RICE UNIVERSITY

The Geometry of Low-Dimensional Signal Models

by

Michael B. Wakin

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Doctor of Philosophy

Approved, Thesis Committee:

Richard G. Baraniuk, Chair,
Victor E. Cameron Professor,
Electrical and Computer Engineering

Michael T. Orchard, Professor,
Electrical and Computer Engineering

Steven J. Cox, Professor,
Computational and Applied Mathematics

Ronald A. DeVore, Robert L. Sumwalt
Distinguished Professor Emeritus,
Mathematics, University of South Carolina

David L. Donoho,
Anne T. and Robert M. Bass Professor,
Statistics, Stanford University

Houston, Texas
AUGUST 2006

Abstract

The Geometry of Low-Dimensional Signal Models
by
Michael B. Wakin

Models in signal processing often deal with some notion of structure or conciseness suggesting that a signal really has "few degrees of freedom" relative to its actual size. Examples include: bandlimited signals, images containing low-dimensional geometric features, or collections of signals observed from multiple viewpoints in a camera or sensor network. In many cases, such signals can be expressed as sparse linear combinations of elements from some dictionary — the sparsity of the representation directly reflects the conciseness of the model and permits efficient algorithms for signal processing. Sparsity also forms the core of the emerging theory of Compressed Sensing (CS), which states that a sparse signal can be recovered from a small number of random linear measurements.

In other cases, however, sparse representations may not suffice to truly capture the underlying structure of a signal. Instead, the conciseness of the signal model may in fact dictate that the signal class forms a low-dimensional manifold as a subset of the high-dimensional ambient signal space. To date, the importance and utility of manifolds for signal processing has been acknowledged largely through a research effort into "learning" manifold structure from a collection of data points. While these methods have proved effective for certain tasks (such as classification and recognition), they also tend to be quite generic and fail to consider the geometric nuances of specific signal classes.

The purpose of this thesis is to develop new methods and understanding for signal processing based on low-dimensional signal models, with a particular focus on the role of geometry. Our key contributions include (i) new models for low-dimensional signal structure, including local parametric models for piecewise smooth signals and joint sparsity models for signal collections; (ii) multiscale representations for piecewise smooth signals designed to accommodate efficient processing; (iii) insight and analysis into the geometry of low-dimensional signal models, including the non-differentiability of certain articulated image manifolds and the behavior of signal manifolds under random low-dimensional projections, and (iv) dimensionality reduction algorithms for image approximation and compression, distributed (multi-signal) CS, parameter estimation, manifold learning, and manifold-based CS.

Acknowledgements

The best part of graduate school has undoubtedly been getting to meet and work with so many amazing people. It has been a great privilege to take part in several exciting and intensive research projects, and I would like to thank my collaborators: Rich Baraniuk, Dror Baron, Rui Castro, Venkat Chandrasekaran, Hyeokho Choi, Albert Cohen, Mark Davenport, Ron DeVore, Dave Donoho, Marco Duarte, Felix Fernandes, Jason Laska, Matthew Moravec, Mike Orchard, Justin Romberg, Chris Rozell, Shri Sarvotham, and Joel Tropp. This thesis owes a great deal to their contributions, as I have also noted on the first page of several chapters.

I am also very grateful to my thesis committee for the many ways in which they contributed to this work: to Mike Orchard for some very challenging but motivating discussions; to Steve Cox for a terrific course introducing me to functional analysis back in my undergraduate days; to Ron DeVore for the time, energy, and humor he generously poured into his yearlong visit to Rice; to my "Dutch uncle" Dave Donoho for his patient but very helpful explanations and for strongly encouraging me to start early on my thesis; and most of all to my advisor Rich Baraniuk for somehow providing me with the right mix of pressure and encouragement. Rich's boundless energy has been a true inspiration, and I thank him for the countless hours he enthusiastically devoted to helping me develop as a speaker, writer, and researcher.

Much of the inspiration for this work came during a Fall 2004 visit to the UCLA Institute for Pure and Applied Mathematics (IPAM) for a program on Multiscale Geometry and Analysis in High Dimensions. I am very grateful to the organizers of that program, particularly to Emmanuel Candès and Dave Donoho for several enlightening conversations and to Peter Jones for having a good sense of humor.

At Rice, I could not have asked for a more fun or interesting group of people to work with: Becca, Chris, Clay, Courtney, Dror, Hyeokho, Ilan, Justin, Jyoti, Kadim, Laska, Lavu, Lexa, Liz, Marco, Mark, Matt G., Matthew M., Mike O., Mike R., Mona, Neelsh, Prashant, Ray, Rich, Rob, Rui, Rutger, Ryan, Shri, Venkat, Vero, Vinay, William C., William M., and many more. Thank you all for making Duncan Hall a place I truly enjoyed coming to work. I will fondly remember our favorite pastimes: long lunchtime conversations, Friday afternoons at Valhalla, waiting for Rich to show up at meetings, etc.

Finally, thank you to all of my other friends and family who helped me make it this far, providing *critical* encouragement, support, and distractions when I needed them most: Mom, Dad, Larry, Jackie, Jason, Katy, Denver, Nasos, Dave, Alex, Clay, Sean, Megan, many good friends from The MOB, and everyone else who helped me along the way. Thank you again.

Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Structure and Models in Signal Processing

Signal processing represents one of the primary interfaces of mathematics and science. The abilities to efficiently and accurately measure, process, understand, quantify, compress, and communicate data and information rely both on accurate models for the situation at hand and on novel techniques inspired by the underlying mathematics. The tools and algorithms that have emerged from such insights have had far-reaching impacts, helping to revolutionize fields from communications [1] and entertainment [2] to biology [3] and medicine [4].

In characterizing a given problem, one is often able to specify a *model* for the signals to be processed. This model may distinguish (either statistically or deterministically) classes of interesting signals from uninteresting ones, typical signals from anomalies, information from noise, etc. The model can also have a fundamental impact on the design and performance of signal processing tools and algorithms. As a simple example, one common assumption is that the signals to be processed are bandlimited, in which case each signal can be written as a different linear combination of low-frequency sinusoids. Based on this assumption, then, the Shannon/Nyquist sampling theorem [5] specifies a minimal sampling rate for preserving the signal information; this powerful result forms the core of modern Digital Signal Processing (DSP).

Like the assumption of bandlimitedness, models in signal processing often deal with some notion of structure, constraint, or conciseness. Roughly speaking, one often believes that a signal has "few degrees of freedom" relative to the size of the signal. This can be caused, for example, by a physical system having few parameters, a limit to the information encoded in a signal, or an oversampling relative to the information content of a signal. This notion of conciseness is a very powerful assumption, and it suggests the potential for dramatic gains via algorithms that capture and exploit the true underlying structure of the signal.

To give a more concrete example, one popular generalization[1] of the bandlimited model in signal processing is *sparsity*, in which each signal is well-approximated as a small linear combination of elements from some basis or dictionary, but the choice of elements may vary from signal to signal [6, 7]. In the frequency domain, a sparse model would suggest that each signal consists of just a few sinusoids, whose ampli-

[1] Or refinement, depending on one's perspective.

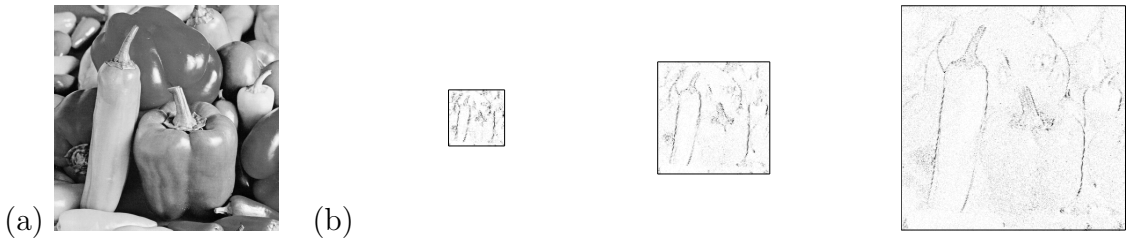

Figure 1.1: *(a)* Peppers *test image. (b) Wavelet coefficient magnitudes in coarse-to-fine scales of analysis (vertical subbands shown). At each scale, the relatively few significant wavelet coefficients tend to cluster around the edges of the objects in the image. This makes possible a variety of effective models for capturing intra- and inter-scale dependencies among the wavelet coefficients but also implies that the locations of significant coefficients will change from image to image.*

tudes, phases, *and* frequencies are variable. (A recording of a musical performance, for example, might be sparse in a dictionary containing sinusoids of limited duration.) Sparsity has also been exploited in fields such as image processing, where the multiscale wavelet transform [5] permits concise, efficiently computable descriptions of images (see Figure 1.1). In a nutshell, wavelets provide a sparse representation for natural images because large smooth image regions require very few wavelets to describe; only the abrupt edges separating smooth regions require large (significant) wavelet coefficients, and those regions occupy a relatively small total area of the image. The key phenomenon to note, however, is that the locations of these significant coefficients may change from image to image.

Sparse representations have proven themselves as a powerful tool for capturing concise signal structure and have led to fast, effective algorithms for solving key problems in signal processing. Wavelets form the core of many state-of-the art methods for data compression and noise removal [8–12] — the multiscale structure of the wavelet transform suggests a top-down tree structure that is particularly effective for computation and modeling. Curvelets have also recently emerged as a multiscale dictionary better suited to edge-like phenomena in two-dimensional (2-D) and three-dimensional (3-D) signals [13–15] and have proven effective, for example, in solving inverse problems for seismic data processing. Inspired by successes such as these, research continues in developing novel sparse dictionaries that are adapted to broader families of signal classes and, again, that are amenable to fast algorithms (often through a multiscale structure).

As we have stated, the notion that many signals have sparse structure is widespread in signal processing and eminently useful. However, sparsity itself can sometimes be a rather restrictive assumption; there are many other interesting and important notions of concise signal structure that may not give rise to representations that are sparse in the conventional sense. Such notions often arise in cases where (i) a small collection of parameters can be identified that carry the relevant information about

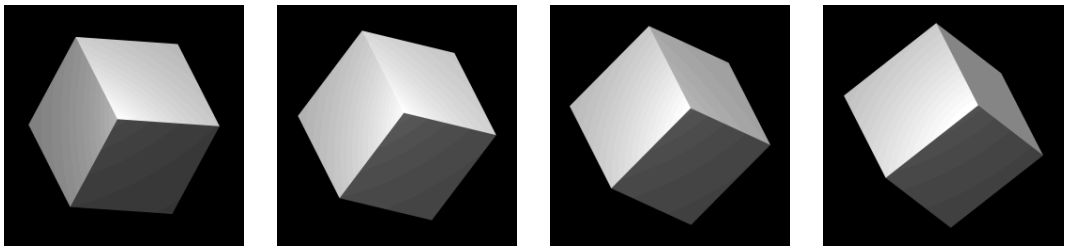

Figure 1.2: *Four $256 \times 256 = 65536$-pixel images of an identical cube, differing only in the position of the cube (10 degrees of rotation between each frame). As the cube rotates, the images change (edges move, shadings differ, etc.), and the resulting images trace out a path on a low-dimensional manifold within the high-dimensional ambient signal space $\mathbb{R}^{65536}$. As we discuss in Chapter 4, this manifold is in fact non-differentiable.*

a signal and (ii) the signal changes as a function of these parameters. Some simple explicit examples include: the time delay of a 1-D signal (parametrized by 1 variable for translation), the configuration of a straight edge in a local image segment (2 parameters: slope and offset), the position of a camera photographing a scene ($\sim$6 parameters), the relative placement of objects in a scene, the duration and chirp rate of a radar pulse, or other parameters governing the output of some articulated physical system [16–19].

In some cases, these parameters may suffice to completely describe the signal; in other cases they may merely serve to distinguish it from other, related signals in the class. (See, for example, the images in Figure 1.2.) The key is that, for a particular problem, the relevant *information* about a signal can often be summarized in a small number of variables (the "degrees of freedom"). While the signal may also happen to have a sparse representation in some dictionary (such as the wavelet transform of an image of a straight edge), this sparsity will rarely reflect the true "information level" of the signal. This motivates a search for novel signal processing representations and algorithms that better exploit the conciseness of such signal models, including cases where the parametric model is only an approximation or where the parametric model is actually unknown.

1.2 Geometry and Low-Dimensional Signal Models

As we have discussed, models play a critical role in signal processing. In a very broad sense, a model can be thought of as an answer to the question: "*What* are the signals of interest?" Based on our understanding of this model, our goal is to develop efficient tools, representations, algorithms, and so on.

As an inspiration for developing these solutions, we believe that significant mathematical insight can often be gained by asking a related geometric question: "*Where* are the signals of interest?" That is, where do signals in the model class reside as a

subset of the ambient signal space (e.g., $\mathbb{R}^N$ for real-valued discrete length-N signals)? Indeed, as we will see, many of the concise signal models discussed in Section 1.1 actually translate to low-dimensional structures within the high-dimensional signal space; again, the low dimension of these structures suggests the potential for fast, powerful algorithms. By studying and understanding the geometry of these low-dimensional structures, we hope to identify new challenges in signal processing and to discover new solutions.

Returning to some specific examples, bandlimited signals live on a low-dimensional *linear subspace* of the ambient signal space (see Section 2.4.1); indeed, the very word "linear" immediately evokes a geometric understanding. It follows immediately, then, that tasks such as optimally removing noise from a signal (in a least-squares sense) would simply involve orthogonal projection onto this subspace.

Sparse signals, on the other hand, live near a *nonlinear* set that is a union of such low-dimensional subspaces. Again, this geometry plays a critical role in the signal processing; Chapter 2 discusses in depth the implications for tasks such as approximation and compression. One of the most surprising implications of the nonlinear, low-dimensional geometry of sparse signal sets comes from the recent theory of *Compressed Sensing* (CS) [20, 21]. The CS theory states that a length-N signal that is K-sparse (it can be written as a sum of K basis elements) can be reconstructed from only cK nonadaptive linear projections onto a second basis that is incoherent with the first, where typically $c \approx 3$ or 4. (A random basis provides such incoherence with very high probability.) This has many promising applications in signal acquisition, compression, medical imaging, and sensor networks [22–33]. A key point is that the CS theory relies heavily on geometric notions such as the n-widths of ℓ^p balls and the properties of randomly projected polytopes [20, 21, 23, 34–40] (see Section 2.8.5).

In more general cases where one has a concise model for signal structure, the resulting signal class often manifests itself as a low-dimensional, nonlinear *manifold* embedded in the high-dimensional signal space.[2] This is the case, in particular, for parametric signal models; as discussed in Section 2.4.3, the dimension of the manifold will match the dimension of the underlying parameter (the number of degrees of freedom). More generally, however, manifolds have also been discovered as useful approximations for signal classes not obeying an explicit parametric model. Examples include the output of dynamical systems having low-dimensional attractors [41, 42] or collections of images such as faces or handwritten digits [43].

Naturally, the geometry of signal manifolds will also have a critical impact on the performance of signal processing methods. To date, the importance and utility of manifolds for signal processing has been acknowledged largely through a research effort into "learning" manifold structure from a collection of data points, typically by constructing "dimensionality reducing" mappings to lower-dimensional space that

[2] A manifold can be thought of as a low-dimensional, nonlinear "surface" within the high-dimensional signal space; Section 2.2 gives a more precise definition. Note that the linear subspace and "union of subspaces" models are essentially special cases of such manifold structure.

reveal the locally Euclidean nature of the manifold or by building functions on the data that reveal its metric structure [41–54] (see also Section 2.7.1). While these methods have proven effective for certain tasks (such as classification and recognition), they also tend to be quite generic. Due to the wide variety of situations in which signal manifolds may arise, however, different signal classes may have different geometric nuances that deserve special attention. Relatively few studies have considered the geometry of specific classes of signals; important exceptions include the work of Lu [55], who empirically studied properties such as the dimension and curvature of image manifolds, Donoho and Grimes [16], who examined the metric structure of articulated image manifolds, and Mumford et al. [56], who used manifolds to model sets of shapes. In general, we feel that the incorporation of the manifold viewpoint into signal processing is only beginning, and more careful studies will both advance our understanding and inspire new solutions.

1.3 Overview and Contributions

The purpose of this thesis is to develop new methods and understanding for signal processing based on low-dimensional signal models, with a particular focus on the role of geometry. To guide our study, we consider two primary application areas:

1. *Image processing*, a research area of broad importance in which concise signal models abound (thanks to the articulations of objects in a scene, the regularity of smooth regions, and the 1-D geometry of edges), and

2. *Compressed Sensing*, a nascent but markedly geometric theory with great promise for applications in signal acquisition, compression, medical imaging, and sensor networks.

Our key contributions include new:

- *concise signal models* that generalize the conventional notion of sparsity;

- *multiscale representations* for sparse approximation and compression;

- *insight and analysis* into the geometry of low-dimensional signal models based on concepts in differential geometry and differential topology; and

- *algorithms* for parameter estimation and dimensionality reduction inspired by the underlying manifold structure.

We outline these contributions chapter-by-chapter.

We begin in **Chapter 2** with a background discussion of low-dimensional signal models. After a short list of mathematical preliminaries and notation, including a brief introduction to manifolds, we discuss the role of signal dictionaries and representations, the geometry of linear, sparse, and manifold-based signal models, and

the implications in problems such as approximation and compression. We also discuss more advanced techniques in dimensionality reduction, manifold learning, and Compressed Sensing.

In **Chapter 3** we consider the task of approximating and compressing two model classes of functions for which traditional harmonic dictionaries fail to provide sparse representations. However, the model itself dictates a low-dimensional structure to the signals, which we capture using a novel parametric multiscale dictionary. The functions we consider are both highly relevant in signal processing and highly structured. In particular, we consider piecewise constant signals in P dimensions where a smooth $(P-1)$-dimensional discontinuity separates the two constant regions, and we also consider the extension of this class to piecewise smooth signals, where a smooth $(P-1)$-dimensional discontinuity separates two smooth regions. These signal classes provide basic models, for example, for images containing edges, video sequences of moving objects, or seismic data containing geological horizons. Despite the underlying (indeed, low-dimensional) structure in each of these classes, classical harmonic dictionaries fail to provide sparse representations for such signals. The problem comes from the $(P-1)$-dimensional discontinuity, whose smooth geometric structure is not captured by local isotropic representations such as wavelets.

As a remedy, we propose a multiscale dictionary consisting of local parametric atoms called *surflets*, each a piecewise constant function with a (tunable) polynomial discontinuity separating the two constant regions. Our surflet dictionary falls outside the traditional realm of bases and frames (where approximations are assembled as linear combinations of atoms from the dictionary). Rather our scheme is perhaps better viewed as a "geometric tiling," where precisely one atom from the dictionary is used to describe the signal at each part of the domain (these atoms "tile" together to cover the domain). We discuss multiscale (top-down, tree-based) schemes for assembling and encoding surflet representations, and we prove that such schemes attain optimal asymptotic approximation and compression performance on our piecewise constant function classes. We also discuss techniques for interfacing surflets with wavelets for representing more general classes of functions. The resulting dictionary, which we term *surfprints*, attains near-optimal asymptotic approximation and compression performance on our piecewise smooth function classes.

In **Chapter 4** we study the geometry of signal manifolds in more detail, particularly in the case of parametrized image manifolds (such as the 2-D surflet manifold). We call these *Image Appearance Manifolds* (IAMs) and let θ denote the parameter controlling the image formation. Our work builds upon a surprising realization [16]: IAMs of continuous images having sharp edges that move as a function of θ are *nowhere differentiable*. This presents an immediate challenge for signal processing algorithms that might assume differentiability or smoothness of such manifolds. Using Newton's method, for example, to estimate the parameter θ for an unlabeled image, would require successive projections onto tangent spaces of the manifold. Because the manifold is not differentiable, however, these tangents do not exist.

Although these IAMs lack differentiability, we identify a multiscale collection of tangent spaces to the manifold, each one associated with both a *location* on the manifold and *scale* of analysis; this multiscale structure can be accessed simply by regularizing the images. Based on this multiscale perspective, we propose a Multiscale Newton algorithm to solve the parameter estimation problem. We also reveal a second, more localized kind of IAM non-differentiability caused by sudden *occlusions* of edges at special values of θ. This type of phenomenon has its own implications in the signal processing and requires a special vigilance; it is not alleviated by merely regularizing the images.

In **Chapter 5** we consider another novel modeling perspective, as we turn our attention toward a suite of signal models designed for simultaneous modeling of *multiple* signals that have a shared concise structure. Our primary motivation for introducing these models is to extend the CS theory and methods to a multi-signal setting — while CS appears promising for applications such as sensor networks, at present it is tailored only for the sensing of a single sparse signal.

We introduce a new theory for *Distributed Compressed Sensing* (DCS) that enables new distributed coding algorithms that exploit both intra- and inter-signal correlation structures. In a typical DCS scenario, a number of sensors measure signals that are each individually sparse in some basis and also correlated from sensor to sensor. Each sensor *independently* encodes its signal by projecting it onto another, incoherent basis (such as a random one) and then transmits just a few of the resulting coefficients to a single collection point. Under the right conditions, a decoder at the collection point can reconstruct each of the signals precisely.

The DCS theory rests on a concept that we term the *joint sparsity* of a signal ensemble. We study in detail three simple models for jointly sparse signals, propose tractable algorithms for joint recovery of signal ensembles from incoherent projections, and characterize theoretically and empirically the number of measurements per sensor required for accurate reconstruction. While the sensors operate entirely without collaboration, our simulations reveal that in practice the savings in the total number of required measurements can be substantial over separate CS decoding, especially when a majority of the sparsity is shared among the signals.

In **Chapter 6**, inspired again by a geometric perspective, we develop new theory and methods for problems involving random projections for dimensionality reduction. In particular, we consider embedding results previously applicable only to finite point clouds (the Johnson-Lindenstrauss lemma; see Section 2.7.2) or to sparse signal models (Compressed Sensing) and generalize these results to include manifold-based signal models. As our primary theoretical contribution (Theorem 6.2), we consider the effect of a random projection operator on a smooth K-dimensional submanifold of $\mathbb{R}^N$, establishing a sufficient number M of random projections to ensure a stable embedding. We explore a number of possible applications of this result, particularly in CS, which we generalize beyond the recovery of sparse signals to include the recovery of manifold-modeled signals from small number of random projections. We also

discuss other possible applications in manifold learning and dimensionality reduction.

We conclude in **Chapter 7** with a final discussion and directions for future research.

This thesis is a reflection of a series of intensive and inspiring collaborations. Where appropriate, the first page of each chapter includes a list of primary collaborators, who share the credit for this work.

Chapter 2
Background on Signal Modeling and Processing

2.1 General Mathematical Preliminaries

2.1.1 Signal notation

We will treat signals as real- or complex-valued functions having domains that are either discrete (and finite) or continuous (and either compact or infinite). Each of these assumptions will be made clear in the particular chapter or section. As a general rule, however, we will use x to denote a discrete signal in $\mathbb{R}^N$ and f to denote a function over a continuous domain $\mathcal{D}$. We also commonly refer to these as discrete- or continuous-*time* signals, though the domain need not actually be temporal in nature. Additional chapter-specific conventions will be specified as necessary.

2.1.2 L_p and ℓ_p norms

As measures for signal energy, fidelity, or sparsity, we will often employ the L_p and ℓ_p norms. For continuous-time functions, the L_p norm is defined as

$$\|f\|_{L_p(\mathcal{D})} = \left(\int_{\mathcal{D}} |f|^p\right)^{1/p}, \quad p \in (0,\infty),$$

and for discrete-time functions, the ℓ_p norm is defined as

$$\|x\|_{\ell_p} = \begin{cases} (\sum_{i=1}^N |x(i)|^p)^{1/p}, & p \in (0,\infty), \\ \max_{i=1,\ldots,N} |x(i)|, & p = \infty, \\ \sum_{i=1}^N \mathbf{1}_{x(i)\neq 0}, & p = 0, \end{cases}$$

where $\mathbf{1}$ denotes the indicator function. (While we often refer to these measures as "norms," they actually do not meet the technical criteria for norms when $p < 1$.)

The *mean-square error* (MSE) between two discrete-time signals $x_1, x_2 \in \mathbb{R}^N$ is given by $\frac{1}{N}\|x_1 - x_2\|_2^2$. The *peak signal-to-noise ratio* (PSNR), another common measure of distortion between two signals, derives directly from the MSE; assuming a maximum possible signal intensity of I, $PSNR := 10\log_{10}\frac{I^2}{MSE}$.

2.1.3 Linear algebra

Let A be a real-valued $M \times N$ matrix. We denote the *nullspace* of A as $\mathcal{N}(A)$ (note that $\mathcal{N}(A)$ is a linear subspace of $\mathbb{R}^N$), and we denote the *transpose* of A as A^T.

We call A an *orthoprojector* from $\mathbb{R}^N$ to $\mathbb{R}^M$ if it has orthonormal rows. From such a matrix we call $A^T A$ the corresponding *orthogonal projection operator* onto the M-dimensional subspace of $\mathbb{R}^N$ spanned by the rows of A.

2.1.4 Lipschitz smoothness

We say a continuous-time function of D variables has smoothness of order $H > 0$, where $H = r+\nu$, r is an integer, and $\nu \in (0, 1]$, if the following criteria are met [57,58]:

- All iterated partial derivatives with respect to the D directions up to order r exist and are continuous.
- All such partial derivatives of order r satisfy a Lipschitz condition of order ν (also known as a Hölder condition).[1]

We will sometimes consider the space of smooth functions whose partial derivatives up to order r are bounded by some constant Ω. We denote the space of such bounded functions with bounded partial derivatives by $\mathcal{C}^H$, where this notation carries an implicit dependence on Ω. Observe that $r = \lceil H - 1 \rceil$, where $\lceil \cdot \rceil$ denotes rounding up. Also, when H is an integer $\mathcal{C}^H$ includes as a subset the traditional space "C^H" (the class of functions that have $H = r + 1$ continuous partial derivatives).

2.1.5 Scale

We will frequently refer to a particular *scale* of analysis for a signal. Suppose our functions f are defined over the continuous domain $\mathcal{D} = [0, 1]^D$. A *dyadic hypercube* $X_j \subseteq [0, 1]^D$ at scale $j \in \mathbb{N}$ is a domain that satisfies

$$X_j = [\beta_1 2^{-j}, (\beta_1 + 1)2^{-j}] \times \cdots \times [\beta_D 2^{-j}, (\beta_D + 1)2^{-j}]$$

with $\beta_1, \beta_2, \ldots, \beta_D \in \{0, 1, \ldots, 2^j - 1\}$. We call X_j a *dyadic interval* when $D = 1$ or a *dyadic square* when $D = 2$ (see Figure 2.1). Note that X_j has sidelength 2^{-j}.

For discrete-time functions the notion of scale is similar. We can imagine, for example, a "voxelization" of the domain $[0, 1]^D$ ("pixelization" when $D = 2$), where each voxel has sidelength 2^{-B}, $B \in \mathbb{N}$, and it takes 2^{BD} voxels to fill $[0, 1]^D$. The relevant scales of analysis for such a signal would simply be $j = 0, 1, \ldots, B$, and each dyadic hypercube X_j would refer to a collection of voxels.

[1] A function $d \in \text{Lip}(\nu)$ if $|d(t_1 + t_2) - d(t_1)| \leq C\|t_2\|^\nu$ for all D-dimensional vectors t_1, t_2.

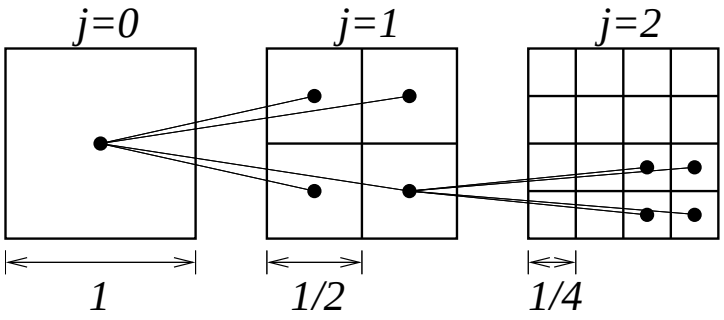

Figure 2.1: *Dyadic partitioning of the unit square at scales $j = 0, 1, 2$. The partitioning induces a coarse-to-fine parent/child relationship that can be modeled using a tree structure.*

2.2 Manifolds

We present here a minimal, introductory set of definitions and terminology from differential geometry and topology, referring the reader to the introductory and classical texts [59–62] for more depth and technical precision.

2.2.1 General terminology

A *K-dimensional manifold* $\mathcal{M}$ is a topological space[2] that is locally homeomorphic[3] to $\mathbb{R}^K$ [61]. This means that there exists an open cover of $\mathcal{M}$ with each such open set mapping homeomorphically to an open ball in $\mathbb{R}^K$. Each such open set, together with its mapping to $\mathbb{R}^K$ is called a *chart*; the set of all charts of a manifold is called an *atlas*.

The general definition of a manifold makes no reference to an ambient space in which the manifold lives. However, as we will often be making use of manifolds as models for sets of signals, it follows that such "signal manifolds" are actually subsets of some larger space (for example, of $L_2(\mathbb{R})$ or $\mathbb{R}^N$). In general, we may think of a K-dimensional submanifold embedded in $\mathbb{R}^N$ as a nonlinear, K-dimensional "surface" within $\mathbb{R}^N$.

2.2.2 Examples of manifolds

One of the simplest examples of a manifold is simply the circle in $\mathbb{R}^2$. A small, open-ended segment cut from the circle could be stretched out and associated with an open interval of the real line (see Figure 2.2). Hence, the circle is a 1-D manifold.

[2] A *topological space* is simply a set X, together with a collection T of subsets of X called open sets, such that: (i) the empty set belongs to T, (ii) X belongs to T, (iii) arbitrary unions of elements of T belong to T, and (iv) finite intersections of elements of T belong to T.

[3] A *homeomorphism* is a function between two topological spaces that is one-to-one, onto, continuous, and has a continuous inverse.

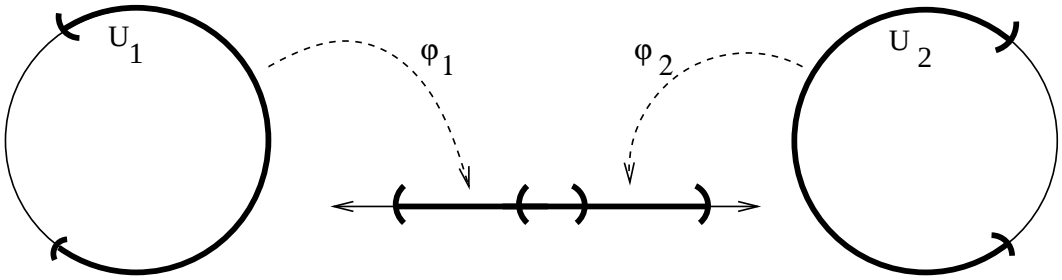

Figure 2.2: *A circle is a manifold because there exists an open cover consisting of the sets U_1, U_2, which are mapped homeomorphically onto open intervals in the real line via the functions φ_1, φ_2. (It is not necessary that the intervals intersect in $\mathbb{R}$.)*

(We note that at least two charts are required to form an atlas for the circle, as the entire circle itself cannot be mapped homeomorphically to an open interval in $\mathbb{R}^1$.)

We refer the reader to [63] for an excellent overview of several manifolds with relevance to signal processing, including the rotation group $SO(3)$, which can be used for representing orientations of objects in 3-D space, and the Grassman manifold $G(K, N)$, which represents all K-dimensional subspaces of $\mathbb{R}^N$. (Without working through the technicalities of the definition of a manifold, it is easy to see that both types of data have a natural notion of neighborhood.)

2.2.3 Tangent spaces

A manifold is *differentiable* if, for any two charts whose open sets on $\mathcal{M}$ overlap, the composition of the corresponding homeomorphisms (from $\mathbb{R}^K$ in one chart to $\mathcal{M}$ and back to $\mathbb{R}^K$ in the other) is differentiable. (In our simple example, the circle is a differentiable manifold.)

To each point x in a differentiable manifold, we may associate a K-dimensional *tangent space* Tan_x. For signal manifolds embedded in L_2 or $\mathbb{R}^N$, it suffices to think of Tan_x as the set of all directional derivatives of smooth paths on $\mathcal{M}$ through x. (Note that Tan_x is a linear subspace and has its origin at 0, rather than at x.)

2.2.4 Distances

One is often interested in measuring distance along a manifold. For abstract differentiable manifolds, this can be accomplished by defining a Riemannian metric on the tangent spaces. A Riemannian metric is a collection of inner products $\langle , \rangle_x$ defined at each point $x \in \mathcal{M}$. The inner product gives a measure for the "length" of a tangent, and one can then compute the length of a path on $\mathcal{M}$ by integrating its tangent lengths along the path.

For differentiable manifolds embedded in $\mathbb{R}^N$, the natural metric is the Euclidean metric inherited from the ambient space. The length of a path $\gamma : [0, 1] \mapsto \mathcal{M}$ can

then be computed simply using the limit

$$\text{length}(\gamma) = \lim_{j \to \infty} \sum_{i=1}^{j} \|\gamma(i/j) - \gamma((i-1)/j)\|_2 .$$

The *geodesic distance* $d_{\mathcal{M}}(x, y)$ between two points $x, y \in \mathcal{M}$ is then given by the length of the shortest path γ on $\mathcal{M}$ joining x and y.

2.2.5 Curvature

Several notions of *curvature* also exist for manifolds. The curvature of a unit-speed path in $\mathbb{R}^N$ is simply given by its second derivative. More generally, for manifolds embedded in $\mathbb{R}^N$, characterizations of curvature generally relate to the second derivatives of paths along $\mathcal{M}$ (in particular, the components of the second derivatives that are normal to $\mathcal{M}$). Section 2.2.6 characterizes the notions of curvature and "twisting" of a manifold that will be most relevant to us.

2.2.6 Condition number

To give ourselves a firm footing for later analysis, we find it helpful assume a certain regularity to the manifold beyond mere differentiability. For this purpose, we adopt the condition number defined recently by Niyogi et al. [51].

Definition 2.1 [51] *Let $\mathcal{M}$ be a compact submanifold of $\mathbb{R}^N$. The* condition number *of $\mathcal{M}$ is defined as $1/\tau$, where τ is the largest number having the following property: The open normal bundle about $\mathcal{M}$ of radius r is imbedded in $\mathbb{R}^N$ for all $r < \tau$.*

The open normal bundle of radius r at a point $x \in \mathcal{M}$ is simply the collection of all vectors of length $< r$ anchored at x and with direction orthogonal to Tan_x.

In addition to controlling local properties (such as curvature) of the manifold, the condition number has a global effect as well, ensuring that the manifold is self-avoiding. These notions are made precise in several lemmata, which we will find helpful for analysis and which we repeat below for completeness.

Lemma 2.1 [51] *If $\mathcal{M}$ is a submanifold of $\mathbb{R}^N$ with condition number $1/\tau$, then the norm of the second fundamental form is bounded by $1/\tau$ in all directions.*

This implies that unit-speed geodesic paths on $\mathcal{M}$ have curvature bounded by $1/\tau$. The second lemma concerns the twisting of tangent spaces.

Lemma 2.2 [51] *Let $\mathcal{M}$ be a submanifold of $\mathbb{R}^N$ with condition number $1/\tau$. Let $p, q \in \mathcal{M}$ be two points with geodesic distance given by $d_{\mathcal{M}}(p, q)$. Let θ be the angle between the tangent spaces Tan_p and Tan_q defined by $\cos(\theta) = \min_{u \in \text{Tan}_p} \max_{v \in \text{Tan}_q} |\langle u, v \rangle|$. Then $\cos(\theta) > 1 - \frac{1}{\tau} d_{\mathcal{M}}(p, q)$.*

The third lemma concerns self-avoidance of $\mathcal{M}$.

Lemma 2.3 [51] *Let $\mathcal{M}$ be a submanifold of $\mathbb{R}^N$ with condition number $1/\tau$. Let $p, q \in \mathcal{M}$ be two points such that $\|p - q\|_2 = d$. Then for all $d \leq \tau/2$, the geodesic distance $d_{\mathcal{M}}(p, q)$ is bounded by $d_{\mathcal{M}}(p, q) \leq \tau - \tau\sqrt{1 - 2d/\tau}$.*

From Lemma 2.3 we have an immediate corollary.

Corollary 2.1 *Let $\mathcal{M}$ be a submanifold of $\mathbb{R}^N$ with condition number $1/\tau$. Let $p, q \in \mathcal{M}$ be two points such that $\|p - q\|_2 = d$. If $d \leq \tau/2$, then $d \geq d_{\mathcal{M}}(p, q) - \frac{(d_{\mathcal{M}}(p,q))^2}{2\tau}$.*

2.2.7 Covering regularity

For future reference, we also introduce a notion of "geodesic covering regularity" for a manifold.

Definition 2.2 *Let $\mathcal{M}$ be a compact submanifold of $\mathbb{R}^N$. Given $T > 0$, the* geodesic covering number *$G(T)$ of $\mathcal{M}$ is defined as the smallest number such that there exists a set A of points, $\#A = G(T)$, so that for all $x \in \mathcal{M}$,*

$$\min_{a \in A} d_{\mathcal{M}}(x, a) \leq T.$$

Definition 2.3 *Let $\mathcal{M}$ be a compact K-dimensional submanifold of $\mathbb{R}^N$ having volume V. We say that $\mathcal{M}$ has* geodesic covering regularity *R if*

$$G(T) \leq \frac{RVK^{K/2}}{T^K} \tag{2.1}$$

for all $T > 0$.

The *volume* referred to above is K-dimensional volume (also known as length when $K = 1$ or surface area when $K = 2$).

The geodesic covering regularity of a manifold is closely related to its ambient distance-based covering number $C(T)$ [51]. In fact, for a manifold with condition number $1/\tau$, we can make this connection explicit. Lemma 2.3 implies that for small d, $d_{\mathcal{M}}(p, q) \leq \tau - \tau\sqrt{1 - 2d/\tau} \leq \tau(1 - (1 - 2d/\tau)) = 2d$. This implies that $G(T) \leq C(T/4)$ for small T. Pages 13–14 of [51] also establish that for small T, the ambient covering number can be bounded by a packing number $P(T)$ of the manifold, from

which we conclude that

$$\begin{aligned} G(T) &\leq C(T/4) \leq P(T/8) \\ &\leq \frac{V}{\cos(\arcsin(\frac{T}{16\tau}))^K \mathrm{vol}(B^K_{T/8})} \\ &\leq \frac{V \cdot \Gamma(K/2+1)}{(1-(\frac{T}{16\tau})^2)^{K/2}\pi^{K/2}(T/8)^K} \\ &\leq \mathrm{Const} \cdot \frac{VK^{K/2}}{T^K}. \end{aligned}$$

Although we point out this connection between the geodesic covering regularity and the condition number, for future reference and flexibility we prefer to specify these as distinct properties in our results in Chapter 6.

2.3 Signal Dictionaries and Representations

For a wide variety of signal processing applications (including analysis, compression, noise removal, and so on) it is useful to consider the representation of a signal in terms of some dictionary [5]. In general, a *dictionary* Ψ is simply a collection of elements drawn from the signal space whose linear combinations can be used to represent or approximate signals.

Considering, for example, signals in $\mathbb{R}^N$, we may collect and represent the elements of the dictionary Ψ as an $N \times Z$ matrix, which we also denote as Ψ. From this dictionary, a signal $x \in \mathbb{R}^N$ can be constructed as a linear combination of the elements (columns) of Ψ. We write

$$x = \Psi\alpha$$

for some $\alpha \in \mathbb{R}^Z$. (For much of our notation in this section, we concentrate on signals in $\mathbb{R}^N$, though the basic concepts translate to other vector spaces.)

Dictionaries appear in a variety of settings. The most common may be the basis, in which case Ψ has exactly N linearly independent columns, and each signal x has a unique set of expansion coefficients $\alpha = \Psi^{-1}x$. The orthonormal basis (where the columns are normalized and orthogonal) is also of particular interest, as the unique set of expansion coefficients $\alpha = \Psi^{-1}x = \Psi^T x$ can be obtained as the inner products of x against the columns of Ψ. That is, $\alpha(i) = \langle x, \psi_i \rangle, i = 1, 2, \ldots, N$, which gives us the expansion

$$x = \sum_{i=1}^{N} \langle x, \psi_i \rangle \psi_i.$$

We also have that $\|x\|_2^2 = \sum_{i=1}^N \langle x, \psi_i \rangle^2$.

Frames are another special type of dictionary [64]. A dictionary Ψ is a frame if

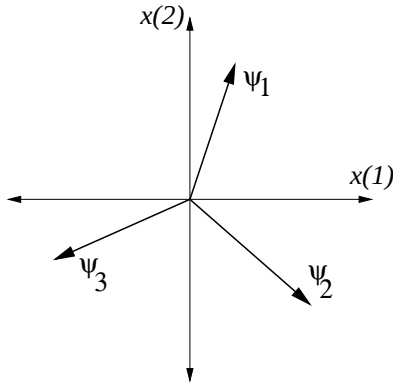

Figure 2.3: *A simple, redundant frame Ψ containing three vectors that span $\mathbb{R}^2$.*

there exist numbers A and B, $0 < A \leq B < \infty$ such that, for any signal x

$$A \|x\|_2^2 \leq \sum_z \langle x, \psi_z \rangle^2 \leq B \|x\|_2^2 .$$

The elements of a frame may be linearly dependent in general (see Figure 2.3), and so there may exist many ways to express a particular signal among the dictionary elements. However, frames do have a useful analysis/synthesis duality: for any frame Ψ there exists a dual frame $\widetilde{\Psi}$ such that

$$x = \sum_z \langle x, \psi_z \rangle \, \widetilde{\psi}_z = \sum_z \left\langle x, \widetilde{\psi}_z \right\rangle \psi_z .$$

A frame is called *tight* if the frame bounds A and B are equal. Tight frames have the special properties of (i) being their own dual frames (after a rescaling by $1/A$) and (ii) preserving norms, i.e., $\sum_{i=1}^{N} \langle x, \psi_i \rangle^2 = A \|x\|_2^2$. The remainder of this section discusses several important dictionaries.

2.3.1 The canonical basis

The standard basis for representing a signal is the canonical (or "spike") basis. In $\mathbb{R}^N$, this corresponds to a dictionary $\Psi = I_N$ (the $N \times N$ identity matrix). When expressed in the canonical basis, signals are often said to be in the "time domain."

2.3.2 Fourier dictionaries

The frequency domain provides one alternative representation to the time domain. The Fourier series and discrete Fourier transform are obtained by letting Ψ contain complex exponentials and allowing the expansion coefficients α to be complex as well. (Such a dictionary can be used to represent real or complex signals.) A related "harmonic" transform to express signals in $\mathbb{R}^N$ is the discrete cosine transform (DCT), in

which Ψ contains real-valued, approximately sinusoidal functions and the coefficients α are real-valued as well.

2.3.3 Wavelets

Closely related to the Fourier transform, wavelets provide a framework for localized harmonic analysis of a signal [5]. Elements of the discrete wavelet dictionary are local, oscillatory functions concentrated approximately on dyadic supports and appear at a discrete collection of scales, locations, and (if the signal dimension $D > 1$) orientations.

The wavelet transform offers a multiscale decomposition of a function into a nested sequence of scaling spaces $V_0 \subset V_1 \subset \cdots \subset V_j \subset \cdots$. Each scaling space is spanned by a discrete collection of dyadic translations of a lowpass scaling function φ_j. The collection of wavelets at a particular scale j spans the difference between adjacent scaling spaces V_j and V_{j-1}. (Each wavelet function at scale j is concentrated approximately on some dyadic hypercube X_j, and between scales, both the wavelets and scaling functions are "self-similar," differing only by rescaling and dyadic dilation.) When $D > 1$, the difference spaces are partitioned into $2^D - 1$ distinct orientations (when $D = 2$ these correspond to vertical, horizontal, and diagonal directions). The wavelet transform can be truncated at any scale j. We then let the basis Ψ consist of all scaling functions at scale j plus all wavelets at scales j and finer.

Wavelets are essentially bandpass functions that detect abrupt changes in a signal. The scale of a wavelet, which controls its support both in time and in frequency, also controls its sensitivity to changes in the signal. This is made more precise by considering the wavelet analysis of smooth signals. Wavelet are often characterized by their number of *vanishing moments*; a wavelet basis function is said to have H vanishing moments if it is orthogonal to (its inner product is zero against) any H-degree polynomial. Section 2.4.2 discusses further the wavelet analysis of smooth and piecewise smooth signals.

The dyadic organization of the wavelet transform lends itself to a multiscale, tree-structured organization of the wavelet coefficients. Each "parent" function, concentrated on a dyadic hypercube X_j of sidelength 2^{-j}, has 2^D "children" whose supports are concentrated on the dyadic subdivisions of X_j. This relationship can be represented in a top-down tree structure. Because the parent and children share a location, they will presumably measure related phenomena about the signal, and so in general, any patterns in their wavelet coefficients tend to be reflected in the connectivity of the tree structure.

In addition to their ease of modeling, wavelets are computationally attractive for signal processing; using a filter bank, the wavelet transform of an N-voxel signal can be computed in just $O(N)$ operations.

2.3.4 Other dictionaries

A wide variety of other dictionaries have been proposed in signal processing and harmonic analysis. As one example, complex-valued wavelet transforms have proven useful for image analysis and modeling [65–71], thanks to a phase component that captures location information at each scale. Just a few of the other harmonic dictionaries popular in image processing include wavelet packets [5], Gabor atoms [5], curvelets [13,14], and contourlets [72,73], all of which involve various space-frequency partitions. We mention additional dictionaries in Section 2.6, and we also discuss in Chapter 3 alternative methods for signal representation such as *tilings*, where precisely one atom from the dictionary is used to describe the signal at each part of the domain (and these atoms "tile" together to cover the entire domain).

2.4 Low-Dimensional Signal Models

We now survey some common and important models in signal processing, each of which involves some notion of conciseness to the signal structure. We see in each case that this conciseness gives rise to a low-dimensional geometry within the ambient signal space.

2.4.1 Linear models

Some of the simplest models in signal processing correspond to *linear subspaces* of the ambient signal space. Bandlimited signals are one such example. Supposing, for example, that a 2π-periodic signal f has Fourier transform $F(\omega) = 0$ for $|\omega| > B$, the Shannon/Nyquist sampling theorem [5] states that such signals can be reconstructed from $2B$ samples. Because the space of B-bandlimited signals is closed under addition and scalar multiplication, it follows that the set of such signals forms a $2B$-dimensional linear subspace of $L^2([0, 2\pi))$.

Linear signal models also appear in cases where a model dictates a *linear constraint* on a signal. Considering a discrete length-N signal x, for example, such a constraint can be written in matrix form as

$$Ax = 0$$

for some $M \times N$ matrix A. Signals obeying such a model are constrained to live in $\mathcal{N}(A)$ (again, obviously, a linear subspace of $\mathbb{R}^N$).

A very similar class of models concerns signals living in an affine space, which can be represented for a discrete signal using

$$Ax = y.$$

The class of such x lives in a shifted nullspace $\widehat{x} + \mathcal{N}(A)$, where $\widehat{x}$ is any solution to the equation $A\widehat{x} = y$.

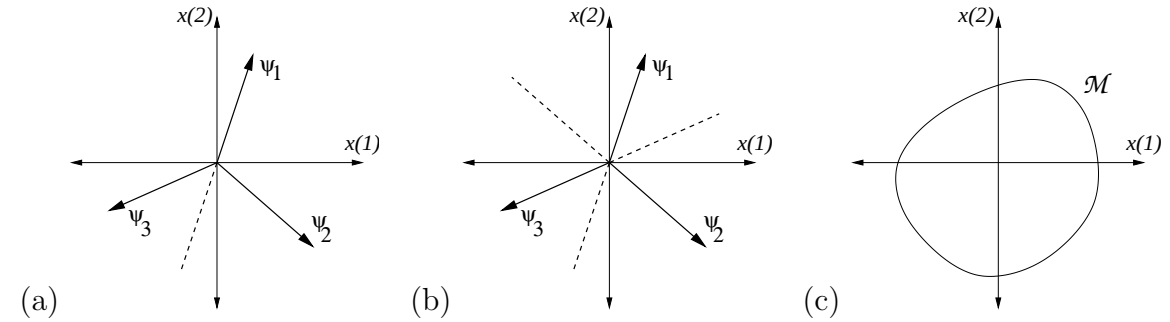

Figure 2.4: *Simple models for signals in* $\mathbb{R}^2$*. (a) The linear space spanned by one element of the dictionary* Ψ*. (b) The nonlinear set of* 1*-sparse signals that can be built using* Ψ*. (c) A manifold* $\mathcal{M}$*.*

Revisiting the dictionary setting (see Section 2.3), one last important linear model arises in cases where we select K specific elements from the dictionary Ψ and then construct signals using linear combinations of only these K elements; in this case the set of possible signals forms a K-dimensional hyperplane in the ambient signal space (see Figure 2.4(a)).

For example, we may construct low-frequency signals using combinations of only the lowest frequency sinusoids from the Fourier dictionary. Similar subsets may be chosen from the wavelet dictionary; in particular, one may choose only elements that span a particular scaling space V_j. As we have mentioned previously, harmonic dictionaries such as sinusoids and wavelets are well-suited to representing smooth signals. This can be seen in the decay of their transform coefficients. For example, we can relate the smoothness of a continuous 1-D function f to the decay of its Fourier coefficients $F(\omega)$; in particular, if $\int |F(\omega)|(1 + |\omega|^H)d\omega < \infty$, then $f \in \mathcal{C}^H$ [5]. Wavelet coefficients exhibit a similar decay for smooth signals: supposing $f \in \mathcal{C}^H$ and the wavelet basis function has at least H vanishing moments, then as the scale $j \to \infty$, the magnitudes of the wavelet coefficients decay as $2^{-j(H+1/2)}$ [5]. (Recall from Section 2.1.4 that $f \in \mathcal{C}^H$ implies f is well-approximated by a polynomial, and so due the vanishing moments this polynomial will have zero contribution to the wavelet coefficients.) Indeed, these results suggest that the largest coefficients tend to concentrate at the coarsest scales (lowest-frequencies). In Section 2.5.1, we see that linear approximations formed from just the lowest frequency elements of the Fourier or wavelet dictionaries provide very accurate approximations to smooth signals.

2.4.2 Sparse (nonlinear) models

Sparse signal models can be viewed as a generalization of linear models. The notion of sparsity comes from the fact that, by the proper choice of dictionary Ψ, many real-world signals $x = \Psi\alpha$ have coefficient vectors α containing few large entries, but across different signals the locations (indices in α) of the large entries may change.

We say a signal is strictly sparse (or "K-sparse") if all but K entries of α are zero.

Some examples of real-world signals for which sparse models have been proposed include neural spike trains (in time), music and other audio recordings (in time and frequency), natural images (in the wavelet or curvelet dictionaries [5, 8–14]), video sequences (in a 3-D wavelet dictionary [74, 75]), and sonar or radar pulses (in a chirplet dictionary [76]). In each of these cases, the relevant information in a sparse representation of a signal is encoded in both the *locations* (indices) of the significant coefficients and the *values* to which they are assigned. This type of uncertainty is an appropriate model for many natural signals with punctuated phenomena.

Sparsity is a *nonlinear* model. In particular, let Σ_K denote the set of all K-sparse signals for a given dictionary. It is easy to see that the set Σ_K is not closed under addition. (In fact, $\Sigma_K + \Sigma_K = \Sigma_{2K}$.) From a geometric perspective, the set of all K-sparse signals from the dictionary Ψ forms not a hyperplane but rather a union of K-dimensional hyperplanes, each spanned by K vectors of Ψ (see Figure 2.4(b)). For a dictionary Ψ with Z entries, there are $\binom{Z}{K}$ such hyperplanes. (The geometry of sparse signal collections has also been described in terms of orthosymmetric sets; see [77].)

Signals that are not strictly sparse but rather have a few "large" and many "small" coefficients are known as *compressible* signals. The notion of compressibility can be made more precise by considering the rate at which the *sorted* magnitudes of the coefficients α decay, and this decay rate can in turn be related to the ℓ_p norm of the coefficient vector α. Letting $\widetilde{\alpha}$ denote a rearrangement of the vector α with the coefficients ordered in terms of decreasing magnitude, then the reordered coefficients satisfy [78]

$$\widetilde{\alpha}_k \leq \|\alpha\|_{\ell_p} k^{-1/p}. \tag{2.2}$$

As we discuss in Section 2.5.2, these decay rates play an important role in *nonlinear approximation*, where adaptive, K-sparse representations from the dictionary are used to approximate a signal.

We recall from Section 2.4.1 that for a smooth signal f, the largest Fourier and wavelet coefficients tend to cluster at coarse scales (low frequencies). Suppose, however, that the function f is piecewise smooth; i.e., it is $\mathcal{C}^H$ at every point $t \in \mathbb{R}$ except for one point t_0, at which it is discontinuous. Naturally, this phenomenon will be reflected in the transform coefficients. In the Fourier domain, this discontinuity will have a global effect, as the overall smoothness of the function f has been reduced dramatically from H to 0. Wavelet coefficients, however, depend only on local signal properties, and so the wavelet basis functions whose supports do not include t_0 will be unaffected by the discontinuity. Coefficients surrounding the singularity will decay only as $2^{-j/2}$, but there are relatively few such coefficients. Indeed, at each scale there are only $O(1)$ wavelets that include t_0 in their supports, but these locations are highly signal-dependent. (For modeling purposes, these significant coefficients will persist through scale down the parent-child tree structure.) After reordering by magnitude, the wavelet coefficients of piecewise smooth signals will have the same general decay

rate as those of smooth signals. In Section 2.5.2, we see that the quality of nonlinear approximations offered by wavelets for smooth 1-D signals is not hampered by the addition of a finite number of discontinuities.

2.4.3 Manifold models

Manifold models generalize the conciseness of sparsity-based signal models. In particular, in many situations where a signal is believed to have a concise description or "few degrees of freedom," the result is that the signal will live on or near a particular submanifold of the ambient signal space.

Parametric models

We begin with an abstract motivation for the manifold perspective. Consider a signal f (such as a natural image), and suppose that we can identify some single 1-D piece of information about that signal that could be variable; that is, other signals might rightly be called "similar" to f if they differ only in this piece of information. (For example, this 1-D parameter could denote the distance from some object in an image to the camera.) We let θ denote the variable parameter and write the signal as f_θ to denote its dependence on θ. In a sense, θ is a single "degree of freedom" driving the generation of the signal f_θ under this simple model. We let Θ denote the set of possible values of the parameter θ. If the mapping between θ and f_θ is well-behaved, then the collection of signals $\{f_\theta : \theta \in \Theta\}$ forms a 1-D path in the ambient signal space.

More generally, when a signal has K degrees of freedom, we may model it as depending on some parameter θ that is chosen from a K-dimensional manifold Θ. (The parameter space Θ could be, for example, a subset of $\mathbb{R}^K$, or it could be a more general manifold such as SO(3).) We again let f_θ denote the signal corresponding to a particular choice of θ, and we let $\mathcal{F} = \{f_\theta : \theta \in \Theta\}$. Assuming the mapping f is continuous and injective over Θ (and its inverse is continuous), then by virtue of the manifold structure of Θ, its image $\mathcal{F}$ will correspond to a K-dimensional manifold embedded in the ambient signal space (see Figure 2.4(c)).

These types of parametric models arise in a number of scenarios in signal processing. Examples include: signals of unknown translation, sinusoids of unknown frequency (across a continuum of possibilities), linear radar chirps described by a starting and ending time and frequency, tomographic or light field images with articulated camera positions, robotic systems with few physical degrees of freedom, dynamical systems with low-dimensional attractors [41, 42], and so on.

In general, parametric signals manifolds are *nonlinear* (by which we mean non-affine as well); this can again be seen by considering the sum of two signals $f_{\theta_0} + f_{\theta_1}$. In many interesting situations, signal manifolds are *non-differentiable* as well. In Chapter 4, we study this issue in much more detail.

Nonparametric models

Manifolds have also been used to model signals for which there is no known parametric model. Examples include images of faces and handwritten digits [43,53], which have been found empirically to cluster near low-dimensional manifolds. Intuitively, because of the configurations of human joints and muscles, it may be conceivable that there are relatively "few" degrees of freedom driving the appearance of a human face or the style of handwriting; however, this inclination is difficult or impossible to make precise. Nonetheless, certain applications in face and handwriting recognition have benefitted from algorithms designed to discover and exploit the nonlinear manifold-like structure of signal collections. Section 2.7.1 discusses such methods for learning parametrizations and other information from data living along manifolds.

Much more generally, one may consider, for example, the set of *all* natural images. Clearly, this set has small volume with respect to the ambient signal space — generating an image randomly pixel-by-pixel will almost certainly produce an unnatural noise-like image. Again, it is conceivable that, at least locally, this set may have a low-dimensional manifold-like structure: from a given image, one may be able to identify only a limited number of meaningful changes that could be performed while still preserving the natural look to the image. Arguably, most work in signal modeling could be interpreted in some way as a search for this overall structure. As part of this thesis, however, we hope to contribute explicitly to the geometric understanding of signal models.

2.5 Approximation

To this point, we have discussed signal representations and models as basic tools for signal processing. In the remainder of this chapter, we discuss the actual application of these tools to tasks such as approximation and compression, and we continue to discuss the geometric implications.

