New Methods for Measuring Interpretable Features in Images

By

BRIAN KNIGHT
DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Applied Mathematics

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Naoki Saito, Chair

Zhi Ding

Carson Jeffres

Committee in Charge

2025

To the french press in the kitchen.

Contents

Abstract

Computer vision is a rapidly growing area of study due to the development and advancement of computational tools in the last 30-40 years. By now, a wide variety of mathematical models and tools have been developed in order to handle large amounts of image data, yet every individual task still requires specific domain knowledge and new algorithm development. In this dissertation several feature extraction problems are studied, and image processing and image analysis methods are developed in order to provide solutions to these. The problems studied here include fine-scale fingerprint registration, weight prediction of juvenile fish from raw images, explainable image classification and texture segmentation, and 3D image registration as it relates to reproductive cell development in fruit flies. For each of these problems, the new solutions proposed rely heavily on ongoing developments in the field of computer vision, thus we begin by providing the necessary background for the reader.

First, a new spatial phase estimate from single-shot fringe pattern images is described. This estimate is useful for analyzing images like fingerprint patterns or images seen in fringe projection profilometry. The analytic signal, and the *split of identity* that it provides, separates local amplitude, phase, and orientation information, which can be used to analyze important local features. Our algorithm improves existing single-shot phase estimates significantly in low signal to noise Signal-to-Noise Ratio (SNR) regimes for a variety of noise models. The spatial phase estimate is then used to solve a fine-scale fingerprint registration problem by relating the differences in spatial phase of two fingerprint patterns to the local deformations between them.

Following this, an explainable image classification framework is developed based on the features of a wavelet scattering transform fed into a logistic regression classifier. By solving an optimization problem related to the probability of inputs to the classifier, images are found which illustrate how a classifier is able to discriminate between classes. This framework is applied to the MNIST data set as a baseline, then to Breast MNIST, a binary classification problem concerned with malignant tumors in ultrasound images of breasts, and finally to a texture segmentation problem based on the Brodatz dataset. The class

The HandsFreeFishing program, a program developed to measure morphometric data of juvenile Chinook salmon (*Onchroynchus tshawytscha*), is then discussed. This program utilizes open-source segmentation models and elliptical Fourier analysis in order to segment fish, compute functions representing their contours, and measure length, surface area, and other features related to their health and development. These features are then used to predict fish weight, which is a particularly difficult quantity to measure in juvenile fish in practice due to instrument limitations and the relative size of the fish compared to a droplet of water.

Lastly, we discuss a 4D (3D + time) image registration problem arising in the early stages of cell formation in fruit flies (*Drosophila melanogaster*) via 3D microscopy videos. The registration problem is solved by training a convolutional neural network (CNN) to predict a displacement vector field between successive time-points. Successful registration allows us to predict the location and shape evolution of individual primordial germ cells, which has not previously been studied quantitatively.

Acknowledgments

I would like to express my sincerest gratitude to my advisor, Professor Naoki Saito, for his support and guidance throughout my time as a PhD student. He was always available, and his expertise and kind demeanor made it easy to ask questions and navigate all the nuances of life as a graduate student. Additionally, he generously provided several quarters of research funding for me, which was incredibly helpful in completing several of the projects included in this dissertation. I would also like to thank Dr. Hayden Nunley and Professor Carson Jeffres for their mentorship, and for introducing me to research in computational biology and computational ecology, respectively. Tina Denena provided amazing administrative and emotional support which I am grateful for; it goes a long way to have a friendly face to talk to when paperwork gets confusing or graduate school gets overwhelming. I am also grateful to all of the members of my qualifying exam committee – Professor Zhi Ding, Professor John Hunter, and Professor Andrew Waldron – and to my dissertation committee – Professor Zhi Ding and Professor Carson Jeffres – for their support and guidance. In addition, I would like to acknowledge Professor Mark Stankus, Professor Charles Camp and Professor Dana Paquin for mentoring me as an undergraduate and then as a masters student, and for encouraging me to apply to PhD programs. I owe Professor Luc Desmedt, Lucy Nikolaeva and Edwin Duerr my gratitude for showing me the beauty of learning early on in my education. I would also like to thank Mount Tamalpais College and the students and staff there for the opportunity to volunteer as an instructor during my time as a graduate student; I have very fond memories of my time there and will never forget those experiences.

As for the funding of these research projects, the research included in Chapter 3 was partially supported by the US National Science Foundation grants DMS-19122747 and CCF-1934568 as well as the US Office of Naval Research Grant (ONR) N00014-20-1-2381. The California Department of Fish and Wildlife funded the development of the project described in Chapter 5 through agreement Q2396019. In addition, Chief Sisk, Marine Sisk, and the Cultural Resource Specialists of the Winnemem-Wintu Tribe are owed thanks for pushing researchers at the Center for Watershed Sciences to innovative in a collaborative space. Dr. Jay Lund is also owed my gratitude for supporting the development of the program. Thanks also to Matt Salvador for data collection, and to all who provided feedback on the development of this program. I would like to thank the Center for Computational Biology at the Flatiron Institute for hospitality while the research included in

Chapter 6 was carried out. Additionally, thanks are due to Dr. Liu Yang, Dr. Celia Smits and the Shvartsman Lab, and to Professor Elizabeth Gavis and the Gavis Lab in the Department of Molecular Biology at Princeton, for providing the data necessary for this work as well as inspiration for the project.

Of course, I'd like to thank all of my friends that supported me over the last 5+ years. My housemates Tait and Regina, for being there to experience all the ups and downs of graduate school with. Darian, Michael, Jeff, Kayleigh, Blake, Nathan, Wai Ho, Gavin, EJ and Alex for always being available to give me advice, help with some job application, or just to talk to about life. Steven, Jackson and Alex, for keeping me constantly entertained in our group chat, and for staying so connected even as life draws us apart. Michael, Joey H., Joey K., Thomas, Tim P., Tim M. and Dario, and many others for always being down for a game of pick up basketball. Dru and Cameron, for checking in on me regularly and running up and down hills with me on holidays. All those at Discovery Church, for providing a loving and thoughtful community throughout my time in Davis.

I also want to thank my family for their unwavering love and support, and especially my mother for teaching me that mathematics is for everyone, and my father, for always taking an interest in the things that interested me.

Finally, I want to thank my partner, Abbie, for being my biggest fan and for making every day a bit better than I could hope for.

List of Symbols and Abbreviations

$\arg(\cdot)$	The argument of a complex number or complex valued function
$\operatorname{atan}(y, x)$	Computes the principal argument of the complex number $x + \mathbb{i}y \in \mathbb{C}$, which lies within the range $(-\pi, \pi]$
$\boldsymbol{e}_i$	A standard basis vector for $\mathbb{R}^n$, or a generator for the Clifford algebra $\mathrm{C}\ell_{3,0,0}$
$\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$	The standard basis vectors in $\mathbb{R}^3$, or the generators for the quaternions
$\boldsymbol{n}^{\perp}$	The orthogonal vector to a unit vector $\boldsymbol{n} \in \mathbb{R}^2$
$\cdot^*$	The complex conjugate of a complex number or complex valued function
$\mathrm{C}\ell_{3,0}$	The Clifford algebra generated by $\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3$
$\delta(\cdot)$	The Dirac delta function
δ_{ij}	The Kronecker delta function
$\ell^p(V)$	The space of p-summable sequences over the (countable) vector space V
$\hat{f}, \hat{f}_{\boldsymbol{k}}$	The Fourier transform of f, and the Fourier coefficient corresponding to index $\boldsymbol{k} \in \mathbb{Z}^n$
$\mathbb{i}$	The imaginary unit in $\mathbb{C}$: $\mathbb{i}^2 = -1$
$\langle \cdot, \cdot \rangle_V$	The inner product on the inner product space V
$\mathbb{D}$	The unit disc in $\mathbb{R}$ or in $\mathbb{C}$
$\mathbb{N}$	The set of all natural numbers
$\mathbb{Q}$	The quaternions
$\mathbb{Z}$	The set of all integers
$\mathcal{F}$	The Fourier transform
$\mathcal{H}$	The Hilbert transform
$\mathcal{I}$	The identity operator
$\mathcal{R}, \mathcal{R}_k$	The Riesz transform and partial Riesz transform
$\overline{f}$	The average value of the function f over a specified domain
$\mathrm{Re}f, \mathrm{Im}f$	The real and imaginary parts, respectively, of a complex valued function f
$\mathbb{R}$	The set of all real numbers

sign	The sign function
$\tau_\alpha, \tau_{\boldsymbol{\alpha}}$	Translation operators on $\mathbb{R}$, $\mathbb{R}^n$, respectively
$\mathbb{T}$	The torus
$\mathit{\Delta}$	The Laplacian
$a \propto b$	a is proportional to b
$D_{\boldsymbol{\beta}}$	A diagonal dilation matrix with parameter $\boldsymbol{\beta}$
f_A	The analytic signal of a given real-valued signal $f : \mathbb{R} \to \mathbb{R}$
f_M	The monogenic signal corresponding to the real-valued function $f : \mathbb{R}^2 \to \mathbb{R}$
$L^p(V_1; V_2)$	The space of L^p integrable functions $f : V_1 \to V_2$ between the two vector spaces V_1 and V_2
$M_S\mathbf{f}$	The structure multivector of the function $\mathbf{f} : \mathbb{R}^2 \to \boldsymbol{e}_3\mathbb{R}$
$p.v.$	Denotes a principal value integral
R_θ	A 2D rotation matrix
$s_\beta, s_{\boldsymbol{\beta}}$	Dilation operators on $\mathbb{R}$, $\mathbb{R}^n$, respectively
$W_\psi f(\alpha, \beta)$	The CWT of f with parameters α, β
1D, 2D, nD	Denoting 1-dimensional, 2-dimensional, n-dimensional respectively
AIC	Akaike information criterion [**3**]
AMD	Average mesh distance
CNN	Convolutional Neural Network
corrcoef	Correlation coefficient, for measuring success of image registration
CWT	Continuous wavelet transform
DVF	Displacement vector field
DWT	Discrete wavelet transform
EB	Equalization of Brightness
i1D, i2D	intrinsically 1-dimensional, intrinsically 2-dimensional
IAP	Instantaneous amplitude and phase
MAE	Mean average error
MAPE	Mean average percentage error
MLR	Multilinear regreassion
MNIST	The dataset of handwritten digits between 0-9 [**30**]

MRA	Multiresolution analysis
ONB	Orthonormal basis
PGC	Primordial germ cell
QAS	Quaternionic analytic signal
RBS	Random B-spline
SAM	Segment anything model [**52**]
SMV	Structure multivector
SNR	Signal-to-noise ratio
ST, WST	Scattering transform, Wavelet scattering transform, used interchangeably
STFT, WFT	Short time Fourier transform, Windowed Fourier transform
STN	Scattering transform network

CHAPTER 1

Introduction

Computer vision is a rapidly growing area of study due to the development and advancement of computational tools in the last 30 - 40 years. Image processing and image analysis are two key components of computer vision, the first being focused on the representation of image data, and the second being concerned with extracting meaningful features from the data; these often go hand in hand and will be the main concern of this dissertation. A feature is defined, very generally, as some piece of information about the content of an image, or sometimes a set of images. Features may take the form of image edges, the outline of a specific object, or a block of text on a solid blue background. Regardless, each feature comes with a story – a reason as to why it is interesting, and how it often requires significant work to measure it appropriately. This dissertation explores several different feature extraction problems and for each it proposes novel methods and algorithms to solve them. Each problem considered is directly related to real, physical data. Briefly, these problems are: (1) finescale image registration of fingerprint images, (2) explainable classification and texture segmentation of real texture images, (3) morphometric feature extraction from images of juvenile fish, and (4) registration and tracking of cells in light-sheet microscopy videos of fruit fly embryos undergoing embryogenesis.

In Chapter 2 we provide the background information–analytic signal processing and wavelet analysis–that is necessary to understand the methods and algorithms proposed in this dissertation. In particular, this covers the development of the analytic signal in one and two dimensions, multiscale analysis, and the *split of identity.* These ideas altogether combine for robust local features as discussed in Chapter 3.

Chapter 3 describes in detail an algorithm developed to estimate spatial phase from single-shot fringe pattern images such as fingerprint patterns or images seen in fringe projection profilometry. This method is derived from the 2D analytic signal and its extensions, leveraging the feature set and so-called *split of identity* which it provides in order to define novel quality maps in a multiscale fashion. Even further improvement is achieved by utilizing the low-rank structure inherent in fringe patterns, resulting in a multiscale low-rank spatial phase estimate which outperforms existing single-shot spatial phase estimation algorithms in low SNR regimes for a variety of noise models.

In Chapter 4 we outline an approach to explainable image classification and texture segmentation. First, an image classification problem and two texture segmentation problems are solved using wavelet scattering transforms combined with logistic regression classifiers. For this, several subsets of the Brodatz texture database are used. This model framework is familiar, but described here in detail. After successful training of the texture segmentation model the approach of recent work by Saito [**74**] is followed, solving a set of inverse classification problems to find inputs which maximize certain classification probabilities. These optimized inputs are then analyzed in order to better explain the classification model. As a comparison, the same approach is applied to the MNIST classification problem of recognizing handwritten digits, where the optimization results can be more readily interpreted, and to the Breast MNIST classification problem, which seeks to separate ultrasound scans of malignant tumors from benign tumors and healthy tissue.

Chapter 5 summarizes work completed in partnership with the Center for Watershed Sciences at UC Davis, building a tool to predict the weight of juvenile Chinook salmon (*Onchroynchus tshawytscha*) from fish viewer images. This project was inspired by a previous morphometric model introduced by Holmes and Jeffres in 2021 [**42**] intended to allow researchers to weigh juvenile fish more accurately and in a less invasive manner. This model was shown to provide accurate morphometric data, but requires trained practitioners to digitize each image (place 16-21 landmark points on different parts of the image of the fish). The landmark point placement is a bottleneck in the data processing pipeline, and so the HandsFreeFishing program was built as a semi-automated feature extraction tool, the features of which allow for accurate weight prediction. The program leverages Meta's Segment Anything Model (SAM) in order to segment each fish. Frther low-level feature extraction methods are then designed in order to measure quantities such as fork length and the surface area of the fish body apart from its fins. These features were chosen based on prior knowledge of juvenile Chinook salmon and showed promising weight prediction results. Additionally, the program created rich data sets and measured previously unrecorded quantities such as eye diameter, a potentially important morphometric measurement.

Chapter 6 describes a project in collaboration with the Center for Computational Biology at the Flatiron institute. Work for this project was conducted during a summer internship and as a guest researcher and visiting scholar for all of the 2024-2025 academic year. In it we propose a solution to a 4D (3D + time) registration and segmentation problem. The problem arose while studying early stages of cell formation in fruit flies *Drosophila melanogaster* via 3D microscopy videos. Our solution trains a convolutional neural network to predict 3D displacement vector fields (DVFs) between successive timepoints in videos. The location and shape-evolution of primordial germ cells (PGCs) are then tracked using these predicted DVFs in conjunction with surface geometry estimates such as mean curvature. Additionally, using a physical surface mesh deformation model developed by Kilwein et al. in 2025 [**51**], physically meaningful DVFs are constructed and then applied to real microscopy measurements in order to generate synthetic ground-truth data. The model trained on these physical simulations is compared to the one trained on randomly generated ground-truth data, and the accuracy of both of these when applied to real data is measured using an average mesh distance function.

Finally, Chapter 7 concludes by reviewing the major contributions of the dissertation, and discussing possible directions for future work. Supplementary material and extra content for all preceding chapters are included in the appendix.

We would like to note that Chapter 3 represents a detailed and expanded version of [**53**]. Additionally, Chapter 5 has been submitted to PLoS One and is currently under review, and the work discussed in Chapter 6 is part of an article in preparation and to be submitted in early 2026.

CHAPTER 2

Analytic Signal Processing

2.1. Classical Feature Extraction Methods

The focus of this chapter is to provide the necessary mathematical background in order to appreciate the contributions of this dissertation. It briefly discusses one-dimensional (1D) signal processing, as this greatly informs many image processing techniques while remaining easier to visualize and describe mathematically. The two-dimensional (2D) extensions of these ideas is then presented with the notation that will be used throughout the dissertation. Some examples are included to illustrate important ideas.

2.1.1. Fourier Analysis. The processing of one dimensional (1D) signals such as time-series or audio signals is a long-studied and robust area of research. For reference texts on this subject see "Introduction to Fourier analysis on Euclidean spaces" [**79**] and "Boundary and Eigenvalue Problems in Mathematical Physics" by Sagan [**73**]. Many of these methods rely heavily on interpreting a signal both in the time domain (or spatial domain) and in the frequency domain, via the Fourier transform (FT) $\mathcal{F} : L^1(\mathbb{R}) \cap L^2(\mathbb{R}) \to L^1(\mathbb{R}) \cap L^2(\mathbb{R})$. Recall that the Fourier transform is defined as follows:

$$\mathcal{F}(f)(\omega) := \int_{-\infty}^{\infty} f(t)\mathrm{e}^{-2\pi\mathbb{i}\omega t}\mathrm{d}t,$$

with inverse

$$\mathcal{F}^{-1}(g)(t) := \int_{-\infty}^{\infty} g(\omega)\mathrm{e}^{2\pi\mathbb{i}\omega t}\mathrm{d}\omega.$$

We denote $\mathcal{F}(f)(\omega) = \hat{f}(\omega)$. The Fourier transform is a linear operator, meaning $\mathcal{F}(\alpha f + \beta g) = \alpha\mathcal{F}(f) + \beta\mathcal{F}(g)$; this allows the transform to give interpretable features. Also, when $\omega = 0$ the integral recovers the *average value* of the function f, or the zero-frequency term, which is given by $\overline{f} := \int_{-\infty}^{\infty} f(t)\mathrm{d}t = \hat{f}(0)$.

In general, is important to consider global transformations applied to the signal such as translations, scalings, and rotations (for signals in 2D or higher), and how signal processing methods and algorithms perceive such transformations. Towards this effort, let τ_α denote a shift right by α, i.e.,

$f \circ \tau_\alpha(t) = f(t-\alpha)$, and s_β denote a dilation by $\beta > 0$, i.e., $f \circ s_\beta = \sqrt{\beta} f(\beta t)$. When $\beta < 0$ this flips f about the y-axis.

The Fourier transform satisfies the following properties:

- $\mathcal{F}(f \circ \tau_\alpha) = \mathrm{e}^{-2\pi \mathbb{i} \alpha\omega} \hat{f}(\omega)$ (*time/space shifting*)
- $\mathcal{F}(f \circ s_\beta) = \frac{1}{\sqrt{\beta}} \hat{f}\left(\frac{\omega}{\beta}\right)$ for $\beta > 0$ (*time/space scaling*)
- $\|f\|_{L^2} = \|\hat{f}\|_{L^2}$ (*Plancharel's theorem*)

Analogous properties can be shown for $\mathcal{F}^{-1}$. In addition to considering important global transformations, it is often beneficial to analyze symmetries that the operation has. For the Fourier transform, if $f(t)$ is a real-valued signal, it follows from the definition of the Fourier transform that $\hat{f}(-\omega) = -\hat{f}(\omega)^*$ where $\cdot^*$ represents the complex conjugate. For a given signal $f(t)$, this conjugate symmetry shows that the negative frequencies of f can be recovered by the positive ones, so in this sense the transform $\hat{f}(\omega)$ has redundancies. This symmetry property will be discussed further when the analytic signal is defined. The last property stated is Plancharel's theorem, and an immediate consequence of this property along with the time/space shifting property is that the absolute value of the FT is shift-invariant.

In two-dimensions (2D), the Fourier transform is defined similarly:

$$\mathcal{F}(f)(\boldsymbol{\omega}) := \iint_{\mathbb{R}^2} f(\boldsymbol{x}) \mathrm{e}^{-2\pi \mathbb{i} \boldsymbol{\omega}^\intercal \boldsymbol{x}} \mathrm{d}x_1 \mathrm{d}x_2,$$

where $\boldsymbol{x} = (x_1, x_2)$ and $\boldsymbol{\omega} = (\omega_1, \omega_2)$. Here $\boldsymbol{x}$ is typically thought of as the spatial variable, and $\boldsymbol{\omega}$ the spatial frequency. Similar results hold when considering scaling and shifting in each variable. For scaling, let $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$ and let $\tau_{\boldsymbol{\alpha}}$ denote the shift operator that shifts right in x_1 and x_2 by α_1 and α_2 respectively. Let $\boldsymbol{\beta} = (\beta_1, \beta_2)$ and let $s_{\boldsymbol{\beta}}$ be the dilation operator in 2D that dilates x_1 and x_2 by factors β_1 and β_2 respectively. The dilation operator $s_{\boldsymbol{\beta}}$ is given by $f \circ s_{\boldsymbol{\beta}} = \sqrt{\beta_1 \beta_2} f(D_{\boldsymbol{\beta}} \boldsymbol{x})$, where the *dilation matrix* $D_{\boldsymbol{\beta}}$ is defined as as follows:

Definition 2.1.1 (Dilation matrix, 2D). *Given $\boldsymbol{\beta} = (\beta_1, \beta_2)$, where $\beta_1, \beta_2 > 0$, the dilation matrix $D_{\boldsymbol{\beta}}$ is*

$$\begin{bmatrix} \beta_1 & 0 \\ 0 & \beta_2 \end{bmatrix}.$$

If $\beta = \beta_1 = \beta_2$, we simply write $D_{\boldsymbol{\beta}} := D_\beta$ where the size of D_β is understood in context.

Additionally, we can consider what happens when a rotation is applied to the spatial variable $\boldsymbol{x}$. Let R_θ denote the 2D rotation matrix which rotate a vector $\boldsymbol{x}$ counterclockwise by θ radians about the origin. For shifting, we have Explicitly we have

$$R_\theta \boldsymbol{x} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1\cos\theta - x_2\sin\theta \\ x_1\sin\theta + x_2\cos\theta \end{bmatrix}, \tag{2.1}$$

with $\det R_\theta = 1$. Now, considering

$$\mathcal{F}(f \circ R_\theta)(\boldsymbol{\omega}) = \iint_{\mathbb{R}^2} f(R_\theta \boldsymbol{x}) \mathrm{e}^{-2\pi \mathrm{i} \boldsymbol{\omega}^\intercal \boldsymbol{x}} \mathrm{d}x_1 \mathrm{d}x_2,$$

we know $R_\theta^{-1} = R_\theta^\intercal = R_{-\theta}$, so, substituting $\boldsymbol{y} = R_\theta \boldsymbol{x}$ yields

$$\mathcal{F}(f \circ R_\theta)(\boldsymbol{\omega}) = \iint_{\mathbb{R}^2} f(\boldsymbol{y}) \mathrm{e}^{-2\pi \mathrm{i} \boldsymbol{\omega}^\intercal (R_\theta^\intercal \boldsymbol{y})} (\det R_{-\theta}) \mathrm{d}y_1 \mathrm{d}y_2.$$

Finally, noticing we can write $\boldsymbol{\omega}^\intercal(R_\theta^\intercal \boldsymbol{y}) = (R_\theta \boldsymbol{\omega})^\intercal \boldsymbol{y}$, we conclude

$$\mathcal{F}(f \circ R_\theta)(\boldsymbol{\omega}) = \iint_{\mathbb{R}^2} f(\boldsymbol{y}) \mathrm{e}^{-2\pi \mathrm{i} (R_\theta \boldsymbol{\omega})^\intercal \boldsymbol{y}} \mathrm{d}y_1 \mathrm{d}y_2.$$

Thus $\mathcal{F}(f \circ R_\theta)(\boldsymbol{\omega}) = \mathcal{F}(f)(\boldsymbol{\omega}) \circ R_\theta$, i.e., the 2D Fourier transform is equivariant with respect to rotations.

All in all for the 2-D Fourier transform w Fourier satisfies the following properties:

- $\mathcal{F}(f \circ \tau_{\boldsymbol{\alpha}}) = \mathrm{e}^{-2\pi \mathrm{i} \boldsymbol{\omega}^\intercal \boldsymbol{\alpha}} \hat{f}(\boldsymbol{\omega})$ (*time/space shifting*)
- $\mathcal{F}(f \circ s_{\boldsymbol{\beta}}) = \frac{1}{\sqrt{\beta_1 \beta_2}} \hat{f}\left(D_{\boldsymbol{\beta}}^{-1} \boldsymbol{\omega}\right)$ for $\beta_1, \beta_2 > 0$ (*time/space scaling*)
- $\mathcal{F}(f \circ R_\theta)(\boldsymbol{\omega}) = \mathcal{F}(f)(\boldsymbol{\omega}) \circ R_\theta$ (*rotation equivariance*)
- $\|f\|_{L^2} = \|\hat{f}\|_{L^2}$ (*Plancharel's theorem*)

For signals on finite intervals, $f : [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_n, b_n] \to \mathbb{R}^n$ for some $-\infty < a_i < b_i < \infty$, $i = 1, \ldots, n$, the Fourier series (FS) is used rather than the FT. Specifically, for a function $f : \mathbb{T}^n \to \mathbb{R}^n$, where $\mathbb{T} = [-\frac{1}{2}, \frac{1}{2}]$, the FS of f is given by

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{k} \in \mathbb{Z}^n} \hat{f}_k \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{x}}, \text{ where } \hat{f}_{\boldsymbol{k}} = \int_{\mathbb{T}^n} f(\boldsymbol{x}) \mathrm{e}^{-2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{x}} \mathrm{d}\boldsymbol{x}, \tag{2.2}$$

where $\boldsymbol{k} \cdot \boldsymbol{x} = \boldsymbol{k}^\intercal \boldsymbol{x}$. The Fourier series representation can be regarded as first applying an analysis operator $T : L^2(\mathbb{T}^n) \to \ell^2(\mathbb{Z}^n)$, with $Tf = \{\hat{f}_{\boldsymbol{k}}\}_{\boldsymbol{k} \in \mathbb{Z}^n}$, followed by a synthesis operator $S : \ell^2(\mathbb{Z}^n) \to$

$L^2(\mathbb{T}^n)$ defined by $S\{a_{\boldsymbol{k}}\}_{\boldsymbol{k}\in\mathbb{Z}} = \sum_{\boldsymbol{k}\in\mathbb{Z}^n} a_{\boldsymbol{k}} \mathrm{e}^{2\pi \mathrm{i}\boldsymbol{x}\cdot\boldsymbol{k}}$. When $n = 1$ we have the standard Fourier series representation $f : \mathbb{T} \to \mathbb{R}$ with $f(t) = \sum_{k\in\mathbb{Z}} \hat{f}_k \mathrm{e}^{2\pi \mathrm{i}tk}$ with $\hat{f}_k = \int_{\mathbb{T}} f(t)e^{-2\pi \mathrm{i}tk}\mathrm{d}t$.

2.1.2. Wavelet Analysis. The success of the Fourier transform to extract all frequency information from a signal f comes at the expense of losing all information about the time (spatial) domain. This means that if we want to be certain our signal has a specific frequency, then we cannot be certain about where that frequency is present in time (space). Formally this is stated via Heisenberg's uncertainty principle [**20**]. Time-frequency analysis explores this trade off. For some standard reference texts on the subject see [**48**], [**20**]. The basic idea that led to the development of wavelet analysis can be illustrated by considering the *short time Fourier transform* (STFT) or *windowed Fourier transform* (WFT) [**38**]. It is this: if we consider only a specific interval of the signal $f(t)$ given by $t \in [t_0, t_1]$ for some $t_1 < t_2$, and compute the FT of this time-limited signal, the resulting spectrum will be localized in time within $[t_0, t_1]$. Sliding this window along, the STFT gives a time-frequency representation of f somewhere between the Formally, it is defined in the continuous setting as

$$\mathrm{STFT}\{f(t)\}(\alpha, \omega) = \int_{-\infty}^{\infty} f(t)w(t-\alpha)\mathrm{e}^{-2\pi \mathrm{i}\omega t}\mathrm{d}\omega.$$

The function w here is the *windowing function*, and the choice of this function determines the resulting output, as well as the resulting properties of the transformation. Notice the STFT is a function of both α and ω. When interpreting the STFT, we can view the windowing function as either acting on the signal $f(t)$ itself, or as acting on the complex exponentials $\mathrm{e}^{-2\pi \mathrm{i}\omega t}$. Viewing it in the latter sense helps connect the idea of the STFT to *wavelet transforms* and wavelet analysis, where a fixed function ψ is dilated and shifted systematically, and correlations with the signal f and these dilated and shifted versions of ψ are measured as the *wavelet coefficients*. Knowledge about the spatial and frequency properties of ψ then determines the interpretation of these coefficients.

