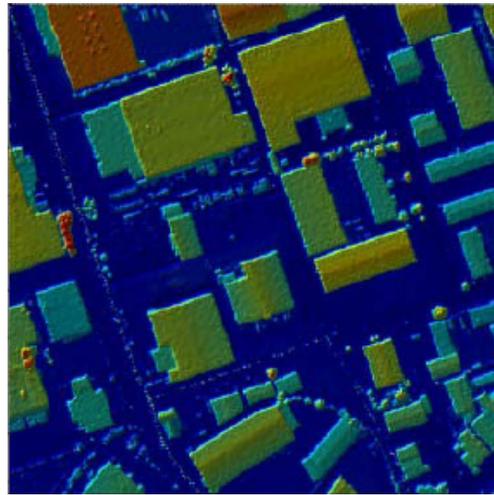


Figure 2.27: Contourlets defects.

This ringing acts to smooth out the gradient, rendering much of the information from the DEM useless.

One way to deal with this phenomenon is to think more intuitively about how we want to represent this type of data. For instance, ask what kind of shapes we are seeing primarily, how much elevation change do we expect, and what are typical neighborhoods in the data resembling? By asking these specific questions we can note that urban DEM contains edges, corners and very few curves. Unless you are analyzing the Guggenheim museum, urban DEM contains flat or constant sloped

rooves. Furthermore, we typically don't find an abundance of natural shapes like trees, or lakes in urban DEM. This means that we are interested in representing simple flat and constant sloped objects that are well surrounded by heavy edge and corner detail.

Figure 2.28: Aligned edges, polygonal edges and flat surfaces.



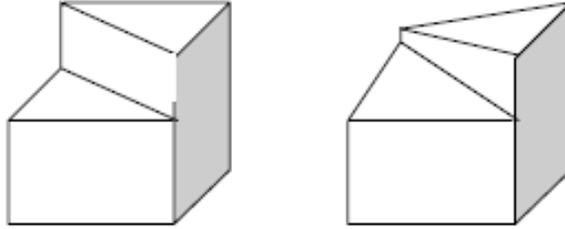
This image, figure 2.28 is as urban as DEM tiles get. The buildings have flat or slightly sloped rooves, the buildings align on a grid that appears to be shifted approximately 20 degrees. There are very few details (circled) to stand in the way of large simple objects to represent.

2.4.1 Wedgelets

In [26] Donoho introduced a multi-scale representation scheme that essentially extracts geometric edge data from an image and encodes it efficiently into piece-wise

constant functions that are supported on a dyadic grid of wedges.

Figure 2.29: Examples of Wedgelet Functions.



This construction is highly amenable to our urban DEM data because of the prevalence of edges and piecewise constant regions in that data.

His aims are to optimally represent the cartoon model sparsely, naturally and in some type of min-max sense. We wish to use this method because intuitively buildings, roads and similar objects share many geometric properties with the cartoon model (except for the C^2 along the boundaries of discontinuities). Rectangular buildings with flat rooves can be perfectly represented by such functions supported on wedges. Collapsing a building down to such a simple function allows a great deal of room for compression.

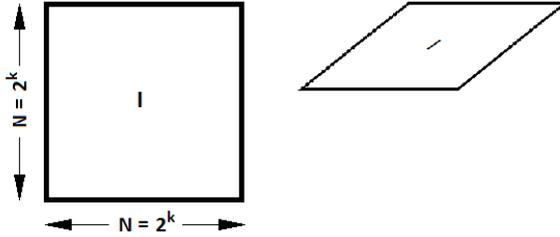
We shall adapt and improve on Donoho's system in three ways:

1. We use a recursive scheme to compute the wedgelet transform in a faster way,
2. We remove the constraint that our representation functions be piece-wise constant, allowing for our surfaces to be planar,
3. We generalize the notion of wedgelets to the underlying data type, LIDAR point clouds.

2.4.2 Piece-Wise Constant Wedgelets

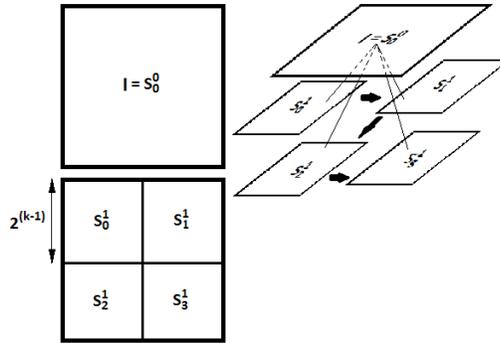
To fix ideas and simplify our notation, we will describe the wedgelet algorithm using a dyadic square image, where $I \in M_{N \times N}(\mathbb{R})$ is an $N \times N$ image, where $N = 2^k$ for some k .

Figure 2.30: The Basic Wedgelet Set-Up.



Next we decompose the image I into a dyadic multi-scale tree, $\{S_l^j\}_{j=0, \dots, k, l=0, \dots, 2^j+1}$, where j represents the depth of the decomposition and l keeps track of the $2^j + 1$ sub-regions at that depth. So at each depth j , there is a partition of $I = \cup_{l=0}^{2^j+1} S_l^j$.

Figure 2.31: First Layer Dyadic Decomposition.



The ordering structure begins at depth 1 as, from left to right, and top to bottom, and then for each of the following depths, the ordering is done recursively.

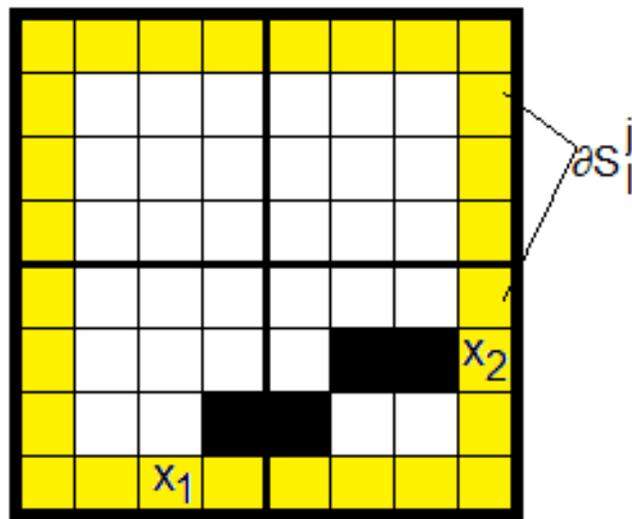
Note that S_l^j is an $\frac{N}{2^j} \times \frac{N}{2^j}$ sub-image of I . This decomposition continues down to pixel-scale, i.e. $j = \log_2(N)$.

Figure 2.32: Illustration of ordering in Layer 2.

S_0^2	S_1^2	S_4^2	S_5^2
S_2^2	S_3^2	S_6^2	S_7^2
S_8^2	S_9^2	S_{12}^2	S_{13}^2
S_{10}^2	S_{11}^2	S_{14}^2	S_{15}^2

The next stage of the wedgelet decomposition of I involves bisecting each sub-region S_l^j by a collection of line segments and creating a pair of wedges supported on S_l^j . Donoho does this by assigning any two non-redundant pixels on the boundary of S_l^j as being end-points of the edgelet $E_l^j(x_1, x_2)$, where $x_1, x_2 \in \partial S_l^j$. Edgelets are directed from x_1 to x_2 as a mechanism to keep different wedges defined by the same edgelet separate.

Figure 2.33: An Edgelet on the sub-square S_l^j .



In practice, parameterizing the bisection in this way can be troublesome com-

putationally. Instead we opt to fix a finite set of angles, $\Theta \subset [-\frac{\pi}{4}, \frac{3\pi}{4})$, and by a finite collection of offsets $k \in \mathbb{Z}$. Since compression is a goal, it is important to see that parameterizing our edgelets in this manner does not cost any extra storage, since θ can be chosen from a finite collection of pre-rendered angles and k is naturally integral. This ordered pair (θ, k) is then a collection of two integers which is precisely the same amount of storage needed to describe edgelets in Donoho's scheme. This choice of parametrization does not come without cost, and accordingly we must build some machinery to deal with this change.

Local vs. Global Coordinates: The first order of business is to recognize the distinction between local and global coordinates when working with sub-regions from the tree $\{S_l^j\}$. When we are working specifically on the region S_l^j , the local coordinates will be taken from the matrix representing the image on that region. That is

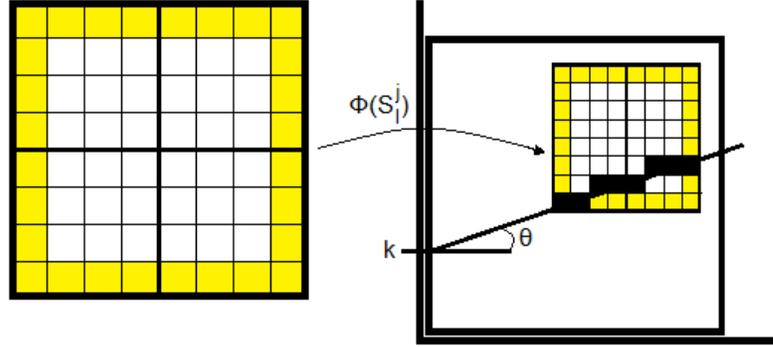
$$(x_1, x_2) \in \mathbb{Z}^2 \cap [0, \frac{N}{2^j} - 1]^2.$$

However, we must remember that these points exist globally within the context of the image I . In fact, all lines and edgelets will be defined globally to save computational time. To remedy this, we introduce the coordinate transform, $\Phi^j : I \rightarrow \cup_{l=0}^{2^j-1} S_l^j$ by

$$\Phi^j(x_1, x_2) = (x_1 - \left\lfloor \frac{x_1 2^j}{N} \right\rfloor \frac{N}{2^j}, x_2 - \left\lfloor \frac{x_2 2^j}{N} \right\rfloor \frac{N}{2^j}).$$

Note that by iteratively applying this transform, we can locate each l corresponding to j so that the point $(x_1, x_2) \in S_l^j$. This is nothing more than an application of the division algorithm. This allows the coordinate transform to be inverted, by stepping up through the depths iteratively as well.

Figure 2.34: An Edgelet on the sub-square S_l^j and its global coordinates.



Definition 1. We will borrow a useful definition of digital line from [33]. Consider an image I that is affixed to a coordinate axis so that the pixels line up with an integral mesh of \mathbb{R}^2 . Consider the line $y = \theta x + k$. We define the digital line $L(\theta, k)$ to be

$$L(\theta, k) = \{(a, b) \in \mathbb{Z}^2 : (k - \frac{1}{2})\delta_\theta < \langle (a, b), v_\theta^\perp \rangle \leq (k + \frac{1}{2})\delta_\theta\}$$

where

$$\delta_\theta = \max\{\cos |\theta|, \sin |\theta|\}$$

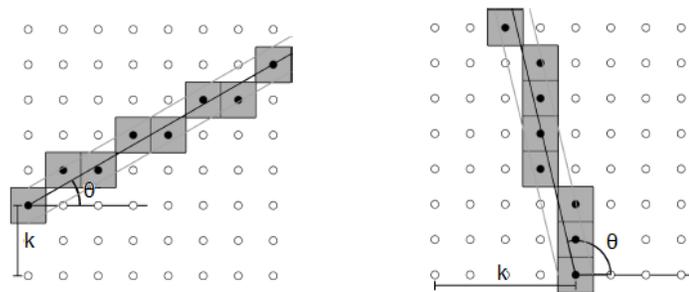
and

$$v_\theta^\perp = \begin{cases} (-\sin \theta, \cos \theta) & |\cos \theta| \geq |\sin \theta| \\ (\sin \theta, -\cos \theta) & \text{otherwise} \end{cases}.$$

Definition 2. Let $\theta \in [-\frac{\pi}{4}, \frac{3\pi}{4})$ and let (j, l) be a valid set of coordinate for the tree $\{S_l^j\}$. We will decompose this range of angles into 4 regions, modified by steep/flat and standard/alternative to

1. We call $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4})$ a flat angle and $\theta \in [\frac{\pi}{4}, \frac{3\pi}{4})$ a steep angle,
2. We call $\theta \in [-\frac{\pi}{4}, 0) \cup [\frac{\pi}{2}, \frac{3\pi}{4})$ a standard increment set and $\theta \in [0, \frac{\pi}{2})$ an alternate increment set.

Figure 2.35: Examples of steep and flat digital lines.



We allow $\theta \in [-\frac{\pi}{4}, \frac{3\pi}{4})$ and we identify four distinct regions in this set.

Region 1. $[-\frac{\pi}{4}, 0)$ are flat lines of standard increment.

Region 2. $[0, \frac{\pi}{4})$ are flat lines of alternate increment.

Region 3. $[\frac{\pi}{4}, \frac{\pi}{2})$ are steep lines of alternate increment.

Region 4. $[\frac{\pi}{2}, \frac{3\pi}{4})$ are steep lines of standard increment.

We make these distinctions for a few important reasons concerning the computational use of (θ, k) . We wish to make sure that as we vary our lines (of fixed angle θ) we do not sweep out redundant wedges.

Note that a flat line is just that; a collection of pixels such that no vertical line will ever cross more than one pixel. Similarly steep lines are collections of contiguous pixels such that no horizontal line will ever cross more than a single pixel.

This means that with a fixed θ , a steep line incremented horizontally will never overlap with a previous or future version of itself. A similar statement is true of flat lines incremented vertically.

Figure 2.36: The four angular regions.

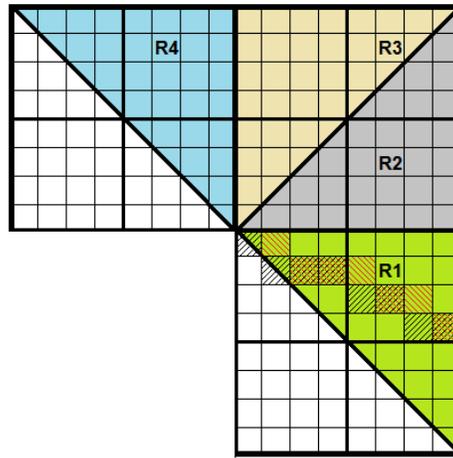
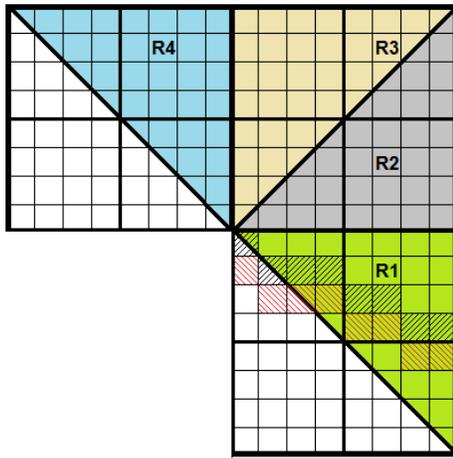
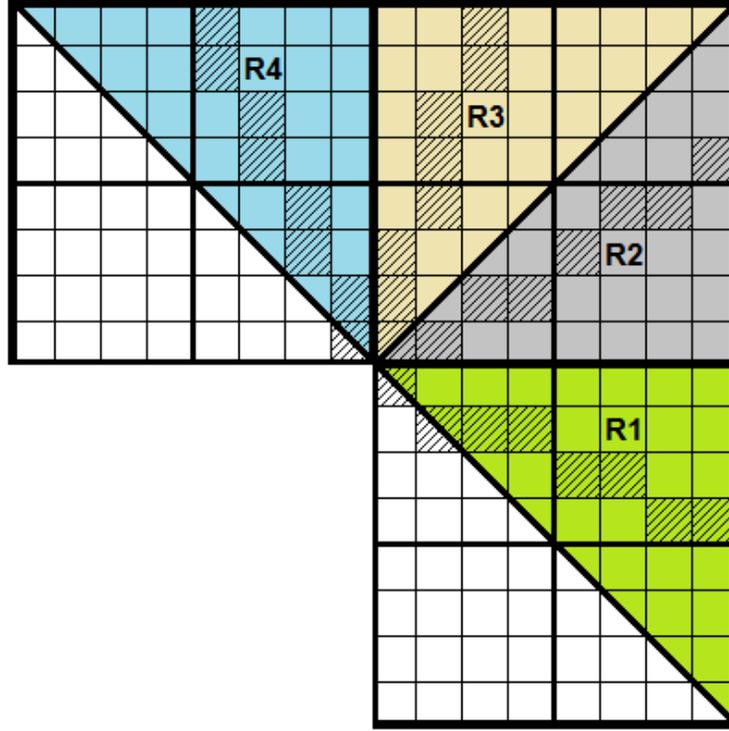


Figure 2.37: Correct Flat Increment.

Figure 2.38: Incorrect Flat Increment.

We do not want to add superfluous calculations, and so we investigate the following cases to lessen our computational workload.

This is where standard and alternative increments come in to play. For $\theta \in$

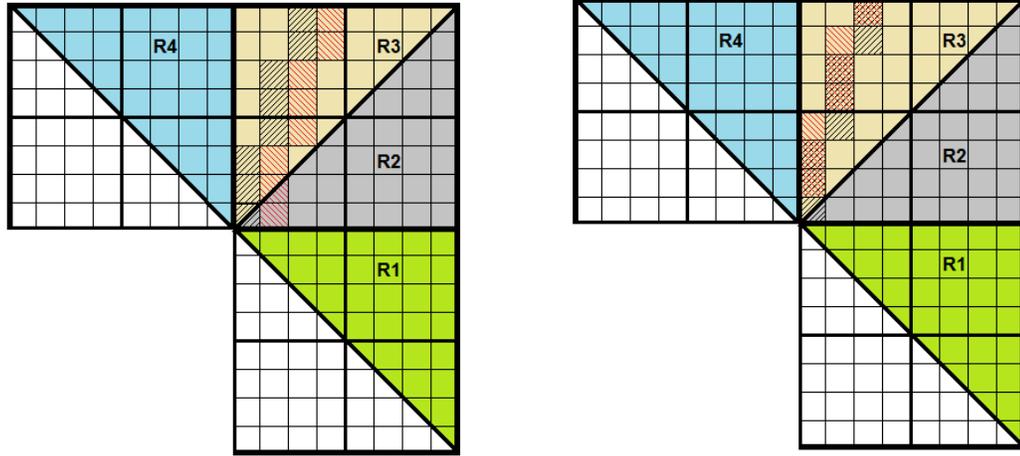
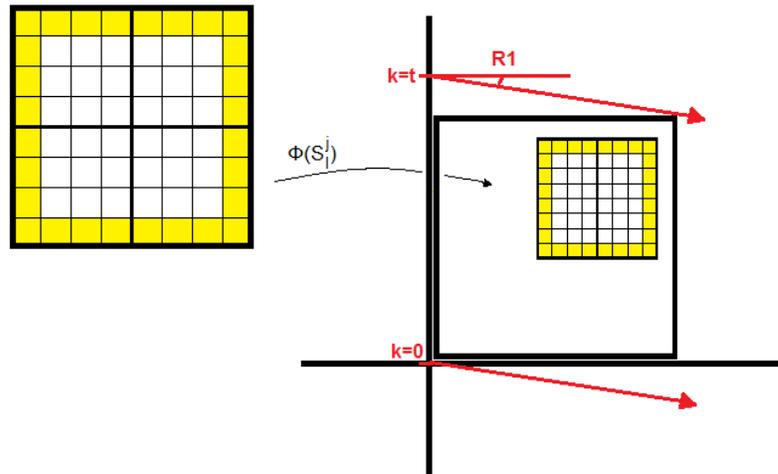


Figure 2.39: Correct Steep Increment. Figure 2.40: Incorrect Steep Increment.

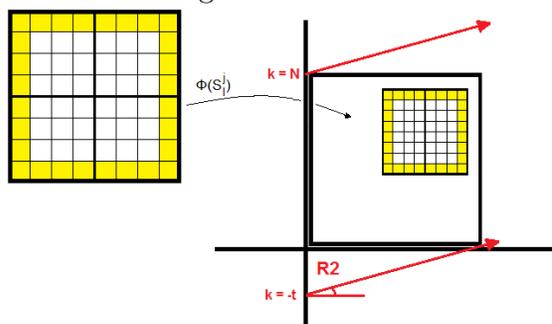
$[-\frac{\pi}{4}, 0)$ (region 1), $L(\theta, 0)$ corresponds to the first allowable digital line and $L(\theta, t)$ is the final line. The increment values for this line would then occupy the range $[0, t] \cap \mathbb{Z}$ where these values are known to live on the vertical axis.

Figure 2.41: R1 Increment Values.



For $\theta \in [0, \frac{\pi}{4})$ (region 2) we have an alternative increment, which means to allow intersection of the box, our initial line must be incremented as a negative value. So the allowable increments for this line type occupy the range $[-t, \frac{N}{2^j}] \cap \mathbb{Z}$.

Figure 2.42: R2 Increment Values.



Region three and four have the following increment ranges.

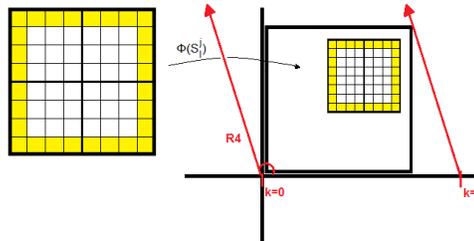
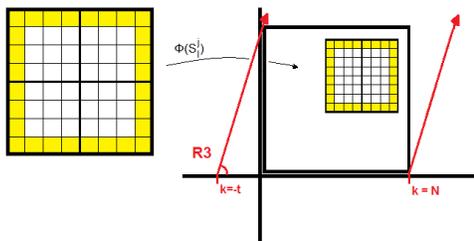


Figure 2.43: R3 Increment Values.

Figure 2.44: R4 Increment Values.

Having these ideas straightened out will make a description of the algorithm exceptionally simple.

Next we define what a wedge and a wedgelet are. Suppose that $\theta \in [-\frac{\pi}{4}, \frac{3\pi}{4})$ and k is a proper increment for that θ . We define the digital line $L(\theta, k)$ and look at the following scenarios.

We define the main wedge to be the region bounded by the line $L(\theta, k)$ and continuing clockwise back to that line. We call the wedge partitioned opposite the main wedge, the sister wedge.

So for any given S_i^j and (θ, k) we can uniquely define the main wedge and the

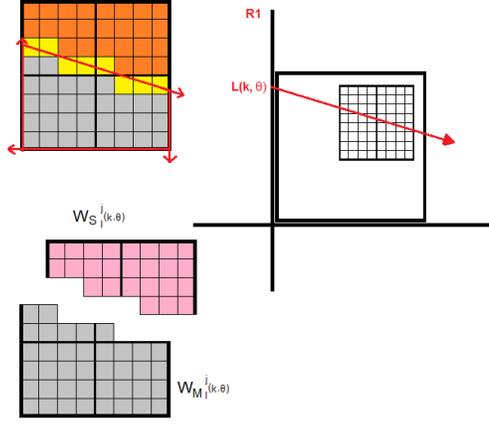


Figure 2.45: R1 Wedgelet.

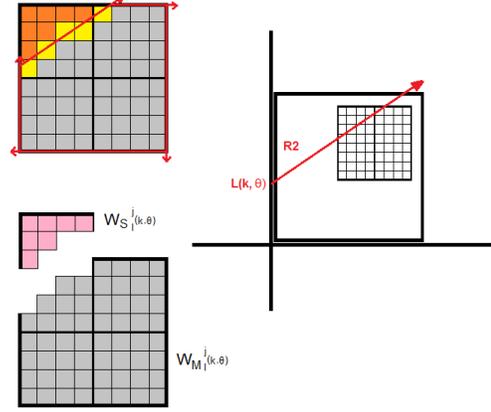


Figure 2.46: R2 Wedgelet.

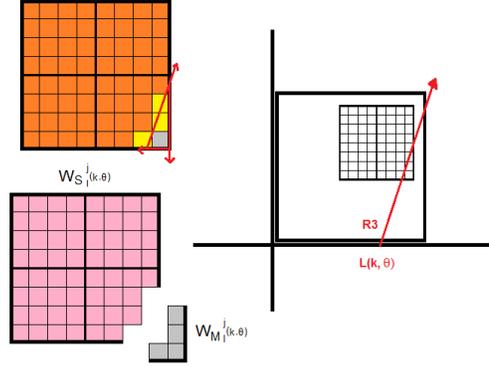


Figure 2.47: R3 Wedgelet.

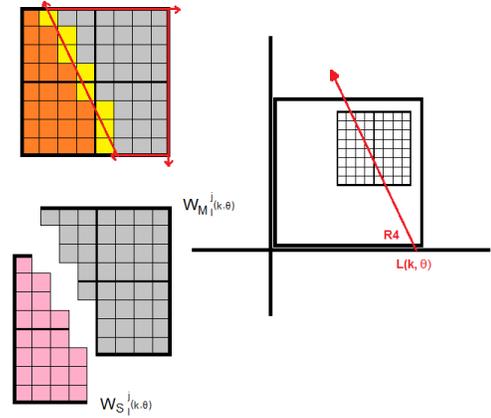


Figure 2.48: R4 Wedgelet.

sister wedge, W_M and W_S . Once we have obtained these wedges, we can define the piece-wise constant functions supported on them as

$$W_l^j(\theta, k) = \left(\frac{1}{|W_M|} \sum S_l^j|_{W_M} \right) \mathbb{1}_{W_M} + \left(\frac{1}{|W_S|} \sum S_l^j|_{W_S} \right) \mathbb{1}_{W_S}.$$

These are all of the tools needed to perform the wedgelet transform on a dyadic squared image I . In the recursive portion of the transform, we make use of a local

error norm. This norm on its own has been the subject of much research, varying from standards such as ℓ^1 , ℓ^2 and total variation norms, to the idea of creating a local scaled version of TSSIM that could somehow act more like a metric instead of an index. Either way, the norm used has to actually satisfy the triangle inequality to allow for the pruning of the full coefficient tree to a sparse reconstruction partition. In practice, it is useful to think of this norm as being either ℓ^1 or ℓ^2 .

2.4.3 The Wedgelet Transform

Algorithm 1 Wedgelet Transform

```

for  $\theta_i \in \Theta$  do

    for  $k \in$  possible increments for  $S_0^0$  do

         $WT(S_0^0)$ 

    end for

end for

```

At the end of the transform we have a coefficient tree, where each node represents an S_l^j and holds the value of the $\theta, k, V_{W_M}, V_{W_S}$ for which $\|S_l^j - W_l^j(\theta, k)\|$ is minimal.

To summarize this process, for each θ and k that makes sense in this context, and in a recursive sense, for each S_l^j that intersects the digital line $L_{(\theta,k)}$, we subdivide the square into a main and sister wedge, and average the values on these wedges and form the wedgelet supported on them. We measure the approximation of that wedgelet to the original and if it is better than our previous 'best approximation', we save the data from this wedgelet at the S_l^j node of our large data structure.

Algorithm 2 $WT(S_i^j)$

Generate $L(\theta, k)$.

Create Main and Sister Wedges W_M and W_S .

Create the wedgelet $W_i^j(\theta, k)$.

Compute the current error

$$\|S_i^j - W_i^j(\theta, k)\| = E_i^j(\theta, k).$$

If $E_i^j(\theta, k) < Err_i^j$ then set the current node as the active node and replace

$$Err_i^j = E_i^j(\theta, k).$$

Subdivide S_i^j into the four children squares $S_{l_1}^{j+1}, S_{l_2}^{j+1}, S_{l_3}^{j+1}, S_{l_4}^{j+1}$.

for $p = 1$ to 4 **do**

IF $S_{l_p}^{j+1} \cap L(\theta, k) \neq \emptyset$ and the side length of $S_{l_p}^{j+1}$ is greater than or equal to the size of a pixel

$WT(S_{l_p}^{j+1})$

end for

For a large image, this is a very sizeable computation. The resulting data structure contains the best wedgelets on each scale, greatly reducing the amount of information that is needed to be stored. However, we still have a large amount of redundant information stored in the data structure. Note that for pretty much any discontinuity that resembles the cartoon model, there will be some collection of resolutions that are able to reproduce that discontinuity very well, as long as there are few other elements in range of the edge and there are no corners or drastic gradients. This is the strength of wedgelets to resolve those cartoon models efficiently.

An example of a tree for the S_{10} data tile could show the following wedge partitions at varying scales.

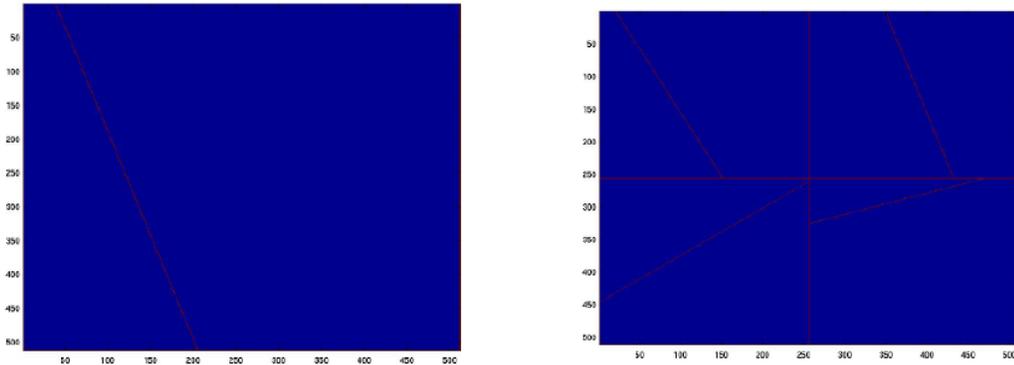


Figure 2.49: Level 0 Wedge Partitions. Figure 2.50: Level 1 Wedge Partitions.

Definition 3. We define a valid reconstruction partition of the quad-tree $\{S_l^j\}$ to be any sub-collection of these sets (nodes in the coefficient tree) for which each pixel in the original image is represented in one square S_l^j once and only once.

From our data structure we can piece together many valid partitions of the original image I . At coarse scales (j is small) we have large wedges that represent

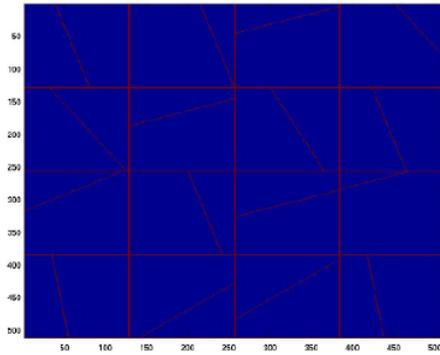


Figure 2.51: Level 2 Wedge Partitions.

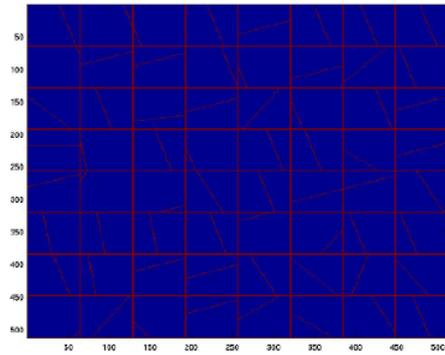


Figure 2.52: Level 3 Wedge Partitions.

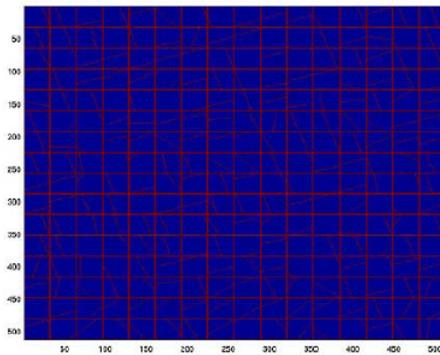


Figure 2.53: Level 4 Wedge Partitions.