2.5.1 Linear approximation

One common prototypical problem in signal processing is to find the best linear approximation to a signal x. By "best linear approximation," we mean the best approximation to x from among a class of signals comprising a linear (or affine) subspace. This situation may arise, for example, when we have a noisy observation of a signal believed to obey a linear model. If we choose an ℓ_2 error criterion, the solution to this optimization problem has a particularly strong geometric interpretation.

To be more concrete, suppose S is a K-dimensional linear subspace of $\mathbb{R}^N$. (The case of an affine subspace follows similarly.) If we seek

$$s^* := \arg\min_{s \in S} \|s - x\|_2 ,$$

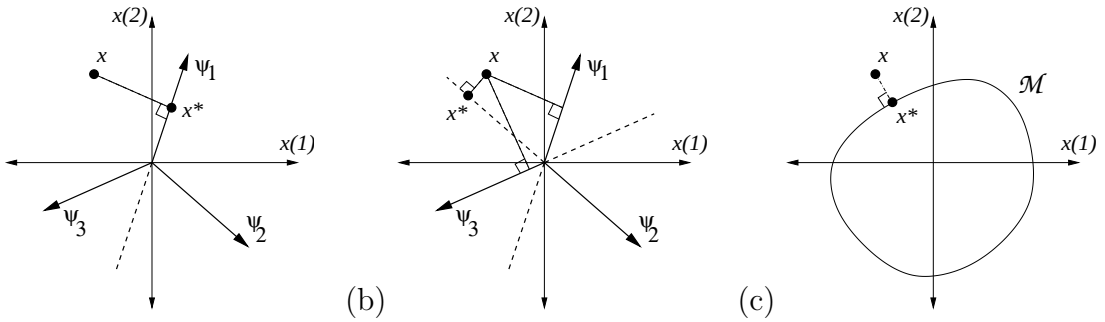

Figure 2.5: *Approximating a signal $x \in \mathbb{R}^2$ with an ℓ_2 error criterion. (a) Linear approximation using one element of the dictionary Ψ. (b) Nonlinear approximation, choosing the best 1-sparse signal that can be built using Ψ. (c) Manifold-based approximation, finding the nearest point on $\mathcal{M}$.*

standard linear algebra results state that the minimizer is given by

$$s^* = A^T Ax, \tag{2.3}$$

where A is a $K \times N$ matrix whose rows form an orthonormal basis for S. Geometrically, one can easily see that this solution corresponds to an orthogonal projection of x onto the subspace S (see Figure 2.5(a)).

The linear approximation problem arises frequently in settings involving signal dictionaries. In some settings, such as the case of an oversampled bandlimited signal, certain coefficients in the vector α may be assumed to be fixed at zero. In the case where the dictionary Ψ forms an orthonormal basis, the linear approximation estimate of the unknown coefficients has a particularly simple form: rows of the matrix A in (2.3) are obtained by selecting and transposing the columns of Ψ whose expansion coefficients are unknown, and consequently, the unknown coefficients can be estimated simply by taking the inner products of x against the appropriate columns of Ψ.

For example, in choosing a fixed subset of the Fourier or wavelet dictionaries, one may rightfully choose the lowest frequency (coarsest scale) basis functions for the set S because, as discussed in Section 2.4.1, the coefficients generally tend to decay at higher frequencies (finer scales). For smooth functions, this strategy is appropriate and effective; functions in Sobolev smoothness spaces are well-approximated using linear approximations from the Fourier or wavelet dictionaries [5]. For piecewise smooth functions, however, even the wavelet-domain linear approximation strategy would miss out on significant coefficients at fine scales. Since the locations of such coefficients are unknown a priori, it is impossible to propose a linear wavelet-domain approximation scheme that could simultaneously capture all piecewise smooth signals.

2.5.2 Nonlinear approximation

A related question often arises in settings involving signal dictionaries. Rather than finding the best approximation to a signal f using a fixed collection of K elements from the dictionary Ψ, one may often seek the best K-term representation to f among all possible expansions that use K terms from the dictionary. Compared to linear approximation, this type of nonlinear approximation [6, 7] utilizes the ability of the dictionary to adapt: different elements may be important for representing different signals.

The K-term nonlinear approximation problem corresponds to the optimization

$$s^*_{K,p} := \arg\min_{s \in \Sigma_K} \|s - f\|_p. \tag{2.4}$$

(For the sake of generality, we consider general L_p and ℓ_p norms in this section.) Due to the nonlinearity of the set Σ_K for a given dictionary, solving this problem can be difficult. Supposing Ψ is an orthonormal basis and $p = 2$, the solution to (2.4) is easily obtained by thresholding: compute the coefficients α and keep the K largest. The approximation error is then given simply by

$$\|s^*_{K,2} - f\|_2 = \left(\sum_{k>K} \widetilde{\alpha}_k^2 \right)^{1/2}.$$

When Ψ is a redundant dictionary, however, the situation is much more complicated. We mention more on this below (see also Figure 2.5(b)).

Measuring approximation quality

One common measure for the quality of a dictionary Ψ in approximating a signal class is the fidelity of its K-term representations. Often one examines the asymptotic rate of decay of the K-term approximation error as K grows large. Defining

$$\sigma_K(f)_p := \|s^*_{K,p} - f\|_p, \tag{2.5}$$

for a given signal f we may consider the asymptotic decay of $\sigma_K(f)_p$ as $K \to \infty$. (We recall the dependence of (2.4) and hence (2.5) on the dictionary Ψ.) In many cases, the function $\sigma_K(f)_p$ will decay as K^{-r} for some r, and when Ψ represents a harmonic dictionary, faster decay rates tend to correspond to smoother functions. Indeed, one can show that when Ψ is an orthonormal basis, then $\sigma_K(f)_2$ will decay as K^{-r} if and only if $\widetilde{\alpha}_k$ decays as $k^{-r+1/2}$ [78].

Nonlinear approximation of piecewise smooth functions

Let $f \in \mathcal{C}^H$ be a 1-D function. Supposing the wavelet dictionary has more than H vanishing moments, then f can be well approximated using its K largest coefficients

(most of which are at coarse scales). As K grows large, the nonlinear approximation error will decay[4] as $\sigma_K(f)_2 \lesssim K^{-H}$.

Supposing that f is piecewise smooth, however, with a finite number of discontinuities, then (as discussed in Section 2.4.2) f will have a limited number of significant wavelet coefficients at fine scales. Because of the concentration of these significant coefficients within each scale, the nonlinear approximation rate will remain $\sigma_K(f)_2 \lesssim K^{-H}$ as if there were no discontinuities present [5].

Unfortunately, this resilience of wavelets to discontinuities does not extend to higher dimensions. Suppose, for example, that f is a $\mathcal{C}^H$ smooth 2-D signal. Assuming the proper number of vanishing moments, a wavelet representation will achieve the optimal nonlinear approximation rate $\sigma_K(f)_2 \lesssim K^{-H/2}$ [5,79]. As in the 1-D case, this approximation rate is maintained when a finite number of point discontinuities are introduced into f. However, when f contains 1-D discontinuities (edges separating the smooth regions), the approximation rate will fall to $\sigma_K(f)_2 \lesssim K^{-1/2}$ [5]. The problem actually arises due to the isotropic, dyadic supports of the wavelets; instead of $O(1)$ significant wavelets at each scale, there are now $O(2^j)$ wavelets overlapping the discontinuity. We revisit this important issue in Section 2.6.

Finding approximations

As mentioned above, in the case where Ψ is an orthonormal basis and $p = 2$, the solution to (2.4) is easily obtained by thresholding: compute the coefficients α and keep the K largest. Thresholding can also be shown to be optimal for arbitrary ℓ_p norms in the special case where Ψ is the canonical basis. While the optimality of thresholding does not generalize to arbitrary norms and bases, thresholding can be shown to be a near-optimal approximation strategy for wavelet bases with arbitrary L_p norms [78].

In the case where Ψ is a redundant dictionary, however, the expansion coefficients α are not unique, and the optimization problem (2.4) can be much more difficult to solve. Indeed, supposing even that an *exact* K-term representation exists for f in the dictionary Ψ, finding that K-term approximation is NP-complete in general, requiring a combinatorial enumeration of the $\binom{Z}{K}$ possible sparse subspaces [28]. This search can be recast as the optimization problem

$$\widehat{\alpha} = \arg\min \|\alpha\|_0 \quad \text{s.t. } f = \Psi\alpha. \tag{2.6}$$

While solving (2.6) is prohibitively complex, a variety of algorithms have been proposed as alternatives. One approach convexifies the optimization problem by replacing the ℓ_0 fidelity criterion by an ℓ_1 criterion

$$\widehat{\alpha} = \arg\min \|\alpha\|_1 \quad \text{s.t. } f = \Psi\alpha.$$

[4]We use the notation $f(\alpha) \lesssim g(\alpha)$, or $f(\alpha) = O(g(\alpha))$, if there exists a constant C, possibly large but not dependent on the argument α, such that $f(\alpha) \leq Cg(\alpha)$.

This problem, known as Basis Pursuit [80], is significantly more approachable and can be solved with traditional linear programming techniques whose computational complexities are polynomial in Z. Iterative greedy algorithms such as Matching Pursuit (MP) and Orthogonal Matching Pursuit (OMP) [5] have also been suggested to find sparse representations α for a signal f. Both MP and OMP iteratively select the columns from Ψ that are most correlated with f, then subtract the contribution of each column, leaving a residual. OMP includes an additional step at each iteration where the residual is orthogonalized against the previously selected columns.

2.5.3 Manifold approximation

We also consider the problem of finding the best manifold-based approximation to a signal (see Figure 2.5(c)). Suppose that $\mathcal{F} = \{f_\theta : \theta \in \Theta\}$ is a parametrized K-dimension manifold and that we are given a signal I that is believed to approximate f_θ for an unknown $\theta \in \Theta$. From I we wish to recover an estimate of θ. Again, we may formulate this parameter estimation problem as an optimization, writing the objective function (here we concentrate solely on the L_2 or ℓ_2 case)

$$D(\theta) = \|f_\theta - I\|_2^2$$

and solving for

$$\theta^* = \arg\min_{\theta \in \Theta} D(\theta).$$

We suppose that the minimum is uniquely defined.

Standard nonlinear parameter estimation [81] tells us that, if D is differentiable, we can use Newton's method to iteratively refine a sequence of guesses $\theta^{(0)}, \theta^{(1)}, \theta^{(2)}, \ldots$ to θ^* and rapidly convergence to the true value. Supposing that $\mathcal{F}$ is a *differentiable* manifold, we would let

$$J = [\partial D/\partial\theta_0 \;\; \partial D/\partial\theta_1 \;\; \ldots \;\; \partial D/\partial\theta_{K-1}]^T$$

be the gradient of D, and let H be the $K \times K$ Hessian, $H_{ij} = \frac{\partial^2 D}{\partial\theta_i \partial\theta_j}$. Assuming D is differentiable, Newton's method specifies the following update step:

$$\theta^{(k+1)} \leftarrow \theta^{(k)} + [H(\theta^{(k)})]^{-1} J(\theta^{(k)}).$$

To relate this method to the structure of the manifold, we can actually express the gradient and Hessian in terms of signals, writing

$$D(\theta) = \|f_\theta - I\|_2^2 = \int (f_\theta - I)^2 \, dx = \int f_\theta^2 - 2 I f_\theta + I^2 \, dx.$$

Differentiating with respect to component θ_i, we obtain

$$\begin{aligned}
\frac{\partial D}{\partial \theta_i} \;=\; J_i \;&=\; \frac{\partial}{\partial \theta_i}\left(\int f_\theta^2 - 2I f_\theta + I^2 \; dx\right) \\
&=\; \int \frac{\partial}{\partial \theta_i}(f_\theta^2) - 2I\frac{\partial}{\partial \theta_i} f_\theta \; dx \\
&=\; \int 2 f_\theta \tau_\theta^i - 2I\tau_\theta^i \; dx \\
&=\; 2\langle f_\theta - I, \tau_\theta^i \rangle,
\end{aligned}$$

where $\tau_\theta^i = \frac{\partial f_\theta}{\partial \theta_i}$ is a tangent signal. Continuing, we examine the Hessian,

$$\begin{aligned}
\frac{\partial^2 D}{\partial \theta_i \partial \theta_j} \;=\; H_{ij} \;&=\; \frac{\partial}{\partial \theta_j}\left(\frac{\partial D}{\partial \theta_i}\right) \\
&=\; \int \frac{\partial}{\partial \theta_j}\left(2 f_\theta \tau_\theta^i - 2I\tau_\theta^i\right) \; dx \\
&=\; \int 2\tau_\theta^i \tau_\theta^j + 2 f_\theta \tau_\theta^{ij} - 2I\tau_\theta^{ij} \; dx \\
&=\; 2\langle \tau_\theta^i, \tau_\theta^j \rangle + 2\langle f_\theta - I, \tau_\theta^{ij} \rangle, \qquad (2.7)
\end{aligned}$$

where $\tau_\theta^{ij} = \frac{\partial^2 f_\theta}{\partial \theta_i \partial \theta_j}$ denotes a second-derivative signal. Thus, we can interpret Newton's method geometrically as (essentially) a sequence of successive projections onto tangent spaces on the manifold.

Again, the above discussion assumes the manifold to be differentiable. However, as we discuss in Chapter 4, many interesting parametric signal manifolds are in fact nowhere differentiable — the tangent spaces demanded by Newton's method do not exist. However, we do identify a type of multiscale tangent structure to the manifold that permits a coarse-to-fine technique for parameter estimation. Section 4.5.2 details our Multiscale Newton method.

2.6 Compression

2.6.1 Transform coding

In Section 2.5.2, we measured the quality of a dictionary in terms of its K-term approximations to signals drawn from some class. One reason that such approximations are desirable is that they provide concise descriptions of the signal that can be easily stored, processed, etc. There is even speculation and evidence that neurons in the human visual system may use sparse coding to represent a scene [82].

For data compression, conciseness is often exploited in a popular technique known as *transform coding*. Given a signal f (for which a concise description may not be readily apparent in its native domain), the idea is simply to use the dictionary Ψ to

transform f to its coefficients α, which can then be efficiently and easily described. As discussed above, perhaps the simplest strategy for summarizing a sparse α is simply to threshold, keeping the K largest coefficients and discarding the rest. A simple encoder would then just encode the positions and quantized values of these K coefficients.

2.6.2 Metric entropy

Suppose f is a function and let $\widehat{f_R}$ be an approximation to f encoded using R bits. To evaluate the quality of a coding strategy, it is common to consider the *asymptotic rate-distortion* (R-D) performance, which measures the decay rate of $\|f - \widehat{f_R}\|_{L_p}$ as $R \to \infty$. The *metric entropy* [57] for a class $\mathcal{F}$ gives the best decay rate that can be achieved uniformly over all functions $f \in \mathcal{F}$. We note that this is a true measure for the complexity of a class and is tied to no particular dictionary or encoding strategy. The metric entropy also has a very geometric interpretation, as it relates to the smallest radius possible for a covering of 2^R balls over the set $\mathcal{F}$.

Metric entropies are known for certain signal classes. For example, the results of Clements [58] (extending those of Kolmogorov and Tihomirov [57]) regarding metric entropy give bounds on the optimal achievable asymptotic rate-distortion performance for D-dimensional $\mathcal{C}^H$-smooth functions f (see also [79]):

$$\left\| f - \widehat{f_R} \right\|_{L_p} \lesssim \left(\frac{1}{R} \right)^{\frac{H}{D}}.$$

Rate-distortion performance measures the complexity of a representation and encoding strategy. In the case of transform coding, for example, R-D results account for the bits required to encode both the values of the significant coefficients *and* their locations. Nonetheless, in many cases transform coding is indeed an effective strategy for encoding signals that have sparse representations [7]. For example, in [79] Cohen et al. propose a wavelet-domain coder that uses a connected-tree structure to efficiently encode the positions of the significant coefficients and prove that this encoding strategy achieves the optimal rate

$$\left\| f - \widehat{f_R} \right\|_{L_p} \lesssim \left(\frac{1}{R} \right)^{\frac{H}{D}}.$$

2.6.3 Compression of piecewise smooth images

In some cases, however, the sparsity of the wavelet transform may not reflect the true underlying structure of a signal. Examples are 2-D piecewise smooth signals with a smooth edge discontinuity separating the smooth regions. As we discussed in Section 2.5.2, wavelets fail to sparsely represent these functions, and so the R-D performance for simple thresholding-based coders will suffer as well. In spite of all

of the benefits of wavelet representations for signal processing (low computational complexity, tree structure, sparse approximations for smooth signals), this failure to efficiently represent edges is a significant drawback. In many images, edges carry some of the most prominent and important information [83], and so it is desirable to have a representation well-suited to compressing edges in images.

To address this concern, recent work in harmonic analysis has focused on developing representations that provide sparse decompositions for certain geometric image classes. Examples include curvelets [13, 14] and contourlets [73], slightly redundant tight frames consisting of anisotropic, "needle-like" atoms. In [84], bandelets are formed by warping an orthonormal wavelet basis to conform to the geometrical structure in the image. A nonlinear multiscale transform that adapts to discontinuities (and can represent a "clean" edge using very few coarse scale coefficients) is proposed in [85]. Each of these new representations has been shown to achieve near-optimal asymptotic approximation and R-D performance for piecewise smooth images consisting of C^H regions separated by discontinuities along C^H curves, with $H = 2$ ($H \geq 2$ for bandelets). Some have also found use in specialized compression applications such as identification photos [86].

In Chapter 3, we propose an alternative approach for representing and compressing piecewise smooth images *in the wavelet domain*, demonstrating that the lack of wavelet sparsity can be overcome by using joint tree-based models for wavelet coefficients. Our scheme is based on the simple yet powerful observation that geometric features can be efficiently approximated using local, geometric atoms in the spatial domain, and that the projection of these geometric primitives onto wavelet subspaces can therefore approximate the corresponding wavelet coefficients. We prove that the resulting dictionary achieves the optimal nonlinear approximation rates for piecewise smooth signal classes. To account for the added complexity of this encoding strategy, we also consider R-D results and prove that this scheme comes within a logarithmic factor of the optimal performance rate. Unlike the techniques mentioned above, our method also generalizes to arbitrary orders of smoothness and arbitrary signal dimension.

2.7 Dimensionality Reduction

Recent years have seen a proliferation of novel techniques for what can loosely be termed "dimensionality reduction." Like the tasks of approximation and compression discussed above, these methods involve some aspect in which low-dimensional information is extracted about a signal or collection of signals in some high-dimensional ambient space. Unlike the tasks of approximation and compression, however, the goal of these methods is not always to maintain a faithful representation of each signal. Instead, the purpose may be to preserve some critical relationships among elements of a data set or to discover information about a manifold on which the data lives.

In this section, we review two general methods for dimensionality reduction. Sec-

tion 2.7.1 begins with a brief overview of techniques for manifold learning. Section 2.7.2 then discusses the Johnson-Lindenstrauss (JL) lemma, which concerns the isometric embedding of a cloud points as it is projected to a lower-dimensional space. Though at first glance the JL lemma does not pertain to any of the low-dimensional signal models we have previously discussed, we later see (Section 2.8.6) that the JL lemma plays a critical role in the core theory of CS, and we also employ the JL lemma in developing a theory for isometric embeddings of manifolds (Theorem 6.2).

2.7.1 Manifold learning

Several techniques have been proposed for *manifold learning* in which a set of points sampled from a K-dimensional submanifold of $\mathbb{R}^N$ are mapped to some lower dimension $\mathbb{R}^M$ (ideally, $M = K$) while preserving some characteristic property of the manifold. Examples include ISOMAP [44], Hessian Eigenmaps (HLLE) [45], and Maximum Variance Unfolding (MVU) [46], which attempt to learn isometric embeddings of the manifold (preserving pairwise geodesic distances); Locally Linear Embedding (LLE) [47], which attempts to preserve local linear neighborhood structures among the embedded points; Local Tangent Space Alignment (LTSA) [48], which attempts to preserve local coordinates in each tangent space; and a method for charting a manifold [49] that attempts to preserve local neighborhood structures. These algorithms can be useful for learning the dimension and parametrizations of manifolds, for sorting data, for visualization and navigation through the data, and as preprocessing to make further analysis more tractable; common demonstrations include analysis of face images and classification of and handwritten digits. A related technique, the Whitney Reduction Network [41, 42], seeks a linear mapping to $\mathbb{R}^M$ that preserves ambient pairwise distances on the manifold and is particularly useful for processing the output of dynamical systems having low-dimensional attractors.

Other algorithms have been proposed for characterizing manifolds from sampled data without constructing an explicit embedding in $\mathbb{R}^M$. The Geodesic Minimal Spanning Tree (GMST) [50] models the data as random samples from the manifold and estimates the corresponding entropy and dimensionality. Another technique [51] has been proposed for using random samples of a manifold to estimate its homology (via the Betti numbers, which essentially characterize its dimension, number of connected components, etc.). Persistence Barcodes [52] are a related technique that involves constructing a type of signature for a manifold (or simply a shape) that uses tangent complexes to detect and characterize local edges and corners.

Additional algorithms have been proposed for constructing meaningful functions on the point samples in $\mathbb{R}^N$. To solve a semi-supervised learning problem, a method called Laplacian Eigenmaps [53] has been proposed that involves forming an adjacency graph for the data in $\mathbb{R}^N$, computing eigenfunctions of the Laplacian operator on the graph (which form a basis for L_2 on the graph), and using these functions to train a classifier on the data. The resulting classifiers have been used for handwritten digit recognition, document classification, and phoneme classification. (The M smoothest

eigenfunctions can also be used to embed the manifold in M, similar to the approaches described above.) A related method called Diffusion Wavelets [54] uses powers of the diffusion operator to model scale on the manifold, then constructs wavelets to capture local behavior at each scale. The result is a wavelet transform adapted not to geodesic distance but to diffusion distance, which measures (roughly) the number of paths connecting two points.

2.7.2 The Johnson-Lindenstrauss lemma

As with the above techniques in manifold learning, the Johnson-Lindenstrauss (JL) lemma [87–90] provides a method for dimensionality reduction of a set of data in $\mathbb{R}^N$. Unlike manifold-based methods, however, the JL lemma can be used for any arbitrary set Q of points in $\mathbb{R}^N$; the data set is not assumed to have any a priori structure.

Despite the apparent lack of structure, the JL lemma suggests that the data set Q *does* carry information that can be preserved when the data is mapped to a lower-dimensional space $\mathbb{R}^M$. In particular, the original formulation of the JL lemma [87] states that there exists a Lipschitz mapping $\Phi : \mathbb{R}^N \mapsto \mathbb{R}^M$ with $M = O(\log(\#Q))$ such that all pairwise distances between points in Q are approximately preserved. This fact is useful for solving problems such as *Approximate Nearest Neighbor* [90], in which one desires the nearest point in Q to some query point $y \in \mathbb{R}^N$ (but a solution not much further than the optimal point is also acceptable). Such problems can be solved significantly more quickly in $\mathbb{R}^M$ than in $\mathbb{R}^N$.

Recent reformulations of the JL lemma propose random linear operators that, with high probability, will ensure a near isometric embedding. These typically build on concentration of measure results such as the following.

Lemma 2.4 [88, 89] *Let $x \in \mathbb{R}^N$, fix $0 < \epsilon < 1$, and let Φ be a matrix constructed in one of the following two manners:*

1. *Φ is a random $M \times N$ matrix with i.i.d. $\mathcal{N}(0, \sigma^2)$ entries, where $\sigma^2 = 1/N$, or*
2. *Φ is random orthoprojector from $\mathbb{R}^N$ to $\mathbb{R}^M$.*

Then with probability exceeding

$$1 - 2\exp\left(-\frac{M(\epsilon^2/2 - \epsilon^3/3)}{2}\right),$$

the following holds:

$$(1-\epsilon)\sqrt{\frac{M}{N}} \leq \frac{\|\Phi x\|_2}{\|x\|_2} \leq (1+\epsilon)\sqrt{\frac{M}{N}}. \tag{2.8}$$

The random orthoprojector referred to above is clearly related to the first case (simple matrix multiplication by a Gaussian Φ) but subtly different; one could think of constructing a random Gaussian Φ, then using Gram-Schmidt to orthonormalize the rows before multiplying x. (This adjustment could even be made *after* computing Φx, a fact which is possibly more relevant for results such as Theorem 6.2.) We note also that simple rescaling of Φ can be used to eliminate the $\sqrt{\frac{M}{N}}$ in (2.8); however we prefer this formulation for later reference.

By using the union bound over all $\binom{\#Q}{2}$ pairs of distinct points in Q, Lemma 2.4 can be used to prove a randomized version of the Johnson-Lindenstrauss lemma.

Lemma 2.5 (Johnson-Lindenstrauss) *Let Q be a finite collection of points in $\mathbb{R}^N$. Fix $0 < \epsilon < 1$ and $\beta > 0$. Set*

$$M \geq \left(\frac{4 + 2\beta}{\epsilon^2/2 - \epsilon^3/3} \right) \ln(\#Q).$$

Let Φ be a matrix constructed in one of the following two manners:

1. *Φ is a random $M \times N$ matrix with i.i.d. $\mathcal{N}(0, \sigma^2)$ entries, where $\sigma^2 = 1/N$, or*
2. *Φ is random orthoprojector from $\mathbb{R}^N$ to $\mathbb{R}^M$.*

Then with probability exceeding $1 - (\#Q)^{-\beta}$, the following statement holds: for every $x, y \in Q$,

$$(1 - \epsilon)\sqrt{\frac{M}{N}} \leq \frac{\|\Phi x - \Phi y\|_2}{\|x - y\|_2} \leq (1 + \epsilon)\sqrt{\frac{M}{N}}.$$

Indeed, [88] establishes that both Lemma 2.4 and Lemma 2.5 also hold when the elements of Φ are chosen i.i.d. from a random Rademacher distribution ($\pm\sigma$ with equal probability 1/2) or from a similar ternary distribution ($\pm\sqrt{3}\sigma$ with equal probability 1/6; 0 with probability 2/3). These can further improve the computational benefits of the JL lemma.

2.8 Compressed Sensing

A new theory known as Compressed Sensing (CS) has recently emerged that can also be categorized as a type of dimensionality reduction. Like manifold learning, CS is strongly model-based (relying on sparsity in particular). However, unlike many of the standard techniques in dimensionality reduction (such as manifold learning or the JL lemma), the goal of CS is to maintain a low-dimensional representation of a signal x from which a faithful approximation to x can be recovered. In a sense, this more closely resembles the traditional problem of data compression (see Section 2.6). In CS, however, the encoder requires no a priori knowledge of the signal structure.

Only the *decoder* uses the model (sparsity) to recover the signal. In Chapter 6, we will indeed see that without changing the CS encoder we can also recover manifold-modeled signals simply by changing the decoder. We justify such an approach again using geometric arguments.

2.8.1 Motivation

Consider a signal $x \in \mathbb{R}^N$, and suppose that the basis Ψ provides a K-sparse representation of x

$$x = \Psi\alpha,$$

with $\|\alpha\|_0 = K$. (In this section, we focus on exactly K-sparse signals, though many of the key ideas translate to compressible signals [20, 21]. In addition, we note that the CS concepts are also extendable to tight frames.)

As we discussed in Section 2.6, the standard procedure for compressing sparse signals, known as transform coding, is to (i) acquire the full N-sample signal x; (ii) compute the complete set of transform coefficients α; (iii) locate the K largest, significant coefficients and discard the (many) small coefficients; (iv) encode the *values and locations* of the largest coefficients.

This procedure has three inherent inefficiencies: First, for a high-dimensional signal, we must start with a large number of samples N. Second, the encoder must compute *all* N of the transform coefficients α, even though it will discard all but K of them. Third, the encoder must encode the locations of the large coefficients, which requires increasing the coding rate since the locations change with each signal.

2.8.2 Incoherent projections

This raises a simple question: For a given signal, is it possible to directly estimate the set of large $\alpha(n)$'s that will not be discarded? While this seems improbable, Candès, Romberg, and Tao [20, 22] and Donoho [21] have shown that a reduced set of projections can contain enough information to reconstruct sparse signals. An offshoot of this work, often referred to as *Compressed Sensing* (CS) [20, 21, 24–27, 29], has emerged that builds on this principle.

In CS, we do not measure or encode the K significant $\alpha(n)$ directly. Rather, we measure and encode $M < N$ projections $y(m) = \langle x, {\phi_m}^T \rangle$ of the signal onto a *second set* of functions $\{\phi_m\}, m = 1, 2, \ldots, M$. In matrix notation, we measure

$$y = \Phi x,$$

where y is an $M \times 1$ column vector and the *measurement basis* matrix Φ is $M \times N$ with each row a basis vector ϕ_m. Since $M < N$, recovery of the signal x from the measurements y is ill-posed in general; however the additional assumption of signal *sparsity* makes recovery possible and practical.

The CS theory tells us that when certain conditions hold, namely that the functions $\{\phi_m\}$ cannot sparsely represent the elements of the basis $\{\psi_n\}$ (a condition known as *incoherence* of the two dictionaries [20–22, 91]) and the number of measurements M is large enough, then it is indeed possible to recover the set of large $\{\alpha(n)\}$ (and thus the signal x) from a similarly sized set of measurements y. This incoherence property holds for many pairs of bases, including for example, delta spikes and the sine waves of a Fourier basis, or the Fourier basis and wavelets. Significantly, this incoherence also holds with high probability between an arbitrary fixed basis and a randomly generated one.

2.8.3 Methods for signal recovery

Although the problem of recovering x from y is ill-posed in general (because $x \in \mathbb{R}^N$, $y \in \mathbb{R}^M$, and $M < N$), it is indeed possible to recover *sparse* signals from CS measurements. Given the measurements $y = \Phi x$, there exist an infinite number of candidate signals in the shifted nullspace $\mathcal{N}(\Phi) + x$ that could generate the same measurements y (see Section 2.4.1). Recovery of the correct signal x can be accomplished by seeking a *sparse* solution among these candidates.

Recovery via ℓ_0 optimization

Supposing that x is exactly K-sparse in the dictionary Ψ, then recovery of x from y can be formulated as the ℓ_0 minimization

$$\widehat{\alpha} = \arg\min \|\alpha\|_0 \quad \text{s.t. } y = \Phi\Psi\alpha. \tag{2.9}$$

Given some technical conditions on Φ and Ψ (see Theorem 2.1 below), then with high probability this optimization problem returns the proper K-sparse solution α, from which the true x may be constructed. (Thanks to the incoherence between the two bases, if the original signal is sparse in the α coefficients, then no other set of sparse signal coefficients α' can yield the same projections y.) We note that the recovery program (2.9) can be interpreted as finding a K-term approximation to y from the columns of the dictionary $\Phi\Psi$, which we call the *holographic basis* because of the complex pattern in which it encodes the sparse signal coefficients [21].

In principle, remarkably few incoherent measurements are required to recover a K-sparse signal via ℓ_0 minimization. Clearly, more than K measurements must be taken to avoid ambiguity; the following theorem establishes that $K + 1$ random measurements will suffice. (Similar results were established by Venkataramani and Bresler [92].)

Theorem 2.1 *Let Ψ be an orthonormal basis for $\mathbb{R}^N$, and let $1 \leq K < N$. Then the following statements hold:*

1. *Let Φ be an $M \times N$ measurement matrix with i.i.d. Gaussian entries with $M \geq 2K$. Then with probability one the following statement holds: all signals $x = \Psi\alpha$ having expansion coefficients $\alpha \in \mathbb{R}^N$ that satisfy $\|\alpha\|_0 = K$ can be recovered uniquely from the M-dimensional measurement vector $y = \Phi x$ via the ℓ_0 optimization (2.9).*

2. *Let $x = \Psi\alpha$ such that $\|\alpha\|_0 = K$. Let Φ be an $M \times N$ measurement matrix with i.i.d. Gaussian entries (notably, independent of x) with $M \geq K + 1$. Then with probability one the following statement holds: x can be recovered uniquely from the M-dimensional measurement vector $y = \Phi x$ via the ℓ_0 optimization (2.9).*

3. *Let Φ be an $M \times N$ measurement matrix, where $M \leq K$. Then, aside from pathological cases (specified in the proof), no signal $x = \Psi\alpha$ with $\|\alpha\|_0 = K$ can be uniquely recovered from the M-dimensional measurement vector $y = \Phi x$.*

Proof: See Appendix A.

The second statement of the theorem differs from the first in the following respect: when $K < M < 2K$, there will necessarily exist K-sparse signals x that cannot be uniquely recovered from the M-dimensional measurement vector $y = \Phi x$. However, these signals form a set of measure zero within the set of *all* K-sparse signals and can safely be avoided if Φ is randomly generated independently of x.

Unfortunately, as discussed in Section 2.5.2, solving this ℓ_0 optimization problem is prohibitively complex. Yet another challenge is robustness; in the setting of Theorem 2.1, the recovery may be very poorly conditioned. In fact, *both* of these considerations (computational complexity and robustness) can be addressed, but at the expense of slightly more measurements.

Recovery via ℓ_1 optimization

The practical revelation that supports the new CS theory is that it is not necessary to solve the ℓ_0-minimization problem to recover α. In fact, a much easier problem yields an equivalent solution (thanks again to the incoherency of the bases); we need only solve for the ℓ_1-sparsest coefficients α that agree with the measurements y [20–22, 24–27, 29]

$$\widehat{\alpha} = \arg\min \|\alpha\|_1 \quad \text{s.t. } y = \Phi\Psi\alpha. \tag{2.10}$$

As discussed in Section 2.5.2, this optimization problem, also known as *Basis Pursuit* [80], is significantly more approachable and can be solved with traditional linear programming techniques whose computational complexities are polynomial in N.

There is no free lunch, however; according to the theory, more than $K + 1$ measurements are required in order to recover sparse signals via Basis Pursuit. Instead, one typically requires $M \geq cK$ measurements, where $c > 1$ is an *oversampling factor*. As an example, we quote a result asymptotic in N. For simplicity, we assume that

the sparsity scales linearly with N; that is, $K = SN$, where we call S the *sparsity rate*.

Theorem 2.2 [28, 38, 39] *Set $K = SN$ with $0 < S \ll 1$. Then there exists an oversampling factor $c(S) = O(\log(1/S))$, $c(S) > 1$, such that, for a K-sparse signal x in the basis Ψ, the following statements hold:*

1. *The probability of recovering x via Basis Pursuit from $(c(S) + \epsilon)K$ random projections, $\epsilon > 0$, converges to one as $N \to \infty$.*
2. *The probability of recovering x via Basis Pursuit from $(c(S) - \epsilon)K$ random projections, $\epsilon > 0$, converges to zero as $N \to \infty$.*

In an illuminating series of recent papers, Donoho and Tanner [38–40] have characterized the oversampling factor $c(S)$ precisely (see also Section 2.8.5). With appropriate oversampling, reconstruction via Basis Pursuit is also provably robust to measurement noise and quantization error [22].

In the remainder of this section and in Chapter 5, we often use the abbreviated notation c to describe the oversampling factor required in various settings even though $c(S)$ depends on the sparsity K and signal length N.

Recovery via greedy pursuit

At the expense of slightly more measurements, iterative greedy algorithms such as Orthogonal Matching Pursuit (OMP) [91], Matching Pursuit (MP) [5], and Tree Matching Pursuit (TMP) [93, 94] have also been proposed to recover the signal x from the measurements y (see Section 2.5.2). In CS applications, OMP requires $c \approx 2\ln(N)$ [91] to succeed with high probability. OMP is also guaranteed to converge within M iterations. In Chapter 5, we will exploit both Basis Pursuit and greedy algorithms for recovering jointly sparse signals from incoherent measurements. We note that Tropp and Gilbert require the OMP algorithm to succeed in the first K iterations [91]; however, in our simulations, we allow the algorithm to run up to the maximum of M possible iterations. While this introduces a potential vulnerability to noise in the measurements, our focus in Chapter 5 is on the noiseless case. The choice of an appropriate practical stopping criterion (likely somewhere between K and M iterations) is a subject of current research in the CS community.

2.8.4 Impact and applications

CS appears to be promising for a number of applications in signal acquisition and compression. Instead of sampling a K-sparse signal N times, only cK incoherent measurements suffice, where K can be orders of magnitude less than N. Therefore, a sensor can transmit far fewer measurements to a receiver, which can reconstruct the signal and then process it in any manner. Moreover, the cK measurements need

not be manipulated in any way before being transmitted, except possibly for some quantization. Finally, independent and identically distributed (i.i.d.) Gaussian or Bernoulli/Rademacher (random ± 1) vectors provide a useful *universal* basis that is incoherent with all others. Hence, when using a random basis, CS is universal in the sense that the sensor can apply the same measurement mechanism no matter what basis the signal is sparse in (and thus the coding algorithm is independent of the sparsity-inducing basis) [20, 21, 95].

These features of CS make it particularly intriguing for applications in remote sensing environments that might involve low-cost battery operated wireless sensors, which have limited computational and communication capabilities. Indeed, in many such environments one may be interested in sensing a *collection* of signals using a network of low-cost signals. In Chapter 5, we propose a series of models for joint sparsity structure among a collection of signals, and we propose the corresponding algorithms for Distributed Compressed Sensing (DCS) of such signals.

Other possible application areas of CS include imaging [33], medical imaging [22, 96], and RF environments (where high-bandwidth signals may contain low-dimensional structures such as radar chirps) [97]. As research continues into practical methods for signal recovery (see Section 2.8.3), additional work has focused on developing physical devices for acquiring random projections. Our group has developed, for example, a prototype digital CS camera based on a digital micromirror design [33]. Additional work suggests that standard components such as filters (with randomized impulse responses) could be useful in CS hardware devices [98].

2.8.5 The geometry of Compressed Sensing

It is important to note that the core theory of CS draws from a number of deep geometric arguments. For example, when viewed together, the CS encoding/decoding process can be interpreted as a linear projection $\Phi : \mathbb{R}^N \mapsto \mathbb{R}^M$ followed by a non-linear mapping $\Delta : \mathbb{R}^M \mapsto \mathbb{R}^N$. In a very general sense, one may naturally ask for a given class of signals $\mathcal{F} \in \mathbb{R}^N$ (such as the set of K-sparse signals or the set of signals with coefficients $\|\alpha\|_{\ell_p} \leq 1$), what encoder/decoder pair Φ, Δ will ensure the best reconstruction (minimax distortion) of all signals in $\mathcal{F}$. This best-case performance is proportional to what is known as the Gluskin n-width [99, 100] of $\mathcal{F}$ (in our setting $n = M$), which in turn has a geometric interpretation. Roughly speaking, the Gluskin n-width seeks the $(N-n)$-dimensional slice through $\mathcal{F}$ that yields signals of greatest energy. This n-width bounds the best-case performance of CS on classes of compressible signals, and one of the hallmarks of CS is that, given a sufficient number of measurements this optimal performance is achieved (to within a constant) [21, 78].

Additionally, one may view the ℓ_0/ℓ_1 equivalence problem geometrically. In particular, given the measurements $y = \Phi x$, we have an $(N-M)$-dimensional hyperplane $\mathcal{H}_y = \{x' \in \mathbb{R}^N : y = \Phi x'\} = \mathcal{N}(\Phi) + x$ of feasible signals that could account for the measurements y. Supposing the original signal x is K-sparse, the ℓ_1 recovery program will recover the correct solution x if and only if $\|x'\|_1 > \|x\|_1$ for every other signal

$x' \in \mathcal{H}_y$ on the hyperplane. This happens only if the hyperplane $\mathcal{H}_y$ (which passes through x) does not "cut into" the ℓ_1-ball of radius $\|x\|_1$. This ℓ_1-ball is a polytope, on which x belongs to a $(K-1)$-dimensional "face." If Φ is a random matrix with i.i.d. Gaussian entries, then the hyperplane $\mathcal{H}_y$ will have random orientation. To answer the question of how M must relate to K in order to ensure reliable recovery, it helps to observe that a randomly generated hyperplane $\mathcal{H}$ will have greater chance to slice into the ℓ_1 ball as $\dim(\mathcal{H}) = N - M$ grows (or as M shrinks) or as the dimension $K-1$ of the face on which x lives grows. Such geometric arguments have been made precise by Donoho and Tanner [38–40] and used to establish a series of sharp bounds on CS recovery.

In Section 6.1.3, we will also present an alternative proof for the first statement in Theorem 2.1 based purely on geometric arguments (following, in fact, from a result about manifold embeddings).

2.8.6 Connections with dimensionality reduction

We have also identified [95] a fundamental connection between the CS and the JL lemma. In order to make this connection, we considered the *Restricted Isometry Property* (RIP), which has been identified as a key property of the CS projection operator Φ to ensure stable signal recovery. We say Φ has RIP of order K if for every K-sparse signal x,

$$(1-\epsilon)\sqrt{\frac{M}{N}} \leq \frac{\|\Phi x\|_2}{\|x\|_2} \leq (1+\epsilon)\sqrt{\frac{M}{N}}.$$

A random $M \times N$ matrix with i.i.d. Gaussian entries can be shown to have this property with high probability if $M = O(K \log(N/K))$.

While the JL lemma concerns pairwise distances within a finite cloud of points, the RIP concerns isometric embedding of an *infinite* number of points (comprising a union of K-dimensional subspaces in $\mathbb{R}^N$). However, the RIP can in fact be derived by constructing an effective *sampling* of K-sparse signals in $\mathbb{R}^N$, using the JL lemma to ensure isometric embeddings for each of these points, and then arguing that the RIP must hold true for *all* K-sparse signals. (See [95] for the full details.)

In Chapter 6, we will again employ the JL lemma to prove that manifolds also have near-isometric embeddings under random projections to lower-dimensional space; this fact will allow us to extend the applicability of CS beyond sparse signal recovery to include parameter estimation and manifold learning from random measurements.

Chapter 3
Parametric Representation and Compression of Multi-Dimensional Piecewise Functions

In this chapter[1] we consider the task of approximating and compressing two model classes of functions for which traditional harmonic dictionaries fail to provide sparse representations. However, the model itself dictates a low-dimensional structure to the signals, which we capture using a novel parametric multiscale dictionary.

The functions we consider are both highly relevant in signal processing and highly structured. In particular, we consider piecewise constant signals in P dimensions where a smooth $(P-1)$-dimensional discontinuity separates the two constant regions, and we also consider the extension of this class to piecewise smooth signals, where a smooth $(P-1)$-dimensional discontinuity separates two smooth regions. These signal classes provide basic models, for example, for images containing edges, video sequences of moving objects, or seismic data containing geological horizons.

Despite the underlying (indeed, low-dimensional) structure in each of these classes, classical harmonic dictionaries fail to provide sparse representations for such signals. The problem comes from the $(P-1)$-dimensional discontinuity, whose smooth geometric structure is not captured by local isotropic representations such as wavelets.

As a remedy, we propose a multiscale dictionary consisting of local parametric atoms called *surflets*, each a piecewise constant function with a (tunable) polynomial discontinuity separating the two constant regions. Our surflet dictionary falls outside the traditional realm of bases and frames (where approximations are assembled as linear combinations of atoms from the dictionary). Rather, our scheme is perhaps better viewed as a "geometric tiling," where precisely one atom from the dictionary is used to describe the signal at each part of the domain (these atoms "tile" together to cover the domain). We discuss multiscale, tree-based schemes for assembling and encoding surflet representations, and we prove that such schemes attain optimal asymptotic approximation and compression performance on our piecewise constant function classes.

We also see limitations to this scheme, however. As designed for piecewise constant functions, our surflet model fails to account for relevant activity away from the discontinuity. Turning our attention, then, to the problem of approximating and compressing piecewise smooth functions, we propose a hybrid scheme combining surflets

[1]This work is in collaboration with Venkat Chandrasekaran, Dror Baron, and Richard Baraniuk [101] and also builds upon earlier work in collaboration with Justin Romberg, Hyeokho Choi, and Richard Baraniuk [102].

with wavelets. Our scheme is based on the simple yet powerful observation that geometric features can be efficiently approximated using local surflet atoms in the spatial domain, and that the projection of these geometric primitives onto wavelet subspaces can therefore approximate the corresponding wavelet coefficients — we dub the resulting projections *surfprints*. Hence we develop an entirely wavelet-domain approximation and compression scheme for piecewise smooth signals, where wavelet coefficients near edges are grouped together (as surfprints) and described parametrically. We prove that surfprint/wavelet schemes attain near-optimal asymptotic approximation and compression performance on our piecewise smooth function classes.

Our work in this chapter can be viewed as a generalization of the wedgelet [103] and wedgeprint [102] representations. (Wedgelets are 2-D atoms localized on dyadic squares with a straight edge separating two constant regions.) Our extensions in this chapter, however, provide fundamental new insights in the following directions:

- The wedgelet and wedgeprint dictionaries are restricted to 2-D signals, while our proposed representations are relevant in higher dimensions.
- Wedgelets and wedgeprints achieve optimal approximation rates only for functions that are $\mathcal{C}^2$-smooth and contain a $\mathcal{C}^2$-smooth discontinuity; our results not only show that surflets and surfprints can be used to achieve optimal rates for more general classes, but also highlight the necessary polynomial orders and quantization scheme (a nontrivial extension from wedgelets).
- We also present a more thorough analysis of discretization effects, including new insights on the multiscale behavior (not revealed by considering wedgelets alone), a new strategy for reducing the surflet dictionary size at fine scales, and the first treatment of wedgeprint/surfprint discretization.

This chapter is organized as follows. In Section 3.1, we define our function models and state the specific goals of our approximation and compression algorithms. We introduce surflets in Section 3.2. In Section 3.3, we describe our surflet-based representation schemes for piecewise constant functions. In Section 3.4, we present our novel dictionary of wavelets and surfprints for effectively representing piecewise smooth functions. Section 3.5 discusses extensions to discrete data and presents numerical experiments.

3.1 Function Classes and Performance Bounds

3.1.1 Multi-dimensional signal models

In this chapter, we consider functions over the continuous domain $[0,1]^P$. We let $\mathbf{x} = [x_1, x_2, \cdots, x_P] \in [0,1]^P$ denote an arbitrary point in this domain. (Note the use boldface characters to denote vectors in this chapter.) We denote the first $P-1$ elements of $\mathbf{x}$ by $\mathbf{y}$, i.e., $\mathbf{y} = [x_1, x_2, \cdots, x_{P-1}] \in [0,1]^{P-1}$.

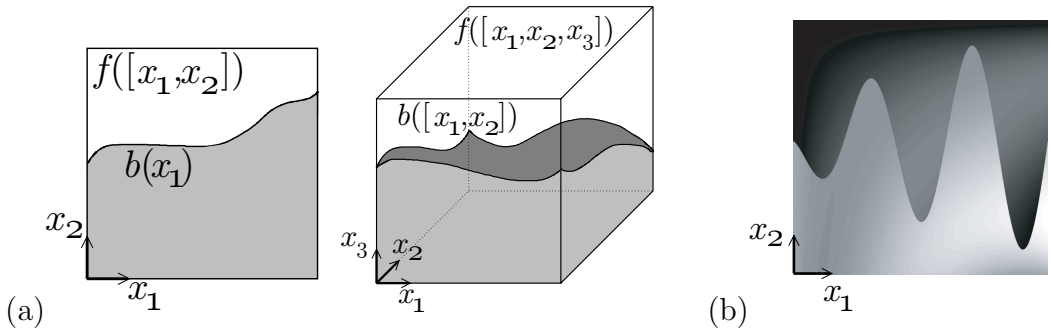

Figure 3.1: *(a) Piecewise constant ("Horizon-class") functions for dimensions $P = 2$ and $P = 3$. (b) Piecewise smooth function for dimension $P = 2$.*

We will often find it useful construct a P-dimensional function by combining two P-dimensional functions separated by a $(P-1)$-dimensional discontinuity. As an example, suppose that g_1 and g_2 are functions of P variables

$$g_1, g_2 : [0,1]^P \to \mathbb{R},$$

and that b is a function of $P-1$ variables

$$b : [0,1]^{P-1} \to \mathbb{R}.$$

We define the function $f : [0,1]^P \to \mathbb{R}$ in the following piecewise manner:

$$f(\mathbf{x}) = \begin{cases} g_1(\mathbf{x}), & x_P \geq b(\mathbf{y}) \\ g_2(\mathbf{x}), & x_P < b(\mathbf{y}). \end{cases}$$

Piecewise constant model

The first class of functions we consider is a "piecewise constant" case where $g_1 = 1$ and $g_2 = 0$. In this case, the $(P-1)$-dimensional discontinuity b defines a boundary between two constant regions in P dimensions. (Piecewise constant functions f defined in this manner are sometimes known as Horizon-class functions [103].) When $b \in \mathcal{C}^{H_\mathrm{d}}$, with $H_\mathrm{d} = r_\mathrm{d} + \nu_\mathrm{d}$, we denote the resulting space of functions f by $\mathcal{F}_\mathrm{C}(P, H_\mathrm{d})$. When $P = 2$, these functions can be interpreted as images containing a $\mathcal{C}^{H_\mathrm{d}}$-smooth one-dimensional discontinuity that separates a 0-valued region below from a 1-valued region above. For $P = 3$, functions in $\mathcal{F}_\mathrm{C}(P, H_\mathrm{d})$ can be interpreted as cubes with a 2-D $\mathcal{C}^{H_\mathrm{d}}$-smooth surface cutting through them, dividing them into two regions — 0-valued below the surface and 1-valued above it (see Figure 3.1(a) for examples in 2-D and 3-D).

We often use f^c to denote an arbitrary function in $\mathcal{F}_\mathrm{C}(P, H_\mathrm{d})$, in such cases we denote its $(P-1)$-dimensional $\mathcal{C}^{H_\mathrm{d}}$-smooth discontinuity by b^c.

Piecewise smooth model

The second class of functions we consider is a "piecewise smooth" model. For this class of functions, we let $g_1, g_2 \in \mathcal{C}^{H_\mathrm{s}}$, with $H_\mathrm{s} = r_\mathrm{s} + \nu_\mathrm{s}$, and $b \in H_\mathrm{d}$, with $H_\mathrm{d} = r_\mathrm{d} + \nu_\mathrm{d}$. The resulting piecewise smooth function f consists of a $(P-1)$-dimensional $\mathcal{C}^{H_\mathrm{d}}$-smooth discontinuity that separates two $\mathcal{C}^{H_\mathrm{s}}$-smooth regions in P dimensions (see Figure 3.1(b) for an example in 2-D). We denote the class of such piecewise smooth functions by $\mathcal{F}_\mathrm{S}(P, H_\mathrm{d}, H_\mathrm{s})$. One can check that both $\mathcal{F}_\mathrm{C}(P, H_\mathrm{d})$ and the space of P-dimensional uniformly $\mathcal{C}^{H_\mathrm{s}}$ functions are subsets of $\mathcal{F}_\mathrm{S}(P, H_\mathrm{d}, H_\mathrm{s})$.

We often use f^s to denote an arbitrary function in $\mathcal{F}_\mathrm{S}(P, H_\mathrm{d}, H_\mathrm{s})$. For such a function, we denote the $(P-1)$-dimensional $\mathcal{C}^{H_\mathrm{d}}$-smooth discontinuity by b^s and the P-dimensional $\mathcal{C}^{H_\mathrm{s}}$-smooth regions by g_1^s and g_2^s.

3.1.2 Optimal approximation and compression rates

In this chapter, we define dictionaries of atoms from which we construct an approximation $\widehat{f}$ to f, which may belong to $\mathcal{F}_\mathrm{C}(P, H_\mathrm{d})$ or $\mathcal{F}_\mathrm{S}(P, H_\mathrm{d}, H_\mathrm{s})$. We analyze the performance of our coding scheme using the squared-L_2 distortion measure between the P-dimensional functions f and $\widehat{f}$. We measure the ability of our dictionary of atoms to represent f sparsely by the asymptotic approximation performance $\|f - \widehat{f_N}\|_{L_2}^2$ as $N \to \infty$, where $\widehat{f_N}$ is the best N-term approximant to f. We also present compression algorithms that encode those atoms from the corresponding dictionaries (depending on whether $f \in \mathcal{F}_\mathrm{C}(P, H_\mathrm{d})$ or $f \in \mathcal{F}_\mathrm{S}(P, H_\mathrm{d}, H_\mathrm{s})$) used to construct $\widehat{f}$. We measure the performance of these compression algorithms by the asymptotic rate-distortion function $\|f - \widehat{f_R}\|_{L_2}^2$ $R \to \infty$, where $\widehat{f_R}$ is the best approximation to f that can be encoded using R bits [104].

A function belonging to either class $\mathcal{F}_\mathrm{C}(P, H_\mathrm{d})$ or $\mathcal{F}_\mathrm{S}(P, H_\mathrm{d}, H_\mathrm{s})$ contains a certain degree of *structure* due to the smooth functions of which it is comprised. (One of these component functions is not only smooth but has lower dimension than f.) Indeed, we see that the optimal approximation and compression performance rates derive directly from these degrees of smoothness.

In [79], Cohen et al. establish the optimal approximation rate for D-dimensional $\mathcal{C}^H$-smooth functions d:

$$\left\| d - \widehat{d_N} \right\|_{L_p}^2 \lesssim \left(\frac{1}{N} \right)^{\frac{2H}{D}}.$$

Similarly, as we discussed in Section 2.6, the optimal achievable asymptotic rate-distortion performance for D-dimensional $\mathcal{C}^H$-smooth functions d is given by

$$\left\| d - \widehat{d_R} \right\|_{L_p}^2 \lesssim \left(\frac{1}{R} \right)^{\frac{2H}{D}}.$$

These results, however, are only useful for characterizing optimal *separate* representations for the $(P-1)$-dimensional discontinuity and the P-dimensional smooth regions.

We extend these results to non-separable representations of the P-dimensional function classes $\mathcal{F}_{\mathrm{C}}(P, H_{\mathrm{d}})$ and $\mathcal{F}_{\mathrm{S}}(P, H_{\mathrm{d}}, H_{\mathrm{s}})$ in Theorems 3.1 and 3.2, respectively.

Theorem 3.1 *The optimal asymptotic approximation performance that can be obtained for all $f^{\mathrm{c}} \in \mathcal{F}_{\mathrm{C}}(P, H_{\mathrm{d}})$ is given by*

$$\left\| f^{\mathrm{c}} - \widehat{f_N^{\mathrm{c}}} \right\|_{L_2}^2 \lesssim \left(\frac{1}{N} \right)^{\frac{H_{\mathrm{d}}}{P-1}}.$$

Similarly, the optimal asymptotic compression performance that can be obtained for all $f^{\mathrm{c}} \in \mathcal{F}_{\mathrm{C}}(P, H_{\mathrm{d}})$ is given by

$$\left\| f^{\mathrm{c}} - \widehat{f_R^{\mathrm{c}}} \right\|_{L_2}^2 \lesssim \left(\frac{1}{R} \right)^{\frac{H_{\mathrm{d}}}{P-1}}.$$

Proof: See [101, Appendix A].

Implicit in the proof of the above theorem is that any scheme that is optimal for representing and compressing the P-dimensional function $f^{\mathrm{c}} \in \mathcal{F}_{\mathrm{C}}(P, H_{\mathrm{d}})$ in the squared-L_2 sense is equivalently optimal for the $(P-1)$-dimensional discontinuity in the L_1 sense. Roughly, the squared-L_2 distance between two Horizon-class functions f_1^{c} and f_2^{c} over a P-dimensional domain $\mathcal{D} = [\mathcal{D}_{\mathrm{b}}^1, \mathcal{D}_{\mathrm{e}}^1] \times \cdots \times [\mathcal{D}_{\mathrm{b}}^P, \mathcal{D}_{\mathrm{e}}^P]$ is equal to the L_1 distance over the $(P-1)$-dimensional subdomain $[\mathcal{D}_{\mathrm{b}}^1, \mathcal{D}_{\mathrm{e}}^1] \times \cdots \times [\mathcal{D}_{\mathrm{b}}^{P-1}, \mathcal{D}_{\mathrm{e}}^{P-1}]$ between the $(P-1)$-dimensional discontinuities b_1^{c} and b_2^{c} in f_1^{c} and f_2^{c} respectively.

More precisely and for future reference, for every $\mathbf{y}$ in the $(P-1)$-dimensional subdomain of $\mathcal{D}$, we define the $\mathcal{D}$-*clipping* of a $(P-1)$-dimensional function b as

$$\overline{b}(\mathbf{y}) = \begin{cases} b(\mathbf{y}), & \mathcal{D}_{\mathrm{b}}^P \le b(\mathbf{y}) \le \mathcal{D}_{\mathrm{e}}^P \\ \mathcal{D}_{\mathrm{e}}^P, & b(\mathbf{y}) > \mathcal{D}_{\mathrm{e}}^P \\ \mathcal{D}_{\mathrm{b}}^P, & b(\mathbf{y}) < \mathcal{D}_{\mathrm{b}}^P. \end{cases}$$

The $\mathcal{D}$-*active region* of b is defined to be

$$\left\{ \mathbf{y} \in [\mathcal{D}_{\mathrm{b}}^1, \mathcal{D}_{\mathrm{e}}^1] \times \cdots \times [\mathcal{D}_{\mathrm{b}}^{P-1}, \mathcal{D}_{\mathrm{e}}^{P-1}] : b(\mathbf{y}) \in [\mathcal{D}_{\mathrm{b}}^P, \mathcal{D}_{\mathrm{e}}^P] \right\},$$

that subset of the subdomain of $\mathcal{D}$ for which the range of b lies in $[\mathcal{D}_{\mathrm{b}}^P, \mathcal{D}_{\mathrm{e}}^P]$. The $\mathcal{D}$-*clipped* L_1 *distance* between b_1^{c} and b_2^{c} is then defined as

$$\overline{L_1}(b_1^{\mathrm{c}}, b_2^{\mathrm{c}}) = \left\| \overline{b_1^{\mathrm{c}}} - \overline{b_2^{\mathrm{c}}} \right\|_{L_1([\mathcal{D}_{\mathrm{b}}^1, \mathcal{D}_{\mathrm{e}}^1] \times \cdots \times [\mathcal{D}_{\mathrm{b}}^{P-1}, \mathcal{D}_{\mathrm{e}}^{P-1}])}.$$

One can check that $\| f_1^{\mathrm{c}} - f_2^{\mathrm{c}} \|_{L_2(\mathcal{D})}^2 = \overline{L_1}(b_1^{\mathrm{c}}, b_2^{\mathrm{c}})$ for any $\mathcal{D}$.

