ABSTRACT

Title of dissertation: MULTISCALE AND DIRECTIONAL REPRESENTATIONS OF HIGH-DIMENSIONAL INFORMATION CONTENT IN REMOTELY SENSED DATA

Daniel Weinberg, Doctor of Philosophy, 2015

Dissertation directed by: Professor Wojciech Czaja
Department of Mathematics

This thesis explores the theory and applications of directional representations in the field of anisotropic harmonic analysis. Although wavelets are optimal for decomposing functions in one dimension, they are unable to achieve the same success in two or more dimensions due to the presence of curves and surfaces of discontinuity. In order to optimally capture the behavior of a function at high-dimensional discontinuities, we must be able to incorporate directional information into our analyzing functions, in addition to location and scale. Examples of such representations are contourlets, curvelets, ridgelets, bandelets, wedgelets, and shearlets. Using directional representations, in particular shearlets, we tackle several challenging problems in the processing of remotely sensed data. First, we detect roads and ditches in LIDAR data of rural scenes. Second, we develop an algorithm for superresolution of optical and hyperspectral data. We conclude by presenting a stochastic particle model in which the probability of movement in a particular direction is neighbor-weighted.

MULTISCALE AND DIRECTIONAL REPRESENTATIONS OF HIGH-DIMENSIONAL INFORMATION CONTENT IN REMOTELY SENSED DATA

by

Daniel Weinberg

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2015

Advisory Committee:
Professor Wojciech Czaja, Chair
Professor John J. Benedetto
Professor Kasso Okoudjou
Professor Radu Balan
Professor Alexander Barg

Dedication

I dedicate this thesis to my parents for their almost infinite patience and emotion support throughout the roller coaster ride that was graduate school. I know you'll be proud of me whatever I do.

Acknowledgments

As this long and arduous journey through graduate school comes to end, I reflect on all of the people without whom this thesis would never have been possible. First and foremost, I would like to thank my advisor Wojciech Czaja for taking me as a student late into my studies. Even if he were busy, he would still always take the time to meet with me to discuss my projects. He would provide many valuable ideas and also be willing to discuss at length any of my own. I also appreciate his support, both emotional and financial, his candor, and sense of humor.

Thanks to John Benedetto for being so welcoming when I was searching for a new home. The Norbert Wiener Center became this home, and my graduate school experience was improved substantially by being a part this community. Thanks to the NWC computer lab crew: Tim Doster, Alex Cloninger, Travis Andrews, Matt Begué, Matt Guay, Chae Clark, James Murphy, Paul Koprowski, and Ben Manning. I enjoyed our discussions, whether they were about math or otherwise.

I also thank my collaborators, who contributed to the ideas and writing of my thesis, including Jeff Sieracki (Chapter 4), James and Ed Bosch (Chapter 5), and Doron Levy (Chapter 6). I also thank Ben and Ariel Hafftka for collaborations that do not appear in this thesis [1, 2]. Special thanks to Nate Strawn for fighting to keep me in the program, as well as recent discussions that improved the results of Chapter 4.

Thank you to the first, and last, alumni of the UMD Math Bootcamp, in addition to the other students in my year. We may not have been the most successful class in terms of finishing the program in a timely manner, but we may have been the closest. Thanks for making the first few years of school fun, that is until we realized that we needed to actually start working hard if we ever wanted to finish the program.

Finally, thanks to my thesis committee members: Wojciech, John, Dr. Radu Balan, Dr. Kasso Okoudjou, and Dr. Alexander Barg.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

In applied harmonic analysis, an important problem is, given a function f that belongs to a certain class of functions in $L^2(\mathbb{R}^d)$, to find analyzing functions $(\psi_i)_{i \in I}$, I countable, such that $f = \sum_{i \in I} c_i(f)\psi_i$. By manipulating the coefficients $(c_i(f))_{i \in I} \subset l^2(I)$ and performing synthesis to obtain a function from the coefficients, we hope to extract important information from the function/signal. Two-dimensional applications include image denoising, enhancement, segmentation, edge detection, and compression. The simplest type of useful manipulation we can perform is thresholding, that is, setting a certain number of coefficients less than a given magnitude to 0.

For many applications, it is helpful if we have a sparse representation in which few coefficients are needed to reconstruct a close approximation to the original function. Let f_N denote the *best N-term approximation* of f in the sense that it minimizes $||f - f_N||_2$, over all choices of N-term reconstructions. A natural question then is, given a class of functions that model images, to find analyzing functions which provide sparse representations that are optimal, or at least very close to optimal. Wavelets, which decompose images by location and scale, do not have this property for images containing lines of discontinuity. Hence, we are led to seek out other representations that additionally take into account directional information.

In this thesis, we explore how the addition of directional information can be leveraged for image processing and mathematical modeling. In Chapter 2, we review some preliminaries from Fourier analysis and finite frame theory that will be used in the following chapters. Next, in Chapter 3, we introduce the continuous shearlet transform as a special case of the continuous wavelet transform. After covering discrete shearlets, we define the cartoon-like images and discuss what it means for a representation to be essentially optimally sparse. We also present the current numerical implementations of the shearlet transform and explore whether the theoretical decay rate occurs in practice.

In Chapter 4, we utilize shearlets to analyze LIDAR data of several rural scenes. Our aim is to detect directional structures such as roads and ditches. The problem has a number of challenging aspects to it, most noticeably the issue of filtering out the trees. In our first approach to the problem, we perform a simple thresholding of the shearlet coefficients. This approach suffers from several shortcomings, such as detecting the edges of trees and being unable to quantify the edge's strength. We are then led to a second approach in which we incorporate information about the local variance, along with the number of small shearlet coefficients. This approach leads to improved results.

In Chapter 5, we consider the problem of single-image superresolution, i.e., increasing an image's resolution without introducing undesired artifacts. Our focus is mainly on identifying the location and orientation of strong edges, so that we can smooth out the jagged edges that result from aliasing. To accomplish this, we utilize the maximum shearlet coefficient, which we prove is effective in an ideal case. We perform experiments on various images, including a color orthophoto, simple shapes, and a band from hyperspectral imagery. We demonstrate the value of our approach both quantitatively and qualitatively by comparing it to existing techniques. This work is published in [3] and [4].

Finally, in Chapter 6, we develop a $2D$ probabilistic particle model for the movement of bacteria in the presence of light, a process called phototaxis. In the model, particles are restricted to a periodic grid and allowed to move at discrete time intervals in any of four directions: up, down, left, or right. The probability that a particle will move in a particular direction is determined by a parameter a and is weighted by the proportion of its neighbors in that direction. A particle may also choose not to move in that time interval with probability b. These rules cause the bacteria to form aggregates, a phenomenon observed in reality. By running the model many times with different parameters, grid sizes, and starting positions, we see that in almost all cases, only horizontal or vertical aggregates (but not both) persist as $t \to \infty$. Next, we derive a system of ordinary differential equations approximating the expected number of particles at each position at time t. Similar to the discrete model, we observe horizontal and vertical aggregates, though now both can occur simultaneously. For some values of the parameters, we observe that the expectations are essentially random. We empirically derive a linear constraint on the parameters, which we then show accurately predicts the transition from aggregates to randomness. This is joint work with Doron Levy and is based on work published in [5].

Chapter 2: Some Preliminaries

2.1 Basic Fourier Analysis

Since there is no universal agreement as to the precise definition of the Fourier transform, we set the one we will be using here:

Definition 2.1.1. Let $f \in L^1(\mathbb{R}^d)$. The *Fourier transform* of f is

$$\hat{f}(\xi) = \int_{\mathbb{R}^d} f(x) e^{-2\pi i \langle x, \xi \rangle} dx.$$

The transform can be extended to a unitary operator $\mathcal{F} : L^2(\mathbb{R}^d) \to L^2(\mathbb{R}^d)$ by the density of $L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$ in $L^2(\mathbb{R}^d)$.

The inverse of the Fourier transform $\mathcal{F}^{-1}$ can be given explicitly for functions in $L^1(\mathbb{R}^d)$:

Definition 2.1.2. Let $f \in L^1(\mathbb{R}^d)$. Then the *inverse Fourier transform* is given by

$$\check{f}(x) = \int_{\mathbb{R}^d} f(\xi) e^{2\pi i \langle x, \xi \rangle} d\xi.$$

The fact that the Fourier transform is unitary follows from the *Parseval-Plancherel Theorem*:

Theorem 2.1.3. *Let $f, g \in L^2(\mathbb{R}^d)$. Then*

$$\langle f, g \rangle = \langle \hat{f}, \hat{g} \rangle,$$

which implies

$$||f||_2 = ||\hat{f}||_2.$$

Remark 2.1.4. Let G be any locally-compact Abelian group (LCAG). We can define a unitary operator $\mathcal{F} : L^2(G) \to L^2(\hat{G})$, where $\hat{G}$ is the Pontryagin dual of G, which generalizes the Fourier transform as defined above [6]. (Note that $\hat{\mathbb{R}} = \mathbb{R}$).

We will be interested in $M \times N$ images, which can be considered as functions on $G = \mathbb{Z}_M \times \mathbb{Z}_N$. Let $f : G \to \mathbb{R}$ represent an image. We can define the discrete Fourier transform (DFT) of this function on $\hat{G} = G$ by

$$\hat{f}(\omega_1, \omega_2) := \sum_{(m,n)\in G} f(m,n) e^{-2\pi i \langle(\omega_1,\omega_2),(m/M,n/N)\rangle}$$

and the inverse DFT by

$$\check{f}(m,n) := \frac{1}{MN} \sum_{(\omega_1,\omega_2)\in G} f(\omega_1,\omega_2) e^{2\pi i \langle(\omega_1,\omega_2),(m/M,n/N)\rangle}.$$

2.2 Finite Frame Theory

Frames, first introduced by Duffin and Schaeffer in [7], are essentially a generalization of bases, where redundancy is allowed. We review some basic definitions and theorems from finite frame theory [8]. Similar definitions hold for infinite-dimensional spaces.

Definition 2.2.1. Let $\mathscr{H}^N$ be an N-dimensional Hilbert space. A family of vectors $(\phi_i)_{i=1}^M$ in $\mathscr{H}^N$ is called a *frame* for $\mathscr{H}^N$ if there exists constants $0 < A \leqslant B < \infty$ such that

$$A||x||^2 \leqslant \sum_{i=1}^{M} |\langle x, \phi_i\rangle|^2 \leqslant B||x||^2 \quad \text{for all } x \in \mathscr{H}^N.$$

The values $(\langle x, \phi_i \rangle)_{i=1}^M$ are the *frame coefficients*. The constants A and B are called the *lower* and *upper frame bounds*, respectively. The largest lower frame bound and smallest upper frame bound, A_{op} and B_{op}, are called the *optimal frame bounds.* If $A_{op} = B_{op}$, then the frame is called *(A-)tight.* A 1-tight frame is called *Parseval*, since in this case Parseval's Identity is satisfied, i.e,

$$\sum_{i=1}^{M} |\langle x, \phi_i \rangle|^2 = ||x||^2.$$

Finite frames are easily characterized since they are simply spanning sets:

Proposition 2.2.2 (Lemma 1.2ii, [8])**.** *Let $(\phi_i)_{i=1}^M \subset \mathscr{H}^N$. $(\phi_i)_{i=1}^M$ is a frame for $\mathscr{H}^N$ if and only if it is a spanning set for $\mathscr{H}^N$.*

Proof. If $(\phi_i)_{i=1}^M$ is not a spanning set, there exists a non-zero x in the orthogonal complement of the span of the $(\phi_i)_{i=1}^M$. Hence, there can be no lower frame bound. Conversely, if $(\phi_i)_{i=1}^M$ is not a frame, then we can find a sequence of normalized vectors $(x_n)_{n=1}^\infty$ such that $\sum_{i=1}^M |\langle x_n, \phi_i \rangle|^2 < 1/n$ for all $n \in \mathbb{N}$. Because all vectors lie on S^{N-1}, by passing to a subsequence, we can assume that the sequence converges to some $x \in S^{N-1}$. Therefore, the orthogonal complement of the span of the (ϕ_i) is nontrivial, implying that $(\phi_i)_{i=1}^M$ is not a spanning set. □

Frames have three important associated operators:

Definition 2.2.3. Let $(\phi_i)_{i=1}^M \subset \mathscr{H}^N$. Let $l_2^M := l_2(\{1, \ldots, M\})$.

1. The *analysis operator* $T : \mathscr{H}^N \to l_2^M$ is defined by

$$Tx := \left(\langle x, \phi_i \rangle\right)_{i=1}^M, \quad x \in \mathscr{H}^N.$$

2. The *synthesis operator* is the adjoint operator $T^* : l_2^M \to \mathscr{H}^N$ given by

$$T^*(a_i)_{i=1}^M = \sum_{i=1}^M a_i \phi_i$$

3. The *frame operator* $S : \mathscr{H}^N \to \mathscr{H}^N$ is defined by

$$Sx := T^*Tx = \sum_{i=1}^M \langle x, \phi_i \rangle \phi_i, \quad x \in \mathscr{H}^N.$$

Much can be said about these operators, but we restrict ourselves to just a few remarks. First, if $(\phi_i)_{i=1}^M$ is a frame for $\mathscr{H}^N$, then a matrix representation for its synthesis operator T^* is given by the matrix whose columns are the frame elements:

$$\begin{bmatrix} | & | & \cdots & | \\ \phi_1 & \phi_2 & \cdots & \phi_M \\ | & | & \cdots & | \end{bmatrix}.$$

Since a finite frame is a spanning set, it follows that $(\phi_i)_{i=1}^M$ is a frame if and only if T^* is surjective, or equivalently that T is injective. Second, the frame operator S associated to a frame is a self-adjoint, positive definite operator, and hence is invertible. That it is self-adjoint is clear from its definition. It is positive definite since

$$\langle Sx, x \rangle = \langle T^*Tx, x \rangle = ||Tx||^2 > 0$$

for all nonzero x by the injectivity of T. Finally, if S is associated with an A-tight frame, for all $x \in \mathscr{H}^N$

$$\langle Sx, x \rangle = ||Tx||^2 = \sum_{i=1}^M |\langle x, \phi_i \rangle|^2 = \langle A \cdot Id\ x, x \rangle$$

implying that $S = A \cdot Id$.

Next, we show that we can reconstruct a vector from its frame coefficients.

Theorem 2.2.4 (Theorem 1.8, [8]). *Let $(\phi_i)_{i=1}^M$ be a frame for $\mathscr{H}^N$ with frame operator S. Then, for all $x \in \mathscr{H}^N$*

$$x = \sum_{i=1}^M \langle x, \phi_i \rangle S^{-1}\phi_i = \sum_{i=1}^M \langle x, S^{-1}\phi_i \rangle \phi_i. \tag{2.1}$$

Proof. The first equality follows immediately from applying S^{-1} to both sides of the definition of S. The second equality is obtained by replacing x with $S^{-1}x$ in the definition and using that S^{-1} is self-adjoint. □

Reconstruction formulas as in (2.1) are not unique in general.

Definition 2.2.5. Let $(\phi_i)_{i=1}^M$ be a frame for $\mathscr{H}^N$. Any sequence of vectors $(\psi_i)_{i=1}^M$ in $\mathscr{H}^N$ such that

$$x = \sum_{i=1}^M \langle x, \phi_i \rangle \psi_i$$

for all $x \in \mathscr{H}^N$ is called a *dual frame* for the (ϕ_i). The dual frame given by $(S^{-1}\phi_i)_{i=1}^M$ is called the *canonical dual frame*.

The next proposition explains the use of the word "dual":

Proposition 2.2.6 (Proposition 1.15, [8]). *Let $(\phi_i)_{i=1}^M$ be a frame for $\mathscr{H}^N$ with dual frame $(\psi_i)_{i=1}^M$. Then $(\psi_i)_{i=1}^M$ is itself a frame with dual frame $(\phi_i)_{i=1}^M$.*

Proof. Let Φ and Ψ be the matrices with columns $(\phi_i)_{i=1}^M$ and $(\psi_i)_{i=1}^M$, respectively. The hypothesis is equivalent to $\Psi\Phi^* = Id$. Taking the adjoint of both sides, we have $\Phi\Psi^* = Id$, from which the conclusion follows. □

Remark 2.2.7. The dual frame for an A-tight frame $(\phi_i)_{i=1}^M$ is given by $(\frac{1}{A}\phi_i)_{i=1}^M$, providing the reconstruction formula

$$x = \frac{1}{A}\sum_{i=1}^M \langle x, \phi_i \rangle \phi_i.$$

In particular, Parseval frames are their own canonical dual, with the simple reconstruction formula

$$x = \sum_{i=1}^{M} \langle x, \phi_i \rangle \phi_i.$$

Chapter 3: Introduction to Shearlets

3.1 The Continuous Wavelet Transform

Before discussing the shearlet transform, we define the continuous wavelet transform, of which the shearlet transform ends up being a special case [9].

Definition 3.1.1. The *continuous affine systems* are

$$\left\{\psi_{M,t} = T_t D_M^{-1}\psi = |\det M|^{1/2}\psi(M(\cdot - t)) : (M,t) \in G \times \mathbb{R}^d\right\}$$

where $\psi \in L^2(\mathbb{R}^d)$, G is a subgroup of $GL_d(\mathbb{R})$, D_M is the dilation operator on $L^2(\mathbb{R}^d)$ defined by

$$D_M\psi(\cdot) = |\det M|^{-1/2}\psi(M^{-1}\cdot),$$

and T_t is the translation operator defined by

$$T_t\psi(\cdot) = \psi(\cdot - t).$$

Definition 3.1.2. The *affine group* $\mathscr{A}_d$ corresponding to an affine system is the set of parameters (M,t) equipped with the group operation

$$(M,t)\cdot(M',t') = (MM', t + Mt').$$

Theorem 3.1.3 (Theorem 1, [9])**.** *Let $d\mu$ be a left-invariant Haar measure on $G \subset GL_d(\mathbb{R})$, and $d\lambda$ be a left Haar measure of $\mathscr{A}_d$. Furthermore, suppose that $\psi \in L^2(\mathbb{R}^d)$*

satisfies the admissibility condition

$$\int_G |\hat{\psi}(M^T\xi)|^2 |\det M| d\mu(M) = 1.$$

Then any function $f \in L^2(\mathbb{R}^d)$ can be recovered via the reproducing formula

$$f = \int_{\mathscr{A}_d} \langle f, \psi_{M,t} \rangle \psi_{M,t} d\lambda(M,t), \tag{3.1}$$

interpreted weakly.

A function ψ satisfying the theorem's conditions is called a *continuous wavelet.* The fact that the conclusion is interpreted weakly means that

$$\langle f, g \rangle = \int_{\mathscr{A}_d} \langle f, \psi_{M,t} \rangle \langle \psi_{M,t}, g \rangle d\lambda(M,t)$$

for all $g \in L^2(\mathbb{R}^d)$, which is equivalent, via the polarization identity to [10]

$$||f||_2^2 = \int_{\mathscr{A}_d} |\langle f, \psi_{M,t} \rangle|^2 d\lambda(M,t).$$

Definition 3.1.4. The *Continuous Wavelet Transform* associated to a continuous affine system is defined by

$$L^2(\mathbb{R}^d) \ni f \mapsto \mathscr{W}_\psi f(M,t) = \langle f, \psi_{M,t} \rangle, \qquad (M,t) \in \mathscr{A}_d.$$

Note that the transform is continuous in its translations, but G is allowed to be discrete. As a special case, letting $G = \{aI_d : a > 0\}$., we have the *(isotropic) Continuous Wavelet Transform.* The admissibility condition for ψ is

$$\int_0^\infty |\hat{\psi}(a\xi)|^2 \frac{da}{a} = 1$$

which is commonly known as the Calderón admissibility condition for wavelets. Letting $d = 1$ and $G = \{2^m : m \in \mathbb{Z}\}$ results in the dyadic 1-dimensional discrete wavelet transform.

3.2 The Continuous Shearlet Transform

We start with a few definitions. Let $\psi \in L^2(\mathbb{R}^2)$. To better analyze anisotropic features, we introduce the *parabolic scaling matrices*

$$A_a = \begin{pmatrix} a & 0 \\ 0 & a^{1/2} \end{pmatrix}, \qquad a > 0$$

and the *shearing matrices*

$$S_s = \begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix}, \qquad s \in \mathbb{R}.$$

Definition 3.2.1. For $\psi \in L^2(\mathbb{R}^2)$, the *continuous shearlet system* $SH(\psi)$ is

$$SH(\psi) = \left\{\psi_{a,s,t} = T_t D_{A_a} D_{S_s} \psi : a > 0, s \in \mathbb{R}, t \in \mathbb{R}^2\right\}.$$

We can give the parameter space a group structure

$$(a, s, t) \cdot (a', s', t') = (aa', s + s'\sqrt{a}, t + S_s A_a t')$$

and we denote this *shearlet group* $\mathbb{S}$.

Definition 3.2.2. Let $\psi \in L^2(\mathbb{R}^2)$. The *Continuous Shearlet Transform* of $f \in L^2(\mathbb{R}^2)$ is

$$f \mapsto \mathcal{SH}_\psi f(a, s, t) = \langle f, T_t D_{A_a} D_{S_s} \psi\rangle, \ (a, s, t) \in \mathbb{S}.$$

For this transform to be useful, we need to be able to invert it, which we can do if ψ satisfies the admissibility property

$$\int_{\mathbb{R}^2} \frac{|\hat{\psi}(\omega_1, \omega_2)|^2}{\omega_1^2} d\omega_2 d\omega_1 < \infty.$$

Such a function ψ is called an *admissible shearlet*. We assume that ψ splits up as

$$\hat{\psi}(\omega) = \hat{\psi}(\omega_1, \omega_2) = \hat{\psi}_1(\omega_1)\hat{\psi}_2\left(\frac{\omega_2}{\omega_1}\right). \tag{3.2}$$

If we further assume that $\hat{\psi}_1$ and $\hat{\psi}_2$ are continuous with $\mathrm{supp}(\hat{\psi}_1) \subset [-c, -b] \cup [b, c]$ and $\mathrm{supp}(\hat{\psi}_2) \subset [-d, d]$ for some $c > b > 0, d > 0$, it follows that ψ is an admissible shearlet. This is clear since the integral would be of a bounded function over a bounded region not containing a neighborhood of 0. Note that

$$\psi_{a,s,t}(x) = a^{-3/4}\psi\left(\begin{pmatrix} 1/a & -s/a \\ 0 & 1/\sqrt{a} \end{pmatrix}(x-t)\right).$$

Define

$$B = \begin{pmatrix} 1/a & -s/a \\ 0 & 1/\sqrt{a} \end{pmatrix}.$$

Then

$$\begin{aligned} \hat{\psi}_{a,s,t}(\omega) &= a^{-3/4}e^{-2\pi i\langle\omega,t\rangle}\mathcal{F}(\psi(Bx)) \\ &= a^{-3/4}e^{-2\pi i\langle\omega,t\rangle}|\det B|^{-1/2}\hat{\psi}((B^{-1})^T\omega) \\ &= a^{-3/4}e^{-2\pi i\langle\omega,t\rangle}\hat{\psi}(a\omega_1, \sqrt{a}(s\omega_1 + \omega_2)) \\ &= a^{-3/4}e^{-2\pi i\langle\omega,t\rangle}\hat{\psi}_1(a\omega_1)\hat{\psi}_2\left(a^{-1/2}\left(\frac{\omega_2}{\omega_1} + s\right)\right). \end{aligned}$$

We can compute the continuous shearlet transform as

$$\begin{aligned}
\mathcal{SH}_\psi(f)(a,s,t) &= \langle f, \psi_{a,s,t}\rangle \\
&= \langle \hat{f}, \hat{\psi}_{a,s,t}\rangle \\
&= a^{3/4}\int_{\mathbb{R}^2} \hat{f}(\omega)\hat{\psi}_1(a\omega_1)\hat{\psi}_2\left(a^{-1/2}\left(\frac{\omega_2}{\omega_1}+s\right)\right)e^{2\pi i\langle\omega,t\rangle}d\omega \\
&= a^{3/4}\mathcal{F}^{-1}\left(\hat{f}(\omega)\hat{\psi}_1(a\omega_1)\hat{\psi}_2\left(a^{-1/2}\left(\frac{\omega_2}{\omega_1}+s\right)\right)\right)(t) \\
&= a^{3/4}\mathcal{F}^{-1}\left(\hat{f}(\omega)\hat{\psi}(a\omega_1,\sqrt{a}(s\omega_1+\omega_2)\right)(t). \qquad (3.3)
\end{aligned}$$

In order to obtain a reproducing formula for shearlets, we impose some assumptions on ψ_1 and ψ_2 [11]. First, we assume that ψ_1 is a real-valued wavelet in the sense that it satisfies the Calderón condition (3.1). Second, we assume that $||\psi_2||_2 = 1$. Additionally, we assume that $\hat{\psi}_1, \hat{\psi}_2 \in C^\infty(\mathbb{R})$ with $\text{supp}(\hat{\psi}_1) \subset [-2,-1/2] \cup [1/2,2]$ and $\text{supp}(\hat{\psi}_2) \subset [-1,1]$. Using our derived expression for the Fourier transform of $\psi_{a,s,t}$, we see that the support of $\hat{\psi}_{a,s,t}$ is contained in

$$\left\{(\omega_1,\omega_2) \in \mathbb{R}^2 : \frac{1}{2a} \leqslant |\omega_1| \leqslant \frac{2}{a}, |s+\frac{\omega_2}{\omega_1}| \leqslant \sqrt{a}\right\}$$

from which it follows that shearlets become elongated as $a \to 0$.

Theorem 3.2.3 (Theorem 1, [11]). *Under the preceding conditions, we have a reproducing formula for the continuous shearlet system $SH(\psi)$.*

Proof. Shearlets are a special case of continuous wavelets where

$$G = \{S_sA_a : (a,s) \in \mathbb{R}^+ \times \mathbb{R}\}.$$

In this case, the admissibility condition (3.1) is

$$\int_{\mathbb{R}}\int_0^\infty |\hat{\psi}(A_a^TS_s^T\omega)|^2a^{-3/2}dads = 1 \qquad \text{for a.e. } \omega \in \mathbb{R}^2.$$

This is true since

$$\begin{aligned}
&\int_{\mathbb{R}}\int_0^\infty |\hat{\psi}(A_a^T S_s^T \omega)|^2 a^{-3/2} da ds \\
&\qquad = \int_{\mathbb{R}}\int_0^\infty |\hat{\psi}_1(a\omega_1)|^2 \left|\hat{\psi}_2\left(a^{-1/2}\left(s+\frac{\omega_2}{\omega_1}\right)\right)\right|^2 a^{-3/2} da ds \\
&\qquad = \int_0^\infty |\hat{\psi}_1(a\omega_1)|^2 \int_{\mathbb{R}} \left|\hat{\psi}_2\left(s + a^{-1/2}\frac{\omega_2}{\omega_1}\right)\right|^2 ds \frac{da}{a} \\
&\qquad = \int_0^\infty |\hat{\psi}_1(a\omega_1)|^2 \frac{da}{a} = 1.
\end{aligned}$$

□

3.3 Discrete Shearlets on $\mathbb{R}^2$

We now turn to defining discrete shearlets. Consider the system

$$\{\psi_{j,k,m} = D_{A_2^j S_k} T_m \psi : j,k \in \mathbb{Z}, m \in \mathbb{Z}^2\} \tag{3.4}$$

$$= \{2^{3j/4}\psi(S_k A_2^j x - m) : j,k \in \mathbb{Z}, m \in \mathbb{Z}^2\}. \tag{3.5}$$

Our goal here is to define $\psi \in L^2(\mathbb{R}^2)$ in such a way that this system forms a Parseval frame. As before, we assume (3.2). We choose $\psi_1 \in L^2(\mathbb{R})$ such that $\hat{\psi}_1 \in C^r(\mathbb{R})$ for some $r \in \mathbb{N}$ and $\text{supp}(\hat{\psi}_1) \subset [-1/2,-1/4] \cup [1/4,1/2]$ satisfying

$$\sum_{j\in\mathbb{Z}} |\hat{\psi}_1(2^j\xi)|^2 = 1 \qquad \text{for } \xi \in \mathbb{R}. \tag{3.6}$$

Note that (3.6) is satisfied by a discrete wavelet. One choice is the Meyer wavelet defined by $\hat{\psi}_M(\xi) = e^{i\pi\xi}b(\xi)$ where

$$b(\xi) = \begin{cases} \sin(\frac{\pi}{2}v(3|\xi|-1)) & : \frac{1}{3} \leqslant |\xi| \leqslant \frac{2}{3} \\ \cos(\frac{\pi}{2}v(\frac{3|\xi|}{2}-1)) & : \frac{2}{3} \leqslant |\xi| \leqslant \frac{4}{3} \\ 0 & : \text{o.w.} \end{cases} \tag{3.7}$$

and v is a function symmetric about $(0.5, 0.5)$, that is, $v(x) + v(1-x) = 1$ for all $x \in \mathbb{R}$. One possibility is

$$v(x) := \begin{cases} 0 & \text{for } x < 0 \\ 35x^4 - 84x^5 + 70x^6 - 20x^7 & \text{for } 0 \leqslant x \leqslant 1 \\ 1 & \text{for } x > 1. \end{cases} \tag{3.8}$$

The choice of v is important since its smoothness is the same as that of $\hat{\psi}_M$, which affects the decay rate of ψ_M. The above v is in C^3, but we can construct a $v \in C^r(\mathbb{R})$ for any $r \in \mathbb{N}$ [12]:

$$v(x) = \begin{cases} 0 & \text{for } x < 0 \\ \left(\int_0^1 t^r(1-t)^r dt\right)^{-1} \int_0^x t^r(1-t)^r dt & \text{for } 0 \leqslant x \leqslant 1 \\ 1 & \text{for } x > 1. \end{cases}$$

For ψ_2, we prescribe that $\hat{\psi}_2 \in C^\infty(\mathbb{R})$, $\text{supp}(\hat{\psi}_2) \subset [-1, 1]$, and that

$$\sum_{k \in \mathbb{Z}} |\hat{\psi}_2(k + \xi)|^2 = 1 \qquad \text{for } \xi \in \mathbb{R}. \tag{3.9}$$

Note that (3.9) implies that the integer translates of ψ_2 are orthonormal. Also, due to the support of $\hat{\psi}_2$, (3.9) is equivalent to

$$\sum_{k=-1,0,1} |\hat{\psi}_2(k + \xi)|^2 = 1 \qquad \text{for } \xi \in [-1, 1]. \tag{3.10}$$

One way to construct such a function is to let ϕ be any $C_c^\infty(\mathbb{R})$ function supported in $[-1, 1]$ and to define

$$\hat{\psi}_2(\xi) = \frac{\phi(\xi)}{\sqrt{\sum_{k \in \mathbb{Z}} |\phi(\xi + k)|^2}}.$$

We have the following result:

Theorem 3.3.1 (Theorem 3, [11]). *Let $\psi \in L^2(\mathbb{R}^2)$ be of the form (3.2) where ψ_1 satisfies (3.6) and ψ_2 satisfies (3.9). Then the system (3.4) is a Parseval frame for $L^2(\mathbb{R}^2)$.*

Proof.

$$\begin{aligned}\psi_{j,k,m} &= |\det A_2^j S_k|^{-1/2}\psi((A_2^j S_k)^{-1}x - m)\\ &= 2^{(-3/4)j}\psi(S_{-k}A_{2^{-j}}x - m)\\ &= 2^{(-3/4)j}\psi(S_{-k}A_{2^{-j}}(x - A_{2^j}S_k m))\end{aligned}$$

and so

$$\begin{aligned}\hat{\psi}_{j,k,m} &= \hat{\psi}(S_k^T A_{2^j}\omega)e^{-2\pi i\langle \omega, A_{2^j}S_k m\rangle}\\ &= \hat{\psi}(S_k^T A_{2^j}\omega)e^{-2\pi i\langle S_k^T A_{2^j}\omega, m\rangle}.\end{aligned}$$

Using (3.2),

$$\begin{aligned}\sum_{j\in\mathbb{Z}}\sum_{k\in\mathbb{Z}}|\hat{\psi}(S_k^T A_{2^j}\omega)|^2 &= \sum_{j\in\mathbb{Z}}\sum_{k\in\mathbb{Z}}|\hat{\psi}_1(2^j\omega_1)|^2\left|\hat{\psi}_2\left(\frac{k2^j\omega_1 + 2^{j/2}\omega_2}{2^j\omega_1}\right)\right|^2\\ &= \sum_{j\in\mathbb{Z}}|\hat{\psi}_1(2^j\omega_1)|^2\sum_{k\in\mathbb{Z}}\left|\hat{\psi}_2\left(2^{j/2}\frac{\omega_2}{\omega_1} + k\right)\right|^2 = 1 \qquad (3.11)\end{aligned}$$

by (3.6) and (3.9). Now, let $f \in L^2(\mathbb{R}^2)$.

$$\begin{aligned}
\sum_{j,k,m} |\langle f, \psi_{j,k,m}\rangle|^2 &= \sum_{j,k,m} |\langle \hat{f}, \hat{\psi}_{j,k,m}\rangle|^2 \\
&= \sum_{j,k,m} \left| \hat{f}(\omega)\hat{\psi}(S_k^T A_{2^j}\omega) e^{2\pi i \langle S_k^T A_{2^j}\omega, m\rangle} \right|^2 \\
&= \sum_{j,k} 2^{-3j} \sum_m \left| \iint_{[-1/2,1/2]^2} \hat{f}(A_{2^{-j}} S_{-k}\omega')\hat{\psi}(\omega') e^{2\pi i\langle \omega', m\rangle} d\omega' \right|^2 \\
&= \sum_{j,k} 2^{-3j} \int_{[-1/2,1/2]^2} |\hat{f}(A_{2^{-j}} S_{-k}\omega')|^2 |\hat{\psi}(\omega')|^2 d\omega' \\
&= \int_{\mathbb{R}^2} |\hat{f}(\omega)|^2 \sum_{j\in\mathbb{Z}} \sum_{k\in\mathbb{Z}} |\hat{\psi}(S_k^T A_{2^j}\omega)|^2 d\omega \\
&= \int_{\mathbb{R}^2} |f(x)|^2 dx \quad \text{by (3.11)}
\end{aligned}$$

where we have used Parseval's theorem to remove the sum over translations and push the absolute value inside the integral. □

We can then define the discrete shearlet transform analogously to the continuous transform.

Definition 3.3.2. The *Discrete Shearlet Transform* of $f \in L^2(\mathbb{R})$ is

$$f \mapsto \mathcal{SH}_\psi f(j,k,m) = \langle f, D_{A_2^j S_k} T_m \psi\rangle, : j,k \in \mathbb{Z}, m \in \mathbb{Z}^2.$$

3.4 Discrete Shearlets on the Cone

In order to numerically implement the transform, clearly we can only consider a finite number of shear/scaling parameters. Toward this end, the previous approach offers several problems. First, we can only cover the vertical axis in the frequency domain in the limit as $j \to \infty$. It is important that we can cover all low frequencies since most images have regions that vary little. Second, we have a directional bias, as directions close to vertical require $|s|$ to become infinitely large. Fixing these problems leads to a modification on shearlets known as *discrete shearlets on the cone* or *cone-adapted discrete shearlets*. There are several different ways to define these functions, but they are all similar in that they involve partitioning the frequency domain, typically into three regions, as depicted in Figure 3.4: the low pass region

$$\mathcal{C}^0 = \{(\xi_1, \xi_2) \in \mathbb{R}^2 : |\xi_1| \leqslant 1, |\xi_2| \leqslant 1\},$$

the horizontal cone

$$\mathcal{C}^h = \{(\xi_1, \xi_2) \in \mathbb{R}^2 : |\xi_2/\xi_1| \leqslant 1, |\xi_1| > 1\},$$

and the vertical cone

$$\mathcal{C}^v = \{(\xi_1, \xi_2) \in \mathbb{R}^2 : |\xi_2/\xi_1| > 1, |\xi_2| > 1\}.$$

We then form Parseval frames for the functions whose frequency support falls within each region. For $\mathcal{C}^0$, we use the integer translates of a single ϕ called the *shearlet scaling function*. For $\mathcal{C}^h$ and $\mathcal{C}^v$, we use the dilates, shears, and translates of *shearlet generators* ψ and $\tilde{\psi}$. Due to symmetry, once we have defined $\psi(\xi_1, \xi_2)$, we can define $\tilde{\psi}$ by $\hat{\tilde{\psi}} = \hat{\psi}(\xi_2, \xi_1)$. Note that we have fixed both problems: $\mathcal{C}^0$ takes care of the low frequencies, while splitting into two cones ensures that we do not have to use large shear parameters in order to cover all directions.

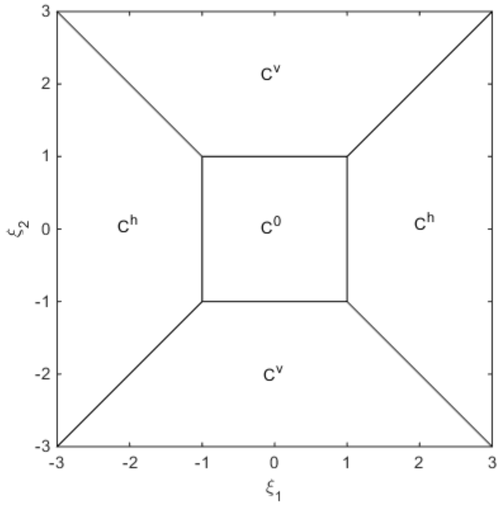


Figure 3.1: Division of frequency domain for cone-adapted shearlets

3.5 Shearlets as Composite Wavelets

Although the above provides a logical succession leading to the discrete shearlets, in fact, shearlets were born through the study of *affine systems with composite dilations* which were originally defined to have the form

$$\Psi_{AB} = \{D_a D_b T_k \psi^l : k \in \mathbb{Z}^n, a \in A, b \in B, l = 1, 2, \ldots, L\}$$

where $A, B \subset GL_n(\mathbb{R})$ and $\Psi = \{\psi_1, \ldots, \psi_L\} \subset L^2(\mathbb{R}^n)$ [13]. If Ψ_{AB} is an orthonormal (ON) basis or a Parseval frame (PF), then Ψ is called an *ON AB-multiwavelet* or a *PF AB-multiwavelet*, respectively. If $L = 1$, we use the term *wavelet* instead of *multiwavelet.*

3.6 Optimally Sparse Representations

A useful model for real images is the class of cartoon-like images.

Definition 3.6.1. The class $\mathcal{E}^2(\mathbb{R}^2)$ of *cartoon-like images* is the set of functions $f : \mathbb{R}^2 \to \mathbb{C}$ that can be decomposed as $f = f_0 + f_1\chi_B$ where $B \subset [0,1]^2$ has a boundary that is a closed C^2-curve with bounded curvature, $f_i \in C^2(\mathbb{R}^2)$ have $\text{supp}(f_i) \subset [0,1]^2$, and $||f_i||_{C^2} \leqslant 1$ for $i = 1, 2$.

Roughly, they are images that are smooth away from a smooth curve of discontinuity. It is known that for cartoon-like images, the optimal asymptotic decay rate of $||f - f_N||_2^2$ is $O(N^{-2})$ as $N \to \infty$, achieved adaptively:

Theorem 3.6.2 (Theorem 2, [9])**.** *Let $f \in \mathcal{E}^2(\mathbb{R}^2)$. There exists a constant C such that, for any N, a triangulation of $[0,1]^2$ with N triangles can be constructed so that the piecewise linear interpolation f_N of these triangles satisfies*

$$||f - f_N||_2^2 \leqslant CN^{-2}, \qquad N \to \infty.$$

We would like to achieve this with a nonadaptive process, that is, with fixed analyzing functions, a feat thought impossible until the invention of curvelets [14]. Wavelets perform optimally on images with point singularities, but only allow asymptotic mean-squared approximation error of $O(N^{-1})$ for $\mathcal{E}^2(\mathbb{R}^2)$ due to their isotropic character [15]. In more detail, let

$$\left\{\psi_{j,n}^l(x) = 2^{-j}\psi^l\left(2^{-j}x_1 - n_1, 2^{-j}x_2 - n_2\right) : j \in \mathbb{Z}, 2^j n \in [0,1)^2, l = 1,2,3\right\}$$

be an orthonormal wavelet basis for $L^2([0,1])$. If we limit the scales so that $j \leqslant J$ for some $J \in \mathbb{Z}$, then we need to use translates of a scaling function ϕ to complete the basis:

$$\mathcal{B} = \{\phi_{J,n}\}_{2^J n \in [0,1)^2} \cup \{\phi_{j,n}^l\}_{j \leqslant J, 2^j n \in [0,1)^2, 1 \leqslant l \leqslant 3}.$$

We consider $f \in BV([0,1]^2)$, i.e.,

$$||f||_V = \int_0^1 \int_0^1 |\nabla f(x_1, x_2)| dx_1 dx_2 < \infty$$

where the gradient is taken in the weak sense.