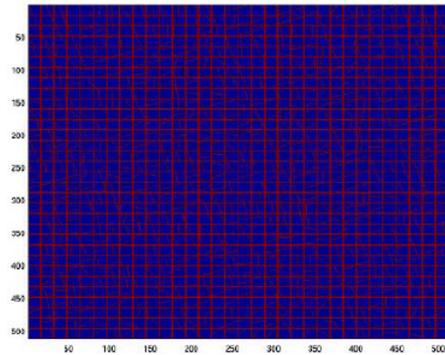
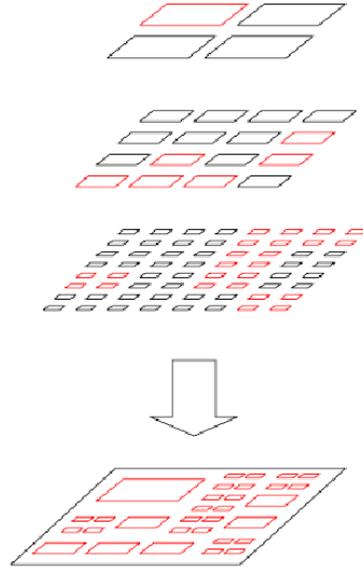


Figure 2.54: Level 5 Wedge Partitions.

large areas, with the benefit of high compression and at the cost of a loss of minute details. On the other hand, at high resolution we can zero in on perfect reconstruction, but do so at the cost of having to store immense amounts of information (Storing a group of 4 pixels requires 4 real numbers to be stored, while representing 4 pixels as a wedgelet requires 2 integers and 2 real numbers, which is a very low savings as opposed to a wedgelet representing a much larger region).

From this data structure we need to construct a representation that is optimal, in some way.

Figure 2.55: Example of a Valid Partition.



2.4.4 Pruning the Quad-Tree

We introduce the parameter $\lambda > 0$. This parameter will balance in our reconstruction, the value of compression vs. precision.

For any valid full partition of I , $\{S_{l_p}^{j_q}\}$ we can define the functional

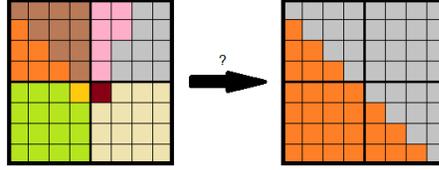
$$F(I_{rec}) = \lambda \left| \{S_{l_p}^{j_q}\} \right| + \|I - I_{rec}\|.$$

The goal is to minimize this functional on the tree for all full valid partitions of I . Clearly a large λ emphasizes having few pieces to represent I , while a small λ tries to force an exact approximation. Since our measure of two images being close is within the context of TSSIM and not actually a metric, the above functional is useful only as a guideline for the pruning procedure. Our modified pruning method will therefore not locate the absolute minimum of the functional F , but will trade that for a local minimum that manages to have a competitively large TSSIM. In the

future, making TSSIM in a more metric-friendly manner will aid in the creation of an algorithm that could find a global minimum of such a functional.

Localized Example:

Figure 2.56: Local Pruning Example.



Starting from the basic approximation of the region of I corresponding to S_l^j by $\{S_{l_p}^{j+1}\}_{p=1,2,3,4}$ versus the approximation of that same region by S_l^j , we can ask the question, when do we wish to pass from the approximation on four squares, to the coarser approximation on a single larger square?

$$I_{rec}|_{S_l^j} = W_l^j(\theta_l^j, k_l^j)$$

and

$$I_{rec}|_{\cup_{p=1}^4 S_{l_p}^{j+1}} = \sum_{p=1}^4 W_{l_p}^{j+1}(\theta_{l_p}^{j+1}, k_{l_p}^{j+1}).$$

Here the values $\theta_l^j, k_l^j, \theta_{l_p}^{j+1}, k_{l_p}^{j+1}$ correspond to the fixed values for each coming from the coefficient data structure at the point specified. Locally we can then write the functional decision as

$$F(I_{rec}|_{S_l^j}) = \lambda + \left\| W_l^j(\theta_l^j, k_l^j) - I|_{S_l^j} \right\| = \lambda + E_l^j$$

$$F(I_{rec}|_{\cup_{p=1}^4 S_{l_p}^{j+1}}) = \lambda \cdot 4 + \left\| \sum_{p=1}^4 W_{l_p}^{j+1}(\theta_{l_p}^{j+1}, k_{l_p}^{j+1}) - I|_{S_l^j} \right\| = 4\lambda + \sum_{p=1}^4 E_{l_p}^{j+1}.$$

So we decide to make the change locally if

$$\lambda + E_l^j < 4\lambda + \sum_{p=1}^4 E_{l_p}^{j+1}$$

or whenever

$$E_l^j - \sum_{p=1}^4 E_{l_p}^{j+1} < 3\lambda.$$

We will now apply this localized example of small-scale pruning and use it to generate a formal pruning procedure. Instead of pruning the tree to a full valid partition by globally minimizing the functional F_λ , we do so by starting out with the finest possible approximation (namely I represented at the pixel level) and consider each grouping of 4 pixels to upgrade to a two by two representation.

Since the approximation at pixel level is perfect, we have the following criteria for a group of 4 pixels to be upgraded to a single two by two wedgelet approximation

$$E_l^j < 3\lambda$$

This algorithm is opportunistic in that it moves from clumps that have the most to gain from becoming a single wedgelet, to those that are least likely to benefit.

The true power of wedgelets is then a very fast ability to sort through reconstructions of varying coarseness based on the parameter λ . This of course comes at the cost of a massive front-end computational overhead.

Algorithm 3 Pruning the Wedgelet Tree

Collect all clumps of 4 pixels all belonging to the same parent node.

Collect the error information from each parent node.

Set the error information from pixel nodes to be zero.

ErrorList \leftarrow The list of errors of the parent nodes, as a vector.

ChildErrorList \leftarrow The four child nodes error information. Here ChildErrorList(1) is a vector with 4 entries.

ErrorList \leftarrow SortDescending(ErrorList). Sort ChildErrorList with the same permutation.

while ErrorList(1) - SUM(ChildErrorList(1)) $<$ 3λ **do**

 Flag the node represented by position 1 as current, turn its children nodes off.

 Remove entry 1 from both lists.

if The new entry itself has 3 neighbors of which all share a common parent node which are current nodes **then**

 Place the error from their common parent into the list ErrorList.

 Place the four errors of these neighbors as children in the corresponding place in ChildErrorList.

 ErrorList \leftarrow SortDescending(ErrorList). Sort ChildErrorList with the same permutation.

end if

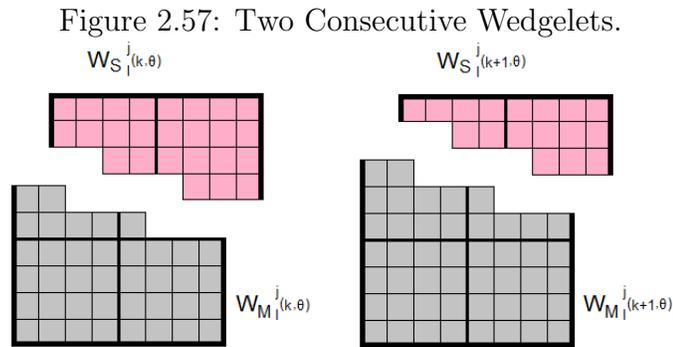
end while

2.4.5 Implementation and a Discrete Green's Theorem

We implemented the wedgelet algorithm in Matlab for testing purposes. A helpful speed-up was implemented from [33], by employing a discrete version of Green's theorem for rapid moment calculation. In essence, since steep or flat lines sweep out non-redundant wedges and partition each of the sub-squares as the index is incremented, the difference between the moment at index k and the moment at index $k + 1$ was the previous moment summed with the information contained on the incremented line $L(\theta, k + 1)$. This can be made clearer with a couple of simple images. Consider that we already know the wedgelet on square S_l^j corresponding to the line $L(\theta, k)$. That is, we have

$$W_l^j(\theta, k) = \left(\frac{1}{|W_M|} \sum S_l^j|_{W_M} \right) \mathbb{1}_{W_M} + \left(\frac{1}{|W_S|} \sum S_l^j|_{W_S} \right) \mathbb{1}_{W_S}. \quad (2.1)$$

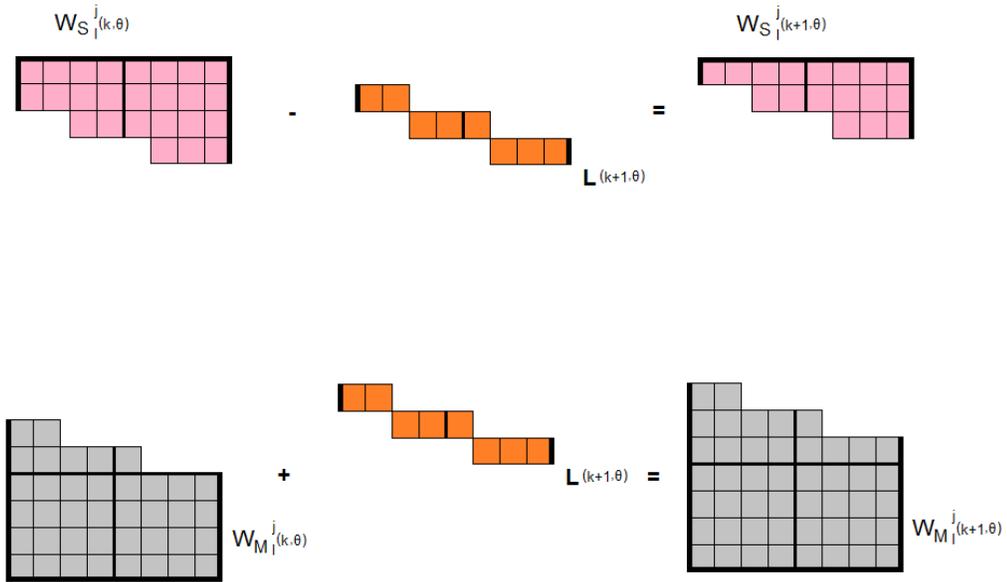
We wish to use this information to speed up the calculation of the wedgelet corresponding to the incremented line $L(\theta, k + 1)$. The following is an image of the wedgelet at k and the wedgelet at $k + 1$.



The digital line sweeps out unique and complete portions of the square, and so

with the following relation, indicating how to convey information about $W_l^j(\theta, k)$ into information about $W_l^j(\theta, k+1)$, we can arrive at a formula which is true independent of k , as long as the line $L(\theta, k)$ and $L(\theta, k+1)$ do not sweep out the same wedgelet on S_l^j .

Figure 2.58: Expression Using $W_l^j(\theta, k)$ to Find $W_l^j(\theta, k+1)$.



That is, if we write the known information as

$$V_{M_l^j}^j(\theta, k) = \left(\frac{1}{|W_{M_l^j}^j|} \sum S_l^j|_{W_M} \right), \quad (2.2)$$

and

$$V_{S_l^j}^j(\theta, k) = \left(\frac{1}{|W_{S_l^j}^j|} \sum S_l^j|_{W_S} \right), \quad (2.3)$$

then we can write the updated moments simply as

$$V_{M_l^j}(\theta, k+1) = \left(\frac{|W_{M_l^j}(\theta, k)| V_{M_l^j}(\theta, k) + \sum S_l^j|_{L(\theta, k+1)}}{|W_{M_l^j}(\theta, k)| + |L(\theta, k+1) \cap S_l^j|} \right) \quad (2.4)$$

and

$$V_{S_l^j}(\theta, k+1) = \left(\frac{|W_{S_l^j}(\theta, k)| V_{S_l^j}(\theta, k) - \sum S_l^j|_{L(\theta, k+1)}}{|W_{M_l^j}(\theta, k)| - |L(\theta, k+1) \cap S_l^j|} \right). \quad (2.5)$$

This leaves us with the updated wedgelet

$$W_l^j(\theta, k+1) = V_{M_l^j}(\theta, k+1) \mathbb{1}_{(W_{M_l^j} \cup L(\theta, k+1))} + V_{S_l^j}(\theta, k+1) \mathbb{1}_{(W_{S_l^j} \setminus L(\theta, k+1))}. \quad (2.6)$$

This means that for this particular square, when the angle is fixed, we can pre-render all of the moments that must be calculated, and we can do so with a simple recursive formula. This is an over simplification of how the pre-computed moments are calculated, as there are variances with changes in angle causing the increments to change, and the line to sweep in a different direction. There is also the issue of how to apply this to nested squares. That is where the discrete Green's theorem comes in. In practice, there are two cumulative matrices that are computed for each image $I \in M^{N \times N}(\mathbb{R})$. These are the vertical and horizontal cumulative matrices C_V and C_H where

$$C_V(i, j) = \sum_{k=1}^i I(k, j),$$

and

$$C_H(i, j) = \sum_{k=1}^j I(i, k).$$

Note that these matrices can easily be adjusted to any of the sub-squares S_l^j simply by a little arithmetic, for instance, with the vertical cumulative matrix on the square S_l^j we have $C_{V_l^j}$ where

$$\begin{aligned}
C_{V_l^j}(i, j) &= \sum_{k=1}^i S_l^j(k, j) = \sum_{k=\Phi_{l_1}^j(1, j)}^{k=\Phi_{l_1}^j(i, j)} I(\Phi_{l_1}^j(k, j), \Phi_{l_2}^j(k, j)) \\
&= \sum_{k=1}^{k=\Phi_{l_1}^j(i, j)} I(\Phi_{l_1}^j(k, j), \Phi_{l_2}^j(k, j)) - \sum_{k=1}^{k=\Phi_{l_1}^j(1, j)+1} I(\Phi_{l_1}^j(k, j), \Phi_{l_2}^j(k, j)) \\
&= C_V(\Phi_{l_1}^j(i, j), j) - C_V(\Phi_{l_1}^j(1, j) + 1, j).
\end{aligned}$$

Here $\Phi_l^j : S_l^j \rightarrow I$ by

$$(i, j) \mapsto (\Phi_{l_1}^j(i, j), \Phi_{l_2}^j(i, j))$$

are the global coordinates of $(i, j) \in S_l^j$ within the image I .

A similar calculation is true of the horizontal cumulative matrix. All that is left to implement this scheme is to calculate an angular moment matrix for each of the finite number of angles in our resolution. This matrix is called C_θ , where

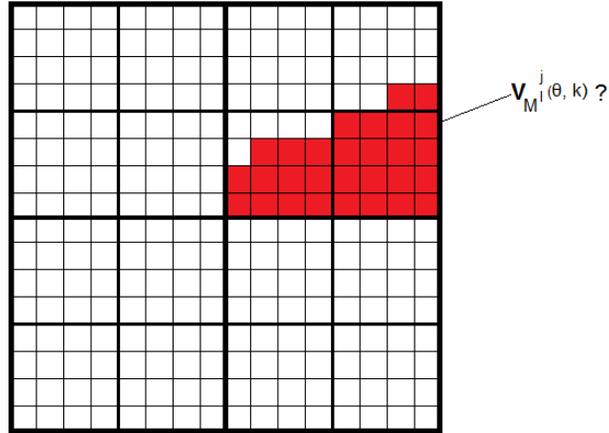
$$C_\theta(i, j) = \sum_{(m, n) \in W_{M_0^0}(\theta, \tilde{k})} I(m, n)$$

where \tilde{k} is the increment that allows $L(\theta, \tilde{k})$ to pass through the point (i, j) . This is just the cumulative moment matrix with angle *theta* that can be computed recursively from equations 2.4, 2.5 and 2.6.

With all of these auxiliary matrices defined, the task of calculating any wedgelet value given a fixed angle θ reduces to adding and subtracting values from C_V, C_H

and C_θ along the boundary of the wedgelet. Here is a pictorial example demonstrating how this process is carried out. We wish to calculate the moment along this specific wedgelet

Figure 2.59: An Unknown Wedgelet.



We then trace around the boundary attempting to isolate the values of this wedgelet.

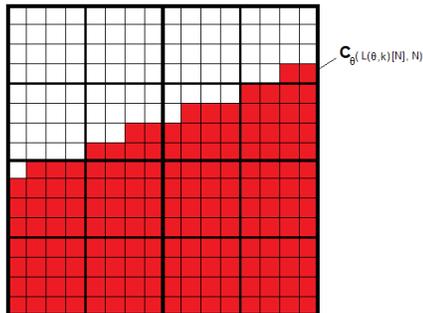


Figure 2.60: Step 2.

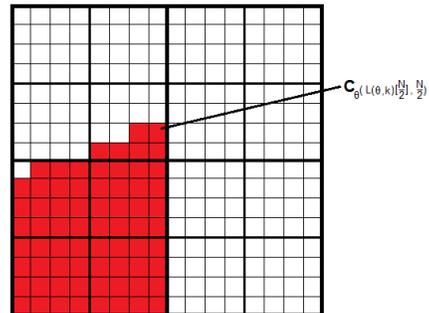


Figure 2.61: Step 3.

And finally we arrive at the line integral around the wedgelet.

The authors in [33] show that these auxiliary matrices speed the wedgelet transform process up by approximately a factor of 50-700 times over the popular

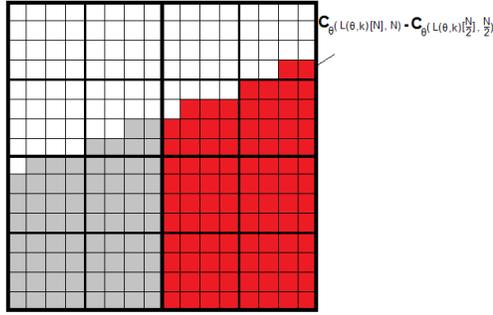


Figure 2.62: Step 4.

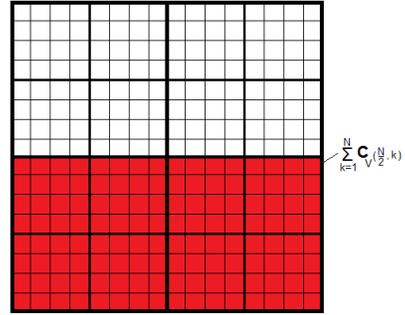


Figure 2.63: Step 5.

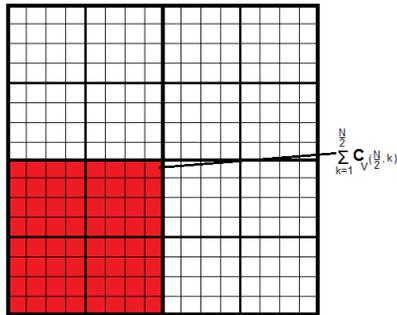


Figure 2.64: Step 6.

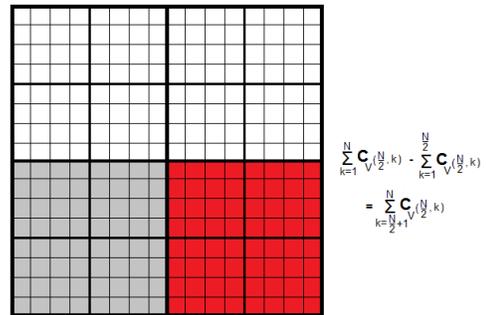


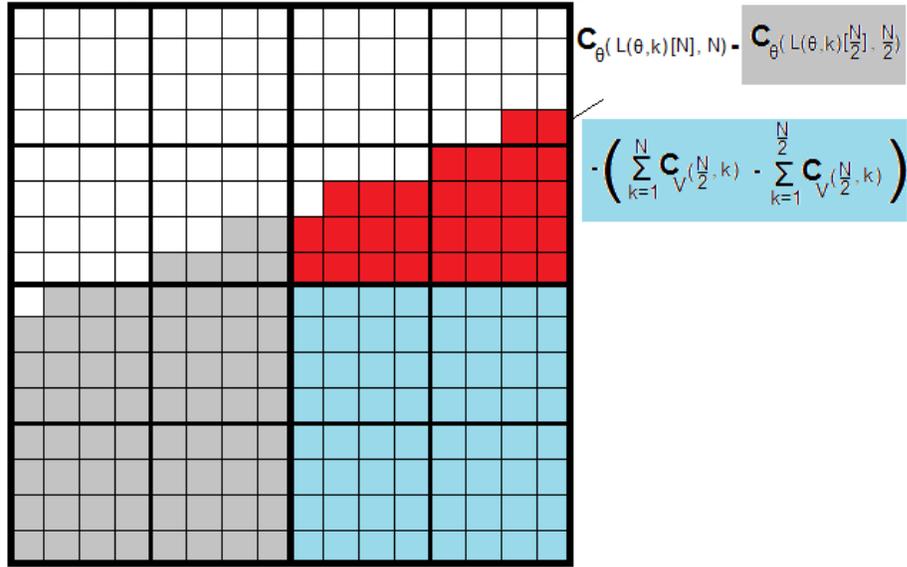
Figure 2.65: Step 7.

Beamlab implementation by Donoho [27]. Unfortunately, the ability to take advantage of this structure is lost when we iterate wedgelets once again and allow the values supported on the wedges to be planar.

2.4.6 Linear Wedgelets

Wedgelets provide a very elegant way of representing images that resemble the cartoon model. Since urban DEM data types so closely resemble these cartoon models it is no wonder that wedgelets can represent them with so few coefficients.

Figure 2.66: The Value of the Wedgelet is Calculated.



There are however a few drawbacks to the design that can be easily remedied with a very simple observation. Most structures in urban DEM have typical roof configurations. That is, most structures in urban DEM have flat rooves, rooves that have a constant gradient or a double gradient (A-frame). Piece-wise constant wedgelets have a very difficult time trying to efficiently encode structures with planar rooves.

The problem is that for a roof with a constant gradient, the average value of the wedgelet will always be the midpoint of the wedge supported on the roof. The error will decrease as the resolution becomes finer, which means that all things being equal, wedgelets will attempt to always reduce to finer resolutions to render the roof. This creates a tiling effect on such structures, and is very obvious to the eye. Aside from being an obstacle to reconstruction, it is an inefficient use of coefficients. Why describe a single large roof as a large collection of small tiles, when

Figure 2.67: A Typical Urban Roof.

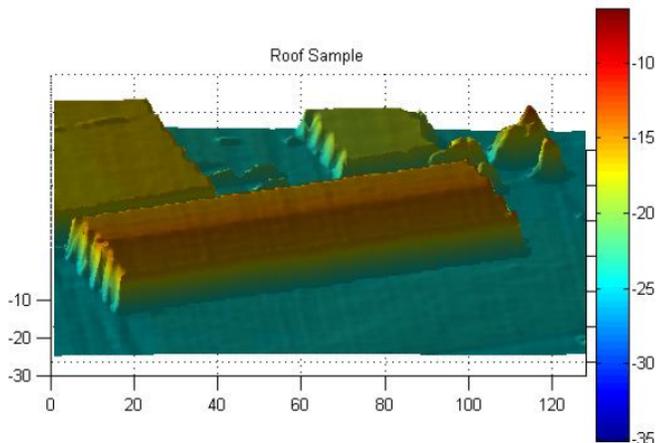
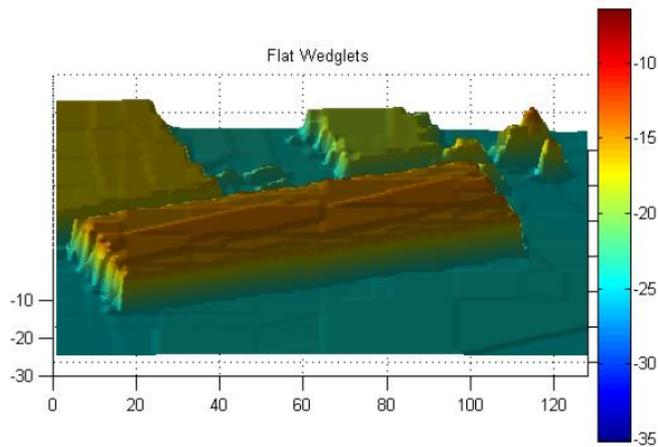


Figure 2.68: Urban Roof Approximation with Wedgelets.



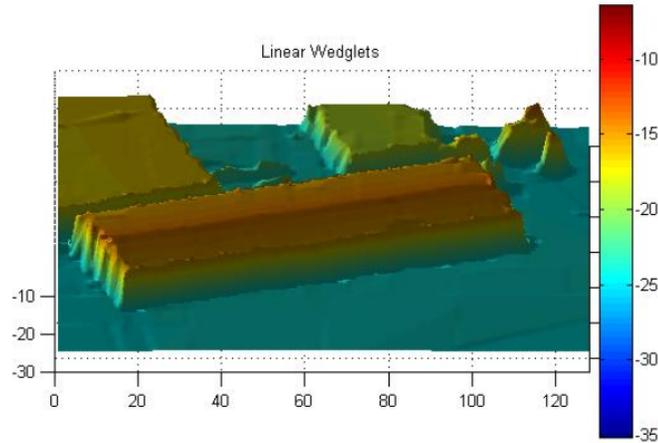
you could describe the roof in one large low-cost chunk of data in a more intrinsic setting? Here we will make a small modification of the wedgelet scheme that allows us to render these roof structures extremely well, at the cost of a more demanding data structure for storage and reconstruction and a more computationally intensive wedgelet transformation.

The idea is simple, the creation of the wedge structures remains unchanged, the only difference is that instead of calculating the value of the wedgelet function on those wedges as being the average value of the image on that wedge, we now apply a

least squares fit to find a plane that best fits the wedge. Note that a single wedgelet on a square S_l^j (both main and sister wedge) needs to store k, θ and the average values $V_M(k, \theta), V_S(k, \theta)$ to be able to reconstruct that wedgelet. The depth and location of the square S_l^j do not need to be stored since they are contained in the position of the quad-tree node. This means that a single wedgelet contains 2 integers and 2 real numbers for storage. Once we have passed to these linear wedgelets, each square will require the storage of k, θ as before, but now the main wedge and the sister wedge will require three real numbers each, to describe the plane that is supported on them. Our data storage has then increased by 4 real numbers per square that need to be stored for reconstruction. This seems to be a counterintuitive change, since our goal is one of compression. But as we shall see, the savings we reap in being able to coarsely represent structures with planar rooves more than makes up for the increase in data storage per wedgelet. There is another small fact that we need to be aware of when considering a change that raises the coefficient count, especially on the small wedgelet scale. In the constant wedgelet case, there is clearly no reason to represent pixels as wedgelets, since we are wasting energy trying to describe in 3 doubles what could be described in 1 double. In the constant wedgelet case, the two-by-twos sometimes benefit from being described as a wedgelet. If the two-by-two were constant or bi-valued, then the wedgelet can represent that two-by-two in 3 doubles as opposed to 4 doubles. However, this all changes in the linear wedgelets case. First, all two-by-two's can be perfectly described by linear wedgelets, assuming nothing strange is going on with the angular resolution. We must realize that this change has made it always inefficient to describe two-by-twos as a linear wedgelet,

since both storing the wedgelet and the pixels are perfect representations. However the linear wedgelets cost 7 doubles per square, and clearly we would be better off storing two-by-twos by pixel, if the parameter λ allows such fine resolution.

Figure 2.69: Urban Roof Approximation with Linear Wedgelets.



2.4.7 LIDAR Wedgelets

We generalized the linear wedgelet process from raster images to a LIDAR point cloud implementation. We have been dealing exclusively so far with the processed form of LIDAR, DEM, and we have had few opportunities to interact with this complex data type. Part of the apprehension with dealing directly with LIDAR is that the process of converting it to DEM is still a mystery. This makes it very difficult to analyze the results of compression, since we have no baseline comparison in the form of a gridded raster image. We possess DEM and LIDAR that physically overlap in geographic location, but either do not line up temporally or have been processed beyond our ability to replicate the data. Even with this inability to easily test our results, there were a few very strong reasons to implement wedgelets

for LIDAR. First, the process of wedgelets is itself a gridding mechanism, giving us another mechanism to try and understand the link between DEM and LIDAR. Second, the averaging effects of linear wedgelets tends to smooth out large wedge segments containing Gaussian or salt-and-pepper noise; LIDAR wedgelets have real noise suppression potential when λ is large. Third, it is reasonable to think of a gridded DEM as a very regular point cloud, allowing us to possibly speed up or improve our standard wedgelet process on urban DEM. The extension of wedgelets to LIDAR is a true generalization since the action on a DEM by the LIDAR wedgelets is equivalent to the standard linear process. This may also leave the door open for studying the effect of perturbations on the representation. The fourth reason for implementing a point cloud version of wedgelets is because the data structures and calculations needed to create such a program are very well suited and easily implemented in C++. Finally, by breaking free of the digital line, we are sometimes able to discern edges to a sub-pixel resolution.

Since comparing compression ratios and quality of reconstruction is so difficult when working with LIDAR, we will simply leave this section as a justification for making wedgelets work with LIDAR. It would be of great benefit to have some analog of TSSIM that can compare a raster version of LIDAR to the point cloud and make some kind of qualified assessment of quality. Unfortunately, since we have been relying on the correlation of TSSIM to the human visual system, it seems unlikely that such an easy fix will be possible. Another immediate benefit of the C++ implementation of LIDAR wedgelets is the tremendous speed-up by implementing efficient structures and using optimized linear algebra routines. The results that follow were

almost all created with the LIDAR wedgelets package, as it is the defacto method of choice due to its superior speed as compared to the Matlab implementation.

2.5 Experiments and Performance

The following four images (figures 2.70, 2.71, 2.72, 2.73) are reconstructions of the DEM tile S_{10} , using linear wedgelets with a ten angle resolution, an ℓ^1 norm for local wedgelet selection and varying the parameter $\lambda = 100, 50, 10, 1$. Generally with a heavily urban setting, a reconstruction using $\lambda = 1$ is indistinguishable from the original DEM. Note that the coarser reconstructions still look reasonable and in fact possess large TSSIM values, but there is the disturbing tendency for the edges of wedgelets that do not match up to appear looming like an eyesore in the reconstruction. One method we used to combat this mismatched edge problem was to introduce a local selection norm that balanced the overall local representation vs. how closely the edges of the wedgelet matched the altitudes in the original DEM. We attempted to accept a slightly inferior ℓ^1 norm locally, to have the obvious visual disturbance removed. Unfortunately, it is not so easy to trick the wedgelets algorithm into accepting wedgelets that the user desires. It seems that in some locations, wedgelets are doomed to poor reconstruction. These occur at corners or multiple edge intersections and also are prevalent when the layout of the scene aligns awkwardly with the less-than-flexible dyadic mesh. A particularly good example of this kind of misalignment with the decomposition grid occurs on the large building completely in the scene farthest north. The dyadic squares center around a nexus

of three edges, and the best wedgelet fit in that position is rarely pleasing to the eye unless λ is sufficiently small.

Figure 2.70: S_{10} with $\lambda = 100$.

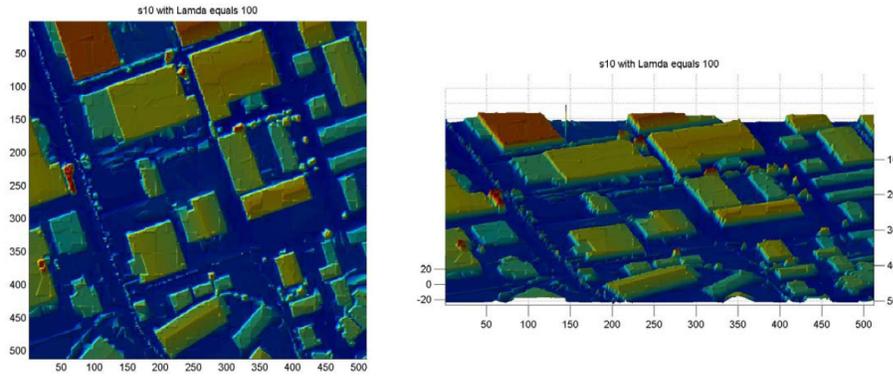


Figure 2.71: S_{10} with $\lambda = 50$.