The following theorem characterizes the optimal achievable asymptotic approximation rate and rate-distortion performance for approximating and encoding elements of the function class $\mathcal{F}_{\mathrm{S}}(P, H_{\mathrm{d}}, H_{\mathrm{s}})$.

Theorem 3.2 *The optimal asymptotic approximation performance that can be obtained for all* $f^{\mathrm{s}} \in \mathcal{F}_{\mathrm{S}}(P, H_{\mathrm{d}}, H_{\mathrm{s}})$ *is given by*

$$\left\| f^{\mathrm{s}} - \widehat{f_N^{\mathrm{s}}} \right\|_{L_2}^2 \lesssim \left(\frac{1}{N} \right)^{\min\left(\frac{H_{\mathrm{d}}}{P-1}, \frac{2H_{\mathrm{s}}}{P} \right)}.$$

Similarly, the optimal asymptotic compression performance that can be obtained for all $f^{\mathrm{s}} \in \mathcal{F}_{\mathrm{S}}(P, H_{\mathrm{d}}, H_{\mathrm{s}})$ *is given by*

$$\left\| f^{\mathrm{s}} - \widehat{f_R^{\mathrm{s}}} \right\|_{L_2}^2 \lesssim \left(\frac{1}{R} \right)^{\min\left(\frac{H_{\mathrm{d}}}{P-1}, \frac{2H_{\mathrm{s}}}{P} \right)}.$$

Proof: See [101, Appendix B].

3.1.3 "Oracle" coders and their limitations

In order to approximate or compress an arbitrary function $f^{\mathrm{c}} \in \mathcal{F}_{\mathrm{C}}(P, H_{\mathrm{d}})$, we presume that an algorithm is given the function f^{c} itself. Again, however, all of the critical information about f^{c} is contained in the discontinuity b^{c}, and one would expect any efficient coder to exploit such a fact. Methods through which this is achieved may vary.

One can imagine a coder that *explicitly* encodes an approximation $\widehat{b^{\mathrm{c}}}$ to b^{c} and then constructs a Horizon approximation $\widehat{f^{\mathrm{c}}}$. Knowledge of b^{c} could be provided from an external "oracle" [105], or b^{c} could conceivably be estimated from the provided data f^{c}. As discussed in Section 2.6.2, a tree-structured wavelet coder could provide one efficient method for compressing the $(P-1)$-dimensional smooth function b^{c} with optimal L_1 rate-distortion performance. Such a wavelet/Horizon coder would then be optimal (in the squared-L_2 sense) for coding instances of f^{c} at the optimal rate of Theorem 3.1. In practice, however, a coder would not provided with explicit information of b^{c}, and a method for estimating b^{c} from f^{c} may be difficult to implement. Estimates for b^{c} may also be quite sensitive to noise in the data.

A similar strategy could also be employed for $f^{\mathrm{s}} \in \mathcal{F}_{\mathrm{S}}(P, H_{\mathrm{d}}, H_{\mathrm{s}})$. Approximations to the discontinuity $\widehat{b^{\mathrm{s}}}$ and the P-dimensional smooth regions $\widehat{g_1^{\mathrm{s}}}$ and $\widehat{g_2^{\mathrm{s}}}$ may be encoded separately and explicitly. This strategy would have disadvantages for the same reasons mentioned above. In fact, estimating the discontinuity in this scenario would be much harder.

In this chapter, we seek and propose representation schemes and algorithms that approximate f^{c} and f^{s} directly in P dimensions. For our surflet and surfprint schemes, we emphasize that *no explicit knowledge of the functions* b^{c}, b^{s}, g_1^{s}, *or* g_2^{s} *is required.* We prove that surflet-based approximation techniques and encoding algorithms for f^{c} achieve the optimal decay rates, while our surfprint-based methods for f^{s} achieve the optimal approximation decay rate and a near-optimal rate-distortion decay rate (within a logarithmic factor of the optimal decay rate of Theorem 3.2). Although

we omit the discussion, our algorithms can be extended to similar piecewise constant and piecewise smooth function spaces. Our spatially localized approach, for example, allows for changes in the variable along which the discontinuity varies (assumed throughout this chapter to be x_P as described in Section 3.1.1).

3.2 The Surflet Dictionary

In this section, we introduce a discrete, multiscale dictionary of P-dimensional atoms called *surflets* that can be used to construct approximations to a function $f^{\mathrm{c}} \in \mathcal{F}_{\mathrm{C}}(P, H_{\mathrm{d}})$. A surflet is a piecewise constant function defined on a P-dimensional dyadic hypercube, where a $(P-1)$-dimensional polynomial specifies the discontinuity. Section 3.3 describes compression using surflets.

3.2.1 Motivation — Taylor's theorem

The surflet atoms are motivated by the following property. If d is a function of D variables in $\mathcal{C}^H$ with $H = r + \nu$, r is a positive integer, and $\nu \in (0, 1]$, then Taylor's theorem states that

$$\begin{aligned} d(\mathbf{z}+\mathbf{h}) &= d(\mathbf{z}) + \frac{1}{1!}\sum_{i_1=1}^{D} d_{z_{i_1}}(\mathbf{z})h_{i_1} + \frac{1}{2!}\sum_{i_1,i_2=1}^{D} d_{z_{i_1},z_{i_2}}(\mathbf{z})h_{i_1}h_{i_2} + \cdots \\ &\quad + \frac{1}{r!}\sum_{i_1,\ldots,i_r=1}^{D} d_{z_{i_1},\cdots,z_{i_r}}(\mathbf{z})h_{i_1}\cdots h_{i_r} + O(\|\mathbf{h}\|^H), \end{aligned} \tag{3.1}$$

where $d_{z_1,\cdots,z_\ell}$ refers to the iterated partial derivatives of d with respect to $z_1, \ldots, z_\ell$ in that order. (Note that there are D^ℓ ℓ'th order derivative terms.) Thus, over a small domain, the function d is well approximated using a polynomial of order r (where the polynomial coefficients correspond to the partial derivatives of d evaluated at $\mathbf{z}$).

Clearly, in the case of f^{c}, one method for approximating the discontinuity b^{c} would be to assemble a *piecewise polynomial* approximation, where each polynomial is derived from the local Taylor approximation of b^{c} (let $D = P - 1$, $H = H_{\mathrm{d}}$, and $d = b^{\mathrm{c}}$ in the above characterization). These piecewise polynomials can be used to assemble a Horizon-class approximation of the function f^{c}. Surflets provide the P-dimensional framework for constructing such approximations and can be implemented without explicit knowledge of b^{c} or its derivatives.

3.2.2 Definition

Recall from Section 2.1.5 that a *dyadic hypercube* $X_j \subseteq [0,1]^P$ at scale $j \in \mathbb{N}$ is a domain that satisfies[2]

$$X_j = [\beta_1 2^{-j}, (\beta_1+1)2^{-j}) \times \cdots \times [\beta_P 2^{-j}, (\beta_P+1)2^{-j})$$

with $\beta_1, \beta_2, \ldots, \beta_P \in \{0, 1, \ldots, 2^j - 1\}$. We explicitly denote the $(P-1)$-dimensional hypercube *subdomain* of X_j as

$$Y_j = [\beta_1 2^{-j}, (\beta_1+1)2^{-j}) \times \cdots \times [\beta_{P-1} 2^{-j}, (\beta_{P-1}+1)2^{-j}). \tag{3.2}$$

The *surflet* $s(X_j; p; \cdot)$ is a Horizon-class function over the dyadic hypercube X_j defined through the $(P-1)$-dimensional polynomial p. For $\mathbf{x} \in X_j$ with corresponding $\mathbf{y} = [x_1, x_2, \cdots, x_{P-1}]$,

$$s(X_j; p; \mathbf{x}) = \begin{cases} 1, & x_P \geq p(\mathbf{y}) \\ 0, & \text{otherwise,} \end{cases}$$

where the polynomial $p(\mathbf{y})$ is defined as

$$p(\mathbf{y}) = p_0 + \sum_{i_1=1}^{P-1} p_{1,i_1}\, y_{i_1} + \sum_{i_1,i_2=1}^{P-1} p_{2,i_1,i_2}\, y_{i_1} y_{i_2} + \cdots + \sum_{i_1,\ldots,i_{r_\mathrm{d}}=1}^{P-1} p_{r_\mathrm{d},i_1,i_2,\ldots,i_{r_\mathrm{d}}}\, y_{i_1} y_{i_2} \cdots y_{i_{r_\mathrm{d}}}.$$

We call the polynomial coefficients $\{p_{\ell,i_1,\ldots,i_\ell}\}_{\ell=0}^{r_\mathrm{d}}$ the *surflet coefficients*.[3] We note here that, in some cases, a surflet may be identically 0 or 1 over the entire domain X_j. We sometimes denote a generic surflet by $s(X_j)$, indicating only its region of support.

A surflet $s(X_j)$ approximates the function f^c over the dyadic hypercube X_j. One can cover the entire domain $[0,1]^P$ with a collection of dyadic hypercubes (possibly at different scales) and use surflets to approximate f^c over each of these smaller domains. For $P = 3$, these surflets tiled together look like piecewise polynomial "surfaces" approximating the discontinuity b^c in the function f^c. Figure 3.2 illustrates a collection of surflets with $P = 2$ and $P = 3$.

[2] In this chapter we use half-open intervals, but in order to cover the entire domain $[0,1]^P$, in the case where $(\beta_i+1)2^{-j} = 1$, $i \in \{1, \ldots, P\}$, we replace the half-open interval $[\beta_i 2^{-j}, (\beta_i+1)2^{-j})$ with the closed interval $[\beta_i 2^{-j}, (\beta_i+1)2^{-j}]$.

[3] Because the ordering of the terms $y_{i_1} y_{i_2} \cdots y_{i_\ell}$ in a monomial is irrelevant, only $\binom{\ell+P-2}{\ell}$ monomial coefficients (not $(P-1)^\ell$) need to be encoded for order ℓ. We preserve the slightly redundant notation for ease of comparison with (3.1).

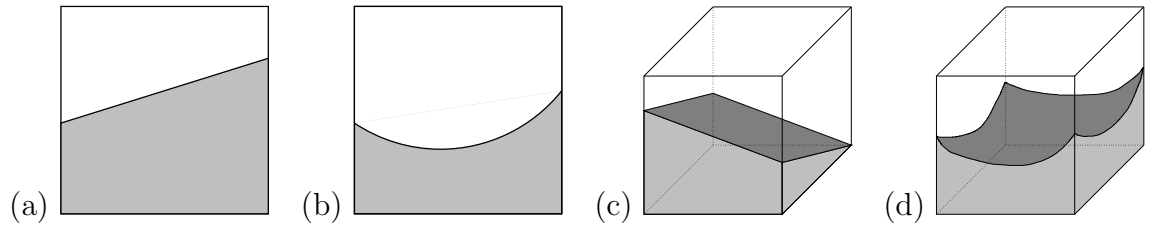

Figure 3.2: *Example surflets, designed for (a) $P = 2$, $H_{\mathrm{d}} \in (1, 2]$; (b) $P = 2$, $H_{\mathrm{d}} \in (2, 3]$; (c) $P = 3$, $H_{\mathrm{d}} \in (1, 2]$; (d) $P = 3$, $H_{\mathrm{d}} \in (2, 3]$.*

3.2.3 Quantization

We obtain a discrete surflet dictionary $\mathcal{S}(j)$ at scale j by quantizing the set of allowable surflet polynomial coefficients. For $\ell \in \{0, 1, \ldots, r_{\mathrm{d}}\}$, the surflet coefficient $p_{\ell,i_1,\ldots,i_\ell}$ at scale $j \in \mathbb{N}$ is restricted to values $\{\mu \cdot \Delta_{\ell,j}^{H_{\mathrm{d}}}\}_{\mu \in \mathbb{Z}}$, where the stepsize satisfies

$$\Delta_{\ell,j}^{H_{\mathrm{d}}} = 2^{-(H_{\mathrm{d}}-\ell)j}. \tag{3.3}$$

The necessary range for μ will depend on the derivative bound Ω (Section 2.1.4). We emphasize that the relevant discrete surflet dictionary $\mathcal{S}(j)$ is *finite* at every scale j.

These quantization stepsizes are carefully chosen to ensure the proper fidelity of surflet approximations without requiring excess bitrate. The key idea is that *higher-order terms can be quantized with lesser precision* without increasing the residual error term in the Taylor approximation (3.1). In fact, Kolmogorov and Tihomirov [57] implicitly used this concept to establish the metric entropy for bounded uniformly smooth functions.

3.3 Approximation and Compression of Piecewise Constant Functions

3.3.1 Overview

We now propose a surflet-based multiresolution geometric tiling approach to approximate and encode an arbitrary function $f^{\mathrm{c}} \in \mathcal{F}_{\mathrm{C}}(P, H_{\mathrm{d}})$. The tiling is arranged on a 2^P-tree, where each node in the tree at scale j corresponds to a hypercube of sidelength 2^{-j}. Each node is labeled with a surflet appropriately chosen from $\mathcal{S}(j)$ and is either a leaf node (hypercube) or has 2^P children nodes (children hypercubes that perfectly tile the volume of the parent hypercube). Leaf nodes provide the actual approximation to the function f^{c}, while interior nodes are useful for predicting and encoding their descendants. This framework enables an *adaptive, multiscale* approximation of f^{c} — many small surflets can be used at fine scales for complicated regions, while few large surflets will suffice to encode simple regions of f^{c} (such as those containing all 0 or 1). Figure 3.3 shows surflet tiling approximations for $P = 2$ and $P = 3$.

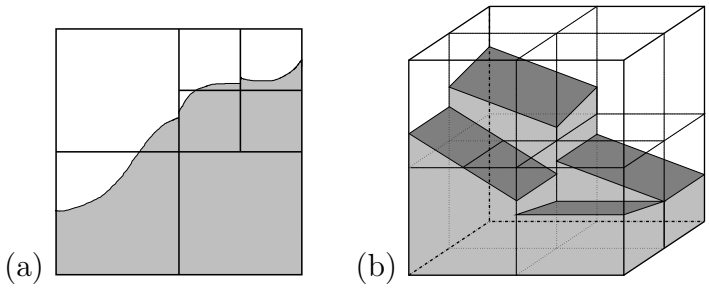

Figure 3.3: *Example surflet tilings, (a) piecewise cubic with $P = 2$ and (b) piecewise linear with $P = 3$.*

Section 3.3.2 discusses techniques for determining the proper surflet at each node. Section 3.3.3 describes a constructive algorithm for building tree-based surflet approximations. Section 3.3.4 describes the performance of a simple surflet encoder acting only on the leaf nodes. Section 3.3.5 presents a more advanced surflet coder, using a top-down predictive technique to exploit the correlation among surflet coefficients. Finally, Section 3.3.6 discusses extensions of our surflet-based representation schemes to broader function classes.

3.3.2 Surflet selection

Consider a node at scale j that corresponds to a dyadic hypercube X_j, and let Y_j be the $(P-1)$-dimensional subdomain of X_j as defined in (3.2).

We first examine a situation where the coder is provided with explicit information about the discontinuity b^{c} and its derivatives. In this case, determination of the surflet at the node that corresponds to X_j can proceed as implied by Section 3.2. The coder constructs the Taylor expansion of b^{c} around any point $\mathbf{y} \in Y_j$ and quantizes the polynomial coefficients (3.3). We choose

$$\mathbf{y}_{ep} = \left[\left(\beta_1 + \frac{1}{2} \right) 2^{-j}, \left(\beta_2 + \frac{1}{2} \right) 2^{-j}, \ldots, \left(\beta_{P-1} + \frac{1}{2} \right) 2^{-j} \right]$$

and call this an *expansion point*. We refer to the resulting surflet as the *quantized Taylor surflet*. From (3.1), it follows that the squared-L_2 error between f^{c} and the quantized Taylor surflet approximation $s(X_j)$ (which equals the X_j-clipped L_1 error between b^{c} and the polynomial defining $s(X_j)$) obeys

$$\|f^{\mathrm{c}} - s(X_j)\|^2_{L_2(X_j)} = \int_{X_j} (f^{\mathrm{c}} - s(X_j))^2 = O\left(2^{-j(H_{\mathrm{d}}+P-1)}\right). \tag{3.4}$$

However, as discussed in Section 3.1.3, our coder is not provided with explicit information about b^{c}. Therefore, approximating functions in $\mathcal{F}_{\mathrm{C}}(P, H_{\mathrm{d}})$ using Taylor

surflets is impractical.[4]

We now define a technique for obtaining a surflet estimate directly from the function f^c. We assume that there exists a method to compute the squared-L_2 error $\|f^c - s(X_j)\|^2_{L_2(X_j)}$ between a given surflet $s(X_j)$ and the function f^c on the dyadic block X_j. In such a case, we can search the finite surflet dictionary $\mathcal{S}(j)$ for the minimizer of this error *without explicit knowledge of* b^c. We refer to the resulting surflet as the *native* L_2*-best surflet*; this surflet will necessarily obey (3.4) as well. Practically speaking, there may be certain challenges to solving this L_2 optimization problem. These challenges are revealed by taking a geometric perspective, viewing the parameter estimation problem as the orthogonal projection from f^c onto the manifold of possible surflets. As we discuss in Chapter 4, this manifold is not differentiable, which poses an apparent barrier to techniques that might invoke the manifold's tangent spaces in order to apply calculus-based optimization. However, in Chapter 4, we also introduce a multiscale estimation algorithm designed to circumvent this difficulty.

Section 3.3.4 discusses the coding implications of using L_2-best surflets from $\mathcal{S}(j)$. Using native L_2-best surflets over dyadic blocks X_j achieves near-optimal performance. As will be made apparent in Section 3.3.5, in order to achieve optimal performance, a coder must exploit correlations among nearby surflets. Unfortunately, these correlations may be difficult to exploit using native L_2-best surflets. The problem arises because surflets with small X_j-active regions (Section 3.1.2) may be close in L_2 distance over X_j yet have vastly different underlying polynomial coefficients. (These coefficients are used explicitly in our encoding strategy.)

To resolve this problem, we suggest computing L_2-best surflet fits to f^c over the L*-extension* of each dyadic hypercube X_j. That is, if $X_j = [\beta_1 2^{-j}, (\beta_1+1)2^{-j}) \times \cdots \times [\beta_P 2^{-j}, (\beta_P+1)2^{-j})$ then the L-extension of X_j is defined to be

$$X_j^L = [(\beta_1 - L)2^{-j}, (\beta_1 + 1 + L)2^{-j}) \times \cdots \times [(\beta_P - L)2^{-j}, (\beta_P + 1 + L)2^{-j}),$$

where $L > 0$ is an extension factor (designed to expand the domain of analysis and increase correlations between scales).[5] An L*-extended surflet* is a surflet from $\mathcal{S}(j)$ that is now defined over X_j^L whose polynomial discontinuity has a non-empty X_j-active region. We define the L*-extended surflet dictionary* $\mathcal{S}_L(j)$ to be the set of L-extended surflets from $\mathcal{S}(j)$ plus the all-zero and all-one surflets $s(X_j) = 0$ and $s(X_j) = 1$. An L*-extended* L_2*-best surflet* fit to f^c over X_j is then defined to be the L_2-best surflet to f^c *over* X_j^L chosen from $\mathcal{S}_L(j)$. Note that even though extended surflets are defined over extended domains X_j^L, they are used to approximate the function only over the associated native domains X_j. Such extended surflet fits (over extended domains) provide sufficient mathematical constraints for a coder to relate nearby surflets, since extended surflets that are close in terms of squared-L_2 distance

[4] We refer the reader to a technical report [106] for a thorough treatment of Taylor surflet-based approximation of piecewise constant multi-dimensional functions.

[5] If necessary, each L-extension is truncated to the hypercube $[0,1]^P$.

over X_j^L have similar polynomial coefficients (even if extended surflets have small X_j-active regions, they have large X_j^L-active regions). In Section 3.3.5, we describe a coder that uses extended surflets from $\mathcal{S}_L(j)$ to achieve optimal performance.

3.3.3 Tree-based surflet approximations

The surflet dictionary consists of P-dimensional atoms at various scales. Thus, a 2^P-tree offers a natural topology for arranging the surflets used in an approximation. Specifically, each node at scale j in a 2^P-tree is labeled by a surflet that approximates the corresponding dyadic hypercube region X_j of the function f^{c}. This surflet can be assigned according to any of the procedures outlined in Section 3.3.2.

Given a method for assigning a surflet to each tree node, it is also necessary to determine the proper dyadic segmentation for the tree approximation. This can be accomplished using the CART algorithm, which is based on dynamic programming, in a process known as *tree-pruning* [103,107]. Tree-pruning proceeds from the bottom up, determining whether to prune the tree beneath each node (causing it to become a leaf node). Various criteria exist for making such a decision. In particular, the approximation-theoretic optimal segmentation can be obtained by minimizing the Lagrangian cost $D + \lambda N$ for a penalty term λ. Similarly, the Lagrangian rate-distortion cost $D + \lambda R$ can be used to obtain the optimal rate-distortion segmentation.

We summarize the construction of a surflet-based approximation as follows:

Surflet-based approximation

- **Choose scale:** Choose a maximal scale $J \in \mathbb{Z}$ for the 2^P-tree.
- **Label all nodes:** For each scale $j = 0, 1, \ldots, J$, label all nodes at scale j with either a native or an extended L_2-best surflet chosen appropriately from either discrete dictionary of surflets $\mathcal{S}(j)$ or $\mathcal{S}_L(j)$.
- **Prune tree:** Starting at the second-finest scale $j = J - 1$, determine whether each node at scale j should be pruned (according to an appropriate pruning rule). Then proceed up to the root of the tree, i.e., until $j = 0$.

The approximation performance of this algorithm is described in the following theorem.

Theorem 3.3 *Using either quantized Taylor surflets or L_2-best surflets (extended or native), a surflet tree-pruned approximation of an element $f^{\mathrm{c}} \in \mathcal{F}_{\mathrm{C}}(P, H_{\mathrm{d}})$ achieves the optimal asymptotic approximation rate of Theorem 3.1:*

$$\left\| f^{\mathrm{c}} - \widehat{f_N^{\mathrm{c}}} \right\|_{L_2}^2 \lesssim \left(\frac{1}{N} \right)^{\frac{H_{\mathrm{d}}}{P-1}}.$$

Proof: See [101, Appendix C].

3.3.4 Leaf encoding

An initial approach toward surflet encoding would involve specification of the tree segmentation map (which denotes the location of the leaf nodes) along with the quantized surflet coefficients at each leaf node. Rate-distortion analysis then yields the following result.

Theorem 3.4 *Using either quantized Taylor surflets or L_2-best surflets (extended or native), a surflet leaf-encoder applied to an element $f^{\mathrm{c}} \in \mathcal{F}_{\mathrm{C}}(P, H_{\mathrm{d}})$ achieves the following rate-distortion performance*

$$\left\| f^{\mathrm{c}} - \widehat{f^{\mathrm{c}}_R} \right\|_{L_2}^2 \lesssim \left(\frac{\log R}{R} \right)^{\frac{H_{\mathrm{d}}}{P-1}}.$$

Proof: See [101, Appendix D].

Comparing with Theorem 3.1, this simple coder is *near-optimal* in terms of rate-distortion performance. The logarithmic factor is due to the fact that it requires $O(j)$ bits to encode each surflet at scale j. In Section 3.3.5, we propose an alternative coder that requires only a constant number of bits to encode each surflet.

3.3.5 Top-down predictive encoding

Achieving the optimal performance of Theorem 3.1 requires a more sophisticated coder that can exploit the correlation among nearby surflets. We now briefly describe a top-down surflet coder that predicts surflet parameters from previously encoded values.

Top-down predictive surflet coder

- **Encode root node:** Encode the best surflet fit $s([0,1]^P)$ to the hypercube $[0,1]^P$. Encode a flag (1-bit) specifying whether this node is interior or a leaf. Set $j \leftarrow 0$.
- **Predict surflets from parent scale:** For every interior node/hypercube X_j at scale j, partition its domain into 2^P children hypercubes at scale $j+1$. Compute the polynomial coefficients on each child hypercube X_{j+1} that agree with the encoded parent surflet $s(X_j^L)$. These serve as "predictions" for the polynomial coefficients at the child.
- **Encode innovations at child nodes:** For each predicted polynomial coefficient, encode the discrepancy with the L-extended surflet fit $s(X_{j+1}^L)$.
- **Descend tree:** Set $j \leftarrow j+1$ and repeat until no interior nodes remain.

This top-down predictive coder encodes an entire tree segmentation starting with the root node, and proceeding from the top down. Given an L-extended surflet $s(X_j^L)$ at an interior node at scale j, we show in [101, Appendix E] that the number of possible L-extended surflets from $\mathcal{S}_L(j)$ that can be used for approximation at scale $j+1$ is *constant*, independent of the scale j. Thus, given a best-fit surflet at scale 0, a constant number of bits is required to encode each surflet at subsequent scales. This prediction is possible because L-extended surflets are defined over L-extended domains, which ensures coherency between the surflet fits (and polynomial coefficients) at a parent and child node.

We note that predicting L-extended best-fit surflets to dyadic hypercube regions around the borders of $[0,1]^P$ may not be possible with a constant number of bits when the discontinuity is not completely contained within the dyadic hypercube. However, we make the mild simplifying assumption that the intersections of the discontinuity with the hyperplanes $x_P = 0$ or $x_P = 1$ can be contained within $O(2^{(P-2)j})$ hypercubes at each scale j. Therefore, using $O(H_{\mathrm{d}} j)$ bits to encode such "border" dyadic hypercubes (with the discontinuity intersecting $x_P = 0$ or $x_P = 1$) does not affect the asymptotic rate-distortion performance of the top-down predictive coder.

Theorem 3.5 *The top-down predictive coder applied to an element $f^{\mathrm{c}} \in \mathcal{F}_{\mathrm{C}}(P, H_{\mathrm{d}})$ using L-extended L_2-best surflets from $\mathcal{S}_L(j)$ achieves the optimal rate-distortion performance of Theorem 3.1:*

$$\left\| f^{\mathrm{c}} - \widehat{f_R^{\mathrm{c}}} \right\|_{L_2}^2 \lesssim \left(\frac{1}{R} \right)^{\frac{H_{\mathrm{d}}}{P-1}}.$$

Proof: See [101, Appendix E].

Although only the leaf nodes provide the ultimate approximation to the function, the additional information encoded at interior nodes provides the key to efficiently encoding the leaf nodes. In addition, unlike the surflet leaf-encoder of Section 3.3.3, this top-down approach yields a *progressive* bitstream — the early bits encode a low-resolution (coarse scale) approximation, which is then refined using subsequent bits.

3.3.6 Extensions to broader function classes

Our results for classes of functions that contain a single discontinuity can be extended to spaces of signals that contain multiple discontinuities. Functions containing multiple discontinuities that do not intersect can be represented using the surflet-based approximation scheme described in Section 3.3.3 at the optimal asymptotic approximation rate. This is because at a sufficiently high scale, dyadic hypercubes that tile signals containing multiple non-intersecting discontinuities contain at most one discontinuity.

Analysis of the surflet-based approximation scheme of Section 3.3.3 applied to signals containing intersecting discontinuities is more involved. Let $f_\sharp^{\mathrm{c}}$ be a P-dimensional piecewise constant function containing two $(P-1)$-dimensional $\mathcal{C}^{H_{\mathrm{d}}}$-smooth discontinuities that intersect each other (the analysis that follows can easily be extended to allow for more than two intersecting discontinuities). Note that the intersection of $(P-1)$-dimensional functions forms a $(P-2)$-dimensional manifold. Again, we make the mild simplifying assumption that the intersection of the discontinuities can be contained in $O(2^{(P-2)j})$ hypercubes at each scale j. The following theorem describes the approximation performance achieved by the scheme in Section 3.3.3 applied to $f_\sharp^{\mathrm{c}}$. A consequence of this theorem is that there exists a smoothness threshold $H_{\mathrm{d}}^{\mathrm{th}}$ that defines the boundary between optimal and sub-optimal approximation performance.

Theorem 3.6 *Using either quantized Taylor surflets or L_2-best surflets (extended or native), the approximation scheme of Section 3.3.3 applied to a piecewise constant P-dimensional function $f_\sharp^{\mathrm{c}}$ that contains two intersecting $\mathcal{C}^{H_{\mathrm{d}}}$-smooth $(P-1)$-dimensional discontinuities achieves performance given by:*

- *$P > 2, H_{\mathrm{d}} \leq \frac{2(P-1)}{P-2}$:*
$$\left\| f_\sharp^{\mathrm{c}} - \widehat{f_{\sharp,N}^{\mathrm{c}}} \right\|_{L_2}^2 \lesssim \left(\frac{1}{N} \right)^{\frac{H_{\mathrm{d}}}{P-1}}.$$
- *$P > 2, H_{\mathrm{d}} > \frac{2(P-1)}{P-2}$:*
$$\left\| f_\sharp^{\mathrm{c}} - \widehat{f_{\sharp,N}^{\mathrm{c}}} \right\|_{L_2}^2 \lesssim \left(\frac{1}{N} \right)^{\frac{2}{P-2}}.$$
- *$P = 2$, any H_{d}:*
$$\left\| f_\sharp^{\mathrm{c}} - \widehat{f_{\sharp,N}^{\mathrm{c}}} \right\|_{L_2}^2 \lesssim \left(\frac{1}{N} \right)^{\frac{H_{\mathrm{d}}}{P-1}}.$$

Proof: See [101, Appendix F].

Thus, the representation scheme in Section 3.3.3 achieves optimal approximation performance for $P = 2$ even in the presence of intersecting discontinuities, while it achieves optimal performance for $P > 2$ up to a smoothness threshold of $H_{\mathrm{d}}^{\mathrm{th}} = \frac{2(P-1)}{P-2}$ (for $H_{\mathrm{d}} > H_{\mathrm{d}}^{\mathrm{th}}$, the scheme performs sub-optimally: $\| f_\sharp^{\mathrm{c}} - \widehat{f_{\sharp,N}^{\mathrm{c}}} \|_{L_2}^2 \lesssim \left(\frac{1}{N}\right)^{\frac{H_{\mathrm{d}}^{\mathrm{th}}}{P-1}}$). This performance of the approximation scheme for $P > 2$ is still superior to that of wavelets, which have $H_{\mathrm{d}}^{\mathrm{th,wl}} = 1$. The reason for this difference in performance between the cases $P = 2$ and $P > 2$ is that intersections of discontinuities when

$P = 2$ correspond to points,[6] while intersections in higher dimensions correspond to low-dimensional manifolds. Hence, the number of hypercubes that contain intersections in the two-dimensional case is constant with scale, whereas the number of hypercubes that contain the intersections when $P > 2$ grows exponentially with scale. The analysis above can clearly be extended to prove analogous results for functions containing piecewise $\mathcal{C}^{H_{\mathrm{d}}}$-smooth discontinuities.

Future work will focus on improving the threshold $H_{\mathrm{d}}^{\mathrm{th}}$ for the case $P > 2$. In order to achieve optimal performance for $P > 2$, one may need a dictionary containing regular surflets and specially-designed "intersection" surflets that are specifically tailored for intersections.

3.4 Approximation and Compression of Piecewise Smooth Functions

In this section, we extend our coding strategies for piecewise constant functions to encoding an arbitrary element f^{s} from the class $\mathcal{F}_{\mathrm{S}}(P, H_{\mathrm{d}}, H_{\mathrm{s}})$ of piecewise smooth functions.

3.4.1 Motivation

For a $\mathcal{C}^{H_{\mathrm{s}}}$-smooth function f in P dimensions, a wavelet basis with sufficient vanishing moments provides approximations at the optimal rate [79] — $\|f - \widehat{f}_N\|_{L_2}^2 \lesssim \left(\frac{1}{N}\right)^{\frac{2H_{\mathrm{s}}}{P}}$ (see also Section 2.5.2). Even if one introduces a finite number of point singularities into the P-dimensional $\mathcal{C}^{H_{\mathrm{s}}}$-smooth function, wavelet-based approximation schemes still attain the optimal rate. Wavelets succeed in approximating smooth functions because most of the wavelet coefficients have small magnitudes and can thus be neglected. Moreover, an arrangement of wavelet coefficients on the nodes of a tree leads to an interesting consequence: wavelet coefficients used in the approximation of P-dimensional smooth functions are *coherent* — often, if a wavelet coefficient has small magnitude, then its children coefficients also have small magnitude. These properties of the wavelet basis have been exploited in state-of-the-art wavelet-based image coders [9, 11].

Although wavelets approximate smooth functions well, the wavelet basis is not well-equipped to approximate functions containing higher-dimensional manifold discontinuities. As discussed in Section 2.5.2, wavelets also do not take advantage of any structure (such as smoothness) that the $(P-1)$-dimensional discontinuity might have, and therefore many high-magnitude coefficients are often required to represent discontinuities. Regardless of the smoothness order of the discontinuity, the approximation rate achieved by wavelets remains the same.

[6] Our analysis also applies to "T-junctions" in images, where one edge terminates at its intersection with another.

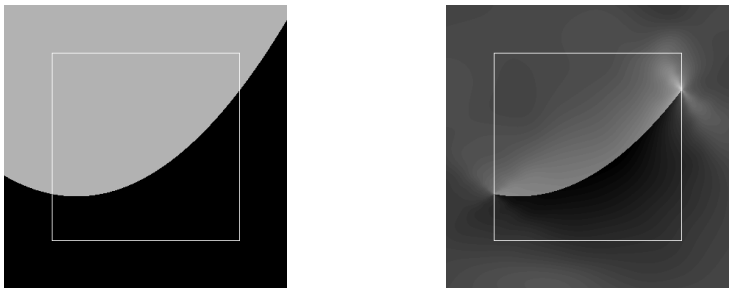

Figure 3.4: *Example surflet and the corresponding surfprint. The white box is the dyadic hypercube in which we define the surflet; note that the removal of coarse scale and neighboring wavelets causes the surfprint to appear different from the surflet.*

Despite this drawback, we desire a wavelet domain solution to approximate $f^{\mathrm{s}} \in \mathcal{F}_{\mathrm{S}}(P, H_{\mathrm{d}}, H_{\mathrm{s}})$ because most of the function f^{s} is smooth in P dimensions, except for a $(P-1)$-dimensional discontinuity. In order to solve the problem posed by the discontinuity, we propose the addition of *surfprint* atoms to the dictionary of wavelet atoms. A surfprint is a weighted sum of wavelet basis functions derived from the projection of a piecewise polynomial surflet atom (a $(P-1)$-dimensional polynomial discontinuity separating two P-dimensional polynomial regions) onto a subspace in the wavelet domain (see Figure 3.4 for an example in 2-D). Surfprints possess all the properties that make surflets well-suited to represent discontinuities. In addition, surfprints coherently model wavelet coefficients that correspond to discontinuities. Thus, we obtain a single unified wavelet-domain framework that is well-equipped to sparsely represent both discontinuities and smooth regions.

The rest of this section is devoted to the definition of surfprints and their use in a wavelet domain framework to represent and encode approximations to elements of $\mathcal{F}_{\mathrm{S}}(P, H_{\mathrm{d}}, H_{\mathrm{s}})$. We do not discuss the extension of our results to classes of piecewise smooth signals containing multiple intersecting discontinuities, but note that such an analysis would be similar to that described in Section 3.3.6.

3.4.2 Surfprints

Let X_{J_o} be a dyadic hypercube at scale J_o. Let v_1, v_2 be P-dimensional polynomials of degree $r_{\mathrm{s}}^{\mathrm{sp}}$, and let v be a P-dimensional function as follows:

$$v_1, v_2, v : X_{J_o} \to \mathbb{R}.$$

Let q be a $(P-1)$-dimensional polynomial of degree $r_{\mathrm{d}}^{\mathrm{sp}}$:

$$q : Y_{J_o} \to \mathbb{R}.$$

As defined in Section 3.1.1, let $\mathbf{x} \in X_{J_o}$ and let $\mathbf{y}$ denote the first $P-1$ elements of $\mathbf{x}$. Let the P-dimensional piecewise polynomial function v be defined as follows:

$$v(\mathbf{x}) = \begin{cases} v_1(\mathbf{x}), & x_P \geq q(\mathbf{y}) \\ v_2(\mathbf{x}), & x_P < q(\mathbf{y}). \end{cases}$$

Next, we describe how this piecewise polynomial function is projected onto a wavelet subspace to obtain a surfprint atom. Let $\mathcal{W}$ be a compactly supported wavelet basis in P dimensions with $H_{\mathrm{s}}^{\mathrm{wl}}$ vanishing moments. A surfprint $\mathrm{sp}(v, X_{J_o}, \mathcal{W})$ is a weighted sum of wavelet basis functions with the weights derived by projecting the piecewise polynomial v onto the subtree of basis functions whose idealized supports nest in the hypercube X_{J_o}:

$$\mathrm{sp}(v, X_{J_o}, \mathcal{W}) = \sum_{j \geq J_o, X_j \subseteq X_{J_o}} \left\langle v, w_{X_j} \right\rangle w_{X_j}, \tag{3.5}$$

where w_{X_j} represents the wavelet basis function having idealized compact support on the hypercube X_j. (The actual support of w_{X_j} may extend slightly beyond X_j.) The hypercube X_{J_o} thus defines the root node (or coarsest scale) of the surfprint atom.

We propose an approximation scheme in Section 3.4.5 where we use wavelet atoms to represent uniformly smooth regions of f^{s} and surfprint atoms to represent regions through which the discontinuity passes. Before presenting our approximation scheme, we begin in Section 3.4.3 by describing how to choose the surfprint polynomial degrees $r_{\mathrm{s}}^{\mathrm{sp}}$ and $r_{\mathrm{d}}^{\mathrm{sp}}$ and the number of vanishing moments $H_{\mathrm{s}}^{\mathrm{wl}}$ for the wavelet basis.

3.4.3 Vanishing moments and polynomial degrees

In general, when approximating elements $f^{\mathrm{s}} \in \mathcal{F}_{\mathrm{S}}(P, H_{\mathrm{d}}, H_{\mathrm{s}})$, the required surfprint polynomial degrees and wavelet vanishing moments are determined by the orders of smoothness H_{d} and H_{s}:

$$H_{\mathrm{s}}^{\mathrm{wl}} \geq H_{\mathrm{s}}, \quad r_{\mathrm{d}}^{\mathrm{sp}} = \lceil H_{\mathrm{d}} - 1 \rceil, \quad \text{and } r_{\mathrm{s}}^{\mathrm{sp}} = \lceil H_{\mathrm{s}} - 1 \rceil.$$

(This is due to Taylor's theorem.) However, the exponent in the expression of Theorem 3.2 for the optimal approximation rate for $\mathcal{F}_{\mathrm{S}}(P, H_{\mathrm{d}}, H_{\mathrm{s}})$ indicates that for every $(H_{\mathrm{d}}, H_{\mathrm{s}})$, either the $(P-1)$-dimensional discontinuity or the P-dimensional smooth region dominates the decay rate. For instance, in two dimensions, the smaller of the two smoothness orders H_{d} and H_{s} defines the decay rate.[7] This implies that the surfprint polynomial degrees and/or the number of wavelet vanishing moments can be relaxed (as if either the discontinuity or the smooth regions had a lower smoothness order), without affecting the approximation rate.

[7] We note also that in the case where the functions g_1 and g_2, which characterize f^{s} above and below the discontinuity, have differing orders of smoothness, the smaller smoothness order will determine both the achievable approximation rates and the appropriate approximation strategies.

Rather than match the surfprint parameters directly to the smoothness orders H_{d} and H_{s}, we let $H_{\mathrm{d}}^{\mathrm{sp}}$ and $H_{\mathrm{s}}^{\mathrm{sp}}$ denote the *operational smoothness orders* to which the surfprint parameters are matched. These operational smoothness orders are selected to ensure the best approximation or rate-distortion performance. The detailed derivations of [101, Appendices G–H] yield the following values for the operational smoothness orders:

- **Discontinuity dominates:** In this case, $\frac{H_{\mathrm{d}}}{P-1} < \frac{2H_{\mathrm{s}}}{P}$. We let $H_{\mathrm{d}}^{\mathrm{sp}} = H_{\mathrm{d}}$ and choose $H_{\mathrm{s}}^{\mathrm{sp}} \in [\frac{H_{\mathrm{d}}-1}{2}, H_{\mathrm{s}}]$ and $H_{\mathrm{s}}^{\mathrm{wl}} \in [\frac{H_{\mathrm{d}}P}{2(P-1)}, H_{\mathrm{s}}]$.
- **Smooth regions dominate:** In this case, $\frac{2H_{\mathrm{s}}}{P} < \frac{H_{\mathrm{d}}}{P-1}$. We let $H_{\mathrm{s}}^{\mathrm{wl}} = H_{\mathrm{s}}$, and choose $H_{\mathrm{s}}^{\mathrm{sp}} \in [H_{\mathrm{s}}(1-\frac{1}{P}) - \frac{1}{2}, H_{\mathrm{s}}]$ and $H_{\mathrm{d}}^{\mathrm{sp}} \in [\frac{2H_{\mathrm{s}}(P-1)}{P}, H_{\mathrm{d}}]$.
- **Both contribute equally:** In this case, $\frac{2H_{\mathrm{s}}}{P} = \frac{H_{\mathrm{d}}}{P-1}$. We let $H_{\mathrm{s}}^{\mathrm{wl}} = H_{\mathrm{s}}$, $H_{\mathrm{d}}^{\mathrm{sp}} = H_{\mathrm{d}}$, and choose $H_{\mathrm{s}}^{\mathrm{sp}} \in [H_{\mathrm{s}}(1-\frac{1}{P}) - \frac{1}{2}, H_{\mathrm{s}}]$.

The surfprint polynomial degrees are given by

$$r_{\mathrm{d}}^{\mathrm{sp}} = \lceil H_{\mathrm{d}}^{\mathrm{sp}} - 1 \rceil \quad \text{and} \quad r_{\mathrm{s}}^{\mathrm{sp}} = \lceil H_{\mathrm{s}}^{\mathrm{sp}} - 1 \rceil.$$

Therefore, if $\lceil H_{\mathrm{d}}^{\mathrm{sp}} - 1 \rceil < \lceil H_{\mathrm{d}} - 1 \rceil$ and $\lceil H_{\mathrm{s}}^{\mathrm{sp}} - 1 \rceil < \lceil H_{\mathrm{s}} - 1 \rceil$, then the required surfprint polynomial degrees for optimal approximations are lower than what one would naturally expect. Note that even in the scenario where both terms in the exponent of the approximation rate match, one can choose $H_{\mathrm{s}}^{\mathrm{sp}}$ slightly smaller than H_{s} while still attaining the optimal approximation rate of Theorem 3.2.

3.4.4 Quantization

In order to construct a discrete surfprint/wavelet dictionary, we quantize the coefficients of the wavelet and surfprint atoms. The quantization step-size $\Delta^{H_{\mathrm{s}}^{\mathrm{wl}}}$ for the wavelet coefficients depends on the specific parameters of an approximation scheme. We present our prototype approximation scheme and discuss the wavelet coefficient step-sizes in Section 3.4.5 (see (3.8) below).

The quantization step-size for the surfprint polynomial coefficients of order ℓ at scale j is analogous to the step-size used to construct a discrete surflet dictionary (3.3):

$$\Delta_{\ell,j}^{H_{\mathrm{d}}^{\mathrm{sp}}} = 2^{-(H_{\mathrm{d}}^{\mathrm{sp}} - \ell)j} \tag{3.6}$$

and

$$\Delta_{\ell,j}^{H_{\mathrm{s}}^{\mathrm{sp}}} = 2^{-(H_{\mathrm{s}}^{\mathrm{sp}} - \ell)j}. \tag{3.7}$$

As before, the key idea is that higher-order polynomial coefficients can be quantized with lesser precision without affecting the error term in the Taylor approximation (3.1).

3.4.5 Surfprint-based approximation

We present a tree-based representation scheme using quantized wavelet and surfprint atoms and prove that this scheme achieves the optimal approximation rate for every function $f^{\mathrm{s}} \in \mathcal{F}_{\mathrm{S}}(P, H_{\mathrm{d}}, H_{\mathrm{s}})$. Let $\mathcal{W}$ be a compactly supported wavelet basis in P dimensions with $H_{\mathrm{s}}^{\mathrm{wl}}$ vanishing moments, as defined in Section 3.4.3. Consider the decomposition of f^{s} into the wavelet basis vectors: $f^{\mathrm{s}} = \sum_j \langle f^{\mathrm{s}}, w_{X_j} \rangle w_{X_j}$. The wavelet coefficients $\langle f^{\mathrm{s}}, w_{X_j} \rangle$ are quantized according to the step-size $\Delta^{H_{\mathrm{s}}^{\mathrm{wl}}}$ defined below. Let these wavelet atoms be arranged on the nodes of a 2^P-tree. We classify the nodes based on the idealized support of the corresponding wavelet basis functions. Nodes whose supports X_j are intersected by the discontinuity b^{s} are called Type D nodes. All other nodes (over which f^{s} is smooth) are classified as Type S. Consider now the following surfprint approximation strategy:[8]

Surfprint approximation

- **Choose scales and wavelet quantization step-size:** Choose a maximal scale $J \in \mathbb{Z}$ and $m, n \in \mathbb{Z}$ such that $\frac{m}{n} = \frac{P}{P-1}$ and both m and n divide J. The quantization step-size for wavelet coefficients at all scales j is given by:

$$\Delta^{H_{\mathrm{s}}^{\mathrm{wl}}} = 2^{-\frac{J}{m}(H_{\mathrm{s}}^{\mathrm{wl}} + \frac{P}{2})} \tag{3.8}$$

  and thus depends only on the maximal scale J and the parameter m.

- **Prune tree:** Keep all wavelet nodes up to scale $\frac{J}{m}$; from scale $\frac{J}{m}$ to scale $\frac{J}{n}$, prune the tree at all Type S nodes (discarding those wavelet coefficients and their descendant subtrees).

- **Select surfprint atoms:** At scale $\frac{J}{n}$ replace the wavelet atom at each Type D discontinuity node and its descendant subtree (up to depth J) by a quantized surfprint atom chosen appropriately from the dictionary with $J_o = \frac{J}{n}$ in (3.5):

  - P-dimensional polynomials: Choose P-dimensional polynomials v_1 and v_2 of degree $r_{\mathrm{s}}^{\mathrm{sp}} = \lceil H_{\mathrm{s}}^{\mathrm{sp}} - 1 \rceil$. These polynomials should approximate the P-dimensional smooth regions up to an absolute (pointwise) error of $O\left(2^{\frac{-H_{\mathrm{s}}^{\mathrm{sp}} J}{n}}\right)$. The existence of such polynomials is guaranteed by Taylor's theorem (3.1) (let $D = P$, $H = H_{\mathrm{s}}^{\mathrm{sp}}$, and $r = r_{\mathrm{s}}^{\mathrm{sp}}$) and the quantization scheme (3.7).
  - $(P-1)$-dimensional polynomial: Choose a $(P-1)$-dimensional polynomial q of degree $r_{\mathrm{d}}^{\mathrm{sp}} = \lceil H_{\mathrm{d}}^{\mathrm{sp}} - 1 \rceil$ such that the discontinuity is approximated up

[8] The wavelet decomposition actually has $2^P - 1$ distinct directional subbands; we assume here that each is treated identically. Also we assume the scaling coefficient at the coarsest scale $j = 0$ is encoded as side information with negligible cost.

to an absolute error of $O\left(2^{\frac{-H_{\mathrm{d}}^{\mathrm{sp}} J}{n}}\right)$. The existence of such a polynomial is guaranteed by Taylor's theorem (3.1) (let $D = P - 1$, $H = H_{\mathrm{d}}^{\mathrm{sp}}$, and $r = r_{\mathrm{d}}^{\mathrm{sp}}$) and the quantization scheme of (3.6).

The following theorem summarizes the performance analysis for such surfprint approximations.

Theorem 3.7 *A surfprint-based approximation of an element $f^{\mathrm{s}} \in \mathcal{F}_{\mathrm{S}}(P, H_{\mathrm{d}}, H_{\mathrm{s}})$ as presented above achieves the optimal asymptotic approximation rate of Theorem 3.2:*

$$\left\| f^{\mathrm{s}} - \widehat{f_N^{\mathrm{s}}} \right\|_{L_2}^2 \lesssim \left(\frac{1}{N}\right)^{\min\left(\frac{H_{\mathrm{d}}}{P-1}, \frac{2H_{\mathrm{s}}}{P}\right)}.$$

Proof: See [101, Appendix G].

An approximation scheme that uses the *best* configuration of N wavelet and surfprint atoms in the L_2 sense would perform at least as well as the scheme suggested above. Hence, surfprint approximation algorithms designed to choose the best N-term approximation (even without explicit knowledge of the discontinuity or the P-dimensional smooth regions) will achieve the optimal approximation rate of Theorem 3.2.

3.4.6 Encoding a surfprint/wavelet approximation

We now consider the problem of encoding the tree-based approximation of Section 3.4.5. A simple top-down coding scheme that specifies the pruned tree topology, quantized wavelet coefficients, and surfprint parameters achieves a near-optimal rate-distortion performance.

Theorem 3.8 *A coding scheme that encodes every element of the surfprint-based approximation of an element $f^{\mathrm{s}} \in \mathcal{F}_{\mathrm{S}}(P, H_{\mathrm{d}}, H_{\mathrm{s}})$ as presented in Section 3.4.5 achieves the near-optimal asymptotic rate-distortion performance (within a logarithmic factor of the optimal performance of Theorem 3.2):*

$$\left\| f^{\mathrm{s}} - \widehat{f_R^{\mathrm{s}}} \right\|_{L_2}^2 \lesssim \left(\frac{\log R}{R}\right)^{\min\left(\frac{H_{\mathrm{d}}}{P-1}, \frac{2H_{\mathrm{s}}}{P}\right)}.$$

Proof: See [101, Appendix H].

Repeating the argument of Section 3.4.5, this near optimal rate-distortion performance serves as an upper bound for an encoding scheme that encodes elements of an L_2-best approximation. We will discuss the extension of these theoretical results to the approximation of discrete data and related issues in Section 3.5.3.

3.5 Extensions to Discrete Data

3.5.1 Overview

In this section, we consider the problem of representing discrete data obtained by "voxelizing" (or pixelizing in 2-D) functions from the classes $\mathcal{F}_{\mathrm{C}}(P, H_{\mathrm{d}})$ and $\mathcal{F}_{\mathrm{S}}(P, H_{\mathrm{d}}, H_{\mathrm{s}})$. Let f be a continuous P-dimensional function. We discretize f according to a vector $\pi = [2^{\pi_1}, \ldots, 2^{\pi_P}] \in \mathbb{Z}^P$, which specifies the number of voxels along each dimension of the discretized P-dimensional function $\widetilde{f}_\pi$. Each entry of $\widetilde{f}_\pi$ is obtained either by averaging f over a P-dimensional voxel or by sampling f at uniformly spaced intervals. (Because of the smoothness characteristics of $\mathcal{F}_{\mathrm{C}}(P, H_{\mathrm{d}})$ and $\mathcal{F}_{\mathrm{S}}(P, H_{\mathrm{d}}, H_{\mathrm{s}})$, both discretization mechanisms provide the same asymptotic performance.) In our analysis, we allow the number of voxels along each dimension to vary in order to provide a framework for analyzing various sampling rates along the different dimensions. Video data, for example, is often sampled differently in the spatial and temporal dimensions. Future research will consider different distortion criteria based on asymmetry in the spatiotemporal response of the human visual system.

For our analysis, we assume that the voxelization vector π is fixed and denote the resulting classes of voxelized functions by $\widetilde{\mathcal{F}_{\mathrm{C}}}(P, H_{\mathrm{d}})$ and $\widetilde{\mathcal{F}_{\mathrm{S}}}(P, H_{\mathrm{d}}, H_{\mathrm{s}})$. Sections 3.5.2 and 3.5.3 describe the sparse representation of elements from $\widetilde{\mathcal{F}_{\mathrm{C}}}(P, H_{\mathrm{d}})$ and $\widetilde{\mathcal{F}_{\mathrm{S}}}(P, H_{\mathrm{d}}, H_{\mathrm{s}})$, respectively. In Section 3.5.4, we discuss the impact of discretization effects on fine scale approximations. Finally, we present our simulation results in Section 3.5.5.

3.5.2 Representing and encoding elements of $\widetilde{\mathcal{F}_{\mathrm{C}}}(P, H_{\mathrm{d}})$

Suppose $f^{\mathrm{c}} \in \mathcal{F}_{\mathrm{C}}(P, H_{\mathrm{d}})$ and let $\widetilde{f}^{\mathrm{c}}_\pi \in \widetilde{\mathcal{F}_{\mathrm{C}}}(P, H_{\mathrm{d}})$ be its discretization. (We view $\widetilde{f}^{\mathrm{c}}_\pi$ as a function on the continuous domain $[0,1]^P$ that is constant over each voxel.) The process of voxelization affects the ability to approximate elements of $\widetilde{\mathcal{F}_{\mathrm{C}}}(P, H_{\mathrm{d}})$. At coarse scales, however, much of the intuition for coding $\mathcal{F}_{\mathrm{C}}(P, H_{\mathrm{d}})$ can be retained. In particular, we can bound the distance from $\widetilde{f}^{\mathrm{c}}_\pi$ to f^{c}. We note that $\widetilde{f}^{\mathrm{c}}_\pi$ differs from f^{c} only over voxels through which b passes. Because each voxel has size $2^{-\pi_1} \times 2^{-\pi_2} \cdots \times 2^{-\pi_P}$, the number of voxels intersected by b is $O\left(2^{\sum_{i=1}^{P-1} \pi_i} \left\lceil \left(\Omega \cdot 2^{-\min(\pi_i)_{i=1}^{P-1}}\right) / \left(2^{-\pi_P}\right) \right\rceil\right)$, where Ω is the universal derivative bound (Section 2.1.4). The squared-L_2 distortion incurred on each such voxel (assuming only that the voxelization process is bounded and local) is $O(2^{-(\pi_1 + \cdots + \pi_P)})$. Summing over all voxels it follows that the (nonsquared) L_2 distance obeys

$$\left\| f^{\mathrm{c}} - \widetilde{f}^{\mathrm{c}}_\pi \right\|_{L_2([0,1]^P)} < C_1 \cdot 2^{-(\min \pi_i)/2} \tag{3.9}$$

where the minimum is taken over all $i \in \{1, \ldots, P\}$.

Now we consider the problem of encoding elements of $\widetilde{\mathcal{F}_{\mathrm{C}}}(P, H_{\mathrm{d}})$. At a particular

bitrate R, we know from Theorem 3.1 that no encoder could represent all elements of $\mathcal{F}_{\mathrm{C}}(P, H_{\mathrm{d}})$ using R bits and incurring L_2 distortion less than $C_2 \cdot \left(\frac{1}{R}\right)^{\frac{H_{\mathrm{d}}}{2(P-1)}}$. (This lower bound for metric entropy is in effect for R sufficiently large, which we assume to be the case.) Suppose we consider a hypothetical encoder for elements of $\widetilde{\mathcal{F}_{\mathrm{C}}}(P, H_{\mathrm{d}})$ that, using R bits, could represent any element with L_2 distortion of $\widetilde{\mathcal{F}_{\mathrm{C}}}(P, H_{\mathrm{d}})$ less than some $D_{\mathrm{hyp}}(R)$. This coder could *also* be used as an encoder for elements of $\mathcal{F}_{\mathrm{C}}(P, H_{\mathrm{d}})$ (by voxelizing each function before encoding). This strategy would yield L_2 distortion no worse than $C_1 \cdot 2^{-(\min \pi_i)/2} + D_{\mathrm{hyp}}(R)$. By applying the metric entropy arguments on $\mathcal{F}_{\mathrm{C}}(P, H_{\mathrm{d}})$, we have the following constraint on $D_{\mathrm{hyp}}(R)$:

$$C_1 \cdot 2^{-(\min \pi_i)/2} + D_{\mathrm{hyp}}(R) \geq C_2 \cdot \left(\frac{1}{R}\right)^{\frac{H_{\mathrm{d}}}{2(P-1)}},$$

or equivalently,

$$D_{\mathrm{hyp}}(R) \geq C_2 \cdot \left(\frac{1}{R}\right)^{\frac{H_{\mathrm{d}}}{2(P-1)}} - C_1 \cdot 2^{-(\min \pi_i)/2}. \tag{3.10}$$

This inequality helps establish a rate-distortion bound for the class $\widetilde{\mathcal{F}_{\mathrm{C}}}(P, H_{\mathrm{d}})$. At sufficiently low rates, the first term on the RHS dominates, and $\widetilde{\mathcal{F}_{\mathrm{C}}}(P, H_{\mathrm{d}})$ faces similar rate-distortion constraints to $\mathcal{F}_{\mathrm{C}}(P, H_{\mathrm{d}})$. At high rates, however, the RHS becomes negative, giving little insight into the coding of $\widetilde{\mathcal{F}_{\mathrm{C}}}(P, H_{\mathrm{d}})$. This breakdown point occurs when $R \sim 2^{(\min \pi_i)(P-1)/H_{\mathrm{d}}}$.