Theorem 3.6.3 (Theorem 9.7, [15]). *Let $f \in BV([0,1]^2)$. Denote by $f_{\mathcal{B}}^r[k]$ the kth largest wavelet coefficient in absolute value, excluding the scaling coefficients. Let f_N be the best N-term approximation of f using wavelet coefficients. Then we have the asymptotic decays*

$$|f_{\mathcal{B}}^r[k]| \leqslant Ck^{-1}$$

and

$$||f - f_N||_2^2 \leqslant C^2 N^{-1}$$

for some constant $C > 0$.

Since cartoon-like images have bounded variation, we have the same result for $\mathcal{E}^2(\mathbb{R}^2)$.

We would need to use analyzing functions that can have small support in many different directions, which traditional wavelets cannot do since they rely on fixed dilations and translations alone. Two additional operations that we could use are rotations and shears. Rotations are used in the curvelet transform, which is essentially optimally sparse for cartoon-like images with mean-squared approximation error $O((\log N)^3 N^{-2}), N \to \infty$ [16]. The drawback to curvelets is that most rotations do not preserve the integer lattice, so there is no direct transition from the continuum to the discrete setting. Shears can more easily preserve the integer lattice and hence provide such a transition. In addition, shearlets can provide the same asymptotic approximation error. This was first proved in [17]. As before, it is assumed that ψ satisfies (3.2), $\hat{\psi}_1, \hat{\psi}_2 \in C^\infty(\mathbb{R})$, $\operatorname{supp}(\hat{\psi}_2) \subset [-1,1]$, and (3.10). The support of $\hat{\psi}_1$ is altered to $[-1/2,-1/16] \cup [1/16,1/2]$ and (3.6) is replaced with

$$\sum_{j\geqslant 0} \left|\hat{\psi}_1\left(2^{-2j}\omega\right)\right|^2 = 1 \qquad \text{for } |\omega| \geqslant 1/8. \tag{3.12}$$

Due to the support of $\hat{\psi}_1$, we could equivalently sum over all $j \in \mathbb{Z}$. Then the shearlet system

$$\left\{\psi_{j,k,m}(x) = 2^{3j/2}\psi\left(S_k A_4^j x - m\right) : j \geqslant 0, -2^j \leqslant k \leqslant 2^j, m \in \mathbb{Z}^2\right\} \tag{3.13}$$

can be shown similarly as above to be a Parseval frame for L^2 functions with frequency support in a horizontal cone. An asymptotic decay on the shearlet coefficients is given by the following theorem.

Theorem 3.6.4 (Theorem 1.1, [17]). *Let $f \in \mathcal{E}^2(\mathbb{R}^2)$ and $\{s_{j,k,m}(f) = \langle f, \psi_{j,k,m}\rangle\}$ be the sequence of shearlet coefficients associated with the horizontal cone. Let $|s(f)|_{(N)}$ denote the Nth largest shearlet coefficient in absolute value. Then*

$$\sup_{f\in\mathcal{E}^2(\mathbb{R}^2)} |s(f)|_{(N)} \leqslant CN^{-3/2}(\log N)^{3/2}.$$

Let f_N be the N-term approximation to f using the largest N coefficients. We have the bound

$$||f - f_N||_2^2 \leqslant \sum_{m>N} |s(f)|_{(m)}^2.$$

Using Theorem 3.6.4 in the above, and the bound

$$\sum_{m>N} m^{-3}(\log m)^3 \leqslant \int_N^\infty x^{-3}(\log x)^3 = O(N^{-2}(\log N)^3)$$

we have the sparsity result:

Theorem 3.6.5 (Theorem 1.2, [17]). *Let $f \in \mathcal{E}^2(\mathbb{R}^2)$ and f_N be the best N term approximation to f. Then*

$$||f - f_N||_2^2 \leqslant CN^{-2}(\log N)^3$$

where C is independent of f and N.

This result was later extended to compactly supported shearlets [18] and 3-dimensional shearlets [19–21], though in the latter case, the optimally sparse reconstruction error is $O(N^{-1}(\log N)^2)$. Shearlet analogs in higher dimensions have also been considered [22, 23].

3.7 Implementations of the Shearlet Transform

We now turn to the various ways the shearlet transform has been implemented to process 2 and 3-dimensional finite signals. There are currently three freely available shearlet toolboxes. The first one was developed by Easley, Labate, and Lim in 2008 [24], which we refer to as the local shearlet toolbox. This 2-D implementation was followed by a 3-D implementation by Negi and Labate in 2012 [25]. The second toolbox is Shearlab developed by Kutyniok et al. in 2012 [26], which led to Shearlab 3D [27]. Most recently, in 2013, Häuser and Steidl released the Fast Finite Shearlet Transform (FFST) [12, 28]. We describe the first and third toolboxes in some detail below.

3.7.1 Local Shearlet Toolbox

The authors of the local shearlet toolbox propose two implementations: one in the frequency domain and another in the time domain. Both implementations utilize the Laplacian pyramid to separate the image by scale/frequency. The Laplacian pyramid applies a low pass filter approximating the Laplacian of the Gaussian to an image, which is then subtracted from the original image, effectively decomposing the image into a high pass and low pass part. The low pass part is then downsampled and the decomposition/downsampling process is continued until the desired number of scales is reached or the image can no longer be downsampled. In the first implementation, the authors perform the pseudo-polar discrete Fourier transform (PDFT), which is an FFT with respect to pseudo-polar coordinates $(u, v) \in \mathbb{R}^2$ defined by

$$(u, v) = \begin{cases} (\xi_1, \frac{\xi_2}{\xi_1}) & : (\xi_1, \xi_2) \in \mathcal{C}^h \\ \\ (\xi_1, \frac{\xi_1}{\xi_2}) & : (\xi_1, \xi_2) \in \mathcal{C}^v \end{cases}$$

where

$$\mathcal{C}^h = \left\{ (\xi_1, \xi_2) \in \hat{\mathbb{R}}^2 : |\xi_1| \geqslant \frac{1}{8}, \left| \frac{\xi_2}{\xi_1} \right| \leqslant 1 \right\}$$

and

$$\mathcal{C}^v = \left\{ (\xi_1, \xi_2) \in \hat{\mathbb{R}}^2 : |\xi_2| \geqslant \frac{1}{8}, \left| \frac{\xi_1}{\xi_2} \right| \leqslant 1 \right\},$$

are the horizontal and vertical cones, respectively. Our assumptions on $\hat{\psi}$ and $\hat{\tilde{\psi}}$ make these coordinates a natural choice. Next, we perform band-pass filtering, keeping the frequencies concentrated in each sheared wedge. Performing an inverse PDFT completes the transform.

The time-domain implementation (the Non-Subsampled Shearlet Transform, or NSST) utilizes a non-subsampled Laplacian pyramid and is performed by convolving the image with shearing filters to perform band-pass filtering. The authors provide two choices of filters: a Meyer wavelet window and a simple characteristic function window. Other choices to be made are the number of scales to consider (determined by the levels of decomposition of the Laplacian pyramid), the size of the filters used, and the number of directional subbands at each scale. Choosing smaller filters makes the transform more local; however, the maximal number of subbands must be less than the size of the filters. In the original paper, the authors vary these choices in a denoising application. They conclude that the best choice is to use Meyer-based shearlet filters of size 16 with 16 directions on the first two levels of decomposition and filters of size 32 with 8 directions on the third and fourth levels of decomposition. Only the code for this implementation is publicly available.

3.7.2 Fast Finite Shearlet Transform

The FFST is in some ways the most intuitive implementation of the shearlet transform. We start by defining the generating shearlet ψ, assuming as allows that it satisfies (3.2). Define a function $b : \mathbb{R} \to \mathbb{R}$ by

$$b(\omega) := \begin{cases} \sin(\frac{\pi}{2}v(|\omega|-1)) & \text{for } 1 \leqslant |\omega| \leqslant 2 \\ \cos(\frac{\pi}{2}v(\frac{1}{2}|\omega|-1)) & \text{for } 2 < |\omega| \leqslant 4 \\ 0 & \text{o.w.} \end{cases}$$

where v is defined by (3.8), cf. (3.7). Then, ψ_1 is given in the frequency domain by

$$\hat{\psi}_1(\omega) := \sqrt{b^2(2\omega) + b^2(\omega)}.$$

It can be shown that $\operatorname{supp}(\hat{\psi}_1) = [-4, -\frac{1}{2}] \cup [\frac{1}{2}, 4]$ and that

$$\sum_{j \geqslant 0} |\hat{\psi}_1(2^{-2j}\omega)|^2 = 1 \quad \text{for } |\omega| > 1.$$

Next, ψ_2 is defined by

$$\hat{\psi}_2(\omega) := \begin{cases} \sqrt{v(1+\omega)} & \text{for } \omega \leqslant 0 \\ \\ \sqrt{v(1-\omega)} & \text{for } \omega > 0. \end{cases}$$

This function has support $[-1, 1]$ and satisfies (3.10).

The FFST uses a modified form of cone-adapted shearlets where the frequency domain is divided into four non-disjoint sets:

$$\mathcal{C}^0 := \{(\omega_1, \omega_{)} \in \mathbb{R}^2 : |\omega_1| < 1, |\omega_2| < 1\}$$

$$\mathcal{C}^h := \{(\omega_1, \omega_2) \in \mathbb{R}^2 : |\omega_2/\omega_1| < 1, |\omega_1| \geqslant 1/2\}$$

$$\mathcal{C}^v := \{(\omega_1, \omega_2) \in \mathbb{R}^2 : |\omega_1/\omega_2| < 1, |\omega_2| \geqslant 1/2\}$$

$$\mathcal{C}^\times := \{(\omega_1, \omega_2) \in \mathbb{R}^2 : |\omega_1| \geqslant 1/2, |\omega_2| \geqslant 1/2, |\omega_1| = |\omega_2|\}.$$

The last region $\mathcal{C}^\times$ is new, representing the intersection of the horizontal and vertical cones (the seam lines). We also note that $\mathcal{C}^0$ now intersects non-trivially with both cones. This allows our shearlets to transition more smoothly across region boundaries. We address each region in turn by defining the generating shearlet.

For the low-pass region, $\mathcal{C}^0$, we use the Meyer scaling function, whose Fourier transform is given by

$$\hat{\varphi}(\omega) := \begin{cases} 1 & \text{for } |\omega| \leqslant 1/2 \\ \cos(\frac{\pi}{2}v(2|\omega|-1)) & \text{for } 1/2 < |\omega| < 1 \\ 0 & \text{o.w.} \end{cases}$$

to define the full scaling function

$$\hat{\phi}(\omega_1, \omega_2) := \begin{cases} \varphi(\omega_1) & \text{for } |\omega_2| \leqslant |\omega_1| \\ \varphi(\omega_2) & \text{for } |\omega_1| < |\omega_2| \end{cases}.$$

For $\mathcal{C}^h$ and $\mathcal{C}^v$, we use

$$\hat{\psi}^h(\omega_1, \omega_2) = \hat{\psi}_1(\omega_1)\hat{\psi}_2\left(\frac{\omega_2}{\omega_1}\right)$$
$$\hat{\psi}^v(\omega_1, \omega_2) = \hat{\psi}_1(\omega_2)\hat{\psi}_2\left(\frac{\omega_1}{\omega_2}\right).$$

Finally, for $\mathcal{C}^\times, \hat{\psi}^h = \hat{\psi}^v$ and we use the common values to define $\hat{\psi}^x$ there.

Now, let f be an $M \times N$ image, considered as a function on the grid $\mathcal{G} := \{(m_1/M, m_2/N) : m_1 = 0, \ldots, M-1, m_2 = 0, \ldots, N-1\}$ with periodic extension. Define $j_0 := \lfloor \log_2 \max\{M, N\} \rfloor$. We discretize the parameters as follows:

$$a_j := 2^{-2j} = \frac{1}{4^j}, \qquad j = 0, \ldots, j_0 - 1,$$
$$s_{j,k} := k2^{-j}, \qquad -2^j \leqslant k \leqslant 2^j,$$
$$t_m := \left(\frac{m_1}{M}, \frac{m_2}{N}\right), \quad m \in \mathcal{G}.$$

Our shearlets then are given by

$$\psi^\kappa_{j,k,m}(x) := \psi^\kappa_{a_j, s_{j,k}, t_m} = \psi^\kappa(A^{-1}_{a_j} S^{-1}_{s_{j,k}}(x - t_m)), \quad \kappa = h, v.$$

With this choice of parameters, the shearlets have support entirely within one of the cones, except for $|k| = 2^j$. In this case, the shearlets in both cones agree, so we can choose either function, which we will call $\psi^{h\times v}$. The low-pass region is handled by translations of the full scaling function: $\phi_t(x) := \phi(x - t)$. With calculations almost identical to those in (3.3) except in the discrete setting, we can develop simple formulas for the shearlet coefficients. For instance, in the horizontal cone, we have

$$\begin{aligned}
\mathcal{SH}(f)(j,k,m) &= \langle f, \psi^h_{j,k,m} \rangle \\
&= \langle \hat{f}, \hat{\psi}^h_{j,k,m} \rangle \\
&= \mathcal{F}^{-1}(\hat{f}(\omega_1,\omega_2)\hat{\psi}(4^{-j}\omega_1, 4^{-j}k\omega_1 + 2^{-j}\omega_2))
\end{aligned}$$

where $\omega_1 = -\lfloor M/2 \rfloor, \ldots, \lceil M/2 \rceil - 1, \omega_2 = -\lfloor N/2 \rfloor, \ldots, \lceil N/2 \rceil - 1$. The Fourier transforms and inverse Fourier transforms are to be interpreted as discrete Fourier transforms, and hence can be implemented by $2D$ FFTs. After a lengthy and technical series of calculations, it can be shown that the shearlets form a Parseval frame for $L^2(\mathcal{G})$.

Theorem 3.7.1 (Theorem 3.1, [12]). *The discrete shearlet system*

$$\begin{aligned}
&\{\psi^h_{j,k,m}(x) : j = 0,\ldots,j_0 - 1, -2^j + 1 \leqslant k \leqslant 2^j - 1, m \in \mathcal{G}\} \\
&\cup \{\psi^v_{j,k,m}(x) : j = 0,\ldots,j_0 - 1, -2^j + 1 \leqslant k \leqslant 2^j - 1, m \in \mathcal{G}\} \\
&\cup \{\psi^{h\times v}_{j,k,m}(x) : j = 0,\ldots,j_0 - 1, |k| = 2^j, m \in \mathcal{G}\} \\
&\cup \{\phi_m(x) : m \in \mathcal{G}\}
\end{aligned}$$

forms a Parseval frame for $L^2(\mathcal{G})$.

3.8 Shearlet Applications in the Literature

Shearlets mainly have applications in image processing tasks. They have been used for image registration [29], inpainting [30, 31], denoising [24, 32], segmentation [33, 34], data reconstruction [35], image interpolation [36], and edge detection/classification [37, 38].

Much of the cited literature promotes shearlets as being better than wavelets at analyzing images with edges, owing to the optimal sparsity of shearlets. It would then be interesting to check whether we can actually obtain the expected approximation error decline using the available implementations. Real images are discrete signals with a limited number of scales, and hence we would not necessarily expect asymptotic results on continuous signals to carry over. In [24], the authors compare their frequency-domain approach to an unspecified discrete wavelet transform, as well as check the rate of error decline as a function of the number of coefficients kept. They note that if we assume $||f - f_N||_2 = CN^{-\alpha}$, then a log-log plot of L^2 approximation error vs. number of coefficients would have slope -1 if we have equality in Theorem 3.6.5. Their first test image is a fingerprint, a prototypical example of an image with many curves of discontinuity. They account for the added redundancy of shearlets by multiplying by the redundancy factor. Their results do show lower error for shearlets as compared to wavelets, at least within the range shown. However, there is no appreciable difference in the error decline; $\alpha \approx .25$ in both cases. The authors then use a second image containing three concentric circles, from which they obtain $\alpha \approx .9634$, but there is no comparison to the wavelet transform for this image.

We decided to investigate this issue further by considering the shearlet transform on two very different images: the cameraman and the mandrill, Figs. 3.2-3.3. Both were converted to 256×256 grayscale images. The former contains few discontinuity curves while the latter contains a large number. For the shearlet transform, the FFST implementation was used with four shearing scales, resulting in 3997696 coefficients. This was chosen since it is most similar to the setup for which the sparsity result was proved. The wavelet transform was a four-level decomposition using the db4 wavelet, resulting in 72034 coefficients. As in [24], we multiply the wavelet coefficients by the redundancy factor (≈ 55.5 here) in order to compare on the same plot.

The results for the cameraman and mandrill image are found in Figs. 3.4-3.5. There are three distinct regions for all curves: the left side which corresponds to the low frequency coefficients, the middle, and the right side where we see the effect of only having a finite number of scales. We are interested in the middle region whose approximate slope gives the asymptotic error decay rate. In the cameraman image, we have $\alpha \approx 1.20$ using shearlets and $\alpha \approx 0.98$ using wavelets. For the mandrill image, we have $\alpha \approx 1.17$ and $\alpha \approx 0.56$, respectively. We observe that shearlets do give a better error decay rate than wavelets. The difference is most apparent in the mandrill image, where wavelets perform half as well as shearlets. Furthermore, shearlets perform about the same regardless of the image's edginess, whereas wavelet performance is strongly influenced by the number of edges. In terms of the error values, wavelets outperform shearlets when few coefficients are kept, but at some point this behavior is reversed. For the cameraman image, shearlets give lower error when more than 80% of coefficients are kept. For the mandrill image, this number is about 52%, again illustrating that shearlets are most useful on images with many edges. We should note, however, that the difference in error values may be hard to interpret due to the fact that 1/256 of the wavelet coefficients are low-pass, compared to 1/61 of the shearlet coefficients.

Unlike wavelets, this shearlet implementation, and in fact all implementations, are highly redundant. Therefore, we would only expect shearlets to be useful in applications for which nonsubsampled wavelets have proved useful, such as denoising [39] and edge detection [40]. The wavelet and shearlet reconstructions of the images resulting in the same error were visually indistinguishable, but the error distributions were quite different. In the cameraman image, keeping 80% of the coefficients resulted in a relative L^2 error of 1.74×10^{-4} using wavelets and 1.78×10^{-4} using shearlets. In the mandrill image, keeping 56% of the coefficients resulted in a relative L^2 error of 2.00×10^{-3} using wavelets and 1.97×10^{-3} using shearlets. Fig. 3.6 and 3.7 show the distribution of error in both images. With wavelets (left plot), the error is small near distinct, strong edges, but is otherwise uniform. With shearlets (right plot), the error is not only small at strong edges, but also at weak edges and textures. Most shearlet error is concentrated in regions that appear constant; in fact, these regions have many small variations that are invisible to the naked eye.

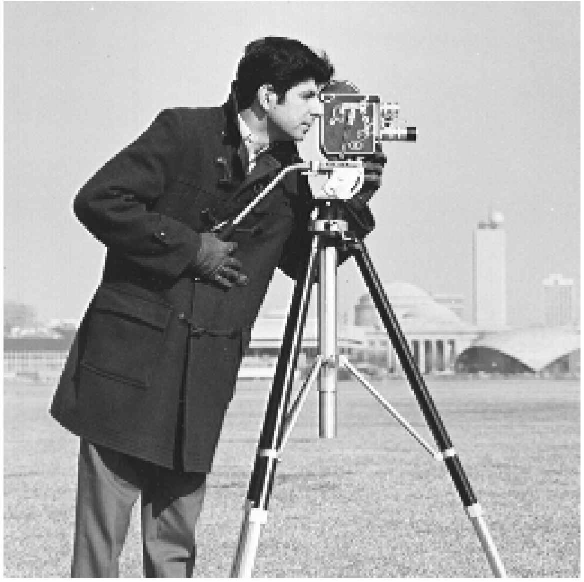


Figure 3.2: The cameraman test image.

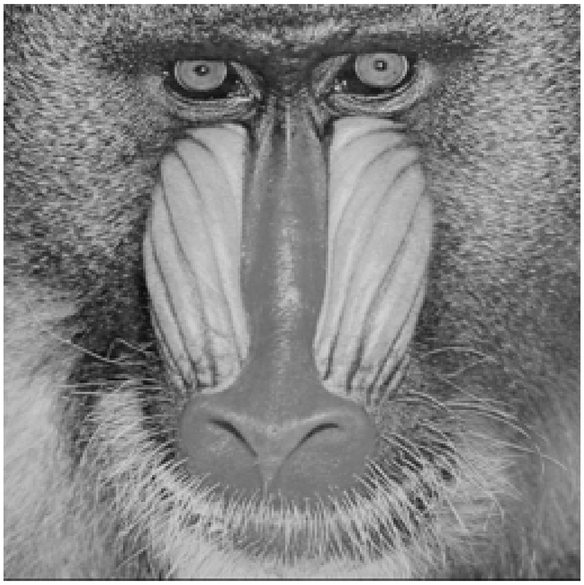


Figure 3.3: The mandrill test image.

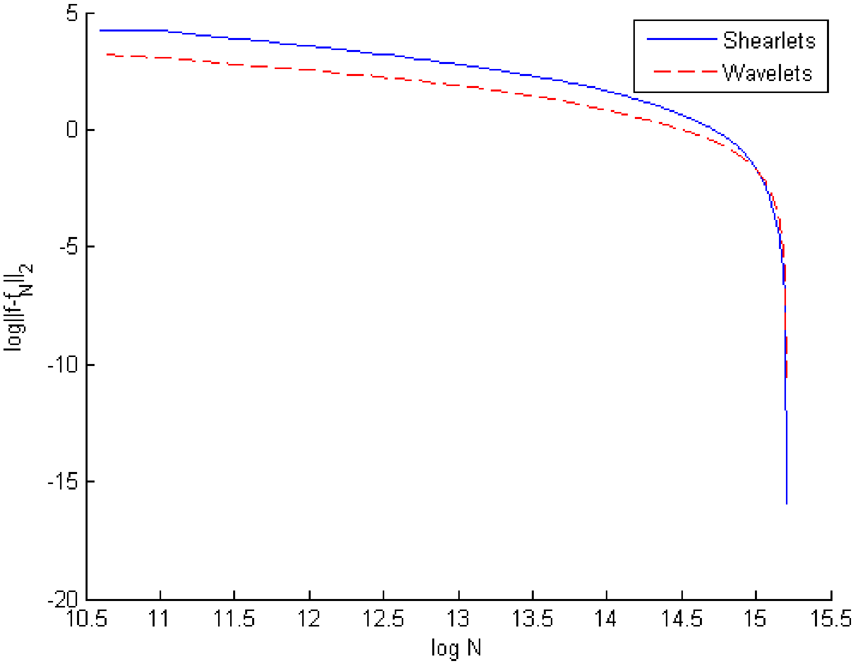


Figure 3.4: A log-log plot of $||f - f_N||_2$ vs. N for the cameraman image. The shearlet curve has $\alpha \approx 0.85$. The wavelet curve has $\alpha \approx 0.72$. The intersection point of the curves corresponds to 82.6%.

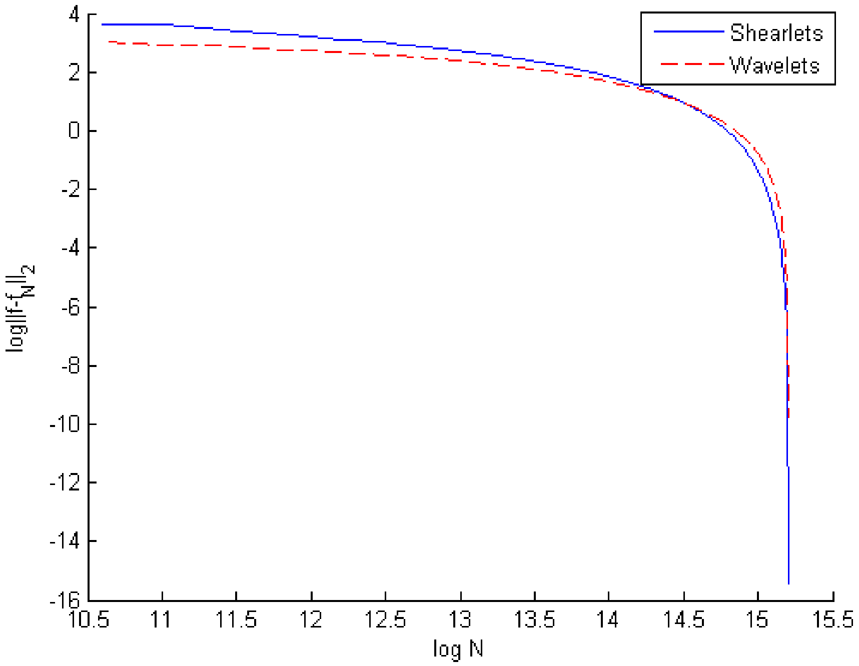


Figure 3.5: A log-log plot of $||f - f_N||_2$ vs. N for the mandrill image. The shearlet curve has $\alpha \approx 0.55$. The wavelet curve has $\alpha \approx 0.42$. The intersection point of the curves corresponds to 49.8%.

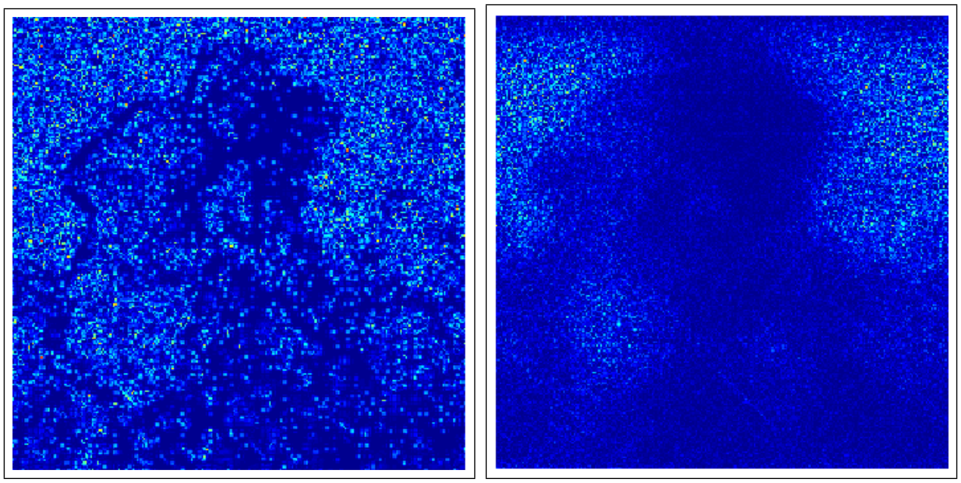


Figure 3.6: Concentration of absolute error in the cameraman image using wavelets (left) and shearlets (right) while keeping 82.6% of the coefficients. The relative L^2 errors are 9.40×10^{-4} and 9.33×10^{-4}, respectively.

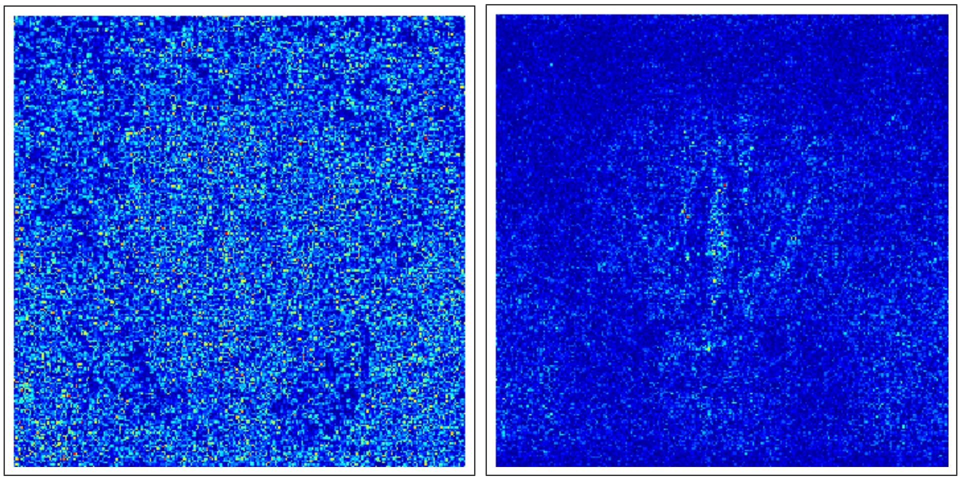


Figure 3.7: Concentration of absolute error in the mandrill image using wavelets (left) and shearlets (right) while keeping 49.8% of the coefficients. The relative L^2 errors are both 1.90×10^{-2}.

Chapter 4: A Shearlet Application to LIDAR

4.1 Introduction

An essential component in the design of high-resolution, multi-resolution, accurate 3D terrain models, is directional information. Traditional wavelets, while able to detect isolated discontinuities, are isotropic objects. As a result, they are not very effective in detecting curves of discontinuity. In response to the limitations of wavelets, many directional representations were proposed, including contourlets [41], curvelets and ridgelets [42], bandelets [43], wedgelets [44], and shearlets [17]. These representations not only include translations and dilations, but also geometric tranformations that provide orientation information. We focus on shearlets since they have a solid theoretical backing as well as several efficient numerical implementations. In this chapter, we utilize shearlets to detect linear and quasilinear structures in LIDAR images of rural areas.

4.2 What is LIDAR?

LIDAR (Light Detection and Ranging) is a remote-sensing technique that uses lasers to acquire elevation data of a scene. LIDAR systems require a collection vehicle, GPS (Global Positioning System), and INS (Inertial Navigation System) [45]. In the data collection process, aerial vehicles, such as planes or helicopters, fly along a survey route while emitting light pulses. The reflected light is received by sensors on the vehicle; elevation data can be inferred based on the time it takes for the light to return. Objects on the ground that allow for the passage of some light, such as trees, result in multiple returns. For instance, some light may be reflected by the top of a tree, a branch further down, or the ground beneath the tree. In this way, we are able to receive information on structures normally hidden under trees. Some LIDAR data also includes intensity values, the strength of the returning signal, which measures material reflectivity. This can be used to classify objects by material composition. Recently, LIDAR has become an indispensable component of self-driving cars with its ability to quickly deliver an accurate map of the surrounding area [46].

4.3 Road Detection

The topic of road detection by remote sensing is an important and challenging problem. According to [47], a road must satisfy six physical and geometric requirements:

1. Its surface must be smooth, without breaks.

2. It must be made of firm material.

3. The steepness must have an upper bound.

4. The width of the road must have an upper bound (in practice, 146 ft).

5. The local radius of curvature must have an upper bound (in practice, 1200 ft).

6. Roads nearby are connected and form a network.

Some examples of road detection algorithms utilizing aerial LIDAR are found in [48–50]. There are also algorithms for LIDAR road detection by autonomous cars [51, 52].

4.4 Edge Detection

Since there is typically an abrupt change in signal intensity at the edge of roads, edge detectors are typically employed in road detection schemes [53]. One of the most popular edge detectors is the Canny algorithm, as evidenced by the fact that the original paper [54] has been cited over $20,000$ times. Essentially, the Canny algorithm looks for local maxima in gradient intensity. The algorithm can be broken up into four steps. First, a Gaussian filter with standard deviation σ is applied to the image to remove noise. Second, a numerical gradient is computed at each point. Third, *non-maximum suppression* is applied to the gradient magnitudes. In this step, a point is determined to not be on a edge if its gradient magnitude is no larger than that of its two neighbors in the gradient's (quantized) direction, resulting in *edge thinning*. Finally, a double thresholding technique called *hysteresis* is performed. Two thresholds, T_{low} and T_{high}, are specified such that $T_{low} < T_{high}$. If the gradient at a point is larger than T_{high}, it is marked as an edge point. The neighbors of an edge point are then checked for whether their gradients are between the thresholds. Any such point is marked as an edge point, and the process continues. This is an example of *edge linking*. Below is the output for the Canny algorithm on the cameraman image with $T_{low} = 0.0375, T_{high} = 0.0938, \sigma = \sqrt{2}$.

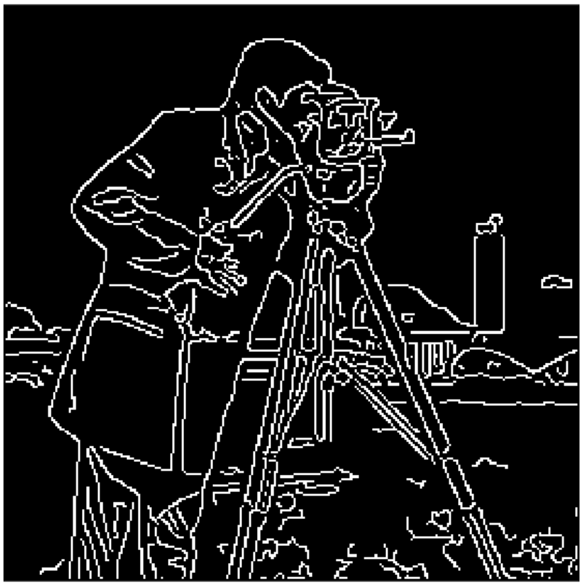


Figure 4.1: Output of the Canny algorithm on the cameraman image. Detected edges are in white.

4.4.1 Theoretical results for edge detection with shearlets

Shearlets have properties that make them effective edge detectors. We review some known theory regarding the ability of shearlets to detect edges [37]. To model an image, let $\Omega = [0,1]^2$ and decompose Ω as:

$$\Omega = \bigcup_{n=1}^{L} \Omega_n \cup \Gamma, \text{ where:}$$

1. Each Ω_n is a connected open set.

2. $\Gamma = \bigcup_{n=1}^{L} \partial\Omega_n$, where each boundary $\partial\Omega_n$ is a smooth rectifiable curve.

In order to represent images, we consider the set of functions

$$I(\Omega) := \{u(x) = \sum_{n=1}^{L} u_n(x)\chi_{\Omega_n}(x),\ x \in \Omega\backslash\Gamma\},$$

where the u_n have bounded partial derivatives and the sets Ω_n are pairwise disjoint in measure.

Theorem 4.4.1 (Theorem II, [37])**.** *Let $u \in I$. If $t \notin \Gamma$, then*

$$\lim_{a\to 0} a^{-\frac{3}{4}}\mathcal{SH}_\psi(a,s,t) = 0.$$

If $t \in \Gamma$ and in a neighborhood of $t = (t_1, t_2)$, the boundary curve is parametrized as $(E(t_2), t_2)$, and $s \neq -E'(t_2)$, then (4.4.1) also holds. Otherwise, if $s = -E'(t_2)$, there is a constant $C > 0$ such that:

$$\lim_{a\to 0} a^{-\frac{3}{4}}\mathcal{SH}_\psi(a,s,t) = C|[u]_t|,$$

where $[u]_t$ is the jump of u at t, occurring in the direction normal to the jump.

Intuitively, this means edges can be determined based on the asymptotic decay of the continuous shearlet transform. This asymptotic decay will be slower when t is on the boundary, and s corresponds to the normal direction to Ω at t. Thus, the asymptotic behavior determines both the location of edges and their local orientation. This result can be improved by incorporating information on the regularity of u.

Theorem 4.4.2 ([37]). *Suppose $u \in I(\Omega)$ is Lipschitz of order α near $t_0 \in \mathbb{R}^2$. Then, for $\alpha \geqslant 0$,*

$$|\mathcal{SH}_\psi u(a, s, t_0)| \leqslant C a^{\frac{1}{2}(\alpha+\frac{3}{2})}, \text{ as } a \to 0.$$

For $\alpha < 0$,

$$|\mathcal{SH}_\psi u(a, s, t_0)| \leqslant C a^{\alpha+\frac{3}{4}}, \text{ as } a \to 0.$$

4.5 Description of the Problem

We are given first return LIDAR data without intensities. The data has already been preprocessed by gridding, i.e., the non-uniformly sampled data has been converted to data sampled on a regular grid via interpolation. We are interested in detecting dirt roads and ditches in rural scenes using shearlets. This problem is quite difficult for a number of reasons. First, since we do not have intensities, we cannot use material information to identify the roads. Second, we only have first return data for scenes with many trees. If we had last return data, we could have an image where most of the trees are removed. Objects such as trees and bushes result in large shearlet coefficients in all directions and at all scales. Filtering and thresholding is needed in order to bring out the features we want. Third, the shearlet's size causes a "bleeding effect" in which large shearlet magnitudes will be found in a sizable region near a directional feature. Fourth, the features we seek only show up in the LIDAR data as weak edges, though optically they would appear much stronger. As a result, we can not utilize any methods which rely on edges persisting across scales such as in [55], since many of the edges are too weak to appear at finer scales.

4.6 Our Initial Approach

At first, we attempted to use the fact that large shearlet coefficients at fine scales are indicative of edges. The FFST was not very effective towards this goal; although we could see our features at the proper scale, the shearlets extended from the trees and obscured the important features. In effect, the shearlets of FFST do not decay fast enough for our purposes. See Fig. 4.2 for a demonstration of this phenomenon.

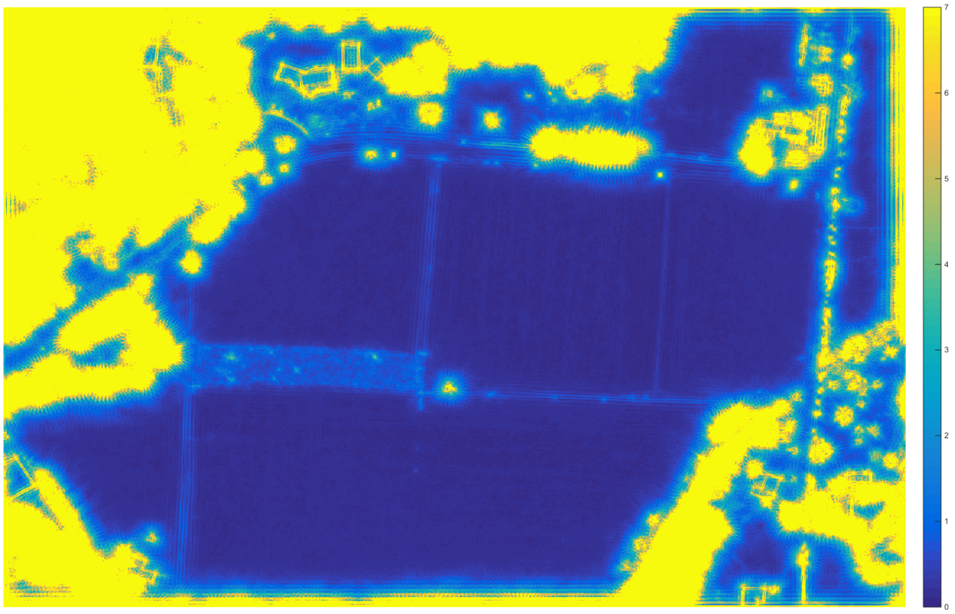


Figure 4.2: Sum of shearlet coefficients using the FFST Implementation. All values larger than 7 have been set to 7 for visualization purposes.

We next tried the local shearlet toolbox, whose implementation via small filters prevents the above phenomenon. We chose 5 scales: excluding the low pass filter, we divided each into 16 directions. This is important for our application because some of our features can only be found at the lower scales, so it is helpful to have many directions available to us. Another restriction is that all input images must be square. We managed this by either cutting out a square part of the image (if it was close to square already) or padding the image with zeros. The latter led to boundary effects where we padded the image, but was otherwise innocuous.