Let $\psi_{\alpha,\beta} = \frac{1}{\sqrt{\beta}}\psi\left(\frac{t-\alpha}{\beta}\right)$. The 1D *continuous wavelet transform* (CWT) of a signal $f(t)$ with respect to the mother wavelet ψ is defined by

$$\begin{aligned} W_\psi f(\alpha, \beta) &:= \int_{-\infty}^{\infty} f(t)\psi^*_{\alpha,\beta}(t)\mathrm{d}t \\ &= \langle f, \psi_{\alpha,\beta} \rangle, \end{aligned}$$

where the mother wavelet ψ satisfies the *admissibility* conditions

$$C_\psi := \int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{|\omega|} \mathrm{d}\omega < \infty,$$
$$\int_{-\infty}^{\infty} |\psi(t)|^2 \mathrm{d}t = 1.$$

These admissibility conditions ensure the CWT is invertible, and the *inverse continuous wavelet transform* (ICWT) is then given by:

$$f(t) := \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_0^{\infty} W_\psi f(\alpha, \beta) \cdot \psi^*_{\alpha,\beta}(t) \frac{\mathrm{d}\beta}{\beta^2} \mathrm{d}\alpha.$$

The function $W_\psi f(\alpha, \beta)$ consists of *wavelet coefficients* for each value of α and β. Because the CWT is a 2D signal defined for all possible translations and scalings, taking the CWT of a time-shifted and time-scaled function $f_{\alpha_0,\beta_0} = \frac{1}{\sqrt{\beta}} f\left(\frac{t-\alpha_0}{\beta_0}\right)$ results in shifting the wavelet transform itself. Explicitly, we have

$$W_\psi f_{\alpha_0,\beta_0}(\alpha, \beta) = W_\psi f(\alpha - \alpha_0, \frac{\beta}{\beta_0}),$$

thus the CWT is equivariant with respect to these transformation, similarly to the FT. Because of this, we have $\|W_\psi f\|_{L^2} = \|W_\psi f_{\alpha_0,\beta_0}\|_{L^2}$, and as such the *energy of the CWT* is invariant to scaling and translation.

Example 2.1.2 (Morlet Wavelets [**43**]). *The Morlet mother wavelet is constructed by subtracting a constant from a plane wave, and then localizing the plane wave via a Gaussian window. It is defined as*

$$\psi_\sigma(t) = c_\sigma \pi^{-\frac{1}{4}} \mathrm{e}^{-\frac{1}{2}t^2} \left(\mathrm{e}^{\mathrm{i}\sigma t} - \kappa_\sigma\right)$$

with $c_\sigma = \left(1 + \mathrm{e}^{-\sigma^2} - 2\mathrm{e}^{-\frac{3}{4}\sigma^2}\right)$ *and* $\kappa_\sigma = \mathrm{e}^{-\frac{1}{2}\sigma}$. *The Fourier transform of* ψ *is given by*

$$\hat{\psi}_\sigma(\omega) = c_\sigma \pi^{-\frac{1}{4}} \left(e^{-\frac{1}{2}(\sigma-\omega)^2} - \kappa_\sigma \mathrm{e}^{-\frac{1}{2}\omega^2}\right).$$

The parameter σ *here controls the trade-off between time and frequency resolution.*

Two dimensional wavelet analysis extends very naturally from the 1D case, again with the addition of rotations. In general, a 2D mother wavelet will be defined as $\psi_{\boldsymbol{\alpha},\beta,\theta}(\boldsymbol{x}) = \beta^{-1}\psi\left(R_{-\theta}\left(\frac{\boldsymbol{x}-\boldsymbol{\alpha}}{\beta}\right)\right)$, and the 2D wavelet transform of $f(\boldsymbol{x})$ by

$$\begin{aligned} W_\psi f(\boldsymbol{\alpha}, \beta, \theta) &:= \iint_{\mathbb{R}^2} f(\boldsymbol{x}) \psi^*_{\alpha,\beta,\theta}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\ &= \langle f, \psi_{\boldsymbol{\alpha},\beta,\theta} \rangle. \end{aligned}$$

In our work, we consider only the case of isotropic wavelets, e.g., mother wavelets satisfying $\psi(\boldsymbol{x}) = \psi(\|\boldsymbol{x}\|)$ along with the standard admissibility conditions

$$C_\psi = \frac{1}{4\pi^2} \iint_{\mathbb{R}^2} \frac{|\hat{\psi}(\boldsymbol{\omega})|^2}{|\boldsymbol{\omega}|^2} \boldsymbol{\omega} < \infty,$$

and $\psi \in L^2(\mathbb{R}^2)$. In the discrete setting, these are constructed by tiling the frequency domain with radial tiles which form a *frame* (Definition 2.1.3) for $\mathbb{R}^2$. In doing so, each wavelet filter represents a specific frequency band of the given image, independent of the orientation of such areas. This becomes very useful if the local orientation and local phase of interest; this is discussed further in Chapter 3.

In practice, we must consider a finite sampling of translations and scalings, which lead to the discrete wavelet transform (DWT). The theoretical ideas behind this restriction are summarized in *multiresolution analysis* (MRA). The DWT and MRA have cemented wavelet analysis as the preferred method of time-frequency analysis in many disciplines. We leave the details for the interested reader to explore on their own, and include here only the basics necessary to understand our contributions. First, we define a frame and discuss some basic terminology in frame theory.

Definition 2.1.3 (Frame). *A frame with frame bounds $0 \leq a \leq A$ for the vector space V is a spanning set $\{\phi_k\}_{k\in\mathbb{N}}$ that, for all $f \in V$ satisfies the frame condition:*

$$a\|f\|^2 \leq \sum_{k\in\mathbb{N}} |\langle f, \phi_k \rangle|^2 \leq A\|f\|^2.$$

The frame is said to be *tight* if $a = A$, and if $a = A = 1$ then it is a Parseval frame. An ONB for V is always a *Parseval frame*, though the converse is not true. For a given frame, there exists the so-called analysis and synthesis operators $T : V \to \ell^2$ and $T^* : \ell^2 \to V$ respectively. The analysis operator computes the coefficients that result by projecting an vector onto each frame element: $Tf = \{\langle f, \phi_k \rangle\}_{k\in\mathbb{N}}$. The synthesis operator is the adjoint of T, and is computed by taking a linear combination of the frame elements of a given sequence of coefficients: $T^*\{c_k\}_{k\in\mathbb{Z}} = \sum_{k\in\mathbb{Z}} c_k \phi_k$. The *frame operator* $S : V \to V$ is given by the composition $S = T^*T$. The frame operator is invertible, and $\{\tilde{\phi}_k = S^{-1}\phi_k\}_{k\in\mathbb{Z}}$ forms the *dual frame.* This dual frame has frame bounds given by $\frac{1}{A} \leq 0 \leq \frac{1}{a}$, and we can see that, given $v \in V$,

$$\begin{aligned} u &= \sum_k \langle v, \phi_k\rangle \tilde{\phi}_k \\ &= \sum_k \langle v, \phi_k\rangle S^{-1}\phi_k \\ &= S^{-1}\left(\sum_k \langle v, \phi_k\rangle \phi_k\right) \\ &= S^{-1}Sv \\ &= v. \end{aligned}$$

Example 2.1.4. *The standard Fourier basis on* $L^2\left(\mathbb{T}^n\right)$ *with inner product* $\langle f, g\rangle := \int_{\mathbb{T}^n} f(\boldsymbol{x})g^*(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$ *is given by* $\{\phi_{\boldsymbol{k}} = e^{2\pi\mathbb{i}(\cdot)^{\intercal}\boldsymbol{k}}\}_{\boldsymbol{k}\in\mathbb{Z}^n}$. *This is an ONB and therefore a Parseval frame. Applying the analysis operator yields the Fourier series coefficients* $\hat{f}(\boldsymbol{k}) = \langle f, \phi_{\boldsymbol{k}}\rangle$. *Additionally, since this is an ONB we have* $\langle \phi_{\boldsymbol{j}}, \phi_{\boldsymbol{k}}\rangle = \delta_{\boldsymbol{jk}}$, *therefore* $(T\phi_{\boldsymbol{k}})_{\boldsymbol{j}} = \delta_{\boldsymbol{jk}}$ *and* $T^*T\phi_{\boldsymbol{k}} = \phi_{\boldsymbol{k}}$, *so* S *is the identity. This implies the dual frame element* $\tilde{\phi}_k = S^{-1}\phi_{\boldsymbol{k}} = \phi_{\boldsymbol{k}}$, *and so we have:*

$$\begin{aligned} f &= \sum_{\boldsymbol{k}\in\mathbb{Z}^n} \langle f, \phi_{\boldsymbol{k}}\rangle \tilde{\phi}_{\boldsymbol{k}} \\ &= \sum_{\boldsymbol{k}\in\mathbb{Z}^n} \hat{f}(\boldsymbol{k}) e^{2\pi\mathbb{i}\boldsymbol{x}^{\intercal}\boldsymbol{k}}, \end{aligned}$$

which is the standard n*-dimensional Fourier series.*

The following theorem, a special case of Theorem 3.1 in [**4**], provides a way to construct wavelet frames for $L^2\left(\mathbb{T}^2\right)$.

Theorem 2.1.5. *[4] Let* $\{\phi_{\boldsymbol{k}}\}_{\boldsymbol{k}\in\mathbb{Z}^n}$ *be an ONB for* $L^2\left(\mathbb{T}^2\right)$ *with inner product as stated in the example above,* ψ *be a mother wavelet with* $\operatorname{supp}\hat{\psi} \subseteq \mathbb{T}^2$, *and* $\hat{\psi}_j(\boldsymbol{\omega}) = D\hat{\psi}_j(\boldsymbol{\omega})$ *where* D *be a dilation operator corresponding to the dilation matrix* D, *where* $Df := |\det(D)^{\frac{1}{2}}|f(D\boldsymbol{\omega})$. *Then if there exists finite constants* $0 < a \leq A$ *such that for all* $\boldsymbol{\omega} \in \mathbb{R}^2$, $p \leq \sum_{j\in\mathbb{Z}} |\hat{\psi}_j(\boldsymbol{\omega})|^2, \leq P$, *it follows that*

$$\{D^jT_{\boldsymbol{k}}\psi : j \in \mathbb{Z}, \boldsymbol{k} \in \mathbb{Z}^2\}$$

is a wavelet frame for $L^2(\mathbb{R}^2)$ with frame bounds a and A.

EXAMPLE 2.1.6 (Isotropic Wavelets [**40**]). *We define a mother wavelet ψ by defining its Fourier transform $\hat{\psi}$ to be the following piecewise function:*

$$\hat{\psi}(\boldsymbol{\omega}) = \begin{cases} \cos\left(2\pi q(\|\boldsymbol{\omega}\|)\right), & \|\boldsymbol{\omega}\| \in (\frac{1}{8}, \frac{1}{4}], \\ \sin\left(2\pi q\left(\frac{1}{2}\|\boldsymbol{\omega}\|\right)\right), & \|\boldsymbol{\omega}\| \in (\frac{1}{4}, \frac{1}{2}], \\ 0, & \textit{otherwise}, \end{cases}$$

where $q \in C^m\left((\frac{1}{8}, \frac{1}{4}]\right)$ and satisfies the following properties:

- $0 \leq q(t) \leq \frac{1}{4}\ \forall t \in (\frac{1}{8}, \frac{1}{4}]$
- $q(\frac{1}{8}) = \frac{1}{4}$ *and* $q(\frac{1}{4}) = 0$
- *if* $m \geq 1$ *then* $q^{(j)}(\frac{1}{8}) = 0 = q^{(j)}(\frac{1}{4})$, $j = 1, \ldots, m$.

These conditions guarantee that $\hat{\psi}$ is m-times continuously differentiable with compact support on $[-\frac{1}{2}, \frac{1}{2}]$. The wavelet $\hat{\psi}$ is isotropic by definition, and it is shown in [40] that $\{D_2^j T_{\boldsymbol{k}}\psi : j \in \mathbb{Z}, \boldsymbol{k} \in \mathbb{Z}^2\}$ generates a tight wavelet frame for $L^2(\mathbb{R}^2)$, where D_2 is the dyadic dilation operator defined by $D_2 f = 2f(2\omega)$. Explicitly, in the frequency domain we have

$$\psi_j(\boldsymbol{\omega}) = \psi(2^k\boldsymbol{\omega}) = \begin{cases} \cos\left(2\pi q(2^k\|\boldsymbol{\omega}\|)\right), & \|\boldsymbol{\omega}\| \in (\frac{2^{-k}}{8}, \frac{2^{-k}}{4}], \\ \sin\left(2\pi q\left(2^{k-1}\|\boldsymbol{\omega}\|\right)\right), & \|\boldsymbol{\omega}\| \in (\frac{2^{-k}}{4}, \frac{2^{-k}}{2}], \cdot \\ 0, & \textit{otherwise}, \end{cases}$$

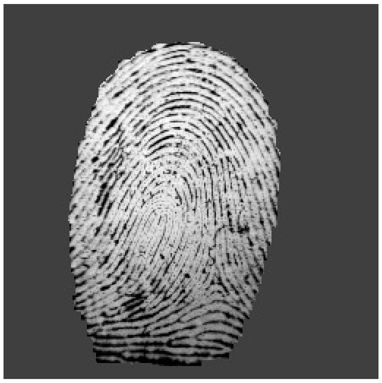
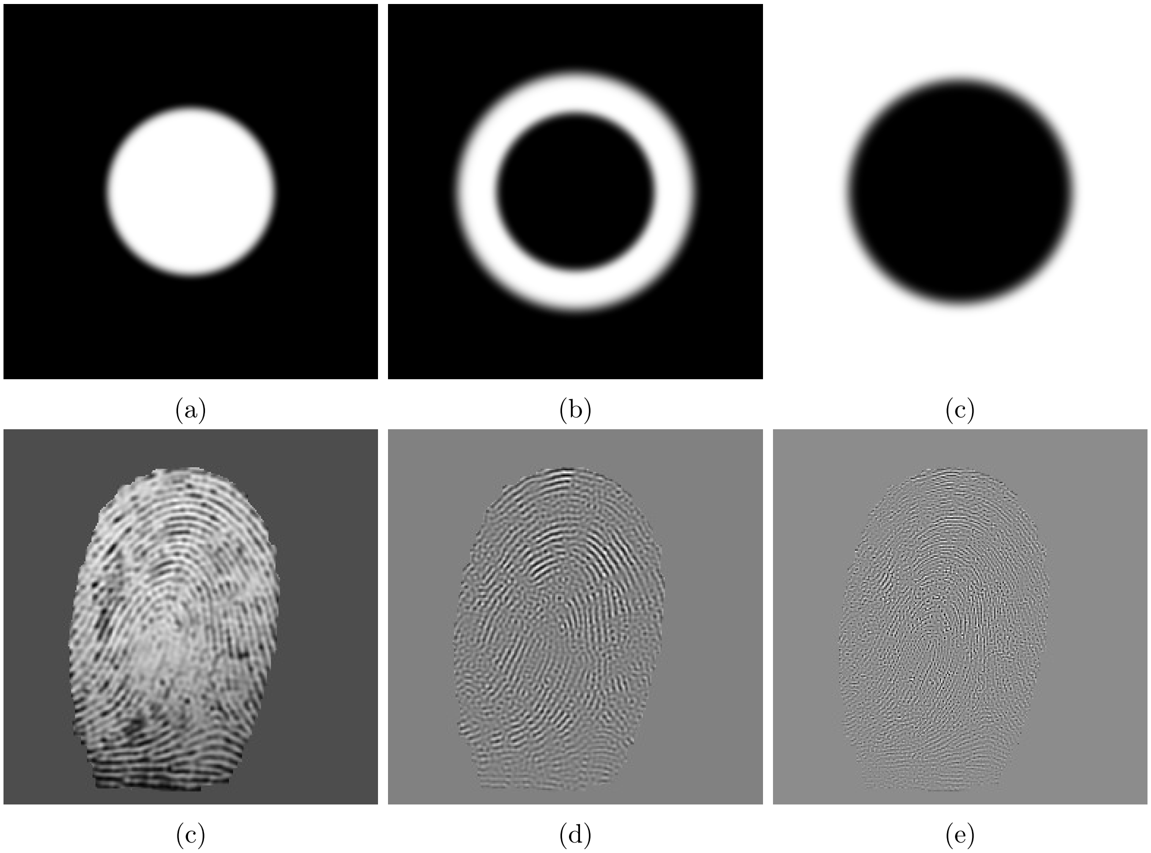


FIGURE 2.1. A real fingerprint image from [**60**].


(a) (b) (c)

(c) (d) (e)

Figure 2.2. Isotropic wavelet frames, (a) low-pass $h_{1,1}$; (b) band-pass $h_{1,2}$; (c) high-pass $h_{1,3}$. Applied to a real fingerprint image: (d) low-pass; (e) band-pass; (f) high-pass.

In [**40**] these isotropic wavelets are further divided into k subchannels, a low-pass filter, $k-1$ band pass filters, and a high pass filter. For a visual aid, consider the image of a fingerprint given in Figure 2.1. Figure 2.2 shows an example of these wavelet scales for this fingerprint image from the FVC2004 DB1-B database [**60**].

2.2. The Analytic Signal

Much of this dissertation is related to the concept of the *monogenic signal*, a quaternion-valued function derived from a real-value 2-dimensional (2D) signal, or an image. The monogenic signal has become a widely used tool for image feature extraction, and the motivation of its development signal is based on 1D *analytic signal* in the analysis of the so-called instantaneous phase, amplitude, and frequency of a real-valued signal. The analytic signal in 1D is first outlined here. We begin from its derivation and then examine many of its important properties and applications.

The underlying assumption is that with any real-valued signal f there is an imaginary-valued function conjugate to it which provides additional structural information about the real signal, such as the envelope of the signal. Consider the signal mode $f(t) = A(t)\cos\varphi(t)$. It will be shown that if $A(t)$ varies slowly with respect to $\varphi(t)$, and that $\varphi(t)$ is smooth, then it is possible to find a complex-valued extension of f whose amplitude is given by $A(t)$ and whose phase is precisely $\varphi(t)$ mod 2π. If a *phase unwrapping* problem can then be solved to remove the modulo 2π ambiguity from the phase φ, then one can further compute the instantaneous frequency given by $\varphi'(t)$. These features, paired with multiscale analysis techniques, have led to many successes in the field of signal processing.

The analytic signal lies at the intersection of the theory of analytic functions and signal processing methods. Making use of the conjugate symmetry of the Fourier transform for real-valued signals, the analytic signal extends a real-valued signal $f : \mathbb{R} \to \mathbb{R}$ to a complex-valued signal $f_A : \mathbb{R} \to \mathbb{C}$, preserving the positive frequencies component and discarding the (redundant) negative frequency component:

$$\hat{f}_A(\omega) = \begin{cases} 0 & \omega < 0, \\ \hat{f}(0) & \omega = 0, \\ 2\hat{f}(\omega) & \omega > 0. \end{cases} \tag{2.3}$$

The benefit of this complex-valued signal is that, assuming a form $f_A(t) = A(t)\mathrm{e}^{\mathbb{i}\phi(t)}$, this yields an *instantaneous amplitude and phase* (IAP) decomposition, and decomposes into $f_A(t) = A(t)(\cos\phi(t) + \mathbb{i}\sin\phi(t))$, with $\mathrm{Re} f_A = f$. These instantaneous amplitude and phase, defined by

$$\begin{aligned} A(t) &:= |f_A(t)|, \\ \phi(t) &:= \operatorname{atan}\left(\mathrm{Im} f_A, \mathrm{Re} f_A\right), \end{aligned}$$

are then meaningful features of the original signal f. Here atan represents the *atan*2 function whose range is given by $(-\pi, \pi]$, so from this representation we recover a wrapped phase, hence a discontinuous phase function. This is particular factorization of f_A, and is not uniquely defined. The Blaschke product, discussed in Section 2.2.1, is another factorization of the analytic signal where the recovered phase function does not have this 2π ambiguity. Recovering the instantaneous frequency of f requires differentiating the phase function, so in order to do this successfully the recovered phase must be unwrapped. The analytic signal, f_A, is commonly defined by way of the *Hilbert transform.*

Definition 2.2.1 (Hilbert transform on $\mathbb{R}$). *The Hilbert transform, denoted $\mathcal{H}$, is defined as the convolution*

$$\mathcal{H}f(t) = \frac{1}{\pi} p.v. \int_{\mathbb{R}} \frac{f(\tau)}{t-\tau} d\tau, \tag{2.4}$$

*where $f \in L^p(\mathbb{R})$, where p.v. denotes the principal value integral. The Hilbert transform is a bounded linear operator on $L^p(\mathbb{R})$ for $p \in (1, \infty)$ [**79**].*

Definition 2.2.2 (Analytic signal, instantaneous amplitude and phase). *Consider a signal $f \in L^2(\mathbb{R})$. The analytic signal $f_A \in L^2(\mathbb{R};\mathbb{C})$ is the complex valued signal $f(t) + \mathbb{i}\mathcal{H}f(t)$. The instantaneous phase of the signal is given by the argument of the corresponding analytic signal: $\varphi(t) = \arg(f_A(t))$. The instantaneous amplitude of the signal is given by the modulus of the corresponding analytic signal: $A(t) = |f_A(t)|$.*

The *Fourier multiplier* of a convolutional operator $\mathcal{K} : L^1(\mathbb{R}) \cap L^2(\mathbb{R}) \to L^1(\mathbb{R}) \cap L^2(\mathbb{R})$ with kernel function k, is the Fourier transform $\hat{k}(\omega)$; the terminology of multiplier comes from the equation $\widehat{\mathcal{K}f}(\omega) = \hat{k}(\omega)\hat{f}(\omega)$. The kernel of the Hilbert transform is given by $h(t) = \frac{1}{\pi t}$, keeping in mind the convolutional operator is taken in the sense of a principal value integral. The Fourier multiplier of the Hilbert transform $\mathcal{H}$, then, is given by $-\mathbb{i}\,\mathrm{sign}(\omega)$, which can be derived by the Dirichlet integral identity $\int_{\mathbb{R}} \frac{\sin(at)}{t} \mathrm{d}t = \pi\,\mathrm{sign}(a)$:

$$\begin{aligned}\hat{h}(\omega) &= \frac{1}{\pi} p.v. \int_{\mathbb{R}} \frac{\mathrm{e}^{-2\pi\mathbb{i}\omega t}}{t} \mathrm{d}t = \frac{1}{\pi} p.v. \int_{\mathbb{R}} \left(\frac{\cos(2\pi\omega t)}{t} - \mathbb{i}\frac{\sin(2\pi\omega t)}{t} \right) \mathrm{d}t \\ &= -\mathbb{i}\,\mathrm{sign}(\omega)\end{aligned}$$

Further, $A(t) = \sqrt{|f(t)|^2 + |\mathcal{H}f(t)|^2}$, $\phi(t) = \mathrm{atan}\left(\mathcal{H}f, f\right)$. Realizing the analytic signal as the pairing of the real-valued signal f with its Hilbert transform provides much insight into the analytic signal itself, as well as higher dimensional analogs, as we will see. Example 2.2.3 shows that the

Hilbert transform performs a *quadrature shift* on pure sinusoids, i.e., assuming $\omega > 0$:

$$\mathcal{H}\cos(2\pi\omega t+\varphi) = \cos(2\pi\omega t+\varphi-\pi/2) = \sin(2\pi\omega t+\varphi),$$
$$\mathcal{H}\sin(2\pi\omega t+\varphi) = \sin(2\pi\omega t+\varphi-\pi/2) = -\cos(2\pi\omega t+\varphi),$$

and sheds light on the amplitude and phase of the analytic signal. It also shows that the Hilbert transform acts as a normalized derivative operator on pure sinusoids, albeit with a sign flip, again we assume $\omega > 0$:

$$\mathcal{H}\cos(2\pi\omega t+\varphi) = -\frac{1}{2\pi\omega}\frac{\mathrm{d}}{\mathrm{d}t}\cos(2\pi\omega t+\varphi),$$
$$\mathcal{H}\sin(2\pi\omega t+\varphi) = -\frac{1}{2\pi\omega}\frac{\mathrm{d}}{\mathrm{d}t}\sin(2\pi\omega t+\varphi).$$

EXAMPLE 2.2.3. *Let $f(t) = A\cos(2\pi\omega t)$, for $\omega > 0$. Rewriting this as the sum of complex exponentials, $f(x) = \frac{A}{2}\left(\mathrm{e}^{2\pi\mathbb{i}\omega t} + \mathrm{e}^{-2\pi\mathbb{i}\omega t}\right)$, which yields the Fourier transform $\hat{f}(\omega) = \frac{A}{2}\left(\delta(\omega)+\delta(-\omega)\right)$, where $\delta(\cdot)$ denotes the Dirac delta function. According to Equation (2.3), this implies the analytic signal $\hat{f}_A(\omega)$ is equal to $2A\delta(\omega)$, and thus $f_A(t) = A\mathrm{e}^{2\pi\mathbb{i}\omega t} = A\left(\cos(2\pi\omega t) + \mathbb{i}\sin(2\pi\omega t)\right)$. According to Def. 2.2.2, this means $\mathcal{H}f(t) = A\sin(2\pi\omega t)$. Additionally, deriving $A(t)$ and $\phi(t)$ here gives $A(t) = A$, and $\phi(t) = 2\pi\omega t \mod 2\pi$. A similar set of calculations shows that $\mathcal{H}(A\sin(2\pi\omega t)) = -A\cos(2\pi\omega t)$.*

It is important to note that $\operatorname{sign}(0) = 0$, so that $\overline{\mathcal{H}f} = \widehat{\mathcal{H}f}(0)$ is always zero. Because of this, before processing a signal f usually the mean is stored and the signal is set to have a mean of zero $f \mapsto f - \overline{f}$. On signals with zero mean, then, the Hilbert transform is an *anti-involution*, meaning that $\mathcal{H}^{-1} = -\mathcal{H}$, or that $\mathcal{H}^2 = -\mathcal{I}$. This is clear form the Fourier multiplier: $\hat{h}$, since $\hat{h}^2(\omega) = -1$ for $\omega \neq 0$. We also note here the property

$$\mathcal{H}\mathrm{e}^{2\pi\mathbb{i}\omega x} = -\mathbb{i}\operatorname{sign}(\omega)\mathrm{e}^{2\pi\mathbb{i}\omega x}. \tag{2.5}$$

If the signal f is instead defined on a finite interval, $\mathbb{T}$, then the Fourier series coefficients of the analytic signal are given by

$$(\hat{f}_A)_k = \begin{cases} 0 & k < 0, \\ \hat{f}_0 & k = 0, \\ 2\hat{f}_k & k > 0. \end{cases}$$

Hence, the Fourier series representation of the analytic signal is given by

$$f_A(t) = \hat{f}_0 + 2\sum_{k\geq 1} \hat{f}_k \mathrm{e}^{2\pi \mathbb{i} tk}.$$

In this case, f_A is also represented as $f_A(t) = f(t) + \mathbb{i}\mathcal{H}f(t)$, where the Hilbert transform on the interval $\mathbb{T}$ is defined as follows.

Definition 2.2.4 (Hilbert Transform on $\mathbb{T}$). *The Hilbert transform, denoted $\mathcal{H}$, is defined as the convolution*

$$\mathcal{H}f(t) = p.v. \int_{-\frac{1}{2}}^{\frac{1}{2}} f(\tau)\cot(\pi(t-\tau))\mathrm{d}\tau, \tag{2.6}$$

where $f \in L^p(\mathbb{T})$, $1 < p < \infty$, $\mathbb{T} = [-\frac{1}{2}, \frac{1}{2}]$.

Equations (2.4) and (2.6) are related to one another through the Poisson summation formula. Explicitly, the formula $\cot(\pi t) = 2\sum_{k\geq 1}\sin(2\pi kt)$ is derived from this summation formula, which is valid in the sense of distributions on the interval $[-\frac{1}{2}, \frac{1}{2}]$. Here, by substituting $z = \mathrm{e}^{2\pi it}$, the connection to analytic functions is seen clearly. In terms of z, we have $f_A(z) = \hat{f}_0 + 2\sum_{k\geq 1}\hat{f}_k z^k$. If z is then taken to be any complex number in the complex unit disk $\mathbb{D}$, $f_A(z)$ is then an analytic function on $\mathbb{D}$, known to satisfy the Cauchy-Riemann equations. This relates the analytic signal to the rich theory of complex analytic functions. Specifically, $f_A(t)$ for $t \in \mathbb{T}$ forms the boundary data of this analytic function $f_A(z)$, and the original signal f and its Hilbert transform $\mathcal{H}f$ form the real and imaginary parts of the boundary data. Extending f and $\mathcal{H}f$ to $\mathbb{D}$, then, defined as $\mathrm{Re}f_A(z)$ and $\mathrm{Im}f_A(z)$ respectively, shows that $f(z)$ and $\mathcal{H}f(z)$ are harmonic conjugates. These ideas provide a nice characterization of the analytic signal of a function f as the boundary value of some complex analytic function on $\mathbb{D}$. For more on this connection the theory of analytic functions, see **[73]**, **[79]**.