We can, in fact, specify a constructive encoding strategy for $\widetilde{\mathcal{F}_{\mathrm{C}}}(P, H_{\mathrm{d}})$ that achieves the optimal compression rate up to this breakdown point. We construct a dictionary of discrete surflet atoms by voxelizing the elements of the continuous quantized surflet dictionary. Assuming there exists a technique to find discrete ℓ_2-best surflet fits to $\widetilde{f}_{\pi}^{\mathrm{c}}$, the tree-based algorithm described in Section 3.3.3 can simply be used to construct an approximation $\widehat{\widetilde{f}}_{\pi}^{\mathrm{c}}$.

Theorem 3.9 *While $R \lesssim 2^{(\min \pi_i)(P-1)/H_{\mathrm{d}}}$, the top-down predictive surflet coder from Section 3.3.5 applied to encode the approximation $\widehat{\widetilde{f}}_{\pi}^{\mathrm{c}}$ to $\widetilde{f}_{\pi}^{\mathrm{c}}$ using discrete ℓ_2-best surflets achieves the rate-distortion performance*

$$\left\| \widetilde{f}_{\pi}^{\mathrm{c}} - \widehat{\widetilde{f}}_{\pi}^{\mathrm{c}} \right\|_{L_2}^2 \lesssim \left(\frac{1}{R}\right)^{\frac{H_{\mathrm{d}}}{P-1}}.$$

Proof: See [101, Appendix I].

As detailed in the proof of this theorem, the breakdown point occurs when using surflets at a critical scale $J_{\mathrm{vox}} = \frac{\min \pi_i}{H_{\mathrm{d}}}$. Up to this scale, all of the familiar approximation and compression rates hold. Beyond this scale, however, voxelization effects

dominate. An interesting corollary to Theorem 3.9 is that, due to the similarities up to scale J_{vox}, the discrete approximation $\widehat{\widetilde{f}}_{\pi}^{\,\text{c}}$ itself provides an effective approximation to the function f^{c}.

Corollary 3.1 *While $R \lesssim 2^{(\min \pi_i)(P-1)/H_{\text{d}}}$, the discrete approximation $\widehat{\widetilde{f}}_{\pi}^{\,\text{c}}$ provides an approximation to f^{c} with the following rate-distortion performance:*

$$\left\| f^{\text{c}} - \widehat{\widetilde{f}}_{\pi}^{\,\text{c}} \right\|_{L_2}^2 \lesssim \left(\frac{1}{R} \right)^{\frac{H_{\text{d}}}{P-1}}.$$

Proof: See [101, Appendix J].

While we have provided an effective strategy for encoding elements of $\widetilde{\mathcal{F}_{\text{C}}}(P, H_{\text{d}})$ at sufficiently low rates (using surflets at scales $j \leq J_{\text{vox}}$), this leaves open the question of how to code $\widetilde{\mathcal{F}_{\text{C}}}(P, H_{\text{d}})$ at higher rates. Unfortunately, (3.10) does not offer much insight. In particular, it is not clear whether surflets are an efficient strategy for encoding $\widetilde{\mathcal{F}_{\text{C}}}(P, H_{\text{d}})$ beyond scale J_{vox}. We revisit this issue in Section 3.5.4.

3.5.3 Representing and encoding elements of $\widetilde{\mathcal{F}_{\text{S}}}(P, H_{\text{d}}, H_{\text{s}})$

Next, let $\widetilde{f}_{\pi}^{\,\text{s}}$ be an arbitrary signal belonging to $\widetilde{\mathcal{F}_{\text{S}}}(P, H_{\text{d}}, H_{\text{s}})$. Similar arguments apply to the voxelization effects for this class. In order to approximate functions in $\widetilde{\mathcal{F}_{\text{S}}}(P, H_{\text{d}}, H_{\text{s}})$, we use a dictionary of compactly supported discrete wavelet basis functions with H_{s}^{wl} vanishing moments and discrete surfprint atoms. A discrete surfprint atom is derived by projecting a discrete piecewise polynomial surflet atom onto a subspace of the discrete wavelet basis.

We use the scheme described in Section 3.4.5 with $\frac{J_{\text{vox}}}{n} = \frac{\min(\pi_i)}{\min\left(H_{\text{d}}^{\text{sp}}, 2H_{\text{s}}^{\text{sp}}+1\right)}$ to approximate $\widetilde{f}_{\pi}^{\,\text{s}}$ by $\widehat{\widetilde{f}}_{\pi}^{\,\text{s}}$. According to [101, Appendix H], this scale corresponds to a range of bitrates up to $O(J_{\text{vox}} 2^{(P-1)\frac{J_{\text{vox}}}{n}})$. Within this range, the approximation is encoded as described in Section 3.4.6. The performance of this scheme appears below.

Theorem 3.10 *While $R \lesssim J_{\text{vox}} 2^{(P-1)\frac{J_{\text{vox}}}{n}}$ where $J_{\text{vox}} = \frac{n \cdot \min(\pi_i)}{\min\left(H_{\text{d}}^{\text{sp}}, 2H_{\text{s}}^{\text{sp}}+1\right)}$, the coding scheme from Section 3.4.5 applied to encode the approximation $\widehat{\widetilde{f}}_{\pi}^{\,\text{s}}$ to $\widetilde{f}_{\pi}^{\,\text{s}}$ using a discrete wavelet/surfprint dictionary achieves the following near-optimal asymptotic rate-distortion performance (within a logarithmic factor of the optimal performance of Theorem 3.2):*

$$\left\| \widetilde{f}_{\pi}^{\,\text{s}} - \widehat{\widetilde{f}}_{\pi}^{\,\text{s}} \right\|_{L_2}^2 \lesssim \left(\frac{\log R}{R} \right)^{\min\left(\frac{H_{\text{d}}}{P-1}, \frac{2H_{\text{s}}}{P}\right)}.$$

Proof: See [101, Appendix K].

Again, a corollary follows naturally.

Corollary 3.2 *While* $R \lesssim J_{\text{vox}} 2^{(P-1)\frac{J_{\text{vox}}}{n}}$, *the discrete approximation* $\widehat{\widetilde{f}}_{\pi}^{\text{s}}$ *provides an approximation to* f^{s} *with the following rate-distortion performance:*

$$\left\| f^{\text{s}} - \widehat{\widetilde{f}}_{\pi}^{\text{s}} \right\|_{L_2}^2 \lesssim \left(\frac{\log R}{R} \right)^{\min\left(\frac{H_{\text{d}}}{P-1}, \frac{2H_{\text{s}}}{P}\right)}.$$

Proof: See [101, Appendix L].

3.5.4 Discretization effects and varying sampling rates

We have proposed surflet algorithms for discrete data at sufficiently coarse scales. Unfortunately, this leaves open the question of how to represent such data at finer scales. In this section, we discuss one perspective on fine scale approximation that leads to a natural surflet coding strategy.

Consider again the class $\widetilde{\mathcal{F}_{\text{C}}}(P, H_{\text{d}})$. Section 3.5.2 provided an effective strategy for encoding elements of $\widetilde{\mathcal{F}_{\text{C}}}(P, H_{\text{d}})$ at sufficiently low rates (using surflets at scales $j \leq J_{\text{vox}} = \frac{\min \pi_i}{H_{\text{d}}}$). Beyond scale J_{vox}, however, the voxelization effects dominate the resolution afforded by surflet approximations. To restore a balance, we suggest a coding strategy for finer scales based on the observation that $\mathcal{F}_{\text{C}}(P, H_{\text{d}}) \subset \mathcal{F}_{\text{C}}(P, H)$ for $H < H_{\text{d}}$. Surflet approximations on the class $\mathcal{F}_{\text{C}}(P, H)$ (tied to the smoothness H) have lower accuracy in general. As a result, $\widetilde{\mathcal{F}_{\text{C}}}(P, H)$ has a higher "breakdown rate" than $\widetilde{\mathcal{F}_{\text{C}}}(P, H_{\text{d}})$, and discrete surflets tailored for smoothness H will achieve the coding rate $O(R^{-H/(P-1)})$ up to scale $\frac{\min \pi_i}{H}$. While this may not be a worthwhile strategy before scale J_{vox}, it could be useful beyond scale J_{vox} and up to scale $\frac{\min \pi_i}{H}$. In fact, beyond that scale, we can *again* reduce H, obtaining a new breakdown rate and a finer scale to code (using lower-order surflets). This gives us a concrete strategy for coding $\widetilde{\mathcal{F}_{\text{C}}}(P, H_{\text{d}})$ at all scales, although our optimality arguments apply only up to scale J_{vox}. At scale j, we use surflets designed for smoothness $H_j = \min\left(H_{\text{d}}, \frac{\min(\pi_i)}{j}\right)$, $0 \leq j \leq \min(\pi_i)$. A surflet dictionary constructed using such scale-adaptive smoothness orders consists of relatively few elements at coarse scales (due to the low value of j in the quantization stepsize) and relatively few at fine scales (due to the decrease of H_j), but many elements at medium scales. This agrees with the following intuitive notions:

- The large block sizes at coarse scales do not provide sufficient resolution to warrant large dictionaries for approximation at these scales.
- The relatively small number of voxels in each block at very fine scales also means that a coder does not require large dictionaries in order to approximate blocks at such scales well.

- At medium scales where the block sizes are small enough to provide good resolution but large enough to contain many voxels, the dictionary contains many elements in order to provide good approximations.

Similar strategies can be proposed, of course, for the class $\widetilde{\mathcal{F}_{\mathrm{S}}}(P, H_{\mathrm{d}}, H_{\mathrm{s}})$.

Finally we note that the interplay between the sampling rate (number of voxels) along the different dimensions and the critical approximation scale J_{vox} can impact the construction of multiscale source coders. As an example of the potential effect of this phenomenon in real-world applications, the sampling rate along the temporal dimension could be the determining factor when designing a surfprint-based video coder because this rate tends to be lower than the sampling rate along the spatial dimensions.

3.5.5 Simulation results

To demonstrate the potential for coding gains based on surflet representations, we perform the following numerical experiments in 2 and 3 dimensions.

2-D coding

We start by coding elements of $\widetilde{\mathcal{F}_{\mathrm{C}}}(P, H_{\mathrm{d}})$ with $P = 2$ and $H_{\mathrm{d}} = 3$. We generate 1024×1024 discretized versions of these images (that is, $\pi_1 = \pi_2 = 10$). Our two example images are shown in Figures 3.5(a) and 3.6(a).

On each image we test three types of surflet dictionaries for encoding.

- Dictionary 1 uses wedgelets as implemented in our previous work [102, 108]. In this dictionary we do not use the quantization stepsizes as specified in (3.3). Rather, we use a quantization stepsize $\Delta_{\ell,j} \sim 2^{-(1-\ell)j}$. As a result, the quantized wedgelet dictionary has the same cardinality at each scale and is self-similar (simply a dyadic scaling of the dictionary at other scales).

- Dictionary 2 adapts with scale. Following the arguments of Section 3.5.4, at a given scale j, we use surflets tailored for smoothness $H_j = \min(2, \frac{\min \pi_i}{j}) = \min(2, \frac{10}{j})$. We use surflets of the appropriate polynomial order and quantize the polynomial coefficients analogous to (3.3); that is, $\Delta_{\ell,j} \sim 2^{-(H_j-\ell)j}$. The limitation $H_j \leq 2$ restricts our surflets to linear polynomials (wedgelets) for comparison with the first dictionary above.

- Dictionary 3 is a surflet dictionary that also adapts with scale. This dictionary is constructed similarly to the second, except that it is tailored to the actual smoothness of f^{c}: we set $H_j = \min(H_{\mathrm{d}}, \frac{\min \pi_i}{j}) = \min(H_{\mathrm{d}}, \frac{10}{j})$. This modification allows quadratic surflets to be used at coarse scales $0 \leq j \leq 5$, beyond which H_j again dictates that wedgelets are used.

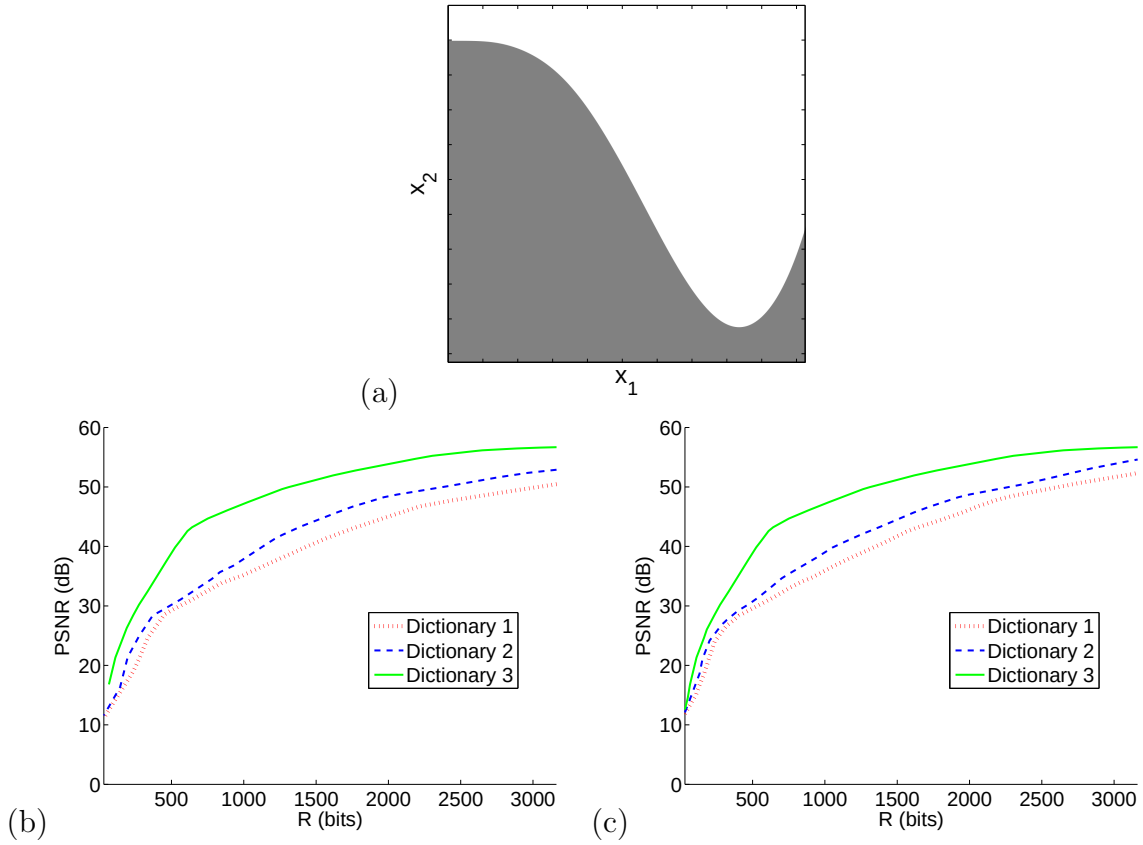

Figure 3.5: *(a) Test function $\widetilde{f_\pi^c}$. (b) Rate-distortion performance for each dictionary (with the best fixed set of dictionary parameters). (c) Rate-distortion performance for each dictionary (selected using best convex hull in R-D plane over all dictionary parameters).*

For each dictionary, we must also specify the range of allowable polynomial coefficients and a constant multiplicative factor on each quantization stepsize. We optimize these parameters through simulation.

Our coding strategy for each dictionary uses a top-down prediction. Based on the prediction from a (previously coded) parent surflet, we partition the set of possible children surflets into two classes for entropy coding. A probability mass of ρ is distributed among the W surflets nearest the predicted surflet (measured using ℓ_2 distance), and a probability mass of $(1-\rho)$ is distributed among the rest to allow for robust encoding. We optimize the choice of W and ρ experimentally.

To find the discrete ℓ_2-best fit surflet to a given block, we use a coarse-to-fine manifold search as suggested in Chapter 4. Based on the costs incurred by this coding scheme, we optimize the surflet tree pruning using a Lagrangian tradeoff parameter λ. We repeat the experiment for various values of λ.

Figure 3.5(b) shows what we judge to be the best R-D curve for each dictionary (Dictionary 1: dotted curve, 2: dashed curve, and 3: solid curve). Each curve is generated by sweeping λ but fixing one combination of polynomial parameters/constants.

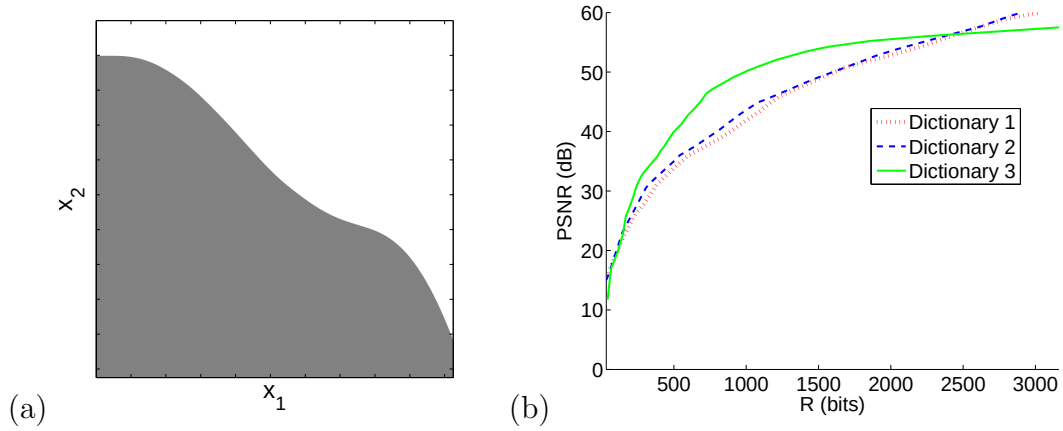

Figure 3.6: *(a) Test function $\widetilde{f_\pi^c}$. (b) Rate-distortion performance for each dictionary (selected using best convex hull in R-D plane over all dictionary parameters).*

Table 3.1: *Surflet dictionary size at each scale (using the surflet parameters chosen to generate Figure 3.5(b)). Our surflet dictionaries (2 and 3) adapt to scale, avoiding unnecessary precision at coarse and fine scales.*

Scale j	0	1	2	3	4	5	6	7	8	9
Dictionary 1	1.8e5	1.8e5	1.8e5	1.8e5	1.8e5	1.8e5	1.8e5	1.8e5	1.8e5	1.8e5
Dictionary 2	2.2e2	4.1e3	6.3e4	9.9e5	9.9e5	2.5e5	6.3e4	1.6e4	4.1e3	1.1e3
Dictionary 3	3.6e2	1.4e4	4.1e5	1.2e7	6.3e6	2.5e5	6.3e4	1.6e4	4.1e3	1.1e3

Over all simulations (all polynomial parameters/constants), we also take the convex hull over all points in the R-D plane. The results are plotted in Figures 3.5(c) and 3.6(b).

We see from the figures that Dictionary 2 outperforms Dictionary 1, requiring 0-20% fewer bits for an equivalent distortion (or improving PSNR by up to 4dB at a given bitrate). Both dictionaries use wedgelets — we conclude that the coding gain comes from the adaptivity through scale. Table 3.1 lists the number of admissible quantized surflets as a function of scale j for each of our three dictionaries.

We also see from the figures that Dictionary 3 often outperforms Dictionary 2, requiring 0-50% fewer bits for an equivalent distortion (or improving PSNR by up to 10dB at a given bitrate). Both dictionaries adapt to scale — we conclude that the coding gain comes from the quadratic surflets used at coarse scales (which are designed to exploit the actual smoothness $H_\mathrm{d} = 3$). Figure 3.7 compares two pruned surflet decompositions using Dictionaries 2 and 3. In this case, the quadratic dictionary offers comparable distortion using 40% fewer bits than the wedgelet dictionary.

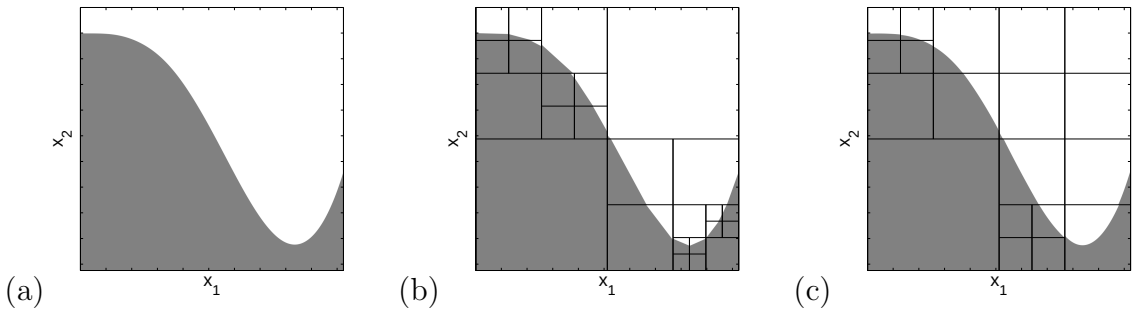

Figure 3.7: *Comparison of pruned surflet tilings using two surflet dictionaries. (a) Test image with $P = 2$ and $H_{\mathrm{d}} = 3$. (b) The wedgelets from Dictionary 2 can be encoded using 482 bits and yields PSNR 29.86dB. (c) The quadratic/wedgelet combination from Dictionary 3 can be encoded using only 275 bits and yields PSNR 30.19dB.*

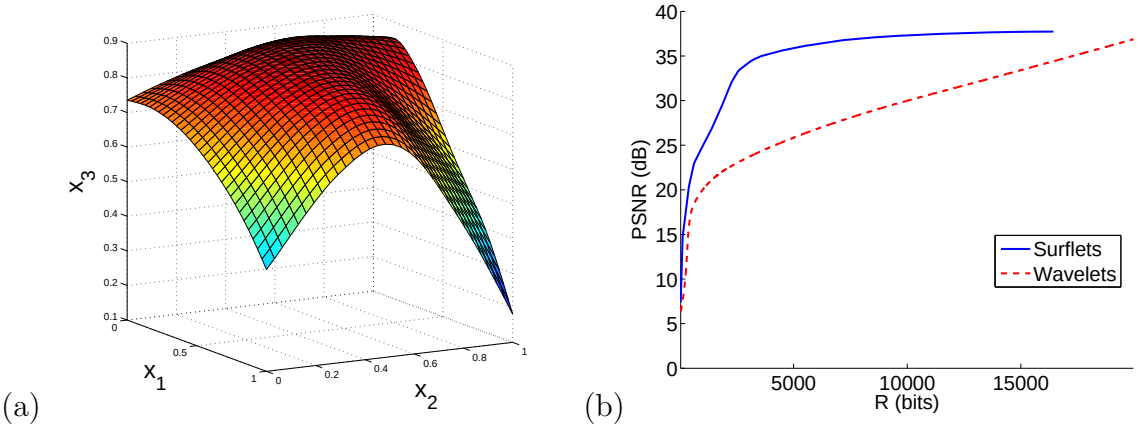

Figure 3.8: *(a) Horizon b^{c} used to generate 3-D test function $\widetilde{f_{\pi}^{\mathrm{c}}}$. (b) Rate-distortion performance for surflet coding compared with wavelet coding.*

3-D coding

We now describe numerical experiments for coding elements of $\widetilde{\mathcal{F}_{\mathrm{C}}}(P, H_{\mathrm{d}})$ and $P = 3$. We generate $64 \times 64 \times 64$ discretized versions of these signals (that is, $\pi_i = 6$). Our two example discontinuities b^{c} are shown in Figure 3.8(a) (for which $H_{\mathrm{d}} = 2$) and Figure 3.10(a) (for which $H_{\mathrm{d}} = \infty$).

For these simulations we compare surflet coding (analogous to Dictionary 2 above, with $H_j = \min(2, \frac{6}{j})$) with wavelet coding. Our wavelet coding is based on a 3-D Haar wavelet transform, which we threshold at a particular level (keeping the largest wavelet coefficients). For the purpose of the plots we assume (optimistically) that each significant wavelet coefficient was coded with zero distortion using only three bits per coefficient. We see from the figures that surflet coding significantly outperforms the wavelet approach, requiring up to 80% fewer bits than our aggressive wavelet estimate (or improving PSNR by up to 10dB a given bitrate). Figure 3.9 shows one

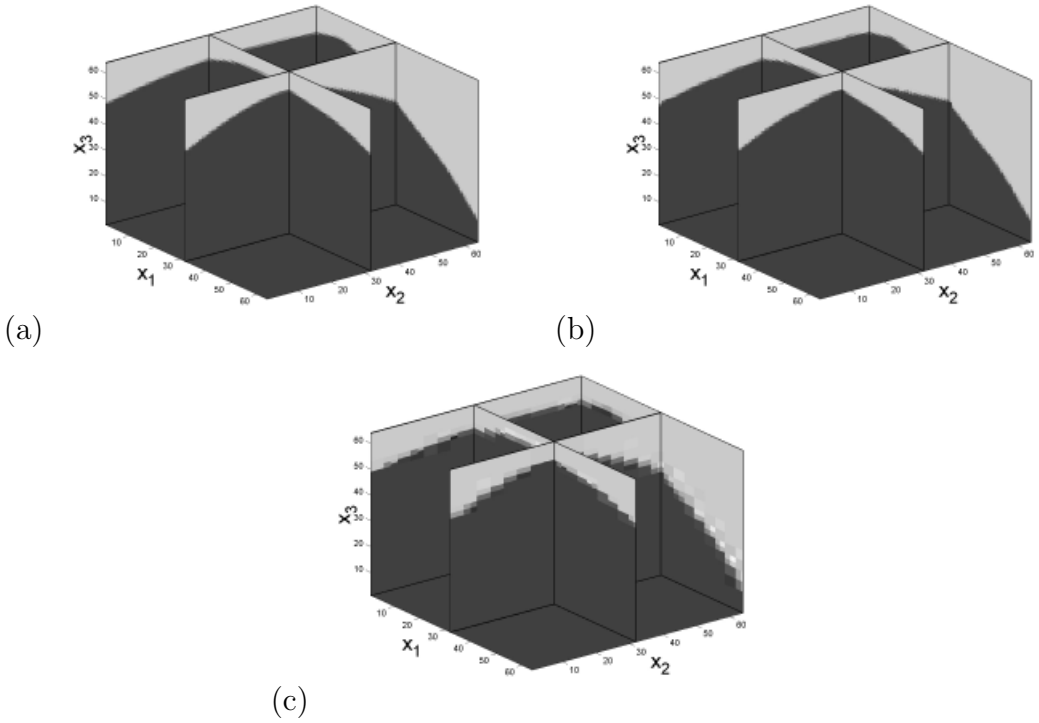

Figure 3.9: *Volumetric slices of 3-D coded functions. (a) Original test function* $\widetilde{f_\pi^c}$ *from Figure 3.8. (b) Surflet-coded function using 2540 bits; PSNR 33.22dB. (c) Wavelet-coded function using approximately 2540 bits; PSNR 23.08dB.*

set of coded results for the function in Figure 3.8; at an equivalent bitrate, we see that surflets offer a significant improvement in PSNR and a dramatic reduction in ringing/blocking artifacts compared with wavelets. We also notice from Figures 3.8 and 3.10, however, that at high bitrates the gains diminish relative to wavelets. We believe this is due to small errors made in the surflet estimates at fine scales using our current implementation of the manifold-based technique.

Future work will focus on improved surflet estimation algorithms; however using even these suboptimal estimates we *still* see superior performance across a wide range of bitrates. In Chapter 7, we discuss additional possible extensions of the multiscale surflet/surfprint framework to incorporate new local models and representations.

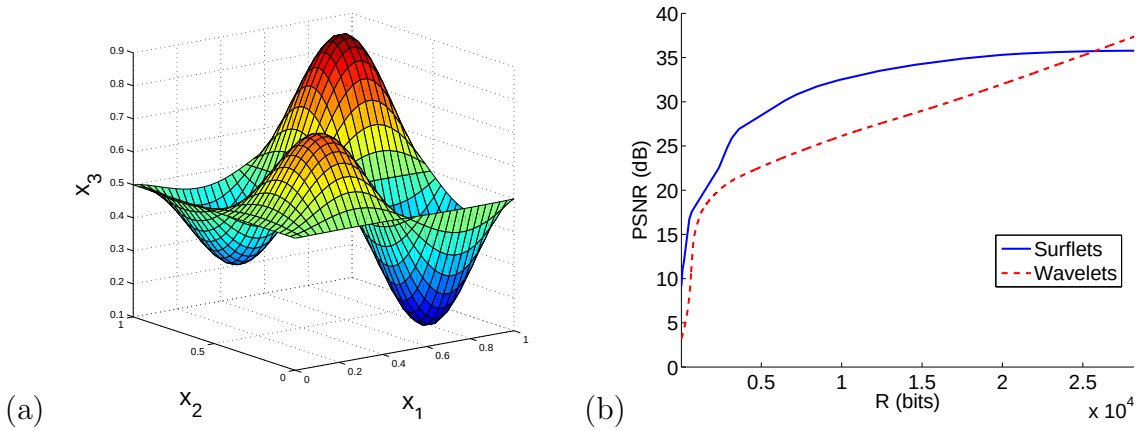

Figure 3.10: *(a) Horizon b^c used to generate 3-D test function $\widetilde{f^c_\pi}$. (b) Rate-distortion performance for surflet coding compared with wavelet coding.*

Chapter 4
The Multiscale Structure of Non-Differentiable Image Manifolds

In Chapter 3, we considered a simple model for real world signals and, observing the shortcomings of sparse representations for such signals, proposed specific parametric atoms designed to provide highly accurate local approximations. Recalling the geometric viewpoint discussed in Section 2.4.3, then, we may argue that such local signal regions live not near a union of low-dimensional hyperplanes (which could be captured by some sparse dictionary), but rather near a low-dimensional manifold generated by considering all possible surflet polynomial parameters.

In this chapter,[1] we study the geometry of signal manifolds in more detail, particularly in the case of image manifolds such as the 2-D surflet manifold. More precisely, we consider specific families of images related by changes of a natural articulation parameter θ controlling the image formation. Examples of such parameters include translation, rotation, and position of an object. Such image families form low-dimensional manifolds in the high-dimensional ambient space. We call them *image appearance manifolds* (IAMs). We let Θ denote the space of parameters and denote by f_θ the image formed by a particular $\theta \in \Theta$. The particular IAM is then given by $\mathcal{F} = \{f_\theta : \theta \in \Theta\}$.

The articulation parameters we consider represent simple and fundamental examples of the prototypical information that comprises an image; our study of IAM geometry gives new insight into the basic structural relationships that relate one image to another.

Our work builds upon a surprising realization [16]: IAMs of continuous images having sharp edges that move as a function of θ are *nowhere differentiable*. This presents an immediate challenge for signal processing algorithms that might assume differentiability or smoothness of such manifolds. As a motivating example, we consider the problem of recovering, from an observed image I on or near the manifold, the parameter θ that best describes that image. (This problem arises, for example, in finding the best surflet fit to a given image segment.) A natural least-squares approach to solving such a problem using Newton's method would involve a sequence of projections onto tangent planes along the manifold. Because the manifold is not differentiable, however, these tangents do not exist.

Although these IAMs lack differentiability in the traditional sense, we identify

[1] This work is in collaboration with David Donoho, Hyeokho Choi, and Richard Baraniuk [19].

a multiscale collection of tangent spaces to the manifold, each one associated with both a *location* on the manifold and *scale* of analysis. (This multiscale characterization of the non-differentiable manifold is not unlike the wavelet analysis of a non-differentiable function [109].) We describe a simple experiment to reveal the multiscale structure, based on local hyperplane fits to neighborhoods on the manifold. At a particular point f_θ on the manifold, as the size ϵ of the neighborhood of analysis shrinks, the planes continue to "twist off" and never converge to a fixed tangent space. We also describe a second technique for accessing this multiscale structure by *regularizing* the individual images f_θ with a kernel of width s. The resulting manifold of regularized images $f_{\theta,s}$ (lacking sharp edges) *is* differentiable and more amenable to computation and analysis.

To address the parameter estimation problem, we then propose a Multiscale Newton search, using a sequence of regularized manifolds and letting the scale parameter $s \to 0$. The algorithm typically converges within just a few iterations and returns very accurate results. Our multiscale approach shares common features with a number of practical "coarse-to-fine differential estimation" methods of image registration [110–115] but can offer new justification and perspective on the relevant issues.

We also reveal a second, more localized kind of IAM non-differentiability caused by *occlusion*. When an occluding surface exists in a scene, there will generally exist special parameter points at which infinitesimal changes in the parameter can make an edge vanish/appear from behind the occlusion (e.g., a rotating cube in 3-D at the point where a face is appearing/disappearing from view). These articulations correspond to multiscale cusps in the IAM with different "left" and "right" approximate tangent spaces; the local dimensionality of the tangent space changes abruptly at such points. This type of phenomenon has its own implications in the signal processing and requires a special vigilance; it is not alleviated by merely regularizing the images.

This chapter is organized as follows. Section 4.1 elaborates on the manifold viewpoint for articulated image families. Section 4.2 explores the first type of non-differentiability, caused by the migration of edges. Section 4.3 analyzes the multiscale tangent twisting behavior in more depth. Section 4.4 explores the second type of non-differentiability, due to occlusion of edges. Section 4.5 considers the problem of parameter estimation given an unlabeled image and includes numerical experiments.

4.1 Image Appearance Manifolds (IAMs)

We consider images both over the unbounded domain $\mathbb{R}^2$ and over bounded domains such as the unit square $[0,1] \times [0,1]$. In this chapter, we use $x = (x_0, x_1)$ to denote the coordinates of the image plane. We are interested in families of images formed by varying a parameter $\theta \in \Theta$ that controls the *articulation* of an object being imaged and thus its *appearance* in each image. For example, θ could be a translation parameter in $\mathbb{R}^3$ specifying the location of the object in a scene; an orientation parameter in SO(3) specifying its pose; or an articulation parameter specifying, for a

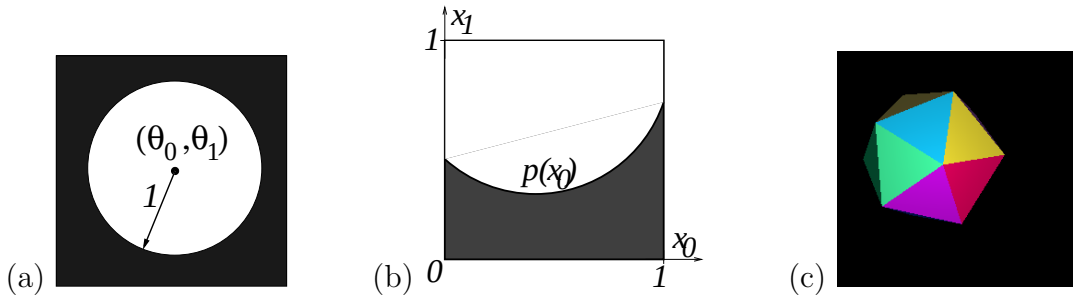

Figure 4.1: *Simple image articulation models. (a) Parametrization of translating disk image f_θ. (b) Parametrization of a surflet. (c) Simulated photograph of a 3-D icosahedron.*

composite object, the relative placement of mobile components. We let K denote the dimension of θ.

The image formed with parameter θ is a function $f_\theta : \mathbb{R}^2 \mapsto \mathbb{R}$; the corresponding family is the K-dimensional *image appearance manifold* (IAM) $\mathcal{F} = \{f_\theta : \theta \in \Theta\}$. We assume that the relation $\theta \mapsto f_\theta$ is one-to-one. The set $\mathcal{F}$ is a collection of functions, and we suppose that all of these functions are square-integrable: $\mathcal{F} \subset L^2(\mathbb{R}^2)$. Equipping $\mathcal{F}$ with the L^2 metric, we induce a metric on Θ

$$\mu\big(\theta^{(0)}, \theta^{(1)}\big) = \|f_{\theta^{(0)}} - f_{\theta^{(1)}}\|_{L^2} . \tag{4.1}$$

Assuming that $\theta \mapsto f_\theta$ is a continuous mapping for the L^2 metric, $M = (\Theta, \mu)$ is a metric space.

We use a range of models to illustrate the structural phenomena of IAMs and highlight the basic challenges that can arise in image processing. Similar models are discussed in [16, 17]; the most elaborate such involve combining models to create, for example, articulating cartoon faces.

4.1.1 Articulations in the image plane

The simplest IAMs are formed by articulating cartoon shapes within the image plane. First, consider *translations of an indicator function* in the image plane. Let f_0 be an indicator function in $\mathbb{R}^2$ — a disk, ellipse, square, or rectangle, for example. Let $\Theta = \mathbb{R}^2$ act on the indicator function according to $f_\theta(x) = f_0(x-\theta)$; see Figure 4.1(a) for an example with the unit disk. Then it is easy to see that $\mu(\theta^{(0)}, \theta^{(1)}) = m(\|\theta^{(0)} - \theta^{(1)}\|)$ for a monotone increasing function $m \geq 0$, $m(0) = 0$. In fact, if we let B_y denote the indicator function centered at $y \in \mathbb{R}^2$, then

$$m(\rho) = \mathrm{Area}(B_{(0,0)} \triangle B_{(\rho,0)})^{1/2},$$

where $\triangle$ denotes the symmetric difference: $A \triangle B = (A \backslash B) \cup (B \backslash A)$.

In a bounded image domain, a translating indicator function will eventually reach one or both frontiers, where it begins changing shape until it finally disappears completely. We will discuss this *occlusion* phenomenon in more detail in Section 4.4.

Surflets offer another bounded domain model (see Chapter 3). If we let $p : [0,1] \mapsto \mathbb{R}$ be a polynomial of degree $r > 1$ and let $\theta \in \mathbb{R}^{r+1}$ denote the set of polynomial coefficients, then the resulting surflet on the unit square is given by $s([0,1]^2; p; x) = \mathbf{1}_{\{x_1 \geq p(x_0),\, x_1 \in [0,1]\}}$ (see Figure 4.1(b)).

4.1.2 Articulations of 3-D objects

Our model is not limited just to articulations in the image plane. Consider, for example, photography of a 3-D object. In this case, the object may be subject to translations ($\Theta = \mathbb{R}^3$), rotations ($\Theta = \mathrm{SO}(3)$), or a combination of both; the metric on Θ simply involves the difference between two rendered images as in (4.1). Figure 4.1(d) shows an example rendering of an icosahedron at an arbitrary position. Additional articulation parameters, such as camera position or lighting conditions [116], can also be considered.

4.2 Non-Differentiability from Edge Migration

Each of the image models mentioned in Section 4.1 involves sharp edges that move as a function of the parameter θ. This simple effect, relevant in many natural settings where images may feature objects having unknown or moving locations, has a profound consequence on the structure of the resulting IAMs: these manifolds are *nowhere* differentiable. This presents an apparent difficulty for image understanding algorithms that might attempt to exploit the local manifold geometry using calculus.

4.2.1 The problem

This lack of differentiability can be seen analytically: the metric spaces resulting from the IAMs in Section 4.1 all have a *non-Lipschitz* relation between the metric distance and the Euclidean distance. As one can check by detailed computations [16], we have

$$\mu\big(\theta^{(0)}, \theta^{(1)}\big) \geq c \left\|\theta^{(0)} - \theta^{(1)}\right\|_2^{1/2} \qquad \text{as } \mu \to 0.$$

The exponent $1/2$ — rather than 1 — implies that the parametrization $\theta \mapsto f_\theta$ is not differentiable. As with a standard function of Hölder regularity $1/2$, we are unable to compute the derivative. For example, to estimate $\frac{\partial f_\theta}{\partial \theta_i}\big|_{\theta=\theta^{(0)}}$, we would let $\theta^{(0)}$ and $\theta^{(1)}$ differ only in component θ_i and would observe that

$$\left\| \frac{f_{\theta^{(1)}} - f_{\theta^{(0)}}}{\theta_i^{(1)} - \theta_i^{(0)}} \right\|_2 \geq c \left\|\theta^{(1)} - \theta^{(0)}\right\|_2^{-1/2} \to \infty \qquad \text{as } \theta^{(1)} \to \theta^{(0)}.$$

This relation is non-differentiable at *every* parameter θ for which local perturbations cause edges to move. Moreover, this failure of differentiability is not something removable by mere reparametrization; no parametrization exists under which there would be a differentiable relationship.

We can also view this geometrically. The metric space $M = (\Theta, \mu)$ is isometric to $\mathcal{F} = (\mathcal{F}, \|\cdot\|_{L^2})$. $\mathcal{F}$ is not a smooth manifold; there simply is no system of charts that can make $\mathcal{F}$ even a C^1 manifold. At base, the lack of differentiability of the manifold $\mathcal{F}$ is due to the lack of *spatial* differentiability of these images [16]. In brief: *images have edges, and if the locations of edges move as the parameters change then the manifold is not smooth.*

4.2.2 Approximate tangent planes via local PCA

An intrinsic way to think about non-smoothness is to consider approximate tangent planes generated by local principal component analysis (PCA) [43]. Suppose we pick an ϵ-neighborhood $N_\epsilon(\theta^{(0)}; \Theta)$ of some $\theta^{(0)} \in \Theta$; this induces a neighborhood $N_\epsilon(f_{\theta^{(0)}}; \mathcal{F})$ around the point $f_{\theta^{(0)}} \in \mathcal{F}$. We define the *$\epsilon$-tangent plane to $\mathcal{F}$ at $f_{\theta^{(0)}}$* as follows. We place a uniform probability measure on $\theta \in N_\epsilon(\theta^{(0)}; \Theta)$, thereby inducing a measure ν on the neighborhood $N_\epsilon(f_{\theta^{(0)}})$. Viewing this measure as a probability measure on a subset of L^2, we can obtain the first K principal components of that probability measure. These K functions span a K-dimensional affine hyperplane, the approximate tangent plane $T^\epsilon_{f_{\theta^{(0)}}}(\mathcal{F})$; it is an approximate least-squares fit to the manifold over the neighborhood $N_\epsilon(f_{\theta^{(0)}})$.

If the manifold were differentiable, then the approximate tangent planes $T^\epsilon_{f_{\theta^{(0)}}}(\mathcal{F})$ would converge to a fixed K-dimensional space as $\epsilon \to 0$; namely, the plane spanned by the K directional derivatives $\frac{\partial}{\partial \theta_i} f_\theta\big|_{\theta=\theta^{(0)}}$, $i = 0, 1, \ldots, K-1$. However, when these do not exist, the approximate tangent planes do not converge as $\epsilon \to 0$, but continually "twist off" into other dimensions.

Consider as an example the translating disk model, where the underlying parametrization is 2-D and the tangent planes are 2-D as well. Figure 4.2(a) shows the approximate tangent plane obtained from this approach at scale $\epsilon = 1/4$. The tangent plane has a basis consisting of two elements, each of which can be considered an image. Figure 4.2(b) shows the tangent plane basis images at the finer scale $\epsilon = 1/8$. It is visually evident that the tangent plane bases at these two scales are different; in fact the angle between the two subspaces is approximately 30°. Moreover, since the basis elements resemble annuli of shrinking width and growing amplitude, it is apparent for continuous-domain images[2] that as $\epsilon \to 0$, the tangent plane bases cannot converge in L^2.

[2]In the case of a *pixelized* image, this phenomenon cannot continue indefinitely. However, the twisting behavior does continue up until the very finest scale, making our analysis relevant for practical algorithms (e.g., see Section 4.5).

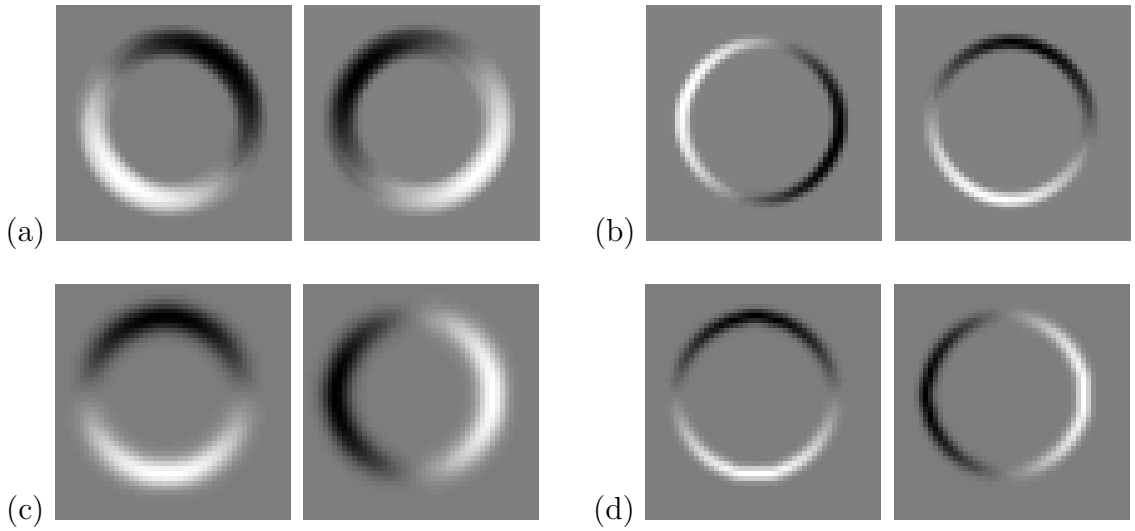

Figure 4.2: *Tangent plane basis vectors of the translating disk IAM estimated: using local PCA at (a) scale $\epsilon = 1/4$ and (b) scale $\epsilon = 1/8$; using image regularization at (c) scale $s = 1/8$ and (d) scale $s = 1/16$.*

4.2.3 Approximate tangent planes via regularization

The lack of IAM differentiability poses an apparent difficulty for image processing: the geometric relationship among images nearby in articulation space seems to be quite complicated. In addition to illuminating this challenge, however, the local PCA experiments in Section 4.2.2 also suggest a way out. Namely, the "twisting off" phenomenon can be understood as the existence of an intrinsic *multiscale structure* to the manifold. Tangent planes, instead of being associated with a location only, as in traditional monoscale analysis, are now associated with a location and a scale.

For a variety of reasons, it is convenient in formalizing this notion to work with a different notion of approximate tangent plane. We first define the *family of regularized manifolds* as follows. Associated with a given IAM, we have a family of *regularization operators* G_s that act on functions $f \in \mathcal{F}$ to smooth them; the parameter $s > 0$ is a scale parameter. For example, for the translated disk model, we let G_s be the operator of convolution with a Gaussian of standard deviation s: $G_s f = g_s * f$, where $g_s(x) = \frac{1}{2\pi s^2} \exp\{\frac{-\|x\|^2}{2s^2}\}$. We also define $f_{\theta,s} = G_s f_\theta$. The functions $f_{\theta,s}$ are smooth, and the collection of such functions for θ varying and $s > 0$ makes a manifold $\mathcal{F}_s$. The operator family $(G_s)_{s>0}$ has the property that, as we smooth less, we do less: $G_s f_\theta \to_{L^2} f_\theta,\ s \to 0$. It follows that, at least on compact subsets of $\mathcal{F}$,

$$\mathcal{F}_s \to_{L^2} \mathcal{F}, \quad s \to 0. \tag{4.2}$$

Because the regularized images contain no sharp edges, it follows that the regularized

IAMs are differentiable. We define the *approximate tangent plane* at scale $s > 0$, $T(s, \theta^{(0)}; \mathcal{F})$, to be the exact tangent plane of the approximate manifold $\mathcal{F}_s$; that is $T_{f_{\theta^{(0)},s}}(\mathcal{F}_s)$.

$T(s, \theta^{(0)})$ is the affine span of the functions $\frac{\partial}{\partial \theta_i} f_{\theta,s}\big|_{\theta=\theta^{(0)}}$, $i = 0, 1, \ldots, K-1$. This notion of approximate tangent plane is different from the more intrinsic local PCA approach but is far more amenable to analysis and computation. In practice, the two notions are similar: regularizing an image averages nearby pixel values, whereas local PCA analyzes a set of images related approximately by small shifts in space.

As an example, consider again the translating disk model. Figures 4.2(c),(d) show the tangent planes obtained from the image regularization process at scales $s = 1/8$ and $s = 1/16$. It is again visually evident that the tangent plane bases at the two scales are different, with behavior analogous to the bases obtained using the local PCA approach in Figures 4.2(a),(b). In this case, the angle between the two tangent planes is 26.4°.

4.2.4 Regularized tangent images

It is instructive to pursue an explicit description for the multiscale tangent images. We begin by deriving the regularized tangents for a restricted class of IAMs, where we have smooth articulations of an indicator set in the plane. This work follows closely certain computations in [16].

Let B denote an indicator set (for example, a disk), and let ∂B denote the boundary of B, which we assume to be C^2. For a point $b \in \partial B$, let $n(b)$ denote the outward-pointing normal vector to ∂B. The set $B = B_\theta$ may change as a function of θ, but we assume the evolution of ∂B_θ to be smooth. Thus we can attach to each boundary point $b \in \partial B_\theta$ a *motion vector* $v_i(b, \theta)$ that indicates the local direction in which the boundary shifts with respect to changes in component θ_i. For example, note that v_i is constant-valued when the articulations simply translate the set B.

From Lemma A.2 in [16], it follows that

$$\frac{\partial}{\partial \theta_i} f_{\theta,s}(x)\bigg|_{\theta=\theta^{(0)}} = \int_{\partial B} g_s(x-b)\, \sigma_i(b)\, db,$$

where $\sigma_i(b) := \langle v_i(b, \theta^{(0)}), n(b) \rangle$ measures the amount of shift in the direction normal to the edge. This can be rewritten as the convolution of the regularization kernel g_s with a *Schwartz distribution* $\gamma_i(x)$. This distribution can be understood as a 1-D ridge of delta functions around the boundary ∂B with "height" $\sigma_i(p)$ for $p \in \partial B$ (and height zero elsewhere). Indeed, this distribution also corresponds to the limiting "tangent image" on the unregularized manifold $\mathcal{F}$. We have essentially justified the last link in this chain of equalities

$$\frac{\partial}{\partial \theta_i} f_{\theta,s} = \frac{\partial}{\partial \theta_i}(g_s * f_\theta) = \left(g_s * \frac{\partial}{\partial \theta_i} f_\theta \right) = g_s * \gamma_i. \tag{4.3}$$

The problem, of course, is that $\gamma_i \notin L^2(\mathbb{R}^2)$, and so we rely on the regularization process. The formula (4.3) is one on which we may rely for general images f_θ — the regularized tangents can be obtained by convolving a Gaussian with the distributional tangent images.

4.3 Multiscale Twisting of IAMs

IAMs of images with sharp edges are non-differentiable, because their tangent planes continually "twist off" into new dimensions. In this section, we examine the multiscale structure of this phenomenon, for the example case of the translating disk IAM. First, we study the twisting phenomenon of the family of smoothed manifolds $\mathcal{F}_s$ as a function of scale s; next we examine twisting at a single scale as a function of position on the manifold. As we will discover, the multiscale characterization of the manifold is not unlike the wavelet analysis of non-differentiable functions.

4.3.1 Tangent bases for translating disk IAM

We can provide some quantitative values for regularized tangent images in the case of a translated disk. For technical reasons we let the image be the full plane $\mathbb{R}^2$ and also let $\Theta = \mathbb{R}^2$.

We start by identifying the boundary ∂B with the circle $[0, 2\pi)$ (we let $b = 0$ denote the rightmost point of B and traverse ∂B in the counterclockwise direction). For clarity, we write $\vec{b}$ when referring to the boundary point in $\mathbb{R}^2$ and write b when referring to the corresponding angle. For example, we have that $n(\vec{b}) = [\cos(b),\ \sin(b)]^T$. For translations we have simply that $v_0(\vec{b}) = [1,\ 0]^T$ and $v_1(\vec{b}) = [0,\ 1]^T$. This gives $\sigma_0(\vec{b}) = \cos(b)$ and $\sigma_1(\vec{b}) = \sin(b)$.

In order to examine the inter-scale twisting of the tangent planes, we use as a basis for the approximate tangent space $T(s, \theta^{(0)})$ the functions

$$\tau_s^i = \frac{\partial}{\partial \theta_i} f_{\theta,s}\bigg|_{\theta=\theta^{(0)}}.$$

The $L^2(\mathbb{R}^2)$ inner product between these tangent images is given by

$$\begin{aligned}
\langle \tau_s^i, \tau_s^j \rangle &= \langle g_s * \gamma_i, g_s * \gamma_j \rangle \\
&= \int_{\mathbb{R}^2} \int_{\partial B} g_s(x - \vec{b})\, \sigma_i(\vec{b})\ d\vec{b} \int_{\partial B} g_s(x - \vec{\beta})\, \sigma_j(\vec{\beta})\ d\vec{\beta}\ dx \\
&= \int_{\partial B} \int_{\partial B} \sigma_i(\vec{b})\, \sigma_j(\vec{\beta}) \int_{\mathbb{R}^2} g_s(x - \vec{b})\, g_s(x - \vec{\beta})\ dx\ d\vec{\beta}\ d\vec{b} \\
&= \int_{\partial B} \int_{\partial B} \sigma_i(\vec{b})\, \sigma_j(\vec{\beta})\, g_{\sqrt{2}s}(\vec{b} - \vec{\beta})\ d\vec{\beta}\ d\vec{b}.
\end{aligned}$$

The last step follows because the convolution of two Gaussians yields another Gaus-

sian; a similar derivation appears in Lemma A.3 of [16]. Considering the case where $i \neq j$, we have

$$\begin{aligned}
\langle \tau_s^0, \tau_s^1 \rangle &= \int_0^{2\pi} \int_0^{2\pi} \cos(b) \sin(\beta)\, g_{\sqrt{2}s}(\vec{b} - \vec{\beta})\; d\beta\; db \\
&= \int_{-\pi/2}^{2\pi-\pi/2} \int_{-\pi/2}^{2\pi-\pi/2} \cos(b + \pi/2)\; \sin(\beta + \pi/2)\, g_{\sqrt{2}s}(\vec{b} - \vec{\beta})\; d\beta\; db \\
&= -\int_0^{2\pi} \int_0^{2\pi} \sin(b)\; \cos(\beta)\, g_{\sqrt{2}s}(\vec{b} - \vec{\beta})\; d\beta\; db \\
&= -\langle \tau_s^1, \tau_s^0 \rangle,
\end{aligned}$$

which implies that $\langle \tau_s^0, \tau_s^1 \rangle = 0$. Thus we have that $\langle \tau_s^i, \tau_s^j \rangle = c_{s,s}\delta_{i,j}$, where, for generality useful below, we set

$$c_{s_0,s_1} := \int_{\partial B} \int_{\partial B} \cos(b)\; \cos(\beta)\, g_{\sqrt{s_0^2+s_1^2}}(\vec{b} - \vec{\beta})\; d\vec{\beta}\; d\vec{b}.$$

Hence, *the* $\{\tau_s^i\}$ *form an orthogonal basis for the approximate tangent plane* $T(s, \theta^{(0)})$ *for every* $s > 0$.

Consider now the bases $\{\tau_{s_0}^i\}_{i=0}^1$, $\{\tau_{s_1}^i\}_{i=0}^1$ at two different scales s_0 and s_1. Then by a similar calculation

$$\langle \tau_{s_0}^i, \tau_{s_1}^j \rangle = c_{s_0,s_1}\delta_{i,j}. \tag{4.4}$$

Hence, a basis element at one scale correlates with only one basis element at another scale.

4.3.2 Inter-scale twist angle

We can give (4.4) a geometric interpretation based on angles between subspaces. At each scale, define the new basis

$$\psi_s^i = c_{s,s}^{-1/2}\, \tau_s^i, \quad i = 0, 1,$$

which is an orthonormal basis for the approximate tangent space $T(s, \theta^{(0)})$. These bases are canonical for measuring the angles between any two tangent spaces. Formally, if we let P_s denote the linear orthogonal projection operator from $L^2(\mathbb{R}^2)$ onto $T(s, \theta^{(0)})$, then the subspace correlation operator $\Gamma_{s_0,s_1} = P_{s_0} P_{s_1}$ has a singular value decomposition using the two bases as left and right singular systems, respectively:

$$\Gamma_{s_0,s_1} = \sum_{i=0}^{1} \psi_{s_0}^i \lambda^i \langle \psi_{s_1}^i,\; \cdot \rangle;$$

or, in an informal but obvious notation,

$$\Gamma_{s_0,s_1} = [\psi^0_{s_0};\ \psi^1_{s_0}]\,\mathrm{diag}(\lambda^0,\lambda^1)\,[\psi^0_{s_1};\ \psi^1_{s_1}]^T.$$

The diagonal entries are given by

$$\lambda^i_{s_0,s_1} = \frac{c_{s_0,s_1}}{c^{1/2}_{s_1,s_1} c^{1/2}_{s_0,s_0}}.$$

Now from the theory of angles between subspaces [117, 118], we have that the angles between the subspaces $T(s_0,\theta^{(0)})$ and $T(s_1,\theta^{(0)})$ are naturally expressed as $\cos(\text{angle } \#i) = \lambda^i_{s_0,s_1},\ i = 0, 1$. In this instance, $\lambda^0 = \lambda^1$, and so we write simply

$$\cos(\text{angle}\{T(s_0,\theta^{(0)}), T(s_1,\theta^{(0)})\}) = \frac{c_{s_0,s_1}}{c^{1/2}_{s_0,s_0}\, c^{1/2}_{s_1,s_1}}.$$

We can perform a simple asymptotic analysis of the c_{s_0,s_1}.