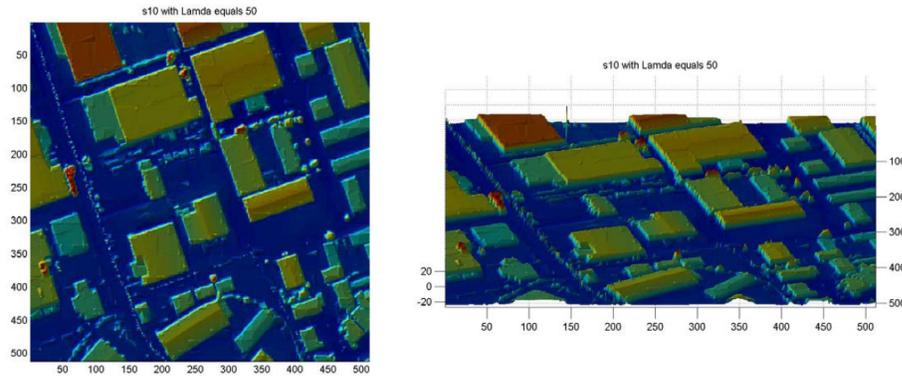


Figure 2.72: S_{10} with $\lambda = 10$.

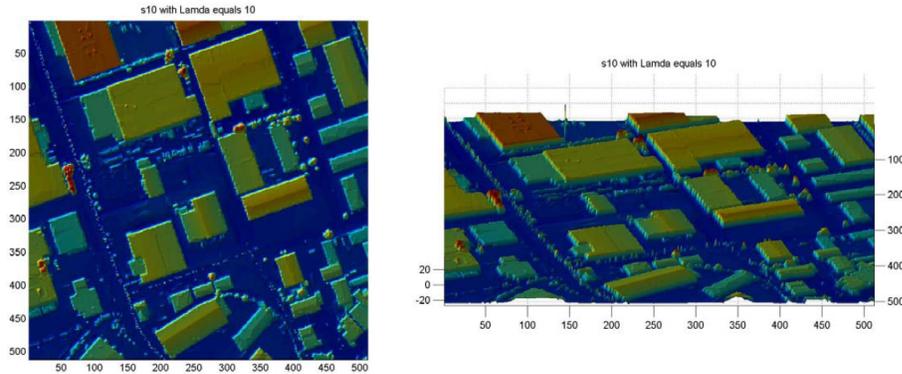
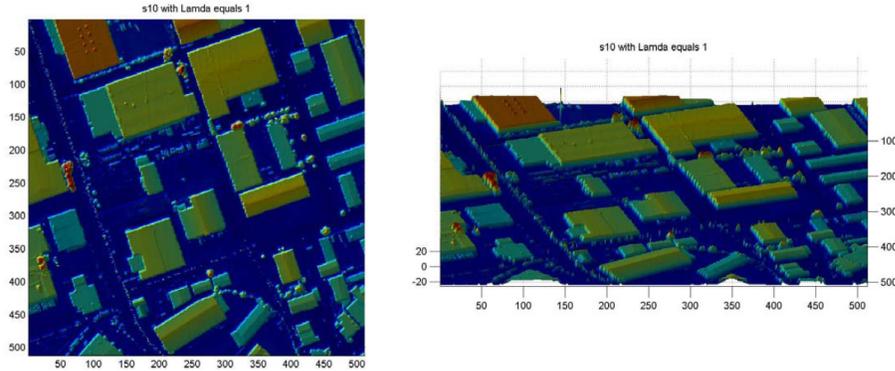


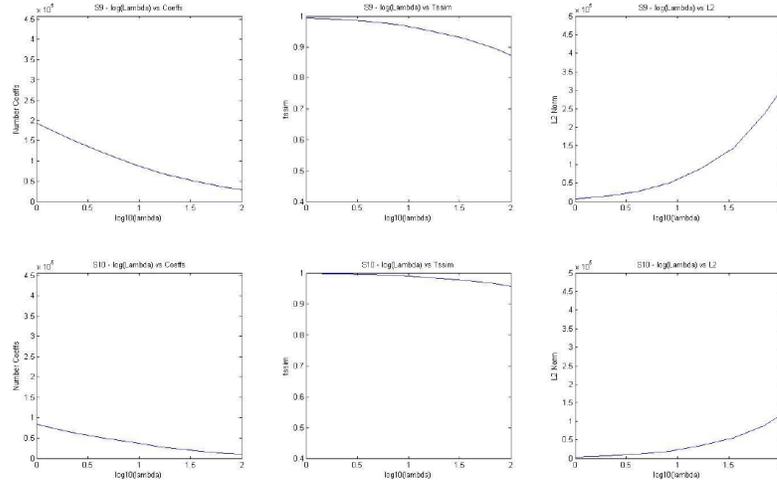
Figure 2.73: S_{10} with $\lambda = 1$.



This next group of six graphs (figure 2.74) attempts to compare important quantities for the wedgelet transform for the tiles S_9 and S_{10} . One of the frustrating factors of compressing a DEM via a parameter λ is the lack of precise control on compression ratios. In fact, λ tends to scale differently with various types of images, leaving a game of guess-and-test when it comes to compressing or reconstructing to specific levels. When studying the interaction of TSSIM to $\log_{10}(\lambda)$ we didn't see a consistent response that gave us a heuristic feel for compression levels. Because of this, we turned our focus away from graphs comparing TSSIM to some variant of λ , and instead compared the quality to the total number of coefficients. In this way we could compare the wedgelets procedure to other methods, like the time-scale methods in section **2.3**.

Now that we have fixed our method of comparison to be between TSSIM and the number or percentage of retained coefficients, we can compare our flat piecewise constant wedgelets to the linear wedgelets. The graph in figure 2.75 compares these two decomposition methods as the percentage of coefficients decrease and the image degrades. The graph used 10 angles for both methods, and local ℓ^1 norm for wedgelet

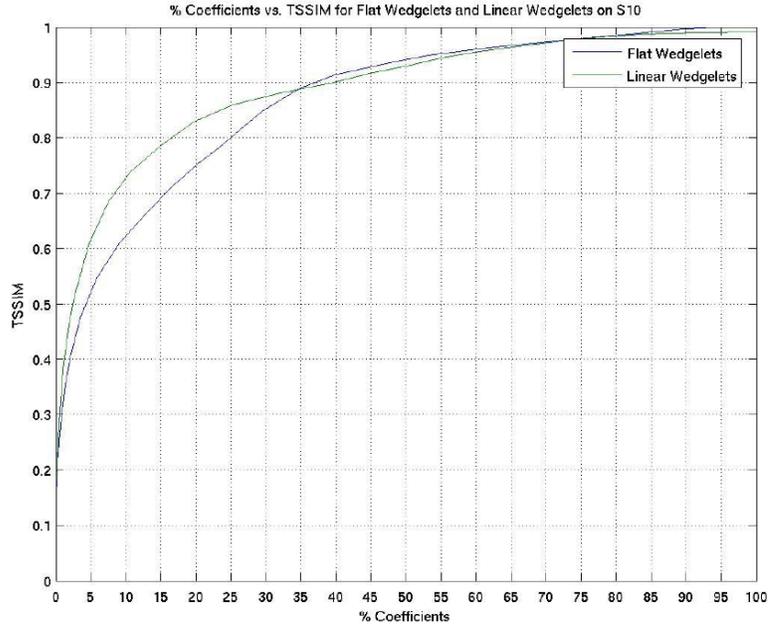
Figure 2.74: S_9 λ vs. Coeffs. vs. TSSIM.



selection. The response to other tiles in the S and T tile sets were not as tight as the curve is for S_{10} . Still, this points out why linear wedgelets are such a competitive compression scheme for urban DEM. Note that at 35% retained coefficients, the flat wedgelets overtake the linear wedgelets in terms of quality reconstruction. This is of course due to the small-scale advantage that flat wedgelets have over linear wedgelets.

In the section on time-scale representation methods for DEM, we discovered that contourlets level three decomposition was our best method for compression with high reconstruction TSSIM. We decided that a TSSIM of 0.8 or above is adequate, and contourlets achieved this with approximately 20% used coefficients. Clearly we are not interested in this region where flat wedgelets overtake the linear version. However, note that linear wedgelets cross the crucial 0.8 TSSIM mark with approximately 17% retained coefficients. This seems to indicate that linear wedgelets enjoys a healthy advantage with moderate sized wedges, when it can utilize its larger number of coefficients to more aptly describe large structures.

Figure 2.75: Flat Wedgelets vs. Linear Wedgelets.



The next two graphs in figures 2.76 and 2.77 illustrate, to some extent, the effects that angular resolution and local wedgelet selection norm have on the reconstruction TSSIM on both S_{10} and S_9 . There were six distinct wedgelet runs for each tile, where each run chose either 4 or 41 angles in its resolution, and each run chose a local ℓ^1 , ℓ^2 or total variation norm. Each of these configurations were degraded by changing the value of λ , and the graph records TSSIM versus retained coefficients. It is fairly difficult to see the graphs, but in the region of interest, the order of quality, while close, steadily gets better in quality from 4 angles ℓ^2 norm, 4 angles ℓ^1 , 41 angles ℓ^1 norm and finally the best performer being 41 angles with the local ℓ^1 wedgelet decision norm. This graph seems to indicate that greater angular resolution is more important than the choice of local wedgelet norm, but they are really so close it is hard to say for certain. All things being equal, we will choose the ℓ^1 norm as it seems to roundly outperform the ℓ^2 norm, especially for strongly

urban DEM. While the choice of norms does not cost us computationally, increasing the angular resolution by a factor of ten increases runtime by a factor of ten as well. The linear growth in computing time per angle makes it an unattractive parameter to adjust.

Figure 2.76: S_{10} Varied Angles and Norms.

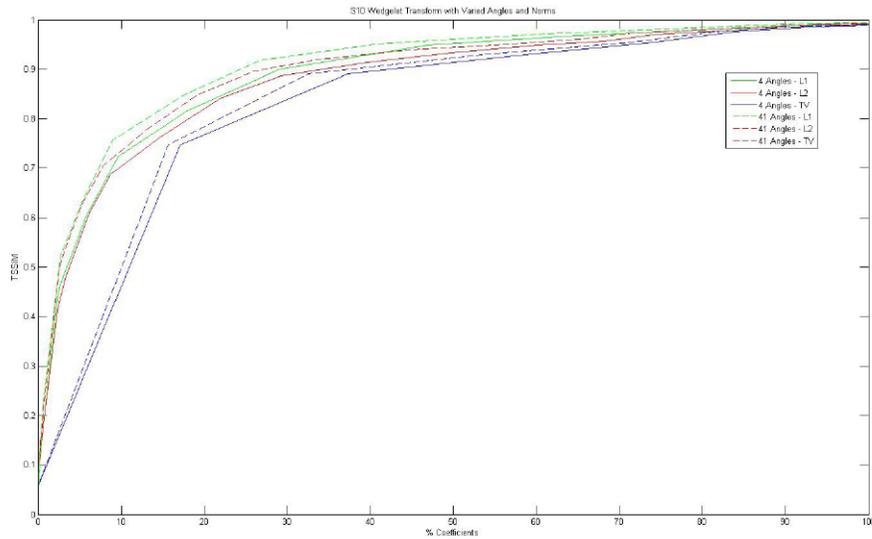
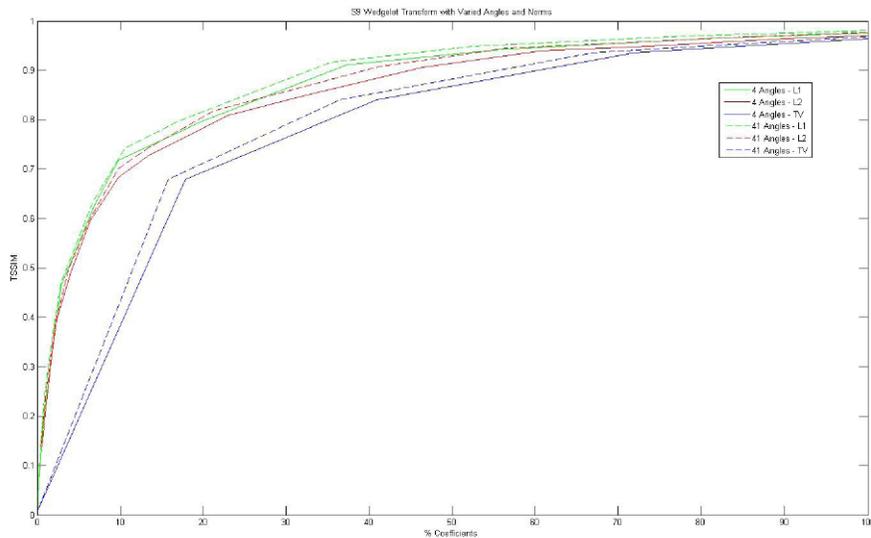


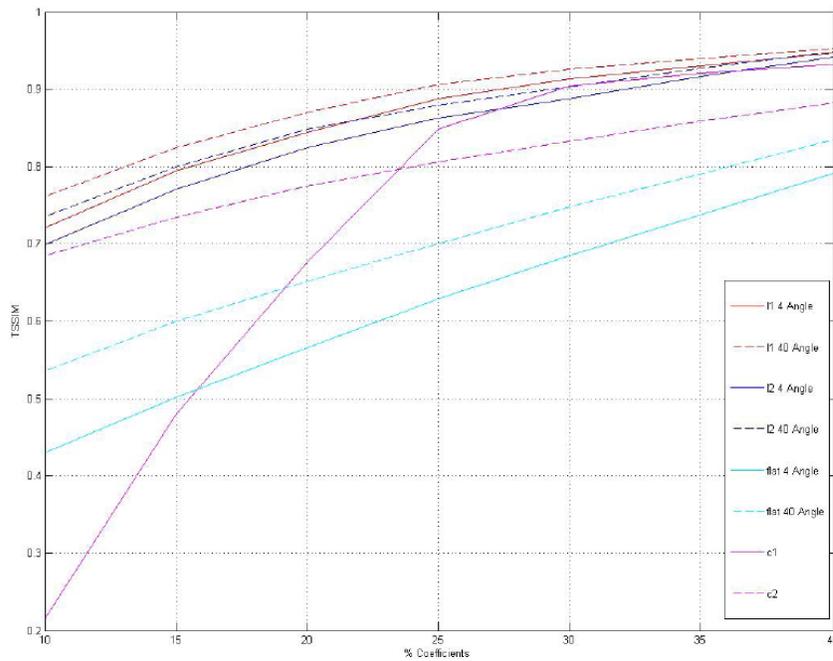
Figure 2.77: S_9 Varied Angles and Norms.



This graph, in figure 2.78, is a first comparison between various permutations

of the wedgelet algorithm compared to the contourlets compression. In the 15 to 25 percent retained coefficient range, linear wedgelets with 41 angles and a local ℓ^1 norm are clear winners in representing the tile: S_{10} . Note that in the region of interest, contourlets perform very well against flat wedgelets.

Figure 2.78: Contourlets vs. Wedgelets.



This image attempts to visualize the effect of angular resolution on visual reproduction of S_{10} . The image uses piecewise constant wedgelets, a local ℓ^2 norm and each retains 25% of the coefficients. While the defects do seem to smooth themselves out from upgrading the angular resolution from 4 angles to 40, the errors don't really go away as much as they don't appear as abrupt and blocky. This image drives home the inability of piecewise constant wedgelets to adequately describe typical urban scenes.

The following 8 images (figures 2.79 - 2.82) are meant to be directly compared

to one another, as to show how the difference in TSSIM from the above graphs, translate into visual quality. Note that in each of the next 6 images, 3 pairs of similar images, the methods being employed are reconstructed using 25% of the coefficients. The contourlets presents a marked visual enhancement of the piecewise constant wedgelets on S_{10} . However, as you progress further down, it is hard to tell that you are looking at a reconstruction at all. The linear wedgelets with 4 or 40 angles, with either norm do exceptionally well visually when retaining 25% of its coefficients. Just to prove how strong the visual effect is, especially compared to piecewise constant wedgelets or contourlets, the seventh image (figure 2.83) retains only 15% of its coefficients and is virtually indistinguishable from the original copy of S_{10} below it in figure 2.84.

Figure 2.79: S_{10} PW Constant Wedgelets 4/40 Angles.

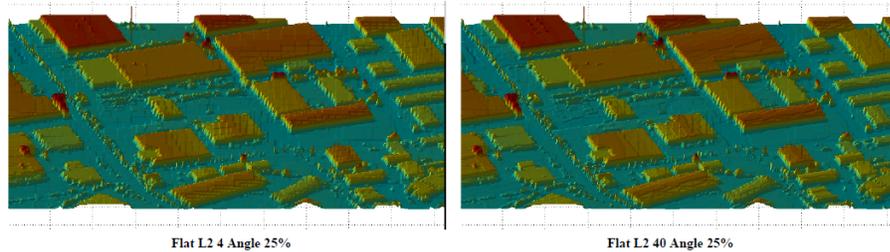
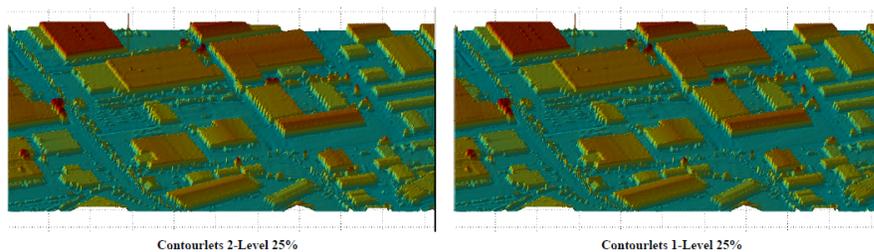


Figure 2.80: S_{10} Contourlets Level 1/2.



The next two graphs (figures 2.85 and 2.86) attempt to drive home the futility (in computing time) for drastically raising the number of analyzed Wedgelet angles.

Figure 2.81: S_{10} Linear Wedgelets 4/40 Angles.

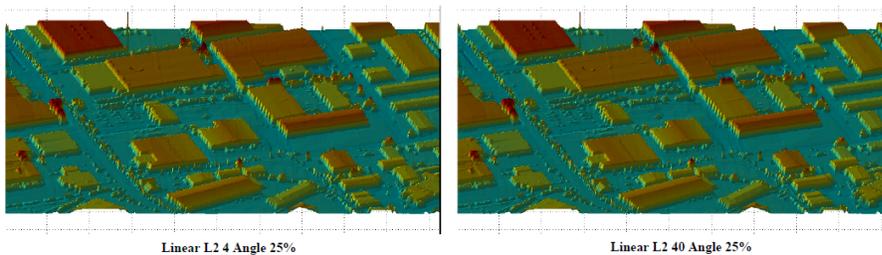


Figure 2.82: S_{10} Linear Wedgelets L1 Norm 4/40 Angles.

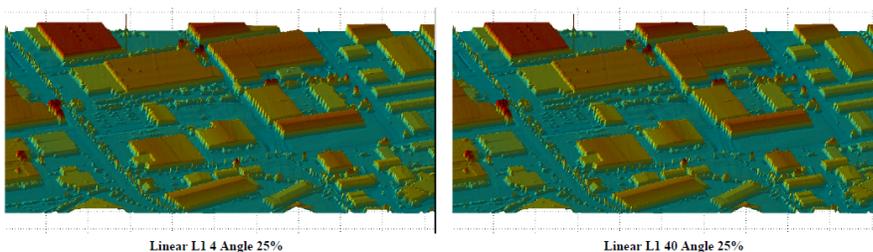
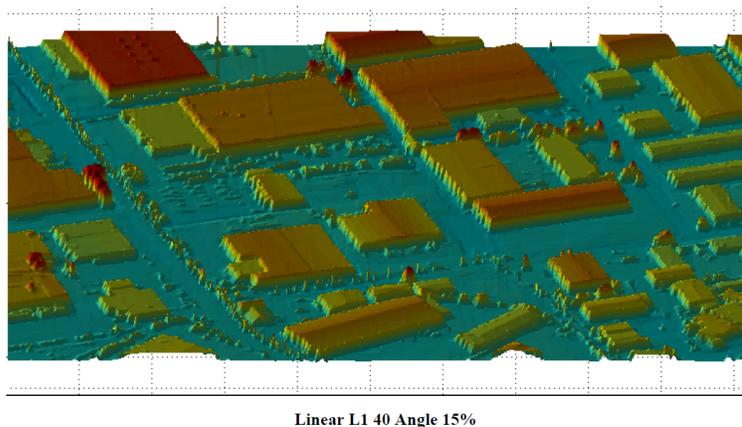
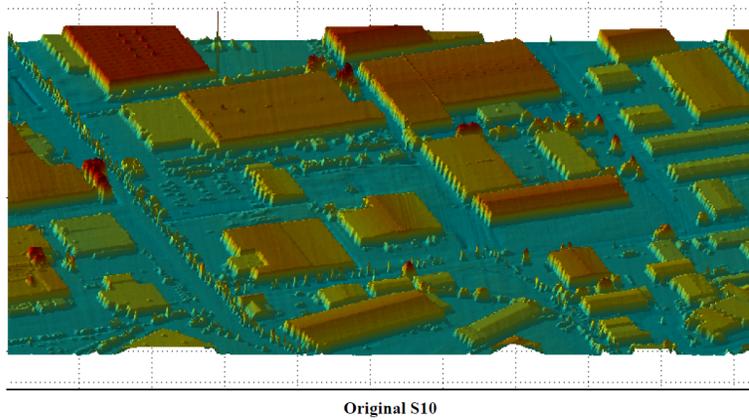


Figure 2.83: S_{10} Linear Wedgelets L1 Norm 40 Angles 15% Retained Coefficients.



Computing time scales linearly with the number of angles used in the decomposition; these graphs illustrate the gain in TSSIM for the two extremal cases for each of the tiles S_{10} and S_9 . The base case is a minimal collection of only two angles (horizontal and vertical), while the second uses 82 angles at each level of the decomposition. Note a very small gain in TSSIM/coefficient with a very large computing computing

Figure 2.84: S_{10} Original.



cost a factor of 41.

Figure 2.85: S_9 Extreme Angular Resolution Test.

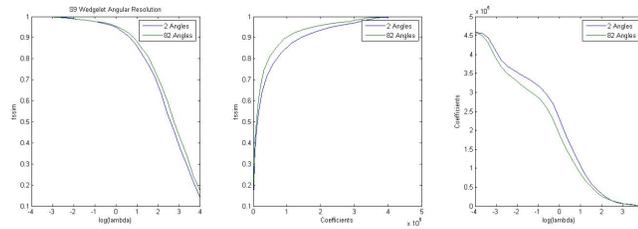
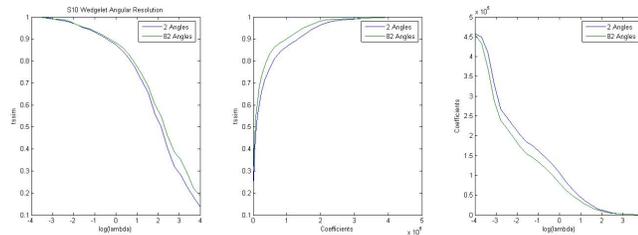


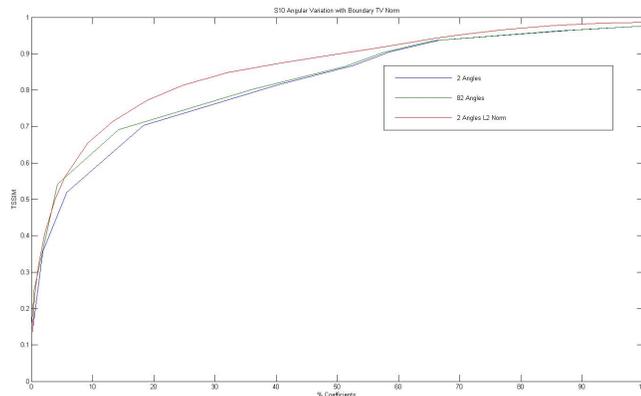
Figure 2.86: S_{10} Extreme Angular Resolution Test.



We noted that many defects in Wedgelet images, to our eyes, took place on the boundaries between dyadic squares. Even though TSSIM indicates quality reconstruction on these defects, our eyes pick them up readily. In an effort to combat this specific problem, a black box modification was implemented in Wedgelets to allow wedge decisions to be based on the particular wedge that minimizes what we call the

original boundary defect (essentially a line integral over the boundary of a dyadic square summing the difference in the original image and the new image). Here is a set of graphs (figures 2.87, 2.88 and 2.89) that display how the modified norm acts to subdue these visual defects. Two extremal angular conditions are used, namely 2 and 82 angles. In the percentage of retained coefficients vs. TSSIM graph, a curve representing S_{10} with 82 angles and a local ℓ^2 norm was added as a reference. Furthermore, two images are included that show the reconstructive differences between wedgelets with a local ℓ^2 norm and the modified boundary total variation norm (BTV) wedgelets, each with 82 angles. They share a similar number of coefficients, approximately 40,000, and have a comparable TSSIM, only differing by about 0.05. Something to note is how accurately the BTV method finds the break line on any slanted roof. Aside from this gimmick, the norm does not seem to be able to find the wedges that match edges with the underlying DEM.

Figure 2.87: S_{10} Angular Variation and Boundary TV-norm.



The next pair of graphs (figures 2.90 and 2.91) put the nail in the coffin for the contourlets versus wedgelets debate. One interesting element of the image is how

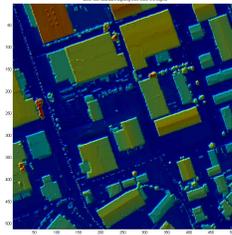
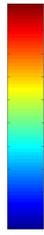
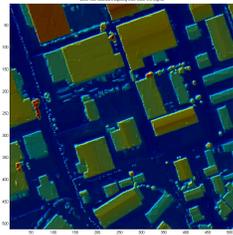
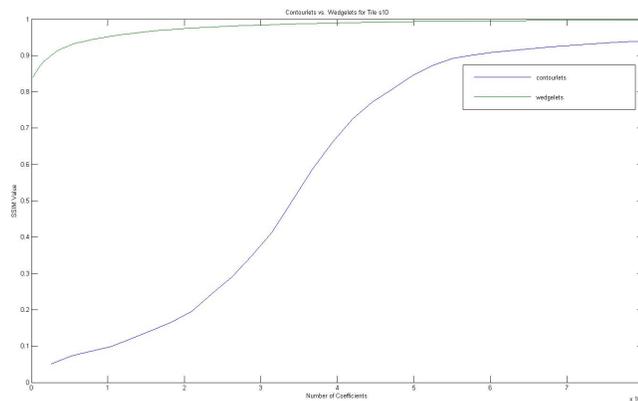


Figure 2.88: S_{10} Boundary TV-norm.

Figure 2.89: S_{10} ℓ^2 -norm.

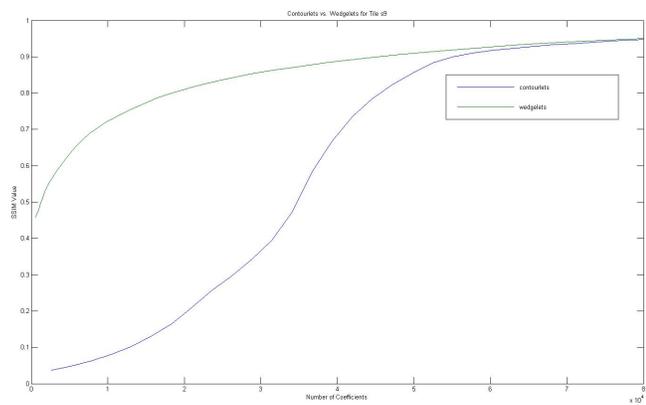
much of a performance hit wedgelets takes to a rural DEM, namely S_9 . Even with the performance degradation due to lack of cartoon-like structures, wedgelets still outperforms contourlets, visual efficacy per coefficient.

Figure 2.90: S_{10} Contourlets vs. Wedgelets.



This lengthy analysis of the wedgelet representation scheme, and its many incarnations has shown it to be a versatile and powerful method for a more optimal and intrinsic technique for reproducing the cartoon model of images. Since our urban DEM data types are similar to this simple kind of structure, and since our quality measure emphasizes the accurate reconstruction of that structure we have found a superior technique for the representation, compression and reconstruction of urban DEM.

Figure 2.91: S_9 Contourlets vs. Wedgelets.



Chapter 3

Dimension Reduction and Material Classification

3.1 Introduction

The problem of unwieldy data is one shared amongst many of the fields that collect and analyze various sources of information. The rapid expansion of computational power and massive data storage has engendered within us a ravenous need to expand our sets, increase resolutions and add on other complexities into our already burgeoning data. This problem is exacerbated in the case of high dimensional data, being more sensitive to the so-called Hughes effect [47]. The popularity of recent compressed sensing techniques are a nod to the need to correct for the over-collection of information. We are simply drowning in this overflow of information, and often, this excess acts to obfuscate the underlying structures in the data.

In many applications of machine learning, data mining and image processing, high dimensional data is collected. Examples of such data sets could be composed of global climate information, stellar or terrestrial spectra or even human genome distributions. Although the collected data is high dimensional, often it is the case that the data is intrinsically low dimensional. The following examples are taken from Maggioni [45] and are excellent illustrations of the power of dimension reduction techniques.

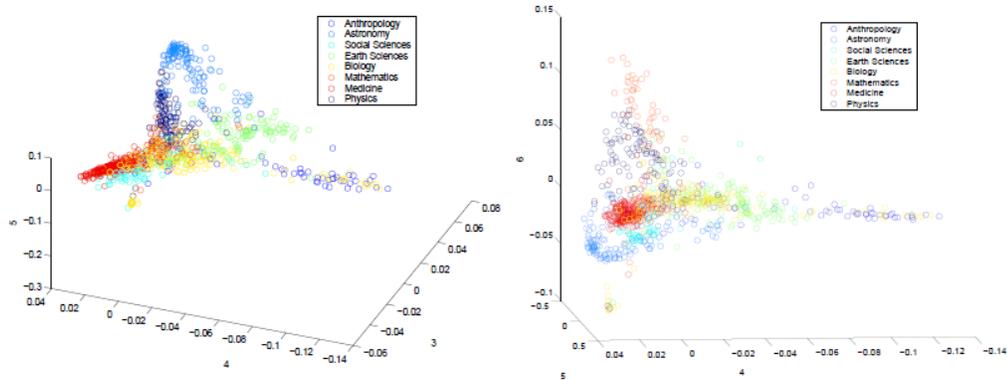
Example: Consider a data cloud consisting of 1047 scholarly articles taken from Science News in some relatively small timeframe. We will collect a dictionary of 2036 relevant words to the articles appearing in this cloud. We order these words in any manner, and then we assign each of the 1047 articles to a vector $x_k \in \mathbb{R}^{2036}$, where the j 'th position of the k 'th vector is the frequency in which word j appears in article k .

We assign each of the vectors to one of the eight classes: Anthropology, Astronomy, Social Sciences, Earth Sciences, Biology, Mathematics, Medicine and Physics. We assign these based on the classification under which the articles were published. We then have $\{x_k\}_{k=1}^{1047} \subset \mathbb{R}^{2036}$, where each x_k is assigned a unique class. As long as the fields in which the papers are published are relatively separated (that is, not electrical engineering and applied mathematics), we would expect the jargon differences would separate the papers in this high dimensional space. Clearly there is a lot of structure in this data set, but the dimension of the ambient space is extremely large for the kind of data and number of points.