The Hilbert transform for functions on $\mathbb{T}$ acts the same as before on pure sinusoids: $\mathcal{H}(A\cos(2\pi kt)) = A\sin(2\pi kt)$, $\mathcal{H}(A\sin(2\pi kt)) = -A\cos(2\pi kt)$ for $k \in \mathbb{Z}$, now, with Fourier multiplier $-\mathbb{i}\,\mathrm{sign}(k)$, so that

$$\mathcal{H}f(t) = \sum_{k\neq 0} 2\hat{f}_k \mathrm{e}^{2\pi\mathbb{i}kt}.$$

which is shown now for the sake of completeness; making use of the

Proof. Let $k \in \mathbb{Z}$, and $f(t) = \cos(2\pi kt)$ for $t \in \mathbb{T}$. Then the Hilbert transform of f, letting $T = t - \tau$ and $\mathrm{d}T = -\mathrm{d}\tau$ in Equation (2.6):

$$\begin{aligned}
\mathcal{H}f(t) &= -\text{p.v.} \int_{\mathbb{T}} \cos(2\pi k(t-T))\cot(\pi T)\mathrm{d}T \\
&= -\sum_{k'\geq 1} \text{p.v.} \int_{\mathbb{T}} \cos(2\pi k(t-T))\sin\big(2\pi k' T\big)\mathrm{d}\tau \\
&= -\sum_{k'\geq 1} \cos(2\pi kt)\cdot \int_{\mathbb{T}} \cos(2\pi kT)\sin\big(2\pi k'\tau\big)\mathrm{d}T + \sin(2\pi kt)\cdot \int_{\mathbb{T}} \sin(2\pi kT)\sin\big(2\pi k'\tau\big)\mathrm{d}T \\
&= \sin(2\pi kt).
\end{aligned}$$

A similar computation shows $\mathcal{H}\sin(2\pi kt) = -\cos(2\pi kt)$. □

The interpretations of $\mathcal{H}$ as a quadrature shift, or a normalized derivative, hold in this setting on $\mathbb{T}$, too. Example 2.2.5 considers a signal $f : \mathbb{T} \to \mathbb{R}$ and its Fourier series representation.

Example 2.2.5 (Hilbert transform and Analytic signal of a trigonometric series). *Consider a real-valued function f and its Fourier series representation*

$$f(x) = \frac{a_0}{2} + \sum_{k\geq 1} \left(a_k \cos(2\pi kx) + b_k \sin(2\pi kx)\right).$$

Then the Hilbert transform of such a signal is given by

$$\mathcal{H}(f)(x) = \sum_{k\geq 1} \left(a_k \sin(2\pi kx) - b_k \cos(2\pi kx)\right).$$

The corresponding analytic signal is given by

$$\begin{aligned}
u(x) &= f(x) + \mathbb{i}\mathcal{H}f(x) \\
&= \frac{a_0}{2} + \sum_{n\geq 1} \left(a_k - \mathbb{i}b_k\right) \mathrm{e}^{2\pi\mathbb{i}kx}.
\end{aligned}$$

So far the features extracted have only been interpretable when derived from pure sinusoids. Bedrosian's theorem [8] is what makes the Hilbert transform and the analytic signal such an important feature extraction tool.

Theorem 2.2.6 (Bedrosian's Theorem [8]). *Let $f, g \in L^2(\mathbb{R})$, such that $\hat{f}(\omega) = 0$ for $|\omega| > A$, and $\hat{g}(\omega) = 0$ for $|\omega| < A$, where A is some positive constant. That is, f is a low frequency function, and g a high frequency function, with mutually exclusive supports in the frequency domain. Then*

$$\mathcal{H}\left[f(t)g(t)\right] = f(t)\mathcal{H}g(t). \tag{2.7}$$

The proof of this theorem is simple and elegant, so it is included here for the reader to enjoy.

Proof. Consider the Fourier transforms

$$\hat{f}(\omega) = \int_{\mathbb{R}} f(t)\mathrm{e}^{-2\pi \mathbb{i}\omega t}\mathrm{d}t, \quad \hat{g}(\omega) = \int_{\mathbb{R}} g(t)\mathrm{e}^{-2\pi \mathbb{i}\omega t}\mathrm{d}t.$$

We may f and g in terms of the inverse Fourier transform as: $f(t) = \mathcal{F}^{-1}\hat{f}(t)$ and $g(t) = \mathcal{F}^{-1}\hat{g}(t)$. Further, the product can be written as

$$\begin{aligned} f(t)g(t) &= \mathcal{F}^{-1}\hat{f}(t) \cdot \mathcal{F}^{-1}\hat{g}(t) \\ &= \int_{\mathbb{R}}\int_{\mathbb{R}} \hat{f}(\omega)\hat{g}(\omega')\mathrm{e}^{2\pi\mathbb{i}\omega t}\mathrm{e}^{2\pi\mathbb{i}\omega' t}\mathrm{d}\omega\mathrm{d}\omega' \\ &= \int_{\mathbb{R}}\int_{\mathbb{R}} \hat{f}(\omega)\hat{g}(\omega')\mathrm{e}^{2\pi\mathbb{i}(\omega+\omega')t}\mathrm{d}\omega\mathrm{d}\omega' \end{aligned}$$

Then, applying $\mathcal{H}$ to this product is just a matter of computing $\mathcal{H}\mathrm{e}^{2\pi\mathbb{i}(\omega+\omega')t} = -\mathbb{i}\,\mathrm{sign}(\omega + \omega')\mathrm{e}^{2\pi\mathbb{i}(\omega+\omega')t}$, so it follows:

$$\mathcal{H}[f(t)g(t)] = \int_{\mathbb{R}}\int_{\mathbb{R}} -\mathbb{i}\,\mathrm{sign}(\omega+\omega')\hat{f}(\omega)\hat{g}(\omega')\mathrm{e}^{2\pi\mathbb{i}(\omega+\omega')t}\mathrm{d}\omega\mathrm{d}\omega'.$$

Now, $\hat{f}(\omega)\hat{g}(\omega')$ is only non-zero when $-A < \omega < A$ and $|\omega'| > A$. When $\omega' > A$, $\mathrm{sign}(\omega + \omega') = 1 = \mathrm{sign}(\omega')$, and, similarly, when $\omega' < A$, $\mathrm{sign}(\omega+\omega') = -1 = \mathrm{sign}(\omega')$. Thus, we may write

$$\begin{aligned} \mathcal{H}[f(t)g(t)] &= f(t)\left(\int_{\mathbb{R}} -\mathbb{i}\,\mathrm{sign}(\omega')\hat{g}(\omega')\mathrm{e}^{2\pi\mathbb{i}\omega' t}\mathrm{d}\omega'\right), \\ &= f(t)\mathcal{H}g(t). \end{aligned}$$

□

Bedrosian's theorem is immediately applicable to phase and amplitude demodulation problems in 1D. In these problems, there is a carrier signal $c(t) = \cos(\omega_c t + \phi_c)$ with a known carrier frequency ω_c and carrier phase ϕ_c, and a message signal $m(t)$, where the message is assumed to be bandlimited with $\hat{m}(\omega) = 0$ for $|\omega| \geq \omega_c$. The *phase modulated signal* $c_p(t)$ is formed by modulated the phase of $c(t)$: $c_p(t) = \cos(\omega_c t + \phi_c + m(t))$. The *amplitude modulated signal* $c_a(t)$ is given by $c_a(t) = m(t)c(t)$, where here its assumed $m(t) > 0$. In either case, the aim of the demodulation problem is to recover the message $m(t)$ from the modulated signal.

Example 2.2.7 (Amplitude demodulation with the analytic signal). *The case of amplitude modulation is a direct instance of Bedrosian's theorem:*

$$\begin{aligned} \mathcal{H}c_a(t) &= m(t)\mathcal{H}(c(t)) \\ (c_a)_A(t) &= m(t)\left[\cos(\omega_c t + \phi_c) + \mathbb{i}\sin(\omega_c t + \phi_c)\right] \\ &= m(t)\mathrm{e}^{\mathbb{i}\omega_c t + \phi_c}. \end{aligned}$$

Taking the modulus of the analytic signal yields $m(t)$ as desired.

Example 2.2.8 (Phase demodulation with the analytic signal). *The case of phase modulation is also an instance of Bedrosian's theorem, though it takes a bit more work to see. First, let $c_p(t)$ be the phase modulated signal as described previously, and rewrite this using trigonometric identities:*

$$\begin{aligned} c_p(t) &= \cos(\omega_c t + \phi_c + m(t)) \\ &= \cos(m(t))\cos(\omega_c t + \phi_c) - \sin(m(t))\sin(\omega_c t + \phi_c). \end{aligned}$$

Then, because $\frac{\mathrm{d}}{\mathrm{dt}}\sin(m(t)) = \cos(m(t))\frac{\mathrm{d}}{\mathrm{dt}}m(t)$, it is clear that $\left|\frac{\mathrm{d}}{\mathrm{dt}}\sin(m(t))\right| \leq |\frac{\mathrm{d}}{\mathrm{dt}}m(t)|$, and, similarly, $\left|\frac{\mathrm{d}}{\mathrm{dt}}\cos(m(t))\right| \leq |\frac{\mathrm{d}}{\mathrm{dt}}m(t)|$. So long as $m(t)$ is sufficiently band-limited, as is assumed, the Hilbert transform of $c_p(t)$ yields the following:

$$\begin{aligned} \mathcal{H}c_p(t) &= \cos(m(t))\mathcal{H}\cos(\omega_c t + \phi_c) - \sin(m(t))\mathcal{H}\sin(\omega_c t + \phi_c) \\ &= \cos(m(t))\sin(\omega_c t + \phi_c) + \sin(m(t))\cos(\omega_c t + \phi_c). \end{aligned}$$

Therefore, the analytic signal of the phase-modulated signal $c_p(t)$ is given by

$$\begin{aligned}(c_p)_A(t) &= \cos(m(t))\mathrm{e}^{\mathbb{i}\omega_c t+\phi_c} + \mathbb{i}\sin(m(t))\mathrm{e}^{\mathbb{i}\omega_c t+\phi_c}\\ &= \mathrm{e}^{\mathbb{i}(m(t)+\omega_c t+\phi_c)}.\end{aligned}$$

Computing the argument of the analytic signal yields $(m(t)+\omega_c t+\phi_c) \mod 2\pi$*, so, after subtracting the know phase of the carrier signal,* $\omega_c t + \phi_c$*, the message* $m(t)$ *is recovered modulo* 2π*. If the resulting phase unwrapping problem can be solved, then the message* $m(t)$ *can be fully recovered.*

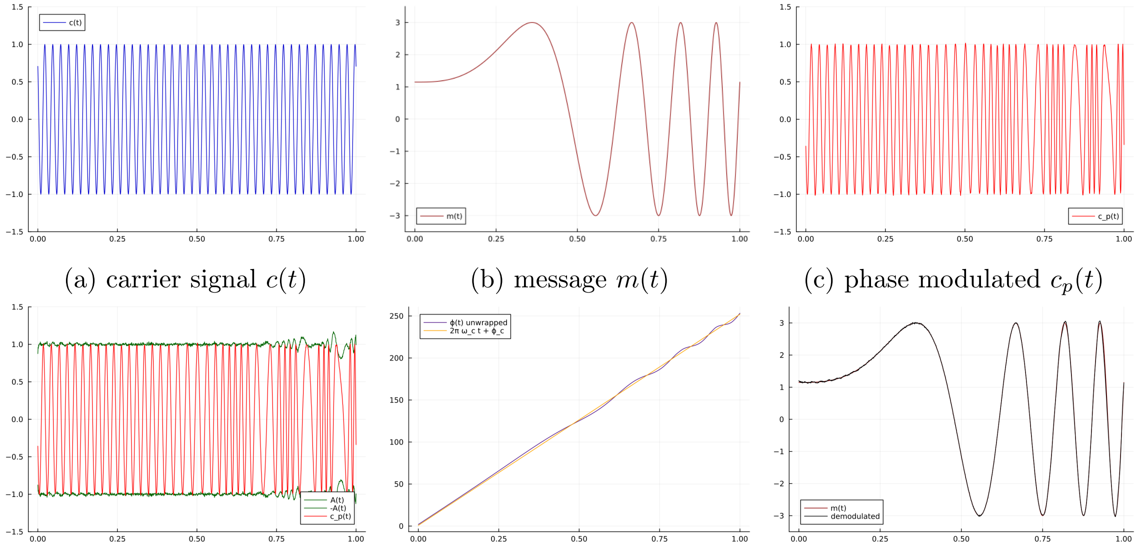
An example of phase and amplitude demodulation using the analytic signal is given in Example 2.2.9.

Example 2.2.9. *Consider the carrier signal* $c(t) = \cos(2\pi\omega_c t + \phi_c)$ *and the message* $m(t) = \alpha_m \sin\big(2\pi\omega_m t^3 + \phi_m\big)$ *on the interval* $[0, 1]$ *with* $\omega_c = 40, \phi_c = \pi/4, \alpha_m = 3, \omega_m = 4$*, and* $\phi_m = \pi/8$*. Notice here that the phase of* $m(t)$ *is not linear, therefore the maximum value of* $m'(t)$ *should be inspected to ensure* $m(t)$ *varies slowly enough with respect to* $c(t)$*. In this case we have* $|m'(t)| \leq 6\pi\alpha_m\omega_m$*, and for the parameters given this means* $|m'(t)| \leq 72\pi$*, whereas* $|c'(t)| < 2\pi\omega_c = 80\pi$*, meaning Bedrosian's theorem should guarantee the phase recovered will correspond to that seen in Example 2.2.8. The signal is sampled well above the Nyquist sampling rate. Indeed, Figure 2.3 shows that we successfully recover* $m(t)$ *using the theory outlined here, where (a)-(c) shows* $c(t)$*,* $m(t)$ *and* $c_p(t)$ *respectively, (d) shows that the envelope given by* $\pm A(t)$ *provides an envelope for* c_p*, where* $A(t)$ *is the amplitude of* $(c_p)_A(t)$*, (e) shows the unwrapped phase, and (f) the successfully demodulated message.*

The main idea of the analytic signal $f_A(t) = f(t) + \mathbb{i}\mathcal{H}(t)$ is to derive the form $A(t)\mathrm{e}^{\mathbb{i}\phi(t)}$, where the features A and ϕ form a *invariance-equivariance decomposition* which decomposes structural (phase) information from energy (amplitude) information:

- the phase depends only on the local structure, e.g., it is invariant to scaling, and equivariant to phase shifts;
- the amplitude depends only on the local energy, e.g., equivariant to scaling, invariant to phase shifts;

More specifically, this yields a *split of identity*, a invariance-equivariance decomposition where the features give a complete description of the signal. The analytic signal f_A, can be written down



(a) carrier signal $c(t)$

(b) message $m(t)$

(c) phase modulated $c_p(t)$

(d) envelope given by $\pm A(t)$

(e) unwrapped phase

(f) demodulated message

FIGURE 2.3. Example of a phase demodulation problem in 1D solved using the analytic signal, relying on Bedrosian's theorem.

in terms of this derived amplitude and phase as $f_A(t) = A(t)e^{\ss\phi(t)}$, and then f is recovered as $\mathrm{Re} f_A(t) = A(t)\cos(\phi(t))$. These are considered the *instantaneous amplitude and phase (IAP) representations* of f_A and f respectively.

We can use this property along with Fourier series representation of an arbitrary signal in $L^1(\mathbb{R}) \cap L^2(\mathbb{R})$ as we demonstrate in Example 2.2.5. Though this is a nice way to understand the action of the Hilbert transform and the resulting analytic signal, in order to derive meaningful local amplitude, phase, and frequency information the analytic signal should be formed after applying an appropriate bandpass filter to the original signal. Because of this, the IAP representation given by the analytic signal should be coupled with a multiscale analysis. This idea leads naturally to the notion of a multiscale phase and amplitude, where a local amplitude and phase is selected from local amplitudes and phases and multiple scales, which is explored more in Chapter 3.

2.2.1. Other 1D instantaneous amplitude and phase methods. For a bandlimited signal f, the analytic signal can extract meaningful local features from f_A by defining the instantaneous amplitude, $A(t) = |f_A|$, and instantaneous phase $\varphi(t) = \arctan \mathcal{H}f, f$. If φ can be successfully unwrapped to produce a smooth phase function $\tilde{\varphi}$, then the instantaneous frequency is given by $\frac{\partial \tilde{\varphi}}{\partial t}$. It should be mentioned that in 1D, another strategy has provided an elegant solution to phase and amplitude decomposition which avoids this issue of a wrapped phase This is accomplished by using the *Blaschke product* [**21**, **22**, **23**, **65**, **91**]. This approach is a different factorization of the analytic signal. Unfortunately, so far higher dimensional extensions have remained elusive. The Blaschke product is derived from considering the analytic signal $f_A(t)$ to be the boundary value of a holomorphic function on the unit disc in the complex plane. With this boundary data given, we have a unique solution $F(z)$ on $|z| < 1$, namely the F which satisfies $\mathit{\Delta} F = 0$, represents the and $F|_{|z|=1} = f_A(t)$. Here $\mathit{\Delta}$ is the Laplace operator defined by $\mathit{\Delta} F = \frac{\partial^2 F}{\partial x^2} + \frac{\partial^2 F}{\partial y^2}$. We then know the existence of the factorization $F(z) = B(z)G(z)$, where

$$B(z) = z^N \prod_{k=1}^{M} \left(\frac{z - \alpha_k}{1 - \overline{\alpha}_k z} \cdot \frac{\overline{\alpha}_k}{|\alpha_k|} \right),$$

where $\{\alpha_k\}_{k=1}^{M}$ are the non-zero roots of $F(z)$, and where $G(z)$ has no roots in $|z| < 1$. We consider $B(z)$ to contain the phase information, and $G(z)$ to contain the amplitude information. Indeed, if $B(\mathrm{e}^{\mathbb{i}t}) = \mathrm{e}^{\mathbb{i}\varphi(t)}$ for some φ we can show that φ is non-decreasing, so that the instantaneous frequency is non-negative and we need not solve a phase unwrapping problem in order to compute this instantaneous frequency. Additionally, some interesting work that has been done in *adaptive Fourier decomposition* [**68**, **69**], which studies a non-linear approximation of positive frequency signals based on greedy algorithms, seeking to represent these signals with very few terms and guaranteeing fast convergence.

2.3. An Introduction to the Monogenic Signal

The notion of a two dimensional (2D) analytic signal has been thoroughly studied [**14**], [**11**], [**33**], [**39**], so that both its uses and limitations in providing interpretable features of 2D signals are well understood. There are two common notions of the analytic signal in 2D, both starting with an image, or a 2D signal, $f : \mathbb{R}^2 \to \mathbb{R}$. The first is the *quaternionic analytic signal* (QAS), [**11**] defined using directional 1D Hilbert transforms along each spatial axis, along with a product of these directional transforms, in order to construct a 2D extension of the Hilbert transform. The second is the monogenic image [**33**], which is defined in terms of the *Riesz transform*, which can be seen as a natural extension of the Hilbert transform in higher dimensions. Explicitly, the Riesz transform is a convolutional operator that gives $\widehat{\mathcal{R}f} = -\mathbb{i}\frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|_2}\hat{f}$. (To be clear, this relates to the sign function in 1D because $\operatorname{sign}(\omega) = \frac{\omega}{|\omega|}$ for $\omega \neq 0$, and 0 when $\omega = 0$.) As written, this is a vector valued operator, and is easily generalized to n dimensions. Due to the natural embedding of the analytic signal into $\mathbb{C}$, however, the 2D Riesz transform is often considered as a quaternion-valued operator by identifying $\boldsymbol{i}$ and $\boldsymbol{j}$ with the standard basis vectors $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$, and writing:

$$\begin{aligned} \mathcal{R}f(\boldsymbol{x}) &= \boldsymbol{i}\mathcal{R}_1 f(\boldsymbol{x}) + \boldsymbol{j}\mathcal{R}_2 f(\boldsymbol{x}), \text{ where} \\ \widehat{\mathcal{R}_k f}(\boldsymbol{\omega}) &= -\mathbb{i}\frac{\omega_k}{\|\boldsymbol{\omega}\|_2}\hat{f}(\boldsymbol{\omega}) \text{ for } k = 1, 2. \end{aligned} \tag{2.8}$$

Using the hypercomplex variables $\boldsymbol{i}, \boldsymbol{j} \in \mathbb{Q}$ and the standard $\mathbb{i} \in \mathbb{C}$ is cumbersome. This is due to some limitations of working in the quaternions, and shortly we will define a Clifford algebra (into which the quaternions embed into naturally) where this issue can be avoided. This idea is presented in [**33**]. For now, consider the Riesz transforms $\mathcal{R}_k$ in the usual sense, and $\boldsymbol{i}$ and $\boldsymbol{j}$ as the standard basis vectors.

Definition 2.3.1 (The Riesz Transform). *The Riesz Transform $\mathcal{R} : L^1(\mathbb{R}^2;\mathbb{R}) \cap L^2(\mathbb{R}^2;\mathbb{R}) \to L^1(\mathbb{R}^2;\mathbb{Q}) \cap L^2(\mathbb{R}^2;\mathbb{Q})$ is defined by (2.8). In the spatial domain the convolutional operators $\mathcal{R}_k$ are given by:*

$$\mathcal{R}_k f(\boldsymbol{x}) = \frac{1}{2\pi} p.v. \int_{\mathbb{R}^2} f(\boldsymbol{y}) \frac{y_k - x_k}{\|\boldsymbol{y} - \boldsymbol{x}\|^3} \mathrm{d}\boldsymbol{y}.$$

From this we see the full Riesz transform is defined as:

$$\mathcal{R}f(\boldsymbol{x}) = \frac{1}{2\pi} p.v. \int_{\mathbb{R}^2} f(\boldsymbol{y}) \left(\boldsymbol{i}\frac{y_1 - x_1}{\|\boldsymbol{y} - \boldsymbol{x}\|^3} + \boldsymbol{j}\frac{y_2 - x_2}{\|\boldsymbol{y} - \boldsymbol{x}\|^3} \right) \mathrm{d}\boldsymbol{y}. \tag{2.9}$$

Definition 2.3.2 (The Monogenic Signal, instantaneous amplitude, phase, and orientation). *The monogenic signal of an image $f : \mathbb{R}^2 \to \mathbb{R}$ is given by the quaternion-valued signal*

$$f_M(\boldsymbol{x}) = f(\boldsymbol{x}) + \mathcal{R}f(\boldsymbol{x}). \tag{2.10}$$

The instantaneous amplitude $A(\boldsymbol{x})$, phase $\phi(\boldsymbol{x})$, and orientation $\boldsymbol{n}(\boldsymbol{x})$ are given by

$$A(\boldsymbol{x}) = |f_M(\boldsymbol{x})|, \quad \boldsymbol{n}(\boldsymbol{x}) = \frac{\mathcal{R}f(\boldsymbol{x})}{|\mathcal{R}f(\boldsymbol{x})|}, \quad \varphi(\boldsymbol{x}) = \arctan\left(|\mathcal{R}f(\boldsymbol{x})|, f(\boldsymbol{x})\right).$$

Example 2.3.3. *Let $f(\boldsymbol{x}) = \cos(\omega(\boldsymbol{x} \cdot \boldsymbol{n}))$ for $\boldsymbol{x}, \boldsymbol{n} \in \mathbb{R}^2$ where $\boldsymbol{n} = (n_1, n_2) = n_1\boldsymbol{i} + n_2\boldsymbol{j}$ is a fixed unit vector. This is a plane wave with orientation $\boldsymbol{n}$ and spatial frequency ω. Again considering* *the Fourier representation of* $f(\boldsymbol{x})$ *gives* $\hat{f}(\boldsymbol{\omega}) = \frac{1}{2}\left[\delta(\omega\boldsymbol{n}) + \delta(-\omega\boldsymbol{n})\right]$*, hence*

$$\begin{aligned}\widehat{\mathcal{R}_k f}(\boldsymbol{\omega}) &= -\frac{\mathbb{i}}{2}\frac{\omega n_k}{\|\omega\boldsymbol{n}\|}\hat{f}(\boldsymbol{\omega}) \\ &= -\frac{\mathbb{i}n_k}{2}\left(\delta(\omega\boldsymbol{n}) - \delta(-\omega\boldsymbol{n})\right).\end{aligned}$$

Therefore we have

$$\mathcal{R}_k f(\boldsymbol{x}) = n_k \sin(\omega\boldsymbol{x}\cdot\boldsymbol{n}).$$

It follows that $\mathcal{R}f(\boldsymbol{x}) = (n_1\boldsymbol{i} + n_2\boldsymbol{j})\sin(\omega\boldsymbol{n})$*. Identifying the vector* $\boldsymbol{n}$ *with the quaternion* $n_1\boldsymbol{i}+n_2\boldsymbol{j}$ *gives* $\mathcal{R}f(\boldsymbol{x}) = \boldsymbol{n}\sin(\omega(\boldsymbol{x}\cdot\boldsymbol{n}))$*. In this way the monogenic signal is given by*

$$f_M(\boldsymbol{x}) = \cos(\omega(\boldsymbol{x}\cdot\boldsymbol{n})) + \boldsymbol{n}\sin(\omega(\boldsymbol{x}\cdot\boldsymbol{n})).$$

In this case the instantaneous amplitude is equal to 1 everywhere, whereas the instantaneous phase is equal to $\omega(\boldsymbol{x}\cdot\boldsymbol{n}) \mod 2\pi$ *and the instantaneous orientation is equal to* $\boldsymbol{n}$*. Figure 2.4 demonstrates this simple example of the amplitude, phase, and orientation of an amplitude modulated plane wave. Here the underlying plane wave is the function*

$$f(\boldsymbol{x}) = \cos\left(\omega(x_1^j\cos\theta + x_2^j\sin\theta)\right),$$

which is modulated in amplitude by the function

$$g(\boldsymbol{x}) = \left(\cos\theta(x_1^j + \pi) + \sin\theta(x_2^j + \pi)\right)^2,$$

for $\omega = 16, \theta = \pi/4$*, and* $x_1^j, x_2^j = -\pi + 2\pi j/512$ *for* $j = 0, \ldots, 511$*. A circular mask is also applied to mitigate edge effects in this case.*

Considering a 2D function of the form $f(\boldsymbol{x}) = A(\boldsymbol{x})\cos(\phi(\boldsymbol{x}))$, where $A(\boldsymbol{x}) \geq 0$ and varies slowly with respect to $\phi(\boldsymbol{x})$, Larkin **[56]** showed that the Riesz transform (or *spiral phase quadrature transform*, as it is referred to in his work) obeys the asymptotic Bedrosian Principle:

$$\mathcal{R}(A(\boldsymbol{x})\cos(\varphi(\boldsymbol{x}))) \approx A(\boldsymbol{x})\mathcal{R}(\cos(\varphi(\boldsymbol{x}))), \tag{2.11}$$

and, further, that

$$\mathcal{R}(\cos(\varphi(\boldsymbol{x}))) \approx \boldsymbol{n}(\boldsymbol{x})\sin(\varphi(\boldsymbol{x})). \tag{2.12}$$

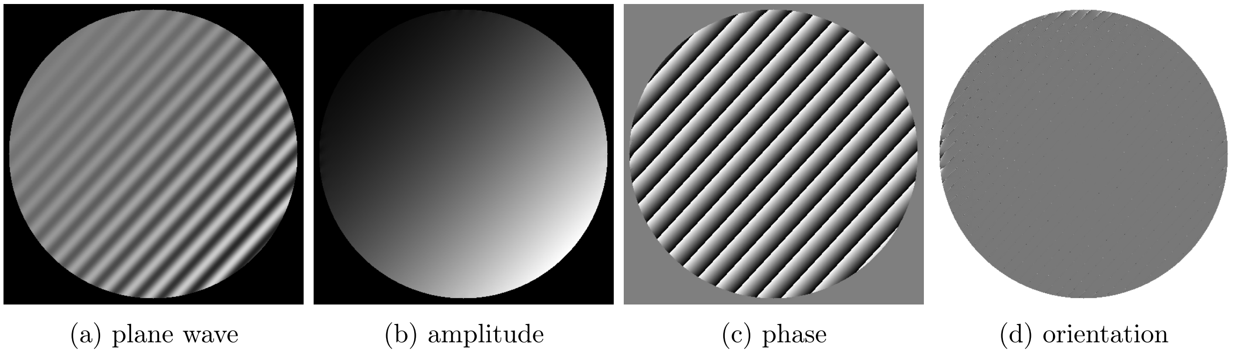

(a) plane wave (b) amplitude (c) phase (d) orientation

FIGURE 2.4. (a) a plane wave with constant orientation and increasing amplitude along the direction of the wave front. (b)-(c) the features of the monogenic signal (b) amplitude, (c) phase, and (d) orientation.

This means that the local amplitude, orientation, and phase defined in Definition 2.3.2 are in some sense robust. This is shown by considering a more complicated signal: a complex chirp with the same amplitude modulation as in Example 2.3.3, which is shown in Figure 2.5. In this case the underlying oscillatory signal is given by

$$f(\boldsymbol{x}) = \cos\left(a^2\omega'(x_1^j+\pi)^2+(x_2^j+\pi)^2\right)$$

with $a = 0.2, \omega' = 64$, and x_1^j, x_2^j as in Example 2.3.3.

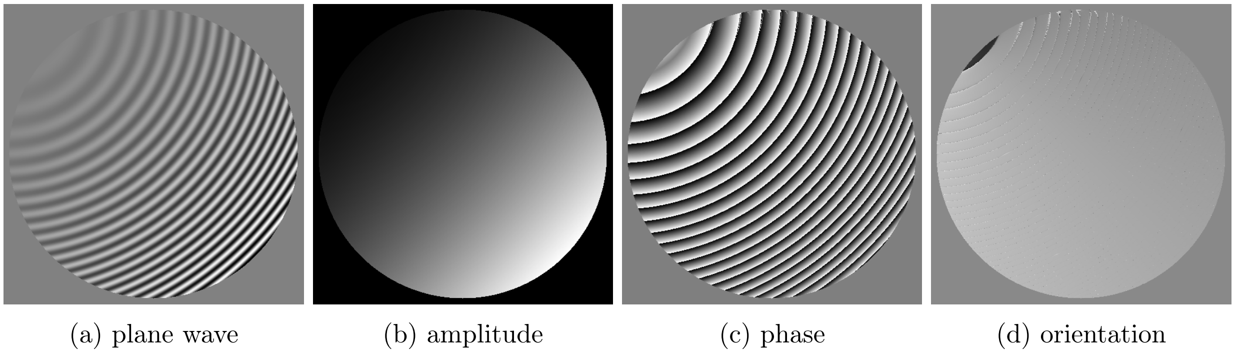

(a) plane wave (b) amplitude (c) phase (d) orientation

FIGURE 2.5. (a) a parabolic chirp signal with constant orientation and the same increasing amplitude as in 2.3.3. (b)-(c) the features of the monogenic signal (b) amplitude, (c) phase, and (d) orientation.