Theorem 4.1 *In the translating disk model, let the regularization kernel g_s be a Gaussian with standard deviation $s > 0$. Fix $0 < \alpha < 1$ and let $s_1 = \alpha s_0$. Then*

$$\lim_{s_0 \to 0} \cos(\text{angle}\{T(s_0,\theta^{(0)}), T(s_1,\theta^{(0)})\}) = \sqrt{\frac{2\alpha}{\alpha^2+1}}. \tag{4.5}$$

Proof: See [19, Appendix].

This analytical result is fully in line with the results found in Sections 4.2.2 and 4.2.3 by empirically calculating angles between subspaces (for the case $\alpha = 1/2$, the formula predicts an angle of 26.6°).

4.3.3 Intra-scale twist angle

We can also examine the twisting phenomenon of the smoothed IAM $\mathcal{F}_s$ at a single scale s as a function of position on the manifold.

A simple experiment reveals the basic effect. We choose $\delta > 0$ and set $\Delta = [\delta;\ 0]^T$. We then compute

$$\text{angle } \{T(s,\theta), T(s,\theta + s\Delta)\}$$

at a variety of scales s. Figure 4.3 shows the experimental results for 256×256 images; tangents are estimated using a local difference between two synthesized images. This experiment reveals the following effects. First, and not surprisingly, larger changes in θ cause a larger twist in the tangent spaces. Second, and more surprisingly, the twist angle is approximately constant across scale when the change in θ is proportional to the scale. This behavior can also be confirmed analytically following the techniques of Section 4.3.1, though the analysis is a bit more complicated.

This experiment pertains to images over the unbounded domain $\mathbb{R}^2$. In case of a bounded domain, the disk will ultimately experience occlusion at the boundary of the image. In this region of occlusion, we have found that the twisting of the manifold $\mathcal{F}_s$ will depend not only on δ, but also more strongly on s and θ, unlike the experiment above.

4.3.4 Sampling

Through the process of regularization, we have defined a continuous multiscale characterization of an IAM tangent space. It is interesting, however, to consider the problem of sampling the multiscale tangent space while still preserving its essential structure. For example, we may be interested in answering the following question: "How finely must we sample in scale s at a fixed $\theta^{(0)}$ so that, between adjacent scales, the manifold twists no more than ρ degrees?" Similarly, "How finely must we sample in θ at a fixed scale s so that, between adjacent samples, the manifold twists no more than ρ degrees?" (For example, the success of our parameter estimation algorithm in Section 4.5 will depend on similar questions.)

From Theorem 4.1, it follows that by choosing a sequence

$$s_i = \alpha^i s_0, \quad i = 1, 2, \ldots$$

with an appropriate $\alpha < 1$, we can ensure that the tangent planes at adjacent scales change by no more than a fixed angle. Within a fixed scale, as we have seen in Section 4.3.3, to obtain a constant angle of twist, the amount of shift should be proportional to the smoothing scale s_i. These "sampling rules" for the multiscale tangent space are reminiscent of the sampling of the continuous wavelet transform to obtain the discrete wavelet transform (a case where $\alpha = 1/2$). Just as a non-differentiable function can be characterized with a multiresolution analysis, the translated disk IAM can be characterized by a multiresolution analysis having a similar scale-space structure. This basic behavior is common among a range of IAM models, though the precise details will vary. For use in an algorithm, additional analytic or experimental investigation may be necessary.

4.4 Non-Differentiability from Edge Occlusion

The first type of non-differentiability, as discussed in Sections 4.2 and 4.3, arises due to the migration of sharp edges. This non-differentiability is global, occurring at every point on the manifold. A second type of non-differentiability, however, can also arise on IAMs. This effect is *local*, occurring at only particular articulations where the tangents (even the regularized tangents) experience a sudden change.

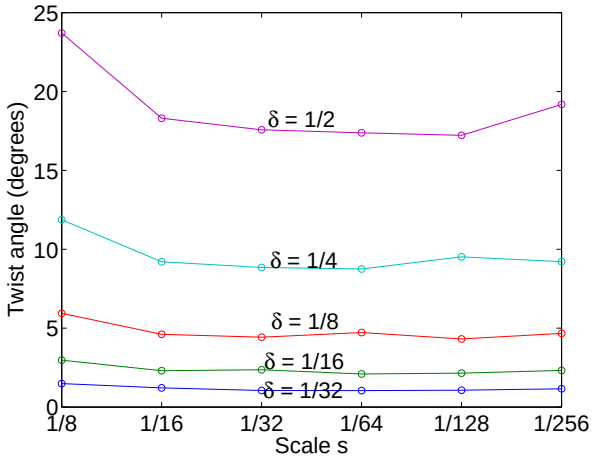

Figure 4.3: *Intra-scale twist angles for translating disk.*

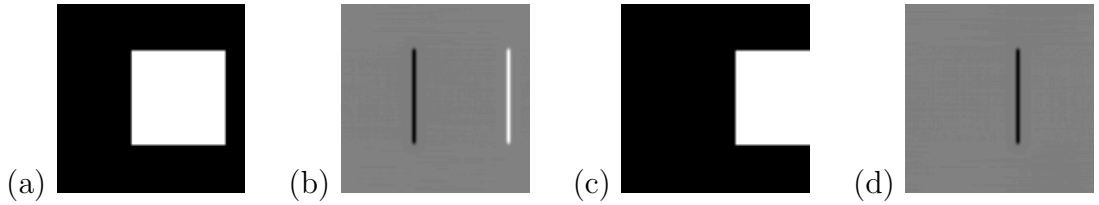

Figure 4.4: *Changing tangent images for translating square before and after occlusion. Pre-occlusion: (a) image and (b) tangent. Post-occlusion: (c) image and (d) tangent.*

4.4.1 Articulations in the image plane

To illustrate the basic effect, consider a simple translating-square image model. We assume a bounded image domain of $[-1,1]\times[-1,1]$; the occlusion of the square at the image border is the critical effect. The square indicator function has sidelength 1 and is centered at $\theta = (\theta_0, \theta_1)$. We will fix $\theta_1 = 0$ and examine the effects of changing component θ_0.

For the non-occluded regime, where $-1/2 < \theta_0 < 1/2$, it is easy to visualize the tangent images: γ_0 consists of two traveling ridges of delta functions, one with height -1 connecting the points $(\theta_0 - 1/2, \pm 1/2)$, and one with height 1 connecting the points $(\theta_0 + 1/2, \pm 1/2)$. These delta ridges are convolved with g_s to obtain the regularized tangent image (see Figure 4.4(a),(b)).

Consider now the occluded regime, for example $1/2 < \theta_0 < 3/2$. In this case, a portion of the square has been eliminated by the image boundary. We can equate the changing image with a *rectangle* sitting against the right side of the image, with width shrinking from the left. In this case γ_0 consists of only *one* traveling delta ridge, having height -1 and connecting the points $(\theta_0 - 1/2, \pm 1/2)$. Again, this ridge is convolved with g_s to obtain the regularized tangent image (see Figure 4.4(c),(d)).

This change in the tangent images is *abrupt*, occurring at precisely $\theta = [1/2,\ 0]^T$. Around this point, the manifold has differing "left" and "right" tangent images. It is simple to compute for this case that, as $s \to \infty$, then at the "moment of occlusion", there is an abrupt 45° change in tangent direction on each regularized manifold. This effect is not merely an artifact of the regularization process; a local PCA approximation would also be sensitive to the direction in which points are sampled.

This example demonstrates that, aside from the global issues of non-differentiability, IAMs may have localized cusps that persist even after regularization. These cusps indicate that the geometric structure relating nearby images can undergo a sudden change.

4.4.2 3-D articulations

Occlusion-based non-differentiability is much more natural in the 3-D case and occurs when an object self-occludes and a new edge appears in view. One example is a 3-D cube viewed face-on and then rotated in some direction. Other examples include polygonal solids, cylinders (when viewed from the end), and so on.

We use two numerical experiments to illustrate this phenomenon. For these experiments, we consider a 3-D cube viewed head-on and examine the tangent space around this point under SO(3) articulations (roll, pitch, yaw) at a fixed scale. For simplicity, we assume an imaging model where the 3-D object has a parallel projection onto the image plane, and we assume that the face of the cube displays a different color/intensity than the sides.

In the first experiment, we compute local tangent approximations on the regularized manifold. We assume θ parametrizes (roll, pitch, yaw) about the face-on appearance $f_{\theta^{(0)}}$. Around $\theta^{(0)}$, we perturb each articulation parameter individually by $+\epsilon$ or $-\epsilon$ and compute the difference relative to the original image (then divide by $\pm\epsilon$ and normalize). The six resulting tangent images are shown in Figure 4.5(a). The leftmost two images are almost identical, suggesting that the tangent space is smooth in the roll variable. The next two images differ significantly from one another, as do the last two. Thus with respect to the pitch and yaw parameters, the "left" and "right" tangents apparently differ. Following the arguments in Section 4.4.1, it is easy to understand what causes this discrepancy. For example, when the cube pitches forward, the image shows two moving edges at the bottom, and one at the top. Yet when the cube pitches back, the reverse is true.

In the second experiment, we perform a local PCA approximation to the manifold. We sample points randomly from the 3-D parameter space and run PCA on the resulting regularized images. Figure 4.5(a) shows a plot of the singular values. This plot suggests that most of the local energy is captured in a 5-D subspace.

These experiments indicate that, at the point where the cube is viewed head-on, we are at a cusp in the IAM with 5 relevant tangent directions — the manifold has a 5-D tangent complex [52] at this point. Clearly, this happens only at a small subset of all possible views of the cube (when only one face is visible). Similar effects (when

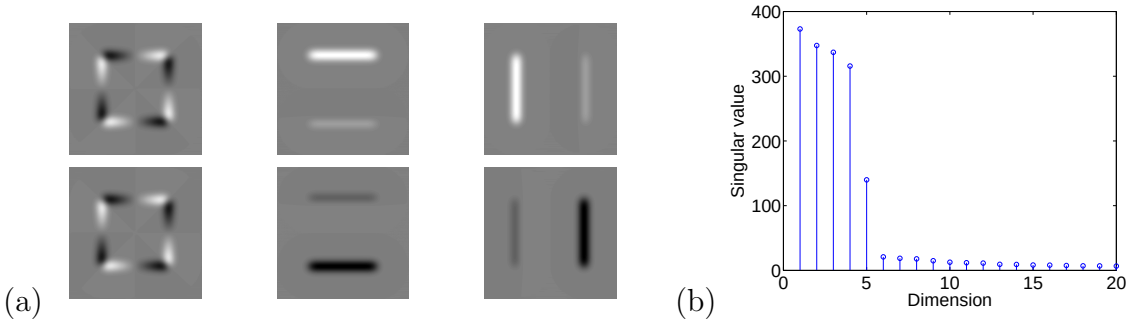

Figure 4.5: *(a) Tangent images for head-on view of a cube in 3-D space. Left: roll (vectors are similar). Middle: pitch (vectors are different). Right: yaw (vectors are different). (b) PCA on regularized cube images; first 20 singular values are shown.*

only two faces are visible) give rise to 4-D tangent complexes. Otherwise, for purely *generic* views of the cube (where three faces are visible), the tangent space has only 3 dimensions, corresponding to the 3 dimensions of Θ. This typical behavior echoes the assumption of "generic view" that is common in models of visual perception [119]: in order to understand a scene, an observer might assume a view to not be accidental (such as seeing a cube face-on).

4.5 Application: High-Resolution Parameter Estimation

With the multiscale viewpoint as background, we now consider the problem of inferring the articulation parameters from individual images. We will see that while the lack of differentiability prevents the application of conventional techniques, the multiscale perspective offers a way out. This perspective offers new justification for similar multiscale approaches employed in techniques such as image registration.

4.5.1 The problem

Let us recall the setup for parameter estimation from Section 2.5.3. Suppose $\mathcal{F} = \{f_\theta : \theta \in \Theta\}$ is an IAM and that we are given a signal I that is believed to approximate f_θ for an unknown $\theta \in \Theta$. From I we wish to recover an estimate of θ. We may formulate this parameter estimation problem as an optimization, writing the objective function (again we concentrate solely on the L_2 or ℓ_2 case)

$$D(\theta) = \|f_\theta - I\|_2^2$$

and solving for

$$\theta^* = \arg\min_{\theta \in \Theta} D(\theta).$$

We suppose that the minimum is uniquely defined. Supposing that $\mathcal{F}$ is a *differentiable* manifold, we could employ Newton's method to solve the minimization using an iterative procedure, where each iteration would involve projecting onto tangent images (as well as second derivative images).

In our setting, however, the tangent vectors τ_θ^i *do not exist* as functions, making it impossible to directly implement such an algorithm. We turn again to the regularization process in order to remedy this situation.

4.5.2 Multiscale Newton algorithm

As discussed in Section 4.2, the lack of differentiability can be alleviated by regularizing the images f_θ. Thus, navigation is possible on any of the regularized manifolds $\mathcal{F}_s$ using Newton's method as described above. This fact, in conjunction with the convergence property (4.2), suggests a multiscale technique for parameter estimation. Note that we focus on dealing with "migration-based" non-differentiability from Section 4.2. In cases where we have occasional occlusion-based non-differentiability as in Section 4.4, it may be necessary to project onto additional tangent images; this adaptation is not difficult, but it does require an awareness of the parameters at which occlusion-based non-differentiability occurs.

The idea is to select a sequence of scales $s_0 > s_1 > \cdots > s_{k_{\max}}$, and to start with an initial guess $\theta^{(0)}$. At each scale we take a Newton-like step on the corresponding smoothed manifold. We find it helpful in practice to ignore the second derivative term from equation (2.7). This is in the typical spirit of making slight changes to Newton's Method; in fact it is similar to the Gauss-Newton method for minimizing D.

To be specific, iteration $k+1$ of the Multiscale Newton algorithm proceeds as follows:

1. Compute the local tangent vectors on the smoothed manifold $\mathcal{F}_{s_k}$ at the point $f_{\theta^{(k)},s_k}$:
$$\tau^i_{\theta^{(k)},s_k} = \frac{\partial}{\partial\theta_i} f_{\theta,s_k}\bigg|_{\theta=\theta^{(k)}}, \quad i = 0, 1, \ldots, K-1.$$

2. Project the estimation error $f_{\theta^{(k)},s_k} - I_{s_k}$ (relative to the *regularized* image $I_{s_k} = g_{s_k} * I$) onto the tangent space $T(s_k, \theta^{(k)})$, setting
$$J_i = 2\langle f_{\theta^{(k)},s_k} - I_{s_k}, \tau^i_{\theta^{(k)},s_k}\rangle.$$

3. Compute the pairwise inner products between tangent vectors
$$H_{ij} = 2\langle \tau^i_{\theta^{(k)},s_k}, \tau^j_{\theta^{(k)},s_k}\rangle.$$

Table 4.1: *Estimation errors of Multiscale Newton iterations, translating disk, no noise.*

s	θ_0 error	θ_1 error	image MSE
Initial	-1.53e-01	1.92e-01	9.75e-02
1/2	-2.98e-02	5.59e-02	3.05e-02
1/4	-4.50e-04	1.39e-03	1.95e-04
1/16	-1.08e-06	8.62e-07	8.29e-10
1/256	1.53e-08	1.55e-07	1.01e-10

Table 4.2: *Estimation errors of Multiscale Newton iterations, translating disk, with noise. MSE between noisy image and true disk* $= 3.996$.

s	θ_0 error	θ_1 error	image MSE
Initial	-1.53e-01	1.93e-01	4.092
1/2	-3.46e-02	7.40e-02	4.033
1/4	-1.45e-02	-2.61e-03	4.003
1/16	-1.55e-03	-1.77e-03	3.997
1/256	-5.22e-04	1.10e-03	3.996

4. Use the projection coefficients to update the estimate

$$\theta^{(k+1)} \leftarrow \theta^{(k)} + H^{-1}J.$$

We note that when the tangent vectors are orthogonal to one another, H is diagonal, and so the update for component $\theta_i^{(k)}$ is simply determined by the inner product of the estimation error vector and the tangent vector $\tau^i_{\theta^{(k)},s_k}$. Moreover, when the regularized manifold $\mathcal{F}_{s_k}$ is linear in the range of interest, the update in Step 4 immediately achieves the minimizer to D at that scale.

Under certain conditions on the accuracy of the initial guess and the sequence $\{s_k\}$ it can be shown that this algorithm provides estimation accuracy $\|\theta - \theta^{(k)}\| < cs_k^2$. Ideally, we would be able to square the scale between successive iterations, $s_{k+1} = s_k^2$. The exact sequence of steps, and the accuracy required of the initial guess $\theta^{(0)}$, will depend on the specific multiscale structure of the IAM under consideration. We omit the convergence analysis in this thesis, instead providing several examples to demonstrate the basic effectiveness of the algorithm.

Table 4.3: *Estimation errors after Multiscale Newton iterations, ellipse.*

s	θ_0 error	θ_1 error	θ_2 error	θ_3 error	θ_4 error	image MSE
Initial	-5.75e-02	3.95e-02	-8.16e+00	7.72e-02	-3.56e-02	8.47e-02
1/2	5.82e-02	1.48e-02	7.91e-01	-3.66e-02	-8.74e-03	3.62e-02
1/4	-4.86e-03	-1.56e-03	-4.14e+00	3.19e-02	-1.28e-02	1.91e-02
1/16	4.25e-04	1.99e-04	-7.95e-01	-2.64e-03	-1.05e-03	1.42e-03
1/256	-3.61e-05	2.71e-05	-3.38e-03	-1.49e-04	-3.86e-05	2.72e-06

Table 4.4: *Estimation errors after Multiscale Newton iterations, 3-D icosahedron. MSE between noisy image and true original* = 2.98.

s	θ_0 error	θ_1 error	θ_2 error	θ_3 error	θ_4 error	θ_5 error	MSE
Initial	-50	-23	20	1.00e-1	-1.00e-1	5.00e-1	3.13
1/2	-8.81e+1	1.53e+0	6.07e+1	-2.60e-2	5.00e-2	-3.28e-1	3.14
1/4	-5.29e+1	4.70e+0	2.44e+1	3.44e-3	2.42e-2	4.24e-2	3.10
1/8	-1.15e+1	1.12e+0	-9.44e-1	-4.34e-3	3.19e-3	1.26e-1	3.03
1/16	8.93e-1	3.00e-1	-1.69e+0	-1.38e-3	2.21e-3	3.40e-2	2.98
1/256	5.28e-1	2.57e-1	-6.68e-1	6.91e-4	2.44e-3	2.12e-2	2.98

4.5.3 Examples

Translating disk

As a basic exercise of the proposed algorithm, we attempt to estimate the articulation parameters for a translated disk. The process is illustrated in Figure 4.6. The observed image I is shown on the far left; the top-left image in the grid is the initial guess $f_{\theta^{(0)}}$. For this experiment, we create 256×256 images with "subpixel" accuracy (each pixel is assigned a value based on the proportion of its support that overlaps the disk). Regularized tangent images are estimated using a local difference of synthesized (and regularized) images.

We run the multiscale estimation algorithm using the sequence of stepsizes $s = 1/2, 1/4, 1/16, 1/256$. Figure 4.6 shows the basic computations of each iteration. Note the geometric significance of the smoothed difference images $I_{s_k} - f_{\theta^{(k)},s_k}$; at each scale this image is projected onto the tangent plane basis vectors. Table 4.1 gives the estimation errors at each iteration, both for the articulation parameters θ and the mean square error (MSE) of the estimated image. Using this sequence of scales, we observe rapid convergence to the correct articulation parameters with accuracy far better than the width of a pixel, $1/256 \approx 3.91\text{e-}03$.

We now run a similar experiment for the case where the observation $I = f_\theta + n$, where n consists of additive white Gaussian noise of variance 4. Using the same

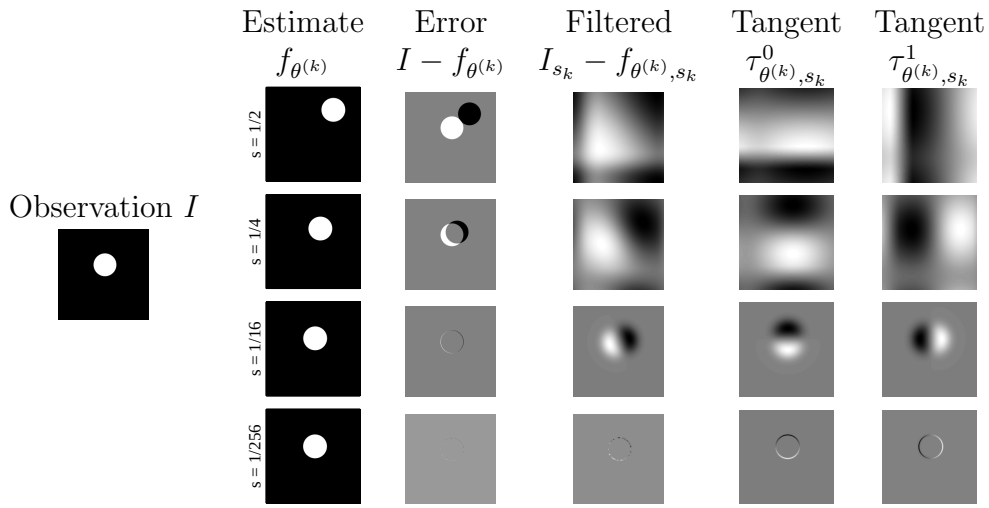

Figure 4.6: *Multiscale estimation of translation parameters for observed disk image. Each row corresponds to the smoothing and tangent basis vectors for one iteration.*

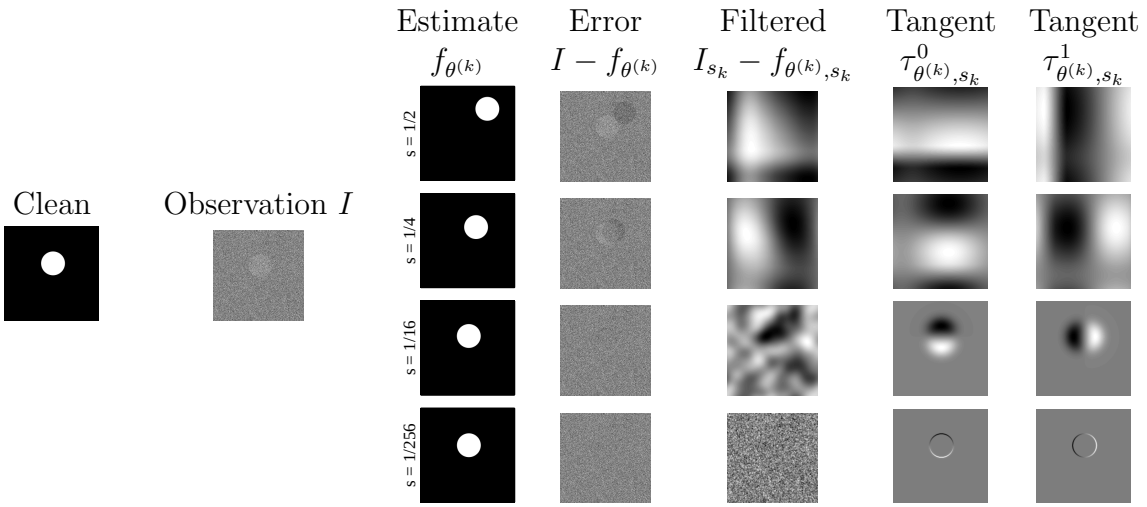

Figure 4.7: *Multiscale estimation of translation parameters for observed disk image with noise.*

sequence of smoothing filter sizes, the results are shown in Figure 4.7 and in Table 4.2. Note that the estimated articulation parameters are approximately the best possible, since the resulting MSE is approximately equal to the noise energy.

Articulated ellipse

We run a similar experiment for an ellipse image. In this case, the parameter space Θ is 5-D, with two directions of translation, one rotation parameter, and two parameters for the axis lengths of the ellipse. Figure 4.8 and Table 4.3 show the estimation results. It is particularly interesting to examine the geometric structure

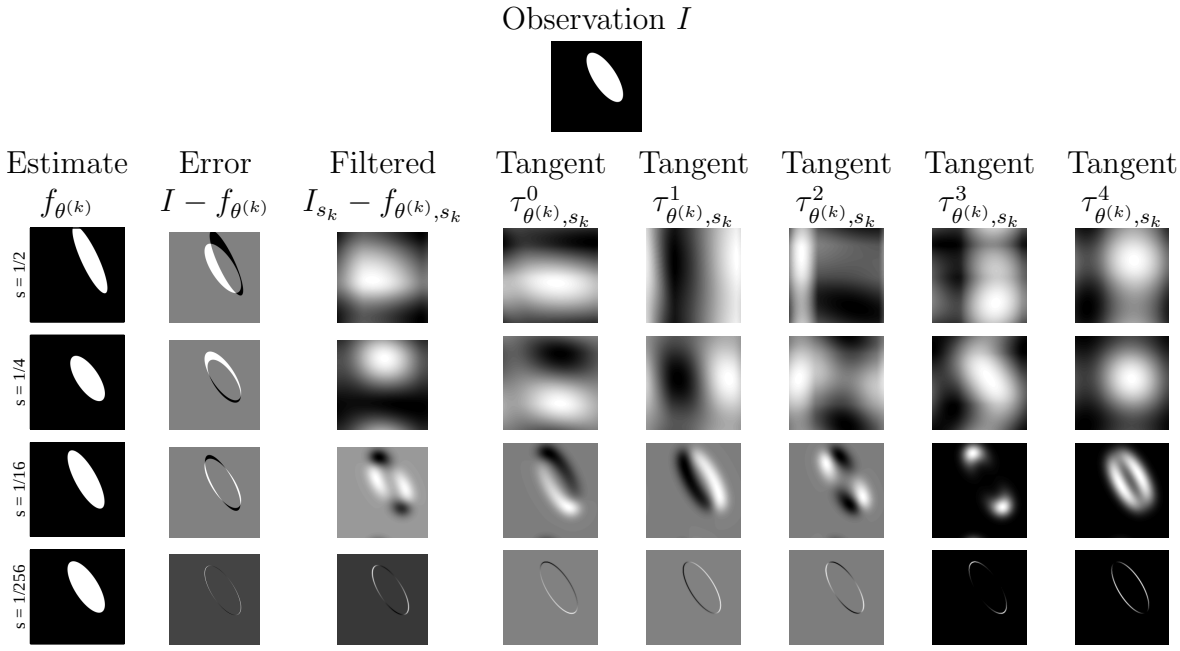

Figure 4.8: *Multiscale estimation of articulation parameters for ellipse.*

among the tangent vectors and how it reflects the different effects of changing the articulation parameters. Again, the algorithm successfully estimates the articulation parameters with high accuracy.

3-D articulations

We now consider a different imaging modality, where we have articulations of a 3-D object. In this case, the parameter space Θ is 6-D; the articulations of the object are three rotational coordinates, two shift coordinates parallel to the image plane, and one shift toward/away from the camera. (We now use a pinhole imaging model, so motions toward the camera make the object appear larger.)

For this example, we consider synthesized photographs of an icosahedron. Our image model includes a directional light source (with location and intensity parameters assumed known). We consider color images, treating each image as an element of $\mathbb{R}^{256\times256\times3}$. Figure 4.9 and Table 4.4 show the successful estimation of the articulation parameters for a noisy image. For this example, we must use a slightly less ambitious sequence of smoothing filters. (In this case, while we successfully ignore the occlusion-based effects of the appearance/disappearance of faces, we find that these should not be ignored in general.)

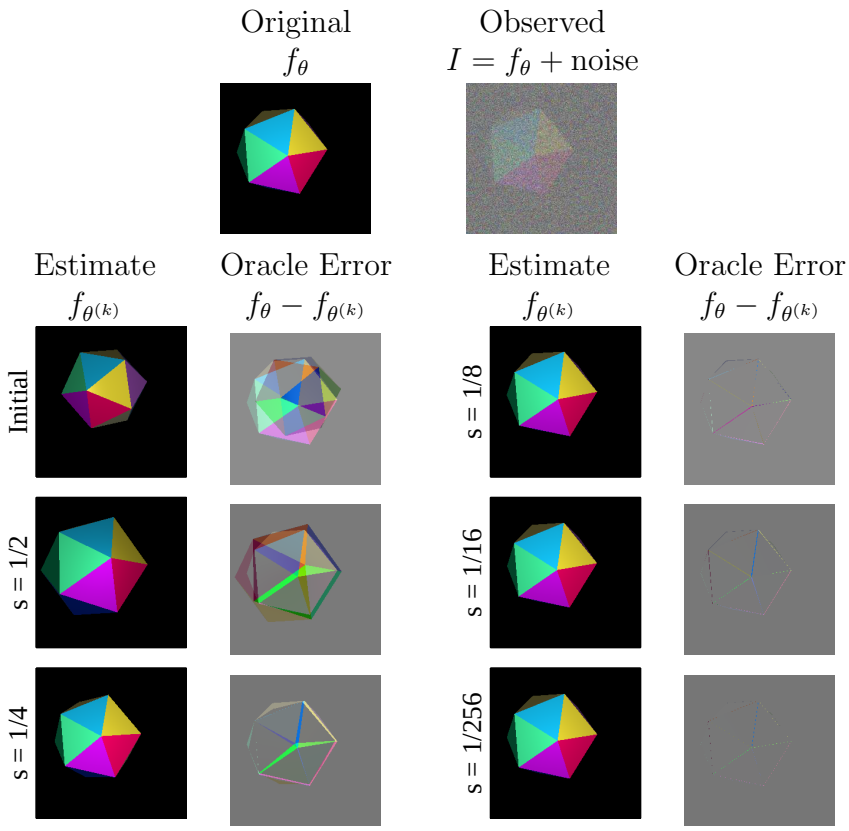

Figure 4.9: *Multiscale estimation of articulation parameters for 3-D icosahedron.*

4.5.4 Related work

Our multiscale framework for estimation with IAMs shares common features with a number of practical image registration algorithms; space considerations permit discussion of only a few here. Irani and Peleg [113] have developed a popular multiscale algorithm for registering an image $I(x)$ with a translated and rotated version for the purposes of super-resolution. They employ a multiscale pyramid to speed up the algorithm and to improve accuracy, but a clear connection is not made with the non-differentiability of the corresponding IAM. While Irani and Peleg compute the tangent basis images with respect to the x_0 and x_1 axes of the image, Keller and Averbach [120] compute them with respect to changes in each of the registration parameters. They also use a multiscale pyramid and conduct a thorough convergence analysis. Belhumeur [116] develops a tangent-based algorithm that estimates not only the pose of a 3-D object, but also its illumination parameters.

Where we differ from these approaches is in deriving the multiscale approach from the structure of the underlying manifold and in explaining the properties of the algorithm (e.g., how quickly the scale can be decreased) in terms of the twisting of the tangent space. More importantly, our approach is general and in principle extends far beyond the registration setting to many other image understanding problems of

learning parameters from example images. We discuss possible extensions of our estimation algorithm in Chapter 7.

Chapter 5
Joint Sparsity Models for Multi-Signal Compressed Sensing

In previous chapters we have looked for new insight and opportunities to be gained by considering a parametric (manifold-based) framework as a model for concise, low-dimensional signal structure. In this chapter,[1] we consider another novel modeling perspective, as we turn our attention toward a suite of signal models designed for simultaneous modeling of *multiple* signals that have a shared concise structure.

As a primary motivation and application area for these models, we consider the extension of Compressed Sensing to the multi-signal environment. At present, the CS theory and methods are tailored for the sensing of a single sparse signal. However, many of the attractive features of CS (with its simple, robust, and universal encoding) make it well-suited to remote sensing environments. In many cases involving remote sensing, however, the data of interest does not consist of a single signal but may instead be comprised of multiple signals, each one measured by a node in a network of low-cost, wireless sensors [122, 123]. As these sensors are often battery-operated, reducing power consumption (especially in communication) is essential. Because the sensors presumably observe related phenomena, however, we can anticipate that there will be some sort of inter-signal structure shared by the sensors in addition to the traditional intra-signal structure observed at a given sensor.

If we suppose that all of the sensors intend to transmit their data to a central collection node, one potential method to reduce communication costs would be for the sensors to collaborate (communicating among themselves) to discover and exploit their shared structure and to jointly encode their data. A number of distributed coding algorithms have been developed that involve collaboration amongst the sensors [124, 125]. Any collaboration, however, involves some amount of inter-sensor communication overhead. The *Slepian-Wolf* framework for lossless distributed coding [126–128] offers a collaboration-free approach in which each sensor node could communicate losslessly at its conditional entropy rate, rather than at its individual entropy rate. Unfortunately, however, most existing coding algorithms [127, 128] exploit only inter-signal correlations and not intra-signal correlations, and there has been only limited progress on distributed coding of so-called "sources with memory." In certain cases, however — in particular when each signal obeys a sparse model

[1] This work is in collaboration with Dror Baron, Marco Duarte, Shriram Sarvotham, and Richard Baraniuk [121].

and the sparsities among the signals are somehow related — we believe that CS can provide a viable solution to the distributed coding problem.

In this chapter, we introduce a new theory for *Distributed Compressed Sensing* (DCS) that enables new distributed coding algorithms that exploit both intra- and inter-signal correlation structures. In a typical DCS scenario, a number of sensors measure signals that are each individually sparse in some basis and also correlated from sensor to sensor. Each sensor *independently* encodes its signal by projecting it onto another, incoherent basis (such as a random one) and then transmits just a few of the resulting coefficients to a single collection point. Under the right conditions, a decoder at the collection point (presumably equipped with more computational resources than the individual sensors) can reconstruct each of the signals precisely.

The DCS theory rests on a concept that we term the *joint sparsity* of a signal ensemble. We study in detail three simple models for jointly sparse signals, propose tractable algorithms for joint recovery of signal ensembles from incoherent projections, and characterize theoretically and empirically the number of measurements per sensor required for accurate reconstruction. While the sensors operate entirely without collaboration, our simulations reveal that in practice the savings in the total number of required measurements can be substantial over separate CS decoding, especially when a majority of the sparsity is shared among the signals.

This chapter is organized as follows. Section 5.1 introduces our three models for joint sparsity: JSM-1, 2, and 3. We provide our detailed analysis and simulation results for these models in Sections 5.2, 5.3, and 5.4, respectively.

5.1 Joint Sparsity Models

In this section, we generalize the notion of a signal being sparse in some basis to the notion of an ensemble of signals being *jointly sparse*. In total, we consider three different *joint sparsity models* (JSMs) that apply in different situations. In the first two models, each signal is itself sparse, and so we could use the CS framework from Section 2.8 to encode and decode each one separately (independently). However, there also exists a framework wherein a *joint representation* for the ensemble uses fewer total vectors. In the third model, no signal is itself sparse, yet there still exists a joint sparsity among the signals that allows recovery from significantly fewer measurements per sensor.

We will use the following notation in this chapter for signal ensembles and our measurement model. Denote the *signals* in the ensemble by x_j, $j \in \{1, 2, \ldots, J\}$, and assume that each signal $x_j \in \mathbb{R}^N$. We use $x_j(n)$ to denote sample n in signal j, and we assume that there exists a known *sparse basis* Ψ for $\mathbb{R}^N$ in which the x_j can be sparsely represented. The coefficients of this sparse representation can take arbitrary real values (both positive and negative). Denote by Φ_j the *measurement matrix* for signal j; Φ_j is $M_j \times N$ and, in general, the entries of Φ_j are different for each j. Thus,

$y_j = \Phi_j x_j$ consists of $M_j < N$ *incoherent measurements* of x_j.[2] We will emphasize random i.i.d. Gaussian matrices Φ_j in the following, but other schemes are possible, including random ± 1 Bernoulli/Rademacher matrices, and so on.

In previous chapters, we discussed signals with intra-signal correlation (within each x_j) or signals with inter-signal correlation (between x_{j_1} and x_{j_2}). The three following models sport both kinds of correlation simultaneously.

5.1.1 JSM-1: Sparse common component + innovations

In this model, all signals share a *common* sparse component while each individual signal contains a sparse *innovation* component; that is,

$$x_j = z_C + z_j, \quad j \in \{1, 2, \ldots, J\}$$

with

$$z_C = \Psi \alpha_C, \quad \|\alpha_C\|_0 = K_C \qquad \text{and} \qquad z_j = \Psi \alpha_j, \quad \|\alpha_j\|_0 = K_j.$$

Thus, the signal z_C is common to all of the x_j and has sparsity K_C in basis Ψ. The signals z_j are the unique portions of the x_j and have sparsity K_j in the same basis. Denote by Ω_C the support set of the nonzero α_C values and by Ω_j the support set of α_j.

A practical situation well-modeled by JSM-1 is a group of sensors measuring temperatures at a number of outdoor locations throughout the day. The temperature readings x_j have both temporal (intra-signal) and spatial (inter-signal) correlations. Global factors, such as the sun and prevailing winds, could have an effect z_C that is both common to all sensors and structured enough to permit sparse representation. More local factors, such as shade, water, or animals, could contribute localized innovations z_j that are also structured (and hence sparse). A similar scenario could be imagined for a network of sensors recording light intensities, air pressure, or other phenomena. All of these scenarios correspond to measuring properties of physical processes that change smoothly in time and in space and thus are highly correlated.

5.1.2 JSM-2: Common sparse supports

In this model, all signals are constructed from the same sparse set of basis vectors, but with different coefficients; that is,

$$x_j = \Psi \alpha_j, \quad j \in \{1, 2, \ldots, J\}, \tag{5.1}$$

where each α_j is nonzero only on the common coefficient set $\Omega \subset \{1, 2, \ldots, N\}$ with $|\Omega| = K$. Hence, all signals have ℓ_0 sparsity of K, and all are constructed from the

[2]Note that the measurements at sensor j can be obtained either indirectly by sampling the signal x_j and then computing the matrix-vector product $y_j = \Phi_j x_j$ or directly by special-purpose hardware that computes y_j without first sampling (see [33], for example).

same K basis elements but with arbitrarily different coefficients.

A practical situation well-modeled by JSM-2 is where multiple sensors acquire replicas of the same Fourier-sparse signal but with phase shifts and attenuations caused by signal propagation. In many cases it is critical to recover each one of the sensed signals, such as in many acoustic localization and array processing algorithms. Another useful application for JSM-2 is MIMO communication [129].

Similar signal models have been considered by different authors in the area of *simultaneous sparse approximation* [129–131]. In this setting, a collection of sparse signals share the same expansion vectors from a redundant dictionary. The sparse approximation can be recovered via greedy algorithms such as *Simultaneous Orthogonal Matching Pursuit* (SOMP) [129, 130] or *MMV Order Recursive Matching Pursuit* (M-ORMP) [131]. We use the SOMP algorithm in our setting (see Section 5.3) to recover from incoherent measurements an ensemble of signals sharing a common sparse structure.

5.1.3 JSM-3: Nonsparse common component + sparse innovations

This model extends JSM-1 so that the common component need no longer be sparse in any basis; that is,

$$x_j = z_C + z_j, \quad j \in \{1, 2, \ldots, J\}$$

with

$$z_C = \Psi\alpha_C \qquad \text{and} \qquad z_j = \Psi\alpha_j, \quad \|\alpha_j\|_0 = K_j,$$

but z_C is not necessarily sparse in the basis Ψ. We also consider the case where the supports of the innovations are shared for all signals, which extends JSM-2. Note that separate CS reconstruction cannot be applied under JSM-3, since the common component is not sparse.

A practical situation well-modeled by JSM-3 is where several sources are recorded by different sensors together with a background signal that is not sparse in any basis. Consider, for example, an idealized computer vision-based verification system in a device production plant. Cameras acquire snapshots of components in the production line; a computer system then checks for failures in the devices for quality control purposes. While each image could be extremely complicated, the ensemble of images will be highly correlated, since each camera is observing the same device with minor (sparse) variations.

JSM-3 could also be useful in some non-distributed scenarios. For example, it motivates the compression of data such as video, where the innovations or differences between video frames may be sparse, even though a single frame may not be very sparse. In this case, JSM-3 suggests that we encode each video frame independently using CS and then decode all frames of the video sequence jointly. This has the advantage of moving the bulk of the computational complexity to the video decoder. Puri and Ramchandran have proposed a similar scheme based on Wyner-Ziv dis-

tributed encoding in their PRISM system [132]. In general, JSM-3 may be invoked for ensembles with significant inter-signal correlations but insignificant intra-signal correlations.

5.1.4 Refinements and extensions

Each of the JSMs proposes a basic framework for joint sparsity among an ensemble of signals. These models are intentionally *generic*; we have not, for example, mentioned the processes by which the index sets and coefficients are assigned. In subsequent sections, to give ourselves a firm footing for analysis, we will often consider specific *stochastic* generative models, in which (for example) the nonzero indices are distributed uniformly at random and the nonzero coefficients are drawn from a random Gaussian distribution. While some of our specific analytical results rely on these assumptions, the basic algorithms we propose should generalize to a wide variety of settings that resemble the JSM-1, 2, and 3 models.

It should also be clear that there are many possible joint sparsity models beyond the three we have introduced. One immediate extension is a combination of JSM-1 and JSM-2, where the signals share a common set of sparse basis vectors but with different expansion coefficients (as in JSM-2) plus additional innovation components (as in JSM-1). For example, consider a number of sensors acquiring different delayed versions of a signal that has a sparse representation in a multiscale basis such as a wavelet basis. The acquired signals will share the same wavelet coefficient support at coarse scales with different values, while the supports at each sensor will be different for coefficients at finer scales. Thus, the coarse scale coefficients can be modeled as the common support component, and the fine scale coefficients can be modeled as the innovation components.

Further work in this area will yield new JSMs suitable for other application scenarios. Applications that could benefit include multiple cameras taking digital photos of a common scene from various angles [133]. Additional extensions are discussed in Chapter 7.

5.2 Recovery Strategies for Sparse Common Component + Innovations Model (JSM-1)

For this model, in order to characterize the measurement rates M_j required to jointly reconstruct the signals x_j, we have proposed an analytical framework inspired by principles of information theory and parallels with the Slepian-Wolf theory. This section gives a basic overview of the ideas and results; we refer the reader to [121] for the full details.

Our information theoretic perspective allows us to formalize the following intuition. Consider the simple case of $J = 2$ signals. By employing the CS machinery, we might expect that (i) $(K_C + K_1)c$ coefficients suffice to reconstruct x_1, (ii) $(K_C + K_2)c$

coefficients suffice to reconstruct x_2, yet only (iii) $(K_C + K_1 + K_2)c$ coefficients should suffice to reconstruct both x_1 and x_2, because we have $K_C + K_1 + K_2$ nonzero elements in x_1 and x_2. In addition, given the $(K_C + K_1)c$ measurements for x_1 as side information, and assuming that the partitioning of x_1 into z and z_1 is known, cK_2 measurements that describe z_2 should allow reconstruction of x_2. (In this sense, we may view K_2 as a "conditional sparsity," in parallel with the notion of conditional entropy.)

Formalizing these arguments allows us to establish theoretical lower bounds on the required measurement rates at each sensor. Like the single-signal CS problem, these measurement rates depend also on the reconstruction scheme. For example, suppose we formulate the recovery problem using matrices and vectors as

$$z \triangleq \begin{bmatrix} z_C \\ z_1 \\ z_2 \end{bmatrix}, \quad x \triangleq \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad y \triangleq \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \quad \Phi \triangleq \begin{bmatrix} \Phi_1 & 0 \\ 0 & \Phi_2 \end{bmatrix}, \tag{5.2}$$

and supposing that $\Psi = I_N$, we can define

$$\widetilde{\Psi} \triangleq \begin{bmatrix} \Psi & \Psi & 0 \\ \Psi & 0 & \Psi \end{bmatrix} \tag{5.3}$$

and write $x = \widetilde{\Psi} z$.

Now, we consider the following reconstruction algorithm that minimizes the total ℓ_0 sparsity among all feasible solutions

$$\widehat{z} = \arg\min \|z_C\|_0 + \|z_1\|_0 + \|z_2\|_0 \quad \text{s.t. } y = \Phi\widetilde{\Psi} z. \tag{5.4}$$

We have proved that, for this algorithm to succeed, it is necessary and sufficient for each measurement rate M_j to be at least one greater than the conditional sparsity K_j and for the total measurement rate $\sum_j M_j$ be at least one greater than the total sparsity $K_C + \sum_j K_j$; such bounds are clear analogues of the Slepian-Wolf theory. In fact, these are lower bounds for *any* reconstruction algorithm to succeed. (This is only the basic idea, and certain technical details must also be considered; see [121, Section 4].) For more tractable recovery algorithms, we establish similar lower bounds on the measurement rates required for ℓ_1 recovery, and we also establish upper bounds on the required measurement rates M_j by proposing a specific algorithm for reconstruction. The algorithm uses carefully designed measurement matrices Φ_j (in which some rows are identical and some differ) so that the resulting measurements can be combined to allow step-by-step recovery of the sparse components.

Figure 5.1 shows such our bounds for the case of $J = 2$ signals, with signal lengths $N = 1000$ and sparsities $K_C = 200$, $K_1 = K_2 = 50$. We see that the theoretical rates M_j are below those required for separable CS recovery of each signal x_j. Our numerical simulations (involving a slightly customized ℓ_1 algorithm where the sparsity of z_C is penalized by a factor γ_C) confirm the potential savings. As

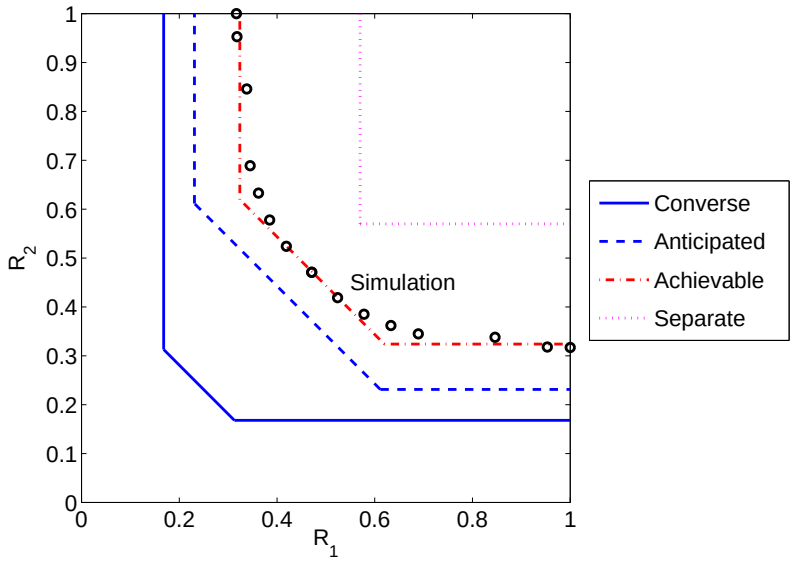

Figure 5.1: *Converse bounds and achievable measurement rates for $J = 2$ signals with common sparse component and sparse innovations (JSM-1). The measurement rates $R_j := M_j/N$ reflect the number of measurements normalized by the signal length. The pink curve denotes the rates required for separable CS signal reconstruction.*

demonstrated in Figure 5.2, the degree to which joint decoding outperforms separate decoding is directly related to the amount of shared information K_C. For $K_C = 11$, $K_1 = K_2 = 2$, M is reduced by approximately 30%. For smaller K_C, joint decoding barely outperforms separate decoding.

5.3 Recovery Strategies for Common Sparse Supports Model (JSM-2)

Under the JSM-2 signal ensemble model from Section 5.1.2, separate recovery of each signal via ℓ_0 minimization would require $K + 1$ measurements per signal, while separate recovery via ℓ_1 minimization would require cK measurements per signal. As we now demonstrate, the total number of measurements can be reduced substantially by employing specially tailored joint reconstruction algorithms that exploit the common structure among the signals, in particular the common coefficient support set Ω.

The algorithms we propose are inspired by conventional greedy pursuit algorithms for CS (such as OMP [91]). In the single-signal case, OMP iteratively constructs the sparse support set Ω; decisions are based on inner products between the columns of $\Phi\Psi$ and a residual. In the multi-signal case, there are more clues available for determining the elements of Ω.

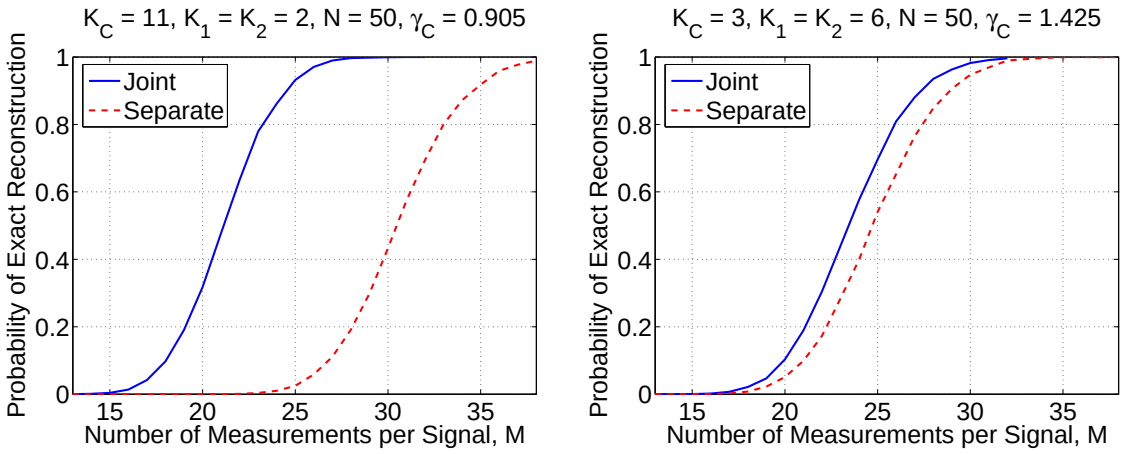

Figure 5.2: *Reconstructing a signal ensemble with common sparse component and sparse innovations (JSM-1). We plot the probability of perfect joint reconstruction (solid lines) and independent CS reconstruction (dashed lines) as a function of the number of measurements per signal M. The advantage of using joint instead of separate reconstruction depends on the common sparsity.*

5.3.1 Recovery via Trivial Pursuit

When there are many correlated signals in the ensemble, a simple non-iterative greedy algorithm based on inner products will suffice to recover the signals jointly. For simplicity but without loss of generality, we again assume that $\Psi = I_N$ and that an equal number of measurements $M_j = M$ are taken of each signal. We write Φ_j in terms of its columns, with $\Phi_j = [\phi_{j,1}, \phi_{j,2}, \ldots, \phi_{j,N}]$.

Trivial Pursuit (TP) Algorithm for JSM-2

1. **Get greedy:** Given all of the measurements, compute the test statistics

$$\xi_n = \frac{1}{J} \sum_{j=1}^{J} \langle y_j, \phi_{j,n} \rangle^2, \qquad n \in \{1, 2, \ldots, N\} \tag{5.5}$$

and estimate the elements of the common coefficient support set by

$$\widehat{\Omega} = \{n \text{ having one of the } K \text{ largest } \xi_n\}.$$

When the sparse, nonzero coefficients are sufficiently generic (as defined below), we have the following surprising result.

Theorem 5.1 *Let Ψ be an orthonormal basis for $\mathbb{R}^N$, let the measurement matrices Φ_j contain i.i.d. Gaussian entries, and assume that the nonzero coefficients in the α_j*

are i.i.d. Gaussian random variables. Then with $M \geq 1$ measurements per signal, TP recovers Ω with probability approaching one as $J \to \infty$.

Proof: See [121, Appendix G].

In words, with *fewer* than K measurements per sensor, it is possible to recover the sparse support set Ω under the JSM-2 model.[3] Of course, this approach does not recover the K coefficient values for each signal; K measurements per sensor are required for this.

Theorem 5.2 *Assume that the nonzero coefficients in the α_j are i.i.d. Gaussian random variables. Then the following statements hold:*

1. *Let the measurement matrices Φ_j contain i.i.d. Gaussian entries, with each matrix having an oversampling factor of $c = 1$ (that is, $M_j = K$ for each measurement matrix Φ_j). Then TP recovers all signals from the ensemble $\{x_j\}$ with probability approaching one as $J \to \infty$.*

2. *Let Φ_j be a measurement matrix with oversampling factor $c < 1$ (that is, $M_j < K$), for some $j \in \{1, 2, \ldots, J\}$. Then with probability one, the signal x_j cannot be uniquely recovered by any algorithm for any value of J.*

The first statement is an immediate corollary of Theorem 5.1; the second statement follows because each equation $y_j = \Phi_j x_j$ would be underdetermined even if the nonzero indices were known. Thus, under the JSM-2 model, the Trivial Pursuit algorithm asymptotically performs as well as an oracle decoder that has prior knowledge of the locations of the sparse coefficients. From an information theoretic perspective, Theorem 5.2 provides tight achievable and converse bounds for JSM-2 signals.

In a technical report [134], we derive an approximate formula for the probability of error in recovering the common support set Ω given J, N, K, and M. Figure 5.3 depicts the performance of the formula in comparison to simulation results. While theoretically interesting and potentially practically useful, these results require J to be large. Our numerical experiments show that TP works well even when M is small, as long as J is sufficiently large. However, in the case of fewer signals (small J), TP performs poorly. We propose next an alternative recovery technique based on simultaneous greedy pursuit that performs well for small J.

5.3.2 Recovery via iterative greedy pursuit

In practice, the common sparse support among the J signals enables a fast iterative algorithm to recover all of the signals jointly. Tropp and Gilbert have proposed

[3] One can also show the somewhat stronger result that, as long as $\sum_j M_j \gg N$, TP recovers Ω with probability approaching one. We have omitted this additional result for brevity.

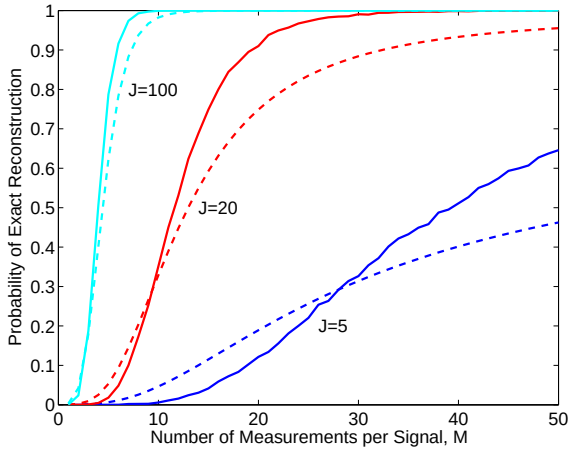

Figure 5.3: *Reconstruction using TP for JSM-2. Approximate formula (dashed lines) for the probability of error in recovering the support set Ω in JSM-2 using TP given J, N, K, and M [134] compared against simulation results (solid) for fixed $N = 50$, $K = 5$ and varying number of measurements M and number of signals $J = 5$, $J = 20$, and $J = 100$.*

one such algorithm, called *Simultaneous Orthogonal Matching Pursuit* (SOMP) [129], which can be readily applied in our DCS framework. SOMP is a variant of OMP that seeks to identify Ω one element at a time. (A similar simultaneous sparse approximation algorithm has been proposed using convex optimization; see [135] for details.) We dub the DCS-tailored SOMP algorithm DCS-SOMP.

To adapt the original SOMP algorithm to our setting, we first extend it to cover a different measurement basis Φ_j for each signal x_j. Then, in each DCS-SOMP iteration, we select the column index $n \in \{1, 2, \ldots, N\}$ that accounts for the greatest amount of residual energy across *all* signals. As in SOMP, we orthogonalize the remaining columns (in each measurement basis) after each step; after convergence we obtain an expansion of the measurement vector on an orthogonalized subset of the holographic basis vectors. To obtain the expansion coefficients in the sparse basis, we then reverse the orthogonalization process using the QR matrix factorization. We assume without loss of generality that $\Psi = I_N$.

DCS-SOMP Algorithm for JSM-2

1. **Initialize:** Set the iteration counter $\ell = 1$. For each signal index $j \in \{1, 2, \ldots, J\}$, initialize the orthogonalized coefficient vectors $\widehat{\beta}_j = 0$, $\widehat{\beta}_j \in \mathbb{R}^M$; also initialize the set of selected indices $\widehat{\Omega} = \emptyset$. Let $r_{j,\ell}$ denote the residual of the measurement y_j remaining after the first ℓ iterations, and initialize $r_{j,0} = y_j$.