Maggioni reduces the dimension of this set with a Laplacian kernel method and then embeds the set into six-dimensions. To visualize the projection, he then constructs two plots, one with coordinates 3, 4, 5 represented as x, y, z , and the other with coordinates 4, 5, 6 represented as x, y, z . The reduced dimension points are colored based on the class from which they derive. These plots show that there is clearly strong geometry connecting similar papers and differentiating papers that are dissimilar. The idea is that the embedding preserved all of the structure from the large dimensional data set, in fact, possibly enhancing the local structures by

embedding it on an underlying manifold.

Figure 3.1: Article Data Projected onto \mathbb{R}^3 : Maggioni.

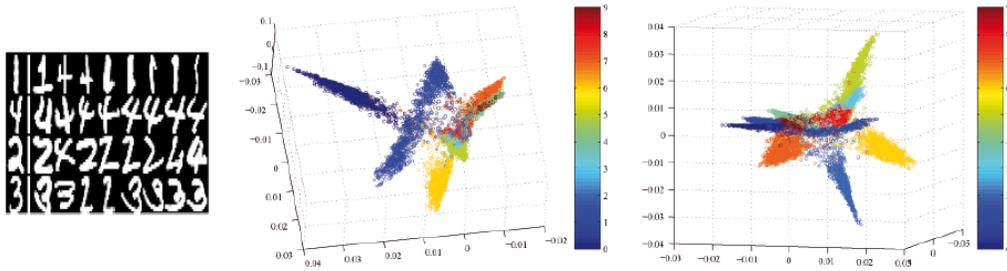


Example: The next example that Maggioni uses to illustrate the uses of dimension reduction comes from a post office data set. The post office has much interest in being able to identify hand-written numbers and letters, with all of the variations that humans can produce. The data set consists of 60,000 28×28 gray-scale images of the digits 1 through 9. That is, we have $\{x_k\}_{k=1}^{60,000} \subset \mathbb{R}^{28^2} = \mathbb{R}^{784}$, where each vector is an unfolded image. We can similarly assign each vector to one of the classes $\{1, 2, \dots, 9\}$. The included image shows a sample of the data set.

The data has its dimension reduced, and then 3 of the remaining coordinates are picked out to visualize the structure in each of the plots. This example shows a tremendous amount of underlying structure; structure that is only enhanced by the reduction of dimension. Such a system could be a very rough form of character recognition, but the important aspect of these examples is that the structure not only survived the embedding, but appears as very coherent.

These examples illustrate the goals of dimension reduction quite admirably.

Figure 3.2: Hand-Written Numeral Data Projected onto \mathbb{R}^3 : Maggioni.



Namely, the desire to embed a high dimensional data set onto a much lower dimensional intrinsic manifold, while retaining the pertinent local geometry such as neighborhoods, tangent spaces and geodesics, see [5] [6]. In the words of C. Burges [12], “Apart from teaching us about the data, dimension reduction can lead to better models of inference.”

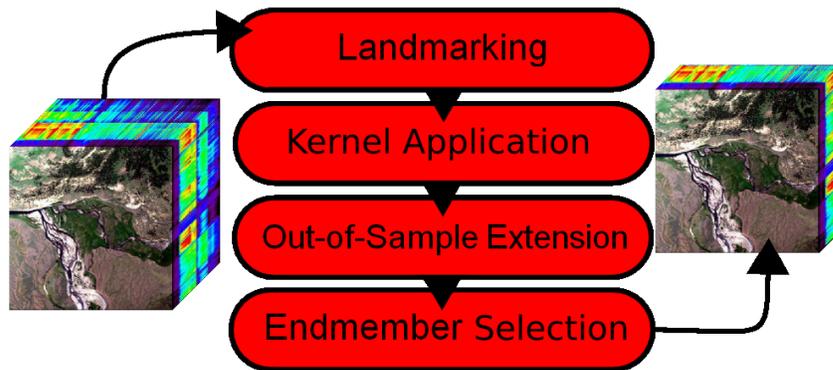
Modern dimension reduction invariably starts with Principal Component Analysis (PCA), but this is by no means the first incarnation of embedding methods. The theory has beautiful classical roots that this thesis will not have the space to document them. PCA will be our baseline linear reduction technique, but our real tools will be the non-linear kernel methods Laplacian Eigenmaps (LE) and Locally Linear Embedding (LLE). Typical kernel methods of dimension reduction consist of representing the data and local geometry within a kernel and approximating that kernel with a low-rank operator found by truncating its spectral series representation.

This chapter is specifically focused on the dimension reduction of terrestrial hyperspectral data, where the purpose of the reduction is to aid in automatic material classification. With this narrow scope in mind, we can see the large importance

we place on the preservation of local geometry through the embedding process, since classification cannot be performed when dissimilar materials bleed into one another’s spectral signatures. We will approach this problem from two distinct perspectives.

The first set of ideas comprises a so-called dimension reduction pipeline. This involves implementing a series of modular stages giving a flexible dimension reduction framework. These stages can be loosely categorized as Landmarking, Kernel application, Out of sample extension (inversion of the landmarking process) and finally the creation of an endmember frame and projection of the data onto that frame.

Figure 3.3: Frame-Based Kernel Dimension Reduction Pipeline.



The landmarking procedure drastically lowers the computational cost of expensive processes like nearest neighbor generation and eigenvalue decomposition of large data sets. This is essentially done by sampling the data and treating the sample like a full data cube for dimension reduction, with a cost in accuracy that tends to be acceptable. The out-of-sample extension essentially inverts this process by extrapolating the dimension reduction to those points that were not sampled. Our

kernel methods for this pipeline are either Laplacian Eigenmaps or Locally Linear Embedding. This stage consists of forming a graph from the sampled data, constructing a kernel that represents the local geometry and then using a truncated spectral change of coordinates to reduce the dimension of the points in the sample. The above steps are all well-known and are only unique in the order in which they are applied. While this portion of the process is vital for preparing the data for classification, the novel part of this pipeline is the introduction of frames to replace traditional endmember constructions. The flexibility that frames allow with the ability to design elements, and the added redundancy of the representations are extremely powerful when finally passing to the classification stage of the process.

The second portion of this chapter will briefly develop a technique that couples wavelet packets with kernel dimension reduction methods to create a process that is sensitive to both spectral and spatial information. Since our data corresponds to physical geographic locations, the individual locations contain important local information that is not otherwise exploited by dimension reduction techniques. This can be seen clearly by observing that if a certain location can be identified as a material, like a road, that is typically contiguous, the neighboring spectral positions (pixels) are more likely to also be road than is a random point in the scene.

This chapter shall be broken down into three sections. Section one shall discuss our data type: hyperspectral data. Section two shall develop the frame-based kernel dimension reduction pipeline and conclude that this process outperforms the corresponding base level technique on its own with respect to material classification. Section three shall develop the spectral-spatial methodology, and it will be shown

that the hybrid method with PCA outperforms PCA on its own.

3.2 Hyperspectral Data

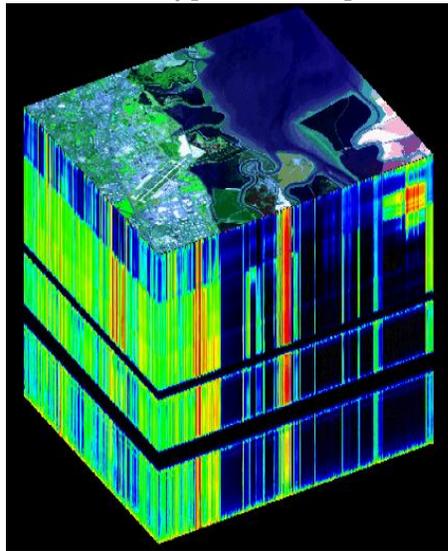
No one seems to be quite sure what the precise distinction is between a multi-spectral data set and a hyper-spectral set. Spectral information has become a ubiquitous member of most government labs and applied harmonic analysis departments all around. That being said, hyperspectral data is one of the rarer forms that this information can take. Most organizations that acquire this type of data do so at tremendous cost, and they tend to keep their data close to the chest. As the public's access to satellites increases and sensor costs decrease, we find an increasing number of spectral bands in our repertoire.

Hyperspectral data is still better described than rigorously defined. Generally speaking, hyperspectral data is a large collection of spectral reflectance information for a single geographic (or in some cases, geological core samples) region. In its most idealized state, hyperspectral data is just the collection of reflectance energy that returns from an exposure to a specific frequency of light. An oversimplified version of the truth is that hyperspectral data is collected by satellites or airplanes that can filter or process multiple separate wavelengths of light at the same time. The data typically takes the form of a collection of what look like black and white images that compose a so-called hypercube. In reality, the geographic regions in each image are the same, but the intensity of a pixel in any particular image corresponds directly to that location's spectral response to a frequency represented by that band. In this

sense, we can think of a single pixel as representing a sampling of the spectra of some material near that pixels location. It is the hyperspectral data that uses a significant number of samples for each pixel.

In reality, hyperspectral data exists for pretty much any application you could want. Geologists use it to study the mineral composition of far away regions, or to study small-scale formations in core samples. Our greatest space telescopes collect dense spectral information all the time, to peer into the mysteries of the universe. In this chapter, hyperspectral data will explicitly refer to hypercubes of spectral reflectance information. A somewhat overused example of a hypercube exhibits the three dimensional character of the data.

Figure 3.4: The Prototypical Example of a Hypercube.



The most important properties of hyperspectral data are their narrow and contiguous measurements, being spectrally over determined and their utility in target detection, material classification and surface mapping with material identification. Two very important and often used hyperspectral data sets are referred to as the

Urban and Smith Island data set respectively.

Figure 3.5: True Color Image of the Urban Data Set.



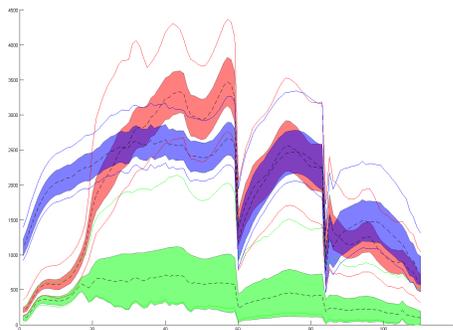
Figure 3.6: True Color Image of Smith Island.



These sets are well known and often used for good reason, they are some of the few collections of data that are not highly classified and restricted from public study. These will be our primary test data sets, and many of our results will be based on these hyperspectral sets, since others are not allowed to be published in this thesis. The Urban data set is 307 by 307 pixels and consists of roughly 161 spectral bands. Note the large prominent Walmart near the center of the image. If

you look closely in the western portion of the image, you can make out pixel/sub-pixel swimming pools which will crop up later in the chapter. The Smith data set [2] is 679 by 944 pixels, and contains 126 spectral bands. If you look really closely in the watery portion of the western half of the island, for several spectral bands you can make out sub-sea level structures below the water. These are the elusive targets that we shall forever endeavor to discover. Figure 3.7 is a statistical envelope for three spectral classes within the Smith Island dataset. The classes represented are water, sand and flora, the dotted line represents an average class spectra and the filled in portion represents a standard deviation in each direction from the mean.

Figure 3.7: Statistical Spectral Envelopes.



Here are a couple of examples (Figures 3.8 and 3.9) of spectral bands from the Urban set. Note that the true color images are acquired from taking bands closest to red, green and blue and put their reflectance together for a color image. This shows that the spectral coverage of the data sits well within the visual wavelengths of light.

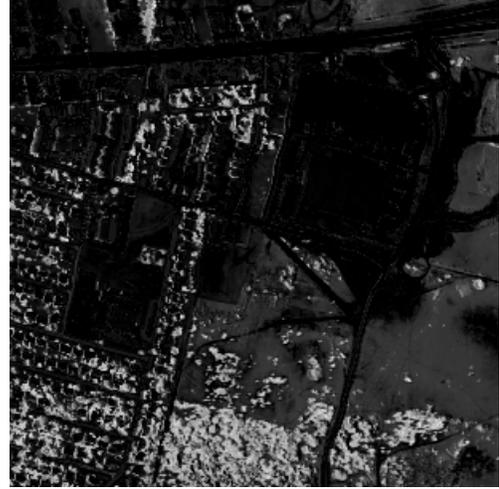
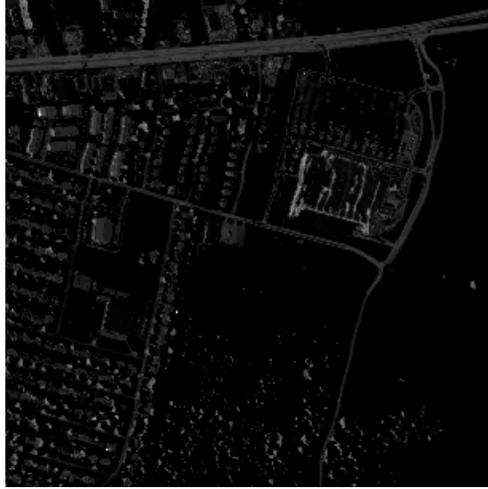


Figure 3.8: Urban Tile Spectral Band. Figure 3.9: Urban Tile Spectral Band.

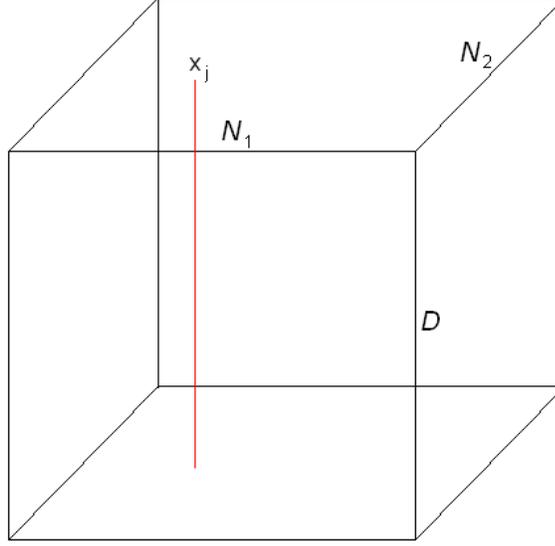
3.3 Kernel-Based Frame DR Methods: The Pipeline

We start with a hypercube, X , that is, a real valued $N_1 \times N_2 \times D$ dimensions. Since our data will almost always contain hyperspectral information, we can think of this cube as D -spectral bands, where each of the bands is $N_1 \times N_2$ representing the same geographical region of space, where a pixel in each band represents some reflectance of that bands wavelength from the region the pixel represents.

We can think of this as a regular sampling of some manifold M , which has an unknown underlying dimension $d \ll D$. The consensus is not yet in on whether this is a helpful perspective to take, concerning an underlying manifold even existing. For instance, clearly the data in X can be embedded into infinitely many manifolds, or weirder yet, into some one-dimensional curve in space by essentially connecting the dots.

This frame of reference is further confused upon the realization that spectral sampling may itself not be possible since the underlying surface, i.e., the surface of

Figure 3.10: Typical Hyperspectral Data Cube Point $x_j \in \mathbb{R}^D$.



the Earth, may in fact be discrete, assuming some smallest form of matter.

Even with these questions looming about the veracity of underlying manifolds existing for hyperspectral data, we will accept that the mathematics are simpler assuming so, and for us, X will be a sampling from a lower dimensional manifold. This allows us to use analytical results about graph Laplacians by exploiting the underlying structure of the manifold.

We unfold our data cube into a list of points $X_{data} = \{x_k\}_{k=1}^{N_1N_2} \subset \mathbb{R}^D$, where the points x_k are D -dimensional, and are columns in the $D \times N_1N_2$ matrix X_{data} .

We take this collection of points and from it create a directed graph, G , using one of the following criteria:

1. k -nearest neighbors. We choose some $1 < k \ll N = N_1N_2$ and generate the set of neighbors to the point x_j , $\Omega_{x_j}^k$ where we define

$$y_1^{x_j} = \operatorname{argmin}_{x \in X \setminus \{x_j\}} \|x - x_j\|$$

has $y_1^{x_j} \in \Omega_{x_j}^k$, and recursively,

$$y_l^{x_j} = \operatorname{argmin}_{x \in X \setminus \{x_j, y_1^{x_j}, \dots, y_{l-1}^{x_j}\}} \|x - x_j\|.$$

Then $\Omega_{x_j}^k = \{y_1^{x_j}, \dots, y_k^{x_j}\}$. In this case, for $x \in G$, $x \rightsquigarrow y$ iff $x \in \Omega_k^y$.

2. ϵ -neighbors. Choose an $\epsilon > 0$. In this simple case, we assert that $x \rightsquigarrow y$ iff $x \in B_\epsilon(y)$.

The choice of neighborhoods for our graph is of great importance. Note that ϵ -neighborhoods will generate non-directed graphs, while k -nearest neighbors can have asymmetries. Laplacians contain a great deal of localized information, essentially characterizing local geometry within the data. With this perspective it is easy to see that by choosing our local neighborhoods arbitrarily and without concern for the underlying geometry of the data, our constructed graph Laplacian will not be as useful and a host of undesirable features can crop up into our results.

This is a relatively deep problem, about detecting local geometry in discrete collections of points and is very much still a complicated issue surrounding the creation of geometrically sensitive operators on graphs. Consider the following simple example.

Here the two-dimensional data in X , in Figure 3.11, is much like what raw data typically looks like; a collection of points with no obvious geometry. Figure

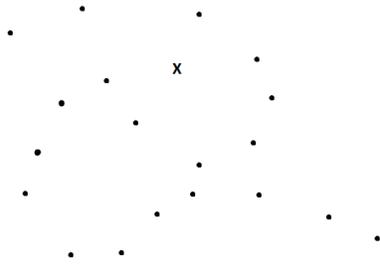


Figure 3.11: Data Points $X \subset \mathbb{R}^2$.

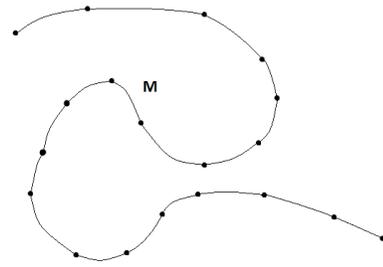
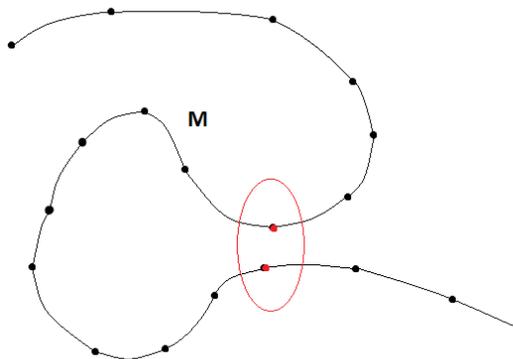


Figure 3.12: Manifold $X \subset M$.

3.12 could be considered the underlying structure of the data, lying on the one-dimensional manifold M . Without knowing M and with only clues provided by the data itself, how do you choose quality neighborhoods that at least approximate the structure of M ? Note that it is possible for this set of data and underlying manifold, to choose an ϵ or k for which the following two data points (Figure 3.13) are considered neighbors.

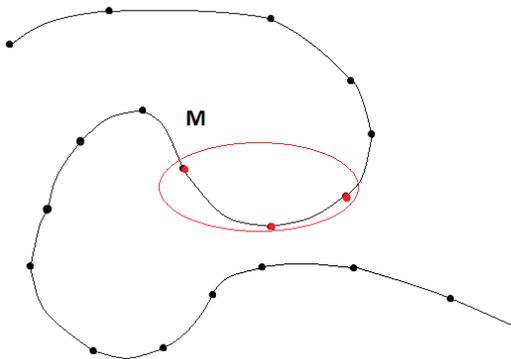
Figure 3.13: A Poorly Chosen Neighborhood.



In fact, any larger ϵ or k will provide the same ‘bad neighbor’ as in Figure 3.13. This is how k -nearest neighbors and ϵ -neighborhoods can fail, since they are really coarse approximations of the underlying local geometry. Ideally, you would like to

find these local neighborhoods by some geodesic distance function, which somehow was able to know the underlying structure. The ideal neighborhood for the point in question would probably look like that in Figure 3.14.

Figure 3.14: A Well Chosen Neighborhood.



There is no reason to think that choosing our neighborhoods in such a clumsy manner could return elegant results, but it turns out that often local Euclidean neighborhoods are enough to encode the important geometry of the data into the Laplacian. While there is ongoing research into finding more intrinsic neighborhoods within data [56], by choosing ϵ or k sufficiently large (at the cost of extra computational expense) we can gather neighborhoods that contain the relevant geometry with a small remainder of non-relevant neighbors, as in Figures 3.15 and 3.16.

For the purposes of this thesis, these neighborhoods will be sufficient.

3.3.1 The Laplacian-Eigenmaps Kernel

Starting with the graph we generated in the previous section, (G, E) , and continuing with the guidelines established in [5] [6], choose $\sigma > 0$ and we define the

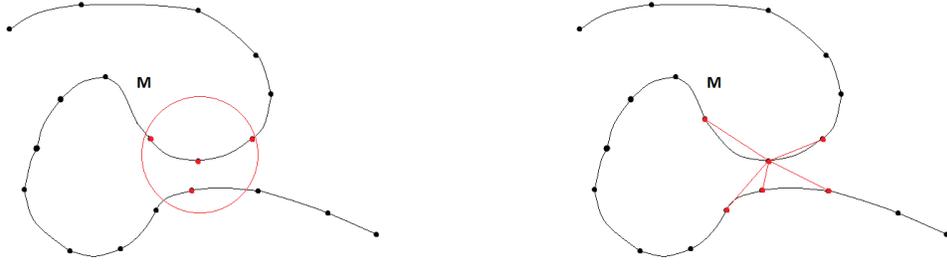


Figure 3.15: A Decent ϵ -Neighborhood. Figure 3.16: A Decent k -Neighborhood.

Laplacian Eigenmaps weights between connected vertices to be

$$W(x_i, x_j) = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{\sigma}} & x_i \rightsquigarrow x_j \\ 0 & \text{otherwise} \end{cases}$$

Next we form the degree matrix, $D = (d_{i,j})$ by

$$d_{i,j} = \begin{cases} 0 & i \neq j \\ \sum_{k=1}^{N=N_1 N_2} W(x_j, x_k) & i = j \end{cases}$$

The graph Laplacian can then be defined as

$$L = D - W.$$

3.3.2 Locally Linear Embedding

Roweis and Saul in [49] developed a nonlinear kernel which attempts to follow the following general steps, which seem to always find a voice with nonlinear phenomenon:

1. Stick to not-too-nonlinear structures,
2. Somehow decompose nonlinear structures into linear subspaces,
3. Generalize the eigenvalue problem of minimizing distortion.

Their method for constructing a kernel consists of forming a graph as in the general case, and finding k -nearest neighbors. From there, the process differs significantly from the Laplacian Eigenmaps case. The next step is also to compute a set of weights, and in this case the weights must satisfy, minimizing the residual sum of squares when reconstructing each point from its neighbors,

$$RSS(w) = \sum_{i=1}^n \left\| x_i - \sum_{j \neq i} w_{i,j} x_j \right\|_{\ell^2}^2 \quad (3.1)$$

subject to the constraints that $w_{i,j} = 0$ unless j is one of i 's neighbors, and having a unit sum $\sum_j w_{i,j} = 1$ for each i . Solving for the new coordinates from the weights then involves minimizing the functional

$$\Phi(Y) = \sum_{i=1}^n \left\| y_i - \sum_{j \neq i} w_{i,j} y_j \right\|_{\ell^2}^2 \quad (3.2)$$

subject to the constraints $\sum_i Y_{i,j} = 0$ for each j and $Y^T Y = I$.

Luckily enough, solving this problem really reduces to the construction of the kernel

$$K = (I - w)^T (I - w), \quad (3.3)$$

and then follows the same procedure in the case of the Laplacian Eigenmaps kernel in order to change the coordinates to the lower dimensional subspace. This is a convenient change, as it really allows for our pipeline to have a modular assortment of reduction kernels.

3.3.3 Dimension Reduction by Change of Coordinates

We have now looked at a couple of examples of how to transform a hypercube X into an operator that faithfully encodes much of the local geometry from the data. We now wish to change the coordinates of our data into those of a lower dimensional space, for which we have preserved many important geometric quarks of the data while projecting it onto a close approximation of its underlying manifold.

Specifically, we are looking for

$$\rho : X \subset \mathbb{R}^D \rightarrow Y \subset \mathbb{R}^d,$$

where we think of \mathbb{R}^d as being the coordinate space of the underlying manifold. Our method of extracting this change of coordinates will involve the spectral theorem and only retaining d of the highest energy spectral directions. These are by no means the only possible coordinates in which the data can be represented. In some sense, reducing the dimension of our data by merely lopping off lower energy spectral directions, from the kernel K , is both not elegant and throws away information that could contain vital details about the data's geometry. If the eigenvalues of the Kernel decay rapidly after some number, then we can say with confidence that

those largest eigenvectors span almost all of the space, at least the most important subspace. The hope is that with data that really does exist on a low-dimensional manifold, and with proper neighborhood selection, the eigenvalues of the Kernel will decay in such a way that we can estimate the dimension of the underlying manifold and project our data onto that space with great accuracy. The reason that we use this particular change of coordinates, is that, at least in the case with Laplacian Eigenmaps, we have convergence results of this kernel to the actual graph Laplacian in probability, given very mild conditions on the graph [5] [6].

We decompose K into its eigen-decomposition,

$$\{(\lambda_i, v_i)\}_{i=1}^N \text{ where } \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$$

We retain the d -eigenvectors corresponding to the largest eigenvalues, and our change our coordinates is given by

$$y_k = \rho(x_k) = (v_1[k], v_2[k], \dots, v_d[k]),$$

where $v_j[k]$ is the k 'th coordinate of the eigenvector corresponding to the j 'th largest eigenvalue.

3.3.4 Landmarking

Kernel methods for dimension reduction are elegant and powerful. However, they inherit a fate similar to their data: information overload. Storage of hyperspectral data is cumbersome, even with coarse resolution and sparse spectral sampling,

the amount of information is simply too great to be processed or stored indefinitely.

The process of creating a kernel for dimension reduction and then reducing the dimension involves a massive computational load, scaling poorly with the spectral bands D and even worse in the geographical bounds N_1 and N_2 . The main bottlenecks in the processing are the neighborhood selection methods ($k^2 \log k$) and finally in the eigenvalue problem. While our kernel is hopefully quite sparse, by choosing small enough neighborhoods, the eigenvalue problem still scales very poorly. One of the biggest challenges in processing massive hypercubes is in the amount of physical memory available to the computer quickly becoming too little to process.

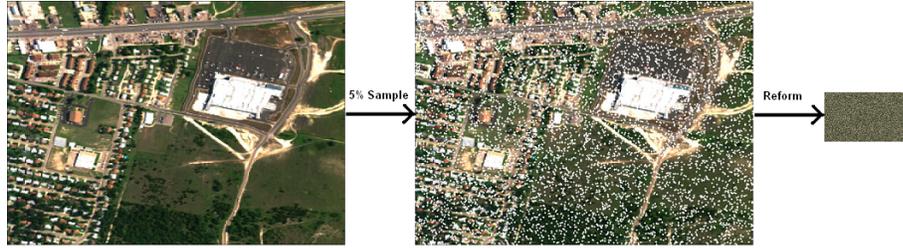
Because of this and other reasons, it is of great benefit to reduce the computational load as well as the memory requirements of the algorithm. Land-marking is a process in which we drastically reduce the computational load in such a way that we preserve most of the relevant geometry and our ability to eventually classify materials.

We first select a sample $\{Z_i\}_{i=1}^M$ of points from X . There are many ways to choose this sample, and the selection method comes with its own list of benefits and consequences. It is an interesting problem to consider how to sample a large scene for the purpose of reducing computational complexity, while not leaving out important anomalies that are represented by few of the data points. We shall select a uniformly random sample from the data, where M is some percentage of the the number of data points X . This percentage is typically two to twenty percent. One advantage of this technique are estimation bounds on error and rates of convergence established by [9]. Of course, one of the disadvantages are a lack of data dependence

and causing the algorithm to be non-deterministic.

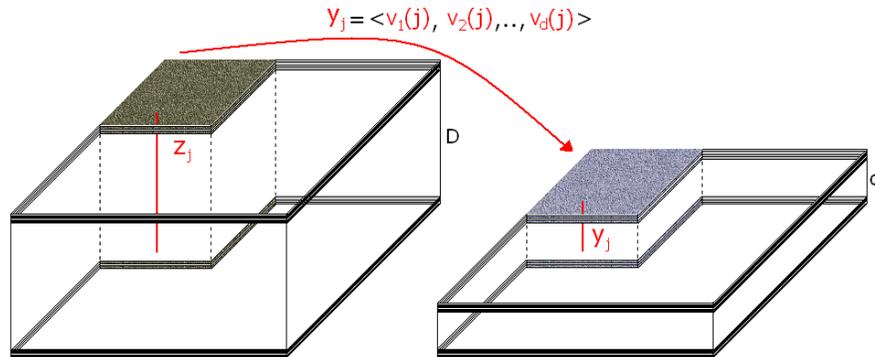
After obtaining this sampling, Z , we generate a graph from the points in Z and create a kernel from this, as if we had reformed the sampled data into a new smaller data cube, as this picture demonstrates.

Figure 3.17: Typical Landmarking Sample.



We proceed as if this were a full data cube, create a kernel and project the data onto the lower dimensional space.

Figure 3.18: Dimension Reduction on the Sampled Set.



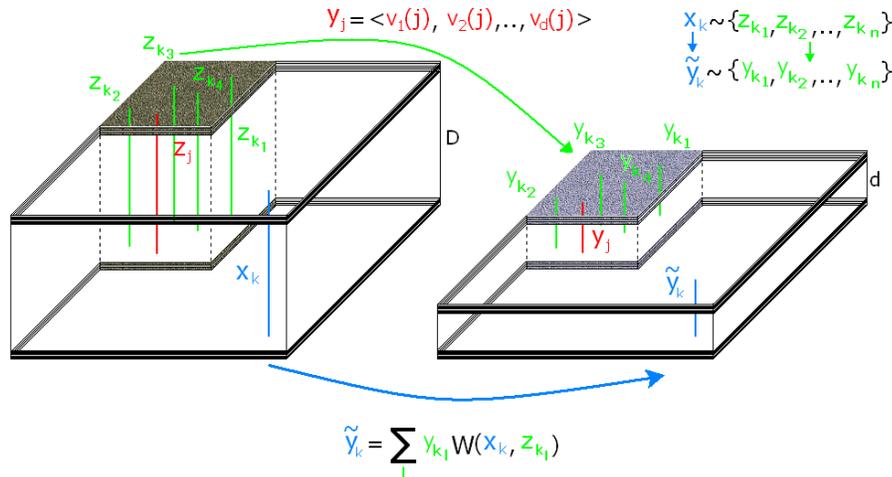
The strength of the land-marking procedure is that it provides a natural mechanism to extrapolate from this sampled dimension reduction, a full projection of the entire data cube onto the lower dimensional coefficient space. This method was developed by Bengio, Paiement and Vincent in [9] and refined by Coifman and La-

fon in [15] for use with LLE. These papers give reasonable estimates relating the sampling percentage to the result fidelity. Choose an $x_k \in X \setminus Z$, since this is not a sampled point, we do not know right away how to project it onto Y . The idea is to represent x_k as a combination of ‘pseudo-neighbors’ from the sampled collection Z . We essentially find a collection of neighbors for x_k , $\{z_{k_1}, z_{k_2}, z_{k_3}, \dots\}$, using one of our criteria for neighbors, where we exclude all of the vectors that are not part of the sampled set.

Luckily we do have information about the projection of the ‘pseudo-neighbors’ onto the space Y , so we will extend the projection onto Y , to the vector x_k , by expressing it as a weighted sum of the projections of those same pseudo-neighbors. The only step remaining is to recalculate the graph weights by taking into account x_k , and then we can define the projection of x_k as

$$\tilde{\rho}(x_k) = \sum_{i=1}^l \tilde{W}(x_j, z_{k_i}) \rho(z_{k_i}).$$

Figure 3.19: A Schematic of the Out-of-Sample Extension.