More discourse about the monogenic signal and some similar 2D signal models is discussed in Chapter 3.

CHAPTER 3

A Structurally Coherent Spatial Phase Estimate with Monogenic Wavelets

Many problems in imaging science rely on spatial phase measurements, e.g., 2D interferometry, interferometric SAR (InSAR), and require a preprocessing step to estimate the true spatial phase of an image or set of images [**93**]. Any improvement to this estimate will thus improve downstream analysis. A standard approach for estimating spatial phase of images is to use the phase of the *monogenic signal* [**55**] [**94**]. The first improvement to this estimate is to produce a multiscale monogenic phase estimate, see Kaseb et al. [**50**], for instance, which makes use of isotropic wavelets [**40**] [**84**], and provides a robust phase estimate in the presence of image corruption. **The main contributions of this chapter are: 1)** to employ the *structure multivector* (SMV) in place of the monogenic signal in order to extract a more robust feature set at any given scale; and **2)** to define a novel quality measure at each scale, based on the features of the SMV, in order to determine the optimal local feature set around a given point in an image. Several experiments on synthetic images are performed to showcase the application of the our multiscale phase estimation, and further solve a phase and amplitude demodulation problem in 2D to display the utility of this estimate. Additionally, the multiscale phase estimate is used to solve a fine-scale fingerprint registration problem as described in [**26**]. Lastly, a Julia module that includes the code needed to reproduce any figures and experiments shown in this chapter is provided, as well as standalone functions to perform the multiscale phase estimation: https://gitlab.com/briancknight/SSVM2025.

The rest of the chapter is organized as follows. In Section 3.1 the standard signal model used to describe fringe patterns is discussed, and the monogenic signal, steerable wavelets, and the structure multivector (SMV) are all defined. Section 3.2 describes the novel multiscale phase estimate derived from the SMV features at each scale, and Section 3.3 concludes with numerical experiments showcasing the improved phase estimation as well as improved accuracy in 2D phase and amplitude demodulation tasks, including one used in fine-scale fingerprint registration. Section 3.6

provides a brief conclusion to the chapter. In the additional appendix, Appendix A provides more details in the construction of the structure multivector and provides an error analysis for the feature set of the SMV when the underlying signal model it assumes is violated.

3.1. Signal Models

3.1.1. The i1D Signal Model and the Monogenic Signal. As discussed in Chapter 2, in 1D signal processing, a real-valued signal can be extended to a complex-valued signal via the Hilbert transform, and this complex-valued extension can provide useful insight into the signal's local amplitude and frequency. Further, in 2D, the monogenic signal is a quaternion-valued signal that extends the real-valued 2D signal via the Riesz transform, the appropriate 2D extension of the Hilbert transform. Here we assume the original signal obeys the signal model

$$f(\boldsymbol{x}) = A(\boldsymbol{x})\cos(\boldsymbol{n}(\boldsymbol{x})\cdot\boldsymbol{x}), \quad \boldsymbol{x}\in\mathbb{R}^2 \tag{3.1}$$

where $A(\boldsymbol{x}) \geq 0$ is the local amplitude function, and $\boldsymbol{n}(\boldsymbol{x})$ is the local orientation, and $\varphi(\boldsymbol{x}) = \boldsymbol{n}(\boldsymbol{x})\cdot\boldsymbol{x}$ is the local phase function, and $A(\boldsymbol{x})$ is assumed to vary slowly with respect to the $\cos(\boldsymbol{n}(\boldsymbol{x})\cdot\boldsymbol{x})$. A may also be referred to as the *local energy*, and the tuple $(\boldsymbol{n}(\boldsymbol{x}), \varphi(\boldsymbol{x}))$ as the *local structure*.

The monogenic signal given by

$$f_M(\boldsymbol{x}) = f(\boldsymbol{x}) + \boldsymbol{i}\mathcal{R}_1 f(\boldsymbol{x}) + \boldsymbol{j}\mathcal{R}_2 f(\boldsymbol{x}),$$

and more compactly written as $f + \mathcal{R}f(\boldsymbol{x}) = \boldsymbol{i}\mathcal{R}_1 f(\boldsymbol{x}) + \boldsymbol{j}\mathcal{R}_2 f(\boldsymbol{x})$, where $\mathcal{R}$ is the (total) Riesz transform. Larkin [**56**] showed that for signals of the *intrinsically 1D* (i1D) model described above, the Riesz transform obeys the asymptotic Bedrosian Principle given in Equations (2.11) and (2.12).

In the case of $\varphi(\boldsymbol{x}) = 2\pi\omega\boldsymbol{n}\cdot\boldsymbol{x}$ for some unit vector $\boldsymbol{n}$ and frequency ω, i.e., a plane wave, so long as $\hat{A}(\boldsymbol{u}) = 0$ for any $\boldsymbol{u}$ such that $|\boldsymbol{u}| > \omega$, the Riesz transform performs the correct quadrature shift for phase estimation of i1D signals in two dimensions.

To be clear, given a signal f of the form described in Equation (3.1), the monogenic signal of f is given by

$$f_M(\boldsymbol{x}) = A(\boldsymbol{x})\left(\cos(\varphi(\boldsymbol{x})) + \boldsymbol{n}(\boldsymbol{x})\sin(\varphi(\boldsymbol{x}))\right),$$

thus we can recover the local amplitude, local orientation, and local phase via

$$A(\boldsymbol{x}) = |f_M(\boldsymbol{x})|, \quad \boldsymbol{n}(\boldsymbol{x}) = \frac{\mathcal{R}f(\boldsymbol{x})}{|\mathcal{R}f(\boldsymbol{x})|}, \quad \varphi(\boldsymbol{x}) = \arctan\left(|\mathcal{R}f(\boldsymbol{x})|, f(\boldsymbol{x})\right).$$

This feature set is considered to be a *split of identity*, in that it separates a signal into independent local features. Specifically, the local structure is invariant to scaling of the local energy, and the local energy is invariant to phase shifts in the local structure. This allows for useful image processing steps, such as equalization of brightness [**40**] as discussed in Section 3.2.1, or 2D phase and amplitude demodulation, as discussed later in this chapter. Similarly, local amplitude information can be used in order to determine important features. This approach is particularly useful when the monogenic signal is paired with an isotropic wavelet decomposition [**50**], which is outlined in the next section.

3.1.2. Steerable Wavelet Frames using Riesz Transforms. The Riesz transform $\mathcal{R}$ is steerable, meaning that if R_θ is a 2D rotation matrix defined by Equation (2.1), then $\mathcal{R}R_\theta f(\boldsymbol{x}) = R_\theta \mathcal{R} f(\boldsymbol{x})$. It also commutes with shifts and dilations so that applying any one of these operations to the signal f can be done before or after computing the monogenic extension. As far as wavelet analysis is concerned, this allows us to construct monogenic wavelets simply by constructing a real isotropic wavelet and then computing its Riesz transform.

Based on this property, Held et al. [**40**] construct steerable wavelet frames for n-dimensional (nD) signals which, for a given image $f \in \mathbb{R}^{2^L \times 2^L}$, yields the decomposition into $L \cdot K$ scales, where L is the number of dyadic scales, and K the number of subscales used for each dyadic scale. For example, the small See Example 2.1.6 given in Section 2.1.2. The component of the decomposition are denoted by $d_{k,l}(\boldsymbol{x})$ for $k = 1, \ldots K$, $l = L', \ldots, L$, where $L' \geq 1$ is the smallest dyadic scale used (typically $L' = 3$), and an approximation component a_L which contains any remaining low-frequency information. Naturally, each scale can be extended via the Riesz transform to obtain:

$$(d_{k,l}(\boldsymbol{x}))_M = A_{k,l}(\boldsymbol{x})\left[\cos(\varphi_{k,l}(\boldsymbol{x})) + \boldsymbol{n}_{k,l}(\boldsymbol{x})\sin(\varphi_{k,l}(\boldsymbol{x}))\right],$$

where $A_{k,l}, \varphi_{k,l}$, and $\boldsymbol{n}_{k,l}$ denote the local amplitude, local phase, and local orientation vector respectively of $d_{k,l}$ for $k = 1, \ldots, K$ and $l = L', \ldots, L$. Additionally, $\theta_{k,l}$ denotes the local orientation. If $K = 1$, each dyadic scale is decomposed via complementary high and low pass filters. For $K > 1$, the dyadic scale is decomposed into $K - 1$ band-pass components, and a high and low frequency component. The specific construction can be found used in [**40**]. Figure 3.1 depicts the filters for $K = 2$, while Figure 3.2 shows the instantaneous amplitude and phase (IAP) representation of the high frequency and band-pass component provided by the structure multivector which is discussed in the next section.

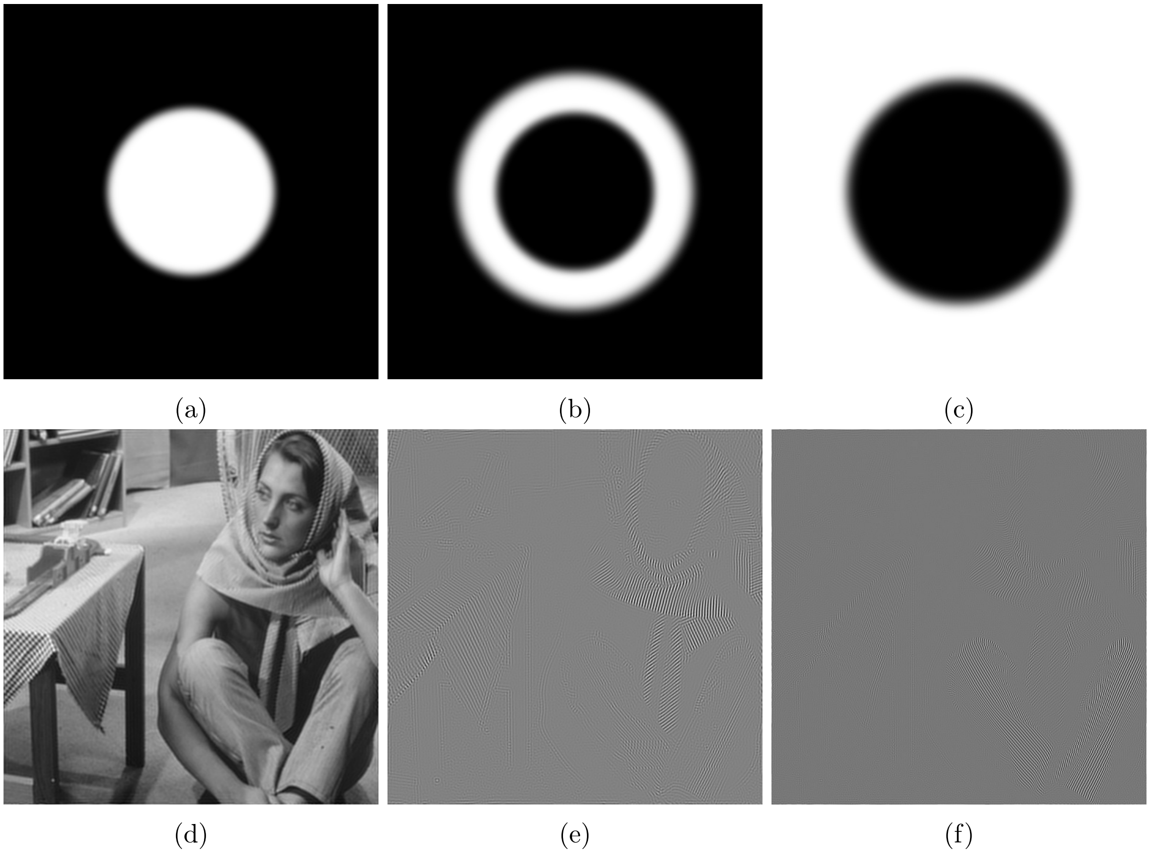
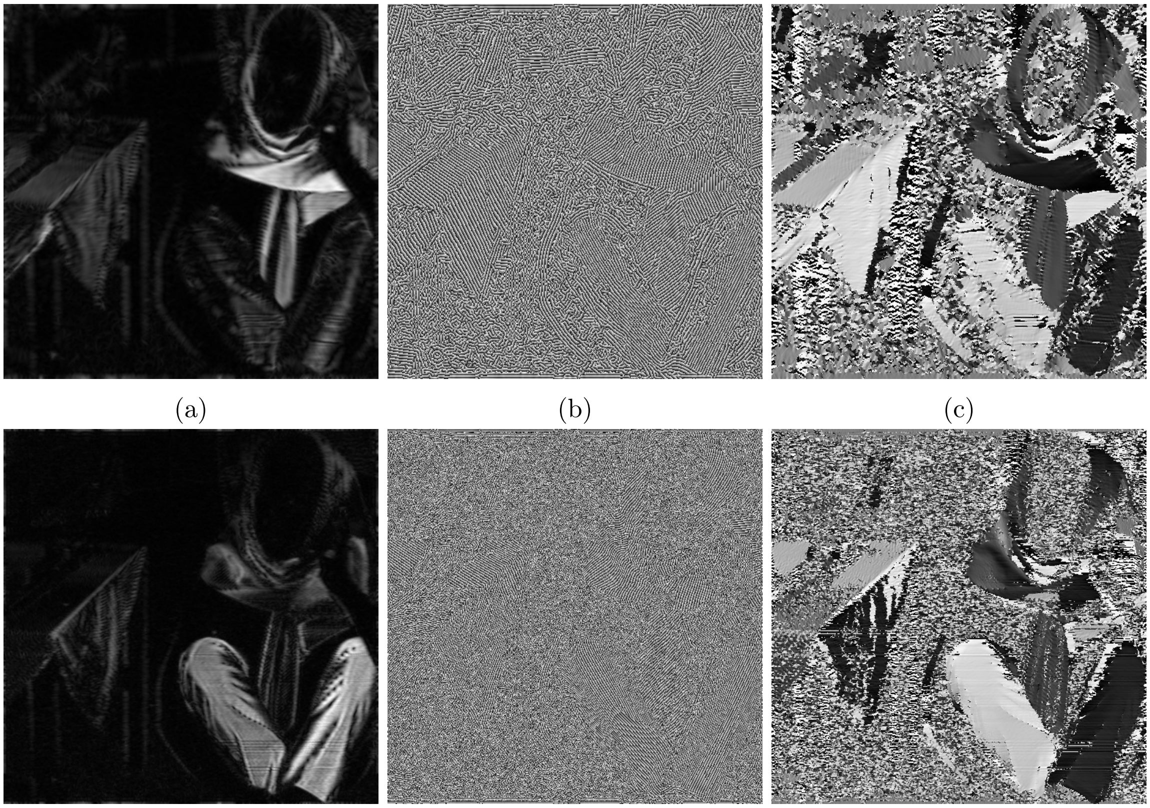


FIGURE 3.1. Left: Isotropic wavelet frames, (a) low-pass $h_{1,1}$, (b) band-pass $h_{1,2}$, (c) high-pass $h_{1,3}$. Applied to Barbara: (d) low-pass, (e) band-pass, (f) high-pass.

Consider the instantaneous amplitude images shown in Figure 3.2(a)(d). Where local amplitude is large in these band-pass images is precisely where coherent local structures are found, such as along the scarf, the pants, or the tablecloth. That is, the local amplitude indicates the local structural information is relevant, and if that local structural information follows the given signal model, then these features provide insight into the whole image. For example, what is the orientation of the fringes on Barbara's right leg, or on her scarf?

3.1.3. The i2D Signal Model and the Structure Multivector. Two-dimensional signals can, of course, vary in two or more orientations in any given local patch, and further work has



(a) (b) (c)

(d) (e) (f)

FIGURE 3.2. IAP representation of isotropic wavelet scales, (a) $A_{1,2}(\boldsymbol{x})$, (b) $\Phi_{1,2}(\boldsymbol{x})$, (c) $\theta_{1,2}(\boldsymbol{x})$, (d) $A_{1,3}(\boldsymbol{x})$, (e) $\Phi_{1,3}(\boldsymbol{x})$, (f) $\theta_{1,3}(\boldsymbol{x})$.

been done using hypercomplex signal processing that can deal with these cases. One extension, the *structure multivector* (SMV), was introduced along with the monogenic signal in the PhD dissertation of Felsberg [**33**]. It is designed to deal with signals of the form

$$f(\boldsymbol{x}) = f_1(\boldsymbol{n}(\boldsymbol{x}) \cdot \boldsymbol{x}) + f_2(\boldsymbol{n}(\boldsymbol{x})^{\perp} \cdot \boldsymbol{x}), \tag{3.2}$$

These are *intrinsically 2D* (i2D) signals with two orientations in each local patch that are orthogonal to one another.

The features of the SMV will essentially be that of two monogenic signals, and to accommodate these additional features the SMV lives in a larger dimensional Clifford algebra, $C\ell_{3,0}$ which subsumes the quaternions. This algebra is generated by the orthonormal basis $\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3$ satisfying the relations $\boldsymbol{e}_i\boldsymbol{e}_j + \boldsymbol{e}_j\boldsymbol{e}_i = 2\delta_{ij}$, and consists of $2^3 = 8$ elements: $1, \boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3, \boldsymbol{e}_{12}, \boldsymbol{e}_{23}, \boldsymbol{e}_{31}, \boldsymbol{e}_{123}$. In general, $\boldsymbol{e}_{i_1}\boldsymbol{e}_{i_2} \cdots \boldsymbol{e}_{i_n} := \boldsymbol{e}_{i_1 i_2 \cdots i_n}$. For more information on Clifford algebras and the construction of the SMV see [29, 33] and Appendix A. Only the minimum details needed to construct the SMV are give here. Consider an image of the form $\mathbf{f} : \mathbb{R}^2 \to \boldsymbol{e}_3\mathbb{R}$, $\mathbf{f}(\boldsymbol{x}) = f(x, y)\boldsymbol{e}_3$.

The corresponding structure multivector (SMV) is given by

$$
\begin{aligned}
M_S\mathbf{f}(\boldsymbol{x}) &= \big[\mathbf{f}(\boldsymbol{x}) + (h_2^1 * \mathbf{f})(\boldsymbol{x})\big] + \boldsymbol{e}_3 \left[(h_2^2 * \mathbf{f})(\boldsymbol{x}) + (h_2^3 * \mathbf{f})(\boldsymbol{x})\right] \\
&= M_0 + M_1\boldsymbol{e}_1 + M_2\boldsymbol{e}_2 + M_3\boldsymbol{e}_3 + M_{23}\boldsymbol{e}_{23} + M_{31}\boldsymbol{e}_{31} + M_{12}\boldsymbol{e}_{12}.
\end{aligned} \tag{3.3}
$$

The explicit definitions of these functions are given below, where $\boldsymbol{x} = x\boldsymbol{e}_1 + y\boldsymbol{e}_2$:

$$
\begin{aligned}
M_1 &= \frac{x}{2\pi|\boldsymbol{x}|^3} * f(\boldsymbol{x}), & M_2 &= \frac{y}{2\pi|\boldsymbol{x}|^3} * f(\boldsymbol{x}), & M_3 &= f(\boldsymbol{x}), \\
M_{23} &= \frac{3(3x^2y - y^3)}{2\pi|\boldsymbol{x}|^5} * f(\boldsymbol{x}), & M_{31} &= \frac{3(3xy^2 - x^3)}{2\pi|\boldsymbol{x}|^5} * f(\boldsymbol{x}), \\
M_0 &= \frac{-2(x^2 - y^2)}{2\pi|\boldsymbol{x}|^4} * f(\boldsymbol{x}), & M_{12} &= \frac{-4xy}{2\pi|\boldsymbol{x}|^4} * f(\boldsymbol{x}).
\end{aligned}
$$

Here $h_2^1 = \mathcal{R}$ denotes the Riesz transform, h_2^3 is the composition of h_2^2 and $h_{2,}^1$, where $H_2^2(\boldsymbol{u}) = \frac{(u^2-v^2)+2uv\boldsymbol{e}_{12}}{\boldsymbol{u}^2}$ is the Fourier transform of h_2^2. H_2^2 responds only to even signals, and any two perpendicular vectors $\boldsymbol{n}$ and $\boldsymbol{n}^\perp$ become antiparallel after action by H_2^2, which means that an even signal according to the model given in Equation (3.2) will yield a response to H_2^2 whose argument is precisely twice that of the main orientation of $\boldsymbol{n}$. Specifically, we can calculate the orientation $\boldsymbol{n}$ given a signal of the form $f(\boldsymbol{x}) = A\cos(\boldsymbol{n} \cdot \boldsymbol{x}) + B\cos\big(\boldsymbol{n}^\perp \cdot \boldsymbol{x}\big)$ directly from this response.

To handle odd structures, or structures of the form $f(\boldsymbol{x}) = A\sin(\boldsymbol{n} \cdot \boldsymbol{x}) + B\sin\big(\boldsymbol{n}^\perp \cdot \boldsymbol{x}\big)$, the Riesz transform of h_2^2 is taken to yield h_3^2. The product of the Riesz response and the response of the third order harmonic estimate provides good orientation estimation for these odd structures, hence the average of these two arguments provides a robust orientation estimate of the structure multivector, as given in Felsberg's dissertation [33]:

$$
\theta_e = \frac{1}{4}\arg\left[(M_0 + M_{12}I_2)^2 + (M_1 + M_2I_2)(M_{31} - M_{23}I_2)\right], \tag{3.4}
$$

where $I_2 = \boldsymbol{e}_{12}$ acts as the imaginary unit $\boldsymbol{i}$.

In [33] the author shows that the extended signal model provides a more robust orientation estimate than that of the monogenic signal. This is further confirmed in [62]. In addition to these facts, we show that: 1) the feature set of the SMV is robust even to i2D signals which violate the orthogonality constraint; and 2) if one of the local i1D signal dominates the local energy, then we can estimate the corresponding orientation well even in the case of large deviation from this constraint. See Appendix A for details.

With this orientation estimate it is then possible to construct a pair of local angular filters that decompose a signal f into two i1D signals, which then yields two local amplitudes, two local orientations, and two local phases that can be used for further processing. Again, see [**33**] or Appendix A for full details. The output of the local angular filtering is two complex i1D signals which we denote by $F_1(\boldsymbol{x})$ and $F_2(\boldsymbol{x})$.

The full feature set of the SMV then is given by the local orientation estimate in Equation (3.4) and:

$$A_i(\boldsymbol{x}) = |F_i(\boldsymbol{x})|, \quad \phi_i(\boldsymbol{x}) = \arg F_i(\boldsymbol{x}), \tag{3.5}$$

for $i = 1, 2$. At each location $\boldsymbol{x}$, we choose the main signal by selecting the pair with the largest local amplitude. This selection is given by the dominant index $d(\boldsymbol{x}) = \arg\max_{1,2}\{A_1(\boldsymbol{x}), A_2(\boldsymbol{x})\}$, so that we have a major and minor IAP representation given by:

$$A(\boldsymbol{x}) = A_{d(\boldsymbol{x})}(\boldsymbol{x}), \quad \Phi(\boldsymbol{x}) = \phi_{d(\boldsymbol{x})}(\boldsymbol{x}), \quad a(\boldsymbol{x}) = A_{3-d(\boldsymbol{x})}(\boldsymbol{x}), \quad \phi(\boldsymbol{x}) = \phi_{3-d(\boldsymbol{x})}(\boldsymbol{x}).$$

Here the capital A and Φ denote the *dominant*, or *major*, local i1D signal. For the rest of the chapter these will be referred to as the *major amplitude* and *major phase* of the SMV. Figure 3.2 depicts the major IAP representation of two scales of Barbara.

3.2. Multiscale Feature Estimation

Let f be the given signal and $(f^{(s)})_{s=1}^{L\cdot K}$ be the isotropic wavelet decomposition of f, and $A^{(s)}, \Phi^{(s)}, a^{(s)}$, and $\phi^{(s)}$ for $s = 1, \dots, K$ denote the major amplitude, major phase, minor amplitude, and minor phase of $f^{(s)}$ respectively. This section explores some applications of these multiscale features. First, these are used to shed some light on some famous optical illusions, which motivates the multiscale phase as an important feature in optics and in local image analysis. We then describe a novel multiscale phase estimation approach which is useful for single-shot spatial phase estimation and for feature extraction more generally.

3.2.1. Optical Illusions and Local Image Structure. This section shows some interesting optical illusions, and how the monogenic phase, or the major phase of the SMV, allow some insight into human optics. To begin, consider the checkerboard grid image shown in Figure 3.3, originally presented by Adelson [**2**]. In this image square A is perceive to be darker than square B, even though the squares have the same gray scale intensity values. To confirm this, we also show cropped images around the squares in Figure 3.5(a)(b).

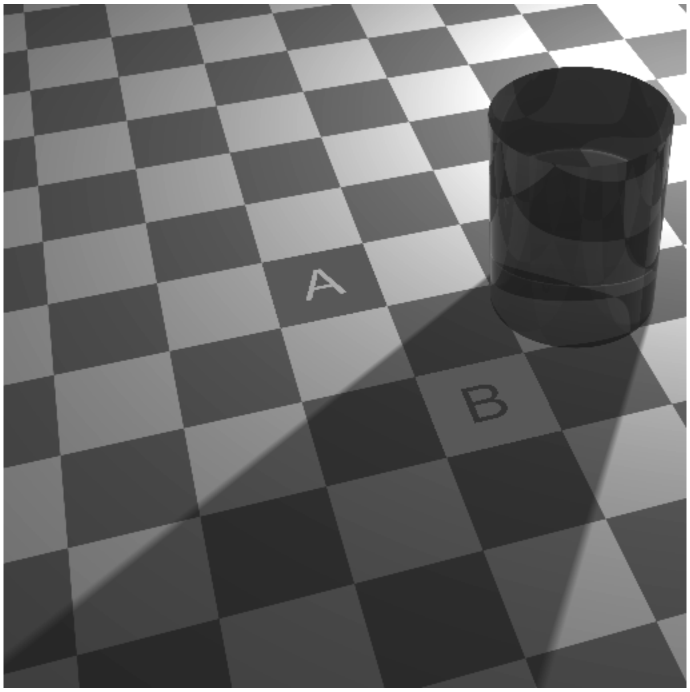


FIGURE 3.3. The checkerboard image, where the background of square A and square B are the same gray scale level, even though A is perceived to be darker than B.

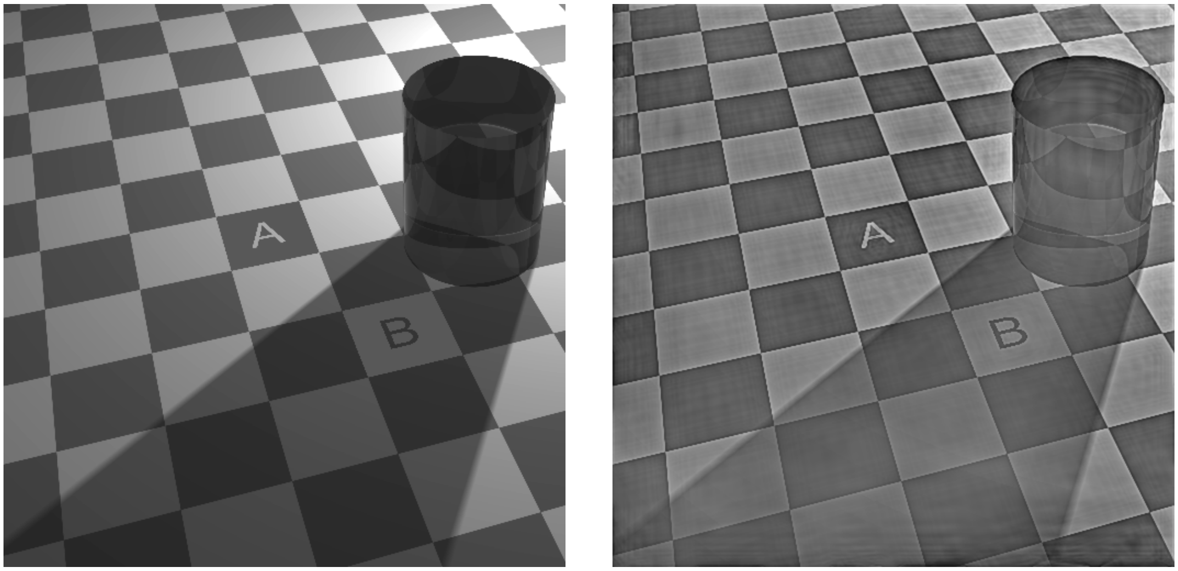
In Figure 3.5(a)(b) we show these same cropped images after applying *equalization of brightness* (EB) to the original image. Equalization of brightness is done as follows: given a multiscale decomposition of an image f into scales $\{f^{(s)}\}_{s=1}^{L\cdot K}$, at each scale s, define $\tilde{f}^{(s)}(\boldsymbol{x})$ as

$$\tilde{f}^{(s)} = \begin{cases} \cos\big(\phi^{(s)}(\boldsymbol{x})\big) & A^{(s)}(\boldsymbol{x}) \geq t, \\ \frac{A^{(s)}(\boldsymbol{x})}{t}\cos\big(\phi^{(s)}(\boldsymbol{x})\big) & A^{(s)}(\boldsymbol{x}) < t, \end{cases} \tag{3.6}$$

where $t = \mu_k + 2\sigma_k$ is a parameter based on the distribution of $A^{(s)}$. The equalized image is then given by summing up these energy-equalized scales:

$$f_{eb}(\boldsymbol{x}) = \sum_{s=1}^{L\cdot K} \tilde{f}^{(s)}(\boldsymbol{x}).$$

The threshold t is used here to ensure that areas of the image with very little energy are not enhanced; that is, the threshold t is where whatever structure present is considered insignificant. Figure 3.4 shows a side-by-side comparison of the original checkerboard image to the brightness-equalized image.