2. **Select** the dictionary vector that maximizes the value of the sum of the magnitudes of the projections of the residual, and add its index to the set of selected

indices

$$\begin{aligned} n_\ell &= \arg\max_{n=1,2,\ldots,N} \sum_{j=1}^{J} \frac{|\langle r_{j,\ell-1}, \phi_{j,n}\rangle|}{\|\phi_{j,n}\|_2}, \\ \widehat{\Omega} &= [\widehat{\Omega} \;\; n_\ell]. \end{aligned}$$

3. **Orthogonalize** the selected basis vector against the orthogonalized set of previously selected dictionary vectors

$$\gamma_{j,\ell} = \phi_{j,n_\ell} - \sum_{t=0}^{\ell-1} \frac{\langle \phi_{j,n_\ell}, \gamma_{j,t}\rangle}{\|\gamma_{j,t}\|_2^2} \gamma_{j,t}.$$

4. **Iterate:** Update the estimate of the coefficients for the selected vector and residuals

$$\begin{aligned} \widehat{\beta}_j(\ell) &= \frac{\langle r_{j,\ell-1}, \gamma_{j,\ell}\rangle}{\|\gamma_{j,\ell}\|_2^2}, \\ r_{j,\ell} &= r_{j,\ell-1} - \frac{\langle r_{j,\ell-1}, \gamma_{j,\ell}\rangle}{\|\gamma_{j,\ell}\|_2^2} \gamma_{j,\ell}. \end{aligned}$$

5. **Check for convergence:** If $\|r_{j,\ell}\|_2 > \epsilon \|y_j\|_2$ for all j, then increment ℓ and go to Step 2; otherwise, continue to Step 6. The parameter ϵ determines the target error power level allowed for algorithm convergence. Note that due to Step 3 the algorithm can only run for up to M iterations.

6. **De-orthogonalize:** Consider the relationship between $\Gamma_j = [\gamma_{j,1}, \gamma_{j,2}, \ldots, \gamma_{j,M}]$ and the Φ_j given by the QR factorization

$$\Phi_{j,\widehat{\Omega}} = \Gamma_j R_j,$$

where $\Phi_{j,\widehat{\Omega}} = [\phi_{j,n_1}, \phi_{j,n_2}, \ldots, \phi_{j,n_M}]$ is the so-called *mutilated basis*.[4] Since $y_j = \Gamma_j \beta_j = \Phi_{j,\widehat{\Omega}} x_{j,\widehat{\Omega}} = \Gamma_j R_j x_{j,\widehat{\Omega}}$, where $x_{j,\widehat{\Omega}}$ is the mutilated coefficient vector, we can compute the signal estimates $\{\widehat{x}_j\}$ as

$$\begin{aligned} \widehat{\alpha}_{j,\widehat{\Omega}} &= R_j^{-1} \widehat{\beta}_j, \\ \widehat{x}_j &= \Psi \widehat{\alpha}_j, \end{aligned}$$

where $\widehat{\alpha}_{j,\widehat{\Omega}}$ is the mutilated version of the sparse coefficient vector $\widehat{\alpha}_j$.

[4]We define a *mutilated basis* Φ_Ω as a subset of the basis vectors from $\Phi = [\phi_1, \phi_2, \ldots, \phi_N]$ corresponding to the indices given by the set $\Omega = \{n_1, n_2, \ldots, n_M\}$, that is, $\Phi_\Omega = [\phi_{n_1}, \phi_{n_2}, \ldots, \phi_{n_M}]$. This concept can be extended to vectors in the same manner.

In practice, each sensor projects its signal x_j via $\Phi_j x_j$ to produce $\widehat{c}K$ measurements for some $\widehat{c}$. The decoder then applies DCS-SOMP to reconstruct the J signals jointly. We orthogonalize because as the number of iterations approaches M the norms of the residues of an orthogonal pursuit decrease faster than for a non-orthogonal pursuit.

Thanks to the common sparsity structure among the signals, we believe (but have not proved) that DCS-SOMP will succeed with $\widehat{c} < c(S)$. Empirically, we have observed that a small number of measurements proportional to K suffices for a moderate number of sensors J. We conjecture that $K + 1$ measurements per sensor suffice as $J \to \infty$; numerical experiments are presented in Section 5.3.3. Thus, in practice, this efficient greedy algorithm enables an oversampling factor $\widehat{c} = (K+1)/K$ that approaches 1 as J, K, and N increase.

5.3.3 Simulations for JSM-2

We now present a simulation comparing separate CS reconstruction versus joint DCS-SOMP reconstruction for a JSM-2 signal ensemble. Figure 5.4 plots the probability of perfect reconstruction corresponding to various numbers of measurements M as the number of sensors varies from $J = 1$ to 32. We fix the signal lengths at $N = 50$ and the sparsity of each signal to $K = 5$.

With DCS-SOMP, for perfect reconstruction of all signals the average number of measurements per signal decreases as a function of J. The trend suggests that, for very large J, close to K measurements per signal should suffice. On the contrary, with separate CS reconstruction, for perfect reconstruction of all signals the number of measurements per sensor *increases* as a function of J. This surprise is due to the fact that each signal will experience an independent probability $p \leq 1$ of successful reconstruction; therefore the overall probability of complete success is p^J. Consequently, each sensor must compensate by making additional measurements. This phenomenon further motivates joint reconstruction under JSM-2.

Finally, we note that we can use algorithms other than DCS-SOMP to recover the signals under the JSM-2 model. Cotter et al. [131] have proposed additional algorithms (such as the M-FOCUSS algorithm) that iteratively eliminate basis vectors from the dictionary and converge to the set of sparse basis vectors over which the signals are supported. We hope to extend such algorithms to JSM-2 in future work.

5.4 Recovery Strategies for Nonsparse Common Component + Sparse Innovations Model (JSM-3)

The JSM-3 signal ensemble model from Section 5.1.3 provides a particularly compelling motivation for joint recovery. Under this model, no individual signal x_j is sparse, and so recovery of each signal separately would require fully N measurements per signal. As in the other JSMs, however, the commonality among the signals makes it possible to substantially reduce this number.

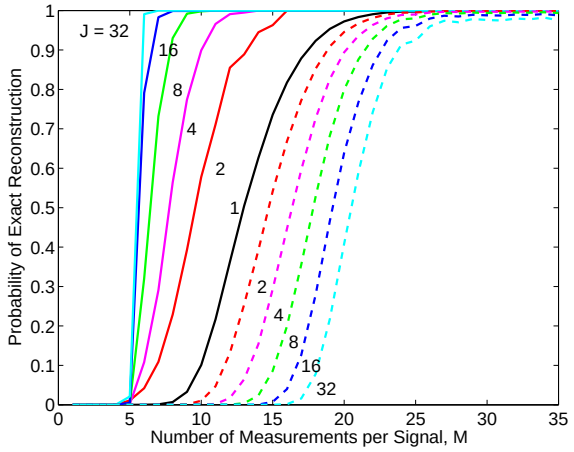

Figure 5.4: *Reconstructing a signal ensemble with common sparse supports (JSM-2). We plot the probability of perfect reconstruction via DCS-SOMP (solid lines) and independent CS reconstruction (dashed lines) as a function of the number of measurements per signal M and the number of signals J. We fix the signal length to $N = 50$, the sparsity to $K = 5$, and average over 1000 simulation runs. An oracle encoder that knows the positions of the large signal expansion coefficients would use 5 measurements per signal.*

5.4.1 Recovery via Transpose Estimation of Common Component

Successful recovery of the signal ensemble $\{x_j\}$ requires recovery of both the non-sparse common component z_C and the sparse innovations $\{z_j\}$. To illustrate the potential for signal recovery using far fewer than N measurements per sensor, consider the following gedankenexperiment. Again, for simplicity but without loss of generality, we assume $\Psi = I_N$.

If z_C were known, then each innovation z_j could be estimated using the standard single-signal CS machinery on the adjusted measurements

$$y_j - \Phi_j z_C = \Phi_j z_j.$$

While z_C is not known in advance, it can be *estimated* from the measurements. In fact, across all J sensors, a total of $\sum_j M_j$ random projections of z_C are observed (each corrupted by a contribution from one of the z_j). Since z_C is not sparse, it cannot be recovered via CS techniques, but when the number of measurements is sufficiently large ($\sum_j M_j \gg N$), z_C can be estimated using standard tools from linear algebra. A key requirement for such a method to succeed in recovering z_C is that each Φ_j be different, so that their rows combine to span all of $\mathbb{R}^N$. In the limit (again, assuming the sparse innovation coefficients are well-behaved), the common component z_C can be recovered while still allowing each sensor to operate at the minimum measurement rate dictated by the $\{z_j\}$. A prototype algorithm is listed below, where we assume that each measurement matrix Φ_j has i.i.d. $\mathcal{N}(0, \sigma_j^2)$ entries.

TECC Algorithm for JSM-3

1. **Estimate common component:** Define the matrix $\widehat{\Phi}$ as the concatenation of the regularized individual measurement matrices $\widehat{\Phi}_j = \frac{1}{M_j \sigma_j^2} \Phi_j$, that is, $\widehat{\Phi} = [\widehat{\Phi}_1, \widehat{\Phi}_2, \ldots, \widehat{\Phi}_J]$. Calculate the estimate of the common component as $\widehat{z_C} = \frac{1}{J} \widehat{\Phi}^T y$.

2. **Estimate measurements generated by innovations:** Using the previous estimate, subtract the contribution of the common part on the measurements and generate estimates for the measurements caused by the innovations for each signal: $\widehat{y}_j = y_j - \Phi_j \widehat{z_C}$.

3. **Reconstruct innovations:** Using a standard single-signal CS reconstruction algorithm, obtain estimates of the innovations $\widehat{z}_j$ from the estimated innovation measurements $\widehat{y}_j$.

4. **Obtain signal estimates:** Estimate each signal as the sum of the common and innovations estimates; that is, $\widehat{x}_j = \widehat{z_C} + \widehat{z}_j$.

The following theorem shows that asymptotically, by using the TECC algorithm, each sensor need only measure at the rate dictated by the sparsity K_j.

Theorem 5.3 *Assume that the nonzero expansion coefficients of the sparse innovations z_j are i.i.d. Gaussian random variables and that their locations are uniformly distributed on $\{1, 2, ..., N\}$. Then the following statements hold:*

1. *Let the measurement matrices Φ_j contain i.i.d. $\mathcal{N}(0, \sigma_j^2)$ entries with $M_j \geq K_j + 1$. Then each signal x_j can be recovered using the TECC algorithm with probability approaching one as $J \to \infty$.*

2. *Let Φ_j be a measurement matrix with $M_j \leq K_j$ for some $j \in \{1, 2, ..., J\}$. Then with probability one, the signal x_j cannot be uniquely recovered by any algorithm for any value of J.*

Proof: See Appendix B.

For large J, the measurement rates permitted by Statement 1 are the lowest possible for *any* reconstruction strategy on JSM-3 signals, even neglecting the presence of the nonsparse component. Thus, Theorem 5.3 provides a tight achievable and converse for JSM-3 signals. The CS technique employed in Theorem 5.3 involves combinatorial searches for estimating the innovation components. More efficient techniques could also be employed (including several proposed for CS in the presence of noise [23, 26, 29, 30, 80]). It is reasonable to expect similar behavior; as the error in estimating the common component diminishes, these techniques should perform similarly to their noiseless analogues (Basis Pursuit [26, 29], for example).

5.4.2 Recovery via Alternating Common and Innovation Estimation

The preceding analysis demonstrates that the number of required measurements in JSM-3 can be substantially reduced through joint recovery. While Theorem 5.3 suggests the theoretical gains as $J \to \infty$, practical gains can also be realized with a moderate number of sensors. For example, suppose in the TECC algorithm that the initial estimate $\widehat{z_C}$ is not accurate enough to enable correct identification of the sparse innovation supports $\{\Omega_j\}$. In such a case, it may still be possible for a rough approximation of the innovations $\{z_j\}$ to help refine the estimate $\widehat{z_C}$. This in turn could help to refine the estimates of the innovations. Since each component helps to estimate the other components, we propose an iterative algorithm for JSM-3 recovery.

The Alternating Common and Innovation Estimation (ACIE) algorithm exploits the observation that once the basis vectors comprising the innovation z_j have been identified in the index set Ω_j, their effect on the measurements y_j can be removed to aid in estimating z_C. Suppose that we have an estimate for these innovation basis vectors in $\widehat{\Omega}_j$. We can then partition the measurements into two parts: the projection into $\text{span}(\{\phi_{j,n}\}_{n\in\widehat{\Omega}_j})$ and the component orthogonal to that span. We build a basis for the $\mathbb{R}^{M_j}$ where y_j lives:

$$\mathbf{B}_j = [\Phi_{j,\widehat{\Omega}_j}\ Q_j],$$

where $\Phi_{j,\widehat{\Omega}_j}$ is the mutilated holographic basis corresponding to the indices in $\widehat{\Omega}_j$, and the $M_j \times (M_j - |\widehat{\Omega}_j|)$ matrix $Q_j = [q_{j,1}\ \ldots\ q_{j,M_j-|\widehat{\Omega}_j|}]$ has orthonormal columns that span the orthogonal complement of $\Phi_{j,\widehat{\Omega}_j}$.

This construction allows us to remove the projection of the measurements into the aforementioned span to obtain measurements caused exclusively by vectors not in $\widehat{\Omega}_j$

$$\widetilde{y}_j = {Q_j}^T y_j, \tag{5.6}$$
$$\widetilde{\Phi}_j = {Q_j}^T \Phi_j. \tag{5.7}$$

These modifications enable the sparse decomposition of the measurement, which now lives in $\mathbb{R}^{M_j-|\widehat{\Omega}_j|}$, to remain unchanged

$$\widetilde{y_j} = \sum_{n=1}^{N} a_j \widetilde{\phi}_{j,n}.$$

Thus, the modified measurements $\widetilde{Y} = \left[{\widetilde{y}_1}^T\ {\widetilde{y}_2}^T\ \ldots\ {\widetilde{y}_J}^T\right]^T$ and modified holographic basis $\widetilde{\Phi} = \left[{\widetilde{\Phi}_1}^T\ {\widetilde{\Phi}_2}^T\ \ldots\ {\widetilde{\Phi}_J}^T\right]^T$ can be used to refine the estimate of the measurements caused by the common part of the signal

$$\widetilde{z_C} = \widetilde{\Phi}^{\dagger}\widetilde{Y}, \tag{5.8}$$

where $A^{\dagger} = A^T(AA^T)^{-1}$ denotes the pseudoinverse of matrix A.

In the case where the innovation support estimate is correct ($\widehat{\Omega}_j = \Omega_j$), the measurements $\widetilde{y_j}$ will describe only the common component z_C. If this is true for every signal j and the number of remaining measurements $\sum_j M_j - KJ \geq N$, then z_C can be perfectly recovered via (5.8). However, it may be difficult to obtain correct estimates for all signal supports in the first iteration of the algorithm, and so we find it preferable to refine the estimate of the support by executing several iterations.

ACIE Algorithm for JSM-3

1. **Initialize:** Set $\widehat{\Omega}_j = \emptyset$ for each j. Set the iteration counter $\ell = 1$.

2. **Estimate common component:** Update estimate $\widetilde{z_C}$ according to (5.6)–(5.8).

3. **Estimate innovation supports:** For each sensor j, after subtracting the contribution $\widetilde{z_C}$ from the measurements, $\widehat{y}_j = y_j - \Phi_j \widetilde{z_C}$, estimate the sparse support of each signal innovation $\widehat{\Omega}_j$.

4. **Iterate:** If $\ell < L$, a preset number of iterations, then increment ℓ and return to Step 2. Otherwise proceed to Step 5.

5. **Estimate innovation coefficients:** For each signal j, estimate the coefficients for the indices in $\widehat{\Omega}_j$
$$\widehat{\alpha}_{j,\widehat{\Omega}_j} = \Phi^{\dagger}_{j,\widehat{\Omega}_j}(y_j - \Phi_j \widetilde{z_C}),$$
where $\widehat{\alpha}_{j,\widehat{\Omega}_j}$ is a mutilated version of the innovation's sparse coefficient vector estimate $\widehat{\alpha}_j$.

6. **Reconstruct signals:** Compute the estimate of each signal as $\widehat{x}_j = \widetilde{z_C} + \widehat{z}_j = \widetilde{z_C} + \Phi_j \widehat{\alpha}_j$.

Estimation of the sparse supports in Step 3 can be accomplished using a variety of techniques. We propose to run ℓ iterations of OMP; if the supports of the innovations are known to match across signals — as in the JSM-2 scenario — then more powerful algorithms like SOMP can be used.

5.4.3 Simulations for JSM-3

We now present simulations of JSM-3 reconstruction in the following scenario. Consider J signals of length $N = 50$ containing a common white noise component $z_C(n) \sim \mathcal{N}(0, 1)$ for $n \in \{1, 2, \ldots, N\}$ that, by definition, is not sparse in any fixed basis. Each innovations component z_j has sparsity $K = 5$ (once again in the time domain), resulting in $x_j = z_C + z_j$. The support for each innovations component is randomly selected with uniform probability from all possible supports for K-sparse,

length-N signals. We draw the values of the innovation coefficients from a unit-variance Gaussian distribution.

We study two different cases. The first is an extension of JSM-1: we select the supports for the various innovations independently and then apply OMP independently to each signal in Step 3 of the ACIE algorithm in order to estimate its innovations component. The second case is an extension of JSM-2: we select one common support for all of the innovations across the signals and then apply the DCS-SOMP algorithm from Section 5.3.2 to estimate the innovations in Step 3. In both cases we set $L = 10$. We test the algorithms for different numbers of signals J and calculate the probability of correct reconstruction as a function of the (same) number of measurements per signal M.

Figure 5.5(a) shows that, for sufficiently large J, we can recover all of the signals with significantly fewer than N measurements per signal. We note the following behavior in the graph. First, as J grows, it becomes more difficult to perfectly reconstruct all J signals. We believe this is inevitable, because even if z_C were known without error, then perfect ensemble recovery would require the successful execution of J *independent* runs of OMP. Second, for small J, the probability of success can decrease at high values of M. We believe this behavior is due to the fact that initial errors in estimating z_C may tend to be somewhat sparse (since $\widehat{z_C}$ roughly becomes an average of the signals $\{x_j\}$), and these sparse errors can mislead the subsequent OMP processes. For more moderate M, it seems that the errors in estimating z_C (though greater) tend to be less sparse. We expect that a more sophisticated algorithm could alleviate such a problem, and we note that the problem is also mitigated at higher J.

Figure 5.5(b) shows that when the sparse innovations share common supports we see an even greater savings. As a point of reference, a traditional approach to signal encoding would require 1600 total measurements to reconstruct these $J = 32$ nonsparse signals of length $N = 50$. Our approach requires only approximately 10 random measurements per sensor for a total of 320 measurements. In Chapter 7 we discuss possible extensions of the DCS framework to incorporate additional models and algorithms.

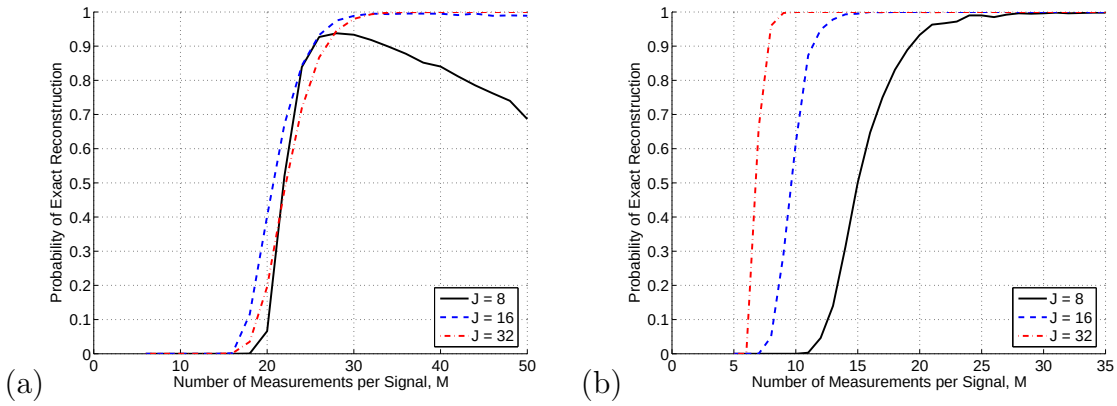

Figure 5.5: *Reconstructing a signal ensemble with nonsparse common component and sparse innovations (JSM-3) using ACIE. (a) Reconstruction using OMP independently on each signal in Step 3 of the ACIE algorithm (innovations have arbitrary supports). (b) Reconstruction using DCS-SOMP jointly on all signals in Step 3 of the ACIE algorithm (innovations have identical supports). Signal length $N = 50$, sparsity $K = 5$. The common structure exploited by DCS-SOMP enables dramatic savings in the number of measurements. We average over 1000 simulation runs.*

Chapter 6
Random Projections of Signal Manifolds

In this chapter,[1] inspired by a geometric perspective, we develop new theory and methods for problems involving random projections for dimensionality reduction. In particular, we consider embedding results previously applicable only to finite point clouds (the JL lemma) or to sparse signal models (Compressed Sensing) and generalize these results to include manifold-based signal models.

As our primary theoretical contribution (Theorem 6.2), we consider the effect of a random projection operator on a smooth K-dimensional submanifold of $\mathbb{R}^N$, establishing a sufficient number M of random projections to ensure a stable embedding of the manifold in $\mathbb{R}^M$. Like the fundamental bound in Compressed Sensing (CS), our requisite M is linear in the "information level" K and logarithmic in the ambient dimension N; additionally we identify a logarithmic dependence on the volume and curvature of the manifold. To establish the result, we use an effective finite "sampling" of the manifold (plus its tangent spaces) to capture its relevant structure and apply the JL lemma.

From a signal processing perspective, this result implies that small numbers of random measurements can capture a great deal of information about manifold-modeled signals. For example, random projections could be used to distinguish one signal from another on the same manifold. This is reminiscent of the CS problem, in which sparse signals can be distinguished from their random projections. This chapter takes the first steps in exploring and formalizing these connections and introducing a framework for manifold-driven CS recovery. As we demonstrate, manifold-modeled signals can *also* be recovered from random projections, where the number of required measurements is proportional to the manifold dimension, rather than the sparsity of the signal.

Our embedding result also implies that signal collections living along a manifold will have their basic neighborhood relationships preserved when projected to lower dimensions. This has promising implications in manifold learning, and we demonstrate that several standard techniques for learning manifold structure from sampled data can also be applied to random projections of that data.

This chapter is organized as follows. Section 6.1 examines theoretical issues concerning the embedding of signal manifolds under random projections. Section 6.2 discusses possible applications of random projections for manifold models in CS. Section 6.3 discusses additional applications in manifold learning.

[1] This work is in collaboration with Richard Baraniuk [136].

6.1 Manifold Embeddings under Random Projections

6.1.1 Inspiration — Whitney's Embedding Theorem

The theoretical inspiration for this work follows from Whitney's (Easy) Embedding Theorem.

Theorem 6.1 [61] *Let $\mathcal{M}$ be a compact Hausdorff C^r K-dimensional manifold, with $2 \leq r \leq \infty$. Then there is a C^r embedding of $\mathcal{M}$ in $\mathbb{R}^{2K+1}$.*

The proof of this theorem is highly insightful; it begins with an embedding of $\mathcal{M}$ in $\mathbb{R}^N$ for some large N and then considers the normalized secant set of the manifold

$$\Gamma = \left\{ \frac{x - x'}{\|x - x'\|_2} : x, x' \in \mathcal{M} \right\}.$$

Roughly speaking, the secant set forms a $2K$-dimensional subset of the $(N-1)$-dimensional unit sphere S^{N-1} (which equates with the space of projections from $\mathbb{R}^N$ to $\mathbb{R}^{N-1}$), and so there exists a projection from $\mathbb{R}^N$ to $\mathbb{R}^{N-1}$ that embeds $\mathcal{M}$ (without overlap). This can be repeated until reaching $\mathbb{R}^{2K+1}$. In signal processing, this secant set has been explicitly employed in order to find the optimal projection vectors for a given manifold (see [41,42], which also provide interesting and insightful discussions).

Our work will build upon the following useful observation: Using identical arguments and assuming mild conditions on the signal manifold $\mathcal{M}$ (ensuring that Γ has zero measure in S^{N-1}), it also follows that with high probability, a *randomly* chosen projection of the manifold from $\mathbb{R}^N$ to $\mathbb{R}^{2K+1}$ will be invertible.

6.1.2 Visualization

As an example, Figure 6.1 shows the random projection of two 1-D manifolds from $\mathbb{R}^N$ onto $\mathbb{R}^3$. In each case, distinct signals from the manifold remain separated in its embedding in $\mathbb{R}^3$. However, it is also clear that the differentiability of the manifold (related to the differentiability of the primitive function g in this example; see also Chapter 4) will play a critical role. We specifically account for the smoothness of the manifold in our embedding results in Section 6.1.4. (Indeed, while non-differentiable manifolds do not meet the criteria of Theorem 6.1, we will be interested in their projections as well. Section 6.2.5 discusses this issue in more detail.)

6.1.3 A geometric connection with Compressed Sensing

Our discussion of random projections and Whitney's Embedding Theorem has an immediate parallel with a basic result in CS. In particular, one may interpret statement two of Theorem 2.1 as follows: Let Σ_K be the set of all K-sparse signals in $\mathbb{R}^N$. With probability one, a random mapping $\Phi : \mathbb{R}^N \mapsto \mathbb{R}^M$ embeds Σ_K in $\mathbb{R}^M$. (Hence, no two K-sparse signals are mapped to the same point.)

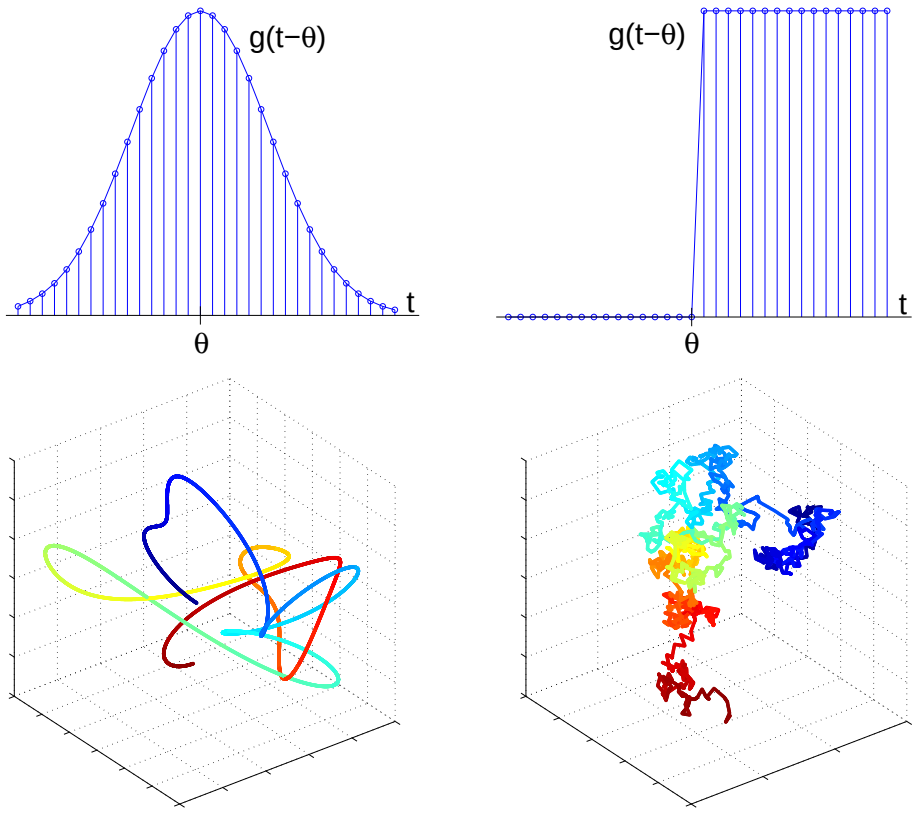

Figure 6.1: *Top row: The articulated signals $f_\theta(t) = g(t-\theta)$ are defined via shifts of a primitive function g, where g is (left) a Gaussian pulse or (right) a step function. Each signal is sampled at N points, and as θ changes, the resulting signals trace out 1-D manifolds in $\mathbb{R}^N$. Bottom row: Projection of manifolds from $\mathbb{R}^N$ onto random 3-D subspace; the color/shading represents different values of $\theta \in \mathbb{R}$.*

While we have already proved this statement in Appendix A, it can also be established using the arguments of Section 6.1.1: The signal set Σ_K consists of a union of K-dimensional hyperplanes. The secant set for Σ_K turns out to be a union of $2K$-dimensional hyperplanes (which loses 1 dimension after normalization). From this, it follows that with probability one, every length-N K-sparse signal can be recovered from just $2K$ random measurements (statement two of Theorem 2.1).

This connection to sparsity-based CS suggests that random projections may indeed be useful for capturing information about manifold-modeled signals as well. As discussed in Section 2.8.3, however, it is often necessary in sparsity-based CS to take more than $2K$ measurements in order to ensure tractable, robust recovery of sparse signals. The Restricted Isometry Property (RIP) gives one condition for such stability (see Section 2.8.6). Geometrically, the RIP can be interpreted as requiring not only that Σ_K embed in $\mathbb{R}^M$ but also that this embedding be "stable" in the sense

that K-sparse signals well separated in $\mathbb{R}^N$ remain well separated in $\mathbb{R}^M$. For similar reasons, we will desire such stability in embeddings of signal manifolds.

6.1.4 Stable embeddings

The following result establishes a sufficient number of random projections to ensure a stable embedding of a well-conditioned manifold. (Recall the terminology given in Sections 2.1.3 and 2.2.)

Theorem 6.2 *Let $\mathcal{M}$ be a compact K-dimensional submanifold of $\mathbb{R}^N$ having condition number $1/\tau$, volume V, and geodesic covering regularity R. Fix $0 < \epsilon < 1$ and $0 < \rho < 1$. Let Φ be a random orthoprojector from $\mathbb{R}^N$ to $\mathbb{R}^M$ with*

$$M = O\left(\frac{K \log(NVR\tau^{-1}\epsilon^{-1}) \log(1/\rho)}{\epsilon^2}\right). \tag{6.1}$$

If $M \leq N$, then with probability at least $1 - \rho$ the following statement holds: For every pair of points $x, y \in \mathcal{M}$,

$$(1-\epsilon)\sqrt{\frac{M}{N}} \leq \frac{\|\Phi x - \Phi y\|_2}{\|x - y\|_2} \leq (1+\epsilon)\sqrt{\frac{M}{N}}. \tag{6.2}$$

Proof: See Appendix C.

Theorem 6.2 concerns the preservation of pairwise ambient distances on the manifold; this can be immediately extended to geodesic distances as well.

Corollary 6.1 *Let $\mathcal{M}$ and Φ be as in Theorem 6.2. Assuming (6.2) holds for all pairs of points on $\mathcal{M}$, then for every pair of points $x, y \in \mathcal{M}$,*

$$(1-\epsilon)\sqrt{\frac{M}{N}} \leq \frac{d_{\Phi\mathcal{M}}(\Phi x, \Phi y)}{d_{\mathcal{M}}(x, y)} \leq (1+\epsilon)\sqrt{\frac{M}{N}}, \tag{6.3}$$

where $d_{\Phi\mathcal{M}}(\Phi x, \Phi y)$ denotes the geodesic distance between the projected points on the image of $\mathcal{M}$.

Proof: See Appendix D.

Before proceeding, we offer some brief remarks on these results.

1. Like the fundamental bound in Compressed Sensing, the requisite number of random projections M to ensure a stable embedding of the manifold is linear in the "information level" K and logarithmic in the ambient dimension N; additionally we identify a logarithmic dependence on the volume and curvature of the manifold.

2. The factor $\sqrt{\frac{M}{N}}$ is easily removed from (6.2) and (6.3) by simple rescaling of Φ.

3. The proof of Theorem 6.2 in fact establishes the bound (6.1) up to actual constants; see (C.10) for the complete result.

4. The $\ln(1/\rho)$ factor in the numerator of (6.1) and (C.10) can be immediately sharpened to

$$\frac{\ln(1/\rho)}{\ln\left(\frac{1900^{2K}K^{K/2}N^{3K/2}RV}{\epsilon^{3K}\tau^{K}}\right)}$$

to dramatically reduce the dependence on the failure probability ρ. (This follows simply from Lemma 2.5 and a more careful accounting in Section C.3 of the proof.)

5. The constant 200 appearing in (C.10) can likely be improved by increasing C_1 and using a more careful analysis in Section C.6.

6. One may also consider extending our results to allow Φ to be a random $M \times N$ matrix with i.i.d. $\mathcal{N}(0, \sigma^2)$ entries, where $\sigma^2 = 1/N$. In order to adapt the proof, one would need to account for the fact that Φ may no longer be nonexpanding; however with high probability the norm $\|\Phi\|_2$ can be bounded by a small constant.

6.2 Applications in Compressed Sensing

We argued in Section 6.1 that certain signal manifolds will have stable embeddings under random projections to low-dimensional spaces, and we drew parallels with the well-conditioned embedding of Σ_K that occurs in the typical CS setting. These parallels suggest that it may indeed be possible to extend the CS theory and methods to include manifold-based signal models.

To be specific, let us consider a length-N signal x that, rather than being K-sparse, we assume lives on or near some known K-dimensional manifold $\mathcal{M} \subset \mathbb{R}^N$. From a collection of measurements $y = \Phi x$, where Φ is a random $M \times N$ matrix, we would like to recover x. As with sparsity-driven CS, there are certain basic questions we must ask:

- How can x be recovered from y?
- How many measurements M are required?
- How stable is the recovery, and how accurately can x be recovered?

In this section, we provide preliminary theoretical insights into these issues and present a series of promising numerical experiments.

6.2.1 Methods for signal recovery

To discuss methods for recovering x from $y = \Phi x$ based on a manifold model $\mathcal{M} \subset \mathbb{R}^N$, we distinguish between the following two cases.

Case 1: $x \in \mathcal{M}$

In the first case, we assume that x lives precisely on the manifold $\mathcal{M}$ in $\mathbb{R}^N$. Assuming that Φ embeds $\mathcal{M}$ into $\mathbb{R}^M$, then, y will live precisely on the image $\Phi\mathcal{M}$ of $\mathcal{M}$ in $\mathbb{R}^M$, and there will exist a unique $\widehat{x} = x$ on $\mathcal{M}$ that can explain the measurements. The recovery problem then reduces to that of estimating the position of a signal on a manifold (in $\mathbb{R}^M$). For differentiable manifolds, methods for solving this problem were discussed in Section 2.5.3. (Even if $\mathcal{M}$ is not explicitly parametric, local parametrizations could be created for $\mathcal{M}$ in $\mathbb{R}^N$ that will translate to local parmaterizations for $\Phi\mathcal{M}$ in $\mathbb{R}^M$.) We defer the topic of recovery for non-differentiable manifolds, however, to Section 6.2.5.

Case 2: $x \notin \mathcal{M}$

A potentially more interesting scenario arises when the manifold is only an *approximation* for the signal class. Examples include edges that are not entirely straight or manifold-based signals corrupted by noise. In this second case, x may not live precisely on the manifold $\mathcal{M}$ in $\mathbb{R}^N$, and so its projection y may not live precisely on $\Phi\mathcal{M}$ in $\mathbb{R}^M$. We propose the following optimization problem as a method for estimating x:

$$\widehat{x} = \arg\min_{x' \in \mathcal{M}} \|y - \Phi x'\|_2 . \tag{6.4}$$

For differentiable manifolds, this problem may again be solved using the methods discussed in Section 2.5.3. Other recovery programs may also be considered, though one advantage of (6.4) is that $\widehat{x}$ itself will belong to the manifold $\mathcal{M}$.

6.2.2 Measurements

To answer the question of how many CS measurements we must take for a manifold-modeled signal, we again consider the two cases of Section 6.2.1. In the first case, when the signal obeys the manifold model precisely, then a unique, correct solution will exist as long as Φ embeds $\mathcal{M}$ in $\mathbb{R}^M$. Though this may be guaranteed with as few as $2K + 1$ measurements, it could also be the case that such an embedding would be very poorly conditioned. Intuitively, if two far-away points $x, x' \in \mathcal{M}$ were to be mapped onto nearby points in $\mathbb{R}^M$, then a recovery algorithm would need to take special care in resolving signals living near x or x'. As indicated in Theorem 6.2, however, additional measurements will ensure a well-conditioned embedding of $\mathcal{M}$. While the theorem provides a useful insight into the interaction of various manifold parameters (dimension, volume, curvature, etc.), we also defer in this section to empirical results

for determining the number of required measurements, as (i) the constants derived for (6.1) are possibly still too loose, and (ii) it may not be known whether a particular signal manifold meets the assumptions of Theorem 6.2 or with what parameters (though this is an important topic for future work).

In the second case, when the signal may only approximately obey the manifold model, we would like our recovery algorithm (6.4) to provide a robust estimate. This robustness will again naturally relate to the quality of the embedding of $\mathcal{M}$ in $\mathbb{R}^M$. Intuitively, if two far-away points $x, x' \in \mathcal{M}$ were to be mapped onto nearby points, then accurate recovery of any signals falling between x and x' would be difficult. Section 6.2.3 makes this notion more precise and proposes specific bounds for stable recovery of manifold-modeled signals.

6.2.3 Stable recovery

Let x^* be the "nearest neighbor" to x on $\mathcal{M}$, i.e.,

$$x^* = \arg\min_{x' \in \mathcal{M}} \|x - x'\|_2 , \tag{6.5}$$

supposing that this point is uniquely defined. To consider this recovery successful, we would like to guarantee that $\|x - \widehat{x}\|_2$ is not much larger than $\|x - x^*\|_2$. As discussed above, this type of stable, robust recovery will depend on a well-conditioned embedding of $\mathcal{M}$. To make this more precise, we consider both deterministic (instance-optimal) and probabilistic bounds for signal recovery.

A deterministic bound

To state a deterministic, instance-optimal bound on signal recovery we use the following measure for the quality of the embedding of $\mathcal{M}$ [41, 42]

$$\kappa := \inf_{x, x' \in \mathcal{M};\ x \neq x'} \frac{\|\Phi x - \Phi x'\|_2}{\|x - x'\|_2}.$$

We have the following theorem.

Theorem 6.3 *Suppose $x \in \mathbb{R}^N$ and that Φ is an orthoprojector from $\mathbb{R}^N$ to $\mathbb{R}^M$. Let $\widehat{x}$ be the estimation recovered from the projection $y = \Phi x$ (according to (6.4)), and let x^* be the optimal estimate of x (according to (6.5)). Then*

$$\frac{\|x - \widehat{x}\|_2}{\|x - x^*\|_2} \leq \sqrt{\frac{4}{\kappa^2} - 3 + 2\sqrt{\frac{1}{\kappa^2} - 1}}.$$

Proof: See Appendix E.

As $\kappa \to 1$, the bound on the right reduces simply to 1, and as $\kappa \to 0$, the bound grows as $2/\kappa$. Supposing that a sufficient number (6.1) of random measurement are taken for a signal manifold, Theorem 6.2 indicates that with high probability, we can expect

$$\kappa > (1-\epsilon)\sqrt{\frac{M}{N}}.$$

Supposing this holds, Theorem 6.3 then gives a deterministic bound on recovery for any $x \in \mathbb{R}^N$. We stress that this is a *worst case* bound, however, and as we discuss below, the accuracy is often significantly better.

We mention also that the algorithms introduced in [41, 42] aim specifically to find projection directions that maximize the quantity κ. However these lack the *universal* applicability of random projections.

Finally, it is worth noting that Theorem 6.3 can be used to derive an ℓ_2 instance-optimal bound for sparsity-driven CS recovery, by noting that the RIP of order $2K$ implies that all distinct K-sparse signals remain well-separated in $\mathbb{R}^M$ and gives a corresponding lower bound on the κ for the embedding of Σ_K. However, this instance-optimal bound would also be quite weak, as it is impossible to derive strong ℓ_2 instance-optimal bounds for CS [137].

A probabilistic bound

Our bound in Theorem 6.3 applies uniformly to any signal in $\mathbb{R}^N$. However, a much sharper bound can be obtained by relaxing the instance-optimal requirement. Such a guarantee comes again from the JL lemma. Assuming that the random orthoprojector Φ is statistically independent of the signal x, then we may recall Section C.3 of the proof of Theorem 6.2 and consider the embedding of the set $\{x\} \cup B$ under Φ. With high probability,[2] each pairwise distance in this set will have compaction isometry ϵ_1. Hence, the distance from x to each anchor point will be well-preserved, and since every manifold point is no more than T from an anchor point, then (assuming $\|x - x^*\|_2$ is sufficiently larger than T) the distance from x to every point on $\mathcal{M}$ will be well-preserved. This guarantees a satisfactory recovery $\widehat{x}$ in the approximate nearest neighbor problem. (By examining, for example, the tangent spaces, this can all be made more precise and extended to consider the case where $\|x - x^*\|_2$ is small.)

6.2.4 Basic examples

In order to illustrate the basic principles in action, we now consider a few examples involving random projections of parametrized manifolds.

[2] By the addition of an extra point to the embedding, there is a nominal increase in the required number of measurements. This increase becomes much more relevant in the case where a large number of signals x would need to be embedded well with respect to the manifold.

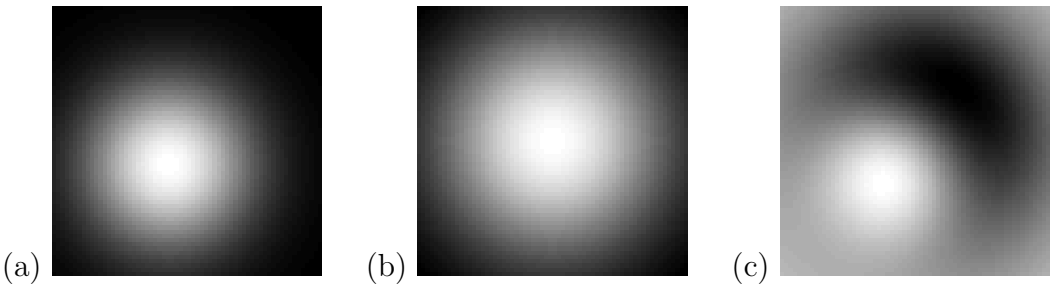

Figure 6.2: *(a) Original image for our experiment, containing a Gaussian bump parametrized by its position and width. (b) Initial guess for parameter estimation. (c) Error image between original and initial guess. From just* 14 *random measurements we can recover the unknown parameters of such an image with very high accuracy and with high probability.*

Gaussian bumps

Our first experiment involves a *smooth* image appearance manifold (IAM) in which each image contains a smooth Gaussian bump. For a given N-pixel image x_θ, the parameter θ describes both the position (2-D) and width (1-D) of the bump; see Figure 6.2(a) for one such image. (Because the bump is smooth, the IAM will be smooth as well.) We fix the amplitude of each bump equal to 1.

We consider the problem of estimating, from a collection of measurements $y = \Phi x_\theta$, the unknown parameter θ. Our test image x_θ is shown in Figure 6.2(a); we choose $N = 64 \times 64 = 4096$. To estimate the unknown parameter, we use 5 iterations of Newton's method, ignoring the second derivative term as discussed in Section 4.5.2. Our starting guess for this iterative algorithm is shown in Figure 6.2(b). (We chose this guess manually, but it could also be obtained, for example, by using a grid search in $\mathbb{R}^M$.) Figure 6.2(c) shows the relative error between the true image and the initial guess. For various values of M, we run 1000 trials over different realizations of the random Gaussian $M \times N$ matrix Φ.

We see in this experiment that the 3-D parameter θ can be recovered with very high accuracy using very few measurements. When $M = 7$ $(= 2 \cdot 3 + 1)$, we recover θ to very high accuracy (image MSE of 10^{-8} or less) in 86% of the trials. Increasing the probability of accurate recovery to 99% requires just $M = 14$ measurements, and surprisingly, with only $M = 3$ we still see accurate recovery in 12% of the trials. It appears that this smooth manifold is very well-behaved under random projections.

Chirps

Our second experiment concerns another smooth (but more challenging) manifold. We consider 1-D, length-N linear chirp signals, for which a 2-D parameter θ describes the starting and ending frequencies. Our test signal of length $N = 256$ is shown in Figure 6.3(a) and has starting and ending frequencies of 5.134Hz and 25.795Hz,

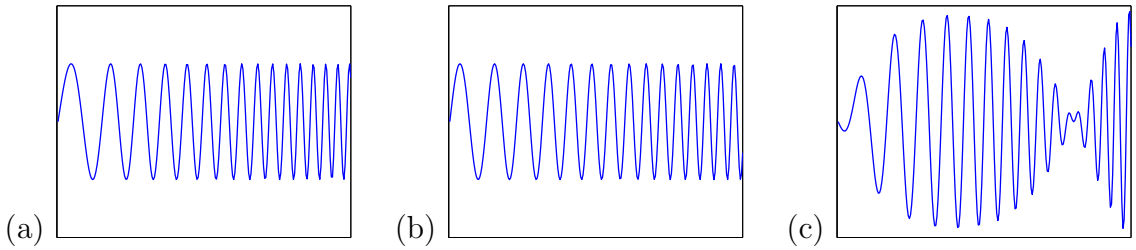

Figure 6.3: *(a) Original signal for our experiment, containing a linear chirp parametrized by its starting and ending frequencies. (b) Initial guess for parameter estimation. (c) Error signal between original and initial guess.*

respectively. To estimate the unknown parameters from random measurements, we use 10 iterations of the modified Newton's method in $\mathbb{R}^M$; our initial guess is shown in Figure 6.3(b) and has starting and ending frequencies of 7Hz and 23Hz, respectively. Figure 6.3(c) shows the relative error between the true signal and the starting guess.

When $M = 5$ $(= 2 \cdot 2 + 1)$, we recover θ to very high accuracy (image MSE of 10^{-8} or less) in 55% of the trials. Increasing the probability of accurate recovery to 99% requires roughly $M = 30$ measurements.

Across additional trials (including much higher N), we have observed that the successful recovery of chirp parameters is highly dependent on an accurate starting guess. Without an accurate initial guess, convergence is rare even with large M. Given an accurate initial guess, however, we often see recovery within the range of M described above. We attribute this sensitivity to the large area of this particular manifold. Indeed, just fixing the starting and ending frequencies to be equal (so that each signal is just a sinusoid, parametrized by its frequency), the manifold will visit all N unit vectors of the Fourier basis (each of which is orthogonal to the others). So, while smooth, this manifold does present a challenging case for parameter estimation.

Edges

We now consider a simple image processing task: given random projections of an N-pixel image segment x, recover an approximation to the local edge structure. As a model for this local edge structure, we adopt the 2-D wedgelet manifold. (Recall from Chapter 3 that a wedgelet is a piecewise constant function defined on a dyadic square block, where a straight edge separates the two constant regions; it can be parametrized by the slope and offset of the edge.) Unlike our experiments above, this manifold is non-differentiable, and so we cannot apply Newton's method. Instead, we sample this manifold to obtain a finite collection of wedgelets, project each wedgelet to $\mathbb{R}^M$ using Φ, and search for the closest match to our measurements $y = \Phi x$. (In Section 6.2.5 we discuss a Multiscale Newton method that could be applied in non-differentiable cases like this.)

As a first experiment (Figure 6.4), we examine a perfect edge originating on the

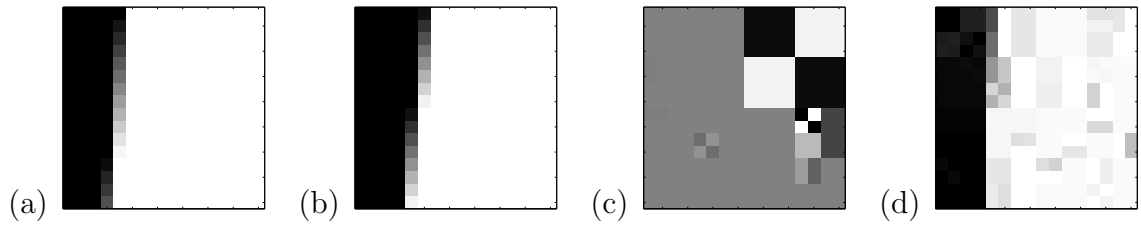

Figure 6.4: *Estimating image edge structure from a* 256*-pixel block. (a) original* 16×16 *block. (b) manifold-based recovery from* 5 *random projections. (c) traditional CS recovery from* 7 *random projections using OMP [91]. (d) OMP recovery from* 50 *random projections. Perfect OMP recovery requires* 70 *or more random projections.*

wedgelet manifold (but one that is not precisely among our discretized samples). We let $N = 16 \times 16 = 256$ and take $M = 5$ $(= 2 \cdot 2 + 1)$ random projections. Although the sampling grid for the manifold search does not contain Φx precisely, we see in Figure 6.4(b) that a very close approximation is recovered. In contrast, using traditional CS techniques to recover x from its random projections (seeking a sparse reconstruction using 2-D Haar wavelets) requires an order of magnitude more measurements.

As a second experiment (Figure 6.5) we analyze the robustness of the recovery process. For this we consider a 256×256 portion of the *Peppers* test image. We break the image into squares of size 16×16, measure each one using 10 random projections, and then search the projected wedgelet samples to fit a wedgelet on each block. (We also include the mean and energy of each block as 2 additional "measurements," which we use to estimate the 2 grayscale values for each wedgelet.) We see from the figure that the recovery is fairly robust and accurately recovers most of the prominent edge structure. The recovery is also fast, taking less than one second for the entire image. For point of comparison we include the best-possible wedgelet approximation, which would require all 256 numbers per block to recover. In spite of the relatively small κ generated by the random projections (approximately 0.05 when computed using the sampled wedgelet grid), the worst case distortion (as measured by $\|x - \widehat{x}\|_2 / \|x - x^*\|_2$ in Theorem 6.3) is approximately 3. For reference, we also include the CS-based recovery from an equivalent number, $(10 + 2) \cdot 256 = 3072$, of *global* random projections. Though slightly better in terms of mean-square error, this approximation fails to prominently represent the edge structure (it also takes several minutes to compute using our software). We stress again, though, that the main purpose of this example is to illustrate the robustness of recovery on natural image segments, some of which are not well-modeled using wedgelets (and so we should not expect high quality wedgelet estimates in every block of the image).

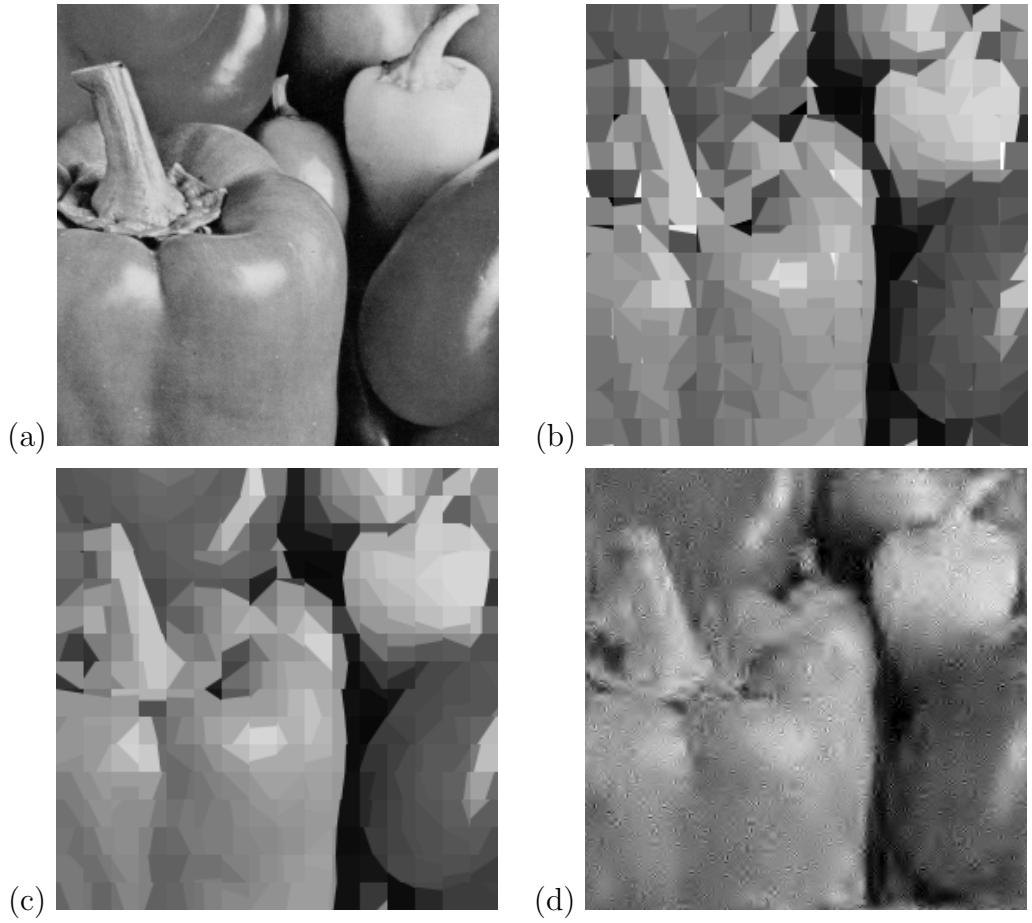

Figure 6.5: *(a) Original* 256×256 Peppers *image. (b) Wedgelet estimation on* 16×16 *pixel tiles, using 10 random projections (plus the mean and energy) on each tile, for a total of* $(10+2) \cdot 256 = 3072$ *measurements. (c) Best-possible wedgelet estimation, which would require all* $256^2 = 65536$ *pixel values. (d) Traditional CS-based recovery (from* 3072 *global random projections) using greedy pursuit to find a sparse approximation in the projected wavelet (D8) basis.*

6.2.5 Non-differentiable manifolds

As discussed in Chapter 4, many interesting signal manifolds are not differentiable. In our setting, this presents a challenge, as Theorem 6.2 does not give any insight into the required number of random projections for a stable embedding, and we can no longer apply Newton's method for parameter estimation. (As shown in Figure 6.1, the projection of a non-differentiable manifold in $\mathbb{R}^N$ typically yields another non-differentiable manifold in $\mathbb{R}^M$.) To address this challenge, we can again rely on the multiscale insight developed in Chapter 4: each non-differentiable IAM can be approximated using a sequence of differentiable manifolds that correspond to various scales of regularization of the original image. To get an approximate understanding

of the behavior of a non-differentiable manifold under random projections, one could study the behavior of its smooth approximations under random projections.

Unfortunately, to solve the parameter estimation problem we cannot immediately apply the Multiscale Newton algorithm to the random measurements $y = \Phi x_\theta$. Letting g_s denote the regularization kernel at scale s, the problem is that the Multiscale Newton algorithm demands computing $(x_\theta * g_s)$, which would live on a differentiable manifold, but the hypothetical measurements $\Phi(x_\theta * g_s)$ of such a signal cannot be computed from the given measurements $y = \Phi x_\theta$.

We propose instead a method for modifying the measurement matrix Φ in advance to accommodate non-differentiable manifolds. Our suggestion is based on the fact that, for a given measurement vector ϕ_i, one can show that

$$\langle \phi_i, x_\theta * g_s \rangle = \langle \phi_i * g_s, x_\theta \rangle .$$

Thus, by regularizing the measurement vectors $\{\phi_i\}$, the resulting image of the manifold in $\mathbb{R}^M$ will be differentiable. To accommodate the Multiscale Newton method, we propose specifically to (i) generate a random Φ, and (ii) partition the rows of Φ into groups, regularizing each group by a kernel g_s from a sequence of scales $\{s_0, s_1, \ldots, s_L\}$. The Multiscale Newton method can then be performed on the regularized random measurements by taking these scales $\{s_0, s_1, \ldots, s_L\}$ in turn.

A similar sequence of randomized, multiscale measurement vectors were proposed in [29] in which the vectors at each scale are chosen as a random linear combination of wavelets at that scale, and the resulting measurements can be used to reconstruct the wavelet transform of a signal scale-by-scale. A similar measurement process would be appropriate for our purposes, preferably by choosing random functions drawn from a coarse-to-fine succession of scaling spaces (rather than difference spaces). Additionally, one may consider using *noiselets* [138] as measurement vectors. Noiselets are deterministic functions designed to appear "noise-like" when expanded in the wavelet domain and can be generated using a simple recursive formula. At each scale j, the noiselet functions give a basis for the Haar scaling space V_j (the space of functions that are constant over every dyadic square at scale j). For a multiscale measurement system, one could simply choose a subset of these vectors at each scale.

At a very high level, we can get a rough idea of the number of measurements required at each scale of the algorithm. Supposing we square the scale between successive iterations, the curvature of the regularized manifolds grows quadratically, and Theorem 6.2 then suggests that finer scales scales require more measurements. However, if we assume quadratic accuracy of the estimates, then the region of uncertainty (in which we are refining our estimate) shrinks. Its volume will shrink quadratically, which will tend to counteract the effect of the increased curvature. A more thorough analysis is required to understand these effects more precisely; in the following demonstration, we choose a conservative sequence of scales but take a constant number of measurements at each scale.