3.3.5 Endmember Selection

The fourth step in our algorithm is to select endmembers for the low dimensional space $Y \subset \mathbb{R}^d$. Endmembers are typically defined to be the constituent elements of a scene, that is, they are generally thought to compose the pure elements in a spectral data cube. Traditional applications of endmember algorithms are run on the original high dimensional data set $X \subset \mathbb{R}^D$ and if s denotes the number of endmembers then $s < D$. Since we are finding endmembers for the space Y , we propose finding $s > d$ endmembers, thus creating a frame $\Phi = \{\phi_i\}_{i=1}^s$ for Y . Frames arise naturally in dimension reduction, and are in fact a generalization of orthonormal bases. There are many endmember selection algorithms available, e.g., N-FINDR [57], ORASIS [11] and Pixel Purity Index [10]; see also [58] and [36]. The results of this section employ the Support Vector Data Description (SVDD), see, e.g., [4] algorithm for selecting endmembers. The core idea of SVDD is to obtain a minimal spherical shaped boundary around the data set, which in turn gives a description of the data in terms of a set of support vectors.

3.3.5.1 Frame Coefficients

Given a frame $\Phi = \{\phi_i\}_{i=1}^s$ for Y , we shall find a set of coefficients $C = \{c_{i,j}\}_{i,j=1}^{N,s}$ that represents Y in terms of Φ :

$$y_i = \sum_{j=1}^s c_{i,j} \phi_j \text{ for all } i = 1, \dots, N. \quad (3.4)$$

We propose two separate ways to find C , allowing for the pursuit of this

modular pipeline. The first is based on the frame operator $S : \mathbb{R}^d \rightarrow \mathbb{R}^d$, which is:

$$Sy = \sum_{i=1}^s \langle y, \phi_i \rangle \phi_i \text{ for all } y \in \mathbb{R}^d. \quad (3.5)$$

For any frame Φ , the frame operator S is invertible, and in fact gives the following representation:

$$y = \sum_{i=1}^s \langle y, S^{-1}\phi_i \rangle \phi_i \text{ for all } y \in \mathbb{R}^d. \quad (3.6)$$

Thus we can define the coefficient set $C = \{c_{i,j}\}_{i,j=1}^{N,s}$ as:

$$c_{i,j} = \langle y_i, S^{-1}\phi_j \rangle \text{ for all } i = 1, \dots, N, j = 1, \dots, s.$$

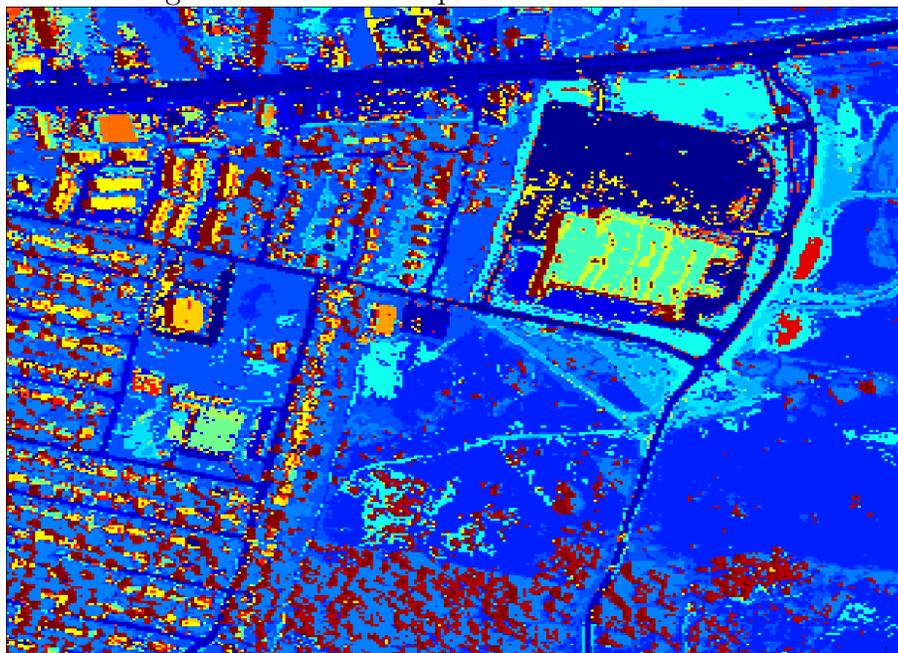
The coefficients $\{c_{i,j}\}$ are called the canonical coefficients and they minimize the ℓ^2 energy of the coefficient set C . An alternative to the canonical coefficient set is to find sparse coefficient representations. Such coefficients can be found by minimizing the ℓ^p energy of the coefficients, where $0 < p \leq 1$:

$$c_{i,\cdot} = \arg \min_{\tilde{c}} \|\tilde{c}\|_{\ell^p} \text{ subject to } y_i = \sum_{j=1}^s \tilde{c}_j \phi_j. \quad (3.7)$$

3.3.6 Frame Based Kernel Method Results

Before moving to the main results of this section, there are a couple of important images that should be singled out. First, the pipeline implemented with the urban data set can generate the following class map in Figure 3.20, just to illustrate the kinds of data we hope to end up with.

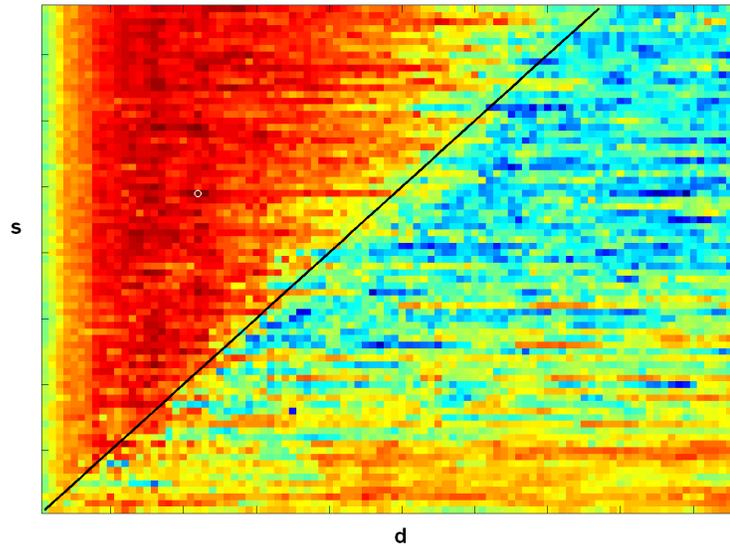
Figure 3.20: Class Map of the Urban Data Tile.



That is, we tell the pipeline which kernel to use, how many neighbors to make the graph implement, how to compute the reduced coefficients and sampling percentages for the landmarking process. Once all of these ideas are fixed, there are only two important numbers to consider. One is the number of dimensions to reduce to, the second is the number of endmembers used to represent that lower dimensional set. A very interesting phenomenon came about while running an experiment on the urban data set, that included permuting those two significant variables with everything else being fixed. The resulting image in Figure 3.21 is due to that experiment.

This image is very special. The x-axis represents the number of dimensions the urban data cube was reduced to, with a range of roughly 10 to 35. The y-axis represents the number of endmembers used to represent that data. Note that when $s = d$ the representing set is a basis, and when $s > d$ the representing set is a frame.

Figure 3.21: The Frame Flag.



The black diagonal line represents the boundary between a frame and a spanning collection of a subspace. This particular data set comes equipped with ground truth consisting of 22 classes divided up into pre-known pixel locations. We can assert the quality of a classification into those 22 classes (like in Figure 3.20) by testing the classified ‘known pixel’ locations against our attempts at classifying them. This gives us a percentage of true classifications and false positives. The color of this flag-image represents the quality of the classification after running the pipeline on the data, where all classes have been amalgamated to a single overall classification percentage. The red represents very strong classification while blue is very poor classification. The fact that the red clusters above the frame-line says that frames really are the key element of this pipeline and the addition of redundancy to the concept of endmember is a valuable one.

The results of this section are based on the Smith Island HSI data described in the data portion of this chapter. Here these are graphically displayed in Figures 3.22, 3.23 and 3.24. We use the following settings in our pipeline: Kernel: LLE, Number of Neighbors: 50, Number of Samples: 40000 pixels, Number of reduced dimensions: 21, Endmember Algorithm: SVDD, Number of Endmembers: 69, Type of Coefficients: Canonical, Classifier: Vector Angle.

3.4 Wavelet Packets and Spectral/Spatial Representation

In this section we will mirror the developments in [7] to give a short introduction to the use of wavelet packets in combining spectral and spatial representations for dimension reduction.

Tools for the analysis of hyperspectral data have successfully been applied to detect and classify objects in areas ranging from human pathology to geophysics and satellite imaging. Hyperspectral image analysis relies on using dimension reduction, because spectral signatures and their mixtures are supposed to lie on low-dimensional manifolds. Principal component analysis (PCA) is a common dimension reduction method, [48]. It is a linear transform that determines the ‘directions’ in the data by maximizing the captured variance. If the manifold is linear, then PCA is the optimal choice.

Dimension reduction methods and classification in hyperspectral imaging use only spectral information. However, the images also have spatial contents that are useful. Wavelets and their multilevel decomposition are well-suited for the analysis

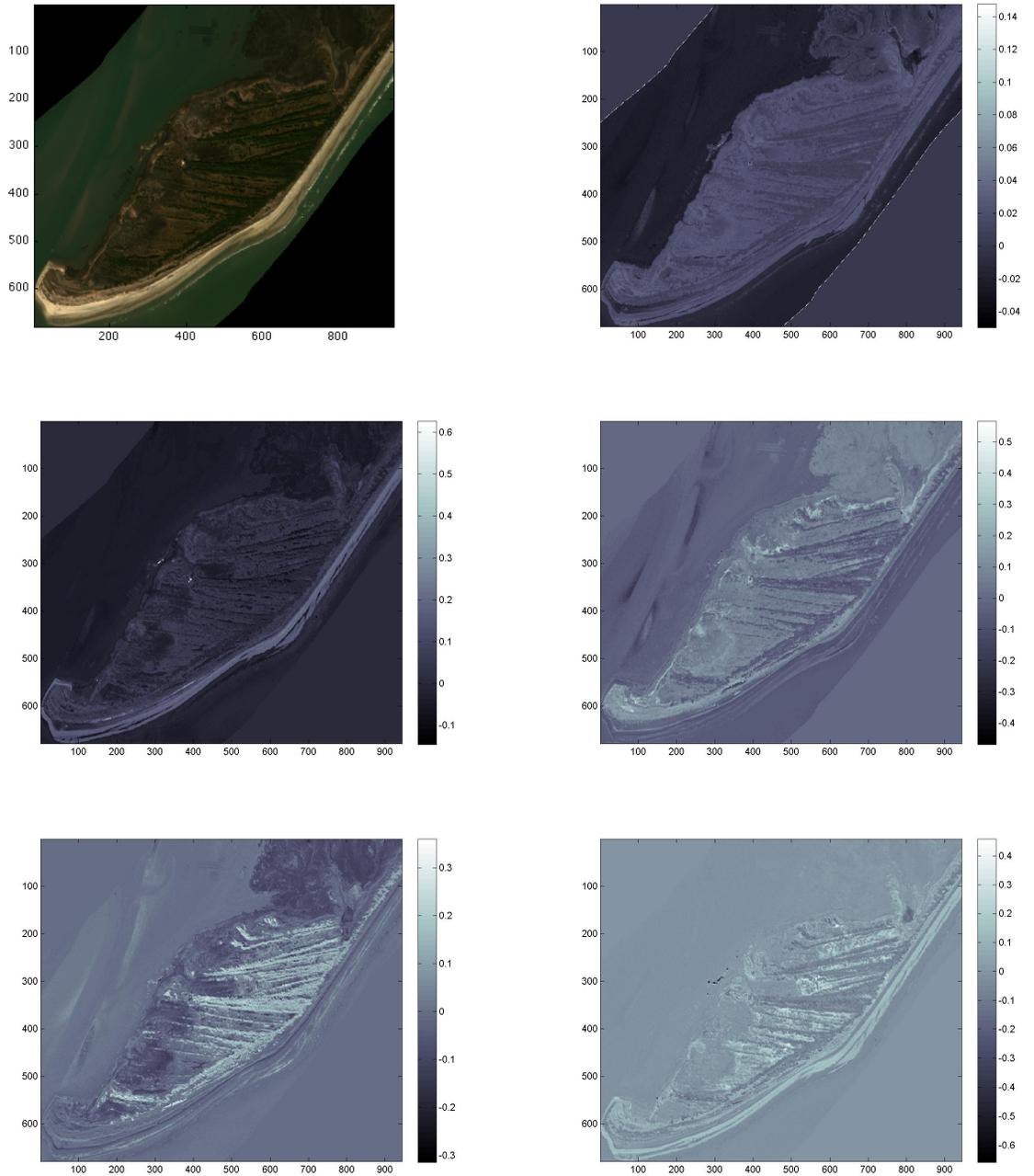


Figure 3.22: Frame Coefficient Maps of Smith Island

Figure 3.23: Smith Trial Ground Truth Results.

	#	#c	%	#fp	#fn
phrag4	196	138	70%	68	58
scirpus	246	155	63%	55	91
juncus	184	116	63%	33	68
patens	66	57	86%	33	9
distichlis	97	90	93%	18	7
andropogon	57	38	67%	9	19
ammophila	32	25	78%	29	7
mud	70	63	90%	25	7
alterniora	200	182	91%	60	18
borrichia	90	84	93%	24	6
salicornia	76	58	76%	3	18
iva	58	49	84%	51	9
pine	166	134	81%	59	32
hardwood_mixed	328	193	59%	41	135
pond_water	105	69	66%	3	36
sand	159	157	99%	0	2
wrack	144	97	67%	11	47
myrica	167	132	79%	54	35
seaoats	18	13	72%	0	5
typha	44	18	41%	59	26
Water_Nshore	206	206	100%	0	0
SubmergedNets1	34	34	100%	0	0
Total	2743	2108	77%	635	635

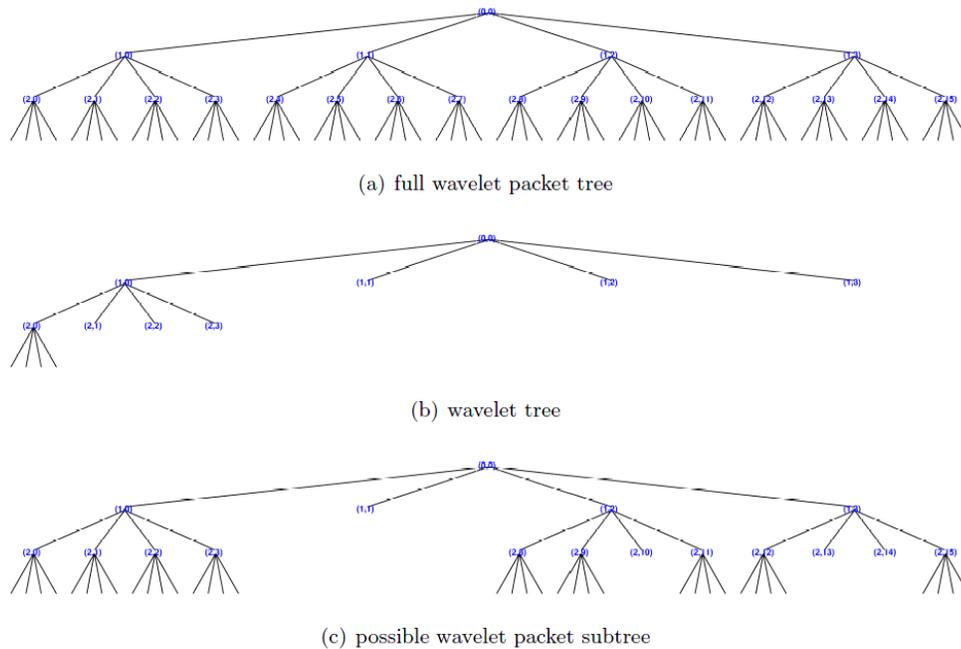
Figure 3.24: Smith Competing Ground Truth Results (total only).

	#	#c	%	#fp	#fn
Raw data	2743	1957	71%	786	786
SVDD	2743	2088	76%	655	655
LLE ($d = 19, \dots, 25$)	2743	2059	75%	684	684

of spatial characteristics. We shall combine wavelet analysis with PCA to capture spatial and spectral data distributions. The naive approach is the sequential wavelet decomposition of each spectral band, forming an hyperspectral dataset of coefficients in the wavelet domain, and performing PCA on this coefficient hypercube. This, however, does not ‘show’ the wavelets that they are in fact decomposing many images from the same scenery. While decomposing band-wise, one must find a way to incorporate spatial information from other spectra.

The Discrete Wavelet Transform (DWT) is an iterative scheme that splits the signal into approximation and detail coefficients. Each level is computed by passing through only the previous approximation coefficients. The Wavelet Packet Transform (WPT), however, decomposes both the approximation and the detail coefficients. Contrary to the DWT, this yields a full wavelet tree decomposition that offers flexibility in the choice of reconstruction coefficients. The best bases algorithm by Coifman and Wickerhauser finds the best subtree for reconstruction, i.e., the best coefficient set according to an entropy measure, [19]. See Figure 3.25 for a visualization of the subtree concept.

Figure 3.25: The Entropy Determines the Subtree that is Chosen for Reconstruction.



Wavelet Packets and the entropy concept offer the flexibility to transfer spatial information from one band to the other. We separately decompose each band by using the WPT, but we consider a common entropy over all bands. The common en-

tropy still respects the correlation between different frequency bands in the wavelet domain. The best 'common' subtree is determined by evaluating the common entropy. Each node of the common subtree then is a set of coefficient vectors whose entries refer to different spectral bands. PCA is applied to these vectors and we then reconstruct by applying the inverse WPT. Fusing spectral and spatial information significantly improves the output of classification schemes.

3.4.1 Wavelet Packets

This subsection will contain a short exposition about wavelet packets and how they will be utilized in the scheme described in the above section. To simplify the notation in this section, we will assume that instead of a traditional data cube, i.e., 3 dimensional matrix containing D -spectral slices, we suppose our 'cube', X , is a $D \times M$ matrix, containing d -linear spectra. We can then define our 'pixels', $\{x_n\}_{n=1}^M \subset \mathbb{R}^D$, as columns of the matrix X .

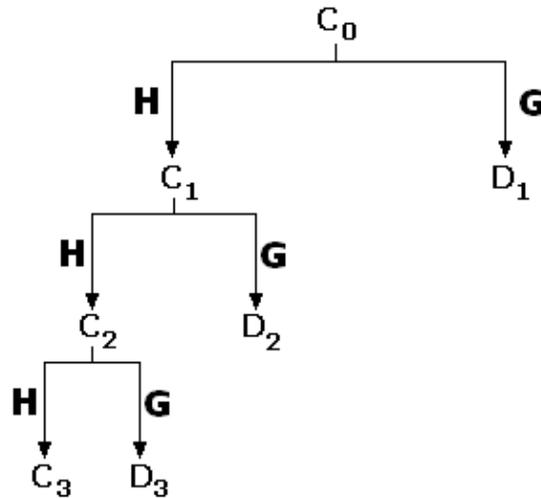
Now choose an orthogonal wavelet, with mother wavelet ϕ and father wavelet ψ with associated scaling filter h and wavelet filter g . Typically, the wavelet decomposition of a vector is done in the following way. Let the initial data, the vector, $x = C_o$. We define the approximation operator H and the detail operator G acting on a vector C by

$$(H C) [k] = \sum_n C[n] \overline{h(n - 2k)},$$

$$(G C) [k] = \sum_n C[n] \overline{g(n - 2k)}.$$

In this case we can compute the wavelet transform of the vector $x = C_o$ by calculating the values of $\{C_n\}$, the approximation coefficients, and $\{D_n\}$, the detail coefficients at the n 'th scale by following the schematic:

Figure 3.26: The Wavelet Coefficient/Filter Tree.



There is nothing keeping us from completing the tree, by applying H and G to each new coefficient. This is precisely the idea behind wavelet packets, and the use of the full tree produces a technique that is rich in complexities. In the wavelet decomposition, calculating the coefficients by applying the operators G and H can be both tedious and not often revealing about the coefficients themselves. Note that if we call $w^o = \phi$ and $w^1 = \psi$ that we can write some of the coefficients from the above picture as

$$C_1[k] = (H C_o)[k] = \langle x, w_{-1,k}^o \rangle,$$

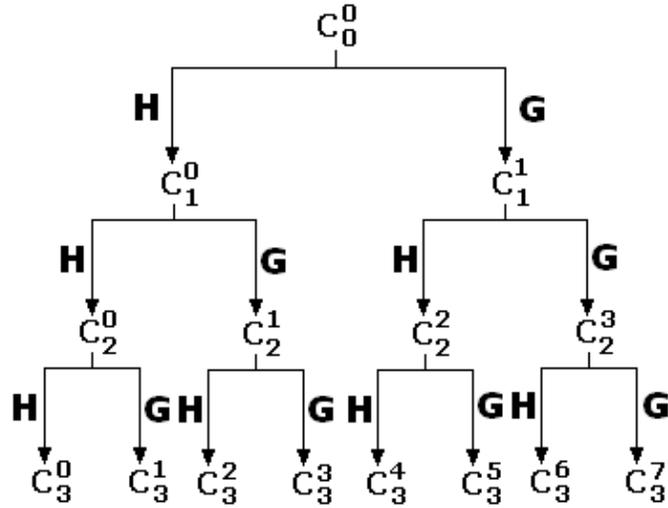
$$D_1[k] = (G C_o)[k] = \langle x, w_{-1,k}^1 \rangle,$$

$$C_2[k] = (H C_1)[k] = \langle x, w_{-2,k}^o \rangle,$$

$$D_2[k] = (G C_1)[k] = \langle x, w_{-2,k}^1 \rangle.$$

The point of writing this, is to ask ourselves how we can calculate coefficients in the wavelet packet sense, in a manner analogous to this. Namely, if we continue our tree and fill it out, as in the following picture

Figure 3.27: The Wavelet Packet Coefficient/Filter Tree.



where now,

$$C_1 = C_1^0,$$

$$D_1 = C_1^1,$$

$$C_2 = C_2^0,$$

$$D_2 = C_2^1,$$

how can we calculate C_2^2 or C_2^3 in as clean and efficient a way as the above calculations?

Definition 4. Let $w^0 = \phi$ and $w^1 = \psi$. We can then define the sequence $\{w^n(x)\}_{n \in \mathbb{Z}^+}$ of wavelet packet functions recursively by

$$w^{2n}(x) = \sum_k h[k]w_{1,k}^n(x),$$

$$w^{2n+1}(x) = \sum_k g[k]w_{1,k}^n(x).$$

In this way, we can define the wavelet packet transform of x as $\{C_k^j[n] = \langle x, w_{-k,n}^j \rangle\}$.

3.4.2 Best Basis Algorithm

Clearly the wavelet packet transform of x is quite overcomplete. We wish to reduce the total information contained in the transform by eliminating nodes in the tree that are both redundant, and in some way add very little to the reconstruction. This is called the best basis algorithm and is a major component for compression of data. To select a best basis, we require some criteria for how to do so, and so we arrive at the so-called entropy (cost) function.

Definition 5. *The entropy function, E , maps vectors (of any length) into the real numbers and has the following properties:*

$$E(\mathbf{0}) = 0,$$

$$E([\mathbf{v}_1 \ \mathbf{v}_2]) = E(\mathbf{v}_1) + E(\mathbf{v}_2),$$

where $[\mathbf{v}_1 \ \mathbf{v}_2]$ is the concatenation of the vectors \mathbf{v}_1 and \mathbf{v}_2 .

Examples:

1. A very simple entropy function is the threshold function with the threshold value a ,

$$E_a(\mathbf{v}) = \sum_j c[j]$$

where $c[j] = 1$ if $|\mathbf{v}[j]| > a$ and $c[j] = 0$ otherwise.

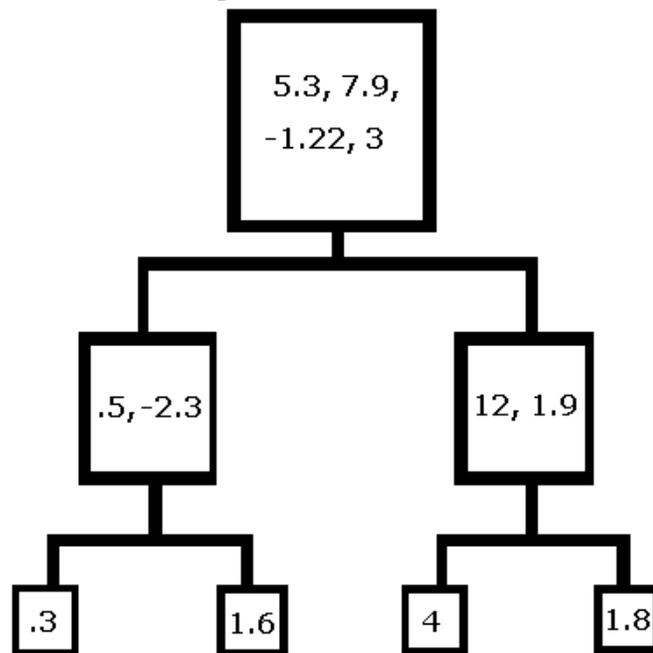
2. A more useful entropy function, from an information theory viewpoint, is the Shannon entropy function,

$$E_{\text{Shannon}}(\mathbf{v}) = - \sum_j \mathbf{v}^2[j] \ln(\mathbf{v}^2[j]).$$

We will use an example to illustrate the mechanics of the best basis algorithm.

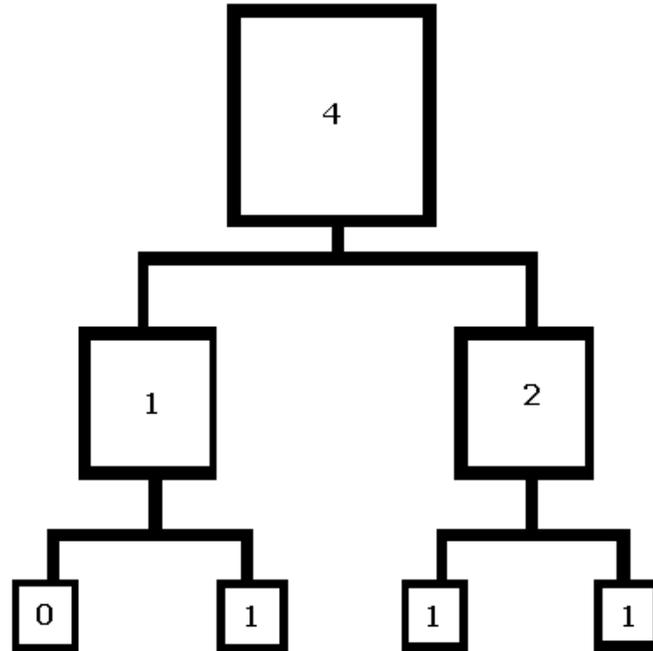
To make matters simpler, we will use the threshold entropy function with threshold value $a = 1.1$. Our wavelet packet transform produces the following coefficients, in Figure 3.28.

Figure 3.28: An Example of a Wavelet Packet Coefficient Tree.



Calculating the threshold entropy function with value $a = 1.1$ is easy, since we must only count the number of entries in each node whose absolute value is greater than a . The first node is therefore assigned the value 4 and the entire entropy can be calculated as in Figure 3.29.

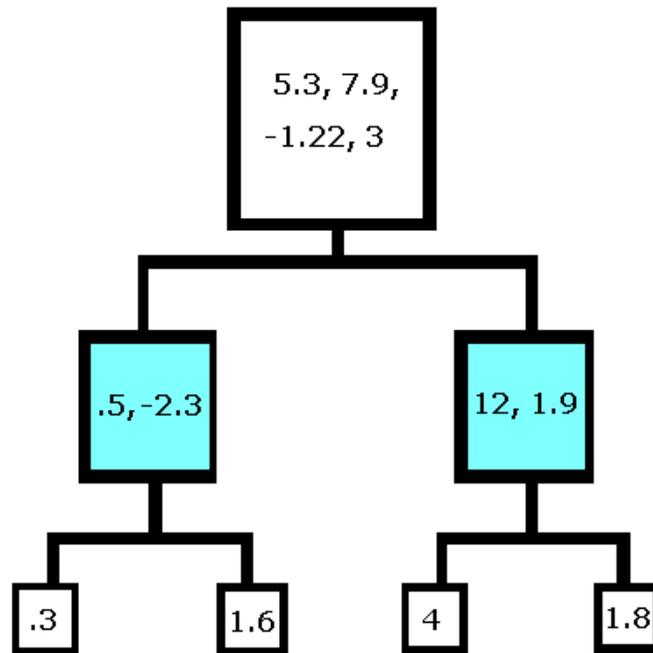
Figure 3.29: An Example of a Corresponding Entropy Function.



The algorithm then starts from the bottom, or leaf nodes, and simply applies the value of the entropy function to those nodes, and considers all of them to be part of the best basis. The next step looks at the parent nodes to those leaves and asks the question: is the parent node's entropy value smaller than or equal to the sum of the offspring entropy functions? If the answer to this question is affirmative, then we mark the parent node as a member of the best basis, and forget its children. If the answer to the question is no, then we replace the entropy function value of the parent node with the sum of the entropy function values of its children. In this

way we can work out way up the tree and at the end have marked our best basis with respect to that particular entropy function. In our example we would end up with Figure 3.30

Figure 3.30: The Best Basis Algorithm.

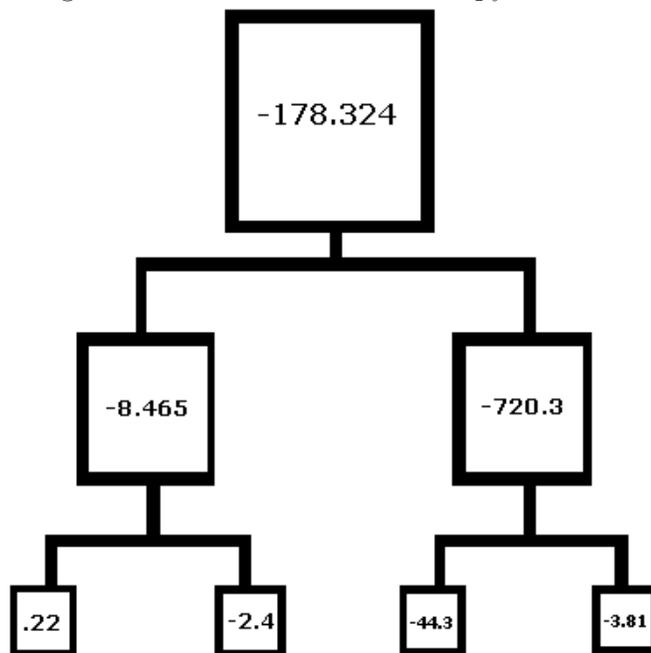


where the colored-in squares are the final best basis. Just for edification, the Shannon entropy function would have given the values in Figure 3.31. resulting in exactly the same best basis in this particular case.

3.4.3 Inverting the Wavelet Packet Transform

The inversion of the wavelet packet transform can be easily (and efficiently) computed by noting the relationship of the operators G and H , and in particular, their adjoints.