We consider data from the Mohawk ditch area, which was provided by the Army Research Labs. Fig. 4.3 shows the original LIDAR data; its resolution, that is the space between neighboring pixels, is 1 m. The most interesting features are the vertical and horizontal roads as well as the narrow ditches. In our first attempt to find the directional features in the data, we sum up the absolute values of the shearlet coefficients over all directions at fixed scales. We look at the log of the sum so that our view is not completely dominated by the trees. Figs. 4.4 – 4.7 show these results at each scale. We count the scales from coarsest to finest, so that scale 1 shows the lowest frequency band (excluding the low-pass region) and scale 4 shows the highest frequency band. At scales 1 and 2, we see the roads and ditches very clearly. There are also vertical and horizontal lines in the fields that may be tillage patterns. By scale 3, these patterns have disappeared, though we now see other non-linear patterns in the bottom fields. At scale 4, some of the ditches have disappeared and we see another pattern appear in the bottom fields.

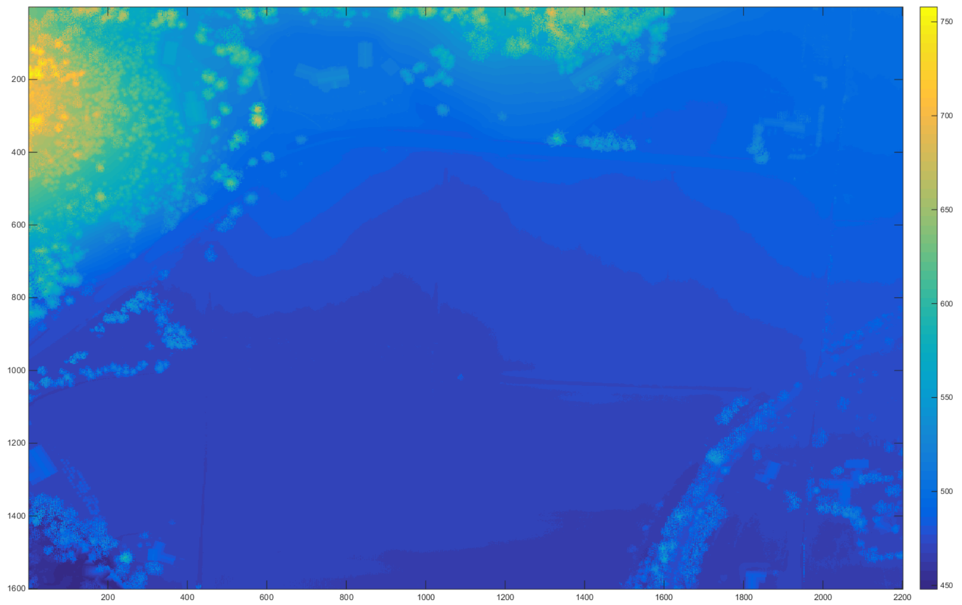


Figure 4.3: Original LIDAR image of the Mohawk ditch scene. The resolution is 1 m.

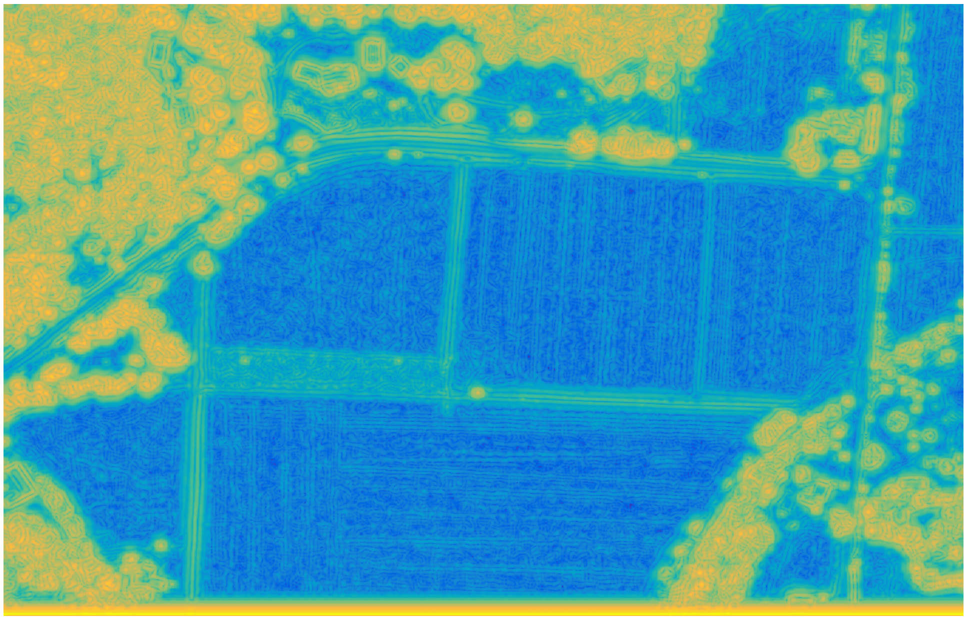


Figure 4.4: Log of sum of absolute values of shearlet coefficients at Scale 1.

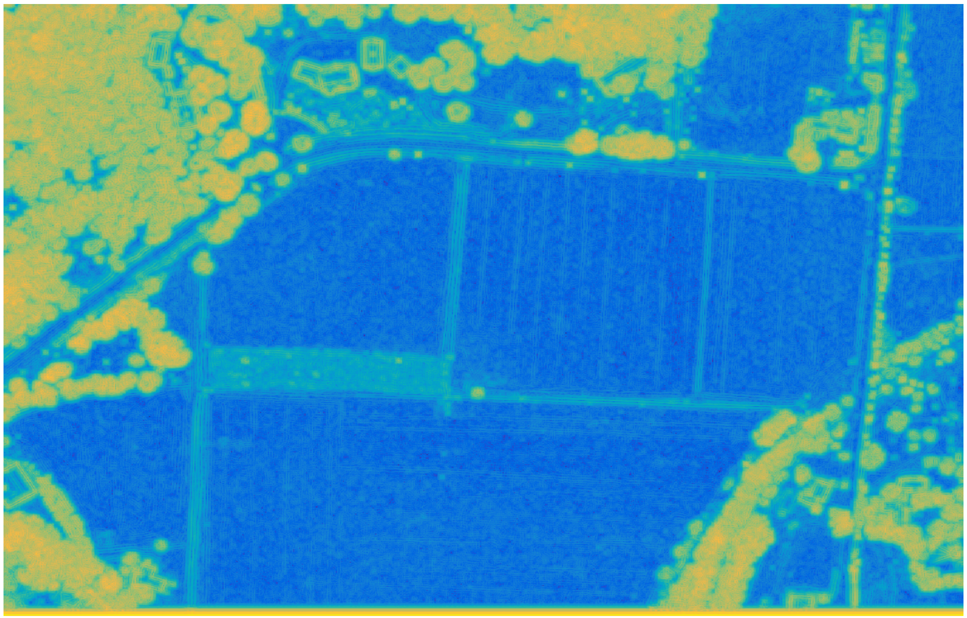


Figure 4.5: Log of sum of absolute values of shearlet coefficients at Scale 2.

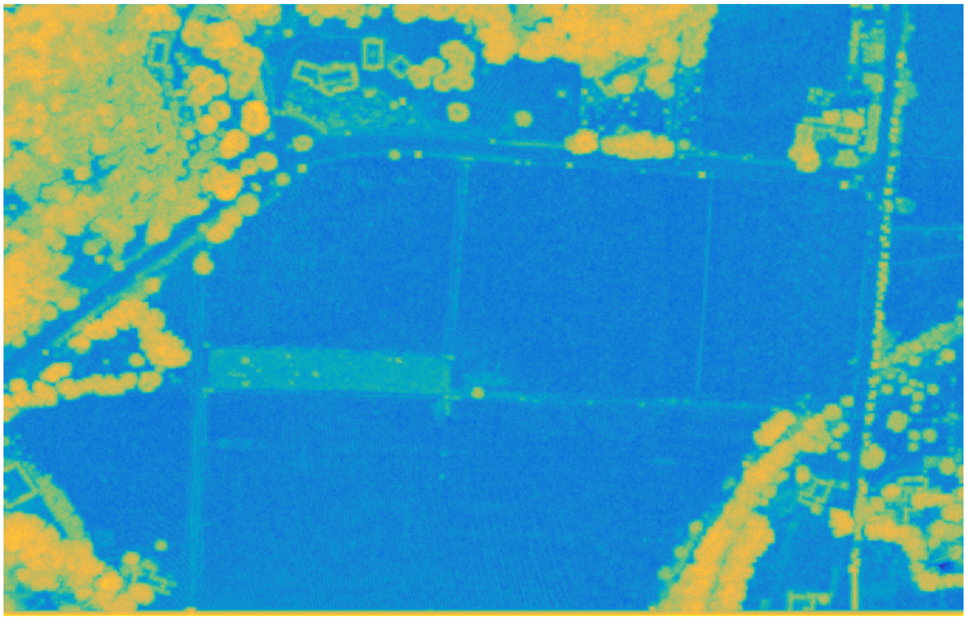


Figure 4.6: Log of sum of absolute values of shearlet coefficients at Scale 3.

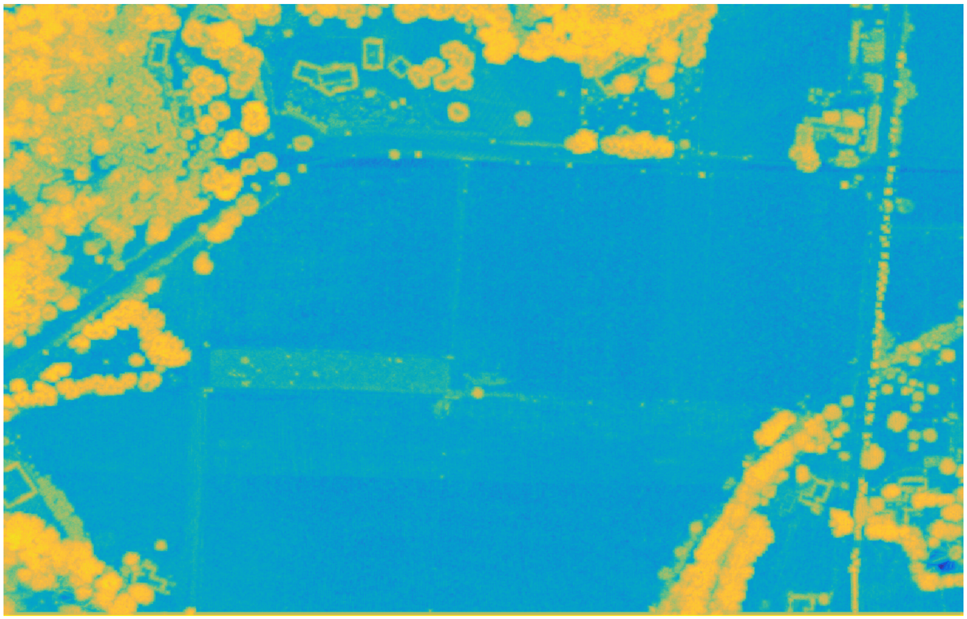


Figure 4.7: Log of sum of absolute values of shearlet coefficients at Scale 4.

We compare these pictures with the result of applying a standard deviation filter to the data and taking the logarithm, which we will refer to as the logstd of the image. This is a good way to determine the directional features that exist in the image, and therefore could potentially be detected. We use the default MATLAB command stdfilt, which computes the standard deviation of all pixels in a 3-by-3 neighborhood, with symmetric padding on the boundary. The results are in Fig. 4.8. The standard deviation-filtered image contains many of the same features we saw in the shearlet coefficients at scales 3 and 4. The horizontal and vertical lines in the field are invisible, however. Comparing the shearlet images to the standard deviation filter image, we do see one disadvantage of the shearlet approach: due to the shearlet support, we can not precisely determine the features' locations. This is less of a problem at finer scales where the shearlet support is thinner, but we tend to lose weak edges at fine scales, so exact detection of weak edges is difficult.

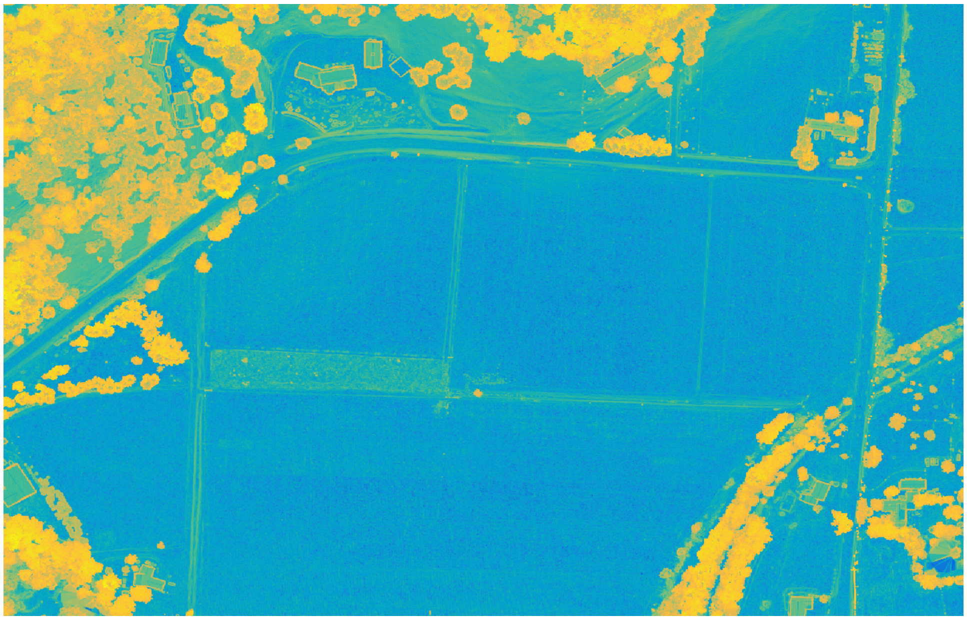


Figure 4.8: Logstd of the Mohawk ditch scene.

We would like to take advantage of the fact that shearlet coefficients are large at points with local direction, say by setting a lower threshold for the sum of the absolute values of the coefficients. However, trees have much larger shearlet coefficients than our objects of interest, so we also need to set an upper threshold. In Fig. 4.9, we present the locations at which the this sum is between 0.12 and 1. The ditches and edges of roads are apparent, but there are quite a few extraneous locations. One of the biggest sources of these extra locations is the edges of trees, which cannot be filtered out by thresholding the shearlet coefficients. We next present an approach that results in fewer false positives.

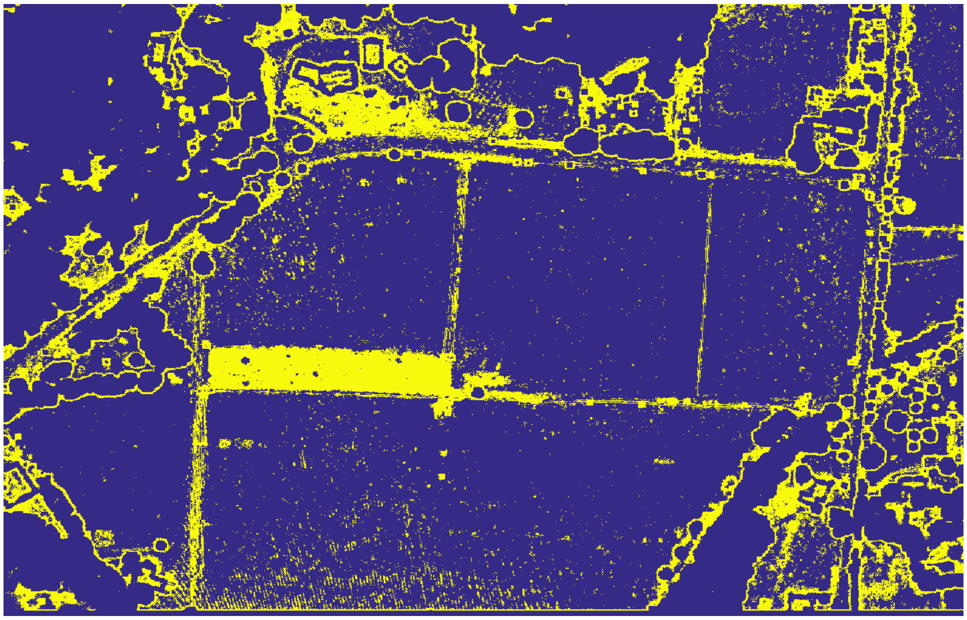


Figure 4.9: Locations where the sum of shearlet coefficients is between 0.12 and 1 at scale 3 (in yellow).

4.7 A New Approach

Since our results were less than satisfactory in the previous section, we developed a different approach. In the ideal case, an image will only have a well-defined local direction at the features we are interested in. Here, the shearlet coefficients would be large in a single direction, while being small in the other directions. Hence, instead of looking for points with large shearlet coefficients, we could equivalently look for points with many small shearlet coefficients and at least one large coefficient. Utilizing this for our LIDAR data, we would need to set a lower threshold on the local standard deviation in order to filter out flat areas. Trees and buildings can be filtered out either by having an upper threshold for the local standard deviation or by having an upper threshold on height with respect to the lowest point. The latter method can only be effective if the terrain is flat, which is generally not the case for large scenes. The steps of the new algorithm are as follows.

1. Let A be the original LIDAR image. Define $B := A - \min(A)$.

2. Compute the shearlet transform of B. Here, we have defaulted back to using the FFST. Restrict to a scale of interest.

3. Specify four thresholding parameters: $loglow, loghi, elevhi$, and $shearhi$. These parameters are thresholds on the logstd, the elevation from the minimum point, and the shearlet coefficient, respectively.

4. Each point that satisfies the standard deviation and elevation thresholds are candidates for the edge points we are interested in. For each of the points, record the number of directions whose shearlet coefficient has magnitude below $shearhi$.

5. Remove the points that have small shearlet coefficients in all directions.

6. Apply a median filter to the resulting image to clean up the image. The median filter is determined by a vector $mfilt = [m, n]$, where each pixel value is replace by the median value in an m by n neighborhood.

4.8 Results

4.8.1 Mohawk Ditch

We return to the Mohawk ditch scene and perform our new algorithm using the shearlet coefficients at scale 3. As parameters, we choose $loglow = -2.4, loghi = 0, shearhi = 0.1, elevhi = 50$, and $mfilt = [5, 3]$. To show the effect of the median filter, we present the output before and after filtering in Figs. 4.10 and 4.11, respectively. Note that the filter removes some unwanted isolated pixels at the cost of removing thin edges.

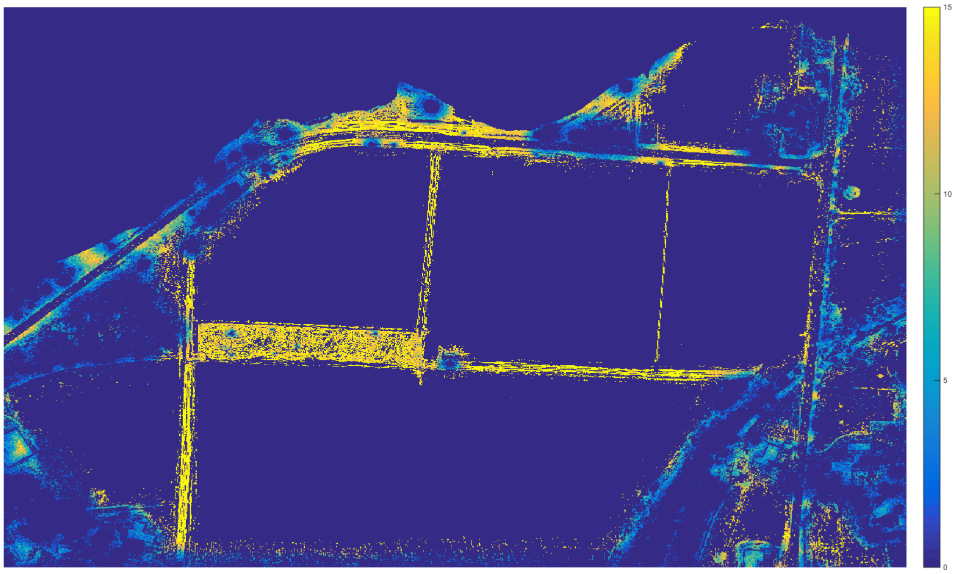


Figure 4.10: Results of the algorithm on the Mohawk ditch scene without the median filter. Yellow pixels correspond to the strongest directional features, being weakly directional in 15 of 16 directions, while light blue pixels are weakly directional in only 1 direction.

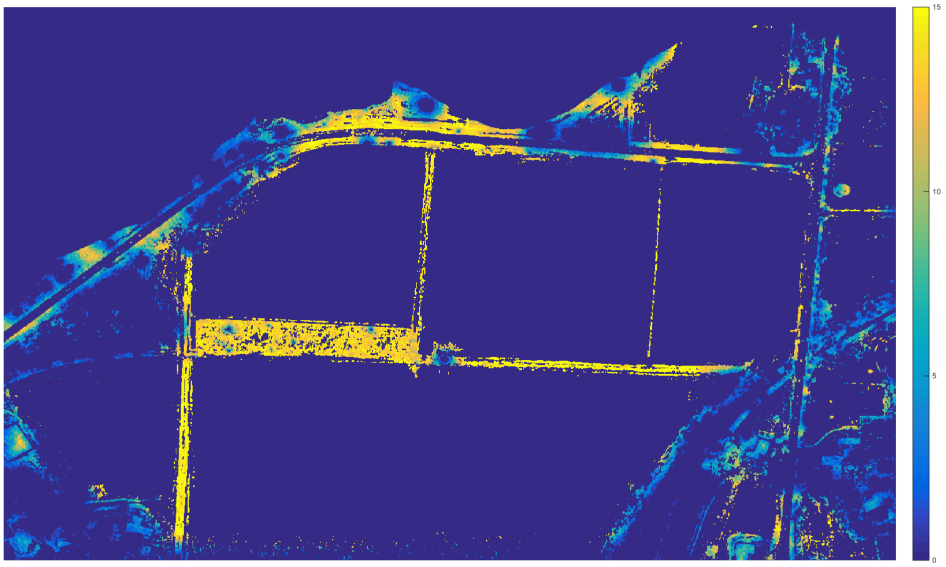


Figure 4.11: Results of the algorithm on the Mohawk ditch scene with the median filter.

We observe that our new algorithm performs fairly well in detecting the roads and ditches, without being misled by tree borders. The strong edges (in yellow) for which 15 of the 16 shearlet coefficients are small correspond to actual roads and ditches for the most part. We still get false positives for buildings in the lower left and right; LIDAR intensities could be used to distinguish these buildings from roads and ditches. The textured area in the middle of the scene is also detected since locally it is similar to what we are looking for. Placing an upper bound on the width of detected features may alleviate this issue.

4.8.2 Mohawk Dirt Tracks

We next consider the Mohawk dirt tracks scene. Fig. 4.12 shows the original LIDAR data and Fig. 4.13 shows the logstd.

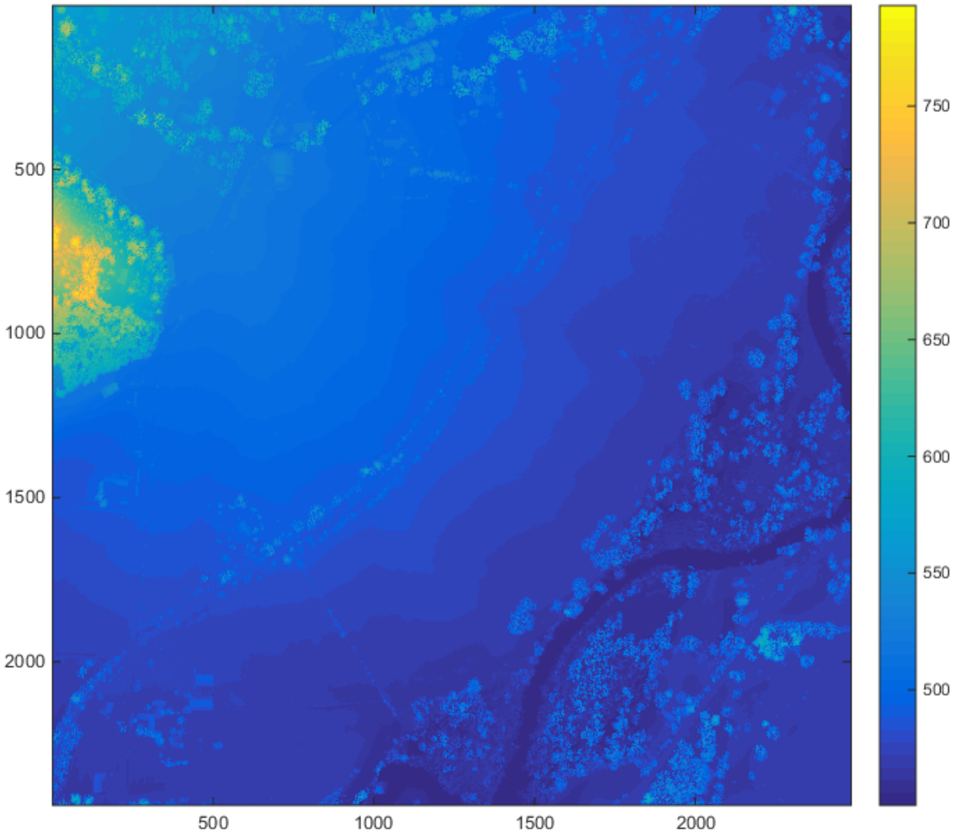


Figure 4.12: Original LIDAR image for the Mohawk dirt tracks scene. The resolution is 1.4 m.

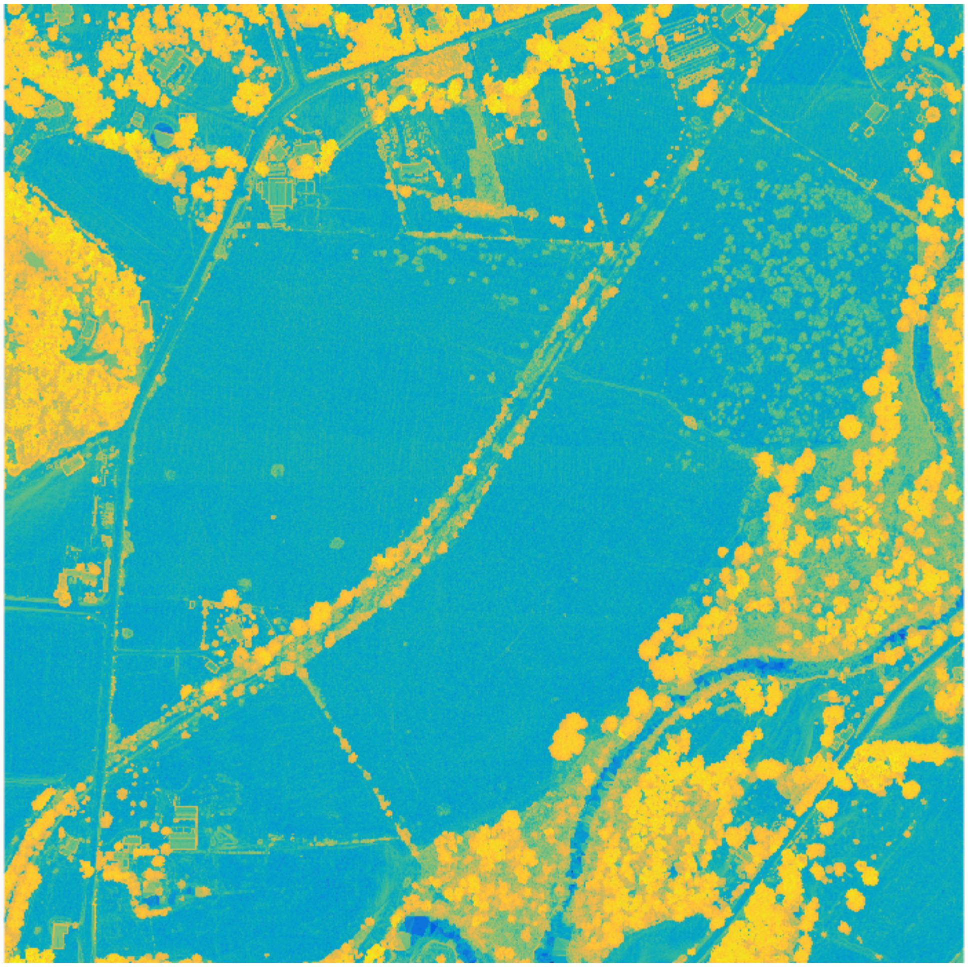


Figure 4.13: Logstd of the Mohawk dirt tracks scene.

We are interested in detecting tracks and drainage ditches. We consider the third scale of shearlet coefficients and use as parameters for the algorithm $loglow = -2, loghi = 0, shearhi = 0.1, elevhi = 100$, and $mfilt = [3,3]$. The results of the algorithm are found in Fig. 4.14.

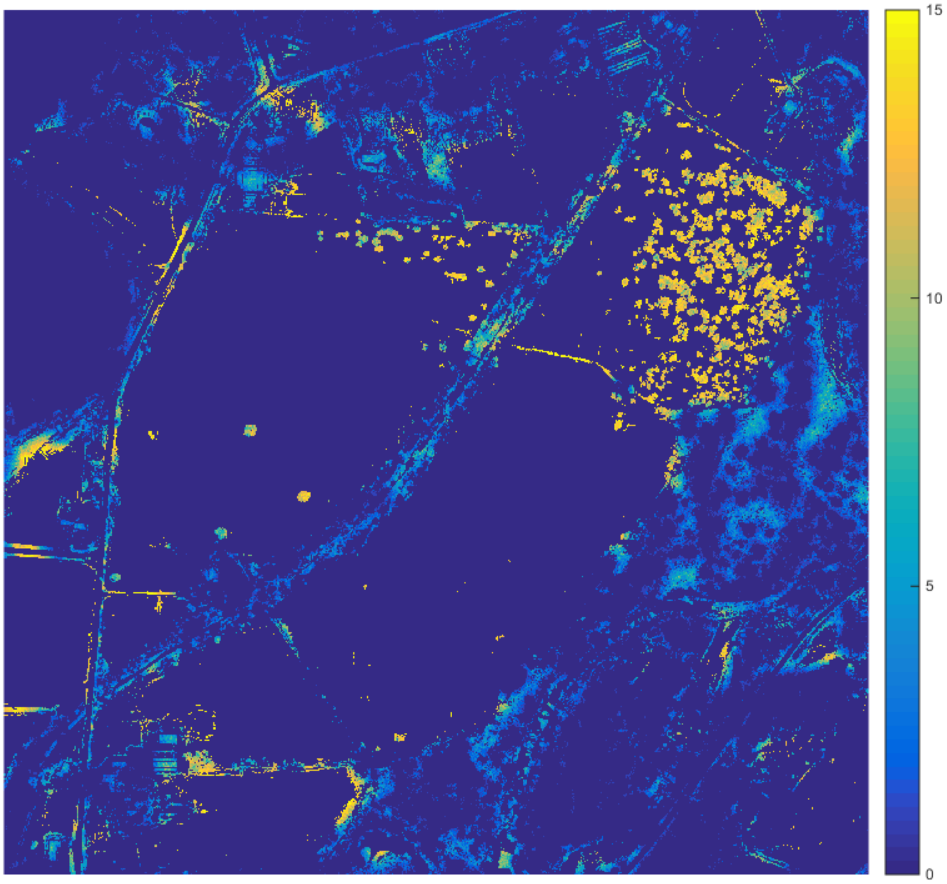


Figure 4.14: Results of the algorithm on the Mohawk dirt tracks scene.

Again, we are able to detect the roads. The main source of false positives for strong edges are the objects that appear to be bushes or other small plants. Some textured areas between trees are also incorrectly detected. Texture analysis techniques such as those found in [56] may be helpful here.

4.8.3 Gainesville Track

Our final scene is the Gainesville track scene, which contains a single dirt road that acts as a weaker edge than those contained in the examples above. As a result, we focus on the second scale of the shearlet coefficients. Figs. 4.15 and 4.16 show the original LIDAR and the logstd of the image.

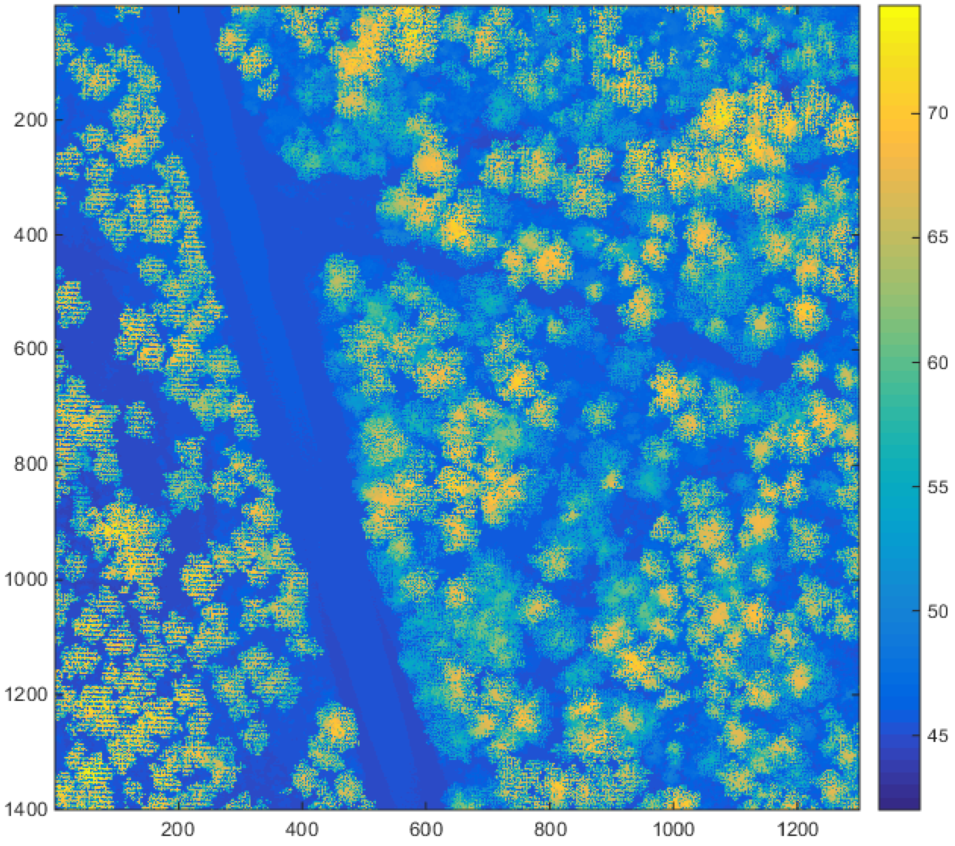


Figure 4.15: Original LIDAR image for the Gainsville track scene. The resolution is 0.1 m.

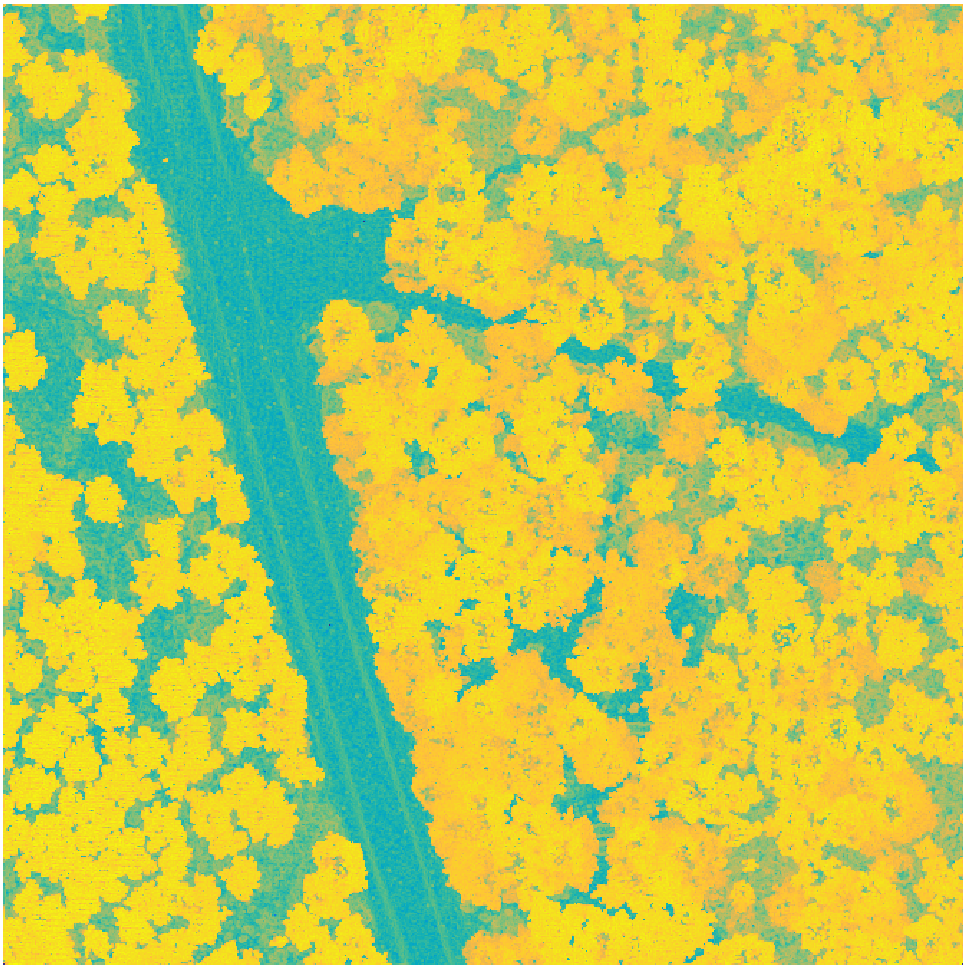


Figure 4.16: Logstd of the Gainsville track scene. Note the barely detectable dirt road.

We apply the algorithm with $loglow = -4, loghi = 0, shearhi = 0.1, elevhi = 5$, and $mfilt = [5, 5]$. The results are in Fig. 4.17

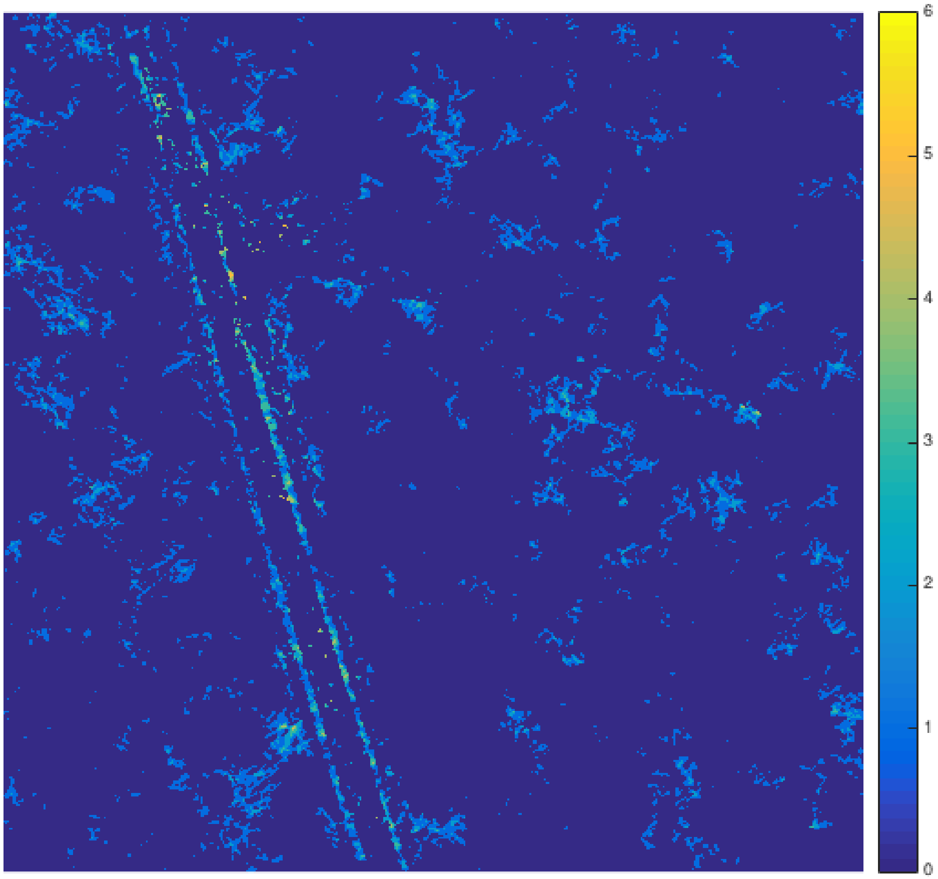


Figure 4.17: Results of the algorithm on the Gainsville track scene. Yellow pixels correspond to having 6 out of 8 small shearlet coefficients.