(a) Original checkerboard image, where A and B have the same grayscale value, though it appears B is brighter than A.

(b) After equalization of brightness B is actually brighter than A, suggesting the underlying structure informs our perception.

FIGURE 3.4. Comparison of the Checkerboard image before and after equalization of brightness.

Another interesting illusion to consider is the Hermann grid illusion show in Figure 3.6, originally studied by Hermann in 1870 [**41**]. There are gray dots that appear at the intersections of the grid lines, but only when one does not look directly at it. Those dots are not there if we consider only the intensity values of the image. Indeed, Figure 3.8(a) shows one of these intersection close up, which seems to close the case. However, after applying the equalization of brightness algorithm to this image, we actually see these gray dots (Figures 3.7, 3.8(b)). The main idea is that *we perceive spatial phase* when we look at images; the local structure of the image informs our understanding

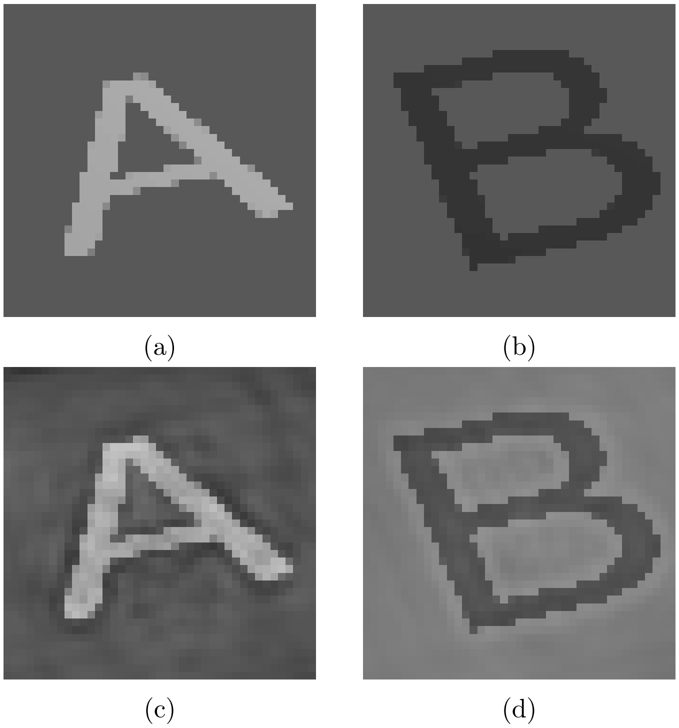


FIGURE 3.5. (a) *A* from original image (b) *B* from original image, (c) *A* after EB, (d) *B* after EB.

of the data, even to the point that it causes us to see things that are not there. It is valuable, then, to extract this local structural information as it provides important insight into the data.

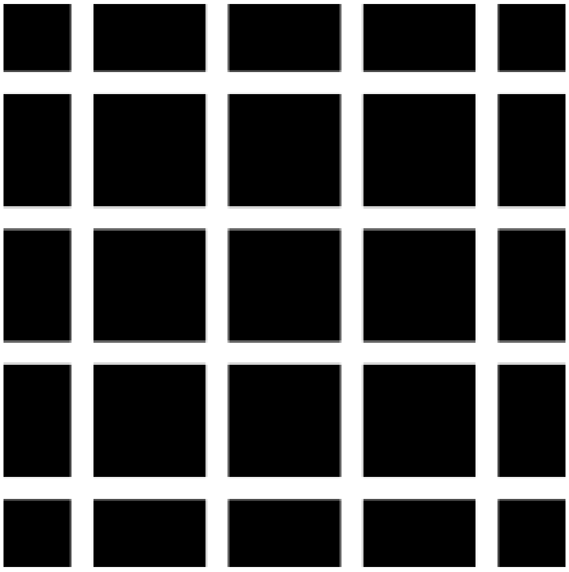
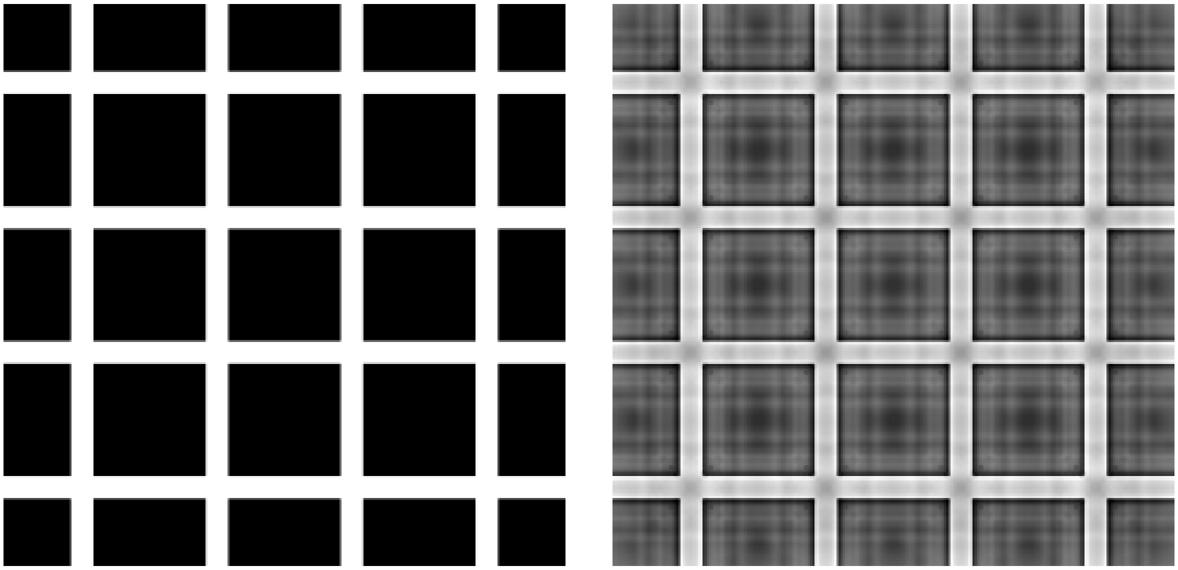


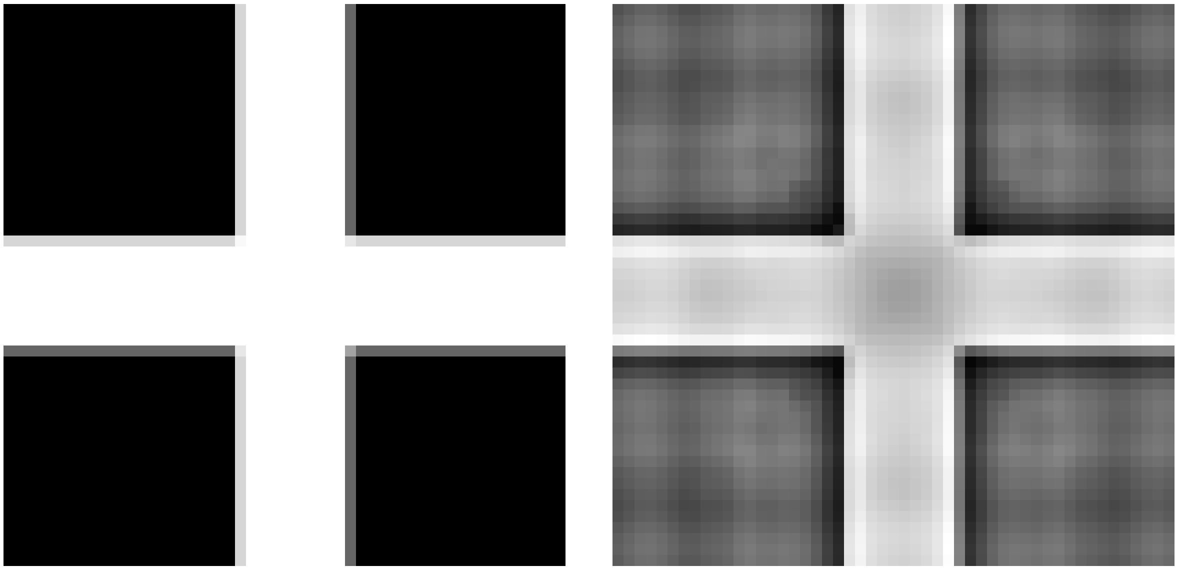
FIGURE 3.6. The Hermann Grid Illusion; elusive gray dots seem to appear at each intersection when you do not look at it directly.



(a) Original Hermann grid image with "ghost-like" gray blobs.

(b) After equalization of brightness, the elusive gray blobs are revealed as structural information.

FIGURE 3.7. Comparison of the Hermann grid image before and after EB.



(a) Zoomed view of an intersection, showing no gray blob.

(b) Zoomed view of the Hermann grid after equalization of brightness.

FIGURE 3.8. Close up comparison of the Hermann grid before and after EB.

3.2.2. A Multiscale Phase Estimate Using the Structure Multivector. Here we extend the multiscale phase extraction algorithm using the feature set of the structure multivector. If we let $\mathcal{Q}(\boldsymbol{x}) \geq 0$ be a *local quality function* which may depend on any aspect of the local multiscale feature set (here provided by steerable wavelets and the SMV), we can define the *local scale* to be given by

$$s^{\mathcal{Q}}(\boldsymbol{x}) = \arg\max_{s}\{\mathcal{Q}^{(s)}(\boldsymbol{x})\},$$

where $\mathcal{Q}^{(s)}$ is the quality function applied to the multiscale features at scale $f^{(s)}$. The corresponding local *multiscale features*

$$\begin{aligned} A_{\mathcal{Q}}(\boldsymbol{x}) &= A^{s^{\mathcal{Q}}(\boldsymbol{x})}(\boldsymbol{x}), & \Phi_{\mathcal{Q}}(\boldsymbol{x}) &= \Phi^{s^{\mathcal{Q}}(\boldsymbol{x})}(\boldsymbol{x}), & \theta_{\mathcal{Q}}(\boldsymbol{x}) &= \theta^{k^{\mathcal{Q}}(\boldsymbol{x})}(\boldsymbol{x}), \\ a_{\mathcal{Q}}(\boldsymbol{x}) &= a^{s^{\mathcal{Q}}(\boldsymbol{x})}(\boldsymbol{x}), & \phi_{\mathcal{Q}}(\boldsymbol{x}) &= \phi^{s^{\mathcal{Q}}(\boldsymbol{x})}(\boldsymbol{x}). \end{aligned}$$

In **[50]** $\mathcal{Q}^{(s)}(\boldsymbol{x})$ is defined to be the amplitude of the monogenic signal at that scale. The analog in this chapter is the major amplitude of the SMV at each scale, which we call the *local amplitude quality.* The idea is to choose the scale with the maximum local energy; if the underlying signal in question is some sort of well-structured fringe pattern, such as an interferogram or a fingerprint, this scale should correspond to the true spatial phase to be estimated and the multiscale feature set should be robust to noise or local signal corruption when the signal-to-noise-ratio (SNR) is sufficiently well behaved.

When the local SNR is close to 1 (0dB), however, the scale with the dominant amplitude is likely to be the noise itself. Still, it may be assumed that the signal corruption does not contain coherent structural information, and so we submit that a local quality metric which makes use of the local structural information should enable good phase estimation even when SNR ≤ 1.

We propose applying a local variance filter $\mathcal{V}_{w_k}$ to each θ^s with an appropriate window size, where lower local variance indicates a more coherent signal. Because the orientation here is modulo 2π, we use circular or directional statistics **[34, 63]**, otherwise differences between values of 0 and 2π are considered to be large variance which is not appropriate for this context. The local (circular) variance filter is defined about the point $\boldsymbol{x} = (x_1, x_2)$ as:

$$\mathcal{V}_{w_s}(\theta)(\boldsymbol{x}) = 1 - \frac{1}{4w_s^2}\left|\sum_{i=-w_s}^{w_s}\sum_{j=-w_s}^{w_s}\exp(\mathbb{i}\theta(\boldsymbol{x}[i,j]))\right|, \tag{3.7}$$

where $\boldsymbol{x}[i,j] = (x_1 + i\Delta x, x_2 + j\Delta x)$ for $-w_s \leq i, j \leq w_s$ and Δx the appropriate grid spacing for the image. The maximum of the the circular variance is 1, denoting the most possible spread of directions, and the minimum value is 0 indicating each point measures the same direction. The *local orientation variance quality* corresponding local scale is then defined to be:

$$\mathcal{Q}_\theta^{(s)}(\boldsymbol{x}) = \frac{1}{1 + \mathcal{V}_{w_k}(\theta^s)(\boldsymbol{x})}, \quad s^{\mathcal{Q}_\theta}(\boldsymbol{x}) = \arg\max_k \{\mathcal{Q}_\theta^{(s)}(\boldsymbol{x})\}.$$

Defined this way, a large local orientation variance leads to a lower local quality, and vice versa. We set the window size w_s to be twice the dyadic scale: $w_s = 2^{l+1}$. We also consider the *local product quality* given by the product of the local amplitude quality and local orientation variance quality, $\mathcal{Q}_\theta^{(s)}(\boldsymbol{x}) \cdot A^{(s)}(\boldsymbol{x})$, as this utilizes both local energy and local structure information. Figure 3.9 compares the results of these three quality functions for the spatial phase estimate of a real fingerprint; the top row shows a patch of the fingerprint along with four different scales of the same image.The next three rows show local phase estimates (left-most column) along with the local quality maps gives by $\mathcal{Q}^{(s)}$, $A^{(s)}$ and $\mathcal{Q}^{(s)} \cdot A^{(s)}$ in rows 2, 3 and 4 respectively. We further compare phase estimation results of a plane wave signal and a parabolic chirp signal, which are shown alongside their respective spatial phases in Figure 3.10. The details of our experiments are outlined further in Section 3.3.

3.3. Spatial Phase Estimation Experiments

This section provides results of the proposed multiscale phase estimation algorithm applied to several standard phase estimation problems: 1) a baseline experiment on estimating the phase of plane waves of different frequencies; 2) estimating the phase of a parabolic chirp, a standard signal with varying local frequency, which provides a more challenging multiscale phase estimation problem; 3) 2D phase demodulation; and 4) a practical example of phase demodulation as it applies to fine-scale fingerprint registration for the problem of fingerprint matching.

3.3.1. Multiscale Phase Estimation Experiments. Given a noisy plane wave of the form $f(\boldsymbol{x}) = \cos(\omega(\boldsymbol{n} \cdot \boldsymbol{x})) + \eta_\sigma(\boldsymbol{x})$, where $\eta_\sigma(\boldsymbol{x})$ is a Gaussian random variable with mean zero and standard deviation σ, the goal is to estimate the true phase function $\omega(\boldsymbol{n} \cdot \boldsymbol{x}) \mod 2\pi$. In our experiment we discretize so that $\boldsymbol{x}[i,j] = [-\pi + \frac{2\pi i}{N}, -\pi + \frac{2\pi j}{N}]$, for $i, j = 0, N-1$, for $N = 2^L$, ω ranges between 2^3 and 2^{L-2}, and $\boldsymbol{n} = [\cos(\pi/4), \sin(\pi/4)]^\intercal$. We use the *structural similarity index measure* (SSIM) [86] to compare the quality of the estimated phase to the ground truth, because it evaluates contrast and structure, and perceptual similarity which is relevant in this setting. PSRN could also be considered for this purpose. For a comparison of these two similarity measures see [44].

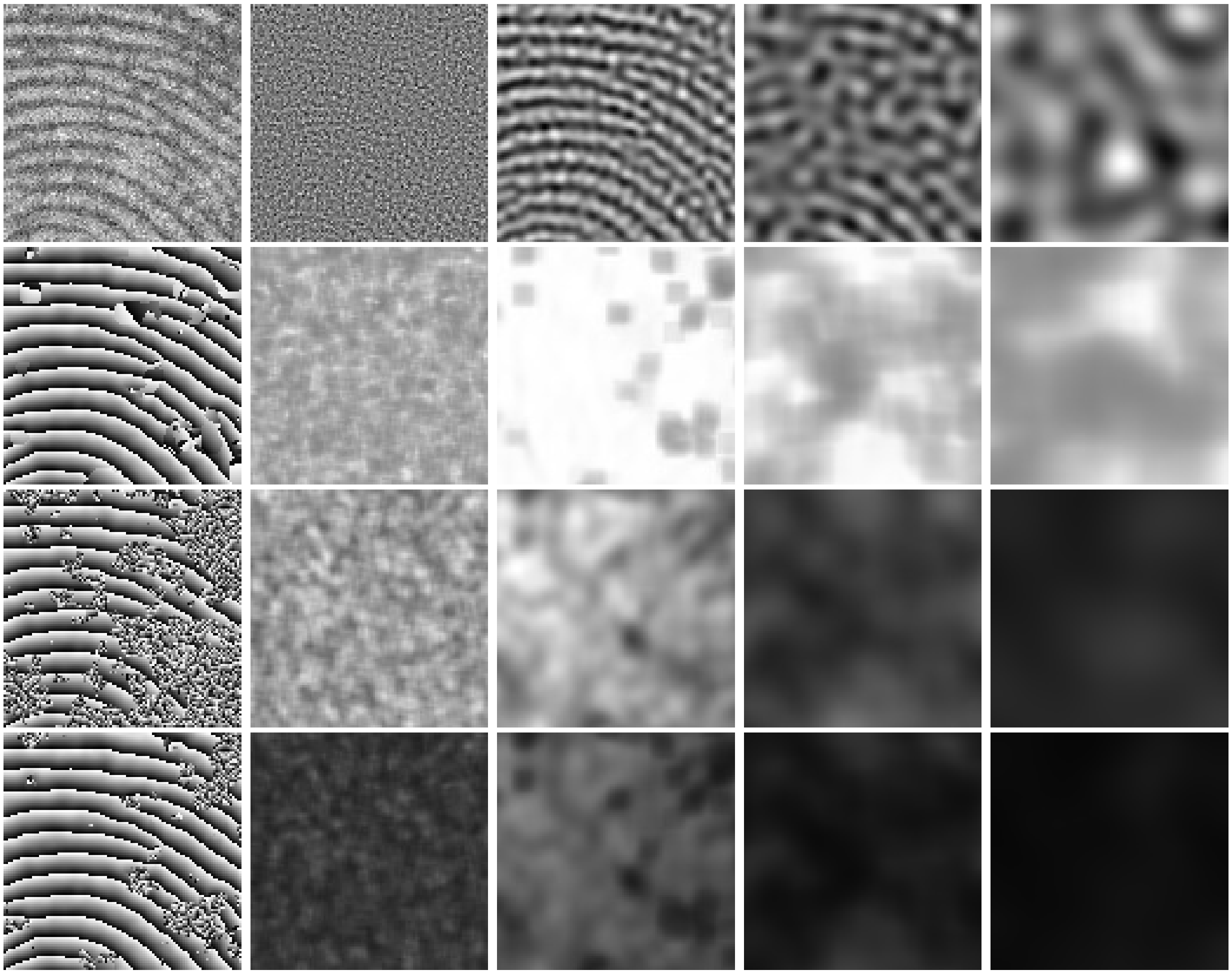
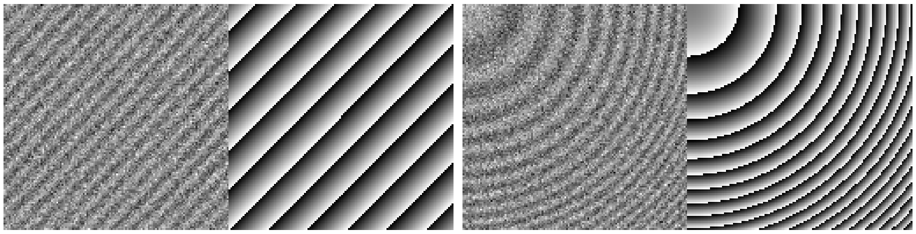


FIGURE 3.9. Top row: noisy fingerprint and decomposition into 4 scales. Second row: multiscale phase computed via $\mathcal{Q}_\theta^{(s)}(\boldsymbol{x})$, local orientation variance quality of each scale. Third row: the multiscale phase computed via $A^{(s)}(\boldsymbol{x})$, local amplitude of each scale. Fourth row: the multiscale phase of the product $\mathcal{Q}_\theta^{(s)}(\boldsymbol{x}) \cdot A^{(s)}(\boldsymbol{x})$, product quality at each scale.

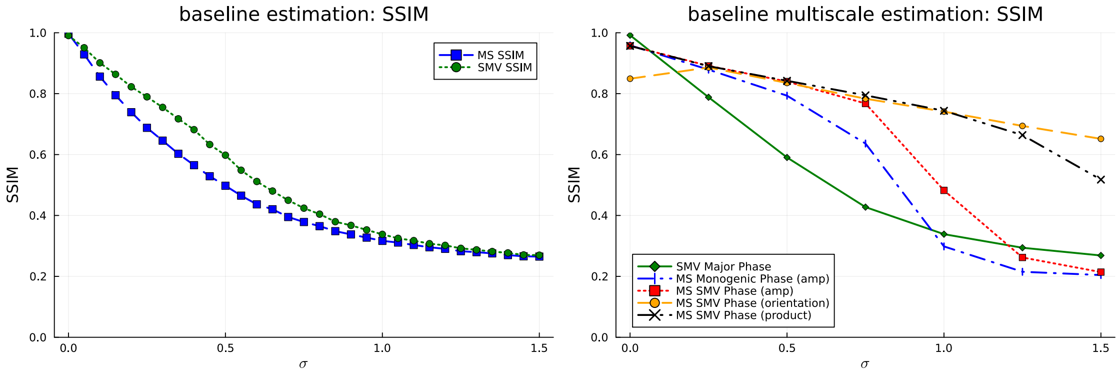
Figure 3.10 shows examples of noisy images alongside the true underlying phase, Figure 3.11 shows the quality of the estimated phase for $0 \leq \sigma \leq 1.5$. Because the true amplitude of the noiseless signal is 1 everywhere, σ can be thought of as the reciprocal of SNR, hence $\sigma = 1$ is the point at which the Gaussian noise begins to dominate the underlying plane wave structure. The amplitude-based multiscale phase of the monogenic signal and SMV perform well until σ surpasses 0.75, beyond which the estimate is unusable. In contrast, the estimate from the local orientation variance quality and the estimate given by the product of the amplitude and local orientation variance quality provide high-quality phase estimates well beyond this point. Additionally, we test our phase estimation procedure on a *parabolic chirp*. The appeal of this signal is that the spatial frequency varies locally,



(a) (b) (c) (d)

FIGURE 3.10. Example of a (a) noisy plane wave and (b) the true underlying phase, and (c) a parabolic chirp signal along with (d) its true phase.

and so it is a more challenging phase estimation task. Estimating the phase of such a signal is actually a known strength of the monogenic signal [**50**]. We demonstrate again that in the presence of signal corruption the monogenic signal alone fails quickly, but our multiscale approach finds the coherent local structure reliably.Furthermore, because both the monogenic signal and the SMV handle locally varying frequency well, we posit using an "overcomplete" set of features can improve phase estimation. We use a set of features which includes the low-pass component at each dyadic scale.

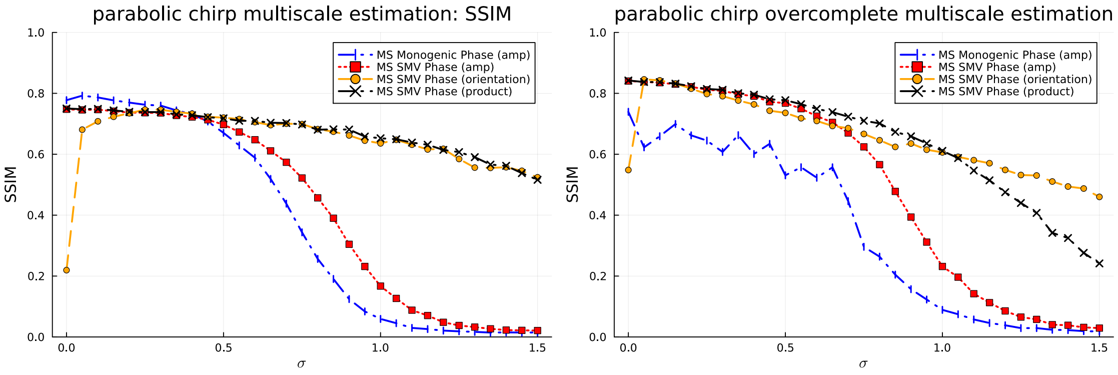


(a) (b)

FIGURE 3.11. phase estimation of noisy plane wave: (a) comparison of Monogenic phase vs. SMV phase (b) comparison of four multiscale phase estimates.

Figure 3.12 shows the results of the multiscale estimate and the overcomplete multiscale estimate for varying SNR.

3.3.2. Phase Demodulation with Multiscale Major Phase. The multiscale major phase estimate given by the the multiscale SMV representation of a signal f motivates further experiments in phase demodulation. In two dimensions, the phase demodulation problem can be stated as



(a)

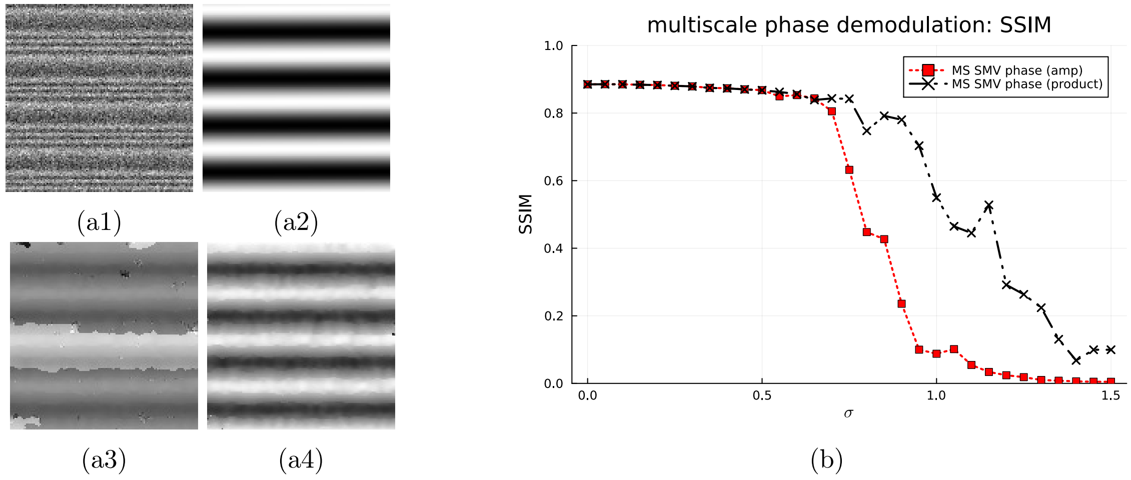
(b)

FIGURE 3.12. Phase estimation of noisy parabolic chirp: (a) comparison of four multiscale phase estimates (b) comparison of same multiscale estimation procedures with an overcomplete set of scales.

follows: Given the carrier wave $c(\boldsymbol{x}) = A\cos(\omega_c \boldsymbol{n}\cdot\boldsymbol{x} + \phi_c)$, and a message $m(\boldsymbol{x})$ where $\hat{m}(\boldsymbol{u}) = 0$ for $|\boldsymbol{u}| \geq \omega_c$, the phase modulated (PM) signal is given by:

$$c_{PM}(\boldsymbol{x}) = \cos(\omega_c \boldsymbol{n}\cdot\boldsymbol{x} + \phi_c + m(\boldsymbol{x})). \tag{3.8}$$

Typically the message is also sinusoidal in structure. For a baseline phase demodulation task, we attempt to recover a sinusoidal message after phase modulation and noise corruption. In this case, motivated by the phase estimation of the parabolic chirp, we use the overcomplete feature set when computing the multiscale phase to more accurately capture the overlapping frequency bands resulting from the phase modulation. After estimating the modulated phase, $\tilde{\Phi}$, it is unwrapped to produce $\tilde{\Phi}^u$. If successful, the message should be well estimated by either $\pm\tilde{\Phi}^u - \omega_c\boldsymbol{n}\cdot\boldsymbol{x}$. The ± here is due to sign ambiguity in the phase demodulation problem; for synthetic experiments the ground truth message allows us to choose the proper sign. Figure 3.13 shows the noisy phase modulated message, the ground truth message, and recovered messages via an amplitude based mulstiscale phase estimate versus the proposed product method. Again our tests conclude the product quality map allows for accurate phase estimation even when SNR is large.