Figure 3.31: The Shannon Entropy Function.



$$(H^*c)[k] = \sum_n c[n] h[k - 2n],$$

$$(G^*c)[k] = \sum_n c[n] g[k - 2n].$$

Since we have the properties that

$$HH^* = GG^* = H^*H + G^*G = I,$$

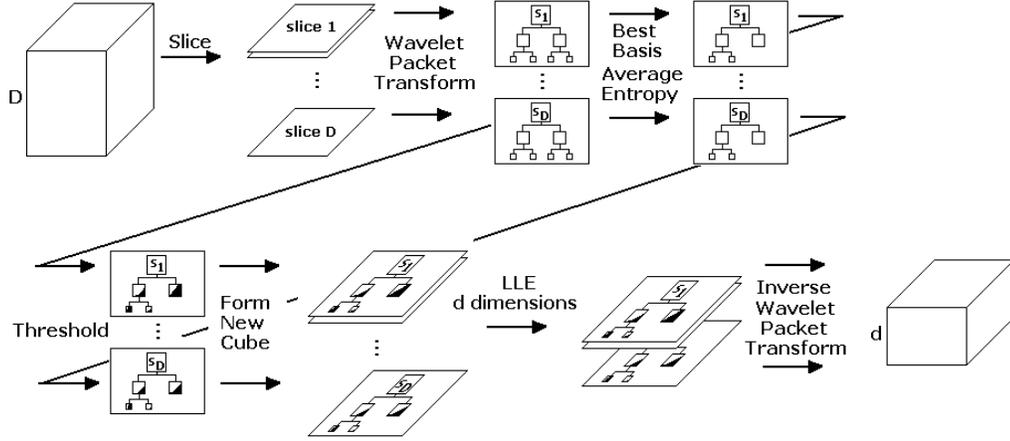
we can recover the coefficient C_k^j by writing

$$C_k^j[n] = H^*C_{k+1}^{2j}[n] + G^*C_{k+1}^{2j+1}[k].$$

3.4.4 Combining LLE and Wavelet Packets

We wish to use the properties of wavelet packets to massage our data cube into a sparse and more natural source of data for LLE to operate upon. The plan roughly follows the following schematic diagram.

Figure 3.32: Spectral/Spatial Dimension Reduction Schematic.



This will a short guide to this process, just keep in mind that the notation will still be within our simpler 2-dimensional framework.

- 1: Begin with a hyperspectral data cube, X , that is $D \times M$, where D is taken as quite large. We then slice the cube into individual layers, which we will call x_n . Each x_n lives in \mathbb{R}^M and there are D of them.
- 2: We choose an orthonormal wavelet system with associated mother wavelet ϕ , father wavelet ψ and assign their filters to be h and g . We assign $w^0 = \phi$ and $w^1 = \psi$ and calculate the remaining associated wavelet packet functions, $\{w^m\}$, by the recursive formulas

$$w^{2m}(x) = \sum_k h[k]w_{1,k}^m(x),$$

$$w^{2m+1}(x) = \sum_k g[k]w_{1,k}^m(x).$$

For each slice, x_n , we calculate the wavelet packet transform independently, which we will denote as $\{C_{n,k}^j\}$, by the formula

$$C_{n,k}^j[m] = \langle x_n, w_{-k,m}^j \rangle.$$

- 3: We then specify an entropy function, E , and compute each of the values $E(C_{n,k}^j)$ for each node in every tree representing an individual slice. We then average the entropy functions together, thereby tying each slice to every other one when it comes time to find a best basis. We write this as

$$\tilde{E}(A_k^j) = \frac{1}{D} \sum_{n=1}^D E(C_{n,k}^j)$$

where we understand that A_k^j represents a position in any of the individual trees. Next we replace the assigned entropy values in each tree with the average version calculated from \tilde{E} . That is, every node that is in the same position, will have precisely the same entropy value, across every tree. We then apply the best basis algorithm to only one of the trees (since they share the same entropy values, the result will be the same for every tree).

4. The next step involves thresholding the best bases that we have just generated. This will add sparsity once we have recreated our cube.
5. Now we wish to take our threshed best bases and force them to look like a data cube that LLE can act upon. One way to do this is to assign nodes earlier precedence based on a lexicographic like ordering. That is, we have the best basis represented by the collection $\{A_k^j\}$ where $k \in \Omega$ and $j \in \Theta$, where these are the appropriate index sets for the best basis, then we say $A_{k_1}^{j_1}$ has precedence over $A_{k_2}^{j_2}$ if $k_1 < k_2$ or if $k_1 = k_2$ and $j_1 < j_2$. This gives a unique ordering, and allows us to create the new vectors y_n as the concatenation of the threshed coefficients that are in the best basis, in order of precedence. We then think of each y_n as a row in the matrix Y , which has D rows.

6. With our new data cube in hand, we assign 'pixel' vectors to be the columns of Y , establish neighbors and neighbor weights and then create an LLE kernel. We then extract the d most important eigenvectors and have effectively reduced our wavelet packet cube's dimension to d .
7. We retain the concept that each row in the LLE reduced version of Y still represents a wavelet packet best basis, and thus each can be reconstructed. We take the rows, \tilde{y}_n , for which there are d , and recalling the ordering we used in step 5, and reverse it, forming true wavelet packet trees. With this in mind, we can invert the wavelet packet transform using the formula

$$C_{n,k}^j[m] = H^* C_{n,k+1}^{2j}[m] + G^* C_{n,k+1}^{2j+1}[m],$$

for each $m = 1, \dots, d$.

What remains after inverting the wavelet packet transform is the desired result; a d -dimensional representation of the original D -dimensional data cube.

3.4.5 Numerical Results

Given a full wavelet tree, let \mathcal{T} denote the collection of all subtrees. The entropy E (or cost function) of a wavelet packet decomposition is a nonnegative map $E : \mathcal{T} \rightarrow \mathbb{R}$. Given n spectral bands, we now decompose each band by the WPT. This gives us an entropy E_i , for $i = 1, \dots, n$, for each band. We define the common entropy E_C through the weighted ℓ^p term

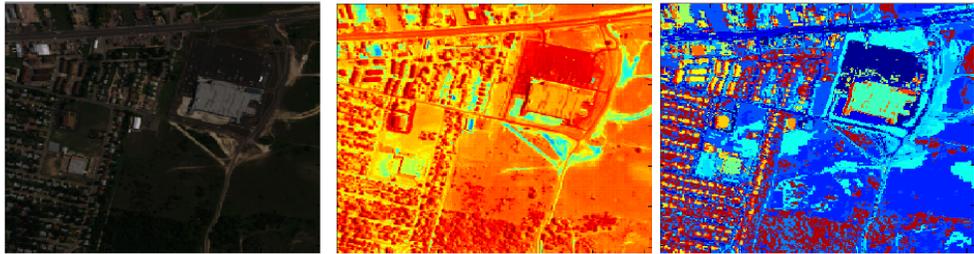
$$E_C = \sum_{i=1}^n w_i |E_i|^p, \quad (3.8)$$

where $0 < p \leq 2$ and a weight vector $\{w_i\}_{i=1}^n$. Minimizing the entropy yields the best subtree for reconstruction.

The assumption that the underlying manifold in the wavelet domain is linear might not hold. It is much more likely that we are in a nonlinear regime. PCA then cannot capture the essential information by itself. There are two ways to introduce nonlinearity. We could replace PCA with a nonlinear dimension reduction such as Locally Linear Embedding (LLE), Laplacian Eigenmaps (LE), Hessian LLE [28], and Diffusion Wavelets/Diffusion maps [18] [14] [15] [17] [16] that try to recover the manifold. However, these methods are computationally expensive. Our simplistic approach is to apply (soft-)shrinkage to the wavelet coefficients the classical concept for compression and denoising. In the context of hyperspectral imagery, however, we use the neutral formulation ‘shrinkage concentrates spatial and spectral information into few coefficients in the wavelet domain’. Through shrinkage, the output of the WPT becomes a nonlinear approximation of the signal. Moreover, shrinkage means smoothing of the underlying manifold. This is a step towards linearization. Finally, we apply PCA to the shrunken coefficients and then transform back. We have applied the method to the Urban dataset. Since we have ground-truth, we can compare our proposed method with standard PCA on the original data set with respect to classification. We use 15 to 50 principal components that capture more than 99% of the variance. For the Figures 3.33 and 3.34, we have chosen the Haar

wavelet, with $p = 1.3$, and $w_i = 1$, for each $i = 1, \dots, n$. Our proposed method outperforms simple PCA, see Figure 3.35.

Figure 3.33: Classification Results on Urban Data Set.



(a) RGB image of Urban dataset (b) reconstruction from the first PCA component of the wavelet domain (c) vector angle classification using 15 PCA components from shrunk wavelet packet coefficients

Figure 3.34: Selection of Classes of the Vector Angle Classification Scheme for the Urban Data Set.

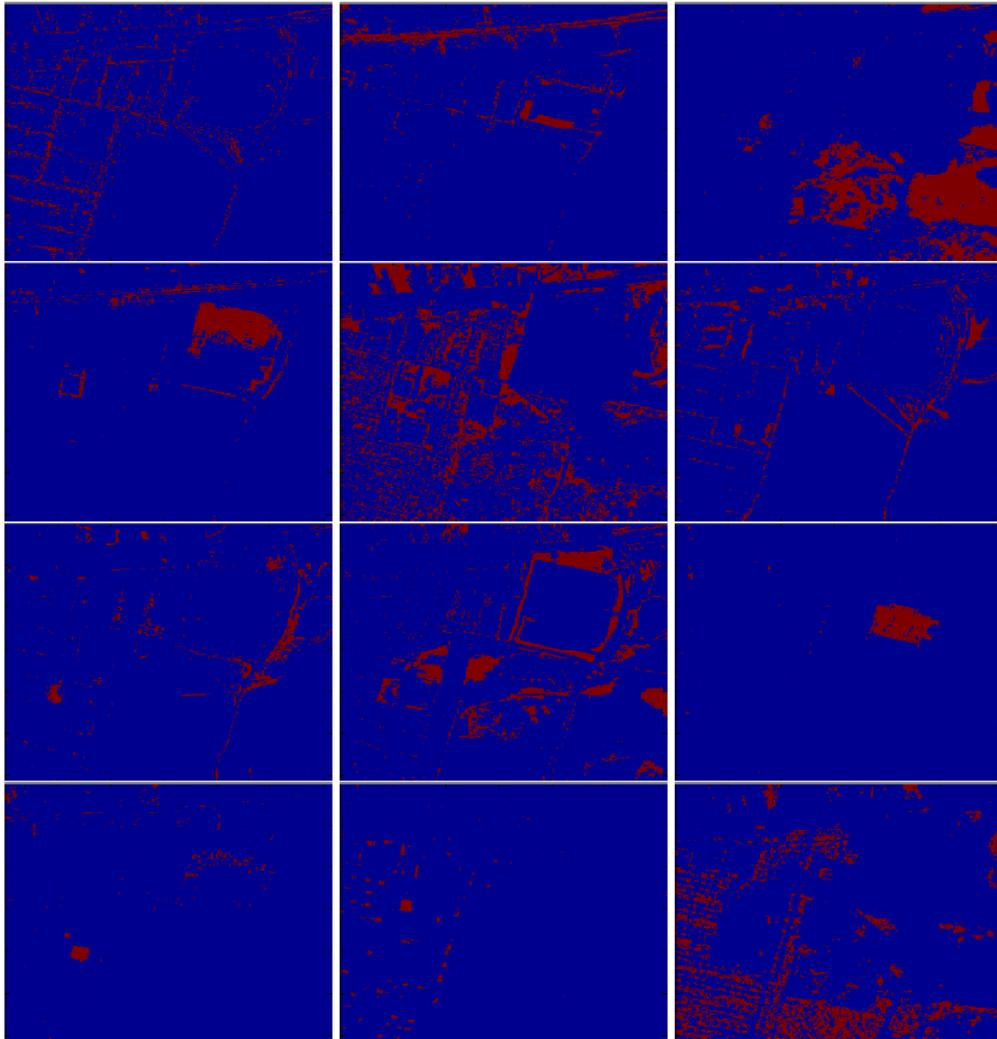
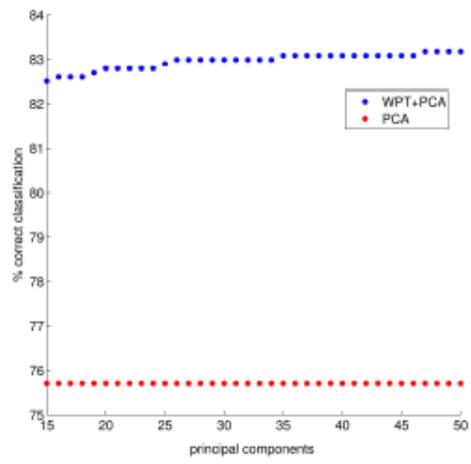


Figure 3.35: The Combination of WPT with PCA Outperforms Simple PCA.



Chapter 4

The Multiplicative Zak Transform

4.1 Introduction

In 1946, D. Gabor [34] and the systems bearing his name facilitated the re-discovery of a useful transform first discovered long ago and utilized by Weil [55], now referred to as the Weil-Brezin, Zak or kq-transforms. Taking our cue from Folland [31], “the transform is referred to in the literature as the Weil-Brezin transform (the name we shall adopt) or the Zak transform.”

The Zak transform demonstrates its remarkable flexibility in the sheer number of applications to which it is applied. These applications include solid state physics, signal processing, and noncommutative algebra. Within the realm of time-frequency analysis, the Zak transform aids in the characterization and creation of new families of Gabor systems and is integral in proving the most general form of the Balian-Low obstruction theorem.

The underlying group structure of Gabor systems is $H_{\mathbb{R}}$, the real Heisenberg group. This structure is clearly represented in the summands of the Zak transform as the Schrödinger representations of the Heisenberg group [32]. A corresponding transformation for wavelet systems would certainly depend on the representations of the affine group. On the surface, the difference in the action of the Heisenberg group and the affine group on $L^2(\mathbb{R})$ may appear superficial, however, due to the

non-transitivity and the dearth of subgroups of the affine group, adapting the Zak transform directly for wavelet systems is likely to fail.

Auslander, Eichmann, Gertner, and Tolimieri defined a multiplicative Zak transform [1], mimicking the construction of the Gabor Zak transform by taking into account the quirks of the affine group. This limits their analysis in the subsequent paper [35] to a subspace on which the action is transitive, namely the Hardy space $L^2((0, \infty))$. The thrust of this work was the characterization and analysis of wavelet systems that form frames for their spans, where the Fourier transform of the generating function is supported on the positive real line. Such wavelets cannot span all of $L^2(\mathbb{R})$ since they do not represent the negative frequencies. Specifically, they show that analyzing the wavelet collections described above with the multiplicative Zak transform reduces to checking the bounds of the function

$$\sum_{j \in J} \left| \hat{\psi}(2^j \gamma) \right|^2$$

almost everywhere on $[1, 2)$, where the index set J has finite cardinality.

Segman and Schempp [50] constructed an alternate form of the multiplicative Zak transform. Their results allow the characterization of wavelet-like frames through the values of the transform on a compact subset of \mathbb{R}^2 . The reason that we do not use their transform is because their goal was the incorporation of scale into the Heisenberg group and not necessarily the characterization of wavelet frames. Because of this, their analyzed wavelet frames do not correspond to those generated by representations of the affine group.

In this chapter we shall extend the transform constructed by Gertner et al.,

weaken the condition requiring support within $(0, \infty)$, allowing limited support in $\widehat{\mathbb{R}}$ and study possible methods to remove the requirement of compact support of the Fourier transform of the generating function. Through the use of superframes [3], we will demonstrate a possible framework within which wavelet frames on the real line could be characterized by the multiplicative Zak transform. Finally, we will introduce a simple inversion formula for the multiplicative Zak transform.

4.2 Notation

We will be discussing both Gabor and wavelet systems. Gabor systems will always be generated by the letter g . All other Latin letters denote functions acting on the time domain. Greek letters will denote functions acting on the Fourier domain with the exception that ψ will be considered as acting in the time domain as a wavelet generating function. Occasionally there will be objects that live on $(0, \infty)$, in each of these cases we mean that $(0, \infty) \subset \widehat{\mathbb{R}}$. The capital Greek letters Φ and Ψ will always denote dilation periodic functions acting on a periodization of the Fourier domain. Often these functions will be restricted to a specific set, on which they are uniquely defined. The context will make this distinction clear.

4.3 Gabor systems and the Zak transform

Before defining the multiplicative Zak transform, it will be instructive to briefly review Gabor systems and the Gabor Zak transform. Our construction and use of the multiplicative Zak transform on wavelet collections will mirror that of the Gabor

situation.

Definition 6. Let $a, b > 0$ and let $g \in L^2(\mathbb{R})$. We call the collection

$$(g, a, b) = \{g_{m,n}(t) = e^{2\pi ibmt} g(t - an) \mid m, n \in \mathbb{Z}\}$$

the uniform Gabor system generated by g with translation parameters a and b .

Gabor originally used a Gaussian in the place of g with parameters $a = b = 1$.

It turns out that this particular collection is complete in $L^2(\mathbb{R})$ but does not form a frame. We will only consider the case when $a = b = 1$.

Definition 7. For $f \in L^2(\mathbb{R})$ define the Zak transform to be

$$Zf(t, \omega) = \sum_{k \in \mathbb{Z}} f(t - k) e^{2\pi i k \omega}, \text{ for } (t, \omega) \in [0, 1)^2.$$

4.3.1 Properties of the Zak transform

Let $Q = [0, 1)^2$ and consider the two orthonormal bases of $L^2(\mathbb{R})$ and $L^2(Q)$, respectively:

$$\{\mathbb{1}_{[m, m+1)}(t) \cdot e^{2\pi i n t}\} \text{ and } \{e^{2\pi i m t} \cdot e^{2\pi i n \omega}\}.$$

Proposition 1. $Z : L^2(\mathbb{R}) \rightarrow L^2(Q)$ is a unitary isomorphism.

Proof: This is clear from the mapping

$$Z : \mathbb{1}_{[m, m+1)}(t) \cdot e^{2\pi i n t} \mapsto e^{2\pi i m t} \cdot e^{2\pi i n \omega}.$$

□

Corollary 1. $\{f_k\}$ is a frame for $L^2(\mathbb{R})$ with frame bounds $A, B > 0$ if and only if $\{(Zf_k)\}$ is a frame for $L^2(Q)$ with frame bounds $A, B > 0$.

Note that for a particular $g_{m,n}(t) = e^{2\pi imt}g(t - n)$, the Zak transform 'pulls' the modulation and translation off of g in the following way:

$$\begin{aligned} (Zg_{m,n})(t, \omega) &= \sum_k g_{m,n}(t - k)e^{2\pi ik\omega} = \sum_k g(t - k - n)e^{2\pi im(t-k)}e^{2\pi ik\omega} \\ &= \sum_{k'=n+k} g(t - k')e^{2\pi im(t-k'+n)}e^{2\pi i(k'-n)\omega} = e^{2\pi imt}e^{-2\pi in\omega} (Zg)(t, \omega). \end{aligned}$$

This allows us to study the properties of the system based solely on the values of the Zak transform of g . Putting these two important facts together leads us to a theorem illustrating the utility of the Zak transform.

Theorem 1. *The Gabor system generated by $g \in L^2(\mathbb{R})$ is a frame for $L^2(\mathbb{R})$ with frame bounds $A, B > 0$ if and only if*

$$A \leq |(Zg)(t, \omega)|^2 \leq B$$

for almost every $(t, \omega) \in Q$.

Proof: This proof is due to Heil and Walnut [38]. Suppose that $\{g_{m,n}\}$ is a frame for $L^2(\mathbb{R})$ with frame bounds $A, B > 0$. Then we have for each $f \in L^2(\mathbb{R})$ that

$$A \|f\|_{L^2(\mathbb{R})}^2 \leq \sum_{m,n} |\langle f, g_{m,n} \rangle|^2 \leq B \|f\|_{L^2(\mathbb{R})}^2.$$

Since the Zak transform is an isometry we know that

$$\|f\|_{L^2(\mathbb{R})}^2 = \|Zf\|_{L^2(Q)}^2.$$

Furthermore, we can use the unitary property of the Zak transform to write

$$\sum_{m,n} |\langle f, g_{m,n} \rangle|^2 = \sum_{m,n} \left| \langle Zf, (Zg_{m,n}) \rangle_{L^2(Q)} \right|^2 = \sum_{m,n} \left| \langle Zf, e^{2\pi imt} e^{-2\pi in\omega} Zg \rangle_{L^2(Q)} \right|^2$$

$$= \sum_{m,n} \left| \langle Zf \cdot \overline{Zg}, e^{2\pi imt} e^{-2\pi in\omega} \rangle_{L^2(Q)} \right|^2 = \|Zf \cdot \overline{Zg}\|_{L^2(Q)}^2 \quad (4.1)$$

since the right part of the inner product in equation 4.1 is an orthonormal basis for $L^2(Q)$ and applying Parseval's theorem. Suppose that

$$|Zg|^2 < A$$

almost everywhere on a set $\Omega \subseteq Q$ of positive measure. Since Z is a surjective isometry, we can find an $f \in L^2(\mathbb{R})$ for which $Zf = F = 1$ on Ω and zero everywhere else. Then we would have

$$A m(\Omega) = A \|F\|_{L^2(Q)}^2 \leq \int_1^2 \int_1^2 |FZg|^2 dm = \int \int_{\Omega} |Zg|^2 dm < A m(\Omega),$$

clearly a contradiction. A similar contradiction arises if you assume that $|Zg|^2 > B$ almost everywhere on a set of positive measure. Thus we have the forward direction.

Now suppose that $A \leq |Zg|^2 \leq B$ almost everywhere on Q . We easily prove this direction by noting that for $f \in L^2(\mathbb{R})$

$$\begin{aligned} A \|f\|_{L^2(\mathbb{R})}^2 &= A \|Zf\|_{L^2(Q)}^2 \leq \|Zf \overline{Zg}\|_{L^2(Q)}^2 \\ &= \sum_{m,n} |\langle f, g_{m,n} \rangle|^2 \\ &= \|Zf \overline{Zg}\|_{L^2(Q)}^2 \leq B \|Zf\|_{L^2(Q)}^2 = B \|f\|_{L^2(\mathbb{R})}^2. \end{aligned}$$

□

This result is a striking example of the power the Zak transform provides when analyzing Gabor systems. A short list of results obtained similarly from Theorem 1 and can be summarized in the following theorem.

Theorem 2. Let $g \in L^2(\mathbb{R})$.

- A. The Gabor system generated by g is complete in $L^2(\mathbb{R})$ if and only if $Zg \neq 0$ a.e. on Q .
- B. The Gabor system generated by g is an orthonormal basis of $L^2(\mathbb{R})$ if and only if $|Zg| = 1$ a.e. on Q .

Even though we have defined the Zak transform onto $L^2(Q)$, there is a lot of analytical information that we can gain by extending the transform to be defined on all of \mathbb{R}^2 . We can achieve this by using the pseudo-periodicity of the Zak transform. Namely, for any $g \in L^2(\mathbb{R})$,

$$\begin{aligned} Zg(t+1, \omega) &= \sum_{k \in \mathbb{Z}} g(t+1-k) e^{2\pi i k \omega} = e^{2\pi i \omega} \sum_{k' = k-1 \in \mathbb{Z}} g(t-k') e^{2\pi i k' \omega} \\ &= e^{2\pi i \omega} Zg(t, \omega), \end{aligned}$$

$$Zg(t, \omega+1) = \sum_{k \in \mathbb{Z}} g(t-k) e^{2\pi i k (\omega+1)} = e^{2\pi i k} \sum_{k \in \mathbb{Z}} g(t-k) e^{2\pi i k \omega} = Zg(t, \omega).$$

The Zak transform is 1-periodic by translation in the variable ω , and is a character away from 1-periodicity by translation in the t variable. Using these relations, we can easily extend the Zak transform to all of \mathbb{R}^2 . An important result concerning this extension is the following famous theorem

Theorem 3. Suppose $g \in L^2(\mathbb{R})$ such that Zg is continuous on all of \mathbb{R}^2 . Then Zg has at least one zero.

In this critical sampling case of Gabor systems, with the Zak transform extended and defined as above, and with this seemingly innocuous result about the continuity of an extended Zak transform forced to take a zero, we arrive at the important Balian-Low obstruction theorem.

Theorem 4. *Suppose that $g \in L^2(\mathbb{R})$ and the Gabor system generated by g is a frame. Then either $xg \notin L^2(\mathbb{R})$ or $\gamma\hat{g} \notin L^2(\hat{\mathbb{R}})$.*

4.4 The Multiplicative Zak transform

In this section we shall introduce the multiplicative Zak transform created by Gertner and Tolimieri [35]. We shall modify the definition so that we have more flexibility in the types of functions we can analyze. It should be noted that the group structures underlying Gabor systems and wavelets are vastly different. Both are non-commutative, but the similarities essentially stop there. The Heisenberg group has many lattice sub-groups while the 2-dilation affine group essentially has very few. The Heisenberg group's representations over $L^2(\mathbb{R})$ (Schrödinger representations) have a beautiful duality through the Fourier transform. The affine group shares no such property. A further complication to extending the ideas from the Gabor setting to wavelets is that the Heisenberg group acts transitively while the affine group has three orbits, taken as actions over $L^2(\mathbb{R})$. These differences will show up many times as obstacles in defining a Zak transform that parallels that of the Gabor Zak transform.

A few papers have approached either the outright construction of a Zak trans-

form for wavelets (Gertner, Tolimieri [35] [1]) or have toyed with representations that include time and frequency with scale (Segman, Schempp [50]). We will consider the Gertner and Tolimieri paper in this section.

We are ultimately interested in the properties of the wavelet system generated by the function $\psi \in L^2(\mathbb{R})$, viz.,

$$\Lambda = \{\psi_{m,n}(t) = 2^{\frac{m}{2}}\psi(2^m t - n) \text{ for } m, n \in \mathbb{Z}\}.$$

Instead of working in the time domain, we are going to switch to the frequency domain for the remainder of this chapter. For ease of writing, θ will represent a generic function supported in $\hat{\mathbb{R}}$, and $\hat{\psi}$ is a specific Fourier transform of a wavelet generating function. Often $\hat{\psi} = \theta$. We can see that if Λ has ‘nice’ properties in $L^2(\mathbb{R})$, then that will also be true of the collection

$$\hat{\Lambda} = \{\hat{\psi}_{m,n}(\gamma) = e^{-2\pi i \gamma n 2^{-m}} 2^{-\frac{m}{2}} \hat{\psi}(2^{-m} \gamma)\} \text{ for } m, n \in \mathbb{Z}$$

in $L^2(\hat{\mathbb{R}})$.

The multiplicative Zak transform developed by Gertner and Tolimieri used dilation periodicity heavily in the framing of their arguments. They use this property and a property about fractional dilations being periodic under certain circumstances to prove results about frames generated by $\hat{\psi}$ that are supported in $(0, \infty)$. This line of reasoning actually gets in the way of extending the transform across $\gamma = 0$. We shall try a slightly different approach, one that allows the characterization of functions having Fourier transforms with more arbitrary support. The concept of dilation periodicity is still useful in constructing the transform as initially done.

Definition 8. We say that a locally square integrable function $\Phi \in L^2_{loc}((0, \infty))$ is dilation-periodic (DP) if

$$2^{\frac{1}{2}}\Phi(2\gamma) = \Phi(\gamma) \text{ for } \gamma > 0.$$

Note: If Φ is DP then it can be completely determined for $\gamma > 0$ by its values on $[1, 2)$ by the formula

$$\Phi(\gamma) = 2^{-\frac{k}{2}}\Phi(2^{-k}\gamma) \text{ for } 2^k \leq \gamma < 2^{k+1}.$$

Here we could replace the set $[1, 2)$ by E where $\{2^k E\}$ partitions $(0, \infty)$.

NB: We can identify a DP function as the periodic extension of a function defined strictly on the sets $[1, 2)$, or more generally E . The context will make domain of the function in question clear.

Definition 9. Let

$$L^2_{DP}((0, \infty)) = \{\Phi \in L^2_{loc}((0, \infty)) \mid \Phi \text{ is DP, } \int_1^2 |\Phi|^2 < \infty\}.$$

This is a Hilbert space with the expected inner product. It is useful to think of the spaces $L^2([1, 2))$ and L^2_{DP} as being the same. We will do most of our analysis in $L^2(\hat{\mathbb{R}})$, but occasionally will work in L^2_{DP} . To make this convention clear, we will use upper-case greek letters to specify when an object is considered part of $L^2([1, 2))$, and will use a tilde on such a letter to show it is a member of L^2_{DP} (technically defined on all of $(0, \infty)$, but determined on $[1, 2)$). The only distinction being that to retrieve $\tilde{\Psi} \in L^2_{DP}$ from $\Psi \in L^2([1, 2))$ requires merely 'periodizing' Ψ in the

following manner

$$\tilde{\Psi}(t) = 2^{-\frac{m}{2}} \Psi(2^{-m}t) \text{ when } t \in [2^m, 2^{m+1}).$$

Definition 10. *We will use the following ONBs of $L^2([1, 2])$*

$$\begin{aligned} \Psi_m(t) &= e^{2\pi i m 2^{-k} t} \text{ for } m \in \mathbb{Z} \\ \Phi_n(\omega) &= \sqrt{\frac{\log_2(e)}{\omega}} e^{2\pi i n \log_2(\omega)} \text{ for } n \in \mathbb{Z}. \end{aligned}$$

Both ONBs obey the rule

$$\Psi_o(\omega) \cdot \Psi_{m+k}(\omega) = \Psi_m(\omega) \cdot \Psi_k(\omega),$$

$$\Phi_o(\omega) \cdot \Phi_{m+k}(\omega) = \Phi_m(\omega) \cdot \Phi_k(\omega).$$

An additional property is that

$$|\Psi_k| = |\Psi_1| \text{ for all } k.$$

Note that

$$\tilde{\Psi}_m(t) = \begin{cases} e^{2\pi i m t} & t \in [1, 2) \\ 2^{-\frac{k}{2}} e^{2\pi i m 2^{-k} t} & t \in [2^k, 2^{k+1}) \end{cases}$$

is the periodized version of Ψ_m . So $\{\tilde{\Psi}_m\}$ and $\{\tilde{\Phi}_n\}$ are ONBs of L^2_{DP} . The definitions of $\{\Psi_m\}$ and $\{\Phi_n\}$ are meant to mimic the behavior of the basis functions from the Gabor Zak case, namely $\{e^{2\pi i n t}\}$ and $\{e^{2\pi i m \omega}\}$. Note that the logarithm is meant to treat dilations much as the Gabor functions treated translations.

The following is an orthonormal basis for $L^2((0, \infty))$ that comes up often in analyzing the multiplicative Zak transform. It is essentially half of a Shannon wavelet in the Fourier domain.

Definition 11. Let $g_{m,n}(\gamma) = 2^{-\frac{m}{2}} \mathbb{1}_{[2^m, 2^{m+1})}(\gamma) e^{2\pi i n 2^{-m} \gamma}$.

Proposition 2. $\{g_{m,n}\}$ is an ONB of $L^2((0, \infty))$.