Unlike in the previous two examples, there are no pixels in Fig. 4.17 that have the maximum number of small shearlet coefficients, in this case seven. For such hard to detect roads, we pick up the edges fairly well. As in the previous example, areas between trees are false positives.

4.9 Conclusions and Future Directions

An aspect of the problem that we are not using is an *a priori* knowledge of the global shape of our objects of interest. Clearly, the road pixels differ from the non-road pixels, which show up as amorphous regions. Image segmentation by shape has been studied [57–59] and we could attempt to use some of their methods. This could be challenging because the shapes we extract have many disconnections.

One of the simplest ways to detect global shape is the Hough transform, which automatically detect lines in an image [60]. It is usually applied to the results of an edge detector and counts the number of nonzero pixels on several lines, parametrized by

$$\rho = x\cos(\theta) + y\sin(\theta).$$

The Hough transform is closely related to the Radon transform, which is defined for $f \in S(\mathbb{R}^2)$ by

$$R(f)(u,s) := \int_{\langle u,x\rangle = s} f(x)dx, u \in S^1, s \in \mathbb{R}.$$

We convert Fig. 4.17 into a binary image by setting all points with one or more counts to 1. This image is then fed into Matlab's built-in Hough transform functions. The result of the transform is in Fig. 4.18.

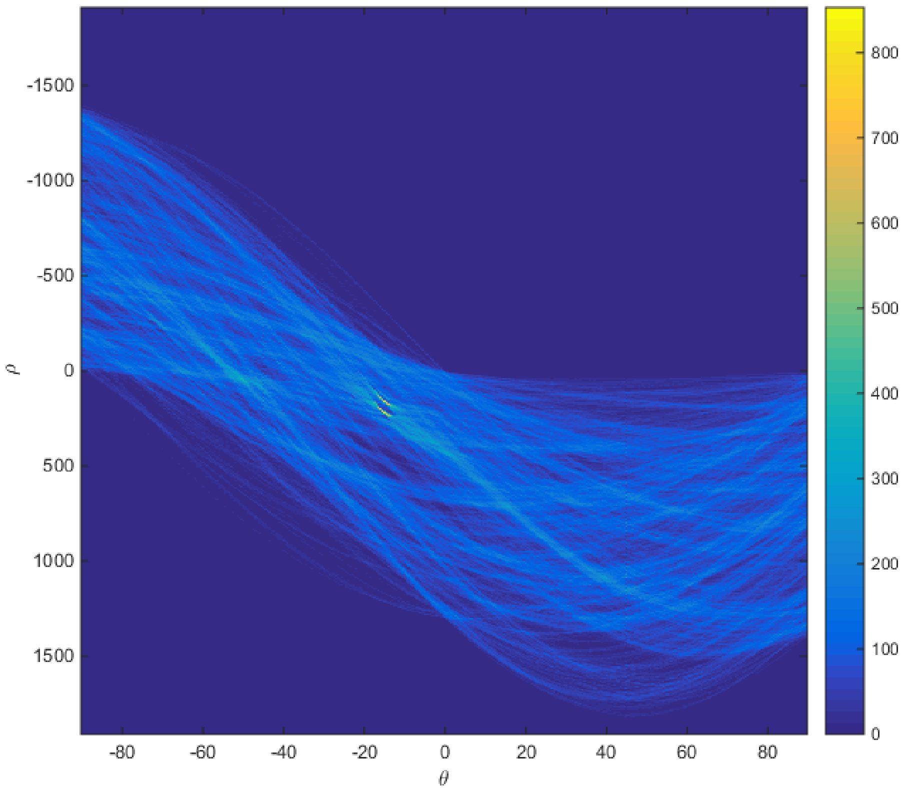


Figure 4.18: The number of pixels lying on lines determined by θ and ρ.

Plotting the ten most significant lines detected by the transform results in Fig. 4.19. These lines in fact only correspond to three distinct lines, that is, having nonconsecutive parameters. Averaging the parameters for each distinct line, we find that the lines are described by $(\theta, \rho) = (-15.5, 200), (-15, 209.5)$, and $(-14, 170)$. The first two of these lines correspond to the right road, while the third corresponds to the left road. Thus, global shape methods show promise in allowing us to remove all false positives from our results.

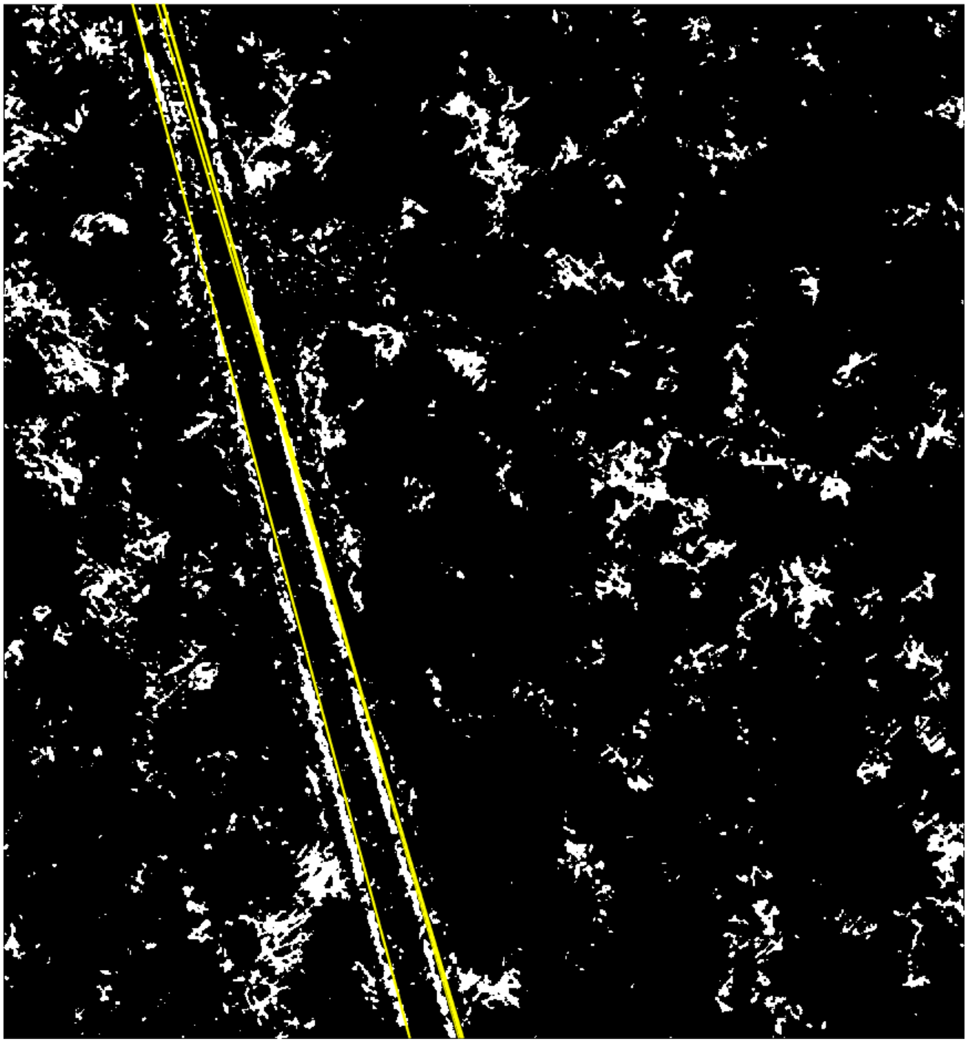


Figure 4.19: Plot of the ten most significant lines as determined by the Hough transform, superimposed on the binary-converted results of the algorithm. We observe that the roads have been detected despite the breaks in the road.

Chapter 5: Directional Superresolution

5.1 Introduction

Superresolution (SR) is the problem of improving the resolution of an image, without introducing artifacts. All sensors have a diffraction limit, which restricts resolving power [61]. Decreasing pixel size improves resolution, but has the drawback of also decreasing light, leading to shot noise [62]. Undersampling leads to aliasing, causing the image to appear blocky. We would like to be able to undo these distortions.

Superresolution is often phrased as an image recovery problem. Let f be the observed image and u the image we want to recover. One possible model for single frame SR is [63]

$$f = D(h * u) + n$$

where h is a possibly unknown blur filter, D is a down-sampling operator (which typically introduces aliasing), and n is Gaussian white noise. If we have several frames, we can use the analogous model

$$f_k = D_k(h * u) + n_k \qquad k = 1, \cdots, N$$

where the downsamplers D_k and noise n_k are allowed to vary. This problem is illposed due to the loss of high frequency information caused by the downsampling. The problem becomes more tractable with additional information. One common technique is to combine multiple low resolution images of the same scene with subpixel shifts [62]. Another is to build a dictionary of known high/low resolution pairs from a set of test images [64]. For this application, we will restrict ourselves to the simpler model

$$f = Du$$

and assume D downsamples by a factor of 2.

Since we do not assume any information beyond the downsampled image, we are attempting what is known as *single-image superresolution*. Simple methods of single-image SR reduce to image interpolation. Some popular interpolation techniques are nearest-neighbor, bilinear, and bicubic [65]. They use weighted averages of the values of 1, 4, and 16 nearest neighbors, respectively, to interpolate new values. All of these techniques lead to jagged edges where aliasing has occurred. More advanced methods rely heavily on statistical techniques. The New Edge-Directed Interpolation (NEDI) scheme [66] uses estimates of local covariance at the low resolution to interpolate the higher resolution. Directional Filtering and Data Fusion (DFDF) [67] fuses two estimates of a pixel's value through linear minimum mean square-error estimation. Soft Decision Adaptive Interpolation (SAI) [68] uses a 2D piecewise autoregressive model, where the model parameters are determined by a soft-decision estimation on groups of pixels. Finally, Kernel Regression (KR) [69] makes use of non-parametric estimation to denoise and interpolate randomly sampled data.

Harmonic analysis plays an important roll in several single-image SR techniques. A few authors have developed iterative procedures to impose sparsity in various transform domains using contourlets [70] and shearlets [36]. The Sparse Mixing Estimator (SME) approach of Mallat and Yu [71] uses a local mixture decomposition on the low resolution signal y

$$y = \sum_{B \in \mathcal{B}} \tilde{a}(B) y_B + y_r$$

where the sum is over small blocks with varying orientations, y_B is the signal restrict to block B, the $\tilde{a}(B)$ are mixing coefficients computed by orthogonal block matching pursuit on the fine scale wavelet coefficients, and y_r is a residue. Based on the size of $\tilde{a}(B)$, a decision is made as to where and how to directionally interpolate the image.

5.2 Superresolution by Tight Frames Constructed from Circulant Matrices

We now summarize the method of constructing tight frames from circulant matrices due to Bosch et al. [72] Below, bolded letters will always refer to column vectors.

Definition 5.2.1. A matrix is called *circulant* if it is in the image of the mapping

$C : \mathbb{R}^m \to \mathbb{R}^{m\times m}$ defined by

$$C(\mathbf{a}^T) = C([a_1, \cdots, a_m]) := \begin{bmatrix} a_1 & a_2 & \cdots & a_m \\ a_m & a_1 & \cdots & a_{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_2 & a_{m-1} & \cdots & a_1 \end{bmatrix}.$$

By a slight abuse of notation, we define C on a column vector $\mathbf{b}$ by $C(\mathbf{b}) = (C(\mathbf{b}^T))^T$.

Definition 5.2.2. Let $\mathbf{e}_2$ denote the standard basis vector for $\mathbb{R}^m$ whose only nonzero component is a 1 in the second slot. The *basic circulant permutation matrix* of order m is

$$V := C(\mathbf{e}_2^T) = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix}.$$

We list some important properties of circulant matrices here that are straightforward to show:

1. V generates a cyclic group of order m with $V^{-1} = V^T$. All elements of the group have trace 0 except the identity.

2. V^T permutes the entries of a column vector cyclicly, i.e,

   $$V^T\mathbf{b} = V^T[b_1, b_2, \ldots, b_m]^T = [b_m, b_1, b_2, \ldots, b_{m-1}]^T.$$

   In particular, the jth column of $C(\mathbf{b})$ is given by $(V^T)^{j-1}\mathbf{b}$.

3. $C(\mathbf{a}^T)C(\mathbf{b}) = C(\mathbf{a}^T C(\mathbf{b}))$.

4. $C(\mathbf{a_1}^T) + C(\mathbf{a_2}^T) = C(\mathbf{a_1}^T + \mathbf{a_2}^T)$.

5. If C_1 and C_2 are circulant, then $C_1C_2 = C_2C_1$, that is, they commute.

We can now prove the following theorem:

Theorem 5.2.3 (Theorem 3.1, [72])**.** *Let $(\mathbf{a}_i)_{i=1}^m$ be a basis for $\mathbb{R}^m$. Define* $A = \begin{bmatrix} \mathbf{a}_1^T \\ \vdots \\ \mathbf{a}_m^T \end{bmatrix} \in \mathbb{R}^{m\times m}$*. Denote its inverse by* $B = [\mathbf{b}_1, \cdots, \mathbf{b}_m] \in \mathbb{R}^{m\times m}$*. Then,*

$$\sum_{i=1}^m C(\mathbf{a}_i^T)C(\mathbf{b}_i) = \sum_{i=1}^m C(\mathbf{b}_i)C(\mathbf{a}_i^T) = m \cdot Id.$$

Proof.

$$\sum_{i=1}^m C(\mathbf{a}_i^T)C(\mathbf{b}_i) = \sum_{i=1}^m C(\mathbf{a}_i^T C(\mathbf{b}_i)) = C\left(\sum_{i=1}^m \mathbf{a}_i^T C(\mathbf{b}_i)\right)$$

so it suffices to show that $\sum_{i=1}^m \mathbf{a}_i^T C(\mathbf{b}_i) = [m, 0, \cdots, 0]$.

$$\begin{aligned}\sum_{i=1}^m \mathbf{a}_i^T C(\mathbf{b}_i) &= \left[\sum_{i=1}^m \mathbf{a}_i^T \mathbf{b}_i, \sum_{i=1}^m \mathbf{a}_i^T V^T \mathbf{b}_i, \cdots, \sum_{i=1}^m \mathbf{a}_i^T (V^T)^{m-1}\mathbf{b}_i\right] \\ &= \left[Tr(AB), Tr(AV^TB), \cdots, Tr(A(V^T)^{m-1}B)\right] \\ &= \left[Tr(Id_m), Tr(V^T), \cdots, Tr((V^T)^{m-1})\right] \\ &= [m, 0, \cdots, 0]\end{aligned}$$

where we have used that the matrix trace satisfies $Tr(A_1A_2) = Tr(A_2A_1)$, whenever both matrix products make sense. □

We immediately have the following corollary:

Corollary 5.2.4. *With the previous hypotheses, the columns of the matrices* $(C(\mathbf{a}_i))_{i=1}^m$ *form a frame for* $\mathbb{R}^m$ *with dual frame given by the columns of the matrices* $(\frac{1}{m}C(\mathbf{b}_i))_{i=1}^m$. *In addition, the columns of the matrices* $(C(\mathbf{a}_i^T))_{i=1}^m$ *form a frame for* $\mathbb{R}^m$ *with dual frame given by the columns of the matrices* $(\frac{1}{m}C(\mathbf{b}_i^T))_{i=1}^m$.

An important special case is if the $(\mathbf{a}_i)_{i=1}^m$ form an orthonormal basis for $\mathbb{R}^m$.

Corollary 5.2.5. *Assume that the vectors* $(\mathbf{a}_i)_{i=1}^m$ *form an orthonormal basis for* $\mathbb{R}^m$. *Then, the columns of the matrix* $[C(\mathbf{a}_1), C(\mathbf{a}_2), \ldots, C(\mathbf{a}_m)]$ *form an* m*-tight frame for* $\mathbb{R}^m$.

Proof. Because A is orthogonal, $A^{-1} = A^T$, so $\mathbf{b}_i = \mathbf{a}_i$ in Corollary 5.2.4. Therefore, the columns of the matrices $(C(\mathbf{a}_i))_{i=1}^m$ form a frame for $\mathbb{R}^m$ with dual frame given by the columns of the matrices $(\frac{1}{m}C(\mathbf{a}_i))_{i=1}^m$, i.e., the columns of the matrices $(C(\mathbf{a}_i))_{i=1}^m$ form an m-tight frame for $\mathbb{R}^n$. □

Note that this frame has synthesis operator $\Phi = [C(\mathbf{a}_1), C(\mathbf{a}_2), \ldots, C(\mathbf{a}_m)]$ and analysis operator $\Phi^* = [C(\mathbf{a}_1), C(\mathbf{a}_2), \ldots, C(\mathbf{a}_m)]^T$, which satisfy $\Phi\Phi^* = m \cdot Id$. For image analysis, we would like a $2D$ frame; this is easily accomplished by using the Kronecker product. Recall that the Kronecker product $\otimes$ has the *mixed-product property*: $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$.

Theorem 5.2.6 (Theorem 4.1, [72]). *Let* Φ *be the analysis operator for an* m*-tight frame of* $\mathbb{R}^n$ *constructed as in Corollary 5.2.5. Then the columns of* $\Phi_{2D} := \Phi \otimes \Phi$ *form an* m^2*-tight frame for* $\mathbb{R}^{n^2}$.

Proof.

$$
\begin{aligned}
\Phi_{2D}\Phi_{2D}^{*} &= (\Phi \otimes \Phi)(\Phi \otimes \Phi)^{*} \\
&= (\Phi \otimes \Phi)(\Phi^{*} \otimes \Phi^{*}) \\
&= (\Phi\Phi^{*}) \otimes (\Phi\Phi^{*}) \\
&= (m \cdot Id) \otimes (m \cdot Id) \\
&= m^{2} \cdot Id.
\end{aligned}
$$

□

This provides us a $2D$ tight frame if we identify $\mathbb{R}^{n^2}$ with $\mathbb{R}^{n \times n}$.

Good choices for the orthonormal vectors that will generate our $2D$ frame are

$$
\begin{aligned}
\mathbf{l} &= [1, 1, 1]^{T}/\sqrt{3} \\
\mathbf{b} &= [1, 0, -1]^{T}/\sqrt{2} \\
\mathbf{h} &= [1, -2, 1]^{T}/\sqrt{6},
\end{aligned}
$$

which represent low-, band-, and high-pass filters, respectively. The construction produces a 9-tight frame for $\mathbb{R}^{3\times 3}$ with analysis operator

$$\Phi_{2D}^{*} = [C(\mathbf{l}), C(\mathbf{b}), C(\mathbf{h})]^{T} \otimes [C(\mathbf{l}), C(\mathbf{b}), C(\mathbf{h})]^{T} \in \mathbb{R}^{81\times 9}.$$

Applying Φ_{2D}^{*} to $Z \in \mathbb{R}^{3\times 3}$, we obtain 9 blocks of size 3×3:

$$
\begin{bmatrix}
C(\mathbf{l})^{T} Z C(\mathbf{l}) & C(\mathbf{l})^{T} Z C(\mathbf{b}) & C(\mathbf{l})^{T} Z C(\mathbf{h}) \\
C(\mathbf{b})^{T} Z C(\mathbf{l}) & C(\mathbf{b})^{T} Z C(\mathbf{b}) & C(\mathbf{b})^{T} Z C(\mathbf{h}) \\
C(\mathbf{h})^{T} Z C(\mathbf{l}) & C(\mathbf{h})^{T} Z C(\mathbf{b}) & C(\mathbf{h})^{T} Z C(\mathbf{h})
\end{bmatrix}.
$$

If Z is a 3×3 block from an image, we only get information about horizontal, vertical, and diagonal edges at the center of the block. For instance, the $C(\mathbf{l})^T ZC(\mathbf{h})$ block will be small in the presence of a vertical edge, while the $C(\mathbf{h})^T ZC(\mathbf{l})$ block will be small in the presence of a horizontal edge. To process a more general direction given by a vector $\mathbf{t} = [t_y, t_x]^T$, we define a "grabbing" function $G_t : \mathbb{R}^{n_1\times n_1} \to \mathbb{R}^{3\times 3}$ by $G_t(\tilde{Z}) = Z$ where $Z_{j,k} = \tilde{Z}_{y(j,k),x(j,k)}$ with

$$y(j,k) := 1 + (j-1)t_y + (k-1)t_x$$

$$x(j,k) := 1 + (k-1)t_y + (n-j)t_x$$

$$n_1 := 1 + (n-1)(t_y + t_x),$$

for $j, k = 1, 2, 3$. For example, in the case of a 45 degree angle, $t_y = t_x = 1$ and $n_1 = 5$, so

$$G_t\left(\begin{bmatrix} z_{11} & z_{12} & z_{13} & z_{14} & z_{15} \\ z_{21} & z_{22} & z_{23} & z_{24} & z_{25} \\ z_{31} & z_{32} & z_{33} & z_{34} & z_{35} \\ z_{41} & z_{42} & z_{43} & z_{44} & z_{45} \\ z_{51} & z_{52} & z_{53} & z_{54} & z_{55} \end{bmatrix}\right) = \begin{bmatrix} z_{13} & z_{24} & z_{35} \\ z_{22} & z_{33} & z_{44} \\ z_{31} & z_{42} & z_{53} \end{bmatrix}.$$

By finding the minimizer of the l^2 energy of the $C(\mathbf{l})^T ZC(\mathbf{h})$ block over all directions, we can assign a local direction. If the direction found is considered significant and the region is not of low variance, nearest-neighbor upsampling is applied with a motion blur in the dominant direction. Otherwise, the algorithm defaults to bicubic interpolation. The resulting superresolved images have much smoother edges as compared to bicubic alone.

5.3 A Shearlet-based Approach to Superresolution

In this section, we develop an algorithm for superresolution utilizing shearlets to detect position and orientation of edges in order to perform edge smoothing. Local direction at a particular pixel can be estimated by finding the shearing parameter which maximizes the magnitude of the shearlet coefficients at a particular scale. This is an important fact, which we use in our algorithms both in this chapter and Chapter 4. While intuitively clear, we prove it rigorously for the simple case of characteristic functions on half-planes. We begin with a few lemmas:

Lemma 5.3.1. *Let $\psi \in S(\mathbb{R}^2), r \in \mathbb{R}$. Then*

$$\int_{\mathbb{R}} \hat{\psi}(-r\omega, \omega) d\omega = \int_{\mathbb{R}} \psi(x, rx) dx.$$

Proof. Define $f(y) = \int_{\mathbb{R}} \psi(x, y + rx) dx$.

$$\begin{aligned}
\int_{\mathbb{R}} \hat{\psi}(-r\omega, \omega) d\omega &= \int_{\mathbb{R}} \int_{\mathbb{R}^2} \psi(x, y) e^{-2\pi i(-rx+y)\omega} dx dy d\omega \\
&= \int_{\mathbb{R}} \int_{\mathbb{R}} \int_{\mathbb{R}} \psi(x, y' + rx) e^{-2\pi i y'\omega} dy' dx d\omega \\
&= \int_{\mathbb{R}} \int_{\mathbb{R}} f(y) e^{-2\pi i y\omega} dy d\omega \\
&= \int_{\mathbb{R}} \hat{f}(\omega) d\omega \\
&= f(0) \\
&= \int_{\mathbb{R}} \psi(x, rx) dx.
\end{aligned}$$

□

Let $H_{y>rx}$ denote the characteristic function of the set $\{(x, y) \in \mathbb{R}^2 | y > rx\}$ for some fixed $r \in \mathbb{R}$.

Lemma 5.3.2.

$$\frac{\partial}{\partial y} H_{y>rx} = \delta_{y-rx}$$

in the sense of distributions.

Proof. Let $\psi \in C_c^\infty(\mathbb{R}^2)$.

$$\begin{aligned}
\langle \frac{\partial}{\partial y} H_{y>rx}, \psi \rangle &= -\langle H_{y>rx}, \psi_y \rangle \\
&= -\int_{\mathbb{R}^2} H_{y>rx} \psi_y dxdy \\
&= -\int_{-\infty}^{\infty} \int_{rx}^{\infty} \psi_y dydx \\
&= \int_{\mathbb{R}} \psi(x, rx) dx \\
&= \langle \delta_{y-rx}, \psi \rangle.
\end{aligned}$$

□

Lemma 5.3.3.

$$\widehat{H_{y>rx}} = \frac{1}{2\pi i \omega_2} \widehat{\delta_{y-rx}}$$

where $\widehat{\delta_{y-rx}}$ acts by

$$\langle \widehat{\delta_{y-rx}}, \hat{\psi} \rangle = \int_{\mathbb{R}} \hat{\psi}(-r\omega, \omega) d\omega, \qquad \hat{\psi} \in C_c^\infty(\mathbb{R}^2).$$

Proof. Note that $\widehat{\frac{\partial}{\partial y} H_{y>rx}} = (2\pi i \omega_2) \widehat{H_{y>rx}} = \widehat{\delta_{y-rx}}$ so the first claim follows. For the second claim, we use Parseval and the above lemma:

$$\begin{aligned}
\langle \widehat{\delta_{y-rx}}, \hat{\psi} \rangle &= \langle \delta_{y-rx}, \psi \rangle \\
&= \int_{\mathbb{R}} \psi(x, rx) dx \\
&= \int_{\mathbb{R}} \hat{\psi}(-r\omega, \omega) d\omega.
\end{aligned}$$

□

We can now prove that shearlets detect the direction of discontinuities in half-planes.

Theorem 5.3.4 (DW)**.** *Discretize the position, dilation, and shear parameters and define ψ as in the FFST. Assume WLOG that $|r| \leqslant 1$. Let $f(x, y) = H_{y>rx}$ and fix a scale j and position m. For almost all values of r, the shearlet transform of $f, \mathcal{SH}(f)(j, k, m)$, is only non-zero for two consecutive values of the shearing parameter k. These values satisfy*

$$|s_{j,k} - r| < \frac{1}{2^j}.$$

Furthermore, for the unique k that maximizes $|\mathcal{SH}(f)(j, k, m)|, s_{j,k}$ is closest to r over all k.

Proof.

$$\begin{aligned}
\mathcal{SH}(f)(j, k, m) &= \langle f, \psi_{jkm} \rangle \\
&= \langle \hat{f}, \hat{\psi}_{jkm} \rangle \\
&= \int_{\mathbb{R}^2} \frac{1}{2\pi i \omega_2} \widehat{\delta_{y-rx}}(\omega_1, \omega_2) \hat{\psi}_{jkm}(\omega_1, \omega_2) d\omega_1 d\omega_2 \\
&= \frac{1}{2\pi i} \int_{\mathbb{R}} \frac{1}{\omega_2} \hat{\psi}_{jkm}(-r\omega_2, \omega_2) d\omega_2.
\end{aligned}$$

By the assumption on r, we are utilizing the vertical cone, hence

$$\hat{\psi}_{jkm}(-r\omega_2, \omega_2) = \hat{\psi}_1(4^{-j}\omega_2)\hat{\psi}_2(-2^j r + k) \exp(-2\pi i(-r\omega_2 m_1/M + \omega_2 m_2/N)).$$

We note that k only occurs in $\hat{\psi}_2(-2^j r + k)$, so maximizing this term will maximize $|\mathcal{SH}(f)(j, k, m)|$. The function $\hat{\psi}_2$ is a non-negative, smooth function supported on $[-1,1]$ that is strictly increasing on $[-1,0]$ and strictly decreasing on $[0,1]$. The support assumption implies that the term is 0 unless $-1 < -2^j r + k < 1$, or equivalently, $|s_{j,k} - r| < \frac{1}{2^j}$. Since the shearlets' slopes differ by $1/2^j$, this occurs for two consecutive values of k, except in the case when r and a shearlet's slope are exactly equal. Here, only one value of k leads to a non-zero shearlet coefficient. Furthermore, the above term will be maximized when $-2^j r + k$ is closest to 0, or when r is closest to $\frac{k}{2^j} = s_{j,k}$, the slope of the shearlet's center. There is a unique such k, unless r is equidistant between two shearlet directions. Here, two values of k give identical shearlet coefficients. This completes the proof when $|r| \leqslant 1$. A symmetric argument, using the definition of ψ in the horizontal cone, shows the result for $|r| > 1$. □

5.4 Description of Algorithm

Our algorithm is described below. Theorem 5.3.4 plays a crucial roll here since it provides guarantees on how well we can resolve the slope of directional features. The input image I may be grayscale or RGB. In the latter case, we perform the algorithm on each slice and reconstruct the color image at the end. Hence, we may assume that I is an $M \times N$ matrix. We scale its entries so that they lie in $[0,1]$.

1. Apply the Fast Finite Shearlet Transform [12], [28] to I. This produces shearlet coefficients up to $\left\lfloor \frac{1}{2}\log_2(\max\{M,N\}) \right\rfloor$ scales. If we label the scales from coarsest to finest scale starting at $j = 1$, we have 2^{j+1} matrices of size $M \times N$ at the jth scale, each corresponding to a different direction approximately

equally spaced. We fix a scale j that best captures the features of interest. In all of our test images, we found that $j = 3$ was optimal, giving us 16 directions. Denote these directional matrices $D_1, ..., D_{2^{j+1}}$.

2. Upsample I by a factor of $n \in \mathbb{N}$ using bicubic interpolation to acquire $\tilde{I}$.

3. Upsample each of the directional matrices by a factor of $n \in \mathbb{N}$ using bicubic interpolation to acquire $\tilde{D}_1, ..., \tilde{D}_{2^{j+1}}$.

4. Assign each pixel in $\tilde{I}$ a local direction based on which directional matrix contains the shearlet coefficient of largest magnitude. Pixels are assigned no dominant direction as determined by one of the following three methods:

   (a) The maximum coefficient is below a certain threshold. We found empirically that a threshold of 0.04 works well for all of our test images, but this parameter can be tuned.

   (b) The local standard deviation in a 5×5 neighborhood is below a certain threshold. We chose a threshold of 0.06.

   (c) It is not picked as an edge by a 'thickened' edge detector. In more detail, we apply the Canny edge detector with low/high thresholds 0.1/0.3 and default σ to $\tilde{I}$, then thicken the edges using the MATLAB function 'imdilate' with a dilation neighborhood of ones(3). Any pixel set to 0 in this process is considered to have no local direction.

   For all three methods, pixels close to the image boundary (say within 10 pixels) are also assigned no direction since the FFST gives spurious directional information near the boundary.

5. Apply a motion blur filter of length $\lfloor 2.5n \rfloor$ in each of the 2^{j+1} directions $\tilde{I}$, to produce $\tilde{I}_1, ..., \tilde{I}_{2^{j+1}}$.

6. Replace the pixel values of $\tilde{I}$ by their corresponding blurred version based on the previously assigned local direction.

7. Output the superresolved image.

5.5 Experiments and Results

For our test image, we chose a 512×512 section of a color orthophoto with 5 cm spatial resolution of an area in Zeebruges, Belgium (Fig. 5.1). The data was provided for the 2015 IEEE GRSS Data Fusion Contest. [1] We downsampled the original by a factor of 2 by removing every odd-indexed row and column, thus obtaining a low-quality/aliased image which we seek to superresolve. Fig. 5.2 shows the result of upsampling through bicubic interpolation alone. Note the jaggedness of the edges. We then performed our algorithm to the image. Fig. 5.3 shows the full direction map for the green channel, i.e., the direction of largest shearlet magnitude for each pixel in the second channel. The other channels have similar direction maps, so we arbitrarily choose the green channel for brevity. Directions vary from 1 (vertical) to 16 (line of slope 4). Recall that we have three different methods for determining which pixels have a locally dominant direction.

[1] http://www.grss-ieee.org/community/technical-committees/data-fusion

We display the direction maps, where the darkest blue represents no local direction, as well as the resulting superresolved images. We also processed the image with Mallat's SME method for comparison, see Fig. 5.10. Fig. 5.11 shows a 100×100 subset comparing bicubic, SME, and our algorithm using method 1. We see that bicubic convolution leads to many jagged edges, which both SME and our algorithm fix. On one hand, SME gives sharper edges than our algorithm, which is expected since we use blurring on the edges. On the other hand, SME caused some artifacts that appear as lines on the roof. Additionally, Mallat's algorithm is much slower than ours: running on a MacBook Pro with a 2.6 GHz Intel Core i5 processor and 16 GB of RAM, SME took 705 seconds, compared to 3 seconds for our methods. We could easily process more/larger images in a reasonable time, with the biggest bottleneck being the initial computation of the shearlet transform.

5.6 Figures

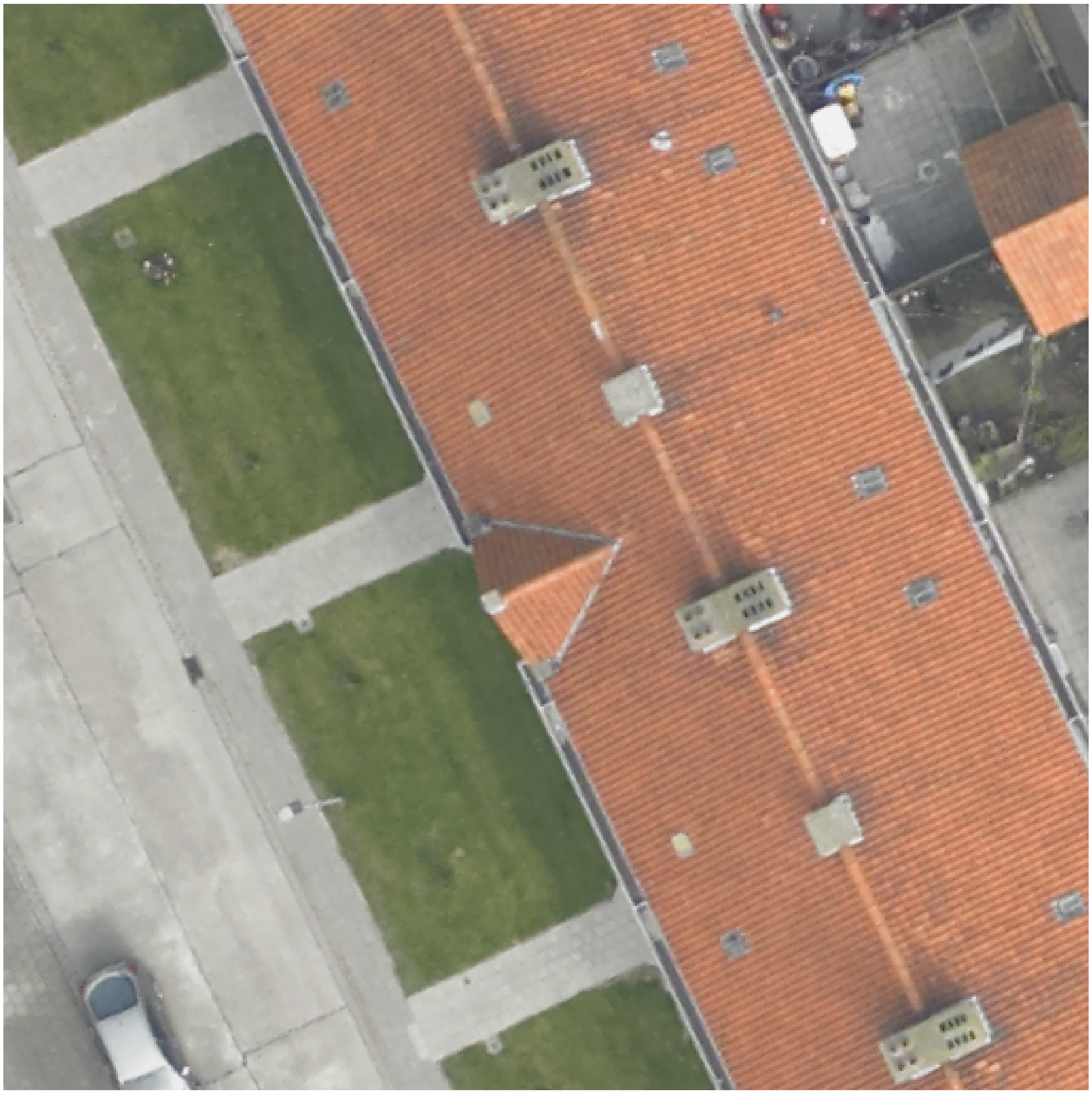


Figure 5.1: The original, high-resolution image of an aerial view of an area in Zeebruges, Belgium.

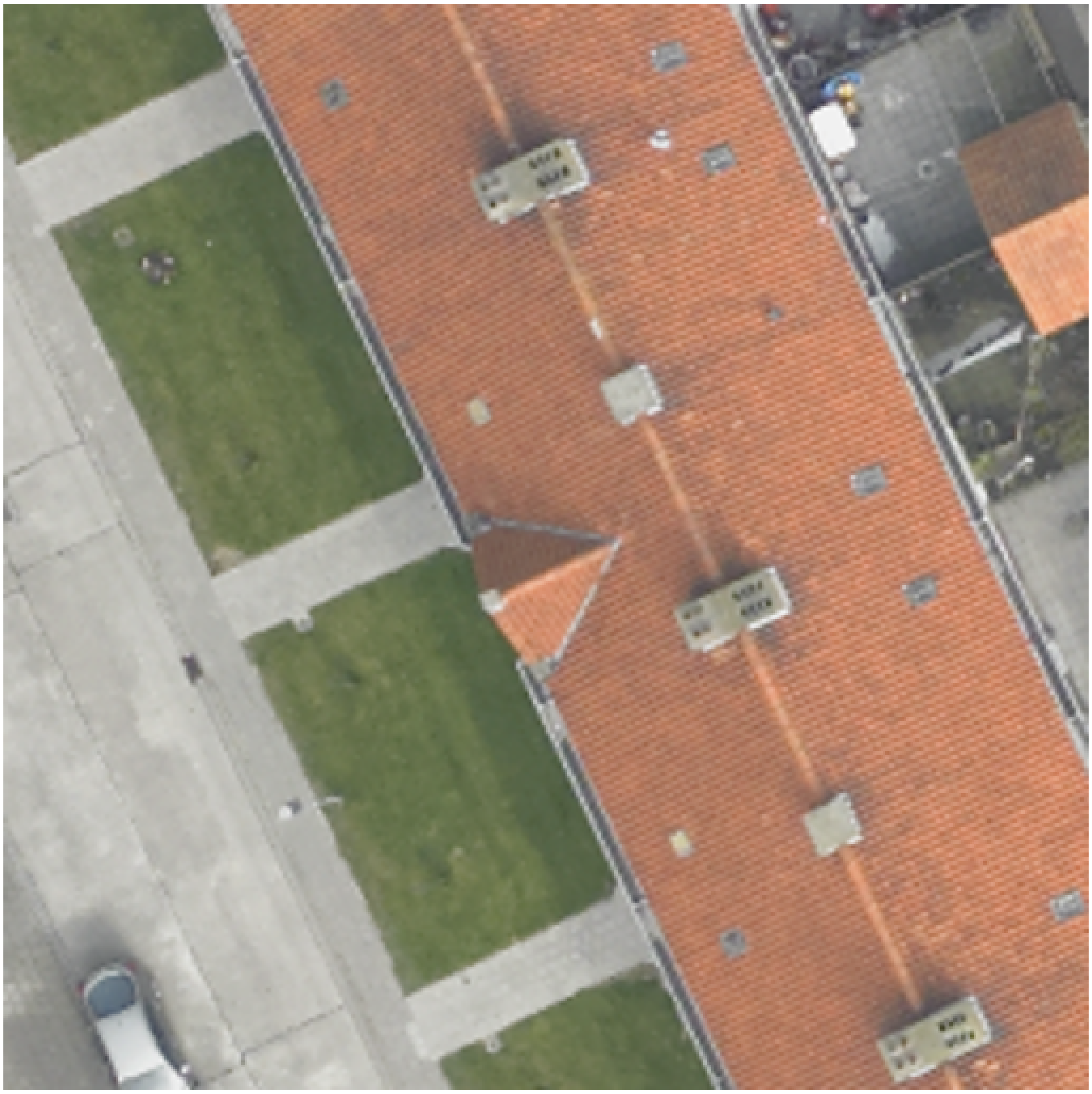


Figure 5.2: The image superresolved by bicubic convolution.