As an experiment, we now consider the non-differentiable IAM consisting of para-

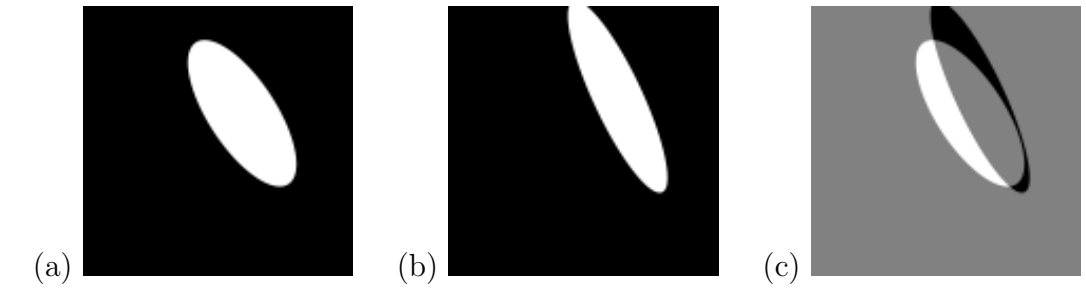

Figure 6.6: *(a) Original image for our experiment, containing an ellipse parametrized by its position, rotation, and major and minor axes. (b) Initial guess for parameter estimation. (c) Error image between original and initial guess.*

Figure 6.7: *Random measurement vectors at a sequence of scales* $s = 1/4, 1/8, 1/16, 1/32, 1/128$.

metrized ellipse images, where the 5-D parameter θ describes the translation, rotation, and major and minor axes of the ellipse. Our test image with $N = 128 \times 128 = 16384$ is shown in Figure 6.6(a); our initial guess for estimation is shown in Figure 6.6(b); and the relative initial error is shown in Figure 6.6(c).

In each trial, we consider multiscale random measurement vectors (regularized Gaussian noise) taken at a sequence of 5 scales $s = 1/4, 1/8, 1/16, 1/32, 1/128$. Figure 6.7 shows one random basis function drawn from each such scale. We take an equal number of random measurements at each scale, and to perform each Newton step we use all measurements taken up to and including the current scale.

Choosing $M = 6$ random measurements *per scale* (for a total of 30 random measurements), we can recover the ellipse parameters with high accuracy (image MSE of 10^{-5} or less) in 57% of trials. With $M = 10$ measurements per scale (50 total), this probability increases to 89%, and with $M = 20$ measurements per scale (100 total), we see high accuracy recovery in 99% of trials.

Using noiselets for our measurement vectors (see Figure 6.8 for example noiselet functions) we see similar performance. Choosing $M = 6$ random noiselets[3] per scale (30 total), we see high accuracy recovery in 13% of trials, but this probability increases

[3]Each noiselet is a complex-valued function; we take $M/2$ per scale, yielding M real measurements.

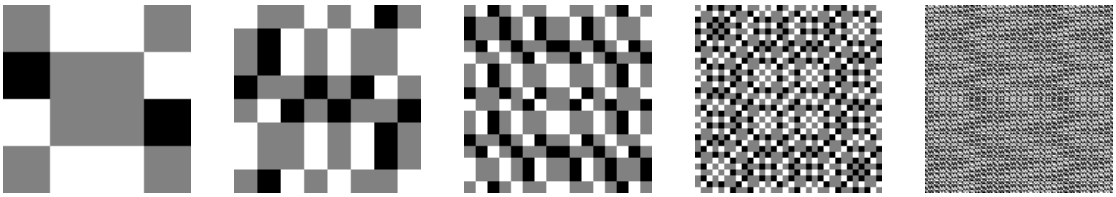

Figure 6.8: *Real components of noiselet measurement vectors at scales $j = 2, 3, 4, 5, 7$.*

to 59% with $M = 10$ random noiselets per scale (50 total) and to 99% with $M = 22$ random noiselets per scale (110 total).

In terms of the number of random measurements required for parameter estimation, it does appear that there is a moderate price to be paid in the case of non-differentiable manifolds. We note, however, that in our ellipse experiments the recovery does seem relatively stable, and that with sufficient measurements, the algorithm rarely diverges far from the true parameters.

6.2.6 Advanced models for signal recovery

In our examples thus far, we have considered the case where a single manifold model is used to describe the signal x. Many manifolds, however, are intended as models for *local* signal structure, and for a given signal x there may in fact be multiple, local manifold models appropriate for describing the different parts of the signal. As an example, we may again consider wedgelets, which are appropriate for modeling locally straight edges in images. For an entire image, a *tiling* of wedgelets is much more appropriate as a model than a single wedgelet. In our CS experiment in Figure 6.5, we used a wedgelet tiling to recover the image, but our random measurements were partitioned to have supports localized on each wedgelet. In general, we cannot expect to have such a partitioning of the measurements, and in fact all of the measurement vectors may be *global*, each being supported over the entire signal. As a proof of concept in this section, we present two methods for joint parameter estimation across multiple manifolds in the case where the CS measurements have global support. As an illustration, we continue to focus on recovering wedgelet tilings.

At first glance, the problem of recovering the parameters for a given wedgelet appears difficult when the measurement vectors have significantly larger support. Writing $y = \Phi x$, where x now represents the entire image, the influence of a particular wedgelet block will be restricted to relatively few columns of Φ, and the rest of an image will have a large influence on the measurements y. Indeed, if one were to estimate the image block-by-block, fitting a wedgelet to each block as if y were a noisy measurement of that block alone, such estimates would be quite poor. Figure 6.9(a), for example, shows a 128×128 test image from which we take $M = 640$ global random measurements, and Figure 6.9(d) shows the block-by-block estimates using 16×16 wedgelets. (For simplicity in this section we use a nearest neighbor grid search to

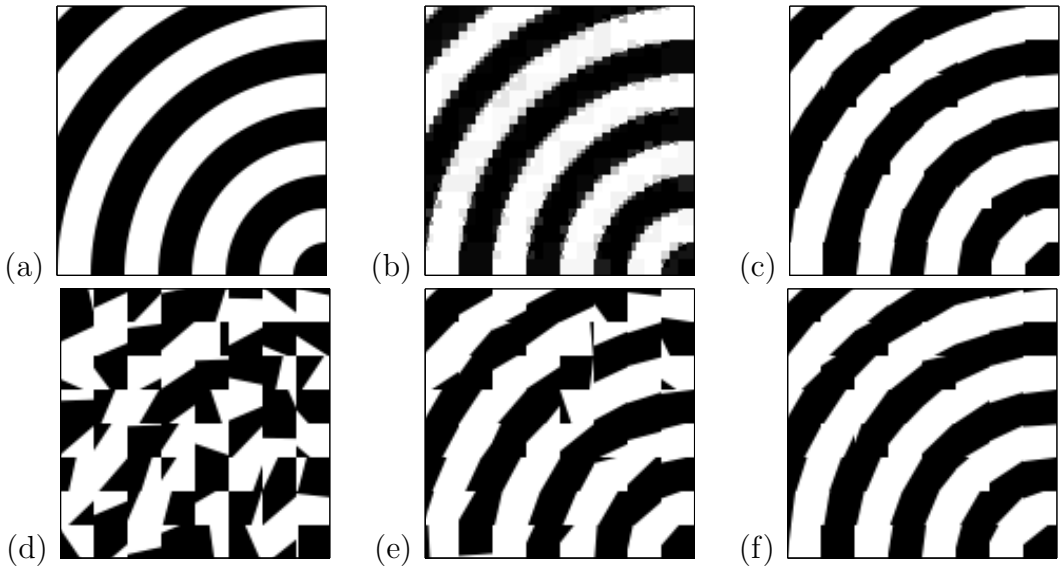

Figure 6.9: *(a) Original* 128×128 *image for our experiment. (b) Wavelet thresholding with* 640 *largest Haar wavelet coefficients, PSNR 18.1dB. (c) Oracle wedgelet approximation to image using wedgelets of size* 16×16 *pixels, PSNR 19.9dB. (d) Wedgelet estimate recovered from* $M = 640$ *global random projections after 1 iteration, PSNR 7.1dB. (e) Estimate after 5 iterations, PSNR 13.6dB. (f) Estimate after 10 iterations, PSNR 19.1dB.*

obtain wedgelet estimates in $\mathbb{R}^M$.)

The above experiment implies that local recovery will not suffice for parameter estimation across multiple manifolds. However, we can propose a very simple but effective algorithm for joint parameter estimation. The algorithm we propose is simply to use the local estimates (shown in Figure 6.9(d)) as an initial guess for the wedgelet on each block, then perform block-by-block estimates again on the residual measurements (subtracting off the best guess from each other block). Figure 6.9(e) and Figure 6.9(f) show the result of this successive estimation procedure after 5 and 10 iterations, respectively. After 10 iterations, the recovered wedgelet estimates approach the quality of oracle estimates for each block (Figure 6.9(c)), which would require all 128×128 pixel values. Instead, our estimates are based on only 640 global random projections, an average of 10 measurements per wedgelet block. For point of comparison, we show in Figure 6.9(b) the best 640-term representation from the 2-D Haar wavelet dictionary; our wedgelet estimates outperform even this upper bound on the performance of sparsity-based CS recovery.

This is encouraging news — we have proposed a simple iterative refinement algorithm that can distill local signal information from the global measurements y. While promising, this technique also has its limitations. Consider for example the 128×128 test image in Figure 6.10(a). For this image we take $M = 384$ global random measurements, and in Figure 6.10(c) we show the collection of 8×8 wedgelet estimates

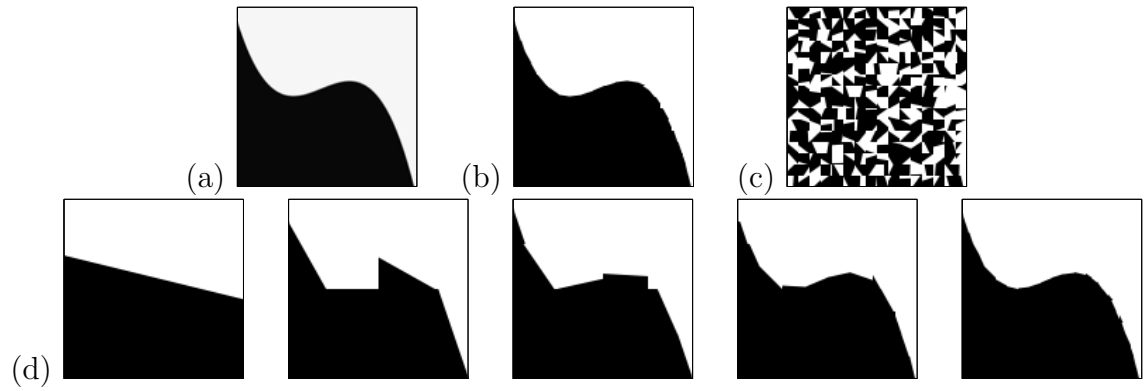

Figure 6.10: *(a) Original* 128×128 *image for our experiment. (b) Oracle wedgelet approximation to image using wedgelets of size* 8×8 *pixels, PSNR 26.9dB. (c)* 8×8 *wedgelet estimates from* $M = 384$ *global random measurements using single-scale iterative algorithm, PSNR 4.1dB. (d) Successive wedgelet estimates from top-down multiscale estimation algorithm. From left to right: wedgelets of size* 128×128, 64×64, 32×32, 16×16, *and* 8×8*; final PSNR 26.5dB.*

returned after 10 iterations of the above algorithm. In this experiment we have an average of only 1.5 measurements per wedgelet block, and the resulting estimates are quite poor.

Ideally we would like to use wedgelets as more than a *local* model for signal structure. While each wedgelet is designed to capture edge structure on a single block, as we discussed in Chapter 3, these blocks are related in space and in scale. A *multiscale* wedgelet model would capture both of these effects and encourage more accurate signal recovery. As a first attempt to access the multiscale structure, we propose a top-down, coarse-to-fine wedgelet estimation algorithm, where at each scale we use the single-scale iterative algorithm described above, but the starting guess for each scale it obtained from the previous (coarser) scale. Returning to our experiment using $M = 384$ global random measurements, Figure 6.10(d) shows our sequence of estimates for wedgelet block sizes 128×128, 64×64, 32×32, 16×16, and finally 8×8. Thanks to the multiscale model, the quality of our ultimate wedgelet estimates on 8×8 blocks is comparable to the best-possible oracle wedgelet estimates (shown in Figure 6.10(b)).

6.3 Applications in Manifold Learning

Theorem 6.2 implies that, in some sense, the structure of a manifold is well preserved when it is mapped under a random projection to a low-dimensional space. In Section 6.2, we discussed possible applications of this fact in CS, where we wish to recover information about a single signal based on its random measurements. In this section, we consider instead possible applications involving collections of multiple

signals.

6.3.1 Manifold learning in $\mathbb{R}^M$

We recall from Section 2.7.1 that the basic problem of manifold learning is to discover some information about a manifold based on a collection of data sampled from that manifold. In standard problems, this data is presented in $\mathbb{R}^N$ (the natural ambient signal space).

For several reasons it may be desirable to reduce the dimension N. First of all, the process of acquiring and storing a large number of manifold samples may be difficult when the dimension N is large. Second, the computational complexity of manifold learning algorithms (e.g., when computing pairwise distances and nearest neighbor graphs) will depend directly on N as well.

Fortunately, Theorem 6.2 and Corollary 6.1 imply that many of the properties of a manifold $\mathcal{M}$ one may wish to discover from sampled data in $\mathbb{R}^N$ are approximately preserved on its image $\Phi\mathcal{M}$ under a random projection to $\mathbb{R}^M$. Among these properties, we have

- ambient and geodesic distances between pairs of points;
- dimension of the manifold;
- topology, local neighborhoods, and local angles;
- lengths and curvature of paths on the manifold; and
- volume of the manifold.

(Some of these follow directly from Theorem 6.2 and Corollary 6.1; others depend on the near-isometry of the projected tangent spaces as discussed in Section C.4.)

These are some of the basic properties sought by the manifold learning algorithms listed in Section 2.7.1 (ISOMAP, LLE, HLLE, MVU, etc.), and and so it appears that we should be able to apply such algorithms to random projections of the original data and get an approximation to the true answer. (While this does involve an initial projection of the data to $\mathbb{R}^M$, we recall from Section 2.8.4 that certain hardware systems are under development for CS that do not require first sampling and storing the data in $\mathbb{R}^N$.)

While we have not conducted a rigorous analysis of the sensitivity of such algorithms to "noise" in the data (as each of the above properties is slightly perturbed during the projection to $\mathbb{R}^M$), we present in the following section a simple experiment as a proof of concept.

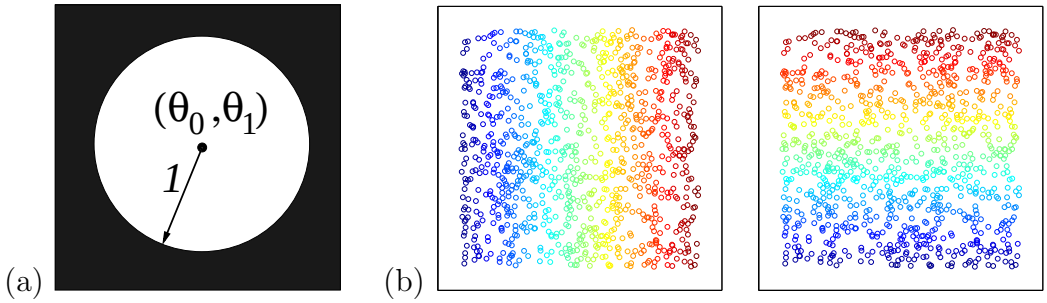

Figure 6.11: *(a) Model for disk. We generate 1000 samples, each of size* $N = 64 \times 64 = 4096$. *(b)* θ_0 *and* θ_1 *values for original data in* $\mathbb{R}^N$.

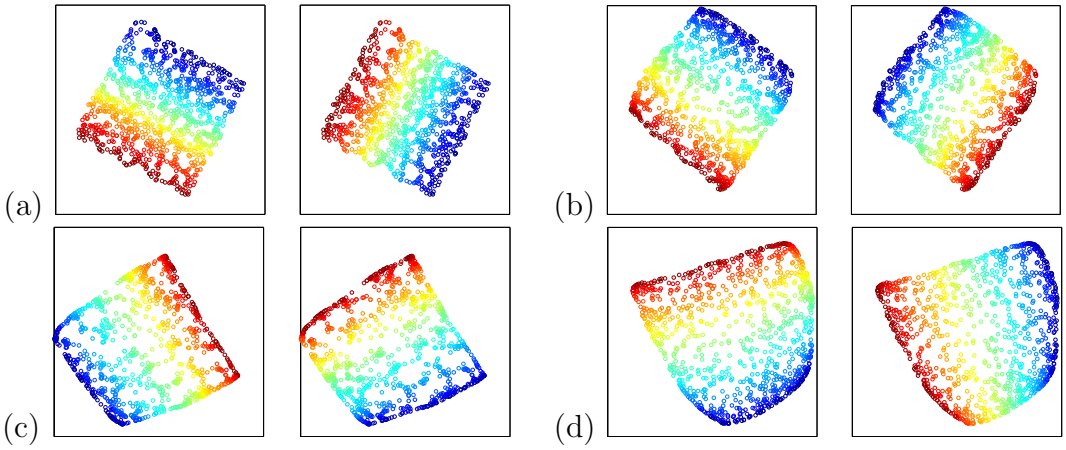

Figure 6.12: *2-D embeddings learned from the data in* $\mathbb{R}^{4096}$ *(see Figure 6.11). (a) ISOMAP, (b) HLLE, (c) Laplacian Eigenmaps, (d) LLE.*

6.3.2 Experiments

To test the performance of several of the manifold learning algorithms on projected data, we consider the problem of learning an isometric embedding of a parametrized image manifold. We generate 1000 images of a translated disk (see Figure 6.11(a)), each of size $N = 64 \times 64 = 4096$. The parameter $\theta = (\theta_0, \theta_1)$ describes the center of each disk; we choose 1000 random values as shown in Figure 6.11(b). In each such plot, the color/shading of the left and right images represent the true values for θ_0 and θ_1 respectively. (We show these colors for the purpose of interpreting the results; the true values of θ_0 and θ_1 are not provided to the manifold learning algorithms.)

Figure 6.12 shows the 2-D embeddings learned by the ISOMAP [44], HLLE [45], Laplacian Eigenmaps [53], and LLE [47] algorithms when presented with the 1000 samples in $\mathbb{R}^{4096}$. Each of these algorithms approximately recovers the true underlying parametrization of the data; the rotations of the square relative to Figure 6.11(b) are irrelevant.

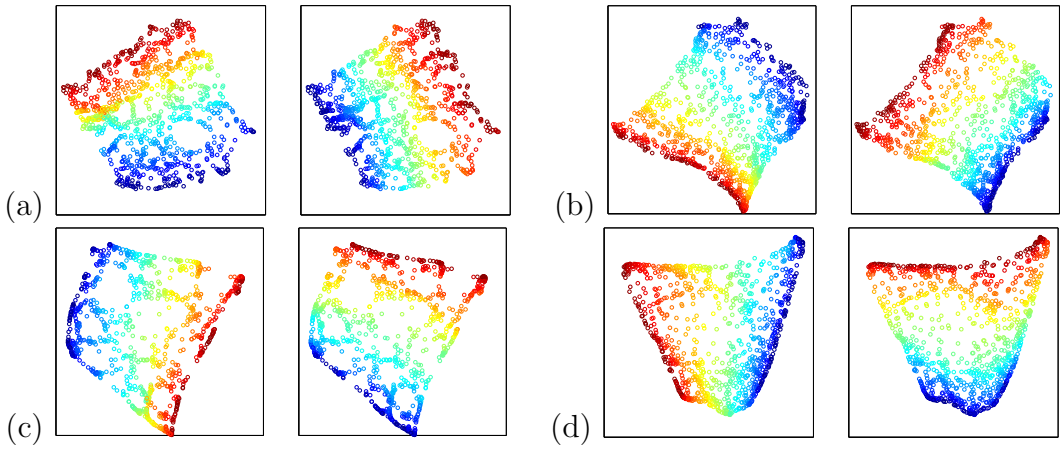

Figure 6.13: *2-D embeddings learned from random projections of the data from* $\mathbb{R}^{4096}$ *to* $\mathbb{R}^M$ *(see Figure 6.11). (a) ISOMAP* ($M = 15$)*, (b) HLLE* ($M = 20$)*, (c) Laplacian Eigenmaps* ($M = 15$)*, (d) LLE* ($M = 200$)*.*

For various values of M, we then construct a random $M \times N$ Gaussian matrix Φ and rerun the algorithms on the projections of the 1000 data points in $\mathbb{R}^M$. Figure 6.13 shows the 2-D embeddings learned by the same algorithms when presented with the samples in $\mathbb{R}^M$. For each algorithm, we show the value of M at which a reasonable embedding is recovered. We see that all algorithms again return an approximation to the true underlying parametrization of the data. With regard to the number of measurements M, the ISOMAP, HLLE, and Laplacian Eigenmaps algorithms appear to be the most stable in this experiment ($M \approx 15$ to 20). In contrast, the LLE algorithm requires a much greater number of measurements $M \approx 200$ but at a level still significantly below the ambient dimension $N = 4096$.

Chapter 7
Conclusions

Many real-world signals have a structure that can be summarized in a "concise" manner relative to the size N of the signal. Efficient processing of such signals relies on representations and algorithms that can access this concise structure. As we have seen, concise models often imply a K-dimensional geometric structure to the signal class within the ambient space $\mathbb{R}^N$, where $K \ll N$, and the geometry of this class itself often holds the clue for developing new and more efficient techniques that operate at the "information level" K of the signal.

Our contributions in this thesis have included: *new models* for low-dimensional signal structure, including local parametric models for piecewise smooth signals and joint sparsity models for signal collections; *new multiscale representations* for piecewise smooth signals designed to accommodate efficient processing; and *new dimensionality reduction algorithms* for problems in approximation, compression, parameter estimation, manifold learning, and Compressed Sensing (CS). There are many possible future directions for this research.

7.1 Models and Representations

7.1.1 Approximation and compression

We demonstrated in Chapter 3 that surflets provide an effective parametric representation for local discontinuities in piecewise constant signals. Because surflets (like wavelets) are organized on dyadic hypercubes, we were able to easily combine the two representations for approximation and compression of piecewise smooth signals (using surfprints — the projections of surflet atoms onto wavelet subspaces). Moreover, an efficient bottom-up tree-pruning algorithm can be used to find the best combination of surfprints and wavelets.

As we discuss in [102], this "plug-and-play" encoding strategy is a generalization of the SFQ algorithm for natural image coding [11]. Given this framework, there is no particular reason our representations must be limited simply to surfprints and wavelets, however. In fact, any local phenomenon amenable to concise modeling and representation would be a candidate for a new type of "print" that could be added to the mix.

As an example, consider the generalization of wedgelets (in which an edge separates two constant-valued regions) to "barlets" (in which a bar of variable width crosses through a constant-valued region). Barlets can be viewed as a superset of

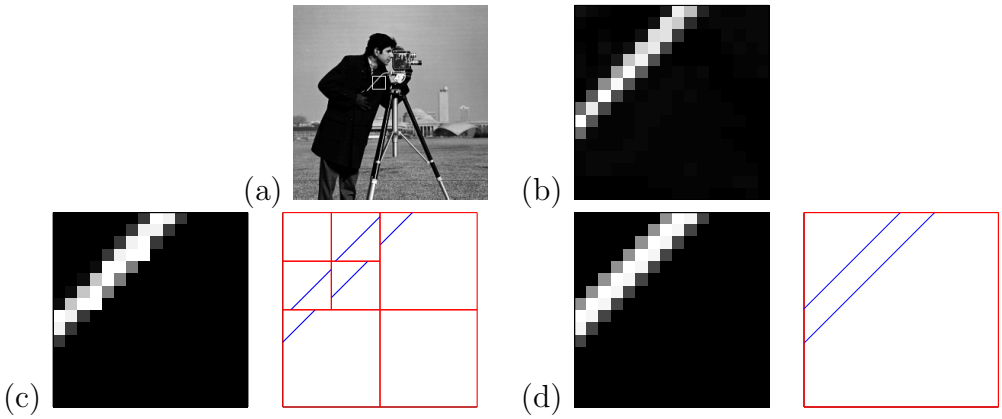

Figure 7.1: *(a) Original* Cameraman *image, from which a square segment is extracted that contains a bar-like feature (see white box near center of image). (b) Image segment extracted from* Cameraman. *(c) Coded wedgelet approximation using 7 wedgelets and requiring approximately 80 bits. (d) Coded approximation using a single barlet and requiring only 22 bits.*

the wedgelet dictionary (or the beamlet dictionary [139, 140]) and are designed to more concisely represent image regions such as the one shown in Figure 7.1(b). As an example, in Figure 7.1(c), we show an approximation to the image segment coded using a local tiling of wedgelets. A total of 7 wedgelets were used to represent the bar, requiring approximately 80 bits to jointly encode using a top-down predictive scheme. In contrast, Figure 7.1(d) shows a simple barlet representation of the same image segment that uses only a single barlet and requires only 22 bits to encode. (Compared with wedgelets, barlets are a more specific parametric model for local signal structure; additional experiments would be required to determine the actual value of this additional parameter in terms of rate-distortion performance, and the result would likely be largely image-dependent.) Because they are organized on dyadic squares, we can immediately imagine the translation of barlets to the wavelet domain, yielding "barprints" to be evaluated among the coding options at each node. Other representations, such as local DCT patches (as used in JPEG coding [141]) could also be considered as candidate primitive representations for additional types of prints.

One primary drawback for such a plug-and-play encoding scheme is the increased computational complexity required to evaluate each coding option at every node. An additional drawback, however, is in the additional bitrate required to distinguish at each node from all of the possible representations. Moreover, given this distinction it may be difficult to code the local region at the proper conditional entropy (given that the other dictionaries do *not* yield efficient representations).

7.1.2 Joint sparsity models

In Chapter 5, we took the first steps towards extending the theory and practice of Compressed Sensing (CS) to multi-signal, distributed settings. Our three simple joint sparsity models (JSMs) for signal ensembles were chosen to capture the essence of real physical scenarios, illustrate the basic analysis and algorithmic techniques, and indicate the significant gains to be realized from joint recovery. To better understand the specific nuances of sensor network problems, additional models should be considered. In particular, one important extension would generalize the JSMs from strictly sparse signals to compressible signals, in which the transform coefficients decay (recall Section 2.4.2). In JSM-2, for example, we can extend the notion of simultaneous sparsity for ℓ_p-sparse signals whose sorted coefficients obey roughly the same ordering. This condition could perhaps be enforced as an ℓ_p constraint on the composite signal

$$\left\{ \sum_{j=1}^{J} |x_j(1)|, \;\; \sum_{j=1}^{J} |x_j(2)|, \;\; \ldots, \;\; \sum_{j=1}^{J} |x_j(N)| \right\}.$$

Other open theoretical questions concern the sensitivity of DCS recovery to noise and quantization, though preliminary experiments on real-world data have been encouraging [142].

7.1.3 Compressed Sensing

In Chapter 6 we demonstrated that signals obeying manifold models can be recovered from small numbers of random projections. In conjunction with the standard results in CS, this suggests that many types of concise signal models may yield signal classes well-preserved under projections to low dimensions.

In CS, the process of recovering a signal from its random measurements depends critically on the *model*. As discussed in Section 2.8.3, given a set of measurements $y = \Phi x$ of a signal x, there are an infinite number of possibilities for the true signal. To distinguish from among these possibilities, one must choose a model (such as sparsity in some dictionary Ψ or nearness to some manifold $\mathcal{M}$). As we have seen in both sparsity-driven CS and manifold-driven CS, the quality of the reconstructed signal will be comparable to the efficiency of the model in representing the signal.

As a general rule, better signal models should lead to better CS recovery. The models adapted to date for CS recovery (sparsity or manifolds), while effective, only represent a basic portion of the total understanding of signal modeling. Signals are not only sparse, but their transform coefficients are often have dependencies. Manifolds often work best as local models for signal regions; the parameters between multiple manifold approximations are often related in space and in scale; and entirely different manifold models could be appropriate for different signal regions. Ultimately, it appears that these more sophisticated models will be key to improving CS reconstruction algorithms. The challenge, of course, will be developing algorithms that

can account for these more sophisticated models in distinguishing among all possible candidates for x. For sparsity-driven CS we have proposed one coarse-to-fine reconstruction scheme in the wavelet domain [93]. For manifold-driven CS we proposed in Section 6.2.6 two techniques for joint recovery of multiple manifold parameters. We believe that much more effective techniques can be developed, leveraging more sophisticated sparse and manifold-based models and perhaps even combining the two, for example, for simultaneous surfprint/wavelet estimation.

7.2 Algorithms

7.2.1 Parameter estimation

In Chapter 4 we presented a Multiscale Newton algorithm for parameter estimation. In addition to the convergence analysis mentioned in Section 4.5.2, a number of issues remain open regarding implementations of this algorithm. For instance, with noisy images the multiscale tangent projections will reach a point of diminishing returns where finer scales will not benefit; we must develop a stopping criterion for such cases. Additional issues revolve around efficient implementation. We believe that a sampling of the tangent planes needed for the projections can be precomputed and stored using the multiscale representation of [63]. Moreover, since many of the computations are local (as evidenced by the support of the tangent basis images in Figure 4.2), we expect that the image projection computations can be implemented in the wavelet domain. This would also lead to a fast method for obtaining the initial guess $\theta^{(0)}$ with the required accuracy.

7.2.2 Distributed Compressed Sensing

Another possible area of future work would be to reduce the computational complexity of reconstruction algorithms for DCS. In some applications, the linear program associated with some DCS decoders (in JSM-1 and JSM-3) could prove too computationally intense. As we saw in JSM-2, efficient iterative and greedy algorithms could come to the rescue, but these need to be extended to the multi-signal case.

7.3 Future Applications in Multi-Signal Processing

In this thesis, we have examined two main problems involving processing multiple signals: DCS and manifold learning. As new capabilities continue to emerge for data acquisition, storage, and communication, and as demand continues to increase for immersive multimedia, medical imaging, remote sensing, and signals intelligence, the importance of effective techniques for multi-signal processing will only continue to grow.

As with single-signal case, the first step in developing efficient algorithms for multi-signal processing is an accurate model for the signals of interest. Ideally, this model

should capture the *joint* structure among the signals in addition to their individual structure. Our JSMs, for example, were intended to capture both types of structure using the notion of sparsity. We can also imagine, however, many settings in which multiple signals may be acquired under very similar conditions (differing only in a few parameters controlling the acquisition of the signals). Some possible examples include:

- frames of a video sequence, differing only in the timestamp,
- radiographic slices from a computed tomographic (CT) scan or cryo-electron microscopy (cryo-EM) image, differing only in the relative position with respect to the subject, or
- images from a surveillance or entertainment camera network, differing only in the position of each camera.

In each of the above cases we have some common phenomenon X that represents the fundamental information of interest (such as the motion of an object in the video or the true 3-D structure of a molecule being imaged), and we collect information via signals that depending both on X and on the parameters θ of the acquisition process. From these signals we may wish to conclude information about X.

If we fix X in the above scenario, then it follows that as θ changes, the various signals will represent samples of some manifold $\mathcal{M}_X$ (e.g., in $\mathbb{R}^N$). We argued in Section 6.3, however, that the structure of a manifold will be well-preserved under random projection to a lower-dimensional space. This suggests that it may be possible to generalize DCS far beyond our JSMs to incorporate a wide variety of manifold-based models. In our above settings, this would involve collecting small number M of random projections from each viewpoint, rather than the size-N signal itself. Depending on the problem, this could significantly reduce the storage or communication demands.

The real challenge in such a generalization of DCS would be developing methods for recovering information about X based on random projections of samples from $\mathcal{M}_X$. While we believe that developing successful methods will likely be highly problem-dependent, we present here one final experiment to as a basic demonstration of feasibility.

Our setting for this experiment involves 1-D signals. We let $X \in \mathbb{R}^N$ denote a signal that we wish to learn. Figure 7.2(a) plots two different X with $N = 32$. Instead of X, we observe random projections of shifts of X. That is, θ represents the amount of shift and $\mathcal{M}_X \subset \mathbb{R}^{32}$ represents all circular shifts of X (including noninteger shifts so that the manifold is continuous). From samples of $\Phi\mathcal{M}_X$ in $\mathbb{R}^M$ we wish to recover X. In a sense, this is a *manifold recovery* problem — there exist an infinite number of candidate manifolds $\mathcal{M} \subset \mathbb{R}^N$ that would project to the same image $\Phi\mathcal{M}_X$. We must use the constraints of our acquisition system as a model and seek a manifold $\mathcal{M} \subset \mathbb{R}^N$ on which each signal is a shift of every other signal.

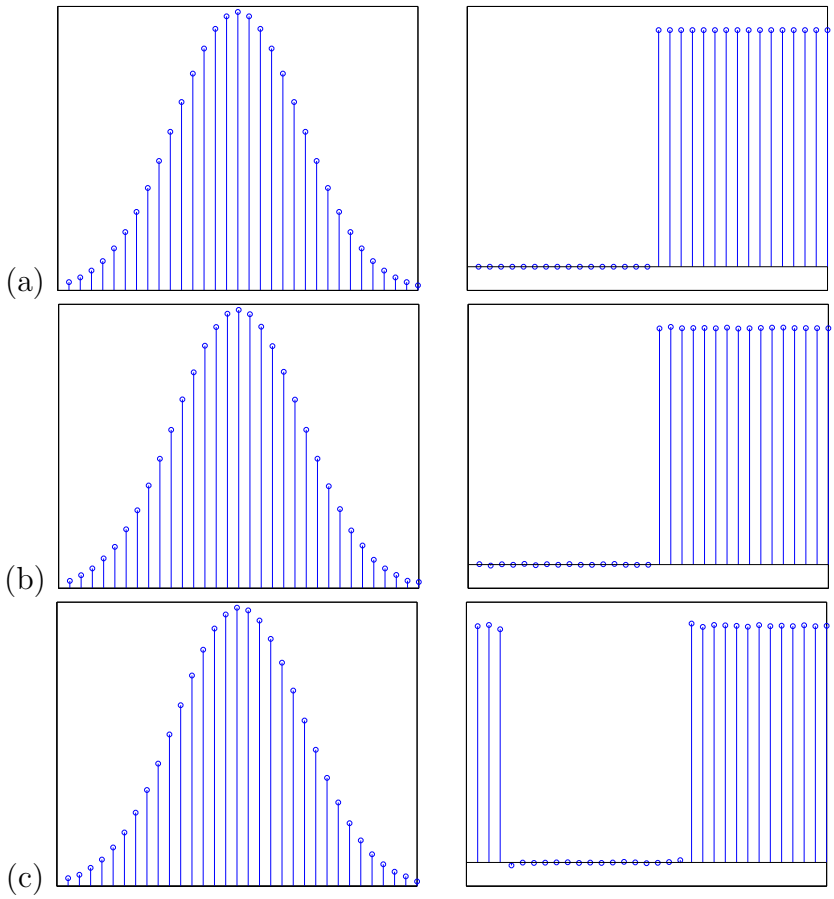

Figure 7.2: *(a) Original length-32 1-D signal X for our experiment. (b) Reconstruction using 20 random projections to $\mathbb{R}^3$ of various known delays of X. (c) Reconstruction using 20 random projections to $\mathbb{R}^{10}$ of various unknown delays of X.*

We begin with the case where each sample is labeled with its shift parameter θ. In this case, we can successfully "lift" the manifold from $\mathbb{R}^M$ back to $\mathbb{R}^N$ using an iterative estimation procedure. We construct an orthonormal basis Ψ in $\mathbb{R}^N$ and estimate the expansion coefficients for X iteratively in order to maximize agreement with the observed data. The results of this algorithm are shown in Figure 7.2(b). Using just $M = 3$ random projections from just 20 labeled samples we recover a highly accurate approximation to X.

The unlabeled case is more difficult, but it is possible to estimate the unknown shift parameters θ as well. We begin by computing geodesic distances among the sampled points in $\mathbb{R}^M$ and use the relative spacing as initial guesses for θ. We then alternate between the above iterative algorithm and refining our estimates for the θ. The results of are shown in Figure 7.2(c). In this case, we require about $M = 10$ random projections from each of 20 unlabeled samples to recover a good approximation to X. (The shift with respect to X of the step function is irrelevant.)

This simple experiment demonstrates that manifold recovery from random projections is indeed possible by enforcing the physical constraints dictated by the data collection process. In future work we will examine more relevant (and complicated) scenarios, particularly applications involving image processing and 3-D scene reconstruction.

Appendix A
Proof of Theorem 2.1

We first prove Statement 2, followed by Statements 1 and 3.

Statement 2 (Achievable, $M \geq K+1$): Since Ψ is an orthonormal basis, it follows that entries of the $M \times N$ matrix $\Phi\Psi$ will be i.i.d. Gaussian. Thus without loss of generality, we assume Ψ to be the identity, $\Psi = I_N$, and so $y = \Phi\alpha$. We concentrate on the "most difficult" case where $M = K+1$; other cases follow similarly.

Let Ω be the index set corresponding to the nonzero entries of α; we have $|\Omega| = K$. Also let Φ_Ω be the $M \times K$ mutilated matrix obtained by selecting the columns of Φ corresponding to the indices Ω. The measurement y is then a linear combination of the K columns of Φ_Ω. With probability one, the columns of Φ_Ω are linearly independent. Thus, Φ_Ω will have rank K and can be used to recover the K nonzero entries of α.

The coefficient vector α can be uniquely determined if no other index set $\widehat{\Omega}$ can be used to explain the measurements y. Let $\widehat{\Omega} \neq \Omega$ be a different set of K indices (possibly with up to $K-1$ indices in common with Ω). We will show that (with probability one) y is not in the column span of $\Phi_{\widehat{\Omega}}$, where the column span of the matrix A is defined as the vector space spanned by the columns of A and denoted by $\text{colspan}(A)$.

First, we note that with probability one, the columns of $\Phi_{\widehat{\Omega}}$ are linearly independent and so $\Phi_{\widehat{\Omega}}$ will have rank K. Now we examine the concatenation of these matrices $\begin{bmatrix}\Phi_\Omega & \Phi_{\widehat{\Omega}}\end{bmatrix}$. The matrix $\begin{bmatrix}\Phi_\Omega & \Phi_{\widehat{\Omega}}\end{bmatrix}$ cannot have rank K unless $\text{colspan}(\Phi_\Omega) = \text{colspan}(\Phi_{\widehat{\Omega}})$, a situation that occurs with probability zero. Since these matrices have $M = K+1$ rows, it follows that $\begin{bmatrix}\Phi_\Omega & \Phi_{\widehat{\Omega}}\end{bmatrix}$ will have rank $K+1$; hence the column span is $\mathbb{R}^{K+1}$.

Since the combined column span of Φ_Ω and $\Phi_{\widehat{\Omega}}$ is $\mathbb{R}^{K+1}$ and since each matrix has rank K, it follows that $\text{colspan}(\Phi_\Omega) \cap \text{colspan}(\Phi_{\widehat{\Omega}})$ is a $(K-1)$-dimensional linear subspace of $\mathbb{R}^{K+1}$. (Each matrix contributes one additional dimension to the column span.) This intersection is the set of measurements in the column span of Φ_Ω that could be confused with signals generated from the vectors $\widehat{\Omega}$. Based on its dimensionality, this set has measure zero in the column span of Φ_Ω; hence the probability that α can be recovered using $\widehat{\Omega}$ is zero. Since the number of sets of K indices is finite, the probability that there exists $\widehat{\Omega} \neq \Omega$ that enables recovery of α is zero.

Statement 1 (Achievable, $M \geq 2K$): We first note that, if $K \geq N/2$, then with probability one, the matrix Φ has rank N, and there is a unique (correct) reconstruction. Thus we assume that $K < N/2$. The proof of Statement 1 follows similarly to the proof of Statement 2. The key fact is that with probability one,

all subsets of up to $2K$ columns drawn from Φ are linearly independent. Assuming this holds, then for two index sets $\Omega \neq \widehat{\Omega}$ such that $|\Omega| = |\widehat{\Omega}| = K$, $\text{colspan}(\Phi_\Omega) \cap \text{colspan}(\Phi_{\widehat{\Omega}})$ has dimension equal to the number of indices common to both Ω and $\widehat{\Omega}$. A signal projects to this common space only if its coefficients are nonzero on exactly these (fewer than K) common indices; since $\|\alpha\|_0 = K$, this does not occur. Thus every K-sparse signal projects to a unique point in $\mathbb{R}^M$.

Statement 3 (Converse, $M \leq K$): If $M < K$, then there is insufficient information in the vector y to recover the K nonzero coefficients of α; thus we assume $M = K$. In this case, there is a single explanation for the measurements only if there is a single set Ω of K linearly independent columns *and* the nonzero indices of α are the elements of Ω. Aside from this pathological case, the rank of subsets $\Phi_{\widehat{\Omega}}$ will generally be less than K (which would prevent robust recovery of signals supported on $\widehat{\Omega}$) or will be equal to K (which would give ambiguous solutions among all such sets $\widehat{\Omega}$). $\square$

Appendix B
Proof of Theorem 5.3

Statement 2 follows trivially from Theorem 2.1 (simply assume that z_C is known a priori). The proof of Statement 1 has two parts. First we argue that $\lim_{J\to\infty} \widehat{z_C} = z_C$. Second we show that this implies vanishing probability of error in recovering each innovation z_j.

Part 1: We can write our estimate as

$$\begin{aligned}\widehat{z_C} &= \frac{1}{J}\widehat{\Phi}^T y = \frac{1}{J}\widehat{\Phi}^T \Phi x = \frac{1}{J}\sum_{j=1}^{J} \frac{1}{M_j\sigma_j^2} {\Phi_j}^T \Phi_j x_j \\ &= \frac{1}{J}\sum_{j=1}^{J} \frac{1}{M_j\sigma_j^2} \sum_{m=1}^{M_j} (\phi_{j,m}^R)^T \phi_{j,m}^R x_i,\end{aligned}$$

where Φ is a diagonal concatenation of the Φ_j's as defined in (5.2), and $\phi_{j,m}^R$ denotes the m-th row of Φ_j, that is, the m-th measurement vector for node j. Since the elements of each Φ_j are Gaussians with variance σ_j^2, the product $(\phi_{j,m}^R)^T\phi_{j,m}^R$ has the property

$$E[(\phi_{j,m}^R)^T \phi_{j,m}^R] = \sigma_j^2 I_N.$$

It follows that

$$E[(\phi_{j,m}^R)^T \phi_{j,m}^R x_j] = \sigma_j^2 E[x_j] = \sigma_j^2 E[z_C + z_j] = \sigma_j^2 z_C$$

and, similarly, that

$$E\left[\frac{1}{M_j\sigma_j^2}\sum_{m=1}^{M_j} (\phi_{j,m}^R)^T \phi_{j,m}^R x_j\right] = z_C.$$

Thus, $\widehat{z_C}$ is a sample mean of J independent random variables with mean z_C. From the LLN, we conclude that

$$\lim_{J\to\infty} \widehat{z_C} = z_C.$$

Part 2: Consider recovery of the innovation z_j from the adjusted measurement vector $\widehat{y_j} = y_j - \Phi_j \widehat{z_C}$. As a recovery scheme, we consider a combinatorial search over all K-sparse index sets drawn from $\{1, 2, \ldots, N\}$. For each such index set Ω', we compute the distance from $\widehat{y}$ to the column span of $\Phi_{j,\Omega'}$, denoted by $d(\widehat{y}, \text{colspan}(\Phi_{j,\Omega'}))$, where $\Phi_{j,\Omega'}$ is the matrix obtained by sampling the columns Ω' from Φ_j. (This distance can

be measured using the pseudoinverse of $\Phi_{j,\Omega'}$.)

For the correct index set Ω, we know that $d(\widehat{y_j}, \text{colspan}(\Phi_{j,\Omega})) \to 0$ as $J \to \infty$. For any other index set Ω', we know from the proof of Theorem 2.1 that $d(\widehat{y_j}, \text{colspan}(\Phi_{j,\Omega'})) > 0$. Let

$$\zeta \triangleq \min_{\Omega' \neq \Omega} d(\widehat{y_j}, \text{colspan}(\Phi_{i,\Omega'})).$$

With probability one, $\zeta > 0$. Thus for sufficiently large J, we will have

$$d(\widehat{y_j}, \text{colspan}(\Phi_{j,\Omega})) < \zeta/2,$$

and so the correct index set Ω can be correctly identified. $\square$

Appendix C
Proof of Theorem 6.2

A quick sketch of the proof is as follows. We first specify a high-resolution sampling of points on the manifold. At each of these points we consider the tangent space to the manifold and specify a sampling of points drawn from this space as well. We then employ the JL lemma to ensure an embedding with satisfactory preservation of all pairwise distances between these points.

Based on the preservation of these pairwise distances, we then ensure isometry for all tangents to the sampled points and then (using the bounded twisting of tangent spaces) ensure isometry for all tangents at all points on the manifold. From this (and using the bounded curvature) we ensure pairwise distance preservation between all nearby points on the manifold.

Finally we establish pairwise distance preservation between distant points on the manifold essentially by using the original pairwise distance preservation between the sample points (plus their nearby tangent points).

C.1 Preliminaries

For shorthand, we say a point $x \in \mathbb{R}^N$ has "compaction isometry ϵ" if the following condition is met:

$$(1-\epsilon)\sqrt{M/N}\,\|x\|_2 \leq \|\Phi x\|_2 \leq (1+\epsilon)\sqrt{M/N}\,\|x\|_2\,.$$

We say a set has compaction isometry ϵ if the above condition is met for every point in the set. We say a point $x \in \mathbb{R}^N$ has "squared compaction isometry ϵ" if the following condition is met:

$$(1-\epsilon)(M/N)\,\|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1+\epsilon)(M/N)\,\|x\|_2^2\,.$$

These notions are very similar — compaction isometry ϵ implies squared compaction isometry 3ϵ, and squared compaction isometry ϵ implies compaction isometry ϵ.

We also note that Φ is a nonexpanding operator (by which we mean that $\|\Phi\|_2 \leq 1$, i.e., $\|\Phi x\|_2 \leq \|x\|_2$ for all $x \in \mathbb{R}^N$).

We will also find the following inequalities useful throughout:

$$\frac{1}{1-s} \leq (1+2s),\ 0 \leq s \leq 1/2, \tag{C.1}$$

and

$$\frac{1}{1+s} \geq (1-s),\ s \geq 0. \tag{C.2}$$

C.2 Sampling the Manifold

Fix $T > 0$. (We later choose a value for T in Section C.7.) Let A be a minimal set of points on the manifold such that, for every $x \in \mathcal{M}$,

$$\min_{a \in A} d_{\mathcal{M}}(x, a) \leq T. \tag{C.3}$$

We call A the set of *anchor points*. From (2.1) we have that $\#A \leq \frac{RVK^{K/2}}{T^K}$.

C.3 Tangent Planes at the Anchor Points

Fix $\delta > 0$ and $\epsilon_1 \geq 2\delta$. (We later choose values for δ and ϵ_1 in Section C.7.) For each anchor point $a \in A$ we consider the tangent space Tan_a to $\mathcal{M}$ at a. We construct a covering of points $Q_1(a) \subset \mathrm{Tan}_a$ such that $\|q\|_2 \leq 1$ for all $q \in Q_1(a)$ and such that for every $u \in \mathrm{Tan}_a$ with $\|u\|_2 \leq 1$,

$$\min_{q \in Q_1(a)} \|u - q\|_2 \leq \delta.$$

This can be accomplished with $\#Q_1(a) \leq (3/\delta)^K$ (see e.g. Chapter 13 of [143]). We then define the renormalized set

$$Q_2(a) = \{Tq :\ q \in Q_1(a)\}$$

and note that $\|q\|_2 \leq T$ for all $q \in Q_2(a)$ and that for every $u \in \mathrm{Tan}_a$ with $\|u\|_2 \leq T$,

$$\min_{q \in Q_2(a)} \|u - q\|_2 \leq T\delta. \tag{C.4}$$

We now define the set

$$B = \bigcup_{a \in A} \{a\} \cup (a + Q_2(a)),$$

where $a + Q_2(a)$ denotes the set of tangents anchored at the point a (rather than at 0).

Now let $\beta = -\ln(\rho)$, set

$$M \geq \left(\frac{4 + 2\beta}{\epsilon_1^2/2 - \epsilon_1^3/3} \right) \ln(\#B), \tag{C.5}$$

and let Φ be as specified in Theorem 6.2. According to Lemma 2.5 (Johnson-

Lindenstrauss), with probability exceeding $1 - (\#B)^{-\beta} > 1 - \rho$, the following statement holds: For all $u, v \in B$, the difference vector $(u - v)$ has compaction isometry ϵ_1. We assume this to hold and must now extend it to show (6.2) for every $x, y \in \mathcal{M}$.

We immediately have that for every $a \in A$, every $q \in Q_2(a)$ has compaction isometry ϵ_1, and because Φ is linear, every $q \in Q_1(a)$ also has compaction isometry ϵ_1.

Following the derivation in Lemma 5.1 of [95] (and recalling that we assume $\delta \leq \epsilon_1/2$), we have that for all $a \in A$, the tangent space Tan_a has compaction isometry

$$\epsilon_2 := 2\epsilon_1.$$

That is, for every $a \in A$, every $u \in \text{Tan}_a$ has compaction isometry ϵ_2.

C.4 Tangent Planes at Arbitrary Points on the Manifold

Suppose $T/\tau < 1/4$. Let x be an arbitrary point on the manifold and let a be its nearest anchor point (in geodesic distance), recalling from (C.3) that $d_{\mathcal{M}}(x, a) \leq T$. Let $v \in \text{Tan}_x$ with $\|v\|_2 = 1$. From Lemma 2.2 it follows that there exists $u \in \text{Tan}_a$ such that $\|u\|_2 = 1$ and $\cos(\text{angle}(u, v)) > 1 - T/\tau$.

Because $\|u\|_2 = \|v\|_2 = 1$, it follows that $\|u - v\|_2 \leq \text{angle}(u, v)$. Define $\theta := \text{angle}(u, v)$; our bound above specifies that $\cos(\theta) > 1 - T/\tau$. Using a Taylor expansion we have that $\cos(\theta) < 1 - \theta^2/2 + \theta^4/24 = 1 - \theta^2/2(1 - \theta^2/12)$, and because we assume $T/\tau < 1/4$, then $\theta < 2$, which implies $\cos(\theta) < 1 - \theta^2/3$. Combining, we have $1 - \theta^2/3 > \cos(\theta) > 1 - T/\tau$, which implies that $T/\tau > \theta^2/3$, and so $\|u - v\|_2 \leq \theta < \sqrt{3T/\tau}$.

Since $u \in \text{Tan}_a$ with $a \in A$, we recall that u has compaction isometry ϵ_2. We aim to determine the compaction isometry for v. Using the triangle inequality and the fact that Φ is nonexpanding, we have $\|\Phi v\|_2 \leq \|\Phi u\|_2 + \|\Phi(u - v)\|_2 \leq (1 + \epsilon_2)\sqrt{M/N} + \sqrt{3T/\tau}$. Similarly, $\|\Phi v\|_2 \geq \|\Phi u\|_2 - \|\Phi(u - v)\|_2 \geq (1 - \epsilon_2)\sqrt{M/N} - \sqrt{3T/\tau}$. Since $\|v\|_2 = 1$, this implies that v has compaction isometry

$$\epsilon_3 := \epsilon_2 + \sqrt{\frac{3TN}{\tau M}}.$$

Because the choices of x and v were arbitrary, it follows that all tangents to the manifold have compaction isometry ϵ_3.

C.5 Differences Between Nearby Points on the Manifold

Let $C_1 > 0$. (We later choose a value for C_1 in Section C.7.) Suppose $C_1 T/\tau < 1/2$. Let x and y be two points on the manifold separated by geodesic distance $\mu := d_{\mathcal{M}}(x, y) \leq C_1 T$. Let $\gamma(t)$ denote a unit speed parametrization of the geodesic path connecting x and y, with $\gamma(0) = x$ and $\gamma(\mu) = y$.

Lemma 2.1 implies that the curvature of γ is bounded by $1/\tau$. From Taylor's theorem we then have that

$$\gamma(\mu) - \gamma(0) = \mu\gamma'(0) + R_1 \tag{C.6}$$

where $\gamma'(0)$ denotes the tangent to the curve γ at 0, and where the norm of the remainder obeys $\|R_1\|_2 \leq \mu^2/\tau$. Using the triangle inequality and the fact that $\|\gamma'(0)\|_2 = 1$, we have that

$$(1 - \mu/\tau)\mu \leq \|\gamma(\mu) - \gamma(0)\|_2 \leq (1 + \mu/\tau)\mu, \tag{C.7}$$

and combining (C.6) with the compaction isometry ϵ_3 of $\gamma'(0)$ and the fact that Φ is nonexpanding we have

$$(1 - (\epsilon_3 + \mu\sqrt{N/M}/\tau))\mu\sqrt{M/N} \leq \|\Phi\gamma(\mu) - \Phi\gamma(0)\|_2 \leq (1 + (\epsilon_3 + \mu\sqrt{N/M}/\tau))\mu\sqrt{M/N}. \tag{C.8}$$

Combining (C.7) and (C.8), the ratio

$$\begin{aligned}
\frac{\|\Phi\gamma(\mu) - \Phi\gamma(0)\|_2}{\|\gamma(\mu) - \gamma(0)\|_2} &\leq \frac{(1 + \epsilon_3 + \mu\sqrt{N/M}/\tau)\mu\sqrt{M/N}}{(1 - \mu/\tau)\mu} \\
&= \frac{(1 + \epsilon_3 + \mu\sqrt{N/M}/\tau)}{(1 - \mu/\tau)}\sqrt{M/N} \\
&\leq \frac{(1 + \epsilon_3 + C_1T\sqrt{N/M}/\tau)}{(1 - C_1T/\tau)}\sqrt{M/N} \\
&\leq (1 + \epsilon_3 + C_1T\sqrt{N/M}/\tau)(1 + 2C_1T/\tau)\sqrt{M/N} \\
&= (1 + \epsilon_3 + C_1T\sqrt{N/M}/\tau + 2C_1T/\tau \\
&\quad + 2\epsilon_3C_1T/\tau + 2C_1^2T^2\sqrt{N/M}/\tau^2)\sqrt{M/N}.
\end{aligned}$$

In the fourth step above we have employed (C.1) and the fact that $C_1T/\tau < 1/2$.

Similarly, the ratio

$$\begin{aligned}
\frac{\|\Phi\gamma(\mu) - \Phi\gamma(0)\|_2}{\|\gamma(\mu) - \gamma(0)\|_2} &\geq \frac{(1 - \epsilon_3 - \mu\sqrt{N/M}/\tau)\mu\sqrt{M/N}}{(1 + \mu/\tau)\mu} \\
&= \frac{(1 - \epsilon_3 - \mu\sqrt{N/M}/\tau)}{(1 + \mu/\tau)}\sqrt{M/N} \\
&\geq \frac{(1 - \epsilon_3 - C_1T\sqrt{N/M}/\tau)}{(1 + C_1T/\tau)}\sqrt{M/N} \\
&\geq (1 - \epsilon_3 - C_1T\sqrt{N/M}/\tau)(1 - C_1T/\tau)\sqrt{M/N} \\
&= (1 - \epsilon_3 - C_1T\sqrt{N/M}/\tau - C_1T/\tau \\
&\quad + \epsilon_3C_1T/\tau + C_1^2T^2\sqrt{N/M}/\tau^2)\sqrt{M/N} \\
&\geq (1 - \epsilon_3 - C_1T\sqrt{N/M}/\tau - C_1T/\tau)\sqrt{M/N}.
\end{aligned}$$

Here the fourth step uses (C.2). Of the bounds we have now derived, the upper bound is the looser of the two, and so it follows that the difference vector $\gamma(\mu) - \gamma(0) = y - x$ has compaction isometry

$$\epsilon_4 := \epsilon_3 + C_1T\sqrt{N/M}/\tau + 2C_1T/\tau + 2\epsilon_3C_1T/\tau + 2C_1^2T^2\sqrt{N/M}/\tau^2.$$

This compaction isometry ϵ_4 will hold for any two points on the manifold separated by geodesic distance $\leq C_1T$.

C.6 Differences Between Distant Points on the Manifold

Suppose $C_1 \geq 10$, $T \leq \tau/C_1$, and $\delta \leq 1/4$. Let x_1 and x_2 be two points on the manifold separated by geodesic distance $d_{\mathcal{M}}(x_1, x_2) > C_1T$. Let a_1 and a_2 be the nearest (in terms of geodesic distance) anchor points to x_1 and x_2, respectively.

We consider the geodesic path from a_1 to x_1 and let $u_1 \in \mathrm{Tan}_{a_1}$ denote the tangent to this path at a_1. (For convenience we scale u_1 to have norm $\|u_1\|_2 = T$.) Similarly, we let $u_2 \in \mathrm{Tan}_{a_2}$ denote the tangent at the start of the geodesic path from a_2 to x_2 (choosing $\|u_2\|_2 = T$).