(a1) (a2)

(a3) (a4)

(b)

FIGURE 3.13. Phase demodulation results: (a1) noisy phase modulated message, $\sigma = 0.75$, (a2) ground truth message, (a3) recovered via MS SMV amplitude quality, (a4) recovered via MS SMV product quality (b) comparison of MS SMV phase estimate using the amplitude quality (square markers) versus product quality (x markers).

3.4. Deformable Fingerprint Registration

In 2018 Cui et al [**26**] proposed a method for fine-scale fingerprint registration via phase demodulation. The method is as follows, given a fixed and moving image, f_f and f_m, which have already been coarsely registered, we consider f_f to be the carrier wave and f_m to be a phase-modulated signal, where the message represents the unknown displacement vector field. Let $T(\boldsymbol{x})$ represent this displacement vector field between $f_f(\boldsymbol{x})$ and $f_m(\boldsymbol{x})$ such that $f_f(\boldsymbol{x} + T(\boldsymbol{x})) = f_m(\boldsymbol{x})$. Let $\boldsymbol{x}' = \boldsymbol{x} + T(\boldsymbol{x})$. Compute ϕ_f and ϕ_m, the (wrapped) spatial phase of the fixed and moving image respectively, and let $\Delta\phi = \phi_f - \phi_m$. Then $\Delta\phi^u$, the unwrapped phase differences, combined with local frequency information can be used to compute a spatial displacement at each coordinate, which enable the fine-scale deformable registration step. More explicitly, we let $f_f(\boldsymbol{x}) = \cos\Big(\phi_f^u(\boldsymbol{x})\Big)$ and $f_m(\boldsymbol{x}) = \cos(\phi_m^u(\boldsymbol{x}))$ and assume that near $\boldsymbol{x} = \boldsymbol{x}_0$ we have the estimate $\phi(\boldsymbol{x}) = 2\pi\omega\boldsymbol{n}\cdot\boldsymbol{x}$. Then $f_m(\boldsymbol{x}) = f_f(\boldsymbol{x}') = \cos(2\pi\omega\boldsymbol{n}\cdot\boldsymbol{x} + 2\pi\omega\boldsymbol{n}\cdot T(\boldsymbol{x}))$ and $\Delta\phi^u(\boldsymbol{x}) = 2\pi\omega\boldsymbol{n}\cdot T(\boldsymbol{x})$, so we recover $\boldsymbol{n}\cdot T(\boldsymbol{x}) = \frac{\Delta\phi^u(\boldsymbol{x})}{2\pi\omega}$, which gives the magnitude of the projection of $T(\boldsymbol{x})$ along the local orientation $\boldsymbol{n} = \begin{bmatrix}\cos(\theta) & \sin(\theta)\end{bmatrix}^\intercal$. If we restrict our displacement along this direction, then, we have

$$d_x(\boldsymbol{x}) = \frac{\Delta\phi^u(\boldsymbol{x})}{2\pi\omega}\cos(\theta), \quad d_y(\boldsymbol{x}) = \frac{\Delta\phi^u(\boldsymbol{x})}{2\pi\omega}\sin(\theta). \tag{3.9}$$

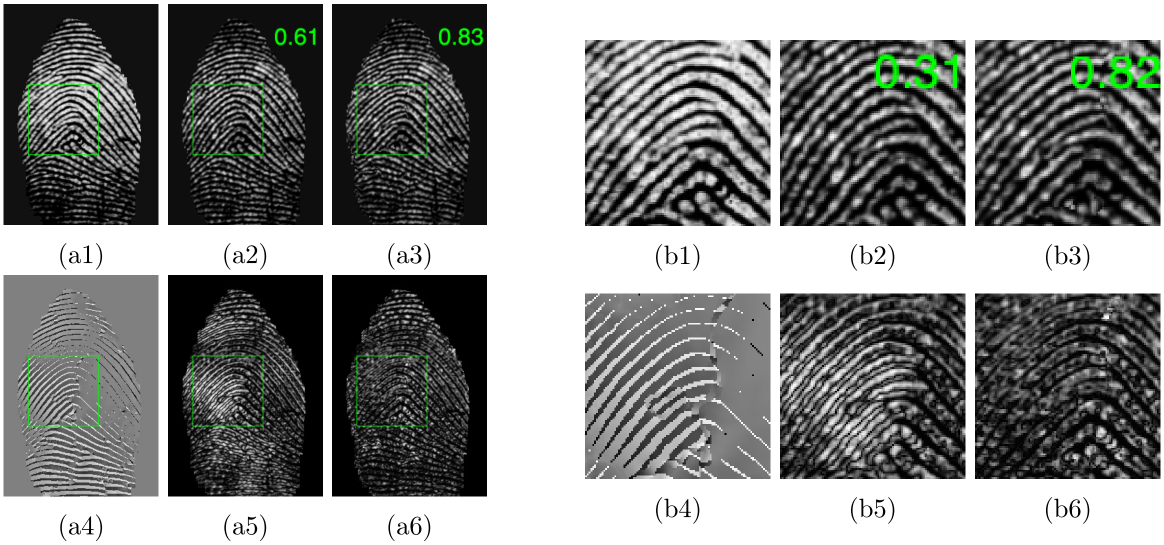


FIGURE 3.14. (a1) Fixed fingerprint image, (a2) moving image, (a3) registered image, (a4) difference of multiscale phases, (a5) $|f_m(\boldsymbol{x}) - f_f(\boldsymbol{x})|$ (a6) $|f_r(\boldsymbol{x}) - f_f(\boldsymbol{x})|$. (b1) fixed image, (b2) moving image, (b3) registered image, (b4) difference of multiscale phases, (b5) $|f_m(\boldsymbol{x}) - f_f(\boldsymbol{x})|$ (b6) $|f_r(\boldsymbol{x}) - f_f(\boldsymbol{x})|$. The overlaid numbers indicate the correlation coefficient with respect to the fixed image.

When we have estimates of the local phase, orientation, and frequency, this becomes

$$d_x(\boldsymbol{x}) = \frac{\Delta\phi^u(\boldsymbol{x})}{2\pi\omega(\boldsymbol{x})}\cos(\theta(\boldsymbol{x})), \quad d_y(\boldsymbol{x}) = \frac{\Delta\phi^u(\boldsymbol{x})}{2\pi\omega(\boldsymbol{x})}\sin(\theta(\boldsymbol{x})). \tag{3.10}$$

Let $T_e(\boldsymbol{x})$ is the estimated displacement vector field given by $T_e(\boldsymbol{x}) = \begin{bmatrix} d_x(\boldsymbol{x}) & d_y(\boldsymbol{x}) \end{bmatrix}^\intercal$, and f_r represent the registered fingerprint image after applying the estimate displacement field and reinterpolating onto the image grid. To measure the success of the registration we use a global measure given by the correlation coefficient:

$$\text{corrcoef}(f, g) = \frac{\sum_{i,j}(f(\boldsymbol{x}[i,j]) - \overline{f})(g(\boldsymbol{x}[i,j]) - \overline{g})}{\sqrt{\sum_{i,j} f(\boldsymbol{x}[i,j] - \overline{f})^2 \sum_{i,j} g(\boldsymbol{x}[i,j] - \overline{g})^2}}. \tag{3.11}$$

Global phase unwrapping algorithms typically have several major discontinuities which provide unreliable frequency information. In our experiments we compute a local frequency estimate $\omega(\boldsymbol{x})$ by differentiation of a windowed phase unwrapping of $\Delta\phi(\boldsymbol{x})$. The phase and orientation values used are those provided by the proposed multiscale method. Figure 3.14 gives an example of the fine-scale registration produced by this method (all images from FVC2004 DB1-B [**60**]).

TABLE 3.1. Correlation Values before and after fine-scale registration for various noise levels.

Type	Affine	Fine-scale	Difference
$\sigma = 0.0$	0.76	0.83	0.070
$\sigma = 0.1$	0.72	0.80	0.076
$\sigma = 0.2$	0.62	0.71	0.086
$\sigma = 0.3$	0.51	0.60	0.091
$\sigma = 0.4$	0.41	0.50	0.085
$\sigma = 0.5$	0.33	0.40	0.070

Table 3.1 expresses the quality of the fingerprint match after successful rigid registration of fingerprints, and then after the additional fine-scale deformable registration is applied. Because we are interested only in the performance of the fine-scale registration algorithm, we add noise only after successful rigid registration. We find that even at large noise levels we are able to estimate accurate fine-scale registration in well-structured areas of the fingerprints.

3.5. A low rank assumption for more robust denoising

Several recent works have shown promise in modeling fringe patterns as a low-rank structure, e.g. [**67**]. To extend the multiscale phase estimate introduced in this chapter, a multiscale low-rank decomposition is considered:

(1) for each scale k, compute the singular value decomposition of $f^{(s)}$ (either in the wavelet domain, or in the spatial domain), and keep r_k of the top singular values, where r_k is either chosen adaptively based on the shape of the decay of the coefficients, or such that $p\%$ of singular values are retained for some $0 < p < 1$. Let $\tilde{f}^{(s)}$ be the new low-rank (r_k rank) approximation of $f^{(s)}$;

(2) construct features $\tilde{A}^{(s)}, \tilde{\phi}^{(s)}$ from $\tilde{f}^{(s)}$;

(3) compute multiscale phase by scale selection according to the quality score Q.

Figure 3.15 and Figure 3.16 show results when applied to the baseline phase estimation task and the phase demodulation task. Though each scale $f^{(s)}$ is sufficiently bandlimited, the scales are formed from isotropic wavelet frames and so there is no guarantee $f^{(s)}$ will be well-approximated by a low-rank matrix. Because of this, some sort of local low-rank approximation might be considered in the future. Alternativelely, applying orientation-dependent wavelet filters would help to ensure better accuracy for a subsequent low-rank approximation.

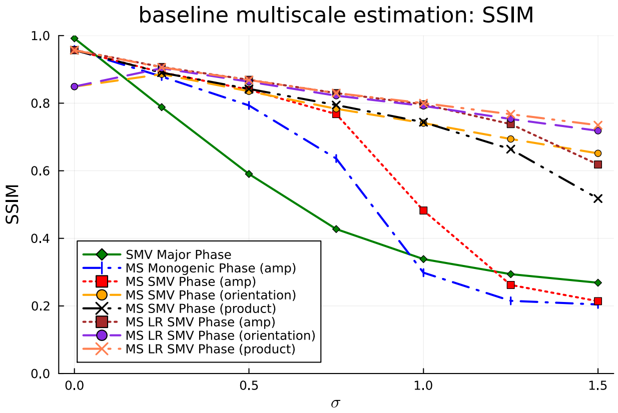
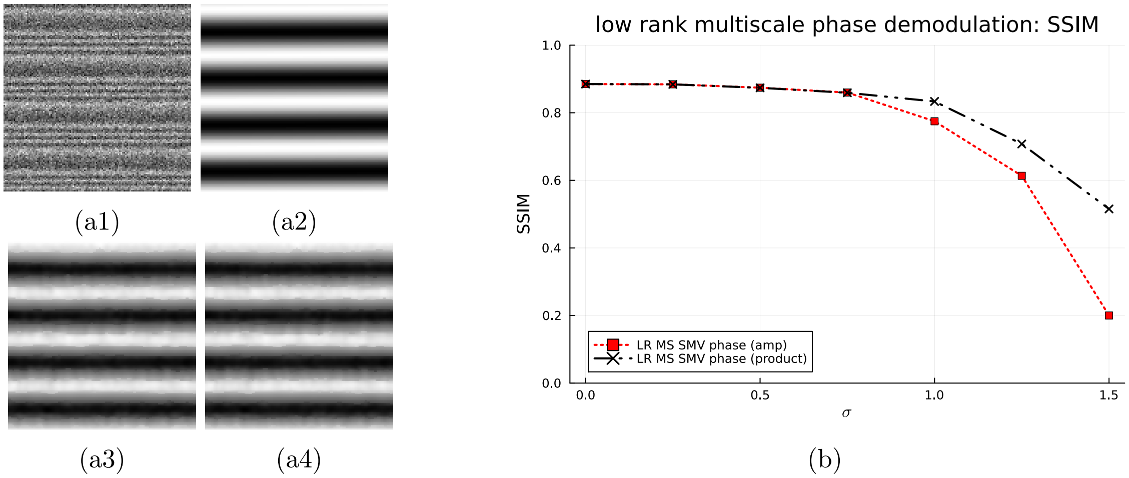


FIGURE 3.15. Baseline multiscale experiment with additional low rank estimates



(a1) (a2)

(a3) (a4)

(b)

FIGURE 3.16. Low rank phase demodulation results: (a1) noisy phase modulated message, $\sigma = 0.75$, (a2) ground truth message, (a3) recovered via MS SMV amplitude quality, (a4) recovered via MS SMV product quality (b) comparison of MS SMV phase estimate using the amplitude quality (square markers) versus product quality (x markers).

3.6. Conclusion

This chapter argued that the Structure Multivector is a robust method for estimating the local energy and structure of fringe and interference patterns, and further defined a local quality metric which rewards areas of coherent local structure. The result is a robust spatial phase estimation algorithm which allows for accurate spatial phase estimation even as noise begins to dominate the signal. This was demonstrated through several synthetic examples and in the practical setting of fine-scale fingerprint registration. Finally, an additional modeling approach of low-rank approximation was used to denoise each band-pass filter in the multiscale approximation, which has been shown to be effective when dealing with fringe pattern structures. This low-rank multiscale spatial phase estimate outperformed the original multiscale phase estimate in the all experiments using synthetic signals, which perhaps is unsurprising, as the noiseless signals are often of low rank structure. For the fingerprint experiments using real data the results varied. Still, this is an interesting direction that deserves further research and experimentation.

CHAPTER 4

Explainable Texture Segmentation with Wavelet Scattering Networks

This chapter outlines an approach to explainable image classification and texture segmentation. This is accomplished by solving an minimization problem concerning inputs to a multiclass logistic classifier. First, we discuss image classification tasks, using the standard MNIST [**30**] classification problem as an example. Next, we discuss an binary image classification using the Breast MNIST dataset [**89**], which consists of breast ultrasound images classified as normal tissue (or benign tumor), and malignant tumors. Additionally, a texture classification problems is considered involving textures from the Brodatz texture dataset [**9**], which are real-life textures commonly used to benchmark texture classification and texture segmentation tasks. Texture classification and segmentation is a long studied computer vision problem [**7, 13, 58**]; this is in part because textures famously elude explicit descriptions, hence explaining how a classifier distinguishes particular textures is challenging. This is another reason why the first problem addressed here is the well known MNIST classification problem; it is relatively easy to explicitly describe the classes, which are digitized images of handwritten digits 0 - 9, and so it is easier to interpret the features extracted by the proposed optimization scheme. The classification task involving the ultrasound images and the texture segmentation task with the Brodatz textures are more challenging; still, the features derived provide insight into how the trained model can discriminate between images belonging to one class from those belonging to any other class.

The extent to which a classification task or texture segmentation task can be explained depends first on the accuracy of the trained model; to this end a classification model based on the scattering transform, combined with a multiclass logistic regression classifier is first described, and classification results are reported. For each problem the classification achieves high accuracy (above 94% for all but the Breast MNIST classification problem, which achieves 78% accuracy) as shown in Table 4.1. The texture segmentation model introduced builds off of previous work of Anibou et al. [**6**], which uses a local wavelet scattering transform along with a score fusion technique in order to segment images based on local texture, and Saito [**74**] which introduces a method of explainable feature extraction for 1D signals.

4.1. Image Classification with the Wavelet Scattering Transform

In image classification problems, there are usually two majors steps. The first is to compute feature vectors from the original image data, which can be as simple as flattening each image in order to vectorize the dataset, or as complicated as training a deep CNN to learn convolutional filters, which are eventually reshaped into feature vectors via fully connected layers. The second step is to train a classifier based on these feature vector. These feature vectors can then be classified with additional layers in a neural network, or fed into a separate classification model. For each of the three problems under consideration here, the wavelet scattering transform [**61**] is used to extract robust features from the original image data, and then a logistic regression classifier is used to map these feature vectors to a specific label. Parameters for the scattering transform are first fixed by the user – the number of scales (J) and the number of angles used (L) – and the scattering transforms are then computed for the training and test sets X and $\tilde{X}$, along with the training labels y and test labels $\tilde{y}$. Then, a logistic regression classifier C is trained using an ℓ_1 penalty. The accuracy of the classifier is then evaluated on test data to ensure it will provide accurate predictions. This step is important in this case, because the minimization problem that is to be solved afterward concerns finding an input to the classifier that maximizes the probability of being in a certain class. Therefore if the classifier itself is not accurate, there is little hope that the resulting optimization problem will provide meaningful results. In our case, for each classification task, we perform 5-fold cross validation and present the mean and standard deviation of the accuracy (see Table 4.1). The coefficients which define the classifier are also inspected; it is assumed that classifiers trained using the ℓ_1 penalty should have a sparser set of coefficients, which impacts the landscape of the proposed optimization scheme.

4.1.1. Wavelet Scattering Transforms. The wavelet scattering transform (WST), or just scattering transform (ST), is a feature extraction tool inspired both by the success of CNNs and wavelet analysis. It was introduced by Stéphen Mallat [**61**] in 2012, where it was proven that the features of the ST are Lipschitz continuous with respect to the action of C^2 diffeomorphisms of the input signal f. Additionally, it has a weak translation invariance property. These theoretical properties, which are described in detail in Section 4.1.2, make the ST a robust feature extraction method appropriate for image classification problems and other machine learning tasks. Similarly to the convolutional layers of a CNN, the scattering transform performs convolution with the predefined filters based on a mother wavelet and its rotated and scaled version, followed by a nonlinearity which is typically taken to be the modulus function. Hence the ST is a nonlinear operation, and as a result it is generally not invertible. The mother wavelet and the rotations and scalings to be used are fixed a priori, so that the constructed filters are fixed, in contrast to a standard CNN where the parameters of each layer are learned. A *scattering transform network* (STN) combines the output of the ST with a classifier or regression model in order to perform prediction tasks. The success of STNs is two-fold: first, because the convolutional filters are computed a priori, a STN consists of far fewer trainable parameters than a standard CNN with the same architecture. Second, the filters themselves are interpretable, and so the scattering coefficients provide insight into how the predictive task is performed.

The scattering transform in 2D is now described in full detail. Essentially, a mother wavelet is paired with a discrete set of rotations and scalings. First, each rotation and scaling pair creates a frame atom which is convolved with f. A nonlinear Lipschitz continuous operator (e.g., the modulus operator) is then applied, followed by a subsampling operator which finally gives the outputs of layer 1. This process is repeated at each layer. Explictly, at layer m, let $\mathcal{O}_m$ be a finite rotation group in $\mathbb{R}^2$. Let Λ_m be the index set at the m-th layer which is made up of the rotation $q \in \mathcal{O}_m$ and the scale parameter $j \in \mathbb{Z}$, and in some given range $j > -J$ for some $J \in \mathbb{Z}$. Let ψ be a mother wavelet. Define $\lambda_m = (q, j) \in \Lambda_m$ to be the multi-index of a generator, which is obtained by rotating and dilating ψ at layer m. Then let ψ_{λ_m} denote the generator, which is given by

$$\psi_{\lambda_m}(\boldsymbol{x}) = 2^{\frac{2j}{Q_m}} \psi(2^{\frac{j}{Q_m}} q^{-1}\boldsymbol{x}),\ \boldsymbol{x} \in \mathbb{R}^2, \tag{4.1}$$

where $Q_m \in \mathbb{R}^+$ is a quality factor that can be tuned for adaptive scale adjustment. For the low frequency portion of the signal which is not accounted for by the generators, the father wavelet φ_0 and further scaling is used:

$$\varphi(\boldsymbol{x}) = 2^{2J}\varphi_0(2^J\boldsymbol{x}).$$

At this point, it is good to note that there are frame bounds A_m and B_m so that for any $f \in L^2(\mathbb{R}^2)$ it follows:

$$A_m\|f\|_2^2 \leq \|f * \varphi\|_2^2 + \sum_{\lambda_m \in \Lambda} \|f * \psi_{\lambda_m}\|_2^2 \leq B_m\|f\|_2^2. \tag{4.2}$$

For a given translation $\boldsymbol{\alpha} \in \mathbb{R}^2$, a *frame atom* is given by

$$\psi_{\boldsymbol{\alpha},\lambda_m} = \psi_{\lambda_m}(\boldsymbol{\alpha} - \boldsymbol{x})^*.$$

Defined in this way we have $\langle f, \psi_{\boldsymbol{\alpha},\lambda_m}\rangle = f * \psi_{\lambda_m}(\boldsymbol{\alpha})$. Let M_m denote the *modulus* operator given by

$$M_m f(\boldsymbol{x}) \;:=\; |f(\boldsymbol{x})|.$$

More generally, the operator M_m can be any Lipschitz continuous contraction operator with Lipschitz bound l_m **[61]**. Then, define the operator $U_m : \Lambda_m \times L^2(\mathbb{R}^2) \to L^2(\mathbb{R}^2)$ from layer $m-1$ to layer m given by:

$$U_m[\lambda_m]f(\boldsymbol{x}) \;:=\; M_m(f * \psi_{\lambda_m})(r_m\boldsymbol{x}),$$

where $r_m \geq 1$ is the subsampling rate. This operator is well defined due to the Lipschitz bound of M_m and the frame bounds given in Equation (4.2). Explicitly, it follows:

$$\begin{aligned}
\|U_m[\lambda_m]f(\boldsymbol{x})\|_2^2 &= \int_{\mathbb{R}^2} |M_m(f * \psi_{\lambda_m})(r_m\boldsymbol{x})| \,\mathrm{d}\boldsymbol{x} \\
&= \frac{1}{r_m^2}\int_{\mathbb{R}^2} |M_m(f * \psi_{\lambda_m})(r_m\boldsymbol{x})| \,\mathrm{d}r_m\boldsymbol{x} \\
&= \frac{1}{r_m^2}\int_{\mathbb{R}^2} |M_m(f * \psi_{\lambda_m})(\boldsymbol{y})| \,\mathrm{d}\boldsymbol{y} \\
&\leq \frac{l_m^2}{r_m^2}\|f * \psi_{\lambda_m}\|_2^2 \\
&\leq \frac{B_m l_m^2}{r_m^2}\|f\|_2^2.
\end{aligned}$$

Given a *scattering path* of indices $\boldsymbol{\lambda} \in \Lambda_m \times \Lambda_{m-1} \cdots \times \Lambda_1$; we define

$$U[\boldsymbol{\lambda}]f(\boldsymbol{x}) \;:=\; U_m[\lambda_m]U_{m-1}[\lambda_{m-1}]\cdots U_1[\lambda_1]f(\boldsymbol{x}).$$

The operator U is well-defined as also, following from the fact that U_m is well-defined at each layer: $\|U[\boldsymbol{\lambda}]f(\boldsymbol{x})\|_2^2 \leq \left(\prod_{k=1}^{m} \frac{B_m l_m^2}{r_m^2}\right)\|f\|_2^2$. Note that we will write $\Lambda^m = \Lambda_m \times \Lambda_{m-1} \cdots \times \Lambda_1$ for brevity.

The feature coefficients of the STN for layer m are given by the operators S_m and Φ_m defined as follows:

$$\begin{aligned} S_m[\boldsymbol{\lambda}]f(\boldsymbol{x}) &:= (\varphi * U[\boldsymbol{\lambda}]f)\,(r'_m\boldsymbol{x}), \\ \Phi_m f(\boldsymbol{x}) &:= \{S_m[\boldsymbol{\lambda}]f(\boldsymbol{x})\}_{\boldsymbol{\lambda}\in\Lambda^m}\,, \end{aligned}$$

where $r'_m \geq 1$ is another subsampling rate.

For $m = 0$ this gives

$$S_m[\emptyset]f(\boldsymbol{x}) = S_0 f(\boldsymbol{x}) := (\varphi_0 * f)\,(r'_m\boldsymbol{x}).$$

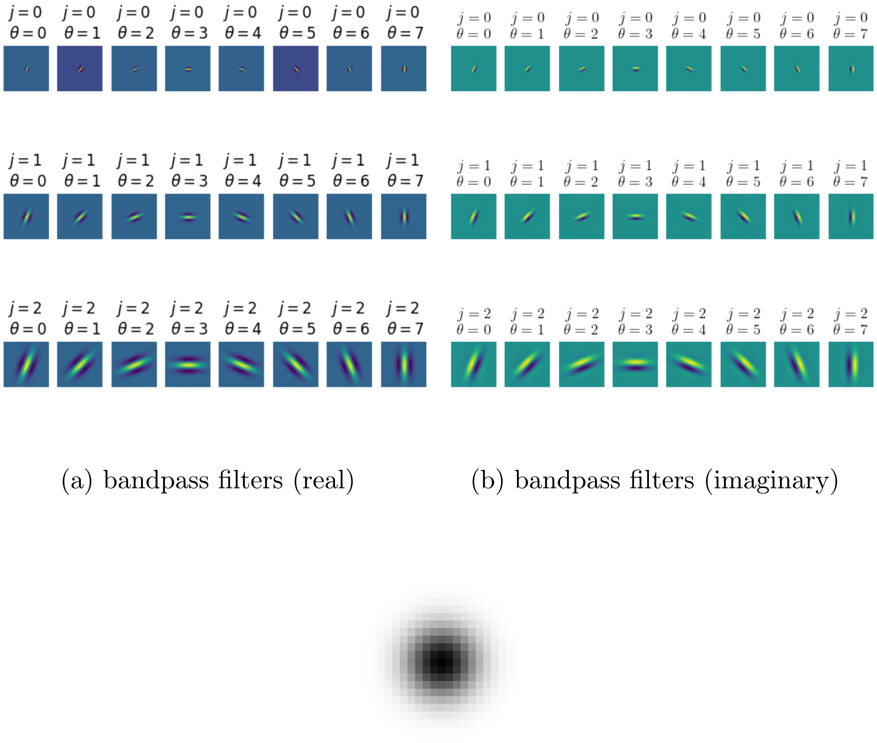
Finally, the features extracted from the entire scattering transform are given by

$$\Phi[f] := \bigcup_{m=0}^{\infty} \Phi_m[f].$$

These vectors are then concatenated, yielding a single feature vector which will also be denoted as $\Phi[f]$.

Example 4.1.1. *We consider a scattering transform with parameters $J = 3$, $L = 8$ for images of size $w \times w$ with $w = 32$, using the Morlet wavelet as implemented in the* `Kymatio` *Python package. In the* `Kymatio` *implementation we take $0 \leq j < J$ and $j \mapsto -j$ when comparing with Equation 4.1, so that lower values of j correspond to higher frequency atoms.*

4.1.2. Theoretical Properties of Wavelet Scattering Transforms. The core idea behind the development of the WST is group invariance; that is, a representation of a signal which is invariant to group actions such as translations and rotations, while also being stable under small perturbations (diffeomorphisms) which deform signals. It is clear that such representations are valuable for performing tasks such as signal or image classification, when the classes one is interested in are invariant with respect to these same group actions and should be unaffected by small local deformations. Stéphen Mallat formulated some of the mathematical properties of the wavelet scattering transform and later extended these ideas to deep CNNs [**10**,**61**]. Wiatowski and Bölcseki extended the result of Mallat when the signal under consideration is bandlimited [**88**]. Informally, the WST is a *deformation insensitive* feature extractor. This statement is made exact through the theory given in the remainder of this section.



(a) bandpass filters (real)

(b) bandpass filters (imaginary)

(c) low pass filter (scaling function)

FIGURE 4.1. Courtesy of Kymatio documentation. (a) Real part of the wavelets for each scale j and angle θ used. (b) Imaginary part (b) The corresponding low-pass filter, also known as the scaling function. Color saturation and color hue respectively denote complex magnitude and complex phase.

First, assume the *weak admissability condition*: that the upper bound B_m given in Equation (4.2) satisfies

$$B_m \leq \min\{1, l_m^{-2} r_m^{-2}\}. \tag{4.3}$$

Also, assume the contraction operator M_m commutes with the translation operator so that

$$M_m(f(\boldsymbol{x} - \boldsymbol{\alpha})) = M_m(f)(\boldsymbol{x} - \boldsymbol{\alpha}),$$

which certainly holds when M_m is the modulus operator. Let $L_a^2(\mathbb{R}^2)$ denote the set of bandlimited square integrable functions on $\mathbb{R}^2$, so that if $f \in L_a^2(\mathbb{R}^2)$ then $\hat{f}(\boldsymbol{\omega}) = 0$ for $\|\boldsymbol{\omega}\|_2 > a$. Let $D_{\boldsymbol{\tau},\omega}$ denote the space-frequency deformation operator, where $\boldsymbol{\tau}(\boldsymbol{x})$ is some nonlinear spatial distortion function, and $\omega(\boldsymbol{x})$ a nonlinear frequency modulation function, so that

$$D_{\boldsymbol{\tau},\omega} f(\boldsymbol{x}) := \mathrm{e}^{2\pi \mathrm{i}\omega(\boldsymbol{x})} f(\boldsymbol{x} - \boldsymbol{\tau}(\boldsymbol{x})) \tag{4.4}$$

The deformation error is then given by the quantity $\|D_{\boldsymbol{\tau},\omega} f - f\|_2^2$.