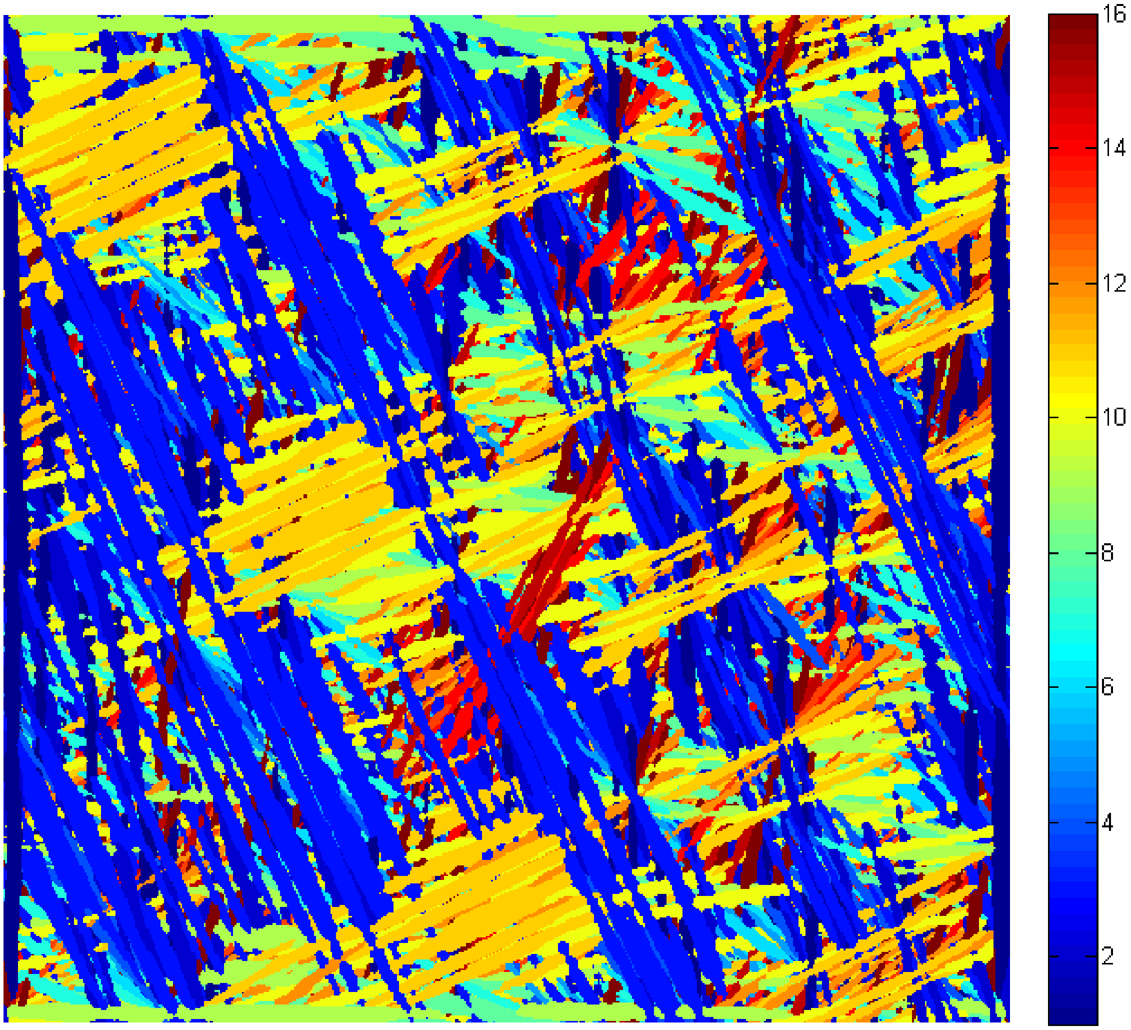


Figure 5.3: The full direction map as determined by the largest shearlet coefficient. The directions vary from 1 (vertical) to 16 (line of slope 4).

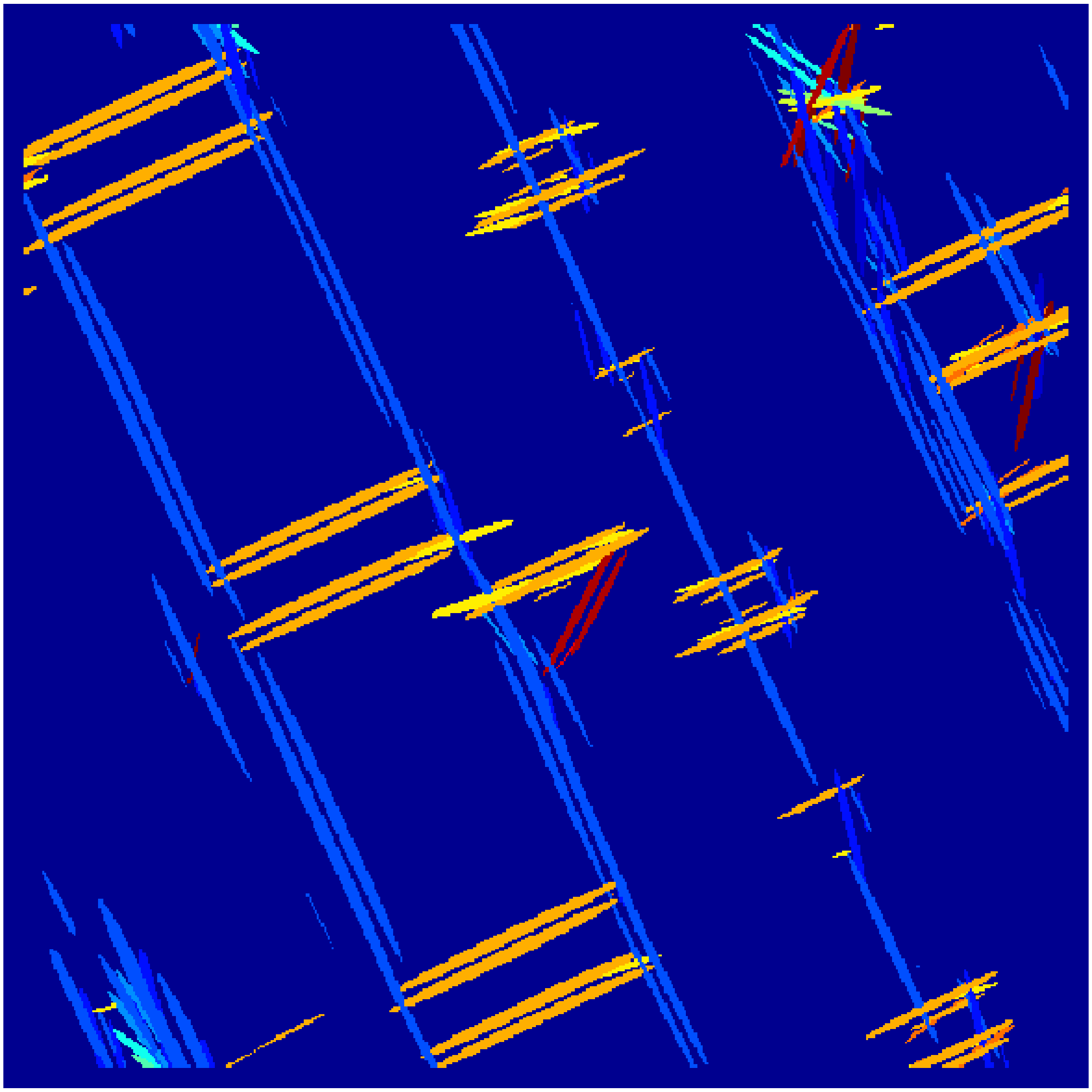


Figure 5.4: The direction map obtained using Method 1 to remove pixels if their largest shearlet coefficient is below 0.04.

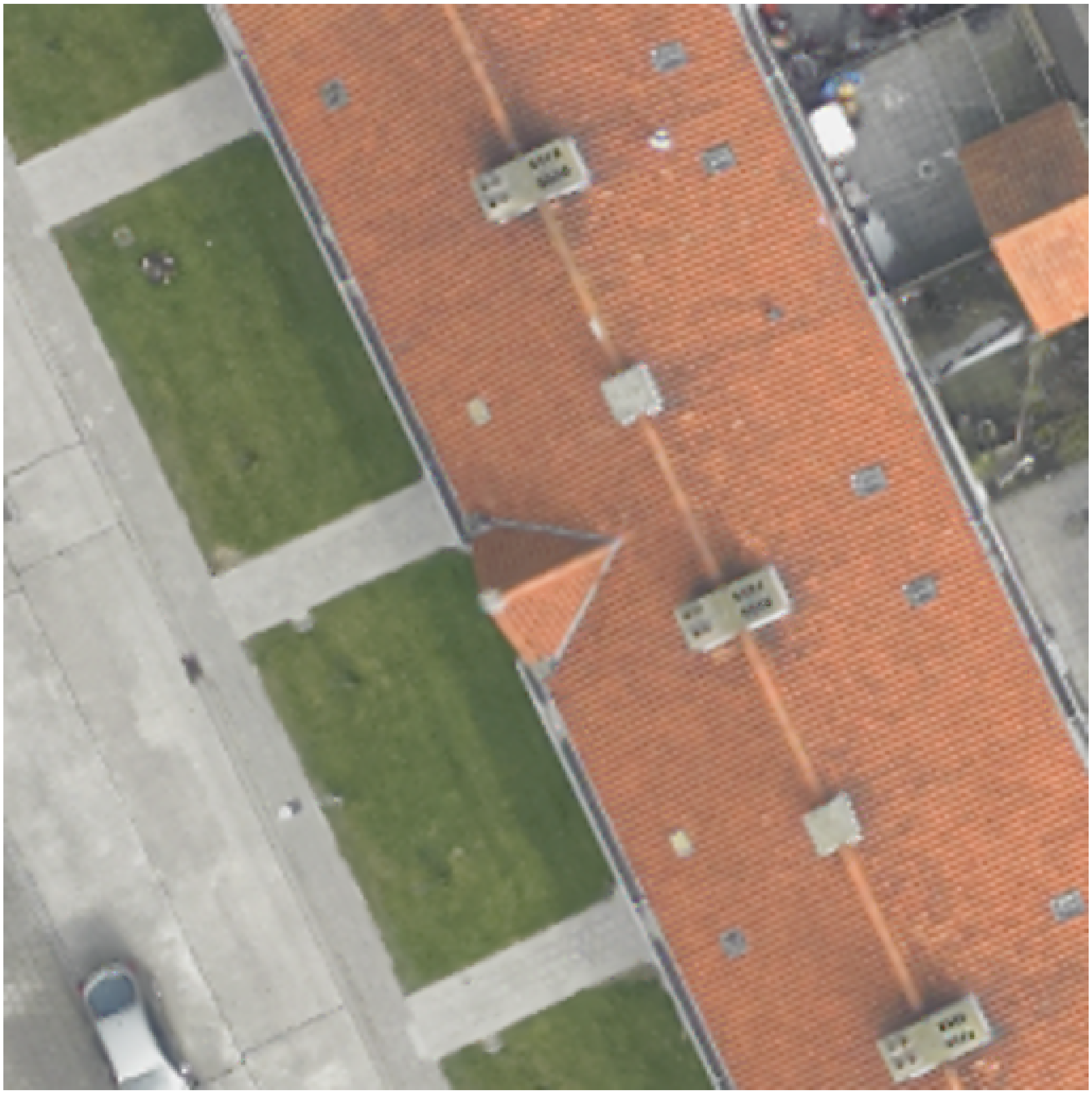


Figure 5.5: The superresolved image using Method 1.

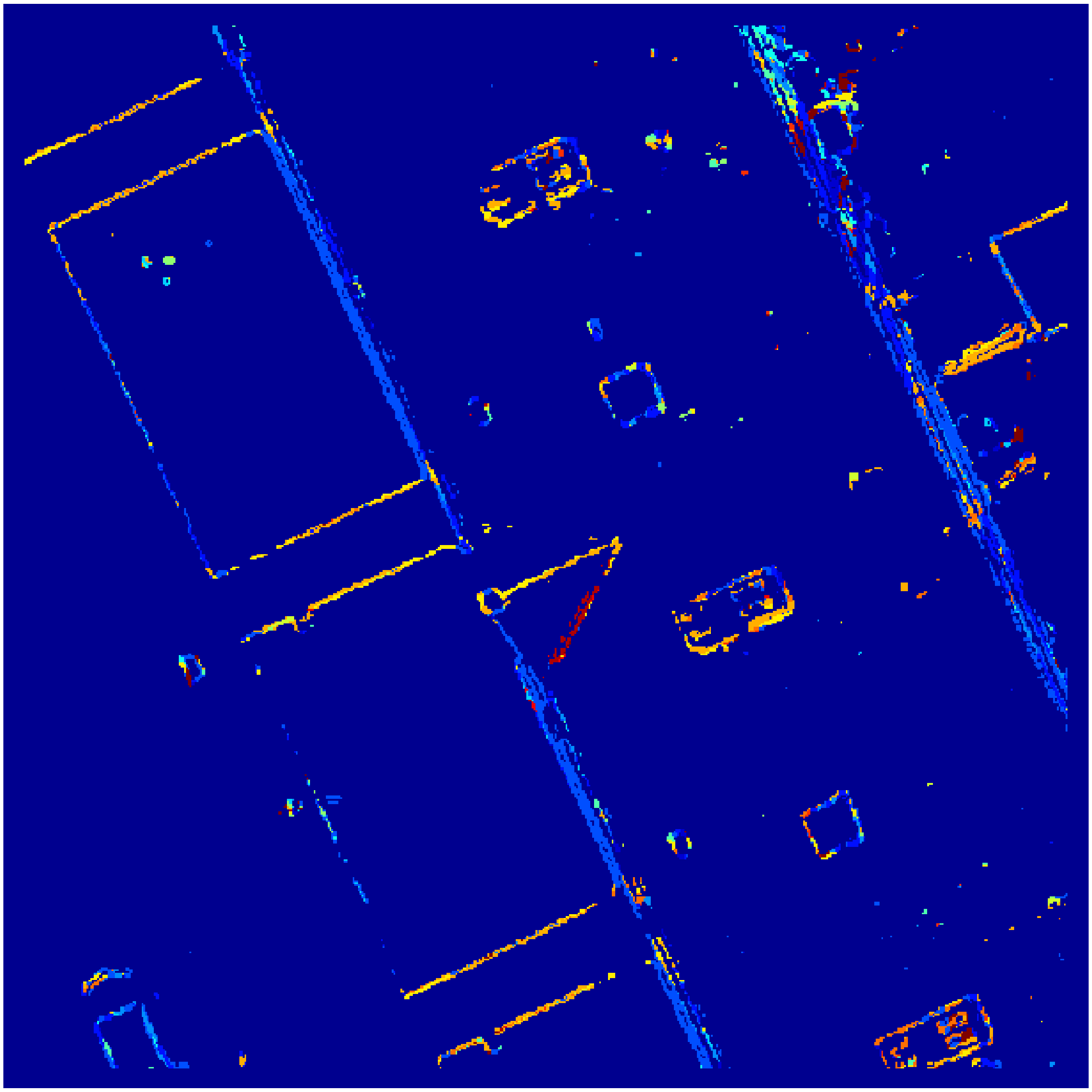


Figure 5.6: The direction map obtained using Method 2 to remove pixels if their local standard deviation in a 5 × 5 neighborhood is below 0.06.

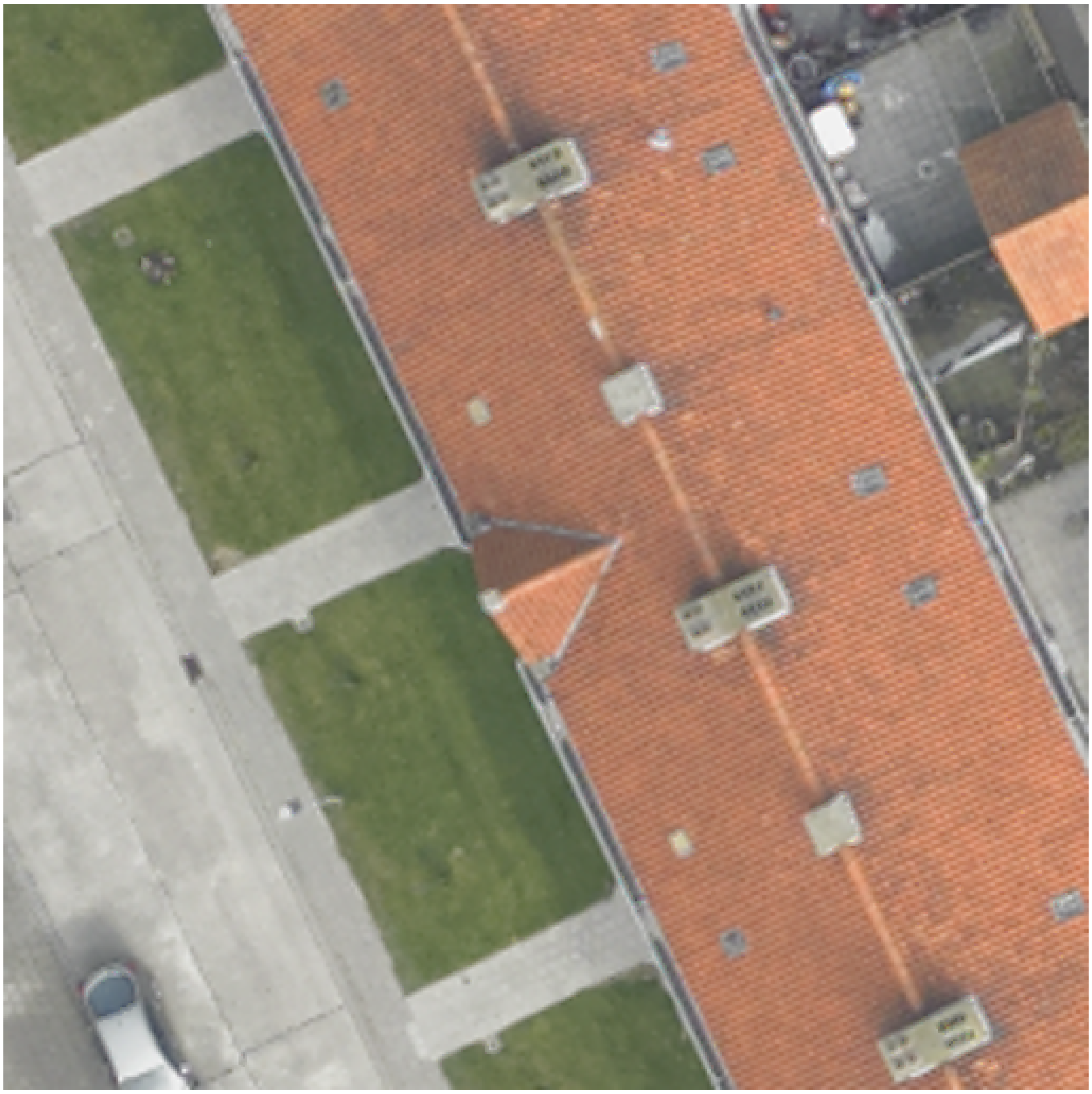


Figure 5.7: The superresolved image using Method 2.

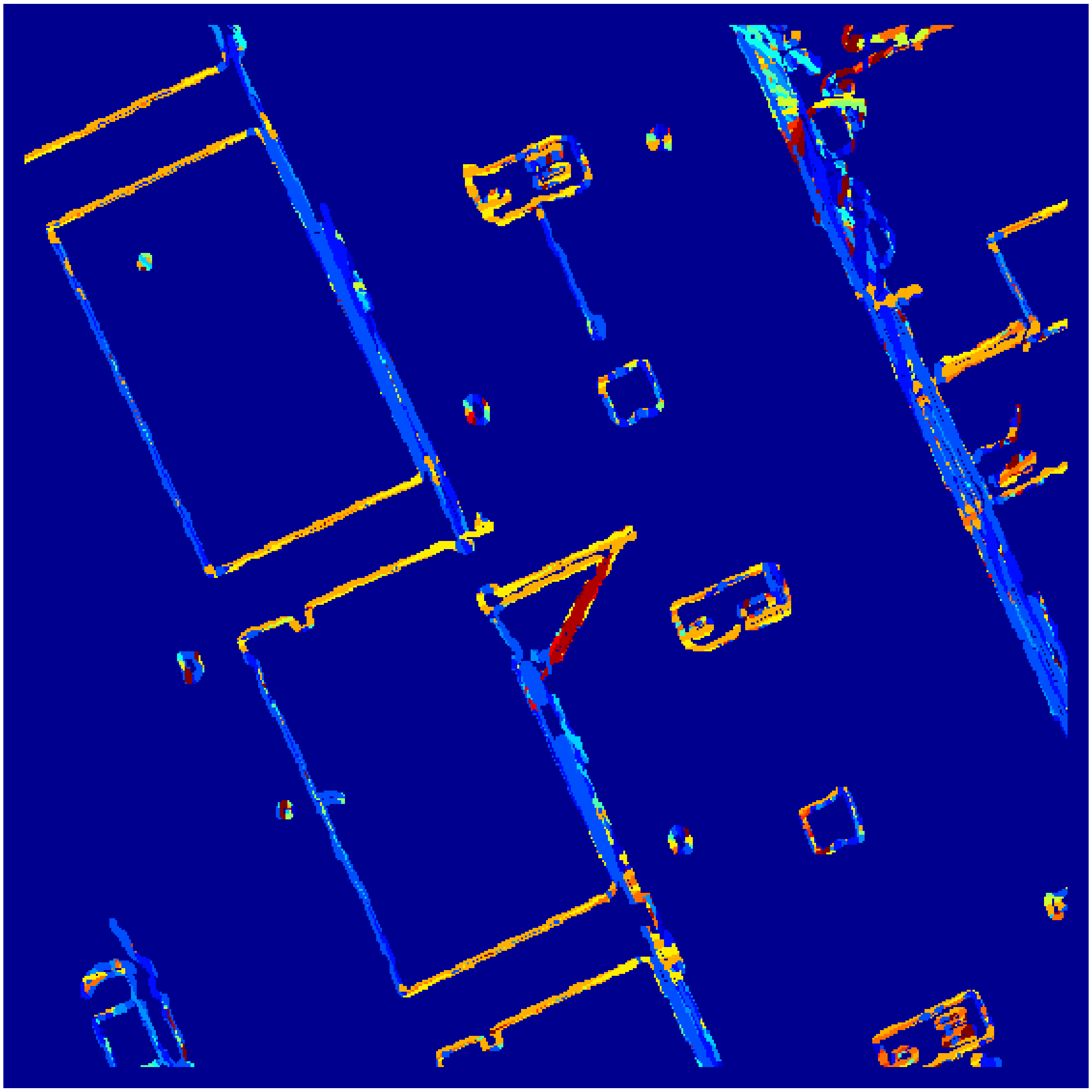


Figure 5.8: The direction map obtained using Method 3 to remove pixels if they are not in a thickened edge, as determined by the Canny algorithm.

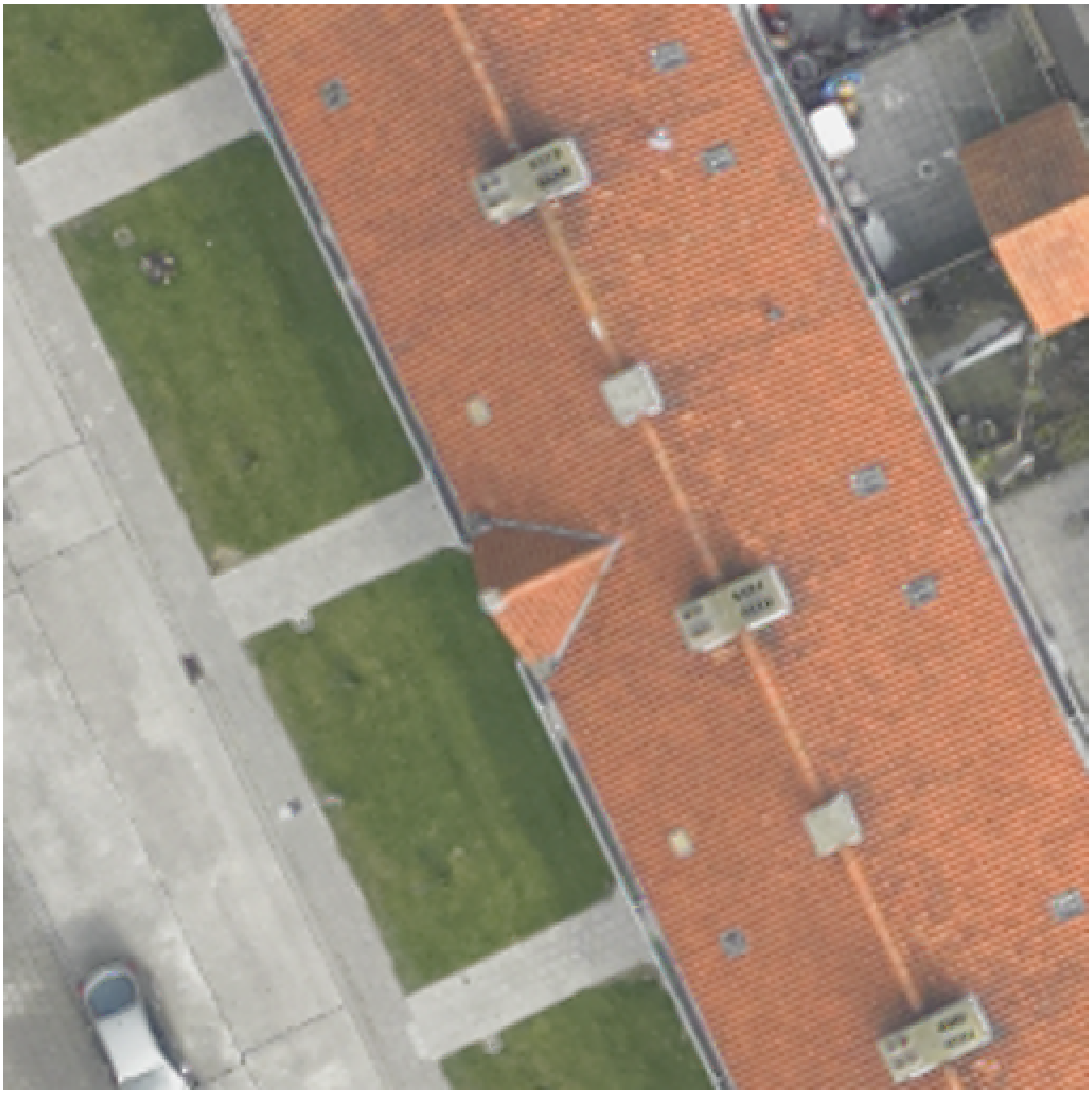


Figure 5.9: The superresolved image using Method 3.

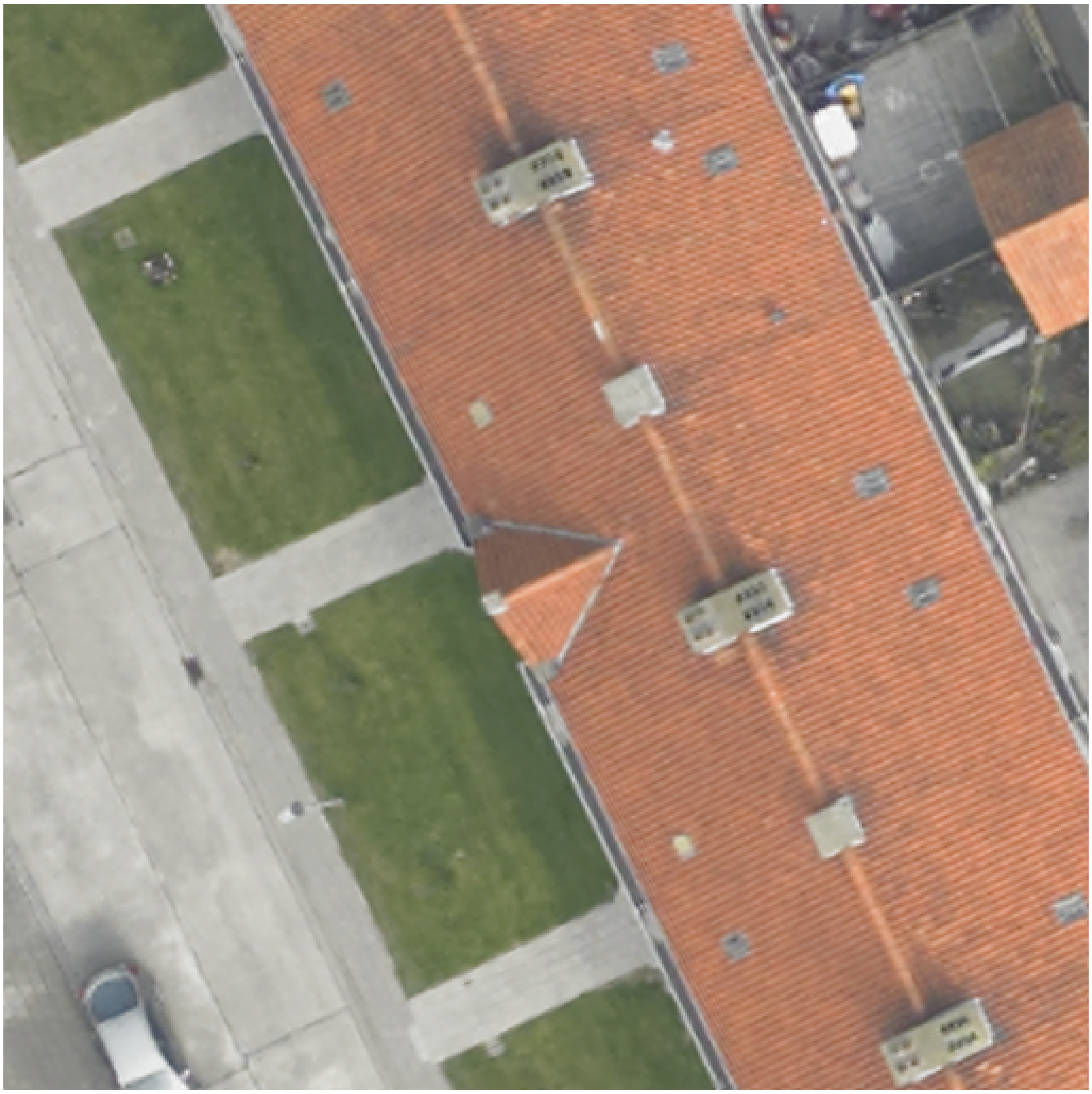
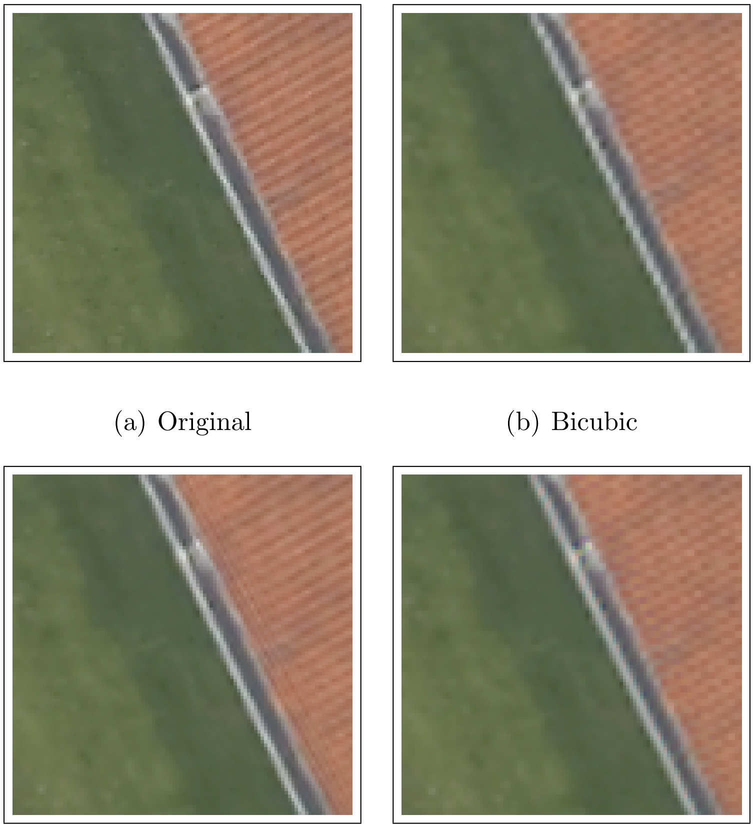


Figure 5.10: The superresolved image using Mallat's Sparse Mixing Estimators.



(a) Original

(b) Bicubic

(c) SME

(d) Our algorithm (Method 1)

Figure 5.11: A comparison of the three methods on a zoomed-in area of the image.

5.7 Further Analysis of Method 1

5.7.1 Synthetic Experiments

We decided to further analyze Method 1 of the above algorithm since it gives good results and relies solely on shearlet information. First, we considered synthetic experiments. Since our algorithm incorporates anisotropic information, we wanted to study their efficacy on images that have very prominent directional content. We constructed 1024×1024 half planes in MATLAB, at various slopes. Two such half planes appear in Figure 5.12.

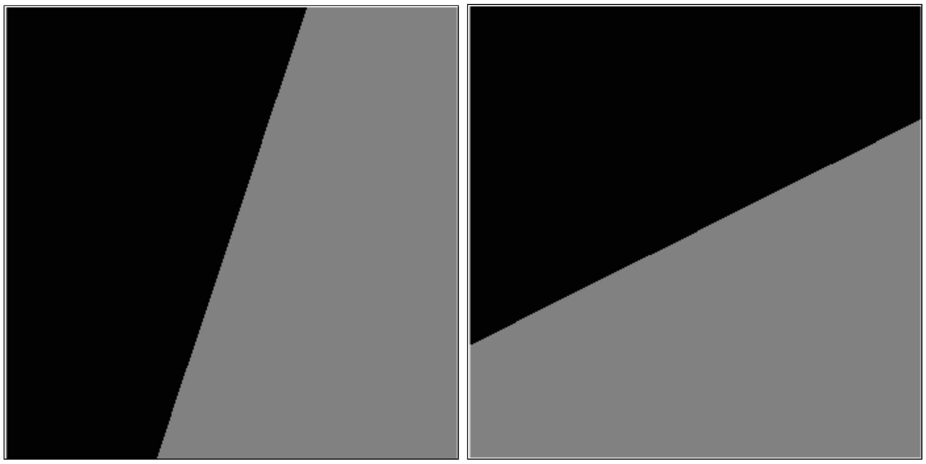


Figure 5.12: Half planes of slope 3 and .5, respectively.

We hypothesized that our anisotropic algorithms would efficiently capture the directional content, and produce high-quality superresolved images. Since there is only one major feature in these images, it may be difficult to interpret the results via visual inspection. Hence, to judge the quality of our superresolution algorithms for these half planes, we compute the *peak signal to noise ratio* (PSNR) for each of of the half planes, superresolved with either our shearlets method, circulant matrices, or bicubic. This measures the overall quality of the method by considering superresolution as a recovery problem. Recall that the PSNR in dB of a degraded image I with respect to the true image I_{ref} is defined by $10\log_{10}(peakval^2/MSE)$ where *peakval* is the largest possible value that can occur in the image and MSE is the Mean Square Error, given by

$$\frac{1}{MN}\sum_{i=1}^{M}\sum_{i=1}^{N}\left[I(i,j)-I_{ref}(i,j)\right].$$

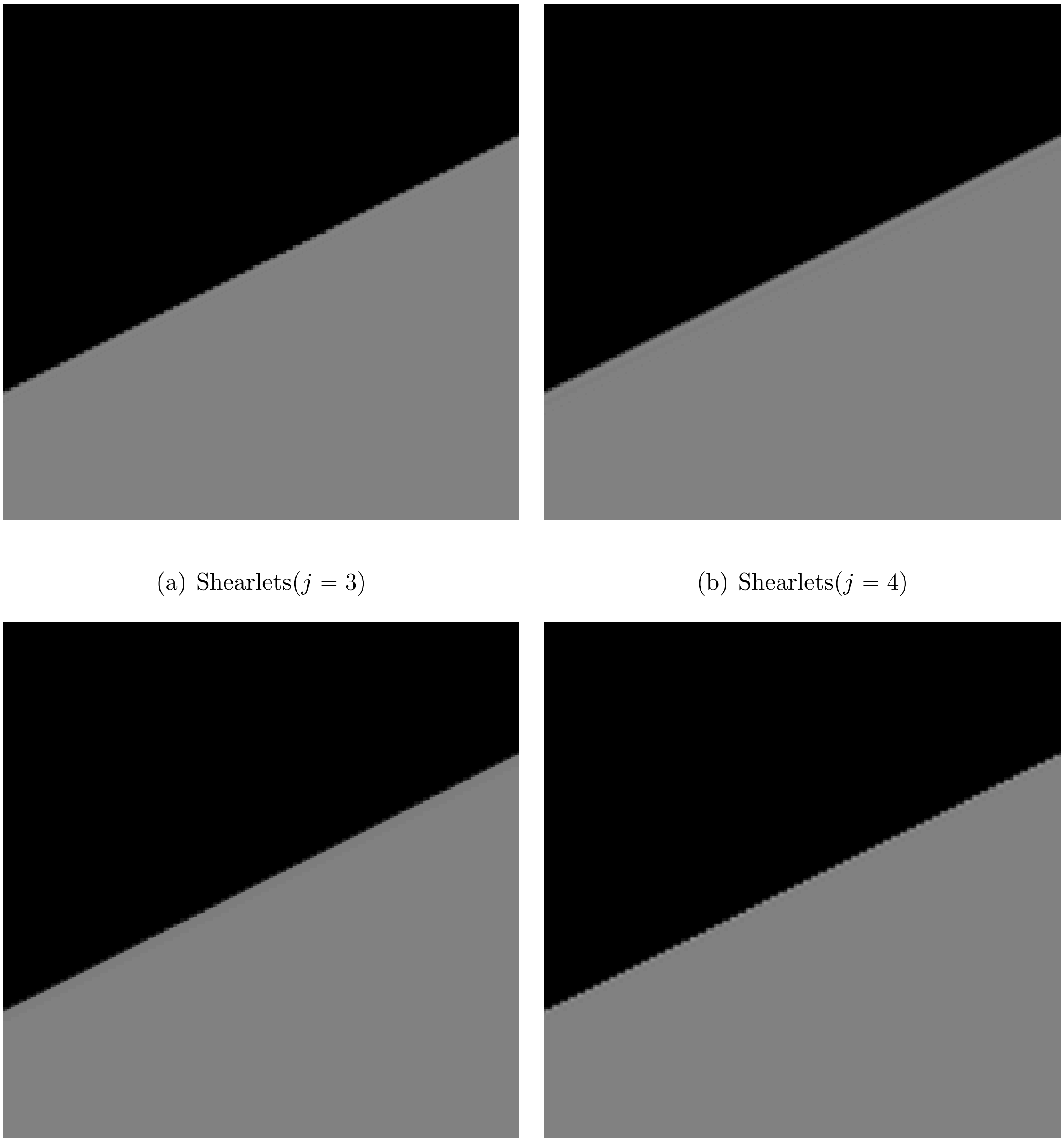
In this case, we downsample the planes to be 512 × 512, then apply our superresolution algorithms to double this resolution back to 1024×1024. We then compare these images with the original planes to compute the PSNR. The values of PSNR for each of our methods for planes at different slopes appear in Table 5.1. We test our shearlet superresolution algorithm with 16 ($j = 3$) and 32 ($j = 4$) directions, as these methods produce slightly different results. The shearlet superresolution algorithms are run with the threshold parameter $T = .05$.

Slope	Shearlet ($j = 3$)	Shearlet ($j = 4$)	Circulant Matrices	Bicubic
1/5	34.25	**34.28**	33.64	34.20
1/4	34.30	**34.41**	33.60	34.21
1/3	34.01	**34.19**	33.36	33.76
1/2	34.59	**35.32**	33.43	34.37
1	35.23	35.38	34.51	**35.40**
2	31.81	**32.18**	31.00	31.52
3	34.41	**34.61**	33.77	34.22
4	33.04	**33.11**	32.24	32.61
5	34.26	**34.29**	33.78	34.21

Table 5.1: The PSNR values for various methods on angled half planes.