We recall from (C.4) that there exists $q_1 \in Q_2(a_1)$ such that $\|u_1 - q_1\|_2 \leq T\delta$ and there exists $q_2 \in Q_2(a_2)$ such that $\|u_2 - q_2\|_2 \leq T\delta$. Additionally, the points $a_1 + q_1$ and $a_2 + q_2$ belong to the set B, and so the difference $(a_1 + q_1) - (a_2 + q_2)$ has compaction isometry ϵ_1.

Recalling the assumption that $T \leq \tau/C_1$, we consider the ambient distance between x_1 and x_2. We have either that $\|x_1 - x_2\|_2 > \tau/2 \geq C_1T/2$ or that $\|x_1 - x_2\|_2 \leq \tau/2$, which by Corollary 2.1 would then imply that $\|x_1 - x_2\|_2 \geq d_{\mathcal{M}}(x_1, x_2) -$

$\frac{(d_{\mathcal{M}}(x_1,x_2))^2}{2\tau}$ with $d_{\mathcal{M}}(x_1,x_2) > C_1T$ by assumption and

$$\begin{aligned} d_{\mathcal{M}}(x_1,x_2) &\leq \tau - \tau\sqrt{1 - 2\left\|x_1 - x_2\right\|_2/\tau} \\ &\leq \tau(1 - (1 - 2\left\|x_1 - x_2\right\|_2/\tau)) = 2\left\|x_1 - x_2\right\|_2 \leq \tau \end{aligned}$$

by Lemma 2.3. In this range $C_1T < d_{\mathcal{M}}(x_1,x_2) \leq \tau$, it follows that $\|x_1 - x_2\|_2 \geq d_{\mathcal{M}}(x_1,x_2) - \frac{(d_{\mathcal{M}}(x_1,x_2))^2}{2\tau} > C_1T/2$. Since we assume $C_1 \geq 10$, then $\|x_1 - x_2\|_2 > 5T$. Using the triangle inequality, $\|a_1 - a_2\|_2 > 3T$ and $\|(a_1 + q_1) - (a_2 + q_2)\|_2 > T$.

Now we consider the compaction isometry of $(a_1 + u_1) - (a_2 + u_2)$. Using the triangle inequality and the fact that Φ is nonexpanding, we have

$$\begin{aligned} \frac{\|\Phi(a_1 + u_1) - \Phi(a_2 + u_2)\|_2}{\|(a_1 + u_1) - (a_2 + u_2)\|_2} &\leq \frac{\|\Phi(a_1 + q_1) - \Phi(a_2 + q_2)\|_2 + 2T\delta}{\|(a_1 + q_1) - (a_2 + q_2)\|_2 - 2T\delta} \\ &\leq \frac{(1+\epsilon_1)\|(a_1 + q_1) - (a_2 + q_2)\|_2\sqrt{M/N} + 2T\delta}{\|(a_1 + q_1) - (a_2 + q_2)\|_2 - 2T\delta} \\ &= \frac{(1+\epsilon_1)\sqrt{M/N} + 2T\delta/\|(a_1 + q_1) - (a_2 + q_2)\|_2}{1 - 2T\delta/\|(a_1 + q_1) - (a_2 + q_2)\|_2} \\ &< \frac{(1+\epsilon_1)\sqrt{M/N} + 2\delta}{1 - 2\delta} \\ &\leq ((1+\epsilon_1)\sqrt{M/N} + 2\delta)(1 + 4\delta) \\ &= (1+\epsilon_1)\sqrt{M/N} + 2\delta + (1+\epsilon_1)4\delta\sqrt{M/N} + 8\delta^2 \\ &= (1 + \epsilon_1 + 4\delta + 4\delta\epsilon_1 \\ &\quad + 2\delta\sqrt{N/M} + 8\delta^2\sqrt{N/M})\sqrt{M/N}. \end{aligned}$$

The fifth step above uses (C.1) and assumes $\delta \leq 1/4$. Similarly,

$$\begin{aligned}
\frac{\|\Phi(a_1+u_1)-\Phi(a_2+u_2)\|_2}{\|(a_1+u_1)-(a_2+u_2)\|_2} &\geq \frac{\|\Phi(a_1+q_1)-\Phi(a_2+q_2)\|_2 - 2T\delta}{\|(a_1+q_1)-(a_2+q_2)\|_2 + 2T\delta} \\
&\geq \frac{(1-\epsilon_1)\,\|(a_1+q_1)-(a_2+q_2)\|_2\sqrt{M/N} - 2T\delta}{\|(a_1+q_1)-(a_2+q_2)\|_2 + 2T\delta} \\
&= \frac{(1-\epsilon_1)\sqrt{M/N} - 2T\delta/\,\|(a_1+q_1)-(a_2+q_2)\|_2}{1 + 2T\delta/\,\|(a_1+q_1)-(a_2+q_2)\|_2} \\
&> \frac{(1-\epsilon_1)\sqrt{M/N} - 2\delta}{1+2\delta} \\
&\geq ((1-\epsilon_1)\sqrt{M/N} - 2\delta)(1-2\delta) \\
&= (1-\epsilon_1)\sqrt{M/N} - 2\delta - (1-\epsilon_1)2\delta\sqrt{M/N} + 4\delta^2 \\
&= (1-\epsilon_1-2\delta+2\delta\epsilon_1 \\
&\quad -2\delta\sqrt{N/M} + 4\delta^2\sqrt{N/M})\sqrt{M/N} \\
&> (1-\epsilon_1-2\delta-2\delta\sqrt{N/M})\sqrt{M/N}.
\end{aligned}$$

Here the fifth step uses (C.2). Of the bounds we have now derived, the upper bound is the looser of the two, and so the difference vector $(a_1+u_1)-(a_2+u_2)$ has compaction isometry

$$\epsilon_5 := \epsilon_1 + 4\delta + 4\delta\epsilon_1 + 2\delta\sqrt{N/M} + 8\delta^2\sqrt{N/M}.$$

Using very similar arguments one can show that the difference vectors $a_1-(a_2+u_2)$ and $(a_1+u_1)-a_2$ also have compaction isometry ϵ_5.

Define $b_i = a_i + u_i$, $\mu_i = d_{\mathcal{M}}(a_i, x_i)$, and $c_i = a_i + (\mu_i/T)u_i$ for $i = 1, 2$. The points c_i represent traversals of length μ_i along the tangent path rather than the geodesic path from a_i to x_i; they can also be expressed as the linear combination

$$c_i = (1-\mu_i/T)a_i + (\mu_i/T)b_i,\ \ i = 1, 2. \tag{C.9}$$

We have established above that all pairwise differences of vectors from the set $\{a_1, a_2, b_1, b_2\}$ have compaction isometry ϵ_5. As we recall from Section C.1, this implies squared compaction isometry $3\epsilon_5$ for each of these difference vectors. We now use this fact to establish a similar bound for the difference $c_1 - c_2$. First, we can express the distance $\|c_1-c_2\|_2^2$ in terms of the distances between the a_i's and b_i's. Define

$$\begin{aligned}
d_{\text{cross}} &= (\mu_1/T)(\mu_2/T)\,\|b_1-b_2\|_2^2 + (1-\mu_1/T)(\mu_2/T)\,\|a_1-b_2\|_2^2 \\
&\quad +(\mu_1/T)(1-\mu_2/T)\,\|b_1-a_2\|_2^2 + (1-\mu_1/T)(1-\mu_2/T)\,\|a_1-a_2\|_2^2
\end{aligned}$$

and

$$d_{\text{local}} = (\mu_1/T)(1-\mu_1/T)\,\|a_1-b_1\|_2^2 + (\mu_2/T)(1-\mu_2/T)\,\|a_2-b_2\|_2^2\,.$$

Then we can use (C.9) to show that

$$\|c_1 - c_2\|_2^2 = d_{\text{cross}} - d_{\text{local}}.$$

Noting that $\|a_1 - b_1\|_2^2 = \|a_2 - b_2\|_2^2 = T^2$, we have that $d_{\text{local}} \leq T^2/2$. Because $\|x_1 - x_2\|_2 > 5T$, a_1 and b_1 are at least distance T from each of a_2 and b_2, which implies that $d_{\text{cross}} > T^2 \geq 2d_{\text{local}}$. We will use this fact below. We can also express

$$\Phi c_i = (1 - \tau_i/T)\Phi a_i + (\tau_i/T)\Phi b_i,\ i = 1, 2,$$

define

$$\begin{aligned}\widehat{d_{\text{cross}}} &= (\mu_1/T)(\mu_2/T)\|\Phi b_1 - \Phi b_2\|_2^2 + (1 - \mu_1/T)(\mu_2/T)\|\Phi a_1 - \Phi b_2\|_2^2 \\ &\quad +(\mu_1/T)(1 - \mu_2/T)\|\Phi b_1 - \Phi a_2\|_2^2 + (1 - \mu_1/T)(1 - \mu_2/T)\|\Phi a_1 - \Phi a_2\|_2^2\end{aligned}$$

and

$$\widehat{d_{\text{local}}} = (\mu_1/T)(1 - \mu_1/T)\|\Phi a_1 - \Phi b_1\|_2^2 + (\mu_2/T)(1 - \mu_2/T)\|\Phi a_2 - \Phi b_2\|_2^2,$$

and establish that

$$\|\Phi c_1 - \Phi c_2\|_2^2 = \widehat{d_{\text{cross}}} - \widehat{d_{\text{local}}}.$$

Using the squared compaction isometry of all pairwise differences of a_1, a_2, b_1, and b_2, we have that

$$\begin{aligned}\|\Phi c_1 - \Phi c_2\|_2^2 &= \widehat{d_{\text{cross}}} - \widehat{d_{\text{local}}} \\ &\leq (1 + 3\epsilon_5)(M/N)d_{\text{cross}} - (1 - 3\epsilon_5)(M/N)d_{\text{local}} \\ &= \left(1 + 3\epsilon_5 + 6\epsilon_5\left(\frac{d_{\text{local}}}{d_{\text{cross}} - d_{\text{local}}}\right)\right)(M/N)(d_{\text{cross}} - d_{\text{local}}) \\ &< (1 + 9\epsilon_5)(M/N)\|c_1 - c_2\|_2^2.\end{aligned}$$

For the last inequality we used the fact that $d_{\text{cross}} > 2d_{\text{local}}$. Similarly, we have that

$$\|\Phi c_1 - \Phi c_2\|_2^2 > (1 - 9\epsilon_5)(M/N)\|c_1 - c_2\|_2^2.$$

Combining, these imply squared compaction isometry $9\epsilon_5$ for the vector $c_1 - c_2$, which also implies compaction isometry $9\epsilon_5$ for $c_1 - c_2$.

Finally, we are ready to compute the compaction isometry for the vector $x_1 - x_2$. Using Taylor's theorem anchored at the points a_i, we have $\|x_i - c_i\|_2 \leq \mu_i^2/\tau \leq$

$T^2/\tau,\ i = 1, 2$. Using the triangle inequality we also have that $\|c_1 - c_2\|_2 > T$. Thus

$$\begin{aligned}
\frac{\|\Phi x_1 - \Phi x_2\|_2}{\|x_1 - x_2\|_2} &\leq \frac{(1+9\epsilon_5)\sqrt{M/N}\,\|c_1 - c_2\|_2 + 2T^2/\tau}{\|c_1 - c_2\|_2 - 2T^2/\tau} \\
&= \left(1 + \frac{9\epsilon_5 + 2T^2/(\tau\,\|c_1 - c_2\|_2) + 2T^2\sqrt{\frac{M}{N}}/(\tau\,\|c_1 - c_2\|_2)}{1 - 2T^2/(\tau\,\|c_1 - c_2\|_2)}\right)\sqrt{\frac{M}{N}} \\
&\leq \left(1 + \frac{9\epsilon_5 + 2T/\tau + 2T\sqrt{N/M}/\tau}{1 - 2T/\tau}\right)\sqrt{\frac{M}{N}}.
\end{aligned}$$

Similarly,

$$\begin{aligned}
\frac{\|\Phi x_1 - \Phi x_2\|_2}{\|x_1 - x_2\|_2} &\geq \frac{(1-9\epsilon_5)\sqrt{M/N}\,\|c_1 - c_2\|_2 - 2T^2/\tau}{\|c_1 - c_2\|_2 + 2T^2/\tau} \\
&= \left(1 - \frac{9\epsilon_5 + 2T^2/(\tau\,\|c_1 - c_2\|_2) + 2T^2\sqrt{\frac{M}{N}}/(\tau\,\|c_1 - c_2\|_2)}{1 + 2T^2/(\tau\,\|c_1 - c_2\|_2)}\right)\sqrt{\frac{M}{N}} \\
&\geq \Big(1 - (9\epsilon_5 + 2T/\tau + 2T\sqrt{N/M}/\tau)\Big)\sqrt{M/N}.
\end{aligned}$$

Considering both bounds, we have

$$\begin{aligned}
9\epsilon_5 + 2T/\tau + 2T\sqrt{N/M}/\tau &\leq \frac{9\epsilon_5 + 2T/\tau + 2T\sqrt{N/M}/\tau}{1 - 2T/\tau} \\
&\leq (9\epsilon_5 + 2T/\tau + 2T\sqrt{N/M}/\tau)(1 + 4T/\tau).
\end{aligned}$$

(For the second inequality, we use the assumption that $T/\tau < 1/4$.) Hence, $x_1 - x_2$ has compaction isometry

$$\epsilon_6 := 9\epsilon_5 + \frac{36\epsilon_5 T}{\tau} + \frac{2T}{\tau} + \frac{8T^2}{\tau^2} + \frac{2T\sqrt{N/M}}{\tau} + \frac{8T^2\sqrt{N/M}}{\tau^2}.$$

C.7 Synthesis

Let $0 < \epsilon < 1$ be the desired compaction isometry for all pairwise distances on the manifold. In the preceding sections, we have established the following compaction

isometries. For nearby points we have compaction isometry

$$\begin{aligned}
\epsilon_4 &= \epsilon_3 + \frac{C_1 T}{\tau}\sqrt{\frac{N}{M}} + \frac{2C_1 T}{\tau} + \frac{2\epsilon_3 C_1 T}{\tau} + \frac{2C_1^2 T^2}{\tau^2}\sqrt{\frac{N}{M}} \\
&= \epsilon_2 + \sqrt{\frac{3TN}{\tau M}} + \frac{C_1 T}{\tau}\sqrt{\frac{N}{M}} + \frac{2C_1 T}{\tau} \\
&\quad +2\left(\epsilon_2 + \sqrt{\frac{3TN}{\tau M}}\right)\left(\frac{C_1 T}{\tau}\right) + \frac{2C_1^2 T^2}{\tau^2}\sqrt{\frac{N}{M}} \\
&= 2\epsilon_1 + \sqrt{\frac{3TN}{\tau M}} + \frac{C_1 T}{\tau}\sqrt{\frac{N}{M}} + \frac{2C_1 T}{\tau} \\
&\quad +2\left(2\epsilon_1 + \sqrt{\frac{3TN}{\tau M}}\right)\left(\frac{C_1 T}{\tau}\right) + \frac{2C_1^2 T^2}{\tau^2}\sqrt{\frac{N}{M}} \\
&= 2\epsilon_1 + \frac{4\epsilon_1 C_1 T}{\tau} + \sqrt{\frac{3TN}{\tau M}} + \frac{C_1 T}{\tau}\sqrt{\frac{N}{M}} \\
&\quad +\frac{2C_1 T}{\tau} + \frac{2C_1 T}{\tau}\sqrt{\frac{3TN}{\tau M}} + \frac{2C_1^2 T^2}{\tau^2}\sqrt{\frac{N}{M}}.
\end{aligned}$$

For distant points we have compaction isometry

$$\begin{aligned}
\epsilon_6 &= 9\epsilon_5 + \frac{36\epsilon_5 T}{\tau} + \frac{2T}{\tau} + \frac{8T^2}{\tau^2} + \frac{2T}{\tau}\sqrt{\frac{N}{M}} + \frac{8T^2}{\tau^2}\sqrt{\frac{N}{M}} \\
&= 9(\epsilon_1 + 4\delta + 4\delta\epsilon_1 + 2\delta\sqrt{N/M} + 8\delta^2\sqrt{N/M}) \\
&\quad +\frac{36(\epsilon_1 + 4\delta + 4\delta\epsilon_1 + 2\delta\sqrt{N/M} + 8\delta^2\sqrt{N/M})T}{\tau} \\
&\quad +\frac{2T}{\tau} + \frac{8T^2}{\tau^2} + \frac{2T}{\tau}\sqrt{\frac{N}{M}} + \frac{8T^2}{\tau^2}\sqrt{\frac{N}{M}} \\
&= 9\epsilon_1 + 36\delta + 36\delta\epsilon_1 + 18\delta\sqrt{N/M} + 72\delta^2\sqrt{N/M} \\
&\quad +\frac{36\epsilon_1 T}{\tau} + \frac{144\delta T}{\tau} + \frac{144\delta\epsilon_1 T}{\tau} + \frac{72\delta T}{\tau}\sqrt{\frac{N}{M}} + \frac{288\delta^2 T}{\tau}\sqrt{\frac{N}{M}} \\
&\quad +\frac{2T}{\tau} + \frac{8T^2}{\tau^2} + \frac{2T}{\tau}\sqrt{\frac{N}{M}} + \frac{8T^2}{\tau^2}\sqrt{\frac{N}{M}}.
\end{aligned}$$

We will now choose values for C_1, ϵ_1, T, and δ that will ensure compaction isometry ϵ for all pairwise distances on the manifold. We first set $C_1 = 10$. For constants C_2, C_3, and C_4 (which we will soon specify), we let

$$\epsilon_1 = C_2\epsilon, \quad T = \frac{C_3\epsilon^2\tau}{N}, \quad \text{and} \quad \delta = \frac{C_4\epsilon}{\sqrt{N}}.$$

Plugging in to the above and using the fact that $\epsilon < 1$, we have

$$\begin{aligned}
\epsilon_4 &\leq 2\epsilon_1 + \frac{4\epsilon_1 C_1 T}{\tau} + \sqrt{\frac{3TN}{\tau M}} + \frac{C_1 T}{\tau}\sqrt{\frac{N}{M}} \\
&\quad + \frac{2C_1 T}{\tau} + \frac{2C_1 T}{\tau}\sqrt{\frac{3TN}{\tau M}} + \frac{2C_1^2 T^2}{\tau^2}\sqrt{\frac{N}{M}} \\
&= 2C_2\epsilon + \frac{40C_2 C_3 \epsilon^3}{N} + \sqrt{\frac{3C_3\epsilon^2}{M}} + \frac{10C_3\epsilon^2}{N}\sqrt{\frac{N}{M}} \\
&\quad + \frac{20C_3\epsilon^2}{N} + \frac{20C_3\epsilon^2}{N}\sqrt{\frac{3C_3\epsilon^2}{M}} + \frac{200C_3^2\epsilon^4}{N^2}\sqrt{\frac{N}{M}} \\
&\leq \epsilon(2C_2 + 40C_2C_3 + \sqrt{3C_3} + 30C_3 + 20\sqrt{3}C_3\sqrt{C_3} + 200C_3^2)
\end{aligned}$$

and

$$\begin{aligned}
\epsilon_6 &\leq 9\epsilon_1 + 36\delta + 36\delta\epsilon_1 + 18\delta\sqrt{N/M} + 72\delta^2\sqrt{N/M} \\
&\quad + \frac{36\epsilon_1 T}{\tau} + \frac{144\delta T}{\tau} + \frac{144\delta\epsilon_1 T}{\tau} + \frac{72\delta T}{\tau}\sqrt{\frac{N}{M}} + \frac{288\delta^2 T}{\tau}\sqrt{\frac{N}{M}} \\
&\quad + \frac{2T}{\tau} + \frac{8T^2}{\tau^2} + \frac{2T}{\tau}\sqrt{\frac{N}{M}} + \frac{8T^2}{\tau^2}\sqrt{\frac{N}{M}} \\
&= 9C_2\epsilon + \frac{36C_4\epsilon}{\sqrt{N}} + \frac{36C_2C_4\epsilon^2}{\sqrt{N}} + \frac{18C_4\epsilon}{\sqrt{M}} + \frac{72C_4^2\epsilon^2}{\sqrt{NM}} \\
&\quad + \frac{36C_2C_3\epsilon^3}{N} + \frac{144C_3C_4\epsilon^3}{N\sqrt{N}} + \frac{144C_2C_3C_4\epsilon^4}{N\sqrt{N}} + \frac{72C_3C_4\epsilon^3}{N\sqrt{M}} + \frac{288C_3C_4^2\epsilon^4}{N\sqrt{NM}} \\
&\quad + \frac{2C_3\epsilon^2}{N} + \frac{8C_3^2\epsilon^4}{N^2} + \frac{2C_3\epsilon^2}{\sqrt{NM}} + \frac{8C_3^2\epsilon^4}{N\sqrt{NM}} \\
&\leq \epsilon(9C_2 + 36C_4 + 36C_2C_4 + 18C_4 + 72C_4^2 + 36C_2C_3 + 144C_3C_4 \\
&\qquad + 144C_2C_3C_4 + 72C_3C_4 + 288C_3C_4^2 + 2C_3 + 8C_3^2 + 2C_3 + 8C_3^2).
\end{aligned}$$

We now must set the constants C_2, C_3, and C_4 to ensure that $\epsilon_4 \leq \epsilon$ and $\epsilon_6 \leq \epsilon$. Due to the role of ϵ_1 in determining our ultimate bound on M, we wish to be most aggressive in setting the constant C_2. To ensure $\epsilon_6 \leq \epsilon$, we must set $C_2 < 1/9$; for neatness we choose $C_2 = 1/10$. For the remaining constants we may choose $C_3 = 1/1900$ and $C_4 = 1/633$ and confirm that both $\epsilon_4 \leq \epsilon$ and $\epsilon_6 \leq \epsilon$. One may also verify that, by using these constants, all of our assumptions at the beginning of each section are met (in particular, that $\epsilon_1 \geq 2\delta$, $T/\tau < 1/4$, $C_1 T/\tau < 1/2$, $C_1 \geq 10$, $T \leq \tau/C_1$, and $\delta \leq 1/4$).

To determine the requisite number of random projections, we must determine the

size of the set B. We have

$$\begin{aligned}
\#B &\leq \sum_{a\in A}(1+\#Q_2(a)) = \sum_{a\in A}(1+\#Q_1(a)) \\
&\leq \left(\frac{RVK^{K/2}}{T^K}\right)(1+(3/\delta)^K) \\
&\leq \left(\frac{RVK^{K/2}}{T^K}\right)\left(1+\left(\frac{3\cdot 633\sqrt{N}}{\epsilon}\right)^K\right) \\
&\leq \left(\frac{RVK^{K/2}1900^K N^K}{\epsilon^{2K}\tau^K}\right)\left(\frac{(3\cdot 633+1)^K N^{K/2}}{\epsilon^K}\right).
\end{aligned}$$

Plugging in to (C.5), we require

$$\begin{aligned}
M &\geq \left(\frac{4+2\beta}{\epsilon_1^2/2-\epsilon_1^3/3}\right)\ln(\#B) \\
&\geq \left(\frac{4-2\ln(\rho)}{\epsilon^2/200-\epsilon^3/3000}\right)\ln\left(\frac{1900^{2K}K^{K/2}N^{3K/2}RV}{\epsilon^{3K}\tau^K}\right). \qquad \text{(C.10)}
\end{aligned}$$

This completes the proof of Theorem 6.2. □

Appendix D
Proof of Corollary 6.1

The corollary follows simply from the fact that length of a smooth curve on the manifold can be written as a limit sum of ambient distances between points on that curve and the observation that (6.2) can be applied to each of these distances.

So if we let $x, y \in \mathcal{M}$, define $\mu = d_{\mathcal{M}}(x, y)$, and let γ denote the unit speed geodesic path joining x and y on $\mathcal{M}$ in $\mathbb{R}^N$, then the length of the image of γ along $\Phi\mathcal{M}$ in $\mathbb{R}^M$ will be bounded above by $(1+\epsilon)\sqrt{M/N}\mu$. Hence, $d_{\Phi\mathcal{M}}(\Phi x, \Phi y) \leq (1+\epsilon)\sqrt{M/N}d_{\mathcal{M}}(x, y)$.

Similarly, if we let $x, y \in \mathcal{M}$, define $\mu_\Phi = d_{\Phi\mathcal{M}}(\Phi x, \Phi y)$, and let γ_Φ denote the unit speed geodesic path joining Φx and Φy on the image of $\mathcal{M}$ in $\mathbb{R}^M$, then the length of the preimage of γ_Φ is bounded above by $\frac{1}{1-\epsilon}\sqrt{N/M}\mu_\Phi$. Hence,

$$d_{\mathcal{M}}(x, y) \leq \frac{1}{1-\epsilon}\sqrt{N/M}\mu_\Phi,$$

which implies that $d_{\Phi\mathcal{M}}(\Phi x, \Phi y) \geq (1-\epsilon)\sqrt{M/N}d_{\mathcal{M}}(x, y)$. □

Appendix E
Proof of Theorem 6.3

Fix $0 < \alpha \leq 1$. We consider two points in $w_a, w_b \in \mathbb{R}^N$ whose distance is compacted by a factor α under Φ, i.e.,

$$\frac{\|\Phi w_a - \Phi w_b\|_2}{\|w_a - w_b\|_2} = \alpha,$$

and supposing that x is closer to w_a, i.e.,

$$\|x - w_a\|_2 \leq \|x - w_b\|_2 ,$$

but Φx is closer to Φw_b, i.e.,

$$\|\Phi x - \Phi w_b\|_2 \leq \|\Phi x - \Phi w_a\|_2 ,$$

we seek the maximum value that

$$\frac{\|x - w_b\|_2}{\|x - w_a\|_2}$$

may take. In other words, we wish to bound the worst possible "mistake" (according to our error criterion) between two candidate points whose distance is compacted by the factor α. Note that all norms in this proof are ℓ_2-norms.

We have the optimization problem

$$\max_{x, w_a, w_b \in \mathbb{R}^N} \frac{\|x - w_b\|_2}{\|x - w_a\|_2} \quad \text{s.t.} \quad \begin{aligned} &\|x - w_a\|_2 \leq \|x - w_b\|_2 , \\ &\|\Phi x - \Phi w_b\|_2 \leq \|\Phi x - \Phi w_a\|_2 , \\ &\frac{\|\Phi w_a - \Phi w_b\|_2}{\|w_a - w_b\|_2} = \alpha. \end{aligned}$$

The constraints and objective function are invariant to adding a constant to all three variables or to a constant rescaling of all three. Hence, without loss of generality, we

set $w_a = \mathbf{0}$ and $\|x\|_2 = 1$. This leaves

$$\begin{aligned}\max_{x,w_b\in\mathbb{R}^N} \|x - w_b\|_2 \quad \text{s.t.} \quad & \|x\|_2 = 1, \\ & \|x - w_b\|_2 \geq \|x\|_2 , \\ & \|\Phi x - \Phi w_b\|_2 \leq \|\Phi x\|_2 , \\ & \frac{\|\Phi w_b\|_2}{\|w_b\|_2} = \alpha.\end{aligned}$$

We may safely ignore the second constraint (because of its relation to the objective function), and we may also square the objective function (to be later undone). We now consider the projection operator and its orthogonal complement separately, noting that $\|w\|_2^2 = \|\Phi w\|_2^2 + \|(I-\Phi)w\|_2^2$. This leads to

$$\max_{x,w_b\in\mathbb{R}^N} \|\Phi x - \Phi w_b\|_2^2 + \|(I-\Phi)x - (I-\Phi)w_b\|_2^2$$

subject to

$$\begin{aligned}&\|\Phi x\|_2^2 + \|(I-\Phi)x\|_2^2 = 1, \\ &\|\Phi x - \Phi w_b\|_2^2 \leq \|\Phi x\|_2^2 , \\ &\frac{\|\Phi w_b\|_2^2}{\|\Phi w_b\|_2^2 + \|(I-\Phi)w_b\|_2^2} = \alpha^2.\end{aligned}$$

We note that the Φ and $(I-\Phi)$ components of each vector may be optimized separately (subject to the listed constraints), again because they are orthogonal components. Now, rewriting the last constraint,

$$\max_{x,w_b\in\mathbb{R}^N} \|\Phi x - \Phi w_b\|_2^2 + \|(I-\Phi)x - (I-\Phi)w_b\|_2^2$$

subject to

$$\begin{aligned}&\|\Phi x\|_2^2 + \|(I-\Phi)x\|_2^2 = 1, \\ &\|\Phi x - \Phi w_b\|_2^2 \leq \|\Phi x\|_2^2 , \\ &\|(I-\Phi)w_b\|_2^2 = \|\Phi w_b\|_2^2 \left(\frac{1}{\alpha^2} - 1\right).\end{aligned}$$

Define β to be the value of $\|(I-\Phi)w_b\|_2$ taken for the optimal solution w_b. We note that the constraints refer to the norm of the vector $(I-\Phi)w_b$ but not its direction. To maximize the objective function, then, $(I-\Phi)w_b$ must be parallel (but with the opposite sign) to $(I-\Phi)x$. Equivalently, it must follow that

$$(I-\Phi)w_b = -\beta \cdot \frac{(I-\Phi)x}{\|(I-\Phi)x\|_2}. \tag{E.1}$$

We now consider the second term in the objective function. From (E.1), it follows that

$$\begin{aligned}\|(I-\Phi)x-(I-\Phi)w_b\|_2^2 &= \left\|(I-\Phi)x\left(1+\frac{\beta}{\|(I-\Phi)x\|_2}\right)\right\|_2^2 \\ &= \|(I-\Phi)x\|_2^2\cdot\left(1+\frac{\beta}{\|(I-\Phi)x\|_2}\right)^2. \qquad \text{(E.2)}\end{aligned}$$

The third constraint also demands that

$$\beta^2 = \|\Phi w_b\|_2^2\left(\frac{1}{\alpha^2}-1\right).$$

Substituting into (E.2), we have

$$\begin{aligned}\|(I-\Phi)x-(I-\Phi)w_b\|_2^2 &= \|(I-\Phi)x\|_2^2\cdot\left(1+2\frac{\beta}{\|(I-\Phi)x\|_2}+\frac{\beta^2}{\|(I-\Phi)x\|_2^2}\right) \\ &= \|(I-\Phi)x\|_2^2+2\|(I-\Phi)x\|_2\|\Phi w_b\|_2\sqrt{\frac{1}{\alpha^2}-1} \\ &\quad +\|\Phi w_b\|_2^2\left(\frac{1}{\alpha^2}-1\right).\end{aligned}$$

This is an increasing function of $\|\Phi w_b\|_2$, and so we seek the maximum value that $\|\Phi w_b\|_2$ may take subject to the constraints. From the second constraint we see that $\|\Phi x-\Phi w_b\|_2^2 \le \|\Phi x\|_2^2$; thus, $\|\Phi w_b\|_2$ is maximized by letting $\Phi w_b = 2\Phi x$. With such a choice of Φw_b we then have

$$\|\Phi x-\Phi w_b\|_2^2 = \|\Phi x\|_2^2$$

We note that this choice of Φw_b *also* maximizes the first term of the objective function subject to the constraints.

We may now rewrite the optimization problem, in light of the above restrictions:

$$\max_{\Phi x,(I-\Phi)x} \|\Phi x\|_2^2+\|(I-\Phi)x\|_2^2+4\|\Phi x\|_2\|(I-\Phi)x\|_2\sqrt{\frac{1}{\alpha^2}-1}+4\|\Phi x\|_2^2\left(\frac{1}{\alpha^2}-1\right)$$
$$\text{s.t. } \|\Phi x\|_2^2+\|(I-\Phi)x\|_2^2=1.$$

We now seek to bound the maximum value that the objective function may take. We note that the single constraint implies that

$$\|\Phi x\|_2\|(I-\Phi)x\|_2 \le \frac{1}{2}$$

and that $\|\Phi x\|_2 \leq 1$ (but because these cannot be simultaneously met with equality, our bound will not be tight). It follows that

$$\begin{aligned}\|\Phi x\|_2^2 + \|(I-\Phi)x\|_2^2 &+ 4\|\Phi x\|_2 \|(I-\Phi)x\|_2 \sqrt{\frac{1}{\alpha^2}-1} + 4\|\Phi x\|_2^2 \left(\frac{1}{\alpha^2}-1\right) \\ &\leq 1 + 2\sqrt{\frac{1}{\alpha^2}-1} + 4\left(\frac{1}{\alpha^2}-1\right) \\ &= \frac{4}{\alpha^2} - 3 + 2\sqrt{\frac{1}{\alpha^2}-1}.\end{aligned}$$

(Although this bound is not tight, we note that

$$\|\Phi x\|_2^2 + \|(I-\Phi)x\|_2^2 + 4\|\Phi x\|_2 \|(I-\Phi)x\|_2 \sqrt{\frac{1}{\alpha^2}-1} + 4\|\Phi x\|_2^2 \left(\frac{1}{\alpha^2}-1\right) = \frac{4}{\alpha^2} - 3$$

is achievable by taking $\|\Phi x\|_2 = 1$ above. This is the interesting case where x falls entirely in the projection subspace.)

Returning to the original optimization problem (for which we must now take a square root), this implies that

$$\frac{\|x - w_b\|_2}{\|x - w_a\|_2} \leq \sqrt{\frac{4}{\alpha^2} - 3 + 2\sqrt{\frac{1}{\alpha^2}-1}}$$

for any observation x that could be mistakenly paired with w_b instead of w_a (under a projection that compacts the distance $\|w_a - w_b\|_2$ by α). Considering all pairs of candidate points in the problem at hand, this bound is maximized by taking $\alpha = \kappa$. □

Bibliography

[1] C. E. Shannon, "A mathematical theory of communication," *Bell Sys. Tech. J.*, vol. 27, pp. 379–423, 623–656, 1948.

[2] K. Brandenburg, "MP3 and AAC explained," in *Proc. AES 17th Int. Conf. High Quality Audio Coding (Florence)*, Sept. 1999.

[3] M. van Heel, B. Gowen, R. Matadeen, E. V. Orlova, R. Finn, T. Pape, D. Cohen, H. Stark, R. Schmidt, M. Schatz, and A. Patwardhan, "Single-particle electron cryo-microscopy: Towards atomic resolution," *Q. Rev. Biophysics*, vol. 33, no. 4, pp. 307–369, 2000.

[4] B. R. Rosen, R. L. Buckner, and A. M. Dale, "Event-related functional MRI: Past, present, and future," *Proc. Nat. Acad. Sci.*, vol. 95, no. 3, pp. 773–780, 1998.

[5] S. Mallat, *A wavelet tour of signal processing*, Academic Press, San Diego, CA, USA, 1999.

[6] R. A. DeVore, "Nonlinear approximation," *Acta Numerica*, vol. 7, pp. 51–150, 1998.

[7] A. Cohen, I. Daubechies, O. G. Guleryuz, and M. T. Orchard, "On the importance of combining wavelet-based nonlinear approximation with coding strategies," *IEEE Trans. Inform. Theory*, vol. 48, no. 7, pp. 1895–1921, July 2002.

[8] R. A. DeVore, B. Jawerth, and B. J. Lucier, "Image compression through wavelet transform coding," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 719–746, Mar. 1992.

[9] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.

[10] S. LoPresto, K. Ramchandran, and M. T. Orchard, "Image coding based on mixture modeling of wavelet coefficients and a fast estimation-quantization framework," in *Proc. Data Compression Conf.*, Snowbird, Utah, March 1997, pp. 221–230.

[11] Z. Xiong, K. Ramchandran, and M. T. Orchard, "Space-frequency quantization for wavelet image coding," *IEEE Trans. Image Processing*, vol. 6, no. 5, pp. 677–693, 1997.

[12] D. L. Donoho, "Denoising by soft-thresholding," *IEEE Trans. Inform. Theory*, vol. 41, no. 3, pp. 613–627, May 1995.

[13] E. J. Candès and D. L. Donoho, "Curvelets — A suprisingly effective non-adaptive representation for objects with edges," in *Curve and Surface Fitting*, A. Cohen, C. Rabut, and L. L. Schumaker, Eds. Vanderbilt University Press, 1999.

[14] E. Candès and D. L. Donoho, "New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities," *Comm. on Pure and Applied Math.*, vol. 57, pp. 219–266, 2004.

[15] L. Ying, L. Demanet, and E. J. Candès, "3D discrete curvelet transform," 2005, Preprint.

[16] D. L. Donoho and C. Grimes, "Image manifolds isometric to Euclidean space," *J. Math. Imaging and Computer Vision*, 2003, To appear.

[17] C. Grimes, *New methods in nonlinear dimensionality reduction*, Ph.D. thesis, Department of Statistics, Stanford University, 2003.

[18] M. B. Wakin, D. L. Donoho, H. Choi, and R. G. Baraniuk, "High-resolution navigation on non-differentiable image manifolds," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Mar. 2005.

[19] M. B. Wakin, D. L. Donoho, H. Choi, and R. G. Baraniuk, "The multiscale structure of non-differentiable image manifolds," in *Proc. Wavelets XI at SPIE Optics and Photonics*, San Diego, August 2005, SPIE.

[20] E. Candès and T. Tao, "Near optimal signal recovery from random projections and universal encoding strategies," *IEEE Trans. Inform. Theory*, 2006, To appear.

[21] D. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, Apr. 2006.

[22] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, Feb. 2006.

[23] E. Candès and T. Tao, "The Dantzig selector: Statistical estimation when p is much larger than n," *Annals of Statistics*, 2006, To appear.

[24] E. Candès and T. Tao, "Decoding by linear programming," *IEEE Trans. Inform. Theory*, vol. 51, no. 12, Dec. 2005.

[25] E. Candès and J. Romberg, "Practical signal recovery from random projections," 2005, Preprint.

[26] E. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, 2006, To appear.

[27] E. Candès and J. Romberg, "Quantitative robust uncertainty principles and optimally sparse decompositions," *Found. of Comp. Math.*, 2006, To appear.

[28] E. Candès and T. Tao, "Error correction via linear programming," *Found. of Comp. Math.*, 2005, Preprint.

[29] D. Donoho and Y. Tsaig, "Extensions of compressed sensing," 2004, Preprint.

[30] J. Haupt and R. Nowak, "Signal reconstruction from noisy random projections," *IEEE Trans. Inform. Theory*, 2006, To appear.

[31] M. B. Wakin, M. F. Duarte, S. Sarvotham, D. Baron, and R. G. Baraniuk, "Recovery of jointly sparse signals from few random projections," in *Proc. Neural Inform. Processing Systems – NIPS*, 2005.

[32] D. Baron, M. F. Duarte, S. Sarvotham, M. B. Wakin, and R. G. Baraniuk, "An information-theoretic approach to distributed compressed sensing," in *Proc. 43rd Allerton Conf. Comm., Control, Comput.*, September 2005.

[33] D. Takhar, V. Bansal, M. Wakin, M. Duarte, D. Baron, K. F. Kelly, and R. G. Baraniuk, "A compressed sensing camera: New theory and an implementation using digital micromirrors," in *Proc. Computational Imaging IV at SPIE Electronic Imaging*, San Jose, January 2006, SPIE.

[34] A. Pinkus, "n-widths and optimal recovery," in *Proc. Symposia Applied Mathematics*, C. de Boor, Ed. 1986, vol. 36, pp. 51–66, American Mathematics Society.

[35] D. Donoho, "For most large underdetermined systems of linear equations, the minimal L1-norm solution is also the sparsest solution," *Communications on Pure and Applied Mathematics*, vol. 59, no. 6, June 2006.

[36] D. Donoho, "Neighborly polytopes and sparse solutions of underdetermined linear equations," 2005, Preprint.

[37] M. Rudelson and R. Vershynin, "Geometric approach to error correcting codes and reconstruction of signals," 2005, Preprint.

[38] D. Donoho, "High-dimensional centrally symmetric polytopes with neighborliness proportional to dimension," Jan. 2005, Preprint.

[39] D. Donoho and J. Tanner, "Neighborliness of randomly-projected simplices in high dimensions," 2005, Preprint.

[40] D. L. Donoho and J. Tanner, "Counting faces of randomly-projected polytopes when then projection radically lowers dimension," Tech. Rep. 2006-11, Stanford University Department of Statistics, 2006.

[41] D. S. Broomhead and M. Kirby, "A new approach for dimensionality reduction: Theory and algorithms," *SIAM J. of Applied Mathematics*, vol. 60, no. 6, 2000.

[42] D. S. Broomhead and M. J. Kirby, "The Whitney Reduction Network: A method for computing autoassociative graphs," *Neural Computation*, vol. 13, pp. 2595–2616, 2001.

[43] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognitive Neuroscience*, vol. 3, no. 1, 1991.

[44] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, December 2000.

[45] D. L. Donoho and C. E. Grimes, "Hessian Eigenmaps: Locally linear embedding techniques for high-dimensional data," *Proc. Natl. Acad. Sci. USA*, vol. 100, no. 10, pp. 5591–5596, May 2003.

[46] K. Q. Weinberger and L. K. Saul, "Unsupervised learning of image manifolds by semidefinite programming," *Int. J. Computer Vision – Special Issue: Computer Vision and Pattern Recognition-CVPR 2004*, vol. 70, no. 1, pp. 77–90, 2006.

[47] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, December 2000.

[48] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimension reduction via tangent space alignment," *SIAM J. Scientific Comput.*, vol. 26, no. 1, 2004.

[49] M. Brand, "Charting a manifold," in *Proc. Neural Inform. Processing Systems – NIPS*, 2002.

[50] J. A. Costa and A. O. Hero, "Geodesic entropic graphs for dimension and entropy estimation in manifold learning," *IEEE Trans. Signal Processing*, vol. 52, no. 8, August 2004.

[51] P. Niyogi, S. Smale, and S. Weinberger, "Finding the homology of submanifolds with confidence from random samples," 2004, Preprint.

[52] G. Carlsson, A. Zomorodian, A. Collins, and L. Guibas, "Persistence barcodes for shapes," *Int. J. of Shape Modeling*, To appear.

[53] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, June 2003.

[54] R. R. Coifman and M. Maggioni, "Diffusion wavelets," *Appl. Comput. Harmon. Anal.*, 2005, To appear.

[55] H. Lu, *Geometric Theory of Images*, Ph.D. thesis, University of California, San Diego, 1998.

[56] D. Mumford and L. Younes (organizers), "University of Minnesota IMA Workshop on Shape Spaces," April 2006.

[57] A. N. Kolmogorov and V. M. Tihomirov, "ϵ-entropy and ϵ-capacity of sets in functional spaces," *Amer. Math. Soc. Transl. (Ser. 2)*, vol. 17, pp. 277–364, 1961.

[58] G. F. Clements, "Entropies of several sets of real valued functions," *Pacific J. Math.*, vol. 13, pp. 1085–1095, 1963.

[59] B. O'Neill, *Elementary Differential Geometry*, Harcourt Academic Press, 2nd edition, 1997.

[60] F. Morgan, *Riemannian Geometry: A Beginner's Guide*, A K Peters, 2nd edition, 1998.

[61] M. W. Hirsch, *Differential Topology*, vol. 33 of *Graduate Texts in Mathematics*, Springer, 1976.

[62] W. M. Boothby, *An Introduction to Differentiable Manifolds and Riemannian Geometry*, Academic Press, revised 2nd edition, 2003.

[63] I. Ur Rahman, I. Drori, V. C. Stodden, D. L. Donoho, and P. Schroeder, "Multiscale representations for manifold-valued data," 2004, Preprint.

[64] J. Kovačević and A. Chebira, "Life beyond bases: The advent of frames," 2006, Preprint.

[65] N. Kingsbury, "Image processing with complex wavelets," *Phil. Trans. R. Soc. Lond. A*, vol. 357, Sept. 1999.

[66] N. Kingsbury, "Complex wavelets for shift invariant analysis and filtering of signals," *Appl. Comp. Harm. Anal.*, vol. 10, pp. 234–253, 2001.

[67] I. W. Selesnick, "The design of approximate Hilbert transform pairs of wavelet bases," *IEEE Trans. Signal Processing*, vol. 50, no. 5, May 2002.

[68] F. C. A. Fernandes, R. L. C. van Spaendonck, and C. S. Burrus, "A new framework for complex wavelet transforms," *IEEE Trans. Signal Processing*, July 2003.

[69] R. van Spaendonck, T. Blu, R. Baraniuk, and M. Vetterli, "Orthogonal Hilbert transform filter banks and wavelets," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, 2003.

[70] M. T. Orchard and H. Ates, "Equiripple design of real and complex filter banks," Tech. Rep., Rice University, 2003.

[71] F. C. A. Fernandes, M. B. Wakin, and R. G. Baraniuk, "Non-Redundant, Linear-Phase, Semi-Orthogonal, Directional Complex Wavelets," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Montreal, Quebec, Canada, May 2004.

[72] M. N. Do and M. Vetterli, "Contourlets: A directional multiresolution image representation," in *Proc. IEEE Int. Conf. Image Proc. (ICIP)*, Rochester, New York, Oct. 2002.

[73] M. N. Do and M. Vetterli, "The contourlet transform: An efficient directional multiresolution image representation," *IEEE Trans. Image Processing*, 2005, To appear.

[74] N. Mehrseresht and D. Taubman, "An efficient content-adaptive motion compensated 3D-DWT with enhanced spatial and temporal scalability," 2004, Preprint.

[75] I. W. Selesnick and K. L. Li, "Video denoising using 2d and 3d dual-tree complex wavelet transforms," in *Proc. SPIE Wavelet Applications Signal Image Processing X*.

[76] R. G. Baraniuk and D. L. Jones, "Shear madness: New orthogonal bases and frames using chirp functions," *IEEE Trans. Signal Proc.*, vol. 41, no. 12, pp. 3543–3549, 1993.

[77] D. L. Donoho, "Unconditional bases are optimal bases for data compression and for statistical estimation," *Appl. Comput. Harmon. Anal.*, vol. 1, no. 1, pp. 100–115, Dec. 1993.

[78] R. A. DeVore, "Lecture notes on Compressed Sensing," *Rice University ELEC 631 Course Notes*, Spring 2006.

[79] A. Cohen, W. Dahmen, I. Daubechies, and R. DeVore, "Tree approximation and optimal encoding," *Appl. Comput. Harmon. Anal.*, vol. 11, pp. 192–226, 2001.

[80] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. on Sci. Comp.*, vol. 20, no. 1, pp. 33–61, 1998.

[81] D. M. Bates and D. G. Watts, *Nonlinear Regression Analysis and Its Applications*, John Wiley and Sons, New York, 1988.

[82] B. Olshausen and D. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?," *Vision Res.*, vol. 37, pp. 311–3325, 1997.

[83] D. Marr, *Vision*, W. H. Freeman and Company, San Francisco, 1982.

[84] E. Le Pennec and S. Mallat, "Sparse geometric image representations with bandelets," *IEEE Trans. Image Processing*, vol. 14, no. 4, pp. 423–438, April 2005.

[85] F. Arandiga, A. Cohen, M. Doblas, R. Donat, and B. Matei, "Sparse representations of images by edge adapted nonlinear multiscale transforms," in *Proc. IEEE Int. Conf. Image Proc. (ICIP)*, Barcelona, Spain, Sept. 2003.

[86] *Let it Wave*, www.letitwave.fr.

[87] W. B Johnson and J. Lindenstrauss, "Extensions of lipschitz mappings into a Hilbert space," in *Proc. Conf. Modern Analysis and Probability*, 1984, pp. 189–206.

[88] D. Achlioptas, "Database-friendly random projections," in *Proc. Symp. Principles of Database Systems*, 2001.

[89] S. Dasgupta and A. Gupta, "An elementary proof of the Johnson-Lindenstrauss lemma," Tech. Rep. TR-99-006, Berkeley, CA, 1999.

[90] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimenstionality," in *Proc. Symp. Theory of Computing*, 1998, pp. 604–613.

[91] J. Tropp and A. C. Gilbert, "Signal recovery from partial information via orthogonal matching pursuit," Apr. 2005, Preprint.

[92] R. Venkataramani and Y. Bresler, "Further results on spectrum blind sampling of 2D signals," in *Proc. IEEE Int. Conf. Image Proc. (ICIP)*, Chicago, Oct. 1998, vol. 2.

[93] M. F. Duarte, M. B. Wakin, and R. G. Baraniuk, "Fast reconstruction of piecewise smooth signals from random projections," in *Proc. SPARS05*, Rennes, France, Nov. 2005.

[94] C. La and M. N. Do, "Signal reconstruction using sparse tree representation," in *Proc. Wavelets XI at SPIE Optics and Photonics*, San Diego, August 2005, SPIE.

[95] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, "The Johnson-Lindenstrauss lemma meets Compressed Sensing," 2006, Preprint.

[96] M. Lustig, D. L. Donoho, and J. M. Pauly, "Rapid MR imaging with Compressed Sensing and randomly under-sampled 3DFT trajectories," in *Proc. 14th Ann. Mtg. ISMRM*, May 2006.

[97] M. F. Duarte, M. A. Davenport, M. B. Wakin, and R. G. Baraniuk, "Sparse signal detection from incoherent projections," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, May 2006.

[98] J. A. Tropp, M. B. Wakin, M. F. Duarte, D. Baron, and R. G. Baraniuk, "Random filters for compressive sampling and reconstruction," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, May 2006.

[99] B. Kashin, "The widths of certain finite dimensional sets and classes of smooth functions," *Izvestia*, , no. 41, pp. 334–351, 1977.

[100] A. Garnaev and E. D. Gluskin, "The widths of Euclidean balls," *Doklady An. SSSR.*, vol. 277, pp. 1048–1052, 1984.

[101] V. Chandrasekaran, M. B. Wakin, D. Baron, and R. Baraniuk, "Representation and compression of multi-dimensional piecewise functions using surflets," submitted to *IEEE Trans. Inf. Theory*, 2006.

[102] M. B. Wakin, J. K. Romberg, H. Choi, and R. G. Baraniuk, "Wavelet-domain approximation and compression of piecewise smooth images," *IEEE Trans. Image Processing*, vol. 15, no. 5, pp. 1071–1087, May 2006.

[103] D. L. Donoho, "Wedgelets: Nearly-minimax estimation of edges," *Annals of Stat.*, vol. 27, pp. 859–897, 1999.

[104] D. L. Donoho, M. Vetterli, R. A. DeVore, and I. Daubechies, "Data compression and harmonic analysis," *IEEE Trans. Information Theory*, vol. 44, no. 6, pp. 2435–2476, 1998.

[105] M. N. Do, P. L. Dragotti, R. Shukla, and M. Vetterli, "On the compression of two-dimensional piecewise smooth functions," in *Proc. IEEE Int. Conf. Image Proc. (ICIP)*, Thessaloniki, Greece, Oct. 2001.

[106] V. Chandrasekaran, M. Wakin, D. Baron, and R. Baraniuk, "Compressing Piecewise Smooth Multidimensional Functions Using Surflets: Rate-Distortion Analysis," Tech. Rep., Rice University ECE Department, Houston, TX, March 2004.

[107] J. Romberg, M. Wakin, and R. Baraniuk, "Multiscale geometric image processing," in *Proc. SPIE Visual Comm. and Image Proc.*, Lugano, Switzerland, July 2003.

[108] J. K. Romberg, M. B. Wakin, and R. G. Baraniuk, "Multiscale wedgelet image analysis: Fast decompositions and modeling," in *Proc. IEEE Int. Conf. Image Proc. (ICIP)*, Rochester, New York, 2002.

[109] M. Holschneider, *Wavelets: An Analysis Tool*, Clarendon Press, Oxford, 1995.

[110] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. 7th Int'l Joint Conf. on Artificial Intelligence*, Vancouver, 1981, pp. 674–679.

[111] L. Quam, "Hierarchical warp stereo," in *Proc. DARPA Image Understanding Workshop*, September 1984, pp. 149–155.

[112] W. Enkelmann, "Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences," *Comp. Vision, Graphics, and Image Processing*, vol. 43, pp. 150–177, 1988.

[113] M. Irani and S. Peleg, "Improving resolution by image registration," *CVGIP: Graphical Models and Image Processing*, vol. 53, no. 3, pp. 231–239, May 1991.

[114] E. P. Simoncelli, "Coarse-to-fine estimation of visual motion," in *Proc. Workshop Image Multidimensional Signal Proc.*, Cannes, France, September 1993, pp. 128–129.

[115] E. P. Simoncelli, "Bayesian multi-scale differential optical flow," in *Handbook of Computer Vision and Applications*, B. Jähne, H. Haussecker, and P. Geissler, Eds., vol. 2, chapter 14, pp. 397–422. Academic Press, San Diego, April 1999.

[116] P. N. Belhumeur and G. D. Hager, "Tracking in 3D: Image variability decomposition for recovering object pose and illumination," *Pattern Analysis and Applications*, vol. 2, pp. 82–91, 1999.

[117] C. Davis and W.M. Kahan, "The rotation of eigenvectors by a perturbation, III," *SIAM J. Numer. Anal.*, vol. 7, no. 1, pp. 1–46, 1970.

[118] Gilbert W. Stewart and Ji guang Sun, *Matrix Perturbation Theory*, Academic Press, Boston, 1990.

[119] W. T. Freeman, "Exploiting the generic viewpoint assumption," *Int. J. Computer Vision*, vol. 20, no. 3, 1996.

[120] Y. Keller and A. Averbach, "Fast motion estimation using bidirectional gradient methods," *IEEE Trans. Image Processing*, vol. 13, no. 8, pp. 1042–1054, August 2004.

[121] D. Baron, M. B. Wakin, M. F. Duarte, S. Sarvotham, and R. G. Baraniuk, "Distributed compressed sensing," 2005, Preprint.

[122] D. Estrin, D. Culler, K. Pister, and G. Sukhatme, "Connecting the physical world with pervasive networks," *IEEE Pervasive Computing*, vol. 1, no. 1, pp. 59–69, 2002.

[123] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Comm. ACM*, vol. 43, no. 5, pp. 51–58, 2000.

[124] H. Luo and G. Pottie, "Routing explicit side information for data compression in wireless sensor networks," in *Proc. Int. Conf. on Distirbuted Computing in Sensor Systems (DCOSS)*, Marina Del Rey, CA, June 2005.

[125] M. Gastpar, P. L. Dragotti, and M. Vetterli, "The distributed Karhunen-Loeve transform," *IEEE Trans. Inform. Theory*, Nov. 2004, Submitted.

[126] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, vol. 19, pp. 471–480, July 1973.

[127] S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," *IEEE Trans. Inform. Theory*, vol. 49, pp. 626–643, Mar. 2003.

[128] Z. Xiong, A. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *IEEE Signal Processing Mag.*, vol. 21, pp. 80–94, Sept. 2004.

[129] J. Tropp, A. C. Gilbert, and M. J. Strauss, "Simulataneous sparse approximation via greedy pursuit," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Mar. 2005.

[130] V. N. Temlyakov, "A remark on simultaneous sparse approximation," *East J. Approx.*, vol. 100, pp. 17–25, 2004.

[131] S. F. Cotter, B. D. Rao, K. Engan, and K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Trans. Signal Processing*, vol. 51, pp. 2477–2488, July 2005.

[132] R. Puri and K. Ramchandran, "PRISM: A new robust video coding architecture based on distributed compression principles," in *Proc. 40th Allerton Conf. Communication, Control, and Computing*, Monticello, IL, Oct. 2002.

[133] R. Wagner, R. G. Baraniuk, and R. D. Nowak, "Distributed image compression for sensor networks using correspondence analysis and super-resolution," in *Proc. Data Compression Conf.*, Mar. 2000.

[134] S. Sarvotham, M. B. Wakin, D. Baron, M. F. Duarte, and R. G. Baraniuk, "Analysis of the DCS one-stage greedy algoritm for common sparse supports," Tech. Rep., Rice University ECE Department, Oct. 2005, available at http://cmc.rice.edu/docs/docinfo.aspx?doc=Sar2005Nov9Analysisof.

[135] J. Tropp, "Algorithms for simultaneous sparse approximation. Part II: Convex relaxation," *EURASIP J. App. Signal Processing*, 2005, To appear.

[136] M. B. Wakin and R. G. Baraniuk, "Random projections of signal manifolds," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, May 2006.

[137] A. Cohen, W. Dahmen, and R. DeVore, "Compressed sensing and best k-term approximation," 2006, Preprint.

[138] R. Coifman, F. Geshwind, and Y. Meyer, "Noiselets," *Appl. Comput. Harmon. Anal.*, vol. 10, pp. 27–44, 2001.

[139] D. L. Donoho and X. Huo, "Beamlet pyramids: A new form of multiresolution analysis, suited for extracting lines, curves, and objects from very noisy image data," in *Proc. SPIE*, July 2000, vol. 4119.

[140] D. L. Donoho and X. Huo, "Beamlets and multiscale image analysis," *Multiscale and Multiresolution Methods,* Ed. T.J. Barth, T. Chan, and R. Haimes, *Springer Lec. Notes Comp. Sci. and Eng., 20*, pp. 149–196, 2002.

[141] W. Pennebaker and J. Mitchell, "JPEG: Still image data compression standard," *Van Nostrand Reinhold*, 1993.

[142] M. F. Duarte, M. B. Wakin, D. Baron, and R. G. Baraniuk, "Universal distributed sensing via random projections," in *Proc. Int. Workshop Inf. Processing in Sensor Networks (IPSN '06)*, 2006.

[143] G. G. Lorentz, M. von Golitschek, and Yu. Makovoz, *Constructive approximation: Advanced problems*, vol. 304, Springer Grundlehren, Berlin, 1996.