Proof: Let $m, n, k, l \in \mathbb{Z}$. Then

$$\begin{aligned} \langle g_{m,n}, g_{k,l} \rangle_{L^2((0, \infty))} &= \int_0^\infty 2^{-\frac{(m+k)}{2}} \left(\mathbb{1}_{[2^m, 2^{m+1})}(t) \cdot \mathbb{1}_{[2^k, 2^{k+1})}(t) \right) \cdot \left(e^{2\pi i (n2^{-m} - l2^{-k})t} \right) dt \\ &= \int_0^\infty 2^{-\frac{(m+k)}{2}} \mathbb{1}_{[2^m, 2^{m+1}) \cap [2^k, 2^{k+1})}(t) e^{2\pi i (n2^{-m} - l2^{-k})t} dt = 0 \end{aligned}$$

when $m \neq k$. Suppose that $m = k$. Then we have that

$$\begin{aligned} \langle g_{m,n}, g_{k,l} \rangle_{L^2((0, \infty))} &= \int_0^\infty 2^{-m} \mathbb{1}_{[2^m, 2^{m+1})}(t) e^{2\pi i (n-l)2^{-m}t} dt \\ &= 2^{-m} \int_{2^m}^{2^{m+1}} e^{2\pi i (n-l)(2^{-m}t)} dt = \int_1^2 e^{2\pi i (n-l)t} dt. \end{aligned}$$

Clearly the above integral is zero when $n \neq l$ and is unity when $n = l$. Therefore $\{g_{m,n}\}$ is orthonormal.

It is well known that the collection $\{e^{2\pi i n t}\}_{n \in \mathbb{Z}}$ is an orthonormal basis of $L^2([1, 2])$. Therefore it is easy to see that for each fixed $m \in \mathbb{Z}$, the collection $\{2^{-\frac{m}{2}} \mathbb{1}_{[2^m, 2^{m+1})}(t) \cdot e^{2\pi i n 2^{-m} t}\}_{n \in \mathbb{Z}}$ is correspondingly an orthonormal basis of $L^2([2^m, 2^{m+1}))$. Thus, since elements having different m 's in the subscript have disjoint support, the collection $\{g_{m,n}\}$ is an orthonormal basis of $L^2(\cup_{m \in \mathbb{Z}} [2^m, 2^{m+1})) = L^2((0, \infty))$. \square

Note: For all $n \in \mathbb{Z}$

$$\tilde{\Psi}_n(t) = \sum_{k \in \mathbb{Z}} g_{k,n}(\omega).$$

Definition 12. Let $C_{[1,2)} = [1, 2) \times [1, 2)$. For $\theta \in L^2((0, \infty))$ define the multiplicative Zak transform $Z^* : L^2((0, \infty)) \rightarrow L^2(C_{[1,2)})$ by

$$(Z^*\theta)(t, \omega) = \sum_{k \in \mathbb{Z}} 2^{\frac{k}{2}} \theta(2^k t) \Phi_k(\omega).$$

Restricting the values of the multiplicative Zak transform to the set $C_{[1,2)}$ serves the same purpose as restricting the Gabor Zak transform to the set Q . However, unlike that case, there end up being many different sets for which this transform will be interesting. Gertner and Tolimieri work only with the set $C_{[1,2)}$ though, and so will we for the remainder of this small section.

Proposition 3. *The multiplicative Zak transform is a surjective isometry from $L^2((0, \infty))$ to $L^2(C_{[1,2)})$.*

Proof: We need only to look at how Z^* operates on the basis $\{g_{m,n}\}$.

$$\begin{aligned} (Z^*g_{m,n})(t, \omega) &= \sum_{k \in \mathbb{Z}} 2^{\frac{k}{2}} g_{m,n}(2^k t) \Phi_k(\omega) \\ &= \sum_{k \in \mathbb{Z}} 2^{\frac{k-m}{2}} \mathbb{1}_{[2^m, 2^{m+1})}(2^k t) e^{2\pi i n 2^{k-m} t} \Phi_k(\omega). \end{aligned}$$

Since t is restricted to $[1, 2)$, we have that $k = m$ and so

$$(Z^*g_{m,n})(t, \omega) = \mathbb{1}_{[1,2)}(t) e^{2\pi i n t} \Phi_m(\omega) = \Psi_n(t) \Phi_m(\omega).$$

□

Alternately, we can define the multiplicative Zak transform on any function through its action on the basis $\{g_{m,n}\}$. For example, given any $\theta \in L^2((0, \infty))$ we can write

$$\theta(t) = \sum_{k,l} \langle \theta, g_{k,l} \rangle g_{k,l}$$

thereby making the multiplicative Zak transform on θ as follows

$$(Z^*\theta)(t, \omega) := Z^* \left[\sum_{k,l} \langle \theta, g_{k,l} \rangle g_{k,l} \right] (t, \omega) = \sum_{k,l} \langle \theta, g_{k,l} \rangle \Phi_k(\omega) \Psi_l(t).$$

The multiplicative Zak transform enjoys the following properties, as a direct parallel to the Gabor Zak transform.

Proposition 4. *P1:*

$$2^{\frac{1}{2}} (Z^*\theta)(2t, \omega) = \frac{\Phi_o(\omega)}{\Phi_1(\omega)} \cdot (Z^*\theta)(t, \omega),$$

P2:

$$2^{\frac{1}{2}} (Z^*\theta)(t, 2\omega) = (Z^*\theta)(t, \omega),$$

P3:

$$\|(Z^*\theta)\|_{L^2_{DP}}^2 = \|\theta\|_{L^2([0, \infty))}^2.$$

Proof: P1: $2^{\frac{1}{2}} (Z^*\theta)(2t, \omega) = 2^{\frac{1}{2}} \sum_k 2^{\frac{k}{2}} \theta(2^k 2t) \Phi_k(\omega)$. Setting $l = k + 1$ we see that we have $\sum_{l \in \mathbb{Z}} 2^{\frac{l}{2}} \theta(2^l t) \Phi_{l-1}(\omega) \cdot \frac{\Phi_o(\omega)}{\Phi_1(\omega)} = \frac{\Phi_o(\omega)}{\Phi_1(\omega)} \sum_{l \in \mathbb{Z}} 2^{\frac{l}{2}} \theta(2^l t) \Phi_l(\omega) = \frac{\Phi_o(\omega)}{\Phi_1(\omega)} Z^*\theta(t, \omega)$.

P2: $2^{\frac{1}{2}} (Z^*\theta)(t, 2\omega) = \sum_{k \in \mathbb{Z}} 2^{\frac{k}{2}} \theta(2^k t) 2^{\frac{1}{2}} \cdot \Phi_k(2\omega) = \sum_{k \in \mathbb{Z}} 2^{\frac{k}{2}} \theta(2^k t) \Phi_k(\omega) = Z^*\theta(t, \omega)$.

P3: $\|Z^*\theta\|^2 = \int_1^2 \int_1^2 \left| \sum_{k \in \mathbb{Z}} 2^{\frac{k}{2}} \theta(2^k t) \Phi_k(\omega) \right|^2$

$$\begin{aligned}
&= \sum_{k \in \mathbb{Z}} \sum_{m \in \mathbb{Z}} 2^{\frac{k+m}{2}} \int_1^2 \theta(2^k t) \overline{\theta(2^m t)} dt \int_1^2 \Phi_k(\omega) \cdot \overline{\Phi_m(\omega)} d\omega \\
&= \sum_{k \in \mathbb{Z}} 2^k \int_1^2 |\theta(2^k t)|^2 dt = \|\theta\|^2. \quad \square
\end{aligned}$$

Let

$$\theta_{m,n}(t) = 2^{-\frac{m}{2}} \theta(2^{-m}t) \frac{\Psi_n(t)}{\Psi_o(t)}.$$

Note that with the Ψ 's above, the $\theta_{m,n}$'s are exactly the elements of $\hat{\Lambda}$. Then the fourth, and most important property of the multiplicative Zak transform is

P4:

$$(Z^* \theta_{m,n})(t, \omega) = \frac{\Phi_m(\omega)}{\Phi_o(\omega)} \cdot \frac{\Psi_n(t)}{\Psi_o(t)} \cdot (Z^* \theta)(t, \omega).$$

Unfortunately P4 is not always true. In fact, this is one of the strongest obstacles to utilizing the multiplicative Zak transform to analyze wavelet sets. This result holds true for any $\theta \in L^2(0, \infty)$ for which $\text{supp}(\theta) \subset [2^l, 2^{l+1})$ for any $l \in \mathbb{Z}$. The twisting action of the multiplicative Zak transform cannot overcome this obstruction in general because unlike the case of Gabor systems, the parameters do not sit on a regular lattice.

This can be seen clearly from the definition of the multiplicative Zak transform as the action on the basis $\{g_{m,n}\}$. Formally

$$g_{m,n} = \left(\sum_{k,l} \langle g, g_{k,l} \rangle g_{k,l} \right)_{m,n} = \sum_{k,l} \langle g, g_{k,l} \rangle (g_{k,l})_{m,n}.$$

We can write

$$(g_{k,l})_{m,n}(\gamma) = \left(2^{-\frac{k}{2}} \mathbb{1}_{[2^k, 2^{k+1})}(\gamma) e^{2\pi i l 2^{-k} \gamma} \right)_{m,n}$$

$$\begin{aligned}
&= e^{-2\pi i n 2^{-m} \gamma} 2^{-\frac{m}{2}} 2^{-\frac{k}{2}} \mathbb{1}_{[2^k, 2^{k+1})}(2^{-m} \gamma) e^{2\pi i l 2^{-k} (2^{-m} \gamma)} \\
&= 2^{-\frac{m+k}{2}} \mathbb{1}_{[2^{k+m}, 2^{k+m+1})}(\gamma) e^{-2\pi i 2^{-(m+k)} (n 2^k - l) \gamma} = g_{m+k, n 2^k - l}.
\end{aligned}$$

The factors representing dilations act additively, while the translation parameter has a non-commutative twist. The real problem is that each scale becomes wrapped up in the translation parameters (this is multi-resolution's forte) and they cannot be separated unless the original function has a very nice support. Putting this all together looks like

$$\begin{aligned}
\psi_{\hat{\psi}, m, n}(\gamma) &= \left(\sum_{k, l} \langle \hat{\psi}, g_{k, l} \rangle g_{k, l} \right)_{m, n}(\gamma) = \sum_{k, l} \langle \hat{\psi}, g_{k, l} \rangle (g_{k, l})_{m, n}(\gamma) \\
&= \sum_{k, l} \langle \hat{\psi}, g_{k, l} \rangle g_{k+m, n 2^k - l}(\gamma),
\end{aligned}$$

and thus

$$Z^* \left(\psi_{\hat{\psi}, m, n} \right) (t, \omega) = \sum_{k, l} \langle \hat{\psi}, g_{k, l} \rangle \Phi_{k+m}(\omega) \Psi_{n 2^k - l}(t).$$

Here it is impossible to neatly split these formal $\Psi_{n 2^k - l}$ apart, unless the original function $\hat{\psi}$ has support in some $[2^s, 2^{s+1})$ for some $s \in \mathbb{N}$ to force all of the dilations to be fixed and allowing the sum to split.

Gertner and Tolimieri constrain their functions to having support in the set $[1, 2^N)$ for some N and alter the dilations of their collection of functions to be fractional with denominator N . This allows them to prove the following classification

theorem, but at the cost of altering the collection of functions from being a true wavelet collection.

Theorem 5. (*Gertner, Tolimieri*) Let $\psi \in L^2(\mathbb{R})$ satisfy $\text{supp}(\hat{\psi}) \subseteq [1, 2^N]$. Then $\{\psi_{m,n}\}$ is a frame for the closed subspace it spans if and only if

$$\exists A, B > 0 \text{ such that } A \leq \sum_{j=0}^{N-1} \left| \hat{\psi}(2^{-j}t) \right|^2 \leq B, \quad \text{a.e. } t \in [1, 2).$$

4.4.1 Superframes and Extension

Part of the problem with the approach in the previous section, is that while analyzing wavelet collections of functions restricted to $(0, \infty)$ is interesting, ideally we want to be able to characterize wavelet frames for all of $L^2(\mathbb{R})$. This is simply not possible when restricted to one half of the real line. This small section will suggest a method that could be used to fuse together wavelets frames for the Hardy spaces $L^2(-\infty, 0)$ and $L^2(0, \infty)$ for the whole space $L^2(\mathbb{R}) = L^2(-\infty, 0) \oplus L^2(0, \infty)$. We shall apply a neat idea called superframes, and attempt to use this structure to glue the pieces together. This tool was developed by Balan in [3] to efficiently represent multiplexed signals.

Definition 13. Suppose that $F_1 = \{x_k^1\}, \dots, F_N = \{x_k^N\}$ are frames for the Hilbert spaces H_1, \dots, H_N where $H = H_1 \oplus \dots \oplus H_N$. We call the collection $F = F_1 \oplus \dots \oplus F_N = \{x_k^1 + \dots + x_k^N\}$ a superframe if it is a frame for the larger space H .

Theorem 6. I. The superset (F_1, \dots, F_N) is a superframe if and only if the following two conditions hold true:

a. F_k is a frame for H_k for $1 \leq k \leq N$,

b. $E_1 \oplus \dots \oplus E_N$ is a direct sum and closed subspace of $\ell^2(Z)$, i.e. $E_k \cap \left(\sum_{l \neq k} E_l \right) = \{0\}$ and $\sum_{k=1}^N E_k$ is closed.

II. Suppose that $F = (F_1, \dots, F_N)$ is a superframe. Then there is a superframe $(\tilde{F}_1, \dots, \tilde{F}_N)$ called the dual superframe of F such that the following reconstruction formula holds for every $f_1 \in H_1, \dots, f_N \in H_N$:

$$f_k = \sum_{n \in Z} \left(\sum_{l=1}^N \langle f_l, x_n^l \rangle \right) \tilde{x}_n^k = \sum_{n \in Z} \left(\sum_{l=1}^N \langle f_l, \tilde{x}_n^l \rangle \right) x_n^k.$$

Moreover, if $\tilde{E}_1, \dots, \tilde{E}_N$ denote the coefficient ranges of the component sets of \tilde{F} , then \tilde{F} is a dual superframe of F if and only if for every $1 \leq k \leq N$, \tilde{F}_k is a dual of F_k and \tilde{E}_k is orthogonal to every E_l with $l \neq k$.

A very natural question that arises in conjunction with this definition and the previous section is, are all wavelet frames for $L^2(\mathbb{R})$ wavelet superframes for the subspaces $L^2(-\infty, 0)$ and $L^2(0, \infty)$? The following theorem shows this to be an affirmative.

Theorem 7. Let $\psi \in L^2(\mathbb{R})$ such that $\{\psi_{m,n}\}$ is a wavelet frame for $L^2(\mathbb{R})$. Let $\psi^-(\gamma) = \hat{\psi}(\gamma) \mathbb{1}_{(-\infty, 0)}(\gamma)$ and $\psi^+(\gamma) = \hat{\psi}(\gamma) \mathbb{1}_{(0, \infty)}(\gamma)$ be projections onto $L^2(-\infty, 0)$ and $L^2(0, \infty)$. Then $\{\psi_{m,n}^-\}$ is a wavelet frame for $L^2(-\infty, 0)$ and $\{\psi_{m,n}^+\}$ is a wavelet frame for $L^2(0, \infty)$ and their sum $\{\psi_{m,n}^- + \psi_{m,n}^+ = \widehat{\psi_{m,n}}\}$ is a superframe for $L^2(\hat{\mathbb{R}})$.

Proof: Define

$$E_1 = \{c \in \ell^2(\mathbb{Z}^2) \mid c_{m,n} = \langle f, \psi_{m,n}^- \rangle \text{ for some } f \in L^2((-\infty, 0))\},$$

$$E_2 = \{c \in \ell^2(\mathbb{Z}^2) \mid c_{m,n} = \langle f, \psi_{m,n}^+ \rangle \text{ for some } f \in L^2((0, \infty))\}.$$

We need that $E_1 \cap E_2 = \{0\}$ and $E_1 + E_2$ is a closed subspace of $\ell^2(\mathbb{Z}^2)$.

Let $c \in E_1 \cap E_2$. Then we have that for some $g \in L^2((-\infty, 0))$ and $f \in L^2((0, \infty))$, for each $m, n \in \mathbb{Z}$

$$\int_{-\infty}^0 g(x) \overline{\widehat{\psi_{m,n}}(x)} dx = \langle g, \psi_{m,n}^- \rangle = c_{m,n} = \langle f, \psi_{m,n}^+ \rangle = \int_0^{\infty} f(x) \overline{\widehat{\psi_{m,n}}(x)} dx.$$

We can extend f, g to all of $L^2(\widehat{\mathbb{R}})$ by defining them to be zero off of their respective natural domains. We then have that

$$\int_{\mathbb{R}} (g(x) - f(x)) \overline{\widehat{\psi_{m,n}}(x)} dx = 0$$

for every $m, n \in \mathbb{Z}$. Since $\{\widehat{\psi_{m,n}}\}$ is a wavelet collection in $L^2(\widehat{\mathbb{R}})$, the only function that is orthogonal to all of these is the zero function. Therefore $g - f \equiv 0$. Since g and f have disjoint support, this forces $g = f = 0$ and therefore, $c \equiv 0$. Thus $E_1 \cap E_2 = \{0\}$.

Now we want to see if the sum $E_1 + E_2$ is a closed subspace of $\ell^2(\mathbb{Z}^2)$. This is in general not true, that is, there are closed sets A, B for which $A + B$ is not closed. For example consider $A = \mathbb{Z}$ and $B = \sqrt{2}\mathbb{Z}$. Then the sum $A + B$ is a dense subset of \mathbb{R} , and so is not closed. Now we will look at the specific sum $E_1 + E_2$. Take a sequence $\{c^k\}_{k \in \mathbb{Z}^2} \subset E_1 + E_2$ such that $c^k \rightarrow c$ in $\ell^2(\mathbb{Z}^2)$. This means that there is a collection $\{f^k\}_{k \in \mathbb{Z}^2} \subset L^2((-\infty, 0))$ and $\{g^k\}_{k \in \mathbb{Z}^2} \subset L^2((0, \infty))$ such that

$$c_{m,n}^k = x_{m,n}^k + y_{m,n}^k = \langle f^k, \psi_{m,n}^- \rangle + \langle g^k, \psi_{m,n}^+ \rangle.$$

Given $\epsilon > 0$, there exists a $K \in \mathbb{N}$ such that for all $k > K$ we have

$$\|c^k - c\|_{\ell^2(\mathbb{Z}^2)}^2 = \sum_{m,n} |c_{m,n}^k - c_{m,n}|^2 = \sum_{m,n} \left| \langle f^k + g^k, \widehat{\psi}_{m,n} \rangle - c_{m,n} \right|^2 < \epsilon.$$

Since the collection $\{\widehat{\psi}_{m,n}\}$ is a wavelet collection in $L^2(\widehat{\mathbb{R}})$, we know that for fixed k , we can represent $f^k + g^k$ by

$$f^k + g^k = \sum_{m,n} \langle f^k + g^k, \widehat{\psi}_{m,n} \rangle S^{-1} \left(\widehat{\psi}_{m,n} \right) = \sum_{m,n} c_{m,n}^k S^{-1} \left(\widehat{\psi}_{m,n} \right),$$

where S is the frame operator for the wavelet collection. Since the collection $\{c^k\}$ converges in the $\ell^2(\mathbb{Z}^2)$ sense, we have that the collection $f^k + g^k$ converges in the $L^2(\widehat{\mathbb{R}})$ sense, to a function $h \in L^2(\widehat{\mathbb{R}})$. We must show that

$$h = \sum_{m,n} c_{m,n} S^{-1} \left(\widehat{\psi}_{m,n} \right) = f + g,$$

for some $f \in L^2((-\infty, 0))$ and $g \in L^2((0, \infty))$. If this is true, then we would have that $c_{m,n} = \langle h, \widehat{\psi}_{m,n} \rangle = \langle h^-, \psi_{m,n}^- \rangle + \langle h^+, \psi_{m,n}^+ \rangle$ for each $m, n \in \mathbb{Z}$.

Let $\epsilon > 0$ be given. Since $g^k + f^k \rightarrow h$ in $L^2(\widehat{\mathbb{R}})$ as $k \rightarrow \infty$, there exists $K_1 \in \mathbb{N}$ for which $\|h - (f^k + g^k)\| < \frac{\epsilon}{2}$ for all $k > K_1$. Similarly, since $c^k \rightarrow c$ in $\ell^2(\mathbb{Z}^2)$ as $k \rightarrow \infty$ we have that there exists $K_2 \in \mathbb{N}$ for which $\|c^k - c\|_{\ell^2(\mathbb{Z}^2)} < \frac{\epsilon A}{2\|\widehat{\psi}\|_{L^2(\widehat{\mathbb{R}})}}$, for all $k > K_2$, where A is the frame's lower bound. Let $k > K = \max\{K_1, K_2\}$. Then we have that

$$\left\| h - \sum_{m,n} c_{m,n} S^{-1} \left(\widehat{\psi} \right) \right\| \leq \|h - (f^k + g^k)\| + \left\| \sum_{m,n} (c_{m,n}^k - c_{m,n}) S^{-1} \left(\widehat{\psi} \right) \right\|$$

$$\leq \frac{\epsilon}{2} + \left\| S^{-1}(\hat{\psi}) \right\| \cdot \sum_{m,n} |c_{m,n}^k - c_{m,n}|^2 \leq \frac{\epsilon}{2} + \frac{\left\| \hat{\psi} \right\|}{A} \frac{\epsilon A}{2 \left\| \hat{\psi} \right\|} = \epsilon.$$

Therefore we must have $h = \sum_{m,n} c_{m,n} S^{-1}(\hat{\psi})$. But this means that since $h = f + g = h \mathbb{1}_{(-\infty, 0)} + h \mathbb{1}_{(0, \infty)}$ where $f \in L^2((-\infty, 0))$ and $g \in L^2((0, \infty))$, then we have shown that $c^k \rightarrow c = a + b$ where $a \in E_1$ and $b \in E_2$. Therefore $E_1 + E_2$ is a closed set. \square

This result generates a lot of hope for being able to analyze wavelets with the multiplicative Zak transform. The schematic for doing so appears to be a straightforward process. Start with some $\psi \in L^2(\mathbb{R})$.

1. Analyze ψ^- , the projection of $\hat{\psi}$ onto $L^2(-\infty, 0)$, with a reflected version of the multiplicative Zak transform and determine if it is a frame for that subspace.
2. Analyze ψ^+ , the projection of $\hat{\psi}$ onto $L^2(0, \infty)$, with the multiplicative Zak transform and determine if it is a frame for that subspace.
3. Evaluate some ‘gluing’ condition for which $\{\widehat{\psi}_{m,n} = \psi_{m,n}^- + \psi_{m,n}^+\}$ is a superframe for $L^2(\hat{\mathbb{R}})$.

This presents two natural challenges. The first challenge is to understand when we can say $\psi_{m,n}^+$ is a frame for $L^2(0, \infty)$ simply by analyzing the values, or other properties of $Z^* \psi^+$, even when $\hat{\psi}$ is not compactly supported. Eventually we would like to solve the following conjecture

Conjecture 1. *Let $\psi \in L^2(\mathbb{R})$ satisfy $\text{supp}(\hat{\psi}) \subseteq (0, \infty)$, not necessarily compact.*

Then $\{\psi_{m,n}\}$ is a frame for the closed subspace it spans if and only if

$$\exists A, B > 0 \text{ such that } A \leq \left| Z^* \hat{\psi} \right|^2 \leq B, \quad a.e. (t, \omega) \in [1, 2)^2.$$

The second challenge is that of finding a gluing condition that allows us to put together two distinct frames, match their parameters and have a superframe for the larger space. An example of this gluing for a specific function (Shannon's wavelet) can be found in [22] pages 67 - 73, along with a collection of well-known estimates for wavelets frames, for which we believe the above three steps will be comparable to. This problem is still open.

4.4.2 The Multiplicative Zak Transform on Wavelet Sets

The results of the previous section are a little unsettling. The affine group forces us to analyze one side of $\hat{\mathbb{R}}$ at a time, and to only do so on special sets. This lack of freedom is in stark contrast to that of the Gabor Zak transform, and therefore we wish to extend the multiplicative Zak transform in some way so that we can generate results resembling that ideal case. One simple way of modifying the multiplicative Zak transform is by allowing ourselves to get away from the target set $C_{[1,2)} = [1, 2) \times [1, 2)$, and allowing a more general analyzing set to be $C_E = E \times [1, 2)$ where E is a wavelet set. An excellent introduction on the theory of wavelet sets can be found in Hernández and Weiss [39].

Definition 14. (*Wavelet Set*) The set $E \subset \hat{\mathbb{R}}$ is a wavelet set if

$$\psi(t) = \mathcal{FT}^{-1} \mathbb{1}_E(t)$$

generates an orthonormal wavelet basis for $L^2(\mathbb{R})$, where \mathcal{FT}^{-1} is the inverse Fourier transform.

Wavelet frames have a rich theory and a variety of properties; they are a tool for which wavelets and geometry can be closely linked. A few key properties of wavelets sets are

Proposition 5. *Suppose that E is a wavelet set. Then the following are true:*

- A. $\{E + n\}_{n \in \mathbb{Z}}$ partitions $\hat{\mathbb{R}}$,
- B. $\{2^m E\}_{m \in \mathbb{Z}}$ partitions $\hat{\mathbb{R}}$,
- C. \exists bijection $\sigma_E : E \rightarrow [-2, -1) \cup [1, 2)$, characterizing the wavelets set.

Definition 15. *Let E be a wavelet set. Let $\theta \in L^2(\hat{\mathbb{R}})$, we define the multiplicative E -Zak transform to be*

$$Z_E^* \theta(t, \omega) = \sum_{k \in \mathbb{Z}} 2^{\frac{k}{2}} \theta(2^k t) \Phi_k(\omega)$$

where $(t, \omega) \in C_E = E \times [1, 2)$.

The multiplicative E -Zak transform is useful because it shares all of the properties of the multiplicative Zak transform, but has a lot more flexibility in the star-crossed fourth property.

Proposition 6. *Let E be a wavelet set. Then the following are true*

- Z_E^* is a surjective isometry from $L^2(\hat{\mathbb{R}})$ to $L^2(C_E)$,

P1:

$$2^{\frac{1}{2}} (Z_E^* \theta) (2t, \omega) = \frac{\Phi_o(\omega)}{\Phi_1(\omega)} \cdot (Z_E^* \theta) (t, \omega),$$

P2:

$$2^{\frac{1}{2}} (Z_E^* \theta) (t, 2\omega) = (Z_E^* \theta) (t, \omega),$$

P3:

$$\|(Z_E^* \theta)\|_{L_{DP}^2}^2 = \|\theta\|_{L^2([0, \infty))}^2.$$

Proof: The result follows from the original multiplicative Zak transform proofs combined with the dilation partition property of the wavelet set E . \square

This gives us a great multitude of multiplicative Zak transforms to work with, to analyze wavelets with. We probably do not need to force the wavelet set structure on ourselves, since we are really only utilizing the dilation partition property of such sets. However, for the time being, we will continue to hold this frame of reference. Also notice that we are not restricted to one half-line of $\hat{\mathbb{R}}$. Note that by freeing ourselves of the static set $C_{[1,2]}$ we can attain the following proposition.

Proposition 7. *Let $\psi \in L^2(\mathbb{R})$ such that $\text{supp}(\hat{\psi}) \subset E$ where E is some wavelet set. Then the following property is true.*

P4:

$$(Z_E^* \theta_{m,n}) (t, \omega) = \frac{\Phi_m(\omega)}{\Phi_o(\omega)} \cdot \frac{\Psi_n(t)}{\Psi_o(t)} \cdot (Z_E^* \theta) (t, \omega).$$

Proof: Let $(t, \omega) \in C_E$. Then

$$(Z_E^* \theta_{m,n}) (t, \omega) = \sum_{k \in \mathbb{Z}} 2^{\frac{k}{2}} \theta_{m,n}(2^k t) \Phi_k(\omega) = \sum_{k \in \mathbb{Z}} 2^{\frac{k}{2}} 2^{-\frac{m}{2}} \theta(2^{k-m} t) e^{-2\pi i n 2^{-m} t} \Phi_k(\omega).$$

Since $t \in E$, and since $\text{supp}(\theta) \subset E$, we must have the summands are zero except when $k = m$. Thus we see

$$\begin{aligned} \sum_{k \in \mathbb{Z}} 2^{\frac{k}{2}} 2^{-\frac{m}{2}} \theta(2^{k-m}t) e^{-2\pi i n 2^{k-m}t} \Phi_k(\omega) &= e^{-2\pi i n t} \Phi_m(\omega) Z_E^* \theta(t, \omega) \\ &= \frac{\Phi_m(\omega)}{\Phi_o(\omega)} \cdot \frac{\Psi_n(t)}{\Psi_o(t)} \cdot (Z_E^* \theta)(t, \omega). \end{aligned}$$

□

Having this fourth property of the multiplicative Zak transform be true for a wider variety of wavelets than originally possible allows us to resurrect the following theorem, which parallels a similar theorem in the Gabor Zak case.

Theorem 8. *Suppose that $\psi \in L^2(\mathbb{R})$ such that $\{\hat{\psi}_{m,n}\}_{m,n \in \mathbb{Z}}$ is a (Fourier-side) wavelet frame for $L^2(\hat{\mathbb{R}})$ with frame bounds $A, B > 0$ and $\text{supp}(\hat{\psi}) \subset E$ for some wavelet set E . Then $\{Z_E^* g_{m,n}\}_{m,n \in \mathbb{Z}}$ is a frame for $L^2(C_E)$ with the same bounds. Furthermore, there exist $0 < A_o \left(= \frac{\log_2(e)A}{2} < A \right), (B <) B_o (= \log_2(e)B) < \infty$ such that*

$$A_o \leq \left| Z_E^* \hat{\psi} \right|^2 \leq B_o$$

a.e. for $(t, \omega) \in C_E$.

Proof: Let $F \in L^2(C_E)$. Since Z_E^* is surjective, there is an $f \in L^2(\hat{\mathbb{R}})$ such that $Z_E^* f = F$ a.e. on C_E . The following calculation justifies the first claim in the theorem:

$$\sum_{m,n \in \mathbb{Z}} \left| \left\langle F, Z_E^* \left(\hat{\psi}_{m,n} \right) \right\rangle_{L^2(C_E)} \right|^2 = \sum_{m,n \in \mathbb{Z}} \left| \left\langle Z_E^* f, Z_E^* \left(\hat{\psi}_{m,n} \right) \right\rangle \right|^2 = \sum_{m,n \in \mathbb{Z}} \left| \left\langle f, \hat{\psi}_{m,n} \right\rangle_{L^2(\hat{\mathbb{R}})} \right|^2,$$

since we know that for any $f \in L^2(\hat{\mathbb{R}})$,

$$A \|f\|_{L^2(\hat{\mathbb{R}})}^2 \leq \sum_{m,n \in \mathbb{Z}} \left| \left\langle f, \hat{\psi}_{m,n} \right\rangle_{L^2(\hat{\mathbb{R}})} \right|^2 \leq B \|f\|_{L^2(\hat{\mathbb{R}})}^2$$

and we know that for all $f \in L^2(\hat{\mathbb{R}})$, $\|f\|_{L^2(\hat{\mathbb{R}})}^2 = \|Z_E^* f\|_{L^2(C_E)}^2$, we can conclude that

$$A \|F\|_{L^2(C_E)}^2 \leq \sum_{m,n \in \mathbb{Z}} \left| \left\langle F, Z_E^* \left(\hat{\psi}_{m,n} \right) \right\rangle \right|^2 \leq B \|F\|_{L^2(C_E)}^2 \quad (4.2)$$

for all $F \in L^2(C_E)$. This means that $\{Z_E^* \left(\hat{\psi}_{m,n} \right)\}_{m,n \in \mathbb{Z}}$ is a frame for $L^2(C_E)$ with bounds $A, B > 0$.