We note that for all cases except one (slope 1), our shearlet-based approach with 32 directions performs the best, followed by the approaches with 16 directions, bicubic convolution, and circulant matrices, in that order. It is not surprising that using higher frequency shearlets on a perfect edge works better than using lower frequency shearlets since the algorithm more precisely locates the edge. The circulant matrices method surprisingly performs the worst of all methods. We hypothesize that this is due to the initial step of nearest-neighbors interpolation, which introduces a large amount of error that can not be fixed later by the motion blur. The effect is most prominent on straight, single-pixel edges. Finally, it is interesting to note that none of our methods improved upon bicubic in the slope 1 case. We believe that this is due to the fact that the ideal line is maximally jagged, so any attempts to remove jaggedness only reduces similarity to the ideal image.

The differences can also be seen visually. In Fig. 5.13, we present the center 256×256 subset of the superresolved half plane of slope 1/2. All directional methods appear better than bicubic interpolation.



(a) Shearlets($j = 3$)

(b) Shearlets($j = 4$)

(c) Circulant matrices

(d) Bicubic

Figure 5.13: Comparison of methods for the half plane of slope 1/2. Only the center 256×256 pixels are shown for ease of visualization. Note that the directional methods produce fewer jagged edges.

We next considered a synthetic experiment on a circle, which appears in Figure

5.14. We again performed superresolution with our shearlet algorithm, the circulant matrix algorithm, and bicubic interpolation, and computed PSNR in a method identical to the one discussed above. The results appear in Table 5.2.

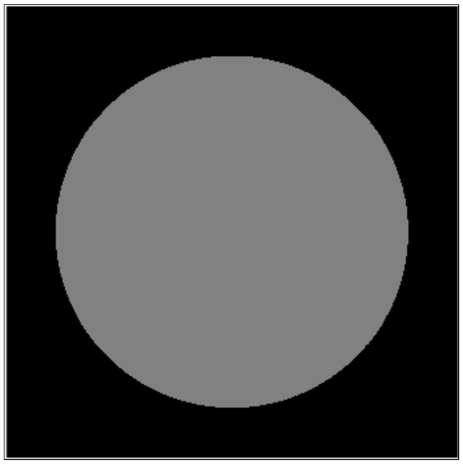


Figure 5.14: Synthetic circle to be tested.

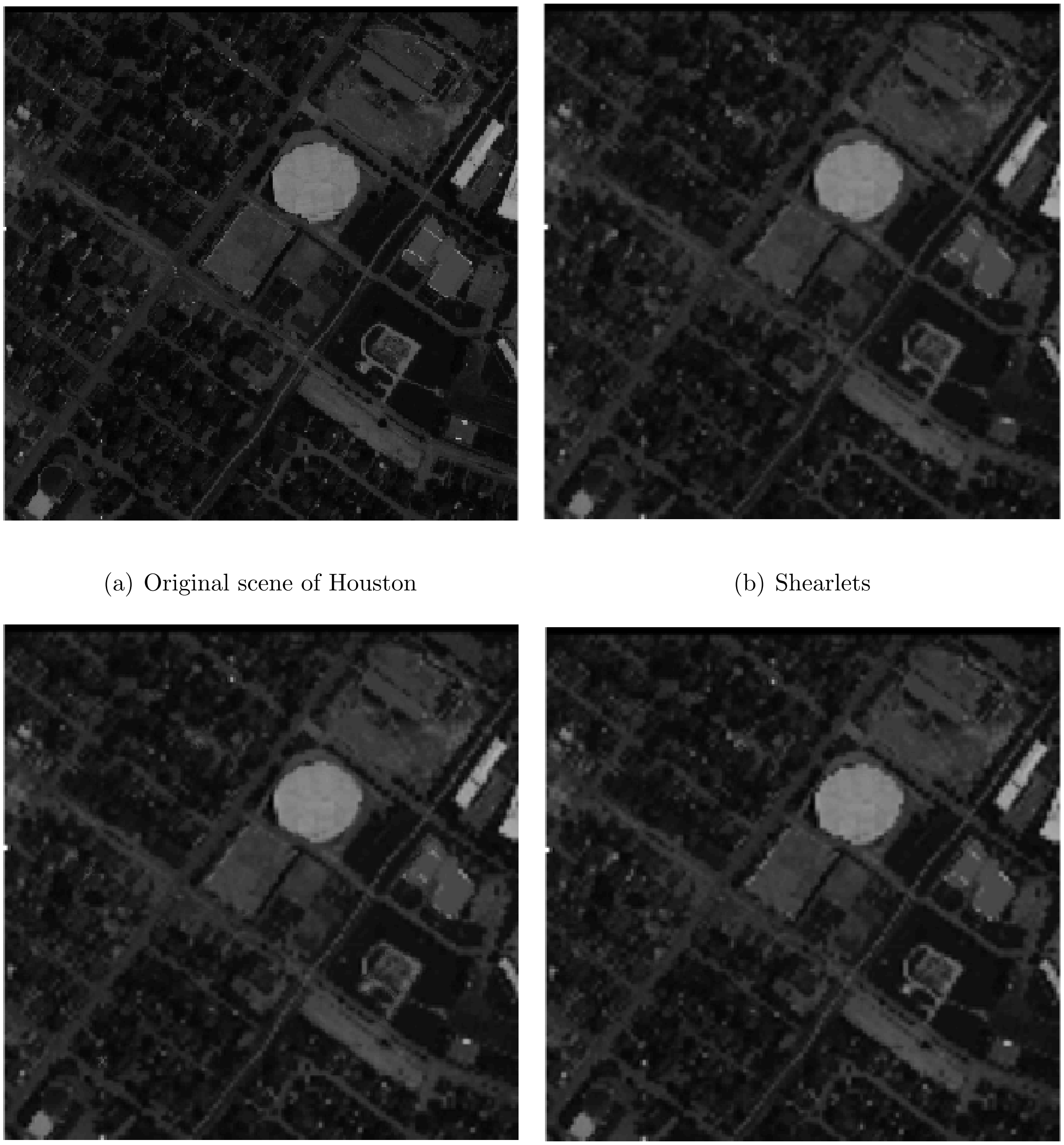
Shearlet ($j = 3$)	Shearlet ($j = 4$)	Circulant Matrices	Bicubic
30.90	30.88	30.56	30.30

Table 5.2: The PSNR values for the circle.

For the circle experiment, the shearlet algorithms give very similar error and all anisotropic methods outperform bicubic interpolation. We conclude that in the case of simple, synthetic experiments, shearlet-based superresolution provides superior performance when compared to conventional bicubic interpolation. The circulant matrix method underperforms on straight lines by PSNR, but works better than bicubic for curved lines.

5.7.2 Hyperspectral Experiments

We also tested our algorithms on real data. We utilized a hyperspectral data set of the University of Houston and surrounding area, which consisted of 144 bands of size 349×1905. This dataset has a spectral resolution of 2.5 m, and spectral resolution of between 380 nm and 1050 nm. We extracted the information from band 70, for its relatively high contrast, and rescaled the data so that all values lay between 0 and 1. For convenience, we extracted a 256×256 subset from the upper right of the image. We performed superresolution experiments with our shearlet algorithm with 16 directions, the circulant matrices algorithm, and bicubic interpolation. We note that in accordance with the bound on the number of directions generated by the shearlet algorithm, this image is too small to consider 32 shearlet directions. To detect pixels with local direction, we set the threshold $T = .02$. The images resulting from these experiments appear in Figure 5.15. The direction map for the shearlet algorithm appears in Figure 5.16.



(a) Original scene of Houston

(b) Shearlets

(c) Circulant matrices

(d) Bicubic

Figure 5.15: Original image of Houston and results of our superresolution experiments. Notice that the algorithms based on anisotropic harmonic analysis produce smoother edges, when compared to bicubic interpolation.

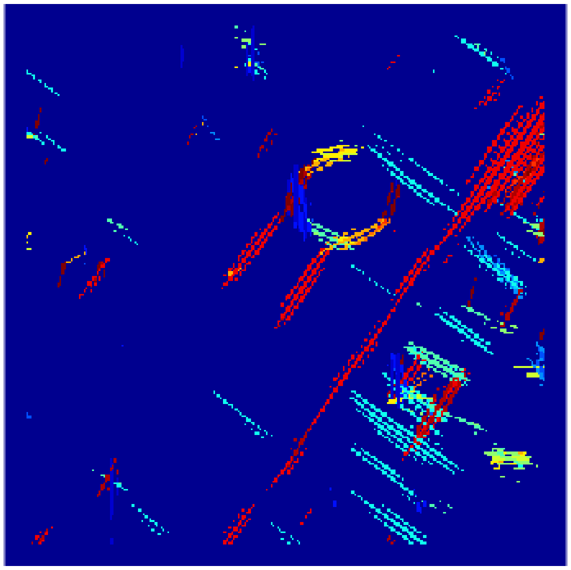


Figure 5.16: Direction map for the shearlet algorithm applied to the Houston scene. There are 16 possible local direction for each pixel, varying from 90° (blue) to $\approx 256°$ (red). The darkest blue corresponds to no assigned local direction.

We quantitively analyzed the results of our algorithm as before, by computing PSNR. The results appear in Table 5.3. We again see from these results that our shearlet algorithm gives superior performance over simple bicubic interpolation. However, we don't have the large gains in PSNR that we saw in the synthetic examples. We find that our algorithm has difficulty distinguishing textures from edges, as both have large shearlet coefficients. As a result, textures become blurred, decreasing PSNR in these areas. Distinguishing textures from edges is notoriously difficult [73]. In addition, textures are hard to superresolve; bicubic interpolation may in fact be one of the best ways to superresolve textures, unless we assume further structure such as self-similarity [74]. To improve upon our method, we would first need to be able to filter out textures. In the Zeebruges image, many of the textures, such as the roof shingles, were ignored by focusing on the second finest scale. Therefore, the method works best when the image's maximum dimension is 256 or larger, allowing us to filter out the textures, while still reasonably begin able to detect the edges.

Shearlet ($j = 3$)	Circulant Matrices	Bicubic
32.23	32.15	32.18

Table 5.3: The PSNR values for the University of Houston scene.

5.8 Conclusions and Future Directions

In general, anisotropic harmonic analysis provides a powerful set of techniques for superresolution in terms of visual quality. One of the greatest challenges for our algorithm is superresolving images with many textures without oversmoothing. In future work, we would like to find a method for filtering out the textures so as to only smooth the edges. In addition, we would like to consider more sophisticated ways of improving edges beyond motion blurring, which tends to decrease image sharpness. We will further study Mallat's SME method, which makes use of directional interpolation, to determine if shearlets can offer any improvements.

Chapter 6: Directionality in Cell Biology: Phototaxis

6.1 Introduction

Directionality is of fundamental importance to science and mathematics. In many image processing applications, it is the most directional features, the edges, which are of primary concern. We seen this in with road detection (Chapter 4) and superresolution (Chapter 5). It has been hypothesized that the edges of an image are sparsely represented in the human visual cortex [75]. The human eye is naturally drawn toward the edges of an object, so preserving them is key for producing high quality images. Remote sensing techniques, such as radar, sonar, and LIDAR, involve emitting signals in the form of sound or light. Signals that return to the emitter provide information for the direction of important features. Dimensionality reduction techniques that require the computation of eigenvectors, such as Laplacian eigenmaps, work by detecting the most significant directions, and hence features, in the data. In this chapter, we explore how directionality is important in the field of cell biology. This is joint work with Doron Levy and is based on work published in [5].

6.2 Preliminaries

In a series of recent works [76–81], mathematical models were developed to describe various aspects of the motion of the cyanobacteria *Synechocystis* sp., which are coccoidal bacteria that move towards light, a motion known as phototaxis. As a result of this motion, finger-like appendages form on a large scale [82, 83]. In contrast, in regions of low and medium density, cells follow a quasi-random pattern of motion in which small aggregates form, yet bacteria may still move in various directions without any observable bias in the direction of the light source.

This quasi-random motion in regions of low-density was the focus of the works in [77, 78] in which mathematical models were developed to describe the emerging patterns of motion. The approach was to construct stochastic particle models in which individual particles could move according to a prescribed set of rules at discrete time steps. The rules of motion allowed the particles to persist in their previous direction of motion, become stationary or start moving if already stationary, and change the direction of their motion. When a particle changes its direction of motion, it can only choose to move towards one of its neighbors. Particles can detect their neighbors within a given detection range.

In order to gain a better understanding of the mathematical model, a one-dimensional version of the stochastic model from [77, 78] was developed, in which particles were constrained to move on a one-dimensional lattice [76]. In this context, it became possible to develop a system of ODEs that quantify the expected number of particles at each position, following the method outlined in [84]. The results of the stochastic model agreed in many cases with the results of the deterministic model, depending on the choice of parameters. In addition, randomly chosen initial conditions in the deterministic model led to the formation of aggregates in most cases.

In this chapter, we generalize the one-dimensional model from [76] to motion on a two-dimensional lattice and use numerical simulations to study the emerging patterns. Similar to [76], our study starts with a stochastic particle system and proceeds with a system of ODEs that captures the averaged behavior of the discrete system.

It is important to note that this study is an example of a flocking model. Mathematical models of flocking phenomena have became very popular in recent years, most of which intend to describe a process in which self-propelled individual organisms act collectively. Examples for such models include flocking models for fish [85–88], birds [89, 90], and insects [91, 92], among many others. Various mechanisms have been proposed in the literature for changing the direction of motion. In [90], Reynolds models a flock of birds using the rules of collision avoidance, velocity matching, and attraction within a certain radius. Vicsek *et al.* propose a simple model where the only rule is for each individual to assume the average direction of its neighbors, with some random perturbation [93]. In the model of Couzin *et al.*, particles have a zone of repulsion, a zone of orientation in which they match their neighbors' directions, and a zone of attraction [94]. The Cucker-Smale model proposes that a bird changes its velocity at each time step by adding a weighted average of the differences between its velocity and those of other birds [89]. In contrast, our approach requires a particle to move towards *one* of its neighbors.

The structure of this chapter is as follows. After reviewing the one-dimensional models in Section 6.3, we introduce the two-dimensional stochastic particle model in Section 6.4.1. Multiple simulations of the stochastic particle model are conducted in Section 6.4.2. We observe the formation of horizontal and vertical aggregates whose lengths depend upon the choice of parameters.

In Section 6.5.1, we derive a system of ODEs that captures the averaged behavior of the stochastic particle model. The correspondence between the stochastic particle model and the ODEs model is demonstrated in Section 6.5.2. The ODEs system also results in a formation of aggregates, at least when the model parameters are confined to a certain range. Concluding remarks are provided in Section 6.6.

6.3 Review of the One-Dimensional Models

We start by reviewing the one-dimensional model from [76]. Consider a set of N particles that occupy the k vertices of a one-dimensional lattice. There are no restrictions on the number of particles that can occupy each bin. We fix a detection radius $D \geqslant 1$, which determines how far away a particle can detect neighboring particles. At every discrete time-step, each particle can either

(i) persist in its last direction with probability a,

(ii) become stationary with probability b,

(iii) choose to move towards another particle within its detection radius with neighbor-weighted probabilities.

If we denote the position of the n^{th} particle at time $t \in \mathbb{N}$ by $x_n(t)$ and the direction of last movement by $p_n(t) \in \{\pm 1\}$, the rules of motion are

$$x_n(t+1) - x_n(t) = \begin{cases} p_n(t), & \text{w.p. } a, \\ 0, & \text{w.p. } b, \\ 1, & \text{w.p. } (1-a-b)\dfrac{\nu_n^r(t)}{\nu_n(t)}, \\ -1, & \text{w.p. } (1-a-b)\dfrac{\nu_n^l(t)}{\nu_n(t)}, \end{cases} \tag{6.1}$$

where $\nu_n^r(t)$ and $\nu_n^l(t)$ denote the number of particles at time t that are positioned within a neighborhood of size D centered at the particle n to the right and to the left, respectively. The sum of these two quantities is denoted by $\nu_n := \nu_n^r + \nu_n^l$. Particles that are located at the same position as particle n are excluded from this sum.

Note that these rules are not enough to determine the process since we must also specify rules for movement on the boundary. A simple choice is of a periodic lattice. By doing so, we have a discrete-time Markov process in which the state at each time consists of the positions of the particles as well as the directions of their last movement. Other types of boundary conditions can be used.

The process is difficult to analyze except by simulation, hence in [76], a reaction-diffusion master equation (RDME) is developed that describes how the probabilities of all of the possible states of the systems change in time. The state of the system at any time can be defined by the vectors $\{\vec{r}, \vec{l}, \vec{r^s}, \vec{l^s}\}$, which records the number of right-moving, left-moving, right-moving but currently stationary, and left-moving stationary particles at each position. Define the probability density function $P(\vec{r}, \vec{l}, \vec{r^s}, \vec{l^s}, t)$, which describes the chance of the system being in a certain state at time t. In this case, the RDME can be described by

$$\begin{aligned}\frac{\partial P}{\partial t}(\vec{r}, \vec{l}, \vec{r^s}, \vec{l^s}, t) = \sum_{i=1}^{k} &\left[Pr\left(\text{ the state is entered by a particle in bin } i \text{ moving right}\right)\right.\\ &\left. - Pr\left(\text{the state is exited by a particle in bin } i \text{ moving right}\right)\right]\\ &+ \sum_{i=1}^{k}\left[Pr\left(\text{the state is entered by a particle in bin } i \text{ moving left}\right)\right.\\ &\left. - Pr\left(\text{the state is exited by a particle in bin } i \text{ moving left}\right)\right]\\ &+ \sum_{i=1}^{k}\left[Pr\left(\text{the state is entered by a particle in bin } i \text{ becoming stationary}\right)\right.\\ &\left. - Pr\left(\text{the state is exited by a particle in bin } i \text{ becoming stationary}\right)\right].\end{aligned} \tag{6.2}$$

Using the explicit forms of the expressions in (6.2), ODEs are derived for the expected number of each type of particle in each bin. For example, multiplying (6.2) by r_i, summing over all possible states, and switching the order of differentiation and summation, results in an ODE for the expected number of right-moving particles in bin i. For more details, we refer to [76]. The resulting system is

$$\begin{aligned}\frac{dR_i}{dt} &= a(R_{i-1} + R^s_{i-1}) + \left\langle n_{i-1}c^r_{i-1}\right\rangle - R_i,\\ \frac{dR^s_i}{dt} &= bR_i - (1-b)R^s_i,\\ \frac{dL_i}{dt} &= a(L_{i+1} - L^s_{i+1}) + \left\langle n_{i+1}c^l_{i+1}\right\rangle - L_i,\\ \frac{dL^s_i}{dt} &= bL_i - (1-b)L^s_i.\end{aligned} \tag{6.3}$$

Here, n_i corresponds to the total number of particles in bin i, and $\langle x\rangle$ is the expected value of x. The first equation in (6.3) corresponds to right moving particles in bin

i, $R_i(t)$. This population consists of right-moving and stationary particles from bin $i-1$ that persist into bin i, and particles in bin $i-1$ moving to the right with a neighbor-weighted probability c^r_{i-1}. All right-moving particles leave the system at every time step, either by persisting in their motion to the right, becoming stationary, or choosing to move toward a neighboring bin. The second equation in (6.3) corresponds to right-stationary particles R^s_i. These consist of right-moving particles that become stationary with stopping probability b, accounting for particles that leave the stationary state with probability $1-b$. Similar expressions are given by equations three and four in (6.3) for the left-moving and the left-stationary particles. In practice, in order to simulate the system (6.3), the expected values of the products are replaced by the products of the expected values. Although this is only true for uncorrelated random variables, qualitative agreement is obtained between the discrete and ODEs model.

6.4 A Two-Dimensional Stochastic Particle Model

6.4.1 Model Formulation

Assume that N particles are located on the vertices of a $k \times k$ periodic lattice. As in the one-dimensional model, we assume that particles remember their previous direction of movement and can either continue in that direction, choose a new direction, or remain stationary. The detection radius D can be generalized to 2-D by counting the particles within a Euclidean distance of D. To simplify the calculations, we fix the detection radius to be 1 so that particles can only detect adjacent particles.

We denote the number of particles detected by the nth particle to the left, right, up, and down at time t by $\nu_n^l(t), \nu_n^r(t), \nu_n^u(t)$, and $\nu_n^d(t)$, respectively, with $\nu_n = \nu_n^l + \nu_n^r + \nu_n^u + \nu_n^d$. Let $x_n(t)$ be the position of the nth particle on the lattice with respect to some arbitrary origin at time t . Let a be the probability of persistence and b the probability of becoming or remaining stationary. Finally, let $p_n(t)$ be the vector representing the last direction of movement for particle n. The resulting rules of motion are

$$x_n(t+1) - x_n(t) = \begin{cases} p_n(t), & \text{w.p. } a, \\ 0, & \text{w.p. } b, \\ (1,0), & \text{w.p. } (1-a-b)\dfrac{\nu_n^r(t)}{\nu_n(t)}, \\ (-1,0), & \text{w.p. } (1-a-b)\dfrac{\nu_n^l(t)}{\nu_n(t)}, \\ (0,1), & \text{w.p. } (1-a-b)\dfrac{\nu_n^u(t)}{\nu_n(t)}, \\ (0,-1), & \text{w.p. } (1-a-b)\dfrac{\nu_n^d(t)}{\nu_n(t)}. \end{cases} \tag{6.4}$$

If a particle has no neighbors, $\nu_n = 0$, then we choose to increase the probability of particle n becoming stationary to $1-a$.

6.4.2 Simulations of the Stochastic Model

All discrete simulations are conducted using the NetLogo multi-agent programable modeling environment (ccl.northwestern.edu/netlogo). We start by uniformly distributing 4000 particles on a 21 × 21 grid. Each particle is assigned a memory in the form of a last direction of motion (left, right, up, or down) with equal probabilities. The initial distribution of particles with their associated memories is shown in Fig. 6.1. We use these initial conditions for all simulations on the 21×21 grid.

The initial configuration is then advanced in time, until $t = 1500$, with different values of the parameters. Figures 6.2-6.6 show snapshots of the simulation for a persistence probability $a \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ in which the probability to remain stationary is set as $b = 0$. In these figures, arrows indicate the direction of movement of particles that are located in any given bin. Bins with no arrows have no particles in them. Each bin may have more than one particle, and hence the total number of arrows in all figures is less than the number of particles (which is 4000).

We define an aggregate as a group of particles occupying horizontally or vertically adjacent bins traveling in the same or opposite directions. In all figures, we see that the particles form horizontal and/or vertical aggregates, which coalesce into fewer aggregates as time elapses. As the value of the persistence probability a increases, the length of the aggregates seem to increase. For example, when $a = 0.1$, the average length of the aggregates (ignoring insignificant ones) is approximately 8, while for $a = 0.2$, the average length of the aggregates is approximately 12. When $a = 0.3$ (Fig. 6.4), the limit aggregates span the entire length of the grid. In general, the average length of significant aggregates does not appear to change significantly after an initial transient period. We can propose an explanation for the relationship between a and aggregate length. The length of an aggregate can only increase if an end particle moves away from its neighboring particles, which occurs with probability a when it is facing away from the aggregates. This increase in length can only be maintained if a significant number of other particles also choose to move to the new endpoint; however, it is much more likely that the new endpoint will move back to its previous position when a is small, so the aggregates length remains unchanged.

When the persistence probability is $a = 0.5$ (Fig. 6.6), aggregates are very slow to form. This is due to the high probability of particles continuing their last movement, as opposed to moving towards other particles. At large times, we are left with two long aggregates: one horizontal and one vertical. They eventually coalesce into a single long aggregate (not shown).

We note that if we assume a constant nonzero probability of continuing in the same direction, the system is never reaches a steady state and, in particular, stable length-one aggregates do not form. If we set both parameters to 0 for our current initial conditions, the system quickly enters steady state with only aggregates of length two (Fig. 6.7).

We now investigate the impact of the probability of remaining stationary, b, on the emerging dynamics. We use the same initial configuration given by Fig. 6.1, fix the persistence probability as $a = 0.3$, and vary $b \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. The results of these simulations are shown in Figs. 6.8−6.12.

Most of these results are quite similar: many long aggregates, both horizontal and vertical, form quickly. As time elapses, they coalesce into fewer aggregates, which are all only horizontal or vertical. Most aggregates span the entire grid, but we can also get smaller aggregates as in Fig. 6.8 and 6.11. As b increases, it takes longer for significant aggregates to form. This makes sense since a particle is more likely to remain stationary. When we continue the simulation past $t = 1500$, we are eventually left with purely horizontal or vertical aggregates, though the exact number varies. These results are not shown for the sake of brevity. We hypothesize that this will always be true for any initial conditions as long as the parameters are large enough, though how big the parameters need to be is dependent on grid size and number of particles. We advanced time for the parameter choice $a = 0.1, b = 0$ to $t = 5000$, but we continue to have both types of aggregates (Fig. 6.13).

It is important to note that since the model is stochastic, the outcomes for a given initial configuration is not unique. This is demonstrated in Fig. 6.14 in which we show different results that are obtained at $t = 1500$ for the same initial conditions. In most cases (Fig 6.14a, c, d, e, f), purely horizontal or vertical aggregates are all that remain. In Fig. 6.14b, however, we have both horizontal and vertical aggregates that will coalesce if we wait sufficiently long.

6.5 An ODEs Model

6.5.1 Model Derivation

Since simulating a large number of particles on a large grid is computationally intensive, we derive a system of ODEs to determine the mean number of particles in each bin. Let $P = P(\Psi)$ denote the probability of the system being in a given state $\Psi = (r, l, u, d, r^s, l^s, u^s, d^s)$. Here, r is a $k \times k$ matrix that denotes the number of right-moving particles at every node. Similarly, $l, u,$and d are matrices that correspond to the number of left, up, and down-moving particles in every node, respectively. The variables with a superscript "s" correspond to the stationary particles and are divided into four groups based on the last direction of motion that brought them to their present location, a direction which they remember.

Generally, the evolution of the probability is given by:

$$\begin{aligned}\frac{\partial P}{\partial t}(\Psi, t) = \sum_{i,j,\mathcal{D}} \Bigg[&\begin{pmatrix}\text{Probability that a particle moves out of node } (i,j) \\ \text{in direction } \mathcal{D} \text{ to enter the state } \{\Psi, t\}\end{pmatrix} \\ &- \begin{pmatrix}\text{Probability that a particle moves out of node } (i,j) \\ \text{in direction } \mathcal{D} \text{ to leave the state } \{\Psi, t\}\end{pmatrix} \Bigg] \\ + \sum_{i,j} \Bigg[&\begin{pmatrix}\text{Probability that a particle in node } (i,j) \text{ becomes stationary} \\ \text{to enter the state } \{\Psi, t\}\end{pmatrix} \\ &- \begin{pmatrix}\text{Probability that a particle in node } (i,j) \text{ becomes stationary} \\ \text{to leave the state } \{\Psi, t\}\end{pmatrix} \Bigg].\end{aligned} \tag{6.5}$$

The summation over the direction $\mathcal{D}$ corresponds to a summation in all directions: left, right, up, and down.

Let $\tilde{P}(.,.)$ denote the probability P of a state with a specified change. For example, $\tilde{P}(r_{ij}+1, r_{i+1,j}-1)$ is P with the number of right-moving particles at node (i,j), r_{ij}, increased by one, and the number of right-moving particles at $(i+1,j)$ decreased by one. Note that $\tilde{P}$ is not a function; it is notation that greatly simplifies the derivation below. We recall that v_{ij}^r denotes the number of particles within the detection range to the right of node (i,j), and that v_{ij} is the total number of particles within the detection range in all directions surrounding node (i,j). Accordingly, we define

$$c_{ij}^r = (1-a-b)\frac{v_{ij}^r}{v_{ij}},$$

as the probability that a particle at (i,j) chooses to move to the right. In addition, we define the probability

$$\overline{c_{ij}^r} = (1-a-b)\frac{v_{ij}^r - 1}{v_{ij} - 1}.$$

$\overline{c_{ij}^r}$ is the probability that a particle at (i,j) chooses to move to the right when the number of particles to the right of (i,j) within the detection range is $v_{ij}^r - 1$ and the total number of particles within the detection range in all directions surrounding (i,j) is $v_{ij} - 1$. Similar quantities are defined for the other directions (left, up, and down).

To address the terms in (6.5), we start by considering the ways for the system to enter the state Ψ with a particle moving right from position (i,j). There are 8 ways in which a right-moving particle from (i,j) will result in entering the given state:

1. A right-moving particle moves out of (i,j) to the right to enter the state Ψ. We assume that in a small time step, only one particle can move at a time and it can only move a single space. Therefore, the only way this can occur is if there were previously $r_{ij} + 1$ right-moving particles at (i,j) and $r_{i+1,j} - 1$ right-moving particles at $(i+1,j)$. Since there is one fewer particle at $(i+1,j)$ compared to state Ψ, there are $\nu_{ij} - 1$ detectable particles and $\nu_{ij}^r - 1$ detectable right-moving particles. There is an additional particle at (i,j), but recall that this is not counted as a detectable particle. Hence, the rate at which these particles could move to the right is the sum of the persistence probability and the neighbor-weighted probability, i.e., $a + \overline{c_{ij}^r}$. Accordingly, the probability that a particle moves out of (i,j) to the right to enter the state

Ψ is $(a + \overline{c^r_{ij}})(r_{ij} + 1)\tilde{P}(r_{ij} + 1, r_{i+1,j} - 1)$.

2. A left-moving particle moves out of (i, j) by choosing to move to the right:

   $\overline{c^r_{ij}}(l_{ij} + 1)\tilde{P}(l_{ij} + 1, r_{i+1,j} - 1)$.

3. An up-moving particle moves out of (i, j) by choosing to move to the right:

   $\overline{c^r_{ij}}(u_{ij} + 1)\tilde{P}(u_{ij} + 1, r_{i+1,j} - 1)$.

4. A down-moving particle moves out of (i, j) by choosing to move to the right:

   $\overline{c^r_{ij}}(d_{ij} + 1)\tilde{P}(d_{ij} + 1, r_{i+1,j} - 1)$.

5. A right-moving but stationary particle initiates a motion to the right:

   $(a + \overline{c^r_{ij}})(r^s_{ij} + 1)\tilde{P}(r^s_{ij} + 1, r_{i+1,j} - 1)$.

6. A left-moving and stationary particle moves to the right:

   $\overline{c^r_{ij}}(l^s_{ij} + 1)\tilde{P}(l^s_{ij} + 1, r_{i+1,j} - 1)$.

7. An up-moving and stationary particle moves to the right:

   $\overline{c^r_{ij}}(u^s_{ij} + 1)\tilde{P}(u^s_{ij} + 1, r_{i+1,j} - 1)$.

8. A down-moving and stationary particle moves to the right:

   $\overline{c^r_{ij}}(d^s_{ij} + 1)\tilde{P}(d^s_{ij} + 1, r_{i+1,j} - 1)$.

Combining these eight paths of entering the state Ψ, we obtain the total probability of entering state Ψ via a particle moving to the right into bin (i,j):

$$\begin{aligned}
&(a+\overline{c^r_{ij}})(r_{ij}+1)\tilde{P}(r_{ij}+1,r_{i+1,j}-1)+\overline{c^r_{ij}}(l_{ij}+1)\tilde{P}(l_{ij}+1,r_{i+1,j}-1)\\
&\quad+\overline{c^r_{ij}}(u_{ij}+1)\tilde{P}(u_{ij}+1,r_{i+1,j}-1)+\overline{c^r_{ij}}(d_{ij}+1)\tilde{P}(d_{ij}+1,r_{i+1,j}-1)\\
&\quad+(a+\overline{c^r_{ij}})(r^s_{ij}+1)\tilde{P}(r^s_{ij}+1,r_{i+1,j}-1)+\overline{c^r_{ij}}(l^s_{ij}+1)\tilde{P}(l^s_{ij}+1,r_{i+1,j}-1) \quad (6.6)\\
&\quad+\overline{c^r_{ij}}(u^s_{ij}+1)\tilde{P}(u^s_{ij}+1,r_{i+1,j}-1)+\overline{c^r_{ij}}(d^s_{ij}+1)\tilde{P}(d^s_{ij}+1,r_{i+1,j}-1).
\end{aligned}$$

Our goal now is to derive an expression for the expectation of the number of the right-moving particles at node (m,n), r_{mn}. Hence, we multiply (6.6) by r_{mn} and sum over i,j, and all possible states Ψ.

Assume for simplicity that neither m nor n are 1 or k. We consider the first term in (6.6), and change variables $r'_{ij}=r_{ij}+1$ and $r'_{i+1,j}=r_{i+1,j}-1$. If $(i,j)\neq(m,n)$ and $(i,j)\neq(m-1,n)$, then (after returning to our original variables) we are left with $r_{mn}(a+c^r_{ij})r_{ij}P$. If $(i,j)=(m,n)$, then we have $(r_{mn}-1)(a+c^r_{mn})r_{mn}P$. If $(i,j)=(m-1,n)$, we get $(r_{mn}+1)(a+c^r_{m-1,n})l_{m-1,n}P$. Therefore, the contribution of first term in (6.6) is

$$\sum_{i,j,\Psi} r_{mn}(a+c^r_{ij})r_{ij}P-(a+c^r_{mn})r_{mn}P+(a+c^r_{m-1,n})r_{m-1,n}P. \qquad (6.7)$$

For the second term in (6.6): If $(i,j)\neq(m-1,n)$, we get $r_{mn}c^r_{ij}l_{ij}P$, while if $(i,j)=(m-1,n)$, we get $(r_{mn}+1)c^r_{m-1,n}r_{m-1,n}P$. Therefore, the contribution of the second term in (6.6) is

$$\sum_{i,j,\Psi} r_{mn}c^r_{ij}l_{ij}P+c^r_{m-1,n}l_{m-1,n}P. \qquad (6.8)$$

Similar expressions hold for the other terms in (6.6).

We now return to (6.5) and consider ways for the system to leave a state due to a particle moving right. There are 8 ways for the system to leave state Ψ due to the motion of a particle to the right. These can be represented by the following expression:

$$-\left[(a+c^r_{ij})r_{ij}P + c^r_{ij}l_{ij}P + c^r_{ij}u_{ij}P + c^r_{ij}d_{ij}P + (a+c^r_{ij})r^s_{ij}P + c^r_{ij}l^s_{ij}P + c^r_{ij}u^s_{ij}P + c^r_{ij}d^s_{ij}P\right]. \tag{6.9}$$

The terms in (6.9) represent a loss for the system, which is the reason for the negative sign. We multiply (6.9) by r_{mn} and sum over i, j and all states Ψ. After combining the result with the contributions obtained by doing the same to (6.6), we have

$$\begin{aligned}&-\langle(a+c^r_{mn})r_{mn}\rangle + \left\langle(a+c^r_{m-1,n})r_{m-1,n}\right\rangle + \left\langle c^r_{m-1,n}l_{m-1,n}\right\rangle + \left\langle c^r_{m-1,n}u_{m-1,n}\right\rangle + \left\langle c^r_{m-1,n}d_{m-1,n}\right\rangle \\ &\quad + \left\langle(a+c^r_{m-1,n})r^s_{m-1,n}\right\rangle + \left\langle c^r_{m-1,n}l^s_{m-1,n}\right\rangle + \left\langle c^r_{m-1,n}u^s_{m-1,n}\right\rangle + \left\langle c^r_{m-1,n}d^s_{m-1,n}\right\rangle,\end{aligned} \tag{6.10}$$

where $\langle x\rangle$ denotes the expectation of x, i.e., $\sum_\Psi xP(\Psi)$. Combining terms multiplied by $c^r_{m-1,n}$ and defining $\eta_{m,n}$ as the sum of all particles at (m,n) (stationary and moving), we can simplify (6.10) as

$$a\left(\langle r_{m-1,n}\rangle + \left\langle r^s_{m-1,n}\right\rangle\right) + \left\langle c^r_{m-1,n}\eta_{m-1,n}\right\rangle - \langle(a+c^r_{mn})r_{mn}\rangle. \tag{6.11}$$

In order to obtain the ODE for $\langle r_{mn}\rangle$ we have to account for additional items, i.e., the right-moving particles that choose to move in a different direction or become stationary. We omit the details for the sake of brevity. Once all terms are accounted for, the resulting equation becomes

$$\frac{d\langle r_{mn}\rangle}{dt} = a\left(\langle r_{m-1,n}\rangle + \left\langle r^s_{m-1,n}\right\rangle\right) + \left\langle c^r_{m-1,n}\eta_{m-1,n}\right\rangle - \langle r_{mn}\rangle. \tag{6.12}$$

We now define $R_{mn} = \langle r_{mn} \rangle$, $R^s_{mn} = \langle r^s_{mn} \rangle$, and use similar notation for the other directions. After switching back to i and j as indices, the ODE for right-moving particles reads

$$\frac{dR_{ij}}{dt} = a\left(R_{i-1,j} + R^s_{i-1,j}\right) + \left\langle c^r_{i-1,j}\eta_{i-1,j}\right\rangle - R_{ij}. \tag{6.13}$$

When i or j is 1 or k, equations (6.13) are valid with the appropriate adjustments due to the periodic boundary conditions.

Repeating the derivation in all other directions we obtain the system

$$\begin{aligned}
\frac{dR_{ij}}{dt} &= a\left(R_{i-1,j} + R^s_{i-1,j}\right) + \left\langle c^r_{i-1,j}\eta_{i-1,j}\right\rangle - R_{ij},\\
\frac{dL_{ij}}{dt} &= a\left(L_{i+1,j} + L^s_{i+1,j}\right) + \left\langle c^l_{i+1,j}\eta_{i+1,j}\right\rangle - L_{ij},\\
\frac{dU_{ij}}{dt} &= a\left(U_{i,j-1} + U^s_{i,j-1}\right) + \left\langle c^u_{i,j-1}\eta_{i,j-1}\right\rangle - U_{ij},\\
\frac{dD_{ij}}{dt} &= a\left(D_{i,j+1} + D^s_{i,j+1}\right) + \left\langle c^d_{i,j+1}\eta_{i,j+1}\right\rangle - D_{ij},\\
\frac{dL^s_{ij}}{dt} &= bL_{ij} - (1-b)L^s_{ij},\\
\frac{dR^s_{ij}}{dt} &= bR_{ij} - (1-b)R^s_{ij},\\
\frac{dU^s_{ij}}{dt} &= bU_{ij} - (1-b)U^s_{ij},\\
\frac{dD^s_{ij}}{dt} &= bD_{ij} - (1-b)D^s_{ij}.
\end{aligned} \tag{6.14}$$

The form of the ODEs in (6.14) is intuitive. For example, consider the equation for L_{ij}. There are two possibilities for the number of left-moving particles at (i, j) to increase. First, a most recently left-moving particle at $(i+1, j)$ could continue in its previous direction with probability a. Second, any particle at $(i+1, j)$ could choose to move to the left with probability $(1-a-b)$ multiplied by the ratio of particles to the left versus all detectable particles. Also, after an infinitesimal amount of time, all particles at (i,j) have either moved elsewhere or become stationary, hence the $-L_{ij}$ term. Similarly, the form of the equation for L^s_{ij} is also clear: increases occur when left-moving particles become stationary with probability b, while decreases occur when stationary particles become non-stationary with probability $1-b$.

6.5.2 Simulations

We follow [76] and approximate expectations of quotients and products of random variables by quotients and products of expectations to close the system. We would like to see how the ODE model compares to an ensemble average of the discrete model. First, we randomly place 1000 particles on an 11×11 grid and assign each particle a memory in the form of its last direction of motion, with equal probabilities for each direction. These initial conditions are then used for simulating the dynamics of both models. We then run the simulations until the configuration reaches a steady state ($t = 1000$ for the ODE model and $t = 500$ for the stochastic particle model). The stochastic particle model is run in NetLogo and the ODE model is run in Matlab using ODE45 with default parameters. We run the stochastic model 5000 times and average over all simulations. The results are shown in Fig. 6.15 for a variety of a and b values.