Proposition 4.1.1. *[88] Let $f \subset L_a^2(\mathbb{R}^2)$, and suppose $\boldsymbol{\tau} \in C^1(\mathbb{R}^2;\mathbb{R}^2)$ with $\|\Delta\boldsymbol{\tau}\|_\infty \le \frac{1}{4}$ and any $\omega \in C(\mathbb{R}^2;\mathbb{R})$. Then it follows that*

$$\|D_{\boldsymbol{\tau},\omega} f - f\|_2 \le C\left(a\|\boldsymbol{\tau}\|_\infty + \|\omega\|_\infty\right)\|f\|_2 \tag{4.5}$$

for some $C > 0$.

Proposition 4.1.2. *For any $f \in L^2(\mathbb{R}^2)$, the norm of the feature extractor Φ is given by:*

$$\|\Phi[f]\| := \sum_{m=0}^{\infty} \sum_{\boldsymbol{\lambda}\in\Lambda^m} \|S_m[\boldsymbol{\lambda}] f(\boldsymbol{x}) f\|_2.$$

By the weak admissability condition (4.3), it follows that Φ is Lipschitz continuous. This means that for any $f, g \in L^2(\mathbb{R}^2)$ it follows:

$$\|\Phi[f] - \Phi[g]\| \le \|f - g\|_2.$$

Corollary 4.1.1. *Let $\eta \in L^2(\mathbb{R})$ denote some additive noise. The feature extractor defined by the WST is robust with respect to additive noise in the sense that*

$$\|\Phi[f + \eta] - \Phi[f]\| \le \|\eta\|_2.$$

Based on these results, it is possible to state precisely how the WST is *deformation insensitive.*

Theorem 4.1.2 (**[88]**, Theorem 1). *Let $f \in L_a^2(\mathbb{R}^2)$ and let $\boldsymbol{\tau}(\boldsymbol{x})$ and $\omega(\boldsymbol{x})$ be as given in Proposition 4.1.1. Then there exists some $C > 0$ independent of Φ so that*

$$\|\Phi[D_{\boldsymbol{\tau},\omega} f] - \Phi[f]\| \le C\left(a\|\boldsymbol{\tau}\|_\infty + \|\omega\|_\infty\right)\|f\|_2. \tag{4.6}$$

Similar results hold for all $f \in L^2(\mathbb{R}^2)$ if $\omega = 0$, i.e., if only nonlinear spatial distortions are considered. Further, similar results are developed in the works cited here **[10,61,88]** for more general feature extractors such as deep CNNs. Lastly, it should be mentioned that Mallat proved that the

WST is *weakly translation invariant*, in the sense that Φ_m becomes more translation invariant as m increases. This is a very relevant fact for understanding group invariance of deep CNNs, and much discussion about this can again be found in the citations just mentioned.

4.1.3. Logistic Regression and Multinomial Logistic Regression Classifiers. Logistic regression is based on the *logistic function*, defined by the sigmoid $\sigma(t) = \frac{1}{1+e^{-t}}$, which takes in any real number and outputs some value between 0 and 1. The variable t is taken to be a linear function of the explanatory variables $x_1, x_2, \ldots, x_N$, i.e. $t = \theta_0 + \boldsymbol{\theta}^\intercal \boldsymbol{x}$. Logistic regression is based on the logistic model $p(\boldsymbol{x}) = \sigma(\theta_0 + \boldsymbol{\theta}^\intercal \boldsymbol{x})$. Given some data points $\{\boldsymbol{x}_k\}_{k=1}^K$ and their corresponding binary labels $\{y_k\}_{k=1}^K$, denote the probability of belonging to label 1 of each data point $\boldsymbol{x}_k$ by $p_k = p(\boldsymbol{x}_k)$. For convenience, let $\boldsymbol{\theta}' = \begin{bmatrix}\theta_0, \ldots, \theta_N\end{bmatrix}^\intercal$ be the full vector of learned model weights. To find an accurate model we then attempt to solve the minimization problem

$$\min_{\theta_0, \boldsymbol{\theta}} \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}(p_k), \tag{4.7}$$

where $\mathcal{L}$ is the *log loss* defined by $\mathcal{L}(p_k) = \begin{cases} -\ln p_k & y_k = 1, \\ -\ln(1-p_k) & y_k = 0. \end{cases}$ Essentially, for labels $y_k = 1$, we penalize the distance of p_k from 1, with a large penalty for very small values of p_k, and lesser penalty for values of p_k close to 1. The same is done for labels $y_k = 0$ by instead considering $1 - p_k$ to be penalized. For a comprehensive reference on the topic, see [**45**].

In addition, it is common to include regularization terms based on the 2-norm and 1-norm of the full coefficient vector $\boldsymbol{\theta}'$. This is most commonly done in the form of Elastic-Net regularization [**95**], which, in this case, yields the minimization problem

$$\min_{\boldsymbol{\theta}'} \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}(p_k) + \lambda \left[\frac{1-\alpha}{2} \|\boldsymbol{\theta}'\|_2^2 + \alpha \|\boldsymbol{\theta}'\|_1 \right].$$

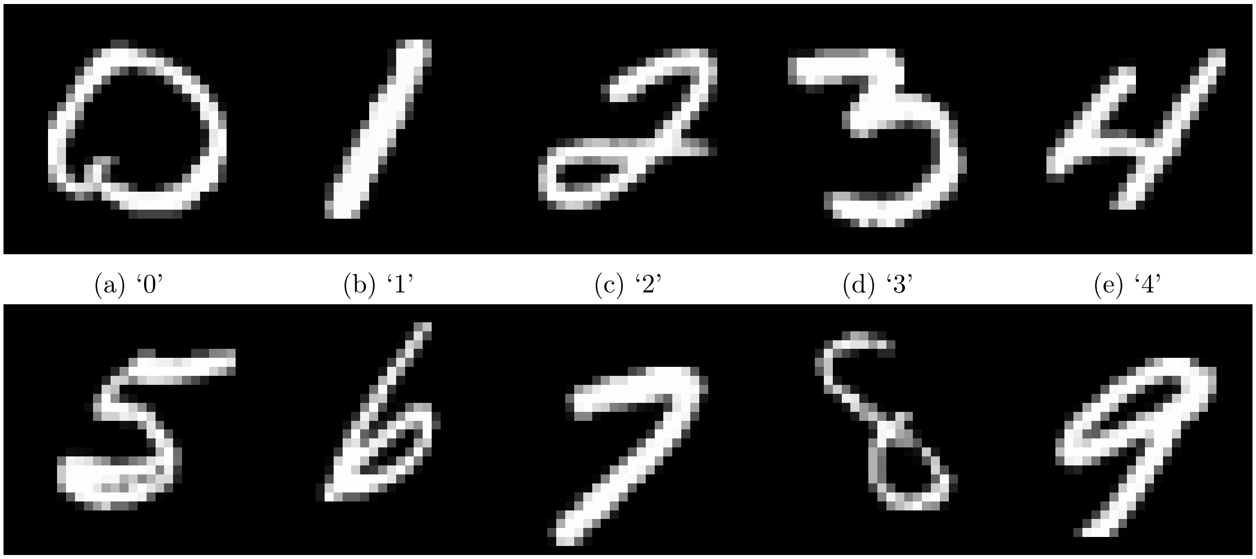
Here α denotes the elastic-net penalty, and $\alpha = 1$ corresponds to Lasso regularization [**82**] whereas $\alpha = 0$ corresponds to Ridge regularization [**83**]. The Lasso penalty encourages sparsity in the learned coefficients, while the Ridge penalty encourages smaller magnitude coefficients overall.

For multiclass problems with classes $1, \ldots, K$, the probability of an input belonging to class $k \in \{1, \ldots, K\}$ is given by the function $p_k(\boldsymbol{x}) := \frac{\exp(\theta_{0,k} + \boldsymbol{\theta}_k^\intercal \boldsymbol{x})}{\sum_{j=1}^K \exp(\theta_{0,j} + \boldsymbol{\theta}_j^\intercal \boldsymbol{x})}$, where $\theta_{0,k}$ is the kth intercept, and $\boldsymbol{\theta}_k$ the regression coefficient vector for class k.

Often, a feature vector $\mathbf{F}[\boldsymbol{x}] \in \mathbb{R}^{N'}$ is computed for each each data point $\boldsymbol{x} \in \mathbb{R}^N$. The regression model is then trained to classify these feature vectors, and so the more general expression for $p_k(\boldsymbol{x})$ is given by $p_k(\boldsymbol{x}) = \frac{\exp\big(\theta_{0,k}+\boldsymbol{\theta}_k^\intercal \mathbf{F}[\boldsymbol{x}]\big)}{\sum_{j=1}^{K} \exp\big(\theta_{0,j}+\boldsymbol{\theta}_j^\intercal \mathbf{F}[\boldsymbol{x}]\big)}$, where $\boldsymbol{\theta}_j \in \mathbb{R}^{N'}$ for $j = 1, \ldots, K$.

4.2. Datasets and Classification Results

4.2.1. MNIST. The MNIST dataset [30] consists of 70,000 images of handwritten digits between 0 and 9, each of size 28 × 28 pixels. For our purposes, these images are zero-padded with a 5-pixel width on each edge, so that each image is size 38 × 38. This is done in order to mitigate edge effects that occur as a consequence of the wavelet filtering done in the frequency domain. The image classification problem here is to train a model to classify each of the 10 digits. There are many approaches to this problem, including deep learning, and it is largely considered a solved task. This is good for the purpose of this chapter, which aims to explain accurate classification results. Figure 4.2 shows an example image from each of the 10 classes present, whereas Figure 4.3 shows the average image for each class.

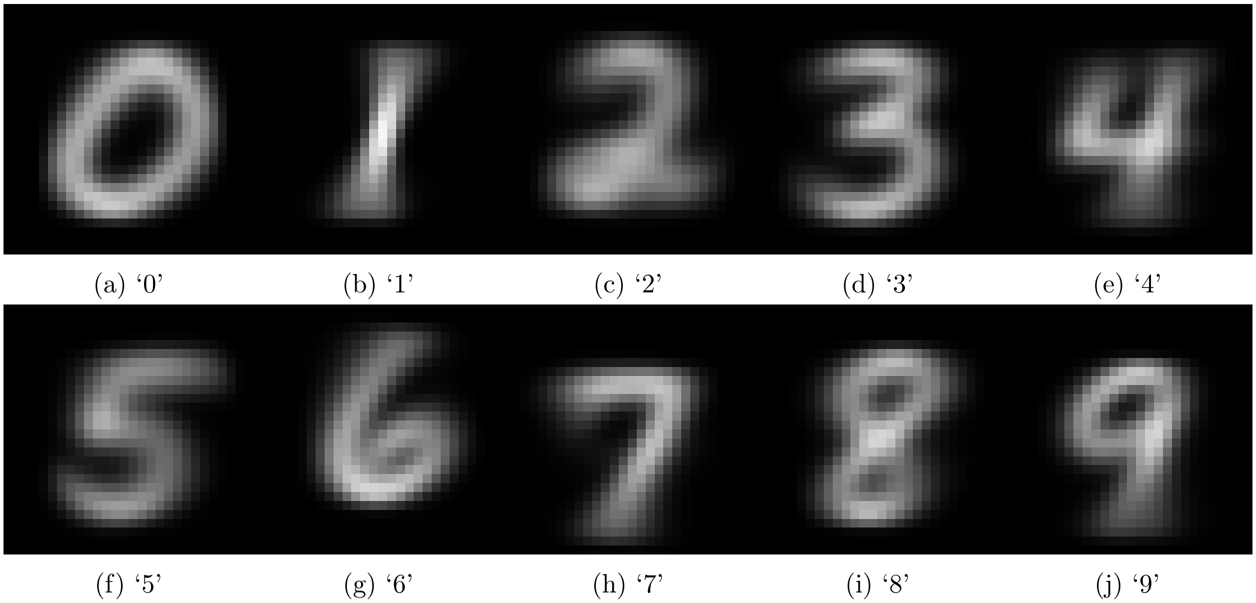


(a) '0' (b) '1' (c) '2' (d) '3' (e) '4'

(f) '5' (g) '6' (h) '7' (i) '8' (j) '9'

FIGURE 4.2. Example images for each digit label of the MNIST dataset.

We also consider a reduced-class classification problem using only five of the ten digits, 4, 5, 6, 7, and 9, and refer to this problem as MNIST (partial).



(a) ‘0’ (b) ‘1’ (c) ‘2’ (d) ‘3’ (e) ‘4’

(f) ‘5’ (g) ‘6’ (h) ‘7’ (i) ‘8’ (j) ‘9’

FIGURE 4.3. Average images for each digit label of the MNIST dataset.

4.2.2. Breast MNIST. The Breast MNIST dataset [89] is part of a larger collection called MedMNIST, or specifically MedMNIST 2D, which is a collection of benchmark data sets for medical image classification problems. The term MNIST is used in this case, because each data set is downsampled to be the same size as the standard MNIST dataset, and these designed to act as benchmark datasets to be used for medical image processing tasks. Breast MNIST consists of 780 breast ultrasound images downsampled to 28×28 patches. Each image is classified as either malignant (0), or normal (or benign) (1). The original dataset consists of higher resolution images with masks that contain the malignant or benign tumors, or no mask for a normal ultrasound. Figure 4.4 shows examples of ultrasounds with malignant tumors and normal tissue (or benign tumors), respectively. Here, because we'd like to capture the true feature of a malignant tumor, the training dataset (546 images) is augmented to include a 90 degree rotation of each image, and then this augmented dataset (1092 images) is then reflected horizontally and vertically. All in all this yields $6 \times 546 = 3276$ training images. It turns out to be the hardest classification problem we consider, with a classifier trained on these 3,276 images having an accuracy around 78% (Table 4.1).

4.2.3. Brodatz Textures. The Brodatz texture database [9] consists of 112 grayscale photographs by Phil Brodatz, labeled D1 to D112 which we accessed here: https://www.ux.uis.no/ tranden/brodatz.html. Each image depicts a unique texture; an example of 5 of these textures is shown

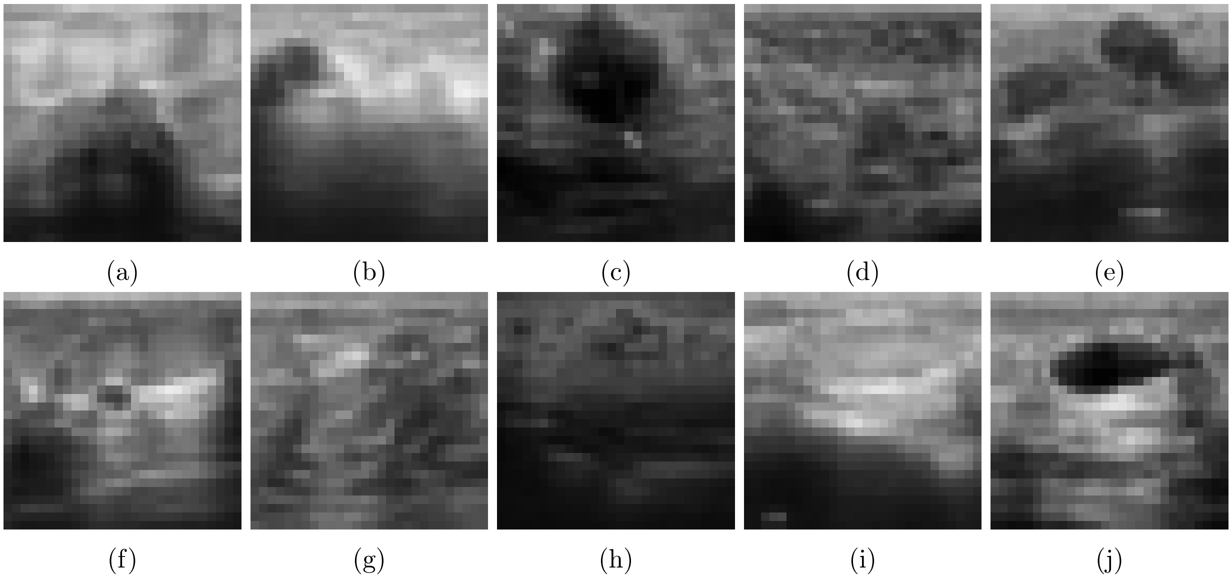


FIGURE 4.4. (a) - (e) Examples of ultrasounds classified as malignant. (f) - (j) Examples of ultrasounds classified as normal or benign.

in Figure 4.5; the 5 chosen here were chosen to represent a variety of texture styles, and we also ensured the spatial frequency exhibited could be captured at some level with the 28 × 28 pixel patches used to train the classifier. This database has become a standard texture database in image processing, and is often used as a texture classification benchmark (e.g., [**1**, **16**, **37**]). Here, the textures stitched together in various patterns, and a texture segmentation problem is considered, as show in Figure 4.6. Patches of size 24 × 24 are extracted from each type of texture image, and these patches are then zero-padded with a 4-pixel width on each edge, so that patches of size 32 × 32 are considered. Again, this is to limit any artificial boundary effects that might occur. For this data we do not apply any rotations or reflections to the patches as we want the optimized inputs to reflect relevant orientation information with respect to the given texture, and for all experiments the accuracy of these classifiers range from 94%-99% as shown in Table 4.1.

4.2.4. Additional Preprocessing and Data Augmentation. For the texture images that are considered, the effects of two additional preprocessing steps are studied: (1) each image patch is centered, and normalized to have a mean of zero and standard deviation of 1; (2) the brightness of each image patch is equalized using Algorithm 3.6. We do not use these additional steps for the MNIST dataset because these images are already normalized and for that reason any variance in intensity data is relevant to the classification problem. In the case of the texture images we consider,

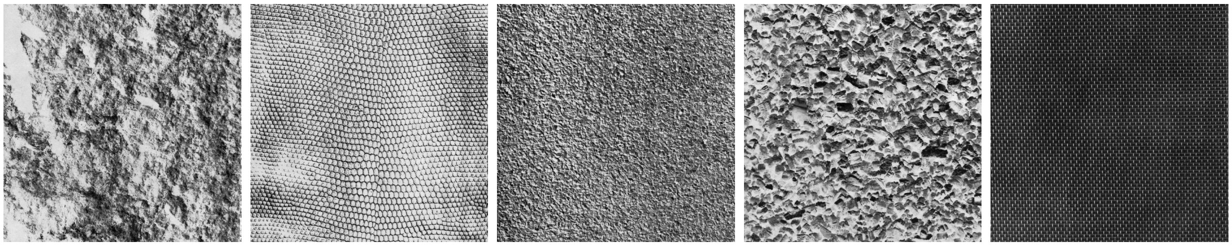

(a) texture 1 (b) texture 2 (c) texture 3 (d) texture 4 (e) texture 5

FIGURE 4.5. Five textures from the Brodatz texture database, from left to right: D2, D3, D4, D5, D6.

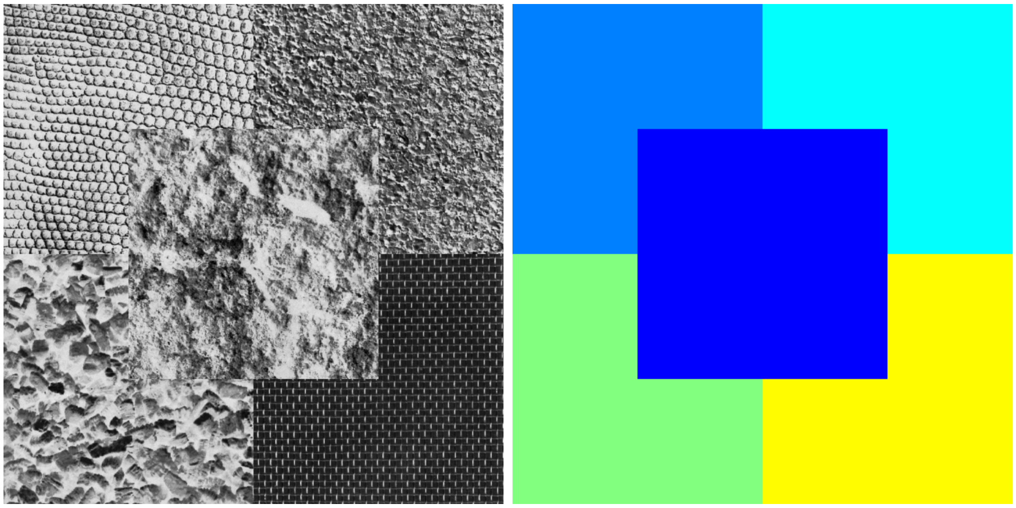

(a) example image (b) labels

FIGURE 4.6. An example of a test image and ground truth texture labels.

it is interesting to compare a classifier trained on patches with varying dynamic range to one trained on patches with fixed dynamic range (centered patches). Similarly, equalization of brightness is also considered here to enhance spatial phase data present in these textures and discard some intensity data, though using equalization of brightness does not guarantee that each patch will have the same dynamic range.

4.2.5. A Texture Segmentation Model Based on a Wavelet Scattering Network. In this section the texture segmentation algorithm is described and applied to some textures from the Brodatz texture database. Figure 4.5 shows some example textures, and Figure 4.6 shows an example of our image segmentation task.

The texture segmentation model is constructed as follows: fix a window size w, and some parameters to define the specific wavelet scattering transform to be used. Then, for each pixel, p_{ij}, compute the scattering transform coefficients of the patch centered about p_{ij}. The vector describing each pixel is derived from the scattering transform coefficients computed for a fixed window size around each pixel of the given textured images. That is, our training data will consist of the $n \times n$ patches from the five images shown in Figure 4.5, along with a unique label corresponding to whichever texture the patch is taken from – in this example then we have labels $\{1, 2, 3, 4, 5\}$. An example of a texture classification task is shown in Figure 4.6.

For now, we use the standard 2D ST given in the `Kymatio` Python package [5], with $J = 3, L = 8$ and a window size of 32×32 ($w = 32$). This is done to create periodic images that are less prone to boundary effects. Additional preprocessing could be done here to smooth the edges but was not implemented in this work. From each image $\approx$ 15k patches are extracted, and the ST coefficients of each patch are computed and stored. A multiclass logistic regression classifier is then trained based on these ST coefficients. The Lasso penalty is used during this training to promote a model which depends on a sparsity of ST coefficients.

Figure 4.7 show the texture segmentation given by the multiclass logistic regression model, and the result after [6] based on the class probabilities of neighboring pixels. Table 4.1 gives the classification results for the patch classification problem denoted by the Brodatz 1 dataset in row 2.

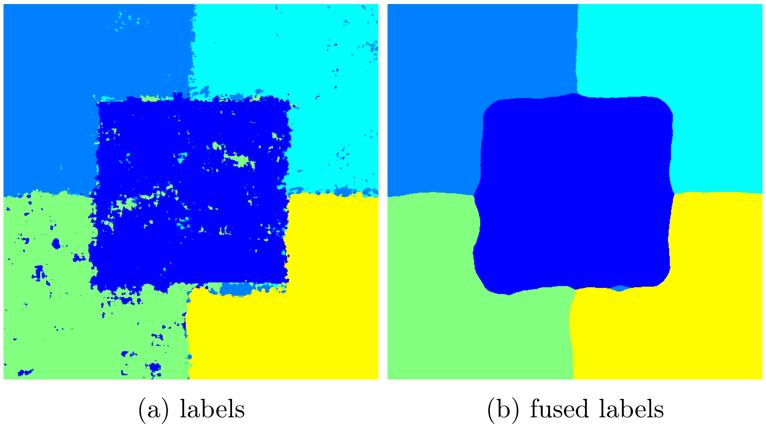

(a) labels (b) fused labels

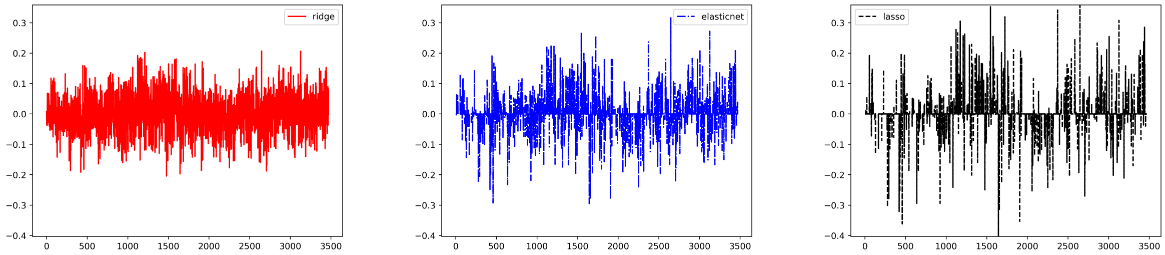
FIGURE 4.7. (a subset of) predicted texture labels and fused labels from our classifier.

4.2.6. Classification Results. Table 4.1 reports the accuracy of all the classifiers trained here. The accuracy of all classifiers is above 94% except for the Breast MNIST classifier, which contains the least amount of training data and is generally a more challenging task than the standard MNIST classification. Further, Figure 4.8 shows the difference in the coefficients of the final logistic regression classifier trained on the scattering transform coefficients. For all classifiers trained with the Ridge penalty, we set the penalty parameter $C = 1$ in `scikit-learn`'s `Logistic_Regression` function, and used the LBFGS solver [**12**]. For those trained with the Lasso penalty we set $C = 0.01$ and used the SAGA solver [**28**].

TABLE 4.1. Classification Results

Dataset	Training Size	Penalty	# classes	ST Parameters	Accuracy (%)
MNIST	10k	Ridge	10	$w = 38, J = 3, L = 8$	98.7 ± 0.3
–	–	Lasso	–	–	95.6 ± 0.6
MNIST (partial)	4905	Ridge	5	$w = 38, J = 3, L = 8$	99.3 ± 0.5
–	–	Lasso	–	–	97.8 ± 0.3
Breast MNIST	3276	Ridge	2	$w = 38, J = 3, L = 8$	78.8 ± 1.3
Brodatz 1	5k	Lasso	5	$w = 32, J = 3, L = 8$	95.7 ± 0.5
Brodatz 1 centered	–	–	5	–	95.1 ± 0.0
Brodatz 1 EB	–		–	–	94.6 ± 0.5
Brodatz 2	5k	Lasso	4	$w = 32, J = 3, L = 8$	99.9 ± 0.1
Brodatz 2 centered	–	–	–	–	99.9 ± 0.1
Brodatz 2 EB	–	–	–	–	99.9 ± 0.1

[1] Accuracy is based on 5-fold cross validation
[2] Here'–' indicates the value is the same as the row above.



(a) Ridge

(b) Elastic-Net

(c) Lasso

FIGURE 4.8. Coefficients of logistic regression classifiers trained on the same data (Breast MNIST). (a) Ridge penalty; (b) Elasticnet penalty with $\alpha = 0.5$ (c) Lasso penalty.

4.3. Extracting Explainable Features via Zeroth-Order Optimization

Once a classifier is successfully trained, it is able to classify images rather accurately as Table 4.1 demonstrates. In order to understand or interpret how the classifier arrives at these predictions, i.e., how it distinguishes one class from another, it is common to inspect the feature vectors involved. In the classifiers trained here, these feature vectors are exactly the feature vectors output by the WST, and hence contain some explainability a priori, as opposed to convolutional filters in a CNN, for example. Another interesting consideration is *what sort of input maximizes the probability of being in class k*. However, the gradient of the WST vector, $\nabla\Phi[\boldsymbol{x}]$ is highly discontinuous, as described in Chapter 4 of the dissertation of David Weber [**87**]. In order to explore this question, then, zeroth-order optimization is used as suggested by Saito [**74**]. Several different regularizing terms are considered in order to shape the landscape of the proposed optimization problem.

4.3.1. Zeroth-Order Optimization. For each class, we solve the optimization problem given by:

$$\boldsymbol{x}_k^\star = \arg\min_{\boldsymbol{x}} \left\{ L(p_k(\boldsymbol{x})) + \boldsymbol{\mu}^\intercal \boldsymbol{r}(\boldsymbol{x}) \right\}, \tag{4.8}$$

where $\boldsymbol{\mu}^\intercal \boldsymbol{r}(x)$ gives the full regularization term. Because a zeroth-order optimization scheme is used here, there is no need for convexity or regularity in either the loss term nor the regularization term. In general, zeroth-order schemes come at the cost of convergence guarantees, but for this they give the user the freedom to include explicit regularization terms and even explicit constraints. For more on zeroth-order optimization we refer the reader to [**24, 57, 59**]. In our case, we utilize the PRIMA solver [**92**] implemented in Python. Finally, we use the loss function $L(p_k) = -\log(p_k)$ in all experiments discussed.