To show the second conclusion of the theorem, consider again the inequality in (4.2). Note that

$$\left\langle F, Z_E^* \left(\hat{\psi}_{m,n} \right) \right\rangle = \left\langle F, Z_E^* \hat{\psi} \cdot \frac{\Phi_m \cdot \Psi_n}{\Phi_o \cdot \Psi_o} \right\rangle = \left\langle F \cdot \left(\frac{Z_E^* \hat{\psi}}{\Phi_o \cdot \Psi_o} \right)^*, \Phi_m \cdot \Psi_n \right\rangle. \quad (4.3)$$

Here the $*$ outside of the parenthesis represents the complex conjugate. Since we know that $\{\Phi_m(\omega) \cdot \Psi_n(t)\}_{m,n \in \mathbb{Z}}$ is an ONB of $L^2(C_E)$, combining (4.2) with (4.3) yields:

$$\begin{aligned} A \|F\|_{L^2(C_E)}^2 &\leq \left\| F \cdot \left(\frac{Z_E^* \hat{\psi}}{\Phi_o \cdot \Psi_o} \right)^* \right\|_{L^2(C_E)}^2 \\ &= \sum_{m,n \in \mathbb{Z}} \left| \left\langle F \cdot \left(\frac{Z_E^* \hat{\psi}}{\Phi_o \cdot \Psi_o} \right)^*, \Phi_m \cdot \Psi_n \right\rangle_{L^2(C_E)} \right|^2 \leq B \|F\|_{L^2(C_E)}^2 \end{aligned}$$

for all $F \in L^2(C_E)$. This shows that

$$\inf_{(t,\omega) \in C_E} \left| \frac{Z_E^* \hat{\psi}}{\Phi_o \cdot \Psi_o} \right|^2 \geq A,$$

$$\sup_{(t,\omega) \in C_E} \left| \frac{Z_E^* \hat{\psi}}{\Phi_o \cdot \Psi_o} \right|^2 \leq B.$$

Since $\Psi_o \equiv 1$ and since $\Phi_o(\omega) = \sqrt{\frac{\log_2(e)}{\omega}}$, we have that

$$\frac{2}{\log_2(e)} \cdot \inf_{(t,\omega) \in C_E} \left| Z_E^* \hat{\psi} \right|^2 \geq \inf_{(t,\omega) \in C_E} \left| \frac{Z_E^* \hat{\psi}}{\Phi_o \cdot \Psi_o} \right|^2 \geq A,$$

$$\frac{1}{\log_2(e)} \cdot \sup_{(t,\omega) \in C_E} \left| Z_E^* \hat{\psi} \right|^2 \leq \sup_{(t,\omega) \in C_E} \left| \frac{Z_E^* \hat{\psi}}{\Phi_o \cdot \Psi_o} \right|^2 \leq B.$$

This means that

$$A_o \left(= \frac{\log_2(e)A}{2} \right) \leq \left| Z_E^* \hat{\psi} \right|^2 \leq (\log_2(e)B =) B_o$$

a.e. $(t, \omega) \in C_E$. □

This result immediately lets us write down a theorem that is a modification of the conjecture in the previous section.

Theorem 9. *Let $\psi \in L^2(\mathbb{R})$ satisfy $\text{supp}(\hat{\psi}) \subseteq E$ for some wavelet set E . Then $\{\psi_{m,n}\}$ is a frame for $L^2(\mathbb{R})$ if and only if*

$$\exists A, B > 0 \text{ such that } A \leq \left| Z_E^* \hat{\psi} \right|^2 \leq B, \quad \text{a.e. } (t, \omega) \in C_E.$$

Note a theorem which resembles the above result, in a slightly different context.

This is taken from [22], page 973.

Theorem 10. *If the $\psi_{m,n}(t) = 2^{\frac{m}{2}} \psi(2^m t - n)$ for $m, n \in \mathbb{Z}$, is a frame for $L^2(\mathbb{R})$ with bounds $A, B > 0$, then*

$$\frac{\ln 2}{2} A \leq \int_0^\infty \frac{|\hat{\psi}(\gamma)|^2}{\gamma} d\gamma \leq \frac{\ln 2}{2} B,$$

$$\frac{\ln 2}{2} A \leq \int_{-\infty}^0 \frac{|\hat{\psi}(\gamma)|^2}{\gamma} d\gamma \leq \frac{\ln 2}{2} B.$$

This shares the bounds with our frame test, and also the ability for a frame on one half, and a frame on another, to not necessarily combine to form a superframe for the whole set.

Note: Unfortunately, while this theorem seems like a large improvement over the conjectures and theorems of Gertner and Tolimieri, we must point out that in the cases we are working with, that is, $\hat{\psi} \subset E$ where E is a wavelet set, the multiplicative E -Zak transform is identity on such functions. That means studying the wavelet generated by $\psi \in L^2(\mathbb{R})$ for which $\text{supp}(\hat{\psi}) \subset E$ for a wavelet set E , amounts to simply checking that $|\hat{\psi}|^2$ is bounded above and below. This is a special case from [20], problem **12.1** on page 281. With this in mind, we wish to test part of this multiplicative Zak transform to see if it can detect the simplest kinds of wavelet frames. Namely, those that are orthonormal bases for $L^2(\mathbb{R})$. That means we should be able to detect the following simple conditions for an orthonormal wavelet basis, since these conditions partially classify such collections of functions. For the rest of this chapter, we will switch the use of the variable t in the multiplicative Zak transform, to γ to make the Fourier transforms looks more natural.

$$\sum_{j \in \mathbb{Z}} \left| \psi(2^j \gamma) \right|^2 = 1 \text{ a.e.}, \quad (4.4)$$

$$\sum_{j=1}^{\infty} \hat{\psi}(2^j \gamma) \overline{\hat{\psi}(2^j(\gamma + l))} = 0 \text{ for } l \in 2\mathbb{Z} + 1. \quad (4.5)$$

To deal with the first of these characteristic formulas, we define the alternate multiplicative E -Zak transform.

Definition 16. We define the alternative multiplicative Zak transform to be

$$\tilde{Z}_E^* \theta(\gamma, \omega) = \sum_{k \in \mathbb{Z}} 2^{-\frac{k}{2}} \theta(2^k \gamma) \Phi_k(\omega),$$

where $(t, \omega) \in C_E$.

Note that in general, we would not expect such a series to converge. Consider the following formal calculation

$$\begin{aligned} \int_1^2 Z_E^* \theta(\gamma, \omega) \overline{\tilde{Z}_E^* \theta(\gamma, \omega)} d\omega &= \int_1^2 \left(\sum_{k \in \mathbb{Z}} 2^{\frac{k}{2}} \theta(2^k \gamma) \Phi_k(\omega) \right) \left(\sum_{l \in \mathbb{Z}} 2^{-\frac{l}{2}} \overline{\theta(2^l \gamma) \Phi_l(\omega)} \right) d\omega \\ &= \sum_{k, l \in \mathbb{Z}} 2^{\frac{k-l}{2}} \theta(2^k \gamma) \overline{\theta(2^l \gamma)} \int_1^2 \Phi_k(\omega) \overline{\Phi_l(\omega)} d\omega = \sum_{k \in \mathbb{Z}} |\theta(2^k \gamma)|^2. \end{aligned}$$

This ‘Parseval-like’ equality seems to show that the information contained in equation 4.4 can be replicated with the multiplicative Zak transform and its alternate form. We can make this precise with the following.

Theorem 11. Let $\psi \in L^2(\mathbb{R})$. Then

$$\int_1^2 Z_E^* \theta(\gamma, \omega) \overline{\tilde{Z}_E^* \theta(\gamma, \omega)} d\omega = \sum_{k \in \mathbb{Z}} |\theta(2^k \gamma)|^2$$

almost everywhere $\gamma \in E$.

Proof: For $N \in \mathbb{N}$, we define the function

$$F_N(\gamma, \omega) = \sum_{k=-N}^N 2^{-\frac{k}{2}} \hat{\psi}(2^k \gamma) \Phi_k(\omega).$$

Clearly $F_N \in L^2(E \times [1, 2))$ for every $N \in \mathbb{N}$. Since $\psi \in L^2(\mathbb{R})$, we know that $Z^* \hat{\psi} \in L^2(E \times [1, 2))$. We can then compute

$$\begin{aligned}
\int_1^2 Z_E^* \hat{\psi}(\gamma, \omega) \overline{F_N(\gamma, \omega)} d\omega &= \sum_{k \in \mathbb{Z}} \sum_{l=-N}^N 2^{\frac{k-l}{2}} \hat{\psi}(2^k \gamma) \overline{\hat{\psi}(2^l \gamma)} \int_1^2 \Phi_k(\omega) \overline{\Phi_l(\omega)} d\omega \\
&= \sum_{k=-N}^N \left| \hat{\psi}(2^k \gamma) \right|^2
\end{aligned}$$

for every $N \in \mathbb{N}$. □

Proposition 8. *Suppose that $\text{supp}(\hat{\psi}) \cap (-\eta, \eta) = \emptyset$ for some $\eta > 0$. Furthermore, assume that $\hat{\psi} \in L^\infty(\mathbb{R})$ (or some mild decay towards infinity). Then $\{F_N(\gamma, \omega)\}$ is a Cauchy sequence and $F_N \rightarrow \tilde{Z}_E^* \hat{\psi}$ pointwise in C_E .*

Proof: Suppose that $N > M$ and consider

$$\begin{aligned}
\int_E \int_1^2 |F_N(\gamma, \omega) - F_M(\gamma, \omega)|^2 d\omega d\gamma &= \int_E \int_1^2 \left| \sum_{|k|=M+1}^N 2^{-\frac{k}{2}} \hat{\psi}(2^k \gamma) \Phi_k(\omega) \right|^2 d\omega d\gamma \\
&= \int_E \int_1^2 \left(\sum_{|k|=M+1}^N 2^{-\frac{k}{2}} \hat{\psi}(2^k \gamma) \Phi_k(\omega) \right) \overline{\left(\sum_{|l|=M+1}^N 2^{-\frac{l}{2}} \hat{\psi}(2^l \gamma) \Phi_l(\omega) \right)} d\omega d\gamma \\
&= \sum_{|k|, |l|=M+1}^N 2^{-\frac{(k+l)}{2}} \int_E \hat{\psi}(2^k \gamma) \overline{\hat{\psi}(2^l \gamma)} d\gamma \int_1^2 \Phi_k(\omega) \overline{\Phi_l(\omega)} d\omega \\
&= \sum_{|k|=M+1}^N 2^{-k} \int_E \left| \hat{\psi}(2^k \gamma) \right|^2 d\gamma.
\end{aligned}$$

Assuming that our wavelet set E is not pathological, we can safely assume that there exists a $K \in \mathbb{N}$ such that if $k > K$, $2^{-k}E \subset (-\eta, \eta)$ and consequently,

$$\begin{aligned}
&= \sum_{k=-M-1}^{-N} 2^{-k} \int_E |\hat{\psi}(2^k \gamma)|^2 d\gamma + \sum_{k=M+1}^N 2^{-k} \int_E |\hat{\psi}(2^k \gamma)|^2 d\gamma \\
&= \sum_{k=M+1}^N 2^{-k} \int_E |\hat{\psi}(2^k \gamma)|^2 d\gamma \leq \sum_{k=M+1}^N 2^k \int_E |\hat{\psi}(2^k \gamma)|^2 d\gamma < \epsilon.
\end{aligned}$$

□

What are the implications of such a formalized calculation? Suppose that θ was the Fourier transform of ψ , and $\{\psi_{m,n}\}$ generated an orthonormal basis for $L^2(\mathbb{R})$. We know that if ψ satisfies equation 4.4, then we have that

$$\int_1^2 Z^* \hat{\psi}(\gamma, \omega) \overline{\tilde{Z}^* \hat{\psi}(\gamma, \omega)} d\omega = \sum_{k \in \mathbb{Z}} |\hat{\psi}(2^k \gamma)|^2 = 1.$$

Since we already know that $Z^* \hat{\psi}$ is well defined and in fact $L^2([1, 2]^2)$, this does say something about the values of the alternative multiplicative Zak transform.

The following is a formal attempt at trying to find the information contained in equation 4.5 from the multiplicative Zak transform and its alternate version. Let $q \in 2\mathbb{Z} + 1$ and $\gamma \in E$. Since E is a wavelet set, $\gamma + q \notin E$. Our periodicity conditions in the multiplicative Zak transform say nothing about additive periodicity. However, there exists a unique $l \in \mathbb{Z}$ and $\gamma_q \in E$ for which $\gamma + q = 2^l \gamma_q$. In this way we can calculate the formal inner product

$$\begin{aligned}
&\int_1^2 Z_E^* \hat{\psi}(\gamma, \omega) \overline{\tilde{Z}_E^* \hat{\psi}(\gamma + q, \omega)} d\omega \\
&= \int_1^2 \left(\sum_{k \in \mathbb{Z}} 2^{\frac{k}{2}} \hat{\psi}(2^k \gamma) \Phi_k(\omega) \right) \overline{\left(\sum_{m \in \mathbb{Z}} 2^{-\frac{m}{2}} \hat{\psi}(2^m(\gamma + q)) \Phi_m(\omega) \right)} d\omega
\end{aligned}$$

$$= \sum_{k,m \in \mathbb{Z}} 2^{\frac{k-m}{2}} \hat{\psi}(2^k \gamma) \overline{\hat{\psi}(2^m(\gamma + q))} \int_1^2 \Phi_k(\omega) \overline{\Phi_m(\omega)} d\omega = \sum_{k \in \mathbb{Z}} \hat{\psi}(2^k \gamma) \overline{\hat{\psi}(2^k(\gamma + q))}.$$

We can also write the expression as

$$\begin{aligned} \int_1^2 Z^* \hat{\psi}(\gamma, \omega) \overline{\tilde{Z}^* \hat{\psi}(\gamma + q, \omega)} d\omega &= \int_1^2 Z^* \hat{\psi}(\gamma, \omega) \overline{\tilde{Z}^* \hat{\psi}(2^l \gamma_q, \omega)} d\omega \\ &= \int_1^2 Z^* \hat{\psi}(\gamma, \omega) 2^{-\frac{l}{2}} \frac{\overline{\Phi_o(\omega)}}{\Phi_l(\omega)} \overline{\tilde{Z}^* \hat{\psi}(\gamma_q, \omega)} d\omega \end{aligned}$$

using our algebraic manipulation $\gamma + q = 2^l \gamma_q$ and the periodicity of the multiplicative Zak transform. We can take this extra term, $2^{-\frac{l}{2}} \frac{\overline{\Phi_o(\omega)}}{\Phi_l(\omega)}$ and note that in the case of our $\{\Phi_k\}$, this can be rewritten as $2^{-\frac{l}{2}} \frac{\Phi_o(\omega)}{\Phi_{-l}(\omega)}$, which will allow us to use the periodicity of Z^* to write

$$\int_1^2 Z^* \hat{\psi}(\gamma, \omega) 2^{-\frac{l}{2}} \frac{\overline{\Phi_o(\omega)}}{\Phi_l(\omega)} \overline{\tilde{Z}^* \hat{\psi}(\gamma_q, \omega)} d\omega = 2^{-l} \int_1^2 Z^* \hat{\psi}(2^{-l} \gamma, \omega) \overline{\tilde{Z}^* \hat{\psi}(\gamma_q, \omega)} d\omega.$$

4.4.3 Inversion Formula

Consider the Multiplicative E -Zak transform on the function $\hat{\psi} \in L^2(\hat{\mathbb{R}})$. Note that since the inner product on $L^2(\hat{\mathbb{R}})$ is defined for Schwartz functions and since the multiplicative E -Zak transform is an isometry onto $L^2(C_E)$ then we can extend the multiplicative E -Zak transform to act on tempered distributions. Given this fact, for $\gamma \in \hat{\mathbb{R}}$ we have

$$\hat{\psi}(\gamma) = \int_{-\infty}^{\infty} \hat{\psi}(s) \overline{\delta_\gamma(s)} ds = \langle \hat{\psi}, \delta_\gamma \rangle = \langle Z_E^* \hat{\psi}, Z_E^* \delta_\gamma \rangle.$$

Therefore, to compute the value of $\hat{\psi}(\gamma)$, it is imperative to calculate $Z_E^* \delta_\gamma$. Note that

$$(Z_E^* \delta_\gamma)(t, \omega) = \sum_{k \in \mathbb{Z}} 2^{-\frac{k}{2}} \delta_\gamma(2^k t) \Phi_k(\omega) = 2^{-\frac{m}{2}} \delta_\gamma(2^m t) \Phi_m(\omega),$$

where $\gamma \in 2^m E$. Given this we can calculate

$$\begin{aligned} \hat{\psi}(\gamma) &= \left\langle Z_E^* \hat{\psi}, Z_E^* \delta_\gamma \right\rangle = \int_1^2 \int_E Z_E^* \hat{\psi}(t, \omega) \overline{2^{-\frac{m}{2}} \delta_\gamma(2^m t) \Phi_m(\omega)} dt d\omega \\ &= 2^{\frac{m}{2}} \int_1^2 Z_E^* \hat{\psi}\left(\frac{\gamma}{2^m}, \omega\right) \overline{\Phi_m(\omega)} d\omega. \end{aligned}$$

This creative formula leads to the following inversion theorem.

Theorem 12. *Let $F \in L^2(C_E)$. Define the function*

$$\theta(\gamma) = \sum_{k \in \mathbb{Z}} \left(2^{-\frac{k}{2}} \int_1^2 F(2^{-k} \gamma, \omega) \overline{\Phi_k(\omega)} d\omega \right) \mathbb{1}_{2^k E}(\gamma).$$

Then $\theta \in L^2(\hat{\mathbb{R}})$ and

$$Z_E^* \theta = F \text{ a.e. } (t, \omega) \in C_E.$$

Proof: Let θ be defined as above. Then we can write the norm

$$\begin{aligned} \|\theta\|_{L^2(\hat{\mathbb{R}})}^2 &= \int_{-\infty}^{\infty} |\theta(\gamma)|^2 d\gamma = \sum_{k \in \mathbb{Z}} \int_{2^k E} |\theta(\gamma)|^2 d\gamma \\ &= \sum_{k \in \mathbb{Z}} \int_{2^k E} \left| 2^{-\frac{k}{2}} \int_1^2 F(\gamma 2^{-k}, \omega) \overline{\Phi_k(\omega)} d\omega \right|^2 d\gamma. \end{aligned}$$

For each $k \in \mathbb{Z}$, we substitute $u = 2^{-k} \gamma$ and obtain

$$= \sum_{k \in \mathbb{Z}} \int_E \left| \int_1^2 F(u, \omega) \overline{\Phi_k(\omega)} d\omega \right|^2 d\gamma.$$

Note that for a fixed $u_o \in E$, we can define

$$G_{u_o}(\omega) = \sum_{k \in \mathbb{Z}} \left(\int_1^2 F(u_o, \omega) \overline{\Phi_k(\omega)} d\omega \right) \Phi_k(\omega).$$

Since the collection $\{\Phi_k(\omega)\}$ is an orthonormal basis for the space $L^2([1, 2])$, we have that the norm of $G_{u_o}(\omega)$ is given by

$$\|G_{u_o}\|^2 = \sum_{k \in \mathbb{Z}} \left| \int_1^2 F(u_o, \omega) \overline{\Phi_k(\omega)} d\omega \right|^2.$$

This sum clearly converges for each $u_o \in [1, 2)$ and thus the sum converges uniformly on the interval $[1, 2)$. But this means that

$$\|\hat{g}\|^2 = \|F\|_{L^2([1, 2]^2)}^2,$$

$$\theta \in L^2(\hat{\mathbb{R}}).$$

Now we wish to calculate

$$\begin{aligned} Z_E^* \theta(t, \omega) &= \sum_{k \in \mathbb{Z}} 2^{\frac{k}{2}} \theta(2^k t) \Phi_k(\omega) = \sum_{k \in \mathbb{Z}} 2^{\frac{k}{2}} \left(2^{-\frac{k}{2}} \int_1^2 F((2^k t) 2^{-k}, s) \overline{\Phi_k(s)} ds \right) \Phi_k(\omega) \\ &= \sum_{k \in \mathbb{Z}} \left(\int_1^2 F(t, s) \overline{\Phi_k(s)} ds \right) \Phi_k(\omega) = G_t(\omega) = F(t, \omega) \end{aligned}$$

for almost all $(t, \omega) \in C_E$.

Example: Consider the function $F(t, \omega) = \sqrt{\frac{\log_2(e)}{\omega}}$ on $C_{[1, 2]}$. Inverting yields

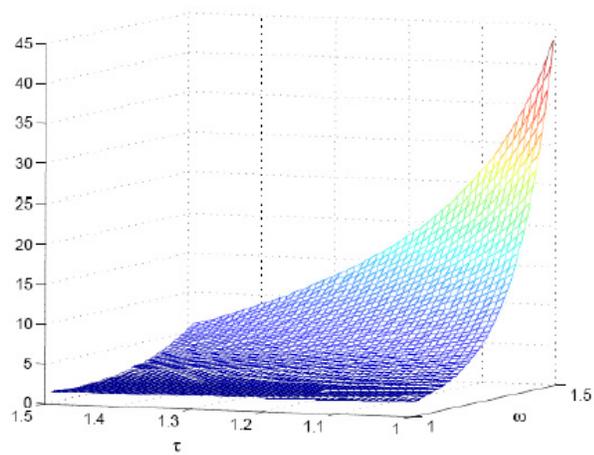
$$\theta(\gamma) = \mathbb{1}_{[1, 2]}(\gamma),$$

where $Z^* \theta(t, \omega) = F(t, \omega)$ almost everywhere on $C_{[1, 2]}$.

Example: Now let θ be the Mexican hat function. That is, $\theta(\gamma) = \gamma^2 e^{-\frac{\gamma^2}{2}}$. Then

we can estimate the values of $|Z^* \theta(t, \omega)|^2$ on $C_{[1, 2]} \cap [1, 1.5]^2$ by

Figure 4.1: Estimate of the MZT of the MHF.



Bibliography

- [1] L. Auslander, G. Eichmann, I. Gertner and R. Tolimieri, "Time-Frequency Analysis and Synthesis of Non-Stationary Signals," Proc. Soc. Photo-Opt. Instrum. Eng. **SPIE1152**, 449 (1989).
- [2] C.M. Bachmann, T.L. Ainsworth and R.A. Fusina, "Improved manifold coordinate representations of large scale hyperspectral imagery," IEEE Trans. on Geoscience and Remote Sensing, **44**(10), 2786 (2006).
- [3] R. Balan, "Multiplexing of Signals using Superframes," Wavelets Applications. in Signal Processing VIII. **4119**, 118 (2000).
- [4] A. Banerjee, P. Burlina, J. Broadwater, "A machine learning approach for finding hyperspectral endmembers," IEEE International Geoscience and Remote Sensing Symposium, 3817 (2007.)
- [5] M. Belkin and P. Niyogi, "Towards a Theoretical Foundation for Laplacian-Based Manifold Methods," **COLT**, 486 (2005).
- [6] M. Belkin and P. Niyogi, "Convergence of Laplacian Eigenmaps," Preprint, (2008).
- [7] J.J. Benedetto, W. Czaja, M. Ehler and C. Flake, "Wavelet packets for multi-and hyperspectral imagery," in *Wavelet Applications in Industrial Processing VII.*, eds. F. Truchetet, and O. Laligant. Proc. SPIE, **7535**, 753508 (2010).
- [8] J.J. Benedetto, W. Czaja, J.C. Flake and M. Hirn, "Frame based kernel methods for automatic classification in hyperspectral data," IEEE International Geoscience and Remote Sensing Symposium, (2009).
- [9] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. Le Roux and M. Ouimet, "Out-of-sample extensions for lle, isomap, mdp, eigenmaps and spectral clustering," Advances in Neural Information Processing Systems, **16**, 177 (2004).
- [10] J. Boardman, F. Kruse and R. Green, "Mapping target signatures via partial unmixing of aviris data," Fifth JPL Airborne Earth Science Workshop, **1** pf JPL Publication 95-1, 23 (1995).
- [11] J. Bowles, P. Palmadesso, J. Antoniadis, M. Baumbeck and L. Rickard, "Use of filter vectors in hyperspectral data analysis," Proc. SPIE, **2553**, 148 (1995).

- [12] C.J.C Burges, “Dimension Reduction: A guided tour,” MSR Tech Report MSR-TR-2009-2013, (2009).
- [13] E.J. Candès and D.L. Donoho, “New Tight Frames of Curvelets and Optimal Representations of Objects with Piecewise C^2 Singularities,” *Comm. Pure and Applied Math.* **LVII**, 0219, (2004).
- [14] R.R. Coifman and S. Lafon, “Diffusion maps,” *Appl. Comput. Harmon. Anal.*, **21**(1), 5 (2006).
- [15] R.R. Coifman and S. Lafon, “Geometric harmonics: A novel tool for multiscale out-of-sample extension of empirical functions,” *Appl. Comput. Harmon. Anal.*, **21**(1), 31 (2006).
- [16] R.R. Coifman, S. Lafon, A.B. Lee, M. Maggioni, B. Nadler, F.J. Warner and S.W. Zucker, “Geometric diffusions as a tool for harmonic analysis and structured definition of data. part i: Diffusion maps,” *Proc. Nat. Acad. Sci.* **102**, 7426 (2005).
- [17] R.R. Coifman, S. Lafon, A.B. Lee, M. Maggioni, B. Nadler, F.J. Warner and S.W. Zucker, “Geometric diffusions as a tool for harmonic analysis and structured definition of data. part ii: Multiscale methods,” *Proc. Nat. Acad. Sci.*, **102**, 7438 (2005).
- [18] R.R. Coifman and M. Maggioni, “Diffusion wavelets,” *Appl. Comput. Harmon. Anal.*, **21**(1), 53 (2006).
- [19] R.R. Coifman and M.V. Wickerhauser, “Entropy-based algorithms for best basis selection,” *IEEE Trans. Inf. Theory*, **38**(2), 713 (1992).
- [20] O. Christensen, *An Introduction to Frames and Riesz Bases* (Birkhäuser, Boston, 2003).
- [21] I. Daubechies, “The wavelet transform, time-frequency localization and signal analysis,” *IEEE Trans. Inform. Theory* **39**, 961 (1990).
- [22] I. Daubechies, *Ten Lectures on Wavelets* (SIAM, Philadelphia, PA, 1992).
- [23] M.N. Do and M. Vetterli, “Contourlets,” in *Beyond Wavelets*, eds. J. Stoeckler and G.V. Welland, pp. 1-27 (2001).

- [24] M.N. Do and M. Vetterli, “The Contourlet Transform: An Efficient Directional Multiresolution Image Representation,” *IEEE Trans. Image Process.* **14**(12), 2091 (2005).
- [25] M.N. Do and M. Vetterli, “The finite ridgelet transform for image representation,” *IEEE Trans. Image Process.* **12**(1), 16 (2003).
- [26] D.L. Donoho, “Wedgelets: nearly minimax estimation of edges,” *Ann. Stat.* **27**(3), (1999).
- [27] D.L. Donoho, H. Xiaoming, I. Jermyn, P. Jones, O. Levi and F. Natterer, “Beamlets and Multiscale Image Analysis,” in *Multiscale and Multiresolution Methods*, Springer, New York, 149 (2001).
- [28] D.L. Donoho and C. Grimes, “Hessian eigenmaps: new locally linear embedding techniques for high-dimensional data,” *Proc. Nat. Acad. Sci.* **100**, 5591 (2003).
- [29] R.J. Duffin and A.C. Schaeffer, “A class of nonharmonic Fourier series,” *Trans. Amer. Math. Soc.* **72**, 341 (1952).
- [30] A.M. Eskicoglu and P.S Fisher, “Image quality measures and their performance,” *IEEE Trans. Commun.* **32**(12), 2959 (1993).
- [31] G.B. Folland, *Harmonic Analysis in Phase Space* Princeton University Press, Princeton, NJ (1989).
- [32] G.B. Folland, *A Course in Abstract Harmonic Analysis* CRC Press, Boca Raton, FL (1995).
- [33] F. Friedrich, L. Demaret, H. Führ and K. Wicker, “Efficient Moment Computation Over Polygonal Domains With an Application to Rapid Wedgelet Approximations,” *SIAM J. Sci. Comput.* **29**(2), 842 (2007).
- [34] D. Gabor, “Theory of communication,” *J. IEE*, **93**(3), 429 (1946).
- [35] I. Gertner and R. Tolimieri, “Short Communication: Multiplicative Zak Transform,” *Journal of Visual Communication and Image Representation* **6**(1), 89 (1995).
- [36] D. Gillis, J. Bowles and M. Winter, “Using endmembers as a coordinate system in hyperspectral imagery,” *Proc. SPIE*, **4816**, 346 (2002).

- [37] K. Gröchenig, *Foundations of Time-Frequency Analysis* (Birkhäuser, Boston, 2000).
- [38] C. Heil and D. Walnut, “Continuous and discrete wavelet transforms,” *Soc. Ind. Appl. Math.* **31**, 628 (1989).
- [39] E. Hernández and G. Weiss, *A First Course on Wavelets* (Studies in Advanced Mathematics: CRC Press, Boca Raton, FL, 1996).
- [40] H.E. Jensen, T. Høholdt and J. Justesen, “Double Series Representation of Bounded Signals,” *IEEE Transactions on Information Theory.* **34**(4), 613 (1988).
- [41] Y. Katznelson, *An Introduction to Harmonic Analysis*. Cambridge University Press, Cambridge, MA, (2004).
- [42] J. Kovacevic and A. Chebira, “Life beyond bases: The advent of frames,” *IEEE Signal Processing Magazine*, (2007).
- [43] G. Kutyniok and W-Q. Lim, “Compactly supported shearlets are optimally sparse,” submitted (2010).
- [44] D. Labate, W-Q. Lim, G. Kutyniok and G. Weiss, “Sparse multidimensional representation using shearlets,” *Wavelets XI* (San Diego, CA, 2005), 254-262, *SPIE Proc.* **5914**, SPIE, Bellingham, WA, (2005).
- [45] M. Maggioni and R. Coifman, “Multiscale analysis of data sets with diffusion wavelets,” 7th SIAM International Conference on Data Mining, Minneapolis, MN (2007).
- [46] S. Mallat, *A Wavelet Tour of Signal Processing* (Academic Press, 1999), Second Ed.
- [47] T. Oommen, D. Misra, N.K.C Twarakavi, A. Prakash, B. Sahoo, S. Bandopadhyay, “An objective analysis of support vector machine based classification for remote sensing,” *Mathematical Geosciences* **40**, 409 (2008).
- [48] K. Pearson, “On lines and planes of closest fit to systems of points in space,” *Philosophical Magazine*, **2**(559), 7 (1901).
- [49] S. Roweis and L. Saul, “Nonlinear dimension reduction by locally linear embedding,” *Science*, **290**(5500), 2323 (2000).

- [50] J. Segman and W. Schempp, "Two Ways to Incorporate Scale in the Heisenberg Group with an Intertwining Operator," *Journal of Mathematical Imaging and Vision* **3**, 79 (1993).
- [51] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Trans. Image Process.* **14**(4), 600 (2004).
- [52] J. Tenenbaum, V.de Silva and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, **290**(5500), 2319 (2000).
- [53] P.C. Teo and D.J. Heeger, "Perceptual image distortion," *Proc. SPIE*, **2179**, 127 (1994).
- [54] A.B. Watson, "Image compression using the discrete cosine transform," *Mathematica J.*, **4**(1), 81 (1994).
- [55] A. Weil, "Sur certains groupes d'opérateurs unitaires," *Acta Math.*, **111**, 143 (1964).
- [56] D.P. Widemann, "Dimensionality Reduction for Hyperspectral Data," *Doctoral Thesis, University of Maryland: College Park*, (2008).
- [57] M. Winter, "N-FINDR: An algorithm for fast autonomous spectral endmember determination in hyperspectral data," *Proc. SPIE*, **3753**, 266 (1999).
- [58] C. Wu and C.-I Chang, "A new simplex growing algorithm for endmember extraction," *IEEE International Geoscience and Remote Sensing Symposium*, (2005).