We note that there are both similarities and differences between the ODE and averaged results. In Fig. 6.15 a, b, and c, the ODEs give a few aggregates that increase in length as a increases, as observed in the discrete model simulations. Most of these aggegates can also be seen in the corresponding ensemble average, but the number of particles found in these positions is much smaller, as the averaging procedure causes the particles to be more spread out. In addition, the averaged pictures have aggregates where no aggregates appear from the ODEs. For instance, in Fig. 6.15 b and c, we have both horizontal and vertical aggregates on the right side, but only horizontal aggregates on the left. We have seen that, except for small values of the parameters, only purely horizontal or vertical aggregates arise. The ODEs, therefore, seem to give results comparable to an individual realization of the model, rather than an average. We also note that in Fig. 6.15 d, the ODEs show little variation in the number of particles found in each position. We discuss this further below.

We now proceed to simulate the ODE model in different setups. We initialize the system by taking the number of left, right, up, and down moving particles on each position to be Poisson with mean 2. We assume no stationary particles initially. Fig. 6.16 shows a time series of the ODE model on a 50×50 grid with 20208 particles. By time $t = 100$, distinct aggregates have begun to form. As time progresses, aggregates coalesce and become more Gaussian.

Fig. 6.17 shows the results obtained at $t = 1000$ when $b = 0$ and $a \in \{0.1, 0.2, 0.3, 0.4\}$. For $a = 0.1$, many small peaks form. For $a = 0.2$, we have fewer and broader peaks. Gaussian behavior is apparent for $a = 0.3$. Finally, for $a = 0.4$, there are no aggregates; this demonstrates a breakdown in the ODEs when the parameters become too large. It may be that this behavior is linked to the loss of preferential locations for aggregates to form in the discrete model.

In Fig. 6.18, we fix $b = 0.1$ and vary a. Results are shown at $t = 1000$.

There is no significant difference between the results shown in Fig. 6.18 and those that were shown in Fig. 6.17. The only noticeable difference is that Fig. 6.18c contains non-Gaussian aggregates. In fact, if we run the simulation for a longer time, Gaussian-type aggregates emerge. This is shown in Fig. 6.19 for time $t = 2000$. Such aggregates take a longer time to form due to the nonzero probability to remain stationary, b.

The simulations shown so far suggest that the only types of long-term behavior we get are Gaussian or random. However, this is not the case: for example, if we let $(a, b) = (0.35, 0)$, we always have non-Gaussian behavior no matter how long we run the simulation (see Fig. 6.20). As demonstrated on the 11×11 grid, what we are seeing in the ODE simulations is reminiscent of an individual simulation rather than an ensemble average. Accordingly, simulations of the discrete model on the larger grid also show the same patterns: horizontal and vertical aggregates that coexist at large times, whose lengths increase with a, and with the number of particles in each bin decreasing with their distance from the middle of their aggregate.

By running simulations for many more choices of parameters, we discover that whether aggregates form does not depend solely on the value of $a + b$. For example, aggregates form for $(a, b) = (0.2, 0.2)$, but not for $(0.4, 0)$. Nevertheless, there does appear to be a constraint on some linear combination of the parameters. To find this constraint, we note that aggregates form for $a = 0$ and any $0 \leqslant b < 1$. In some cases, we have numerical problems solving the ODE with $a = 0$ for large times, so instead we stop these simulations at $t = 100$. We also note that aggregates form for $(a, b) = (0.367, 0)$, but not for $(0.368, 0)$. Hence, we hypothesize an approximate constraint for aggregate formation as $a/.368 + b < 1$. This constraint accurately predicts aggregate formation in every instance when we vary both a and b over the set $\{0, 0.1, 0.2, \ldots, 1\}$ (see Fig. 6.21). In addition, the constraint correctly predicts that aggregates do not form for $(a, b) = (0.3, 0.19)$ but do form for $(0.3, 0.182)$ (see Fig. 6.22).

In order to test whether this constraint depended on the specific initial conditions, we rerun the above calculations with new Poison-distributed initial conditions. We find that the same approximate constraint holds true for these cases as well, even if we alter the mean of the distribution. We also checked whether the constraint depended on the specific distribution used to generate the intial conditions. Choosing the distribution to be uniform on $\{0, 1, 2, \ldots, 8\}$ led to the constraint $a/.357 + b < 1$.

6.6 Conclusions

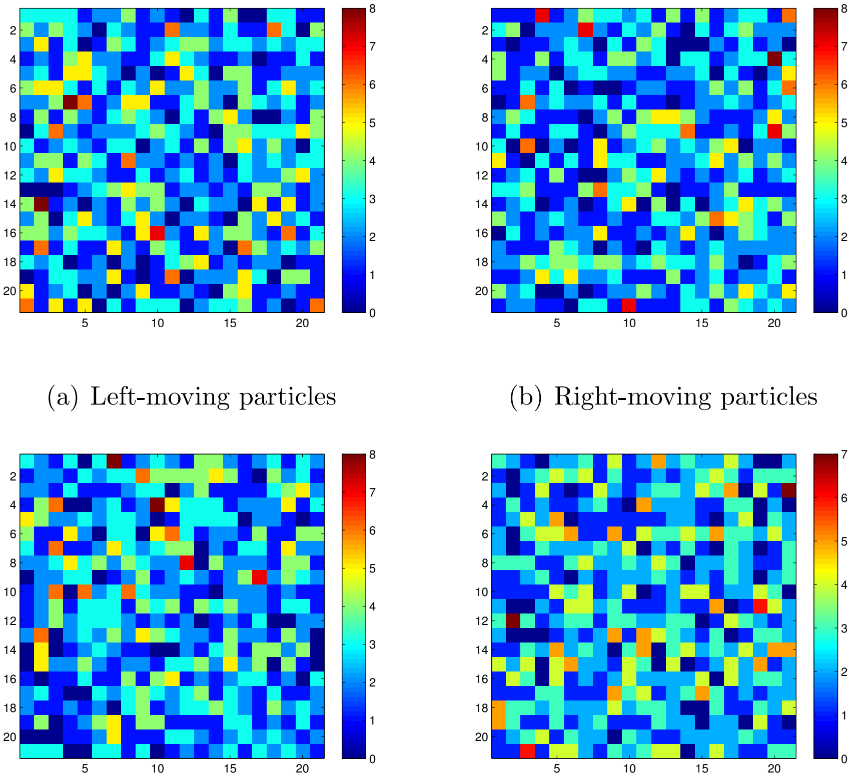
In this chapter, we generalized the one-dimensional model of Galante and Levy [76] to two dimensions. At every time step, particles may persist their motion in their current direction with probability a, remain stationary with probability b, or move toward one of their neighboring particles with equal probabilities. Since there are no exclusion principles in place, multiple particles are allowed to occupy every spot on the lattice, and hence when a particle changes its direction of motion, the new direction is chosen based on a probability that is proportional to the fraction of the neighboring particles in any given direction.

All simulations demonstrated that the limit pattern that emerges is a collection of vertical and/or horizontal aggregates. The lengths of the aggregates increases with the persistence probability a. This can be explained by the fact that aggregates can only increase in length when a particle on the aggregate edge persists, which is unlikely for small a. When a is fixed and the probability to remain stationary b varies, we note again the emergence of aggregates, the number of which steadily decreases as time elapses and aggregates coalesce. This occurs more rapidly for small values of b as expected. Our simulations suggest that we will always be left with purely vertical or horizontal aggregates if the simulations are run for a sufficiently long time, assuming there's a non-zero probability of particles moving towards their neighbors and the parameters are not too small.

The discrete system was followed by a system of ODEs that was derived in order to capture the time evolution of the expected number of particles. Since this requires us to estimate expectations of products and quotients by products and quotients of expectations, we run Monte Carlo simulations of the discrete model to compare to the ODE results. Though quantitatively different, the ODE model captures much of the behevior seen in individual runs of the particle model. We then run the ODE model for fixed b and varying a. As in the discrete simulations, we obtain aggregates whose lengths increase with a. Unlike the discrete simulations, the ODEs transition into randomness when a linear combination of the parameters becomes too large. This raises some questions including: Why does this transition occur? Why is the constraint linear? Why is the constraint largely unaffected by the initial conditions, including the particular distribution used? These questions can form the topic of further study.

In addition, we would like to study the effects of increasing the detection radius, as was done in [76]. This is non-trivial since particles would then be able to detect other particles in directions other than those in which they are allowed to move. If, say, $D = \sqrt{2}$, four diagonal positions would also be detectable. We would need to develop new rules for how these particles would affect the neighbor-weighted probabilities. For example, particles in the upper-left diagonal should contribute to the probability of motion in both the up and left directions. These contributions should be inversely proportional to their distance. Another possible addition to the model would be allowing particles to move in the diagonal directions. Incorporating both of these modifications would make the model more realistic and give rise to more complex dynamics. However, the development of a corresponding ODEs model from these rules is a much more difficult task, and the derivation was already quite complicated for our simple rules, as seen in 6.5.1. Instead, we should more thoroughly investigate the discrete model by determining and mathematically proving the limit dynamics when we allow for more directional freedom.

6.7 Figures



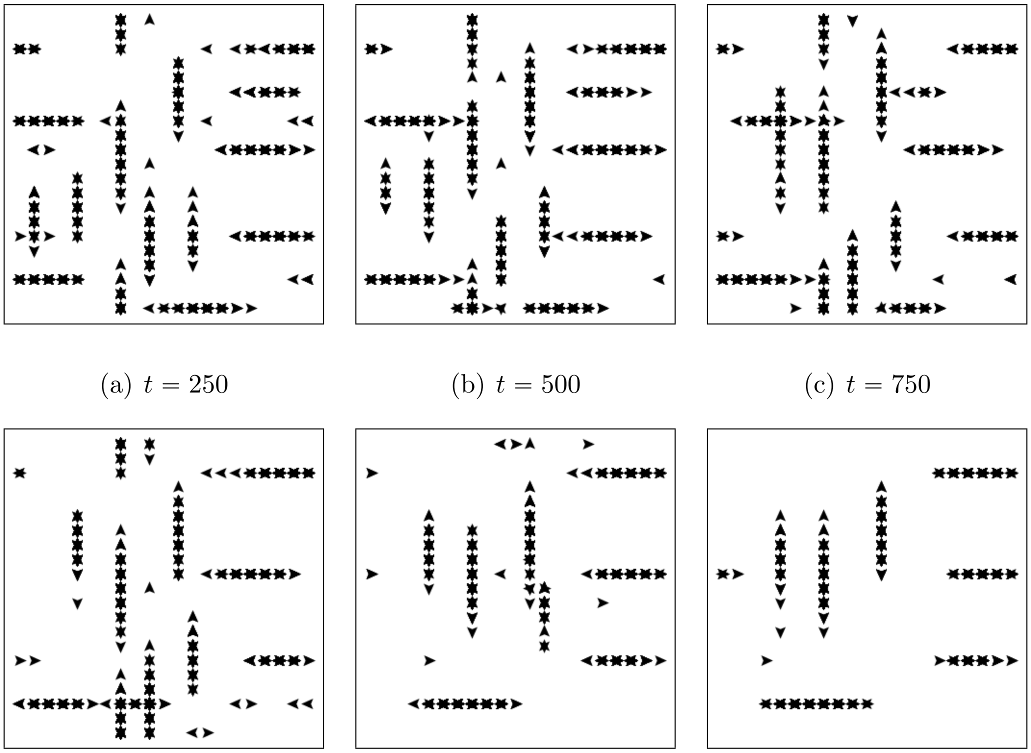
(a) Left-moving particles

(b) Right-moving particles

(c) Upward-moving particles

(d) Downward-moving particles

Figure 6.1: Initial configurations for the discrete model. 4000 particles are uniformly placed on a 21 x 21 grid and are assigned memory (their last direction of motion). The number of particles in every grid-point is shown based on their assigned memory



(a) $t = 250$

(b) $t = 500$

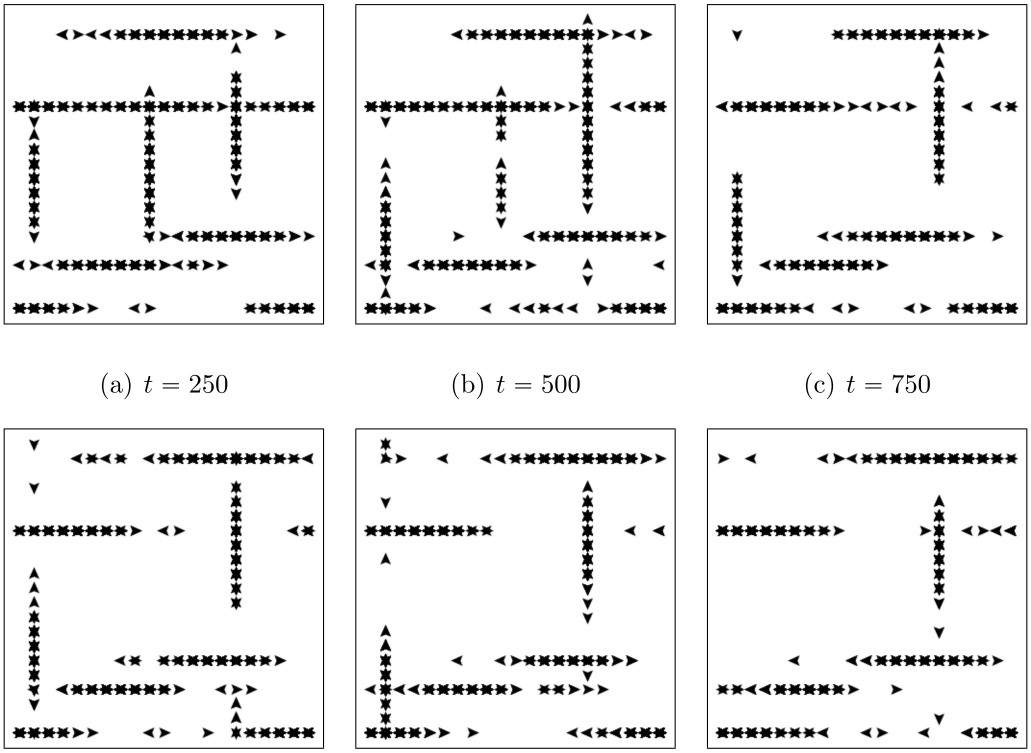
(c) $t = 750$

(d) $t = 1000$

(e) $t = 1250$

(f) $t = 1500$

Figure 6.2: Snapshots of the discrete model with a=0.1, b=0. Initially, 4000 particles are randomly placed on a 21 x 21 grid and given a last direction. The arrows indicate the direction of movement. Each bin may have more than one particle.



(a) $t = 250$

(b) $t = 500$

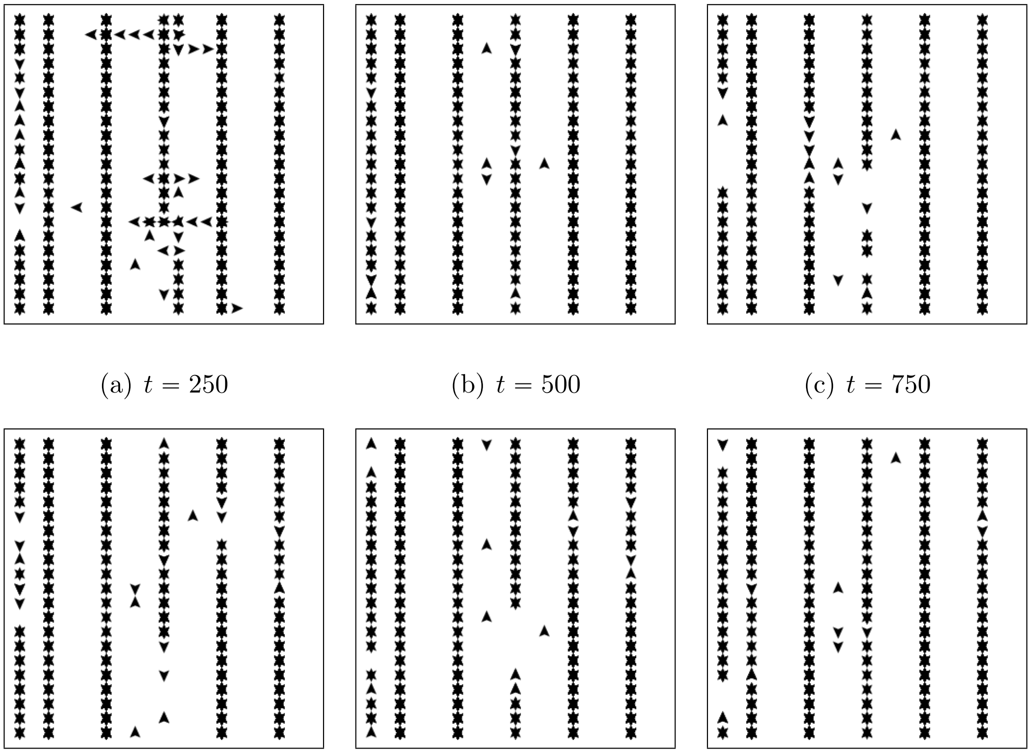
(c) $t = 750$

(d) $t = 1000$

(e) $t = 1250$

(f) $t = 1500$

Figure 6.3: Snapshots of the discrete model with a=0.2, b=0.



(a) $t = 250$

(b) $t = 500$

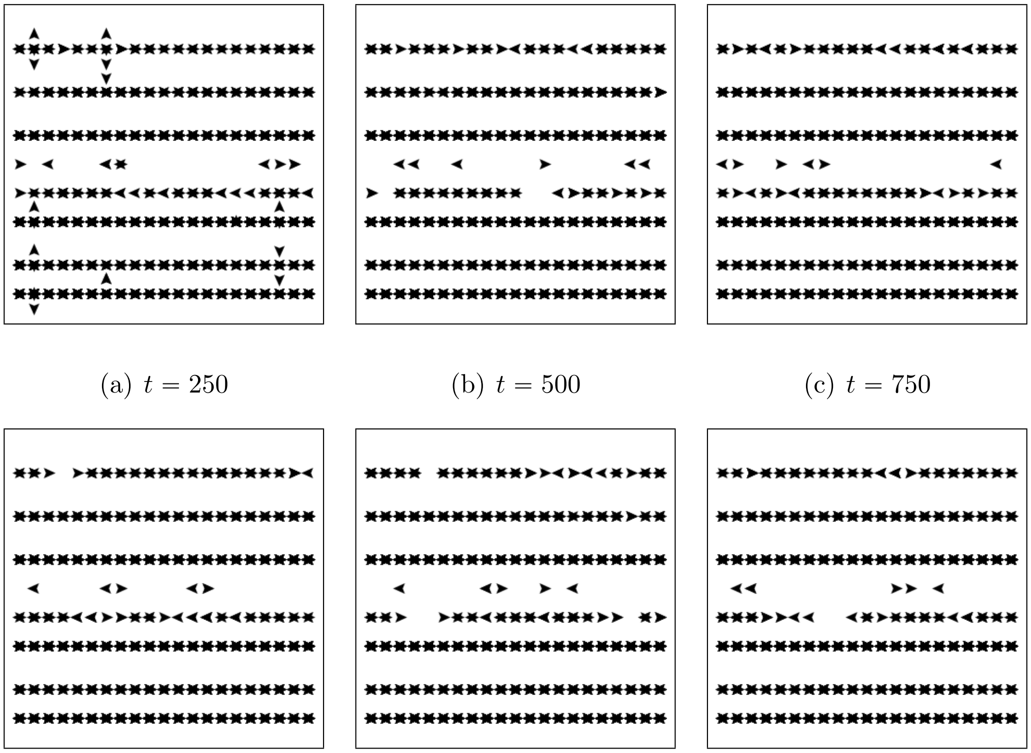
(c) $t = 750$

(d) $t = 1000$

(e) $t = 1250$

(f) $t = 1500$

Figure 6.4: Snapshots of the discrete model with a=0.3, b=0.



(a) $t = 250$

(b) $t = 500$

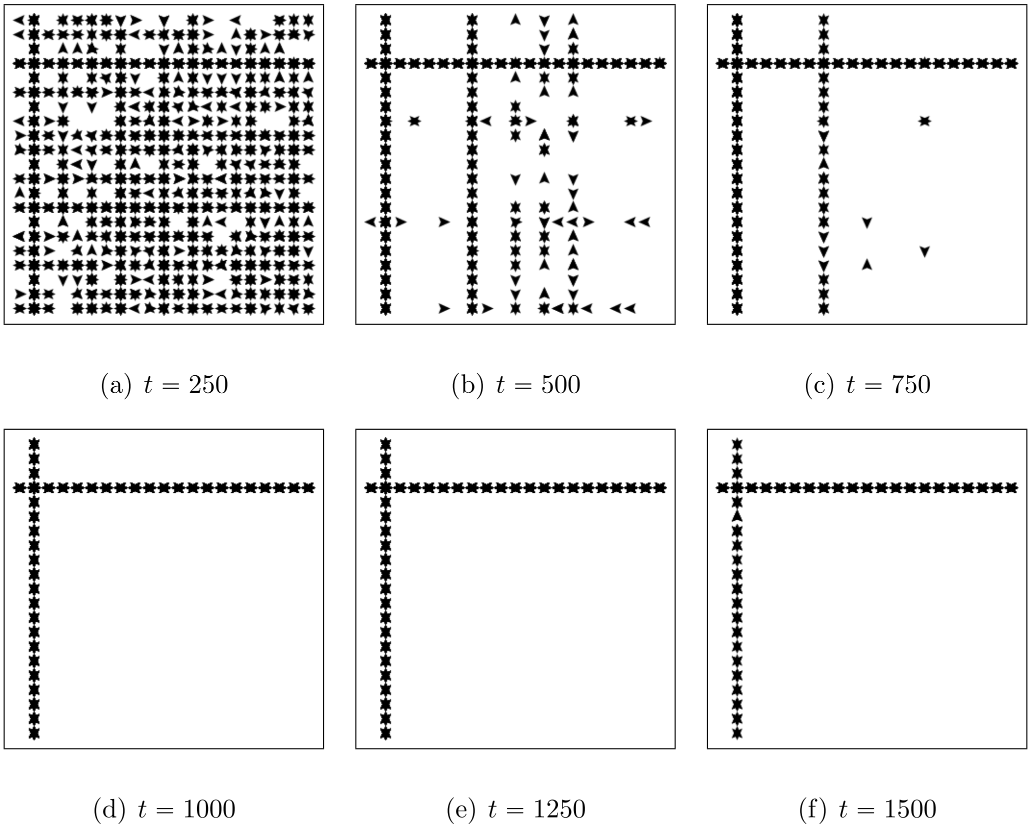
(c) $t = 750$

(d) $t = 1000$

(e) $t = 1250$

(f) $t = 1500$

Figure 6.5: Snapshots of the discrete model with a=0.4, b=0.



(a) $t = 250$

(b) $t = 500$

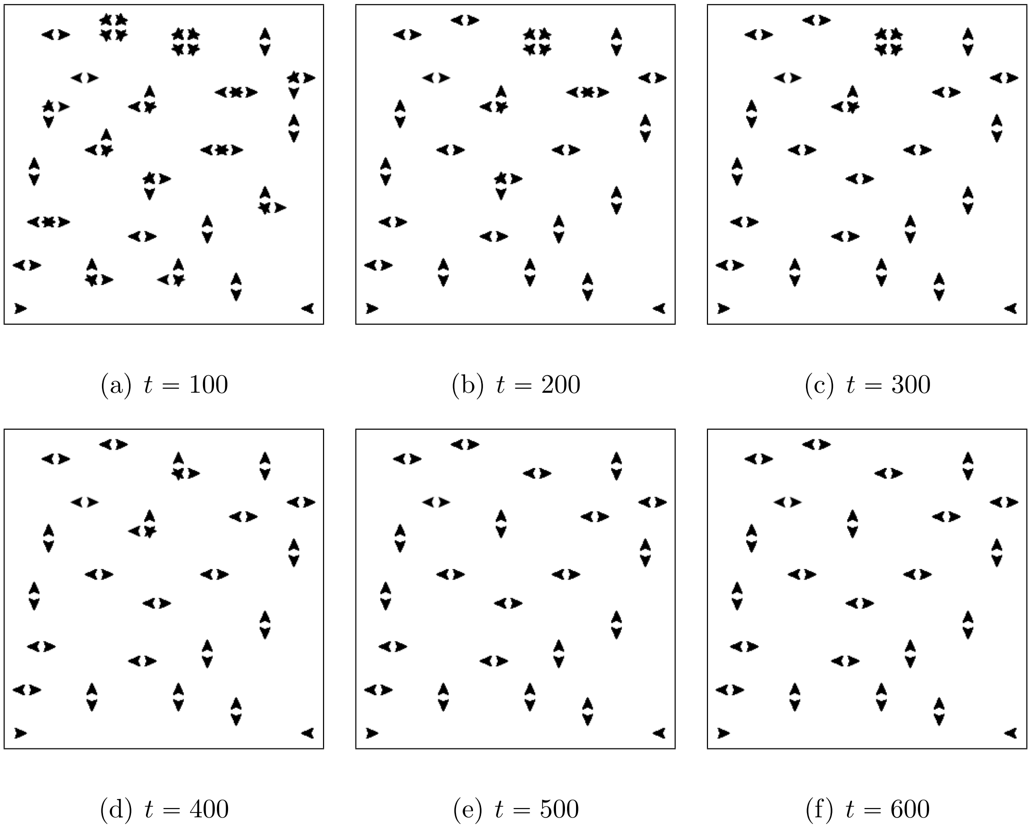
(c) $t = 750$

(d) $t = 1000$

(e) $t = 1250$

(f) $t = 1500$

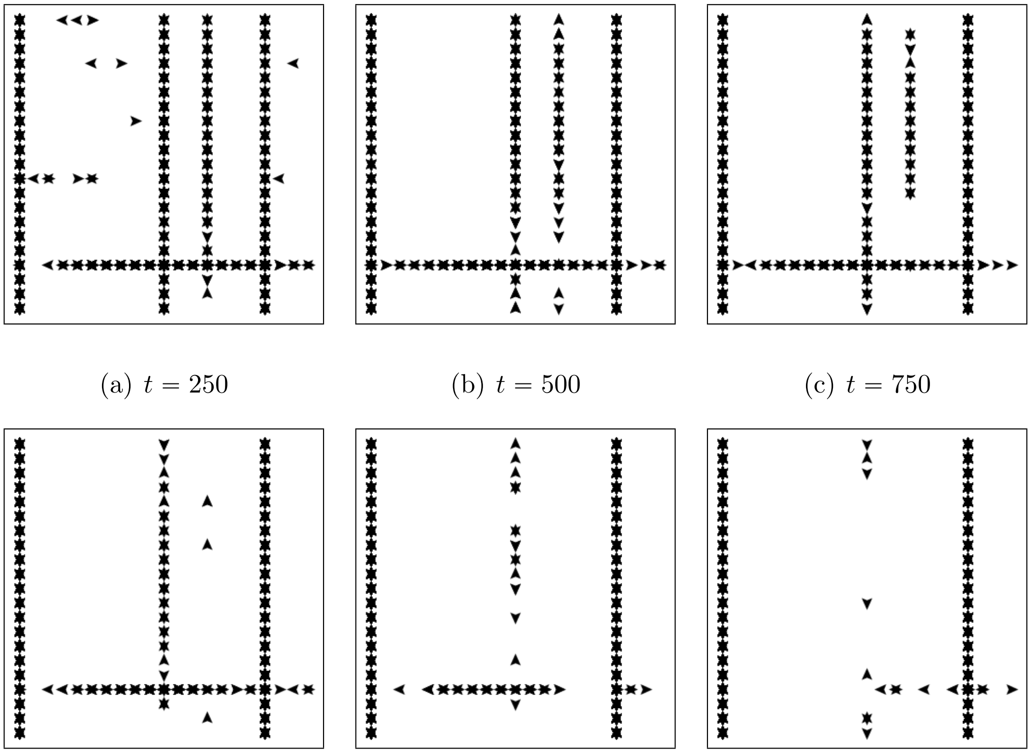
Figure 6.6: Snapshots of the discrete model with a=0.5, b=0.



(a) $t = 100$ (b) $t = 200$ (c) $t = 300$

(d) $t = 400$ (e) $t = 500$ (f) $t = 600$

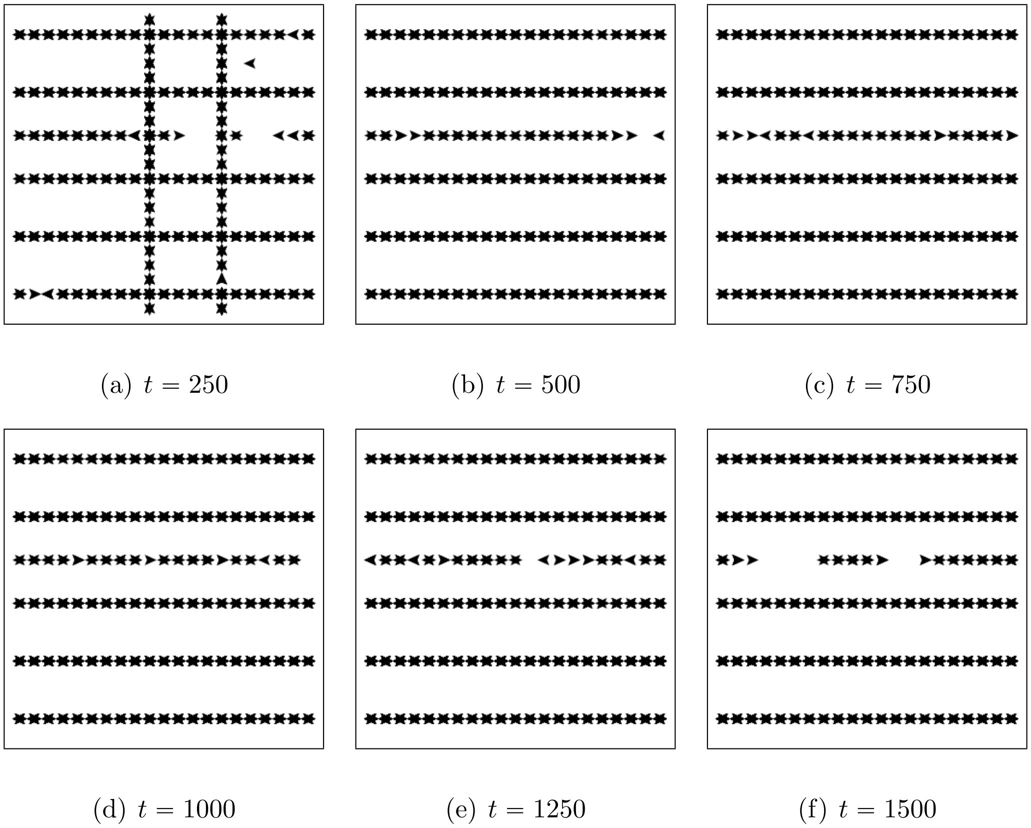
Figure 6.7: Snapshots of the discrete model with a=0, b=0.



(a) $t = 250$ (b) $t = 500$ (c) $t = 750$

(d) $t = 1000$ (e) $t = 1250$ (f) $t = 1500$

Figure 6.8: Snapshots of the discrete model with a=0.3, b=0.1.



(a) $t = 250$

(b) $t = 500$

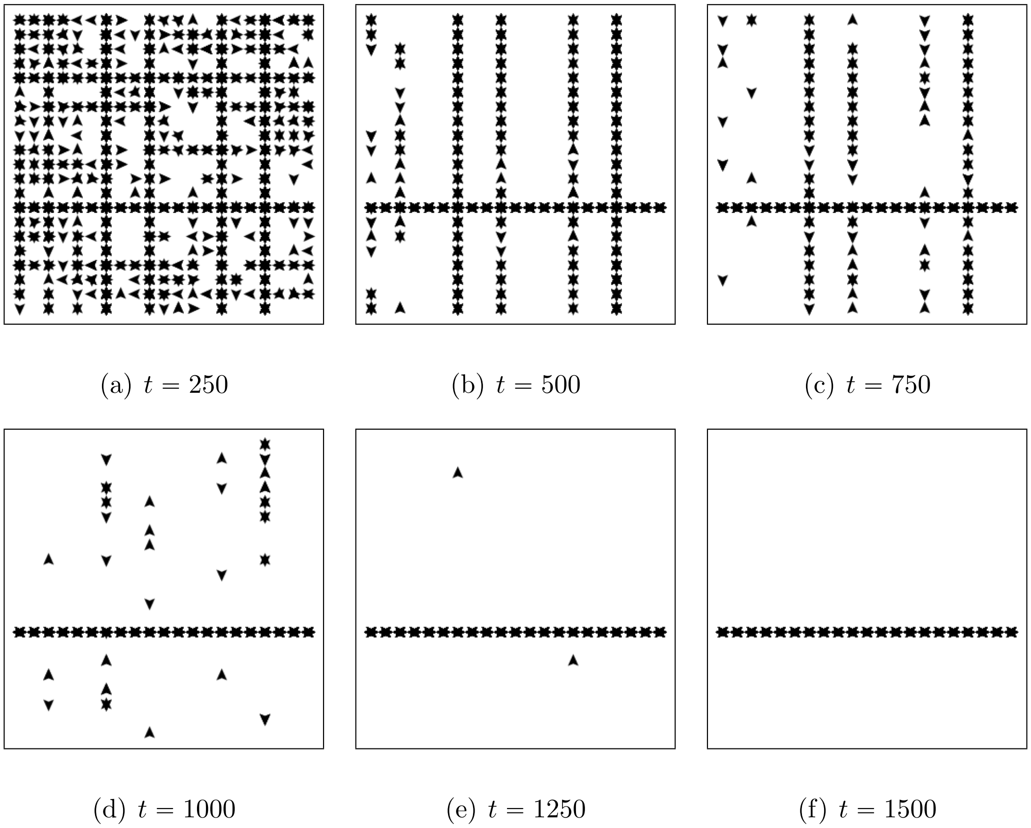
(c) $t = 750$

(d) $t = 1000$

(e) $t = 1250$

(f) $t = 1500$

Figure 6.9: Snapshots of the discrete model with a=0.3, b=0.2.



(a) $t = 250$

(b) $t = 500$

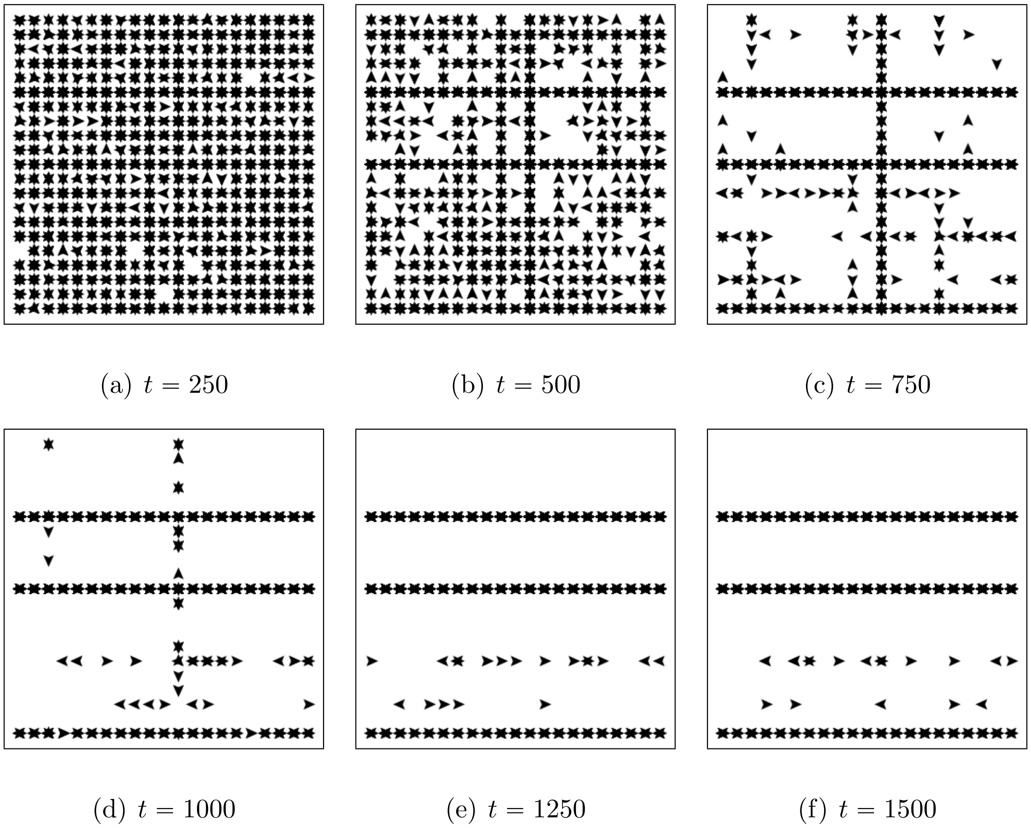
(c) $t = 750$

(d) $t = 1000$

(e) $t = 1250$

(f) $t = 1500$

Figure 6.10: Snapshots of the discrete model with a=0.3, b=0.3.



(a) $t = 250$

(b) $t = 500$

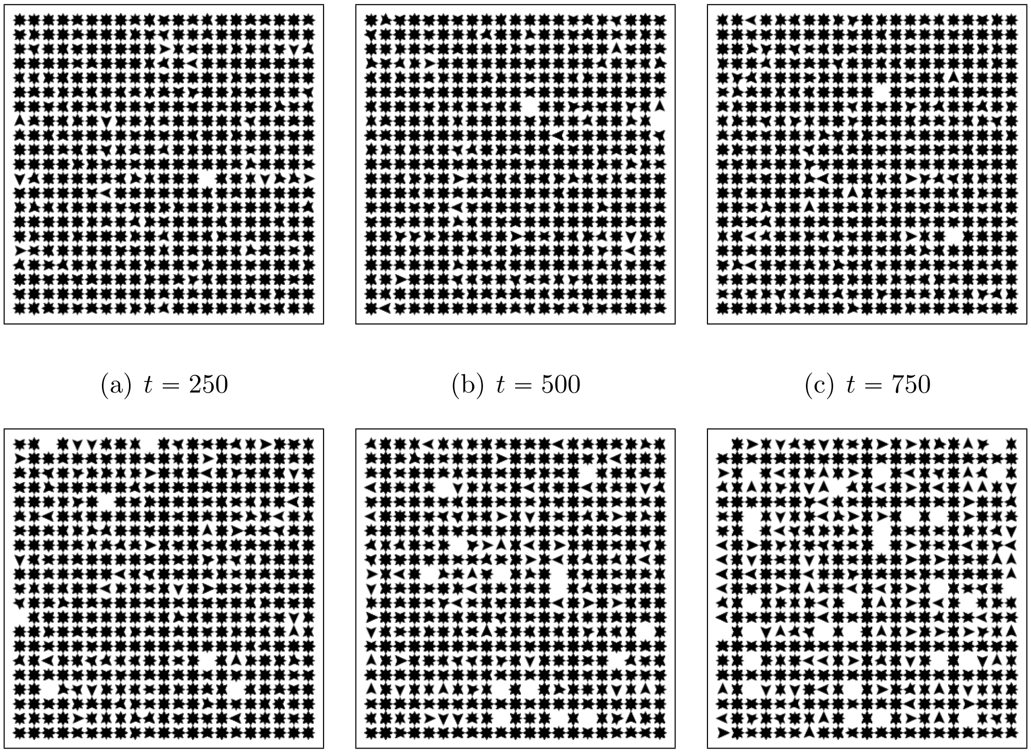
(c) $t = 750$

(d) $t = 1000$

(e) $t = 1250$

(f) $t = 1500$

Figure 6.11: Snapshots of the discrete model with a=0.3, b=0.4.



(a) $t = 250$ (b) $t = 500$ (c) $t = 750$

(d) $t = 1000$ (e) $t = 1250$ (f) $t = 1500$

Figure 6.12: Snapshots of the discrete model with a=0.3, b=0.5.

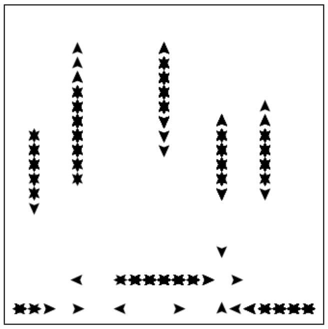


Figure 6.13: The discrete model with $a = 0.1, b = 0$ advanced to $t = 5000$. Both types of aggregates are still observed.

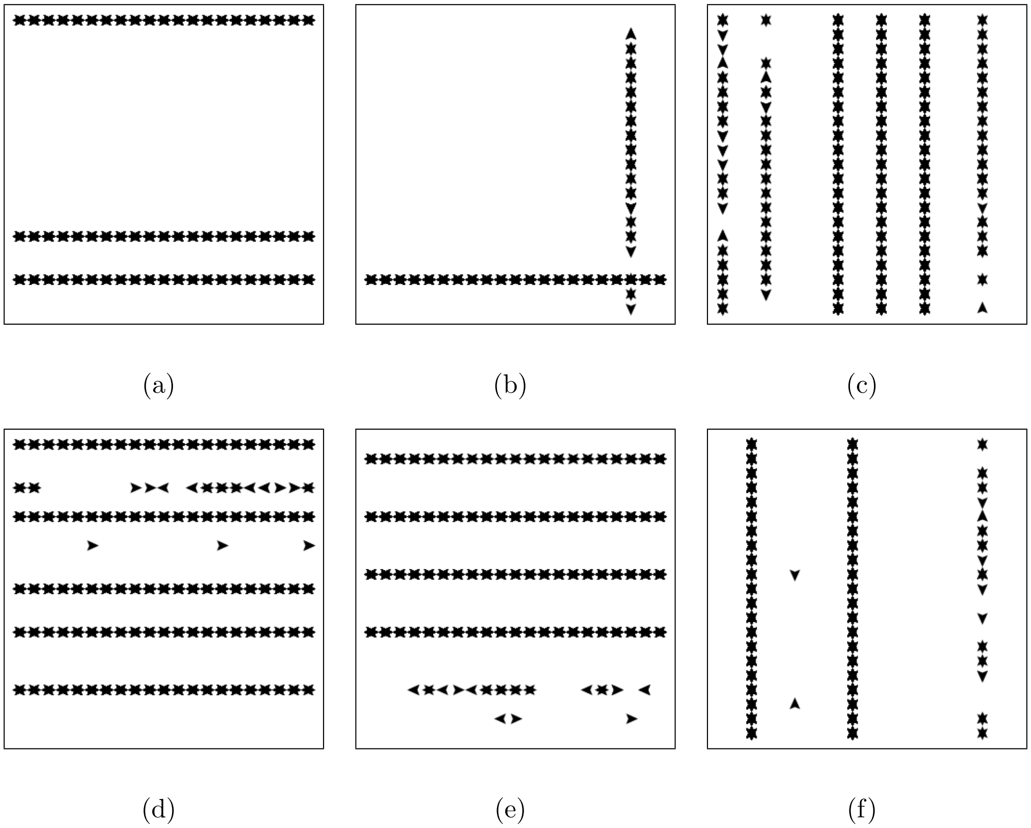
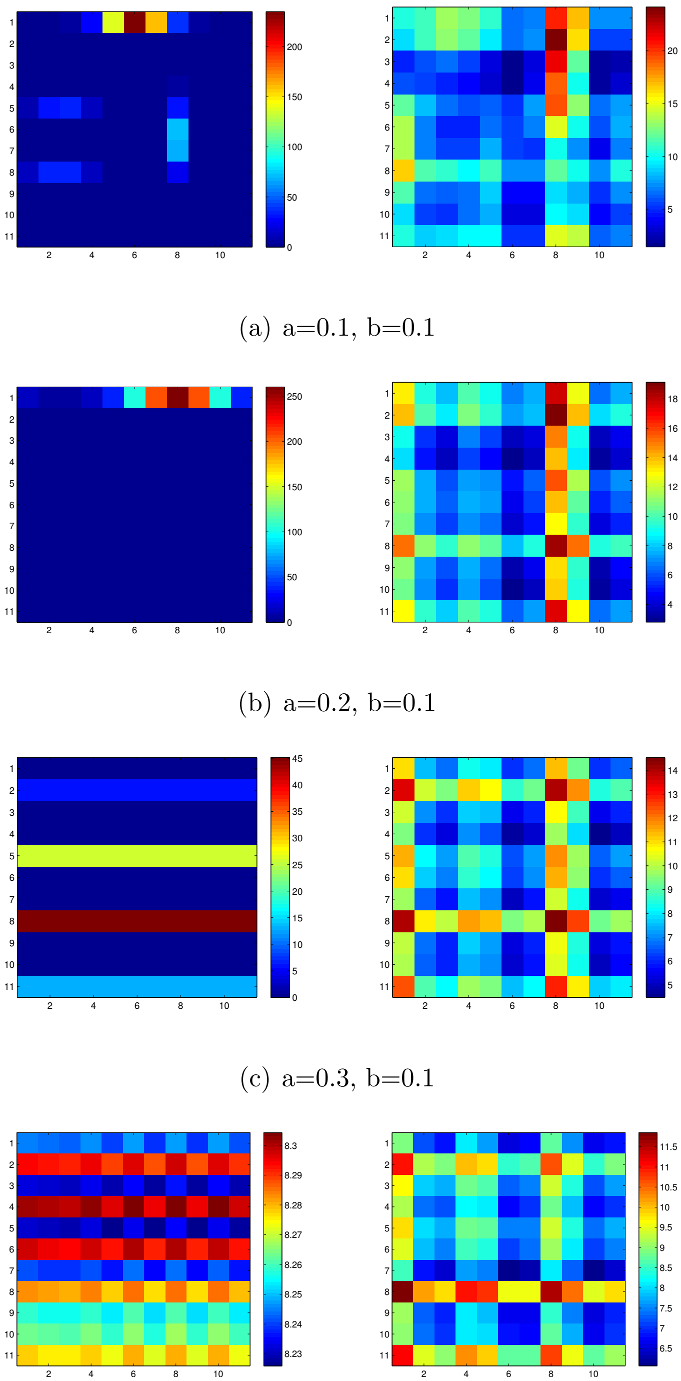


Figure 6.14: Various possible outcomes at t=1500 for the same initial condition from Fig. 6.1. Here, a=0.3, b=0.1.



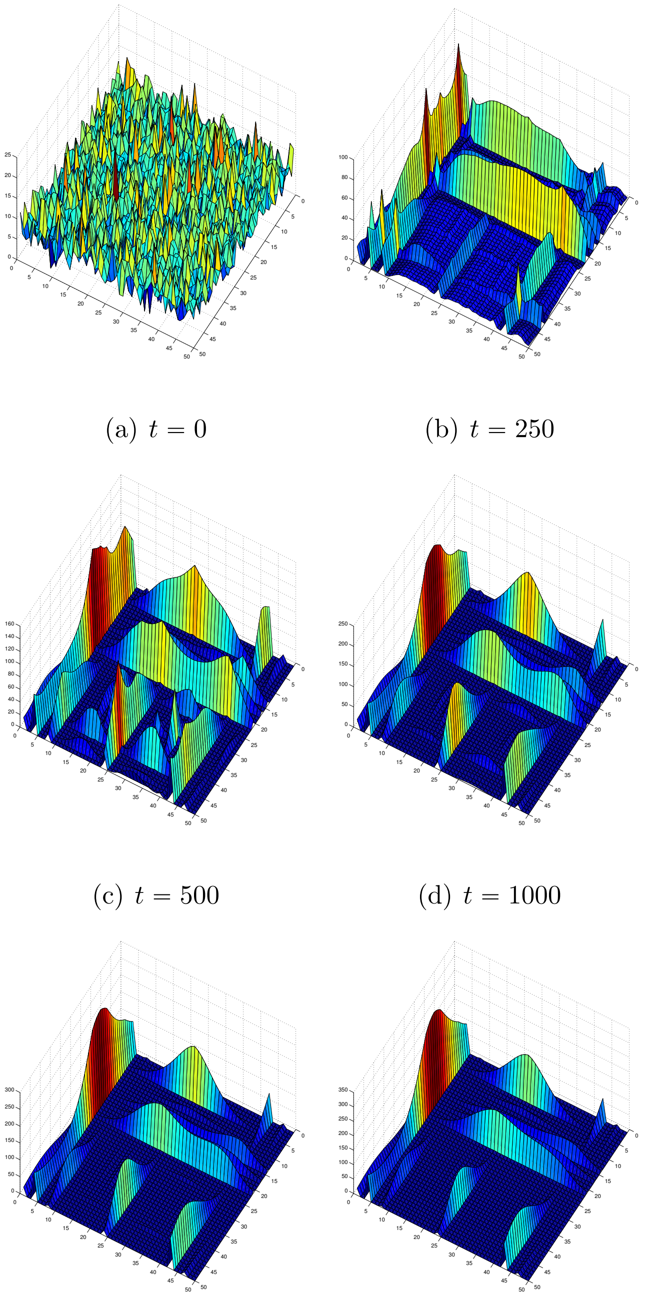
(a) a=0.1, b=0.1

(b) a=0.2, b=0.1

(c) a=0.3, b=0.1

(d) a=0.4, b=0.1

Figure 6.15: Comparisons between the ODE model (left) and the stochastic model averaged over 5000 runs (right)



(a) $t = 0$

(b) $t = 250$

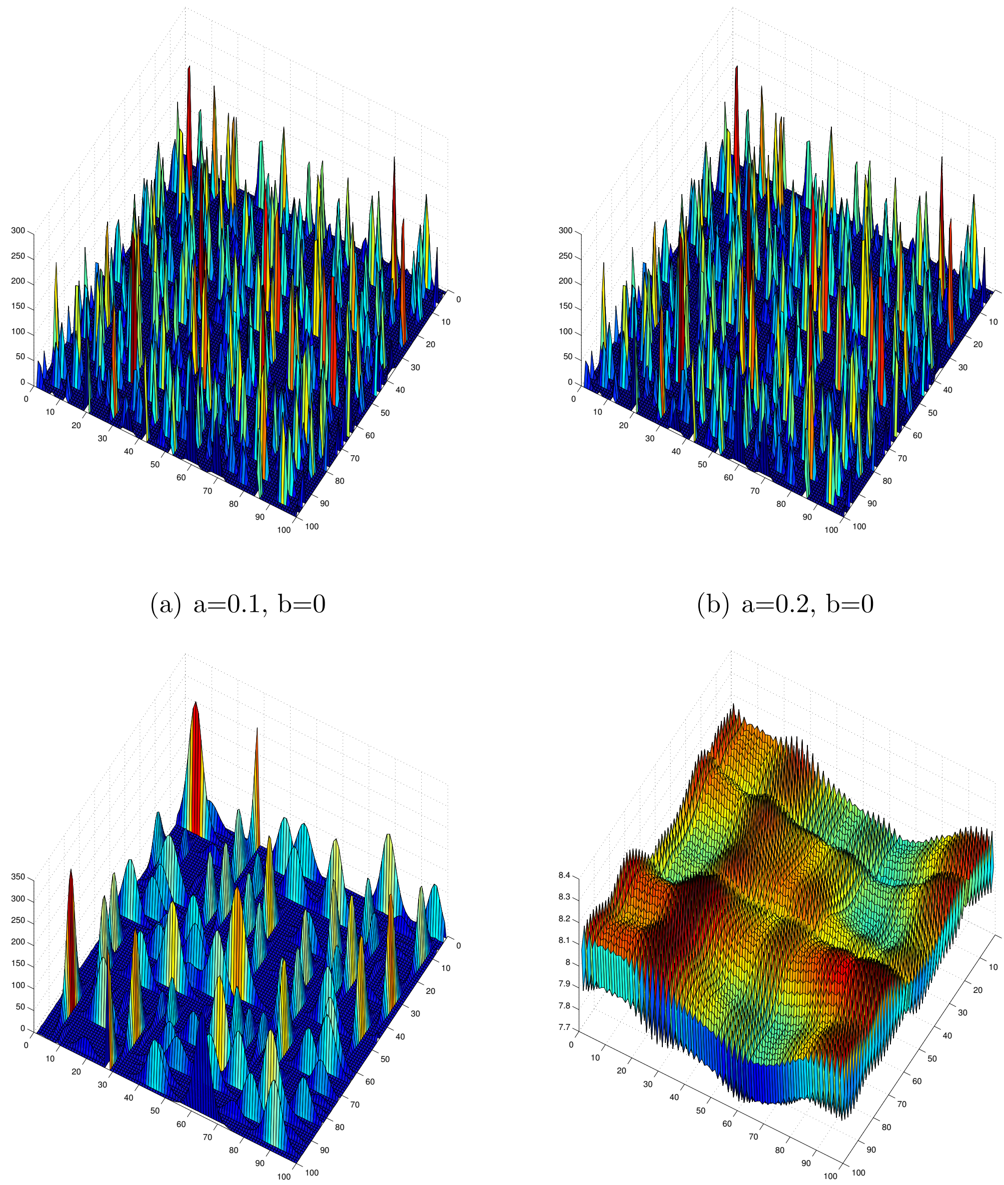
(c) $t = 500$

(d) $t = 1000$

(e) $t = 1500$

(f) $t = 2000$

Figure 6.16: Time Evolution of ODE with a=0.3, b=0 on a 50 x 50 grid.



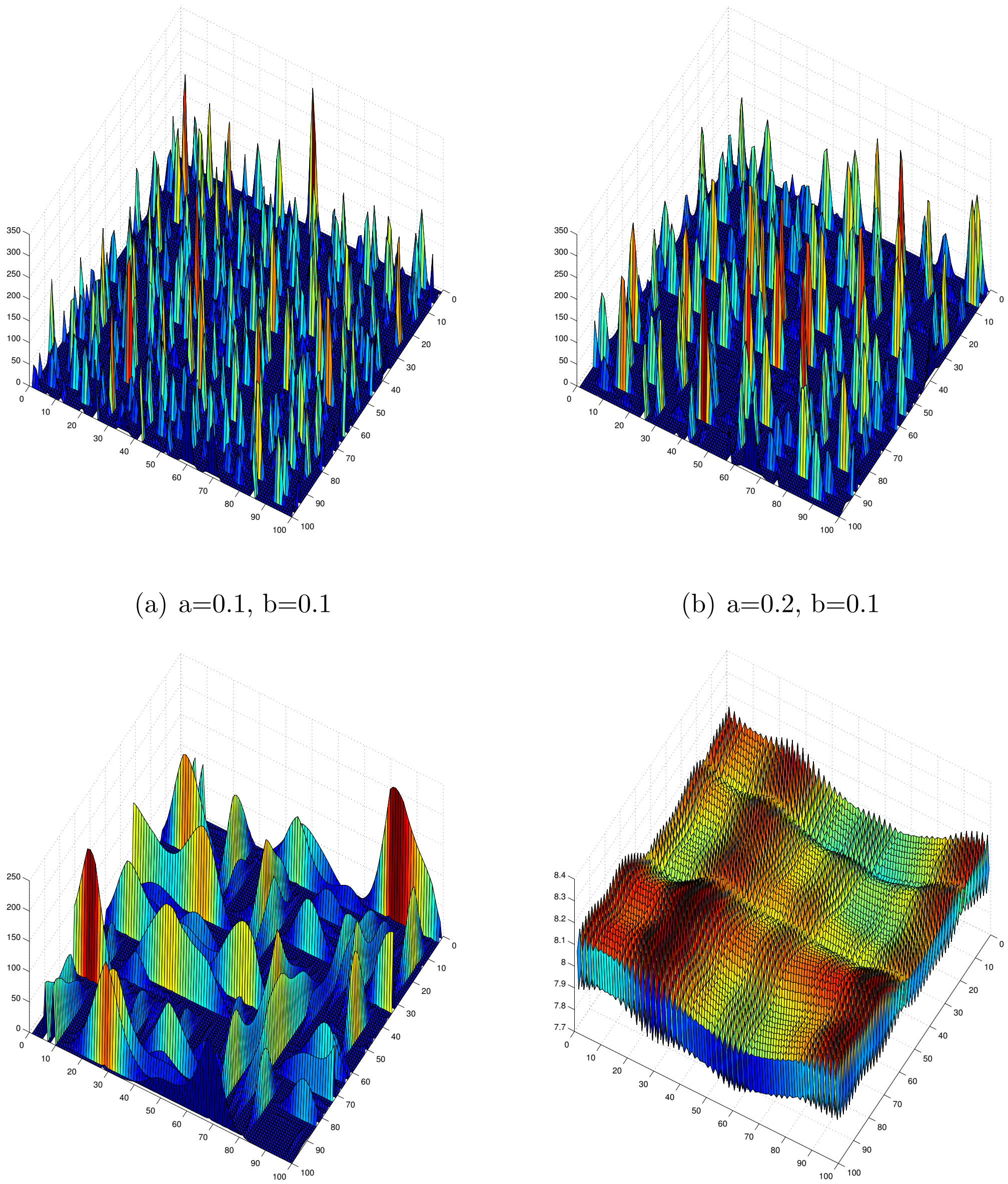
(a) a=0.1, b=0

(b) a=0.2, b=0

(c) a=0.3, b=0

(d) a=0.4, b=0

Figure 6.17: Results from the ODE model at $t = 1000$ for $b = 0$ and $a = 0.1, 0.2, 0.3, 0.4$.



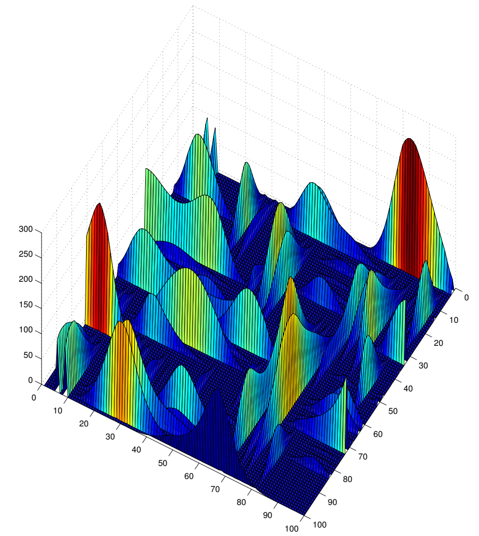
(a) a=0.1, b=0.1

(b) a=0.2, b=0.1

(c) a=0.3, b=0.1

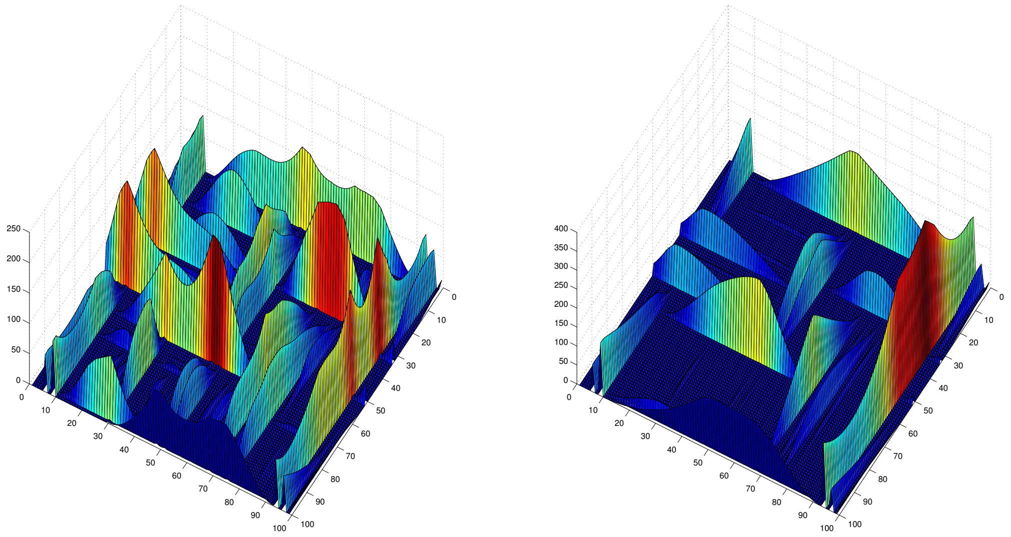
(d) a=0.4, b=0.1

Figure 6.18: Results from the ODE model at $t = 1000$ for $b = 0.1$ and $a = 0.1, 0.2, 0.3, 0.4$.



(a) a=0.3, b=0.1

Figure 6.19: Results from the ODE model at $t = 2000$ for $a = 0.3, b = 0.1$.



(a) t=1000

(b) t=5000

Figure 6.20: Results from the ODE model at $t = 1000$ and 5000 for $a = 0.35, b = 0$

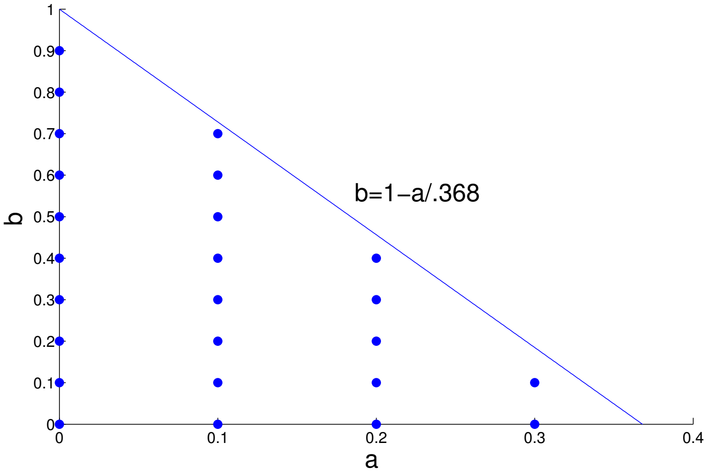
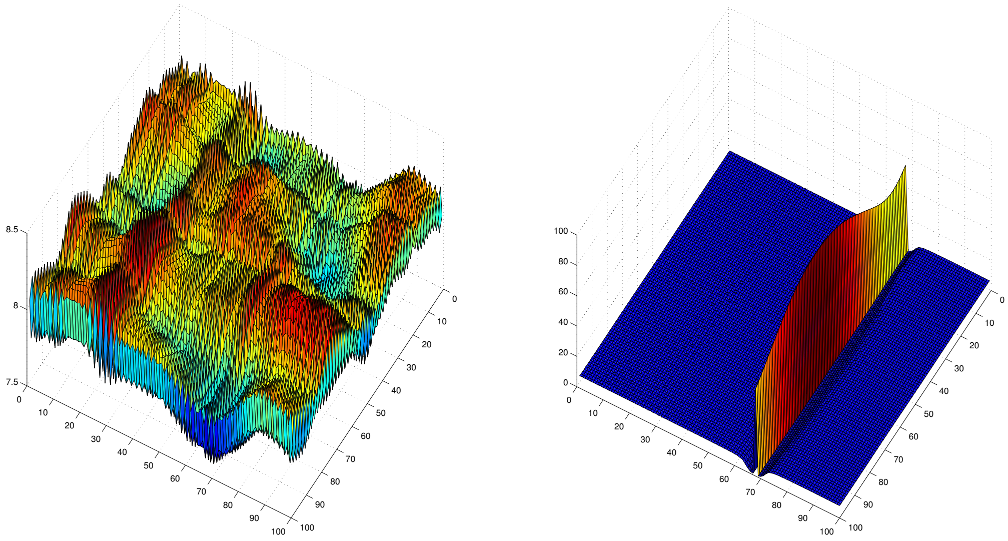


Figure 6.21: The region of parameter space for which aggregates form in the ODE model. The model was run, varying a and b over $\{0, 0.1, 0.2, \ldots, 1\}$. The dark circles represent parameter values for which aggregates form. The constraint line is provided for reference.



(a) a=0.3,b=0.19

(b) a=0.3,b=0.182

Figure 6.22: Demonstration that the constraint accurately predicts the lack of aggregates for $(a, b) = (0.3, 0.19)$ and the existence of aggregates for $(0.3, 0.182)$.

Bibliography

[1] W. Czaja, A. Hafftka, B. Manning, and D. Weinberg. Randomized approximations of operators and their spectral decomposition for diffusion based embeddings of heterogeneous data. *Proceedings of CoSeRa*, 2015.

[2] W. Czaja, K. Kochersberger, B. Manning, T. McCullough, M. McLane, J. Murphy, and D. Weinberg. Spectral decomposition of data-driven operators for nuclear anomaly detection, in prep.

[3] Wojciech Czaja, JM Murphy, and D Weinberg. Superresolution of remotely sensed images with anisotropic features. *Proceedings of SAMPTA*, 2015.

[4] Edward H Bosch, Wojciech Czaja, James M Murphy, and Daniel Weinberg. Anisotropic representations for superresolution of hyperspectral data. In *SPIE Defense+ Security*, pages 947213–947213. International Society for Optics and Photonics, 2015.

[5] D. Weinberg and D. Levy. Modeling selective local interactions with memory: Motion on a 2D lattice. *Physica D*, 278-279:13–30, 2014.

[6] Anton Deitmar and Siegfried Echterhoff. *Principles of harmonic analysis*. Springer, 2014.

[7] Richard J Duffin and Albert C Schaeffer. A class of nonharmonic Fourier series. *Transactions of the American Mathematical Society*, pages 341–366, 1952.

[8] P. Casazza, G. Kutyniok, and F. Philipp. Introduction to finite frame theory. In *Finite Frames*. Springer, 2013.

[9] G. Kutyniok and D. Labate. *Shearlets: Multiscale Analysis for Multivariate Data*, chapter Introduction to Shearlets, pages 1–38. Applied and Numerical Harmonic Analysis. Birkhäuser Boston, 2012.

[10] R. Laugesen, N. Weaver, G. Weiss, and E. Wilson. A characterization of the higher dimensional groups associated with continuous wavelets. *J. Geom. Anal.*, 12(1):89–102, 2002.

[11] K. Guo, G. Kutyniok, and D. Labate. Sparse multidimensional representations using anisotropic dilation and shear operators. In *Wavelets and splines: Athens 2005*, Mod. Methods Math., pages 189–201. Nashboro Press, Brentwood, TN, 2006.

[12] S. Häuser and G. Steidl. Fast finite shearlet transform: a tutorial. *arXiv preprint arXiv:1202.1773*, 2014.

[13] Kanghui Guo, Demetrio Labate, Wang-Q Lim, Guido Weiss, and Edward Wilson. Wavelets with composite dilations. *Electronic research announcements of the American Mathematical Society*, 10(9):78–87, 2004.

[14] E. Candès and D. Donoho. Curvelets: A surprisingly effective nonadaptive represenatation for objects with edges. In A. Cohen and C. Rabut, editors, *Curve and Surface Fitting*, pages 105–120. Vanderbilt Univerity Press, 2000.

[15] S. Mallat. *A Wavelet Tour of Signal Processing.* Wavelet Analysis & Its Applications. Academic Press, 2nd edition, 1999.

[16] E. J. Candès and D. L. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise C2 singularities. *Comm. Pure and Appl. Math*, 56(2):216–266, 2004.

[17] K. Guo and D. Labate. Optimally sparse multidimensional representation using shearlets. *SIAM J. Math Anal.*, 39(1):298–318, 2007.

[18] G. Kutyniok and W.-Q Lim. Compactly supported shearlets are optimally sparse. *J. Approx. Theory*, 163(11):1564–1589, 2011.

[19] K. Guo and D. Labate. Optimally sparse 3D approximations using shearlet representations. *Electronic Research Announcements in Mathematical Sciences*, 17:126–138, 2010.

[20] K. Guo and D. Labate. Optimally sparse representations of 3D data with C2 surface singularities using Parseval frames of shearlets.3d data with C2 surface singularities using Parseval frames of shearlets. *SIAM J Math. Anal.*, 44(2):851–886, 2012.

[21] G. Kutyniok, J. Lemvig, and W.-Q Lim. Optimally sparse approximations of 3D functions by compactly supported shearlet frames. *SIAM J. Math. Anal.*, 44(4):2962–3017, 2012.

[22] Wojciech Czaja and Emily J King. Isotropic shearlet analogs for L2(Rk) and localization operators. *Numerical functional analysis and optimization*, 33(7-9):872–905, 2012.

[23] Wojciech Czaja and Emily J King. Anisotropic shearlet transforms for L2(Rk). *Mathematische Nachrichten*, 287(8-9):903–916, 2014.

[24] G. Easley, D. Labate, and W. Lim. Sparse directional image representations using the discrete shearlet transform. *Applied and Computational Harmonic Analysis*, 25(1):25–46, 2008.

[25] P. Negi and D. Labate. 3-D discrete shearlet transform and video processing. *IEEE Transactions on Image processing*, 21(6):2944–2954, 2012.

[26] G. Kutyniok, M. Shahram, and X. Zhuang. Shearlab: A rational design of a digital parabolic scaling algorithm. *SIAM J. Imaging Sci.*, 5(4):1291–1332, 2012.

[27] G. Kutyniok, W.-Q Lim, and R. Reisenhofer. Shearlab 3D: Faithful digital shearlet transforms based on compactly supported shearlets. *arXiv preprint arXiv:1402.5670*, 2014.

[28] S. Häuser and G. Steidl. Convex multiclass segmentation with shearlet regularization. *International Journal of Computer Mathematics*, 90(1):62–81, 2013.

[29] James M Murphy. *Anisotropic Harmonic Analysis and Integration of Remotely Sensed Data*. PhD thesis, University of Maryland, 2015.

[30] Emily J King, Gitta Kutyniok, and Wang-Q Lim. Image inpainting: theoretical analysis and comparison of algorithms. In *SPIE Optical Engineering+ Applications*, pages 885802–885802. International Society for Optics and Photonics, 2013.

[31] Emily J King, Gitta Kutyniok, and Xiaosheng Zhuang. Analysis of inpainting via clustered sparsity and microlocal analysis. *Journal of mathematical imaging and vision*, 48(2):205–234, 2014.

[32] Glenn R Easley, Demetrio Labate, and Flavia Colonna. Shearlet-based total variation diffusion for denoising. *Image Processing, IEEE Transactions on*, 18(2):260–268, 2009.

[33] David Donoho and Gitta Kutyniok. Geometric separation using a wavelet-shearlet dictionary. In *SAMPTA'09*, 2009.

[34] Gitta Kutyniok and Wang-Q Lim. Image separation using wavelets and shearlets. In *Curves and surfaces*, pages 416–430. Springer, 2012.

[35] S Hauser and Jianwei Ma. Seismic data reconstruction via shearlet-regularized directional inpainting, 2012.

[36] H. Lakshman, W.-Q Lim, H. Schwarz, D. Marpe, G. Kutyniok, and T. Wiegand. Image interpolation using shearlet based interative refinement. *arXiv preprint arXiv:1308.1126*, 2013.

[37] S. Yi, D. Labate, G. Easley, and H. Krim. A shearlet approach to edge analysis and detection. *IEEE Transactions on Image processing*, 18(5):929–941, 2009.

[38] Gitta Kutyniok and Philipp Petersen. Classification of edges using compactly supported shearlets. *arXiv preprint arXiv:1411.5657*, 2014.

[39] M. Lang, H. Guo, J.E. Odegard, C.S. Burrus, and R.O. Wells Jr. Noise reduction using an undecimated discrete wavelet transform. *Signal Process. Newslett.*, 3(1):10–13, 1996.

[40] S. Mallat and S. Zhong. Characterization of signals from multi-scale edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(7):710–732, 1992.

[41] M. N. Do and M. Vetterli. *Contourlets*, volume 10 of *Studies in Computational Mathematics*, pages 1–27. Academic Press/Elsevier Science, San Diego, CA, 2003.

[42] E. J. Candés and F. Guo. New multiscale transforms, minimum total variation synthesis: applications to edge-preserving image reconstruction. *Signal Processing*, 82(11):1519–1543, 2002.

[43] E. L. Pennec and S. Mallat. Bandelet image approximation and compression. *SIAM Journal of Multiscale Modeling and Simulation*, 4(3):992–1039, 2005.

[44] D. L. Donoho. Wedgelets: nearly minimax estimation of edges. *Ann. Statist.*, 27(3):859–897, 1999.

[45] ArcGIS Resources. What is lidar data? http://resources.arcgis.com/en/help/main/10.1/index.html#/What_is_lidar/015w00000041000000/, June 2013.

[46] Mike Ramsey. Quanergy says it will offer low cost sensors for autonomous cars next year. blogs.wsj.com/digits/2015/04/21/quanergy-says-it-will-offer-low-cost-sensors-for-autonomous-cars-next-year/, April 2015.

[47] Ruzena Bajcsy and Mohamad Tavakoli. Computer recognition of roads from satellite pictures. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-6(9):623–637, 1976.

[48] Simon Clode, Franz Rottensteiner, Peter Kootsookos, and Emanuel Zelniker. Detection and vectorization of roads from lidar data. *Photogrammetric Engineering & Remote Sensing*, 73(5):517–535, 2007.

[49] Xiangyun Hu, C Vincent Tao, and Yong Hu. Automatic road extraction from dense urban area by integrated processing of high resolution imagery and lidar data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences. Istanbul, Turkey*, 35:B3, 2004.

[50] Yun-Woong Choi, Young Woon Jang, Hyo Jong Lee, and Gi-Sung Cho. Heuristic road extraction. In *Information Technology Convergence, 2007. ISITC 2007. International Symposium on*, pages 338–342. IEEE, 2007.

[51] Kevin Peterson, Jason Ziglar, and Paul E Rybski. Fast feature detection and stochastic parameter estimation of road shape using multiple lidar. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 612–619. IEEE, 2008.

[52] W. Zhang. LIDAR-based road and road-edge detection. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 845–848. IEEE, 2010.

[53] W Rieger, M Kerschner, T Reiter, and F Rottensteiner. Roads and buildings from laser scanner data within a forest enterprise. *International Archives of Photogrammetry and Remote Sensing*, 32(Part 3):W14, 1999.

[54] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(6):679–698, 1986.

[55] Albert Baumgartner and H Hinz. Multi-scale road extraction using local and global grouping criteria. *International Archives of Photogrammetry and Remote Sensing*, 33(B3/1; PART 3):58–65, 2000.

[56] Mihran Tuceryan and Anil Jain. Texture analysis. In *Handbook of Pattern Recognition & Computer Vision*. World Scientific Publishing Company Incorporated, 1999.

[57] Andy Tsai, Anthony Yezzi Jr, William Wells, Clare Tempany, Dewey Tucker, Ayres Fan, W Eric Grimson, and Alan Willsky. A shape-based approach to the segmentation of medical imagery using level sets. *Medical Imaging, IEEE Transactions on*, 22(2):137–154, 2003.

[58] Daniel Cremers, Mikael Rousson, and Rachid Deriche. A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. *International journal of computer vision*, 72(2):195–215, 2007.

[59] Daniel Freedman and Tao Zhang. Interactive graph cut based segmentation with shape priors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

[60] Richard O Duda and Peter E Hart. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.

[61] RW Gerchberg. Super-resolution through error energy reduction. *Journal of Modern Optics*, 21(9):709–720, 1974.

[62] S. Park, M. Park, and M. Kang. Super-resolution image reconstruction: a technical overview. *IEEE Signal Processing Magazine*, 20(3):21–36, 2003.

[63] A. Marquina and S. Osher. Image super-resolution by TV-regularization and Bregman iteration. *J. Sci. Comp.*, 37(3):367–382, 2008.

[64] W. Freeman, T. Jones, and E. Pasztor. Example-based super-resolution. *Computer graphics and applications*, 22(3):56–65, 2002.

[65] R. Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech and Signal processing*, 29(6):1153–1160, 1981.

[66] X. Li and M. Orchard. New edge-directed interpolation. *IEEE Transactions on IMage processing*, 10(10):1521–1527, 2001.

[67] L. Zhang and X. Wu. An edge-guided image interpolation algorithm via directional filtering and data fusion. *IEEE Transactions on Image processing*, 15(8):2226–2238, 2006.

[68] X. Zhang and X. Wu. Image interpolation by adaptive 2-D autoregressive modeling and soft-decision estimation. *IEEE Trans. Image Process.*, 17(6):887–896, 2008.

[69] H. Takeda, S. Farsiu, and P. Milanfar. Kernel regression for image processing and reconstruction. *IEEE Trans. Image Process.*, 16(2):349–366, 2007.

[70] N. Mueller, Y. Lu, and M. Do. Image interpolation using multiscale geometric representations. In *Proc. SPIE Conf. on Electronic Imaging*, San Jose, USA, 2007.

[71] S. Mallat and G. Yu. Super-resolution with sparse mixing estimators. *IEEE Trans. Image Process.*, 19(11):2889–2900, 2010.

[72] Edward H Bosch, Alexey Castrodad, John S Cooper, Wojtek Czaja, and Julia Dobrosotskaya. Tight frames for multiscale and multidirectional image analysis. In *SPIE Defense, Security, and Sensing*, pages 875004–875004. International Society for Optics and Photonics, 2013.

[73] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *International journal of computer vision*, 43(1):7–27, 2001.

[74] Ido Zachevsky and Yehoshua Y Zeevi. Single-image superresolution of self-similar textures. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 952–956. IEEE, 2013.

[75] David J Field. Relations between the statistics of natural images and the response properties of cortical cells. *JOSA A*, 4(12):2379–2394, 1987.

[76] A. Galante and D. Levy. Modeling selective local interactions with memory. *Physica D*, 260:176–190, 2012.

[77] A. Galante, S. Wisen, D. Bhaya, and D. Levy. Modeling local interactions during the motion of cyanobacteria. *J. Theoret. Biol.*, 309:147–158, 2012.

[78] A. Galante, S. Wisen, D. Bhaya, and D. Levy. Stochastic models and simulations of phototaxis. In H. Sayama, A.A. Minai, D. Braha, and Y. Bar-Yam, editors, *Unifying Themes in Complex Systems Volume VIII: Proceedings of the Eighth International Conference on Complex Systems*, volume 8 of *New England Complex Systems Institute Series on Complexity*, pages 105–119, Boston, MA, 2011. NECSI Knowledge Press.

[79] S.-Y. Ha and D. Levy. Particle, kinetic, and fluid models for phototaxis. *Disc. Cont. Dynam. Syst. B*, 12(1):77–108, 2009.

[80] D. Levy and T. Requeijo. Modeling group dynamics of phototaxis: from particles to PDEs. *Disc. Cont. Dynam. Syst. B*, 9:108–128, 2008.

[81] D. Levy and T. Requeijo. Stochastic models for phototaxis. *Bull. Math. Biol.*, 70(6):1684–1706, 2008.

[82] D. Bhaya. Light matters: phototaxis and signal transduction in unicellular cyanobacteria. *Molecul. Microbiol.*, 53(3):745–754, 2004.

[83] M. Burriesci and D. Bhaya. Tracking phototactic responses and modeling motility of synechnocystis sp. strain pcc6803. *J. Photochem. Photobiol. B*, 91(2-3):77–86, 2008.

[84] R.E. Baker, C.A. Yates, and R. Erban. From microscopic to macroscopic descriptions of cell migration on growing domains,. *Bull. Math. Biol.*, 72(3):719–762, 2010.

[85] I. Aoki. A simulation study on the schooling mechanism in fish,. *Bull. Japan Soc. Sci. Fisher.*, 48(8):1081–1088, 1982.

[86] T. Oboshi, S. Kato, A. Mutoh, and H. Itoh. A simulation study on the form of fish schooling for escape from predator. *Forma*, 18, 2003.

[87] J.K. Parrisha, S.V. Viscido, and D. Grunbaum. Self-organized fish schools: an examination of emergent properties. *Biol. Bull.*, 202:296–305, 2002.

[88] S. Stöcker. Models for tuna school formation. *Math. Biosci.*, 156(1-2):167–190, 1999.

[89] F. Cucker and S. Smale. Emergent behavior in flocks. *IEEE Trans. Automat. Control*, 52(5):852–862, 2007.

[90] C. W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. In *Proc. Comput. Graphics*, pages 25–34. ACMSIGGRAPH, 1987.

[91] T. Ikawa and H. Okabe. Three-dimensional measurements of swarming mosquitoes: a probabilistic model, measuring system, and example results. In *Animal groups in three dimensions*. Cambridge, 1997.

[92] A. Okubo and H.C. Chiang. An analysis of the kinematics of swarming of anarete pritchardi kim (diptera: cecidomyiidae). *Res. Popul. Ecol.*, 16:1–42, 1974.

[93] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet. Novel type of phase transition in a system of self-driven particles. *Phys. Rev. Lett.*, 6:1226–1229, 1995.

[94] I.D. Couzin, J. Krause, R. James, G.D. Ruxton, and N.R. Franks. Collective memory and spatial sorting in animal groups,. *J. Theoret. Biol.*, 218(1):1–11, 2002.