4.3.1.1. *Sparsity.* An important constraint considered in [**74**] is the ℓ_1 norm, which encouraged sparsity in the solution $\boldsymbol{x}_k^\star$. This is because the ℓ_1 norm acts as a convex relaxation of the ℓ_0 and, which counts the number of non-zero entries in a vector. In 2009 Hurley and Rickard [**46**] explored several measures of sparsity including the ℓ_0 and ℓ_1 norms. Here sparsity is thought of as a measurement of wealth disparity, where a perfect wealth distribution (all people have an equal share of wealth) minimizes sparsity, and one person containing all wealth would minimize sparsity. To measure this idea of sparsity, six properties are proposed that sparsity measurements *should* contain. It turns out that both the Gini index [**36**] and the pq-mean (when $p \leq 1$ and $q > 1$) satisfy all six of these constraints. Because the convexity of these measures is not relevant to our optimization scheme, we use the pq-mean as a regularizer. For a vector $\boldsymbol{x} \in \mathbb{R}^N$, the pq-mean is defined as:

$$\|\boldsymbol{x}\|_{pq-\text{mean}} := -\left(\frac{1}{N}\sum_{j=1}^{N} x_j^p\right)^{\frac{1}{p}}\left(\frac{1}{N}\sum_{j=1}^{N} x_j^q\right)^{-\frac{1}{q}}, \text{ where } 0 < p < q. \tag{4.9}$$

For this problem, we consider five such explicit constraints, namely, the the gradient of $\boldsymbol{x}$, the gradient of $\hat{\boldsymbol{x}}$, the pq-mean of $\boldsymbol{x}$, the pq-mean of $\hat{\boldsymbol{x}}$, and an edge energy constraint, which penalizes energy near the boundary of each $\boldsymbol{x}$. For each pq-mean we set $p = 1$ and $q = 3$. In general, including the gradient term as a regularizer will encourage smooth solutions. Here, we pair this gradient term with the sparsity term in the original image space and the Fourier space. In doing so, we aim to produce optimized inputs with interpretable Gabor-like features, localized both in time and spatial frequency [**35**, **75**]. These are written explicitly as:

$$\boldsymbol{\mu}^\intercal \boldsymbol{r} = \mu_1 \cdot \|\nabla \boldsymbol{x}\|_2 + \mu_2 \cdot \|\nabla \hat{\boldsymbol{x}}\|_2 + \mu_3 \cdot \|\boldsymbol{x}_{\text{flat}}\|_{pq} + \mu_4 \cdot \|\hat{\boldsymbol{x}}_{\text{flat}}\|_{pq} + \mu_5 \cdot \|M_{\text{pad}}(w) \otimes \boldsymbol{x}\|_2, \tag{4.10}$$

where $\boldsymbol{x}_{\text{flat}}$ and $\hat{\boldsymbol{x}}_{\text{flat}}$ denote the 1D vectors that result from flattening the 2D arrays $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ respectively, and $M_{\text{pad}}(w)$ denotes the mask which zeros out all but the edge pixels, i.e.,

$$(M_{\text{pad}}(w) \otimes \boldsymbol{x})[i, j] = \begin{cases} 0 & w < i, j < n - w, \\ \boldsymbol{x}[i, j] & \text{otherwise.} \end{cases}$$

By encouraging both smoothness and sparsity in the spatial and frequency domain, the preferred solution will tend toward smooth wave packets, which should highlight discriminating features representative of the given class. The values of these parameters vary slightly between experiments, and in some the smoothness (gradient) parameters are set to 0. The edge penalty is meant to mitigate unwanted edge effects from the wavelet filters applied in the frequency domain. In general, how to set these parameters remains an open question which we plan to address in future work.

4.3.2. Explainable Feature Extraction with MNIST. Figure 4.9 shows several results of this optimization scheme applied to a classifier trained on MNIST to predict classes 4, 5, 6, 7 and 9. For these results we set $\mu_i = 1 \times 10^{-5}$ for $i = 1, 2, 3, 4$ and $\mu_5 = 0.01$, and consider a classifier trained with both Ridge penalty and the Lasso penalty. The probability of each of the optimized images is given in the figure captions; in this case each probability was at or above 99% in the Ridge regression case, and at or above 96% in the Lasso regression case.

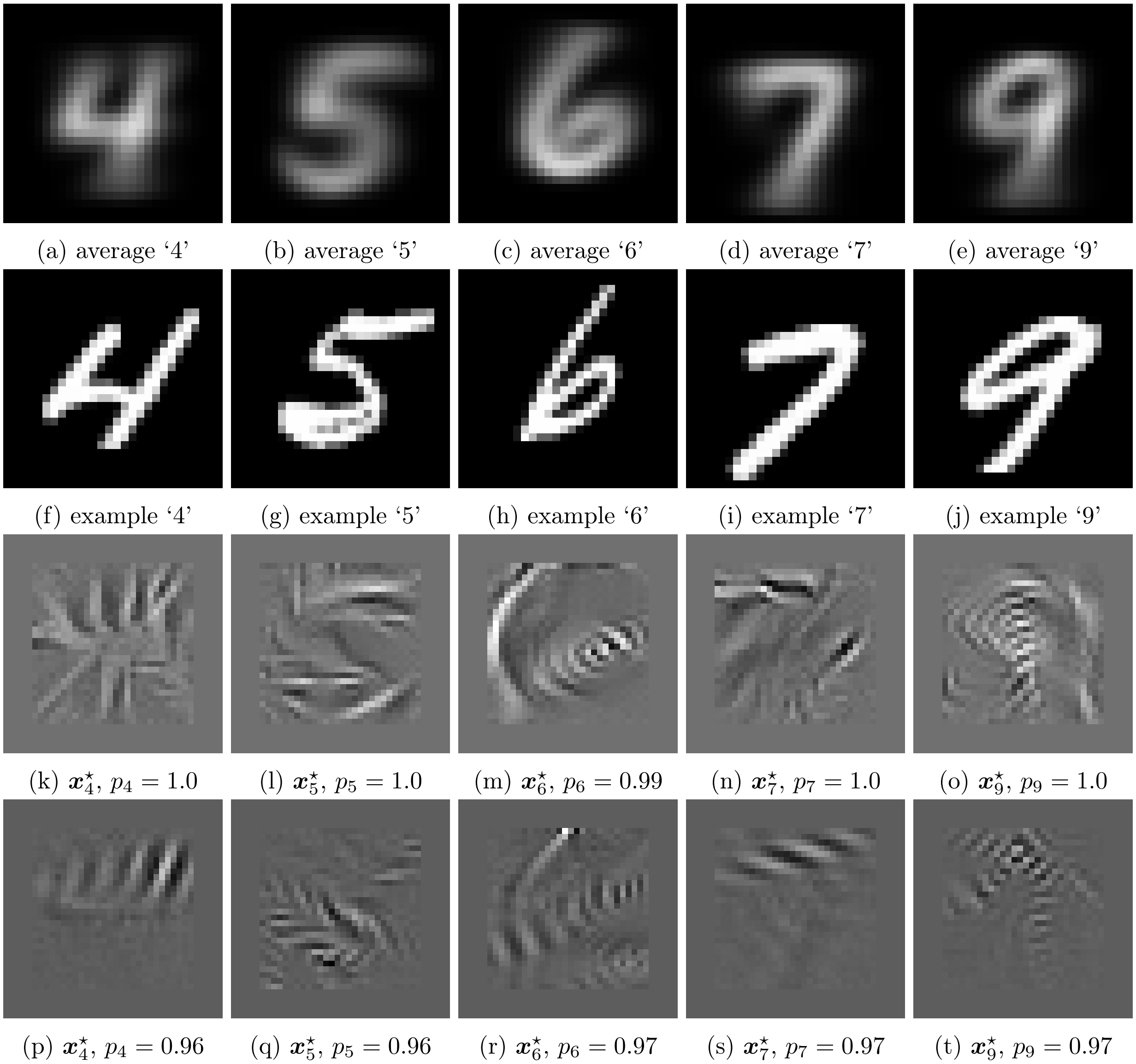
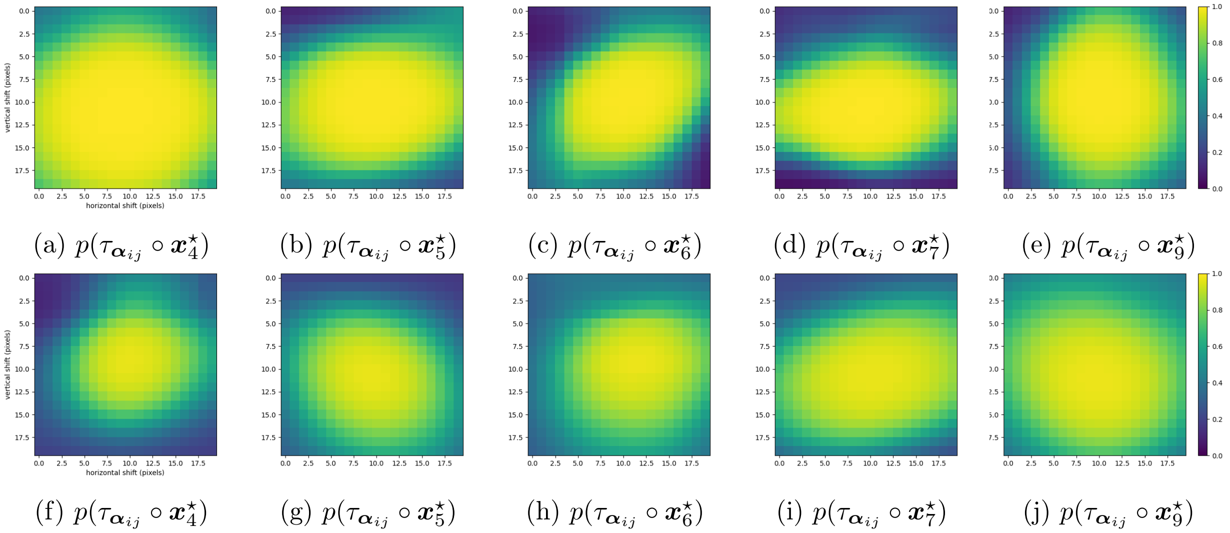

(a) average '4'
(b) average '5'
(c) average '6'
(d) average '7'
(e) average '9'
(f) example '4'
(g) example '5'
(h) example '6'
(i) example '7'
(j) example '9'
(k) $\boldsymbol{x}_4^\star$, $p_4 = 1.0$
(l) $\boldsymbol{x}_5^\star$, $p_5 = 1.0$
(m) $\boldsymbol{x}_6^\star$, $p_6 = 0.99$
(n) $\boldsymbol{x}_7^\star$, $p_7 = 1.0$
(o) $\boldsymbol{x}_9^\star$, $p_9 = 1.0$
(p) $\boldsymbol{x}_4^\star$, $p_4 = 0.96$
(q) $\boldsymbol{x}_5^\star$, $p_5 = 0.96$
(r) $\boldsymbol{x}_6^\star$, $p_6 = 0.97$
(s) $\boldsymbol{x}_7^\star$, $p_7 = 0.97$
(t) $\boldsymbol{x}_9^\star$, $p_9 = 0.97$

FIGURE 4.9. (a) - (e) Average image for each class considered; (f) - (j) Example image from each class considered; (k) - (o) Explainable features derived via Equation 4.8 from each of the five classes with regularization parameters $\mu_i = 1 \times 10^{-5}$ for $i = 1, 2, 3, 4$ and $\mu_5 = 0.01$, trained with Ridge penalty. (p)-(t) The same results but for a classifier trained with the Lasso penalty.

Because the handwritten '6' and '9' digits are related through a rotation by 180 degrees, these are interesting classes to consider. The optimal solutions $\boldsymbol{x}_6^\star$ and $\boldsymbol{x}_9^\star$ exhibit a similar symmetry, specifically around their respective loops. The handwritten '4' has two standard fonts, one where the top comes to a single point, and the other where the top comes to two disconnected points, and the optimized patch is concentrated here about the top region where these two fonts would differ. The handwritten '7' digit seems to be distinguished by the classifier based mainly on the diagonal line that forms the top of the 7. The handwritten '5' seems to have the most complicated solution, with specific emphasis along the horizontal line at the top, and along the bottom curve. The way in which each of the optimized solutions seems to smear out may be indicative of the weak translational invariance of the WST, which should be robust to small deformations and translations. Indeed, we can confirm that these patches are robust to translations by computing the predicted class probabilities of shifted version of these patches. Figure 4.10 shows the probability of the optimized patch for class k, written as $\boldsymbol{x}_k^\star$, of being in class k after the patch is shifted i pixels horizontally and j pixels vertically. We consider $-10 \leq i \leq j \leq 10$ in this case, and denote the shifted patch as $\tau_{\boldsymbol{\alpha}_{i,j}} \circ \boldsymbol{x}_k^\star$. When $i = j = 0$ this is the original optimized patch, and we can see that concentrated around the center of these images are all high probabilities, demonstrating the translational invariance of the ST, regardless of the regression penalty term used. It is also interesting to note the directional asymmetry shown for some of the classes translational invariance. For example, the optimized patch associated with $\boldsymbol{x}_7^\star$ worsens in probability with vertical shifts faster than it does with horizontal ones, particularly in the case of the Ridge penalty.



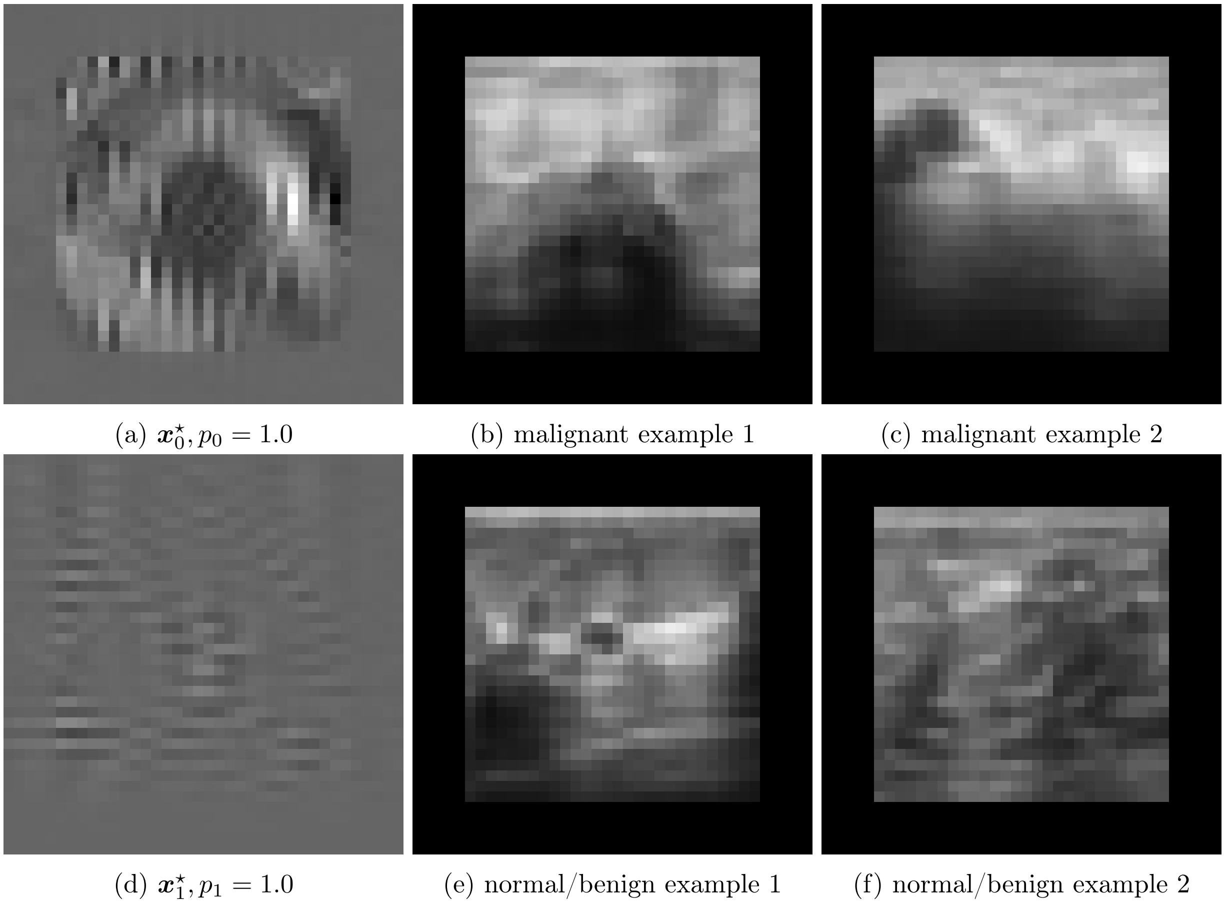
(a) $p(\tau_{\boldsymbol{\alpha}_{ij}} \circ \boldsymbol{x}_4^\star)$ (b) $p(\tau_{\boldsymbol{\alpha}_{ij}} \circ \boldsymbol{x}_5^\star)$ (c) $p(\tau_{\boldsymbol{\alpha}_{ij}} \circ \boldsymbol{x}_6^\star)$ (d) $p(\tau_{\boldsymbol{\alpha}_{ij}} \circ \boldsymbol{x}_7^\star)$ (e) $p(\tau_{\boldsymbol{\alpha}_{ij}} \circ \boldsymbol{x}_9^\star)$

(f) $p(\tau_{\boldsymbol{\alpha}_{ij}} \circ \boldsymbol{x}_4^\star)$ (g) $p(\tau_{\boldsymbol{\alpha}_{ij}} \circ \boldsymbol{x}_5^\star)$ (h) $p(\tau_{\boldsymbol{\alpha}_{ij}} \circ \boldsymbol{x}_6^\star)$ (i) $p(\tau_{\boldsymbol{\alpha}_{ij}} \circ \boldsymbol{x}_7^\star)$ (j) $p(\tau_{\boldsymbol{\alpha}_{ij}} \circ \boldsymbol{x}_9^\star)$

FIGURE 4.10. (a) - (e) The probability that patch $\boldsymbol{x}_k^\star$ belongs to class k after being shifted by i pixels horizontally and j pixels vertically, for the patches derived from the Ridge penalty classifier. (f) - (j) The same for the patches derived from the Lasso penalty classifier.

4.3.3. Explainable Feature Extraction with Breast MNIST. Figure 4.11 shows the optimized inputs for a classifier trained to distinguish normal and benign tumors from malignant ones.

As was done for the MNIST classification, 5 pixels of padding width were added to all training images and a regularization term ($\mu_5 = 0.01$) was used to ensure low energy around the border of the optimized patches. We set all other regularization parameters in this case to 1×10^{-4}, i.e., $\mu_i = 1 \times 10^{-4}$ for $i = 1, 2, 3, 4$.



(a) $\boldsymbol{x}_0^\star, p_0 = 1.0$

(b) malignant example 1

(c) malignant example 2

(d) $\boldsymbol{x}_1^\star, p_1 = 1.0$

(e) normal/benign example 1

(f) normal/benign example 2

FIGURE 4.11. (a) Optimized patch for malignant class; (b)-(c) examples of malignant ultrasounds shown previously; (d) Optimized patch for normal (or benign) class; (e)-(f) examples of normal (or benign) ultrasounds shown previously. Classifier trained with Ridge regression, with regularization parameters $\mu_i = 1 \times 10^{-4}$ for $i = 1, 2, 3, 4$ and $\mu_5 = 0.01$.

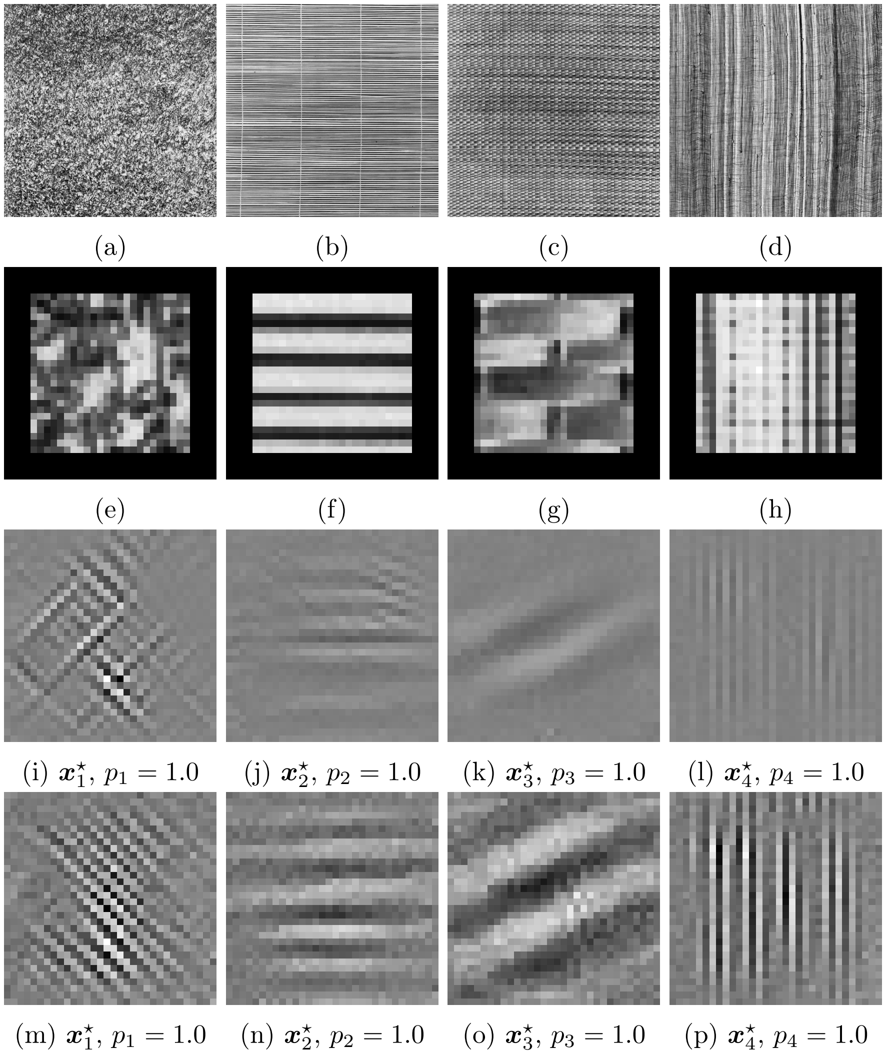
The optimized malignant patch shown in Figure 4.11(a) contains a dark low frequency pattern in the center, whereas the optimized normal (or benign) patch contains no such pattern. Instead, $\boldsymbol{x}_1^\star$ contains only faint horizontal textures which might be present in the absence of an tumor. It seems unlikely that this would discriminate malignant and benign patches, however, as the dark patch shown in Figure 4.4(j) is a benign tumor. The original dataset that Breast MNIST comes from is a set of ultrasound images, each roughly 500 × 500 pixels in resolution. Each image has one of three labels: malignant tumor, benign tumor, or normal tissue. Additionally, for each image there is a corresponding binary which segments any tumorous growth present. It would be interesting, then, to extend the texture segmentation approach outline in the next section to this dataset, and this is an aim of future work with this explainable classification framework.

4.3.4. Explainable Feature Extraction with Brodatz Textures. Before discussing the textures previously shown for the texture segmentation example, Figure 4.12 shows the optimized inputs for a classifier trained on 4 different texture images, chosen for their directional features. D9 has features along many different orientations, whereas D49, D55 and D106 all have fixed orientations in their patterns. We employ pq-norm based sparsity regularization along with the boundary energy regularization and the logarithmic penalty function when solving the optimization problem given in Equation (4.8). For textures with consistent spatial frequencies and orientations, such as D49 and D106, the optimized patches emulate these patterns. However, magnitude fluctuation mitigates these textures, and first centering each patch to have a mean of 0 and standard deviation of 1 accentuates this frequency information. Because the classifier is concerned only with discriminating one class from the other three classes, it may only use the most prominent or unique features of a given class to do so; D9, for example, is a complicated texture. When optimizing the input for this texture without any preprocessing, the output contains multiple orientations. After centering, or applying equalization of brightness, however, a specific direction and frequency seems to be descriptive enough of the texture to discriminate it from the others.

Further, the equalized brightness preprocessing step is also used here, and the corresponding results are shown in Figure 4.13, including example patches after performing EB.

For each experiment run with the Brodatz textures, the regularization parameters were set to $\mu_1 = \mu_2 = \mu_4 = 0$, and $\mu_3 = \mu_5 = 1 \times 10^{-4}$. That is, we encourage sparsity in the original image space, and low edge energy.

4.3.5. Discussion. The explainable features extracted from these two classification tasks shows the potential to understand local features which discriminate a class from all others under consideration. In the case of the Brodatz dataset, textures with distinct orientations and spatial frequency information are easiest to discriminate between, which we find to be most clearly shown in Figure 4.12, Figure 4.13 and Figure 4.14. Of the Brodatz 2 textures, D9, D49, D55 and D106, all



(a) (b) (c) (d)

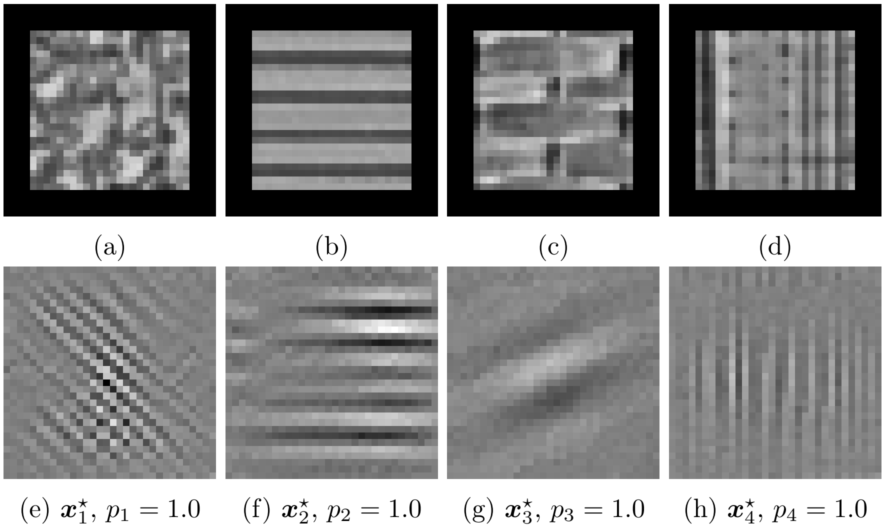
(e) (f) (g) (h)

(i) $\boldsymbol{x}_1^\star$, $p_1 = 1.0$ (j) $\boldsymbol{x}_2^\star$, $p_2 = 1.0$ (k) $\boldsymbol{x}_3^\star$, $p_3 = 1.0$ (l) $\boldsymbol{x}_4^\star$, $p_4 = 1.0$

(m) $\boldsymbol{x}_1^\star$, $p_1 = 1.0$ (n) $\boldsymbol{x}_2^\star$, $p_2 = 1.0$ (o) $\boldsymbol{x}_3^\star$, $p_3 = 1.0$ (p) $\boldsymbol{x}_4^\star$, $p_4 = 1.0$

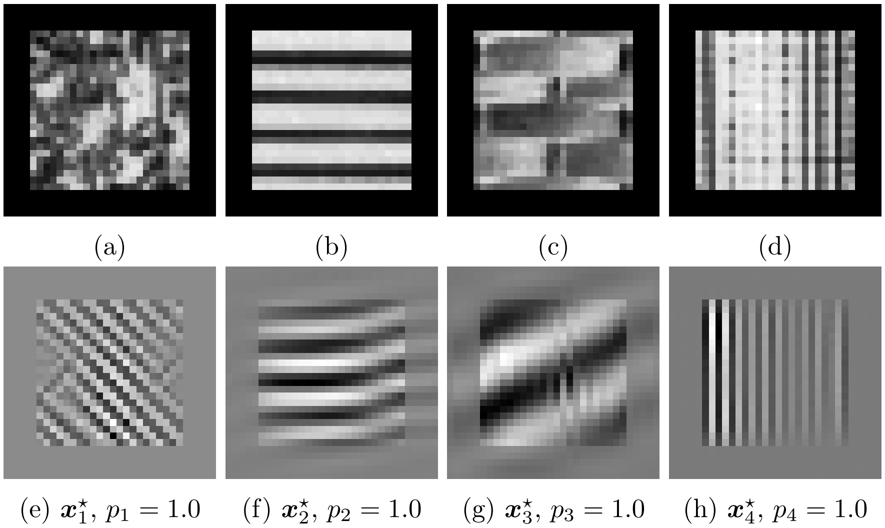
FIGURE 4.12. (a) - (d) Brodatz textures D9, D49, D55, D106; (e) - (h) 32×32 patches from each of the four textures, 4 pixel zero padding. (i) - (l) Explainable features derived via Equation 4.8 from each of the four textures, with $\mu_3 = \mu_5 = 1 \times 10^{-4}$, and $L(p_k) = -\log(p_k)$; (m) - (p) The derived features for a classifier trained on centered patches, each with mean 0 and standard deviation 1, using the same regularization parameters and loss function.

but D9 have clearly oriented patterns, and the optimized patches, regardless of which data augmentation was performed, exhibit the same patterns. The optimized patches for D9 (e.g., Figure 4.12(i)(m)) still exhibit clearly orientation preferences, although without data augmentation several orthogonal orientations appear. Finally, Figure 4.15 shows similar results for the Brodatz 1 textures, used previously to describe texture segmentation problem (see Figure 4.5). The textures in this problem have more complicated pattern structure. Still, the optimization problem converges


(a) (b) (c) (d)

(e) $\boldsymbol{x}_1^\star$, $p_1 = 1.0$ (f) $\boldsymbol{x}_2^\star$, $p_2 = 1.0$ (g) $\boldsymbol{x}_3^\star$, $p_3 = 1.0$ (h) $\boldsymbol{x}_4^\star$, $p_4 = 1.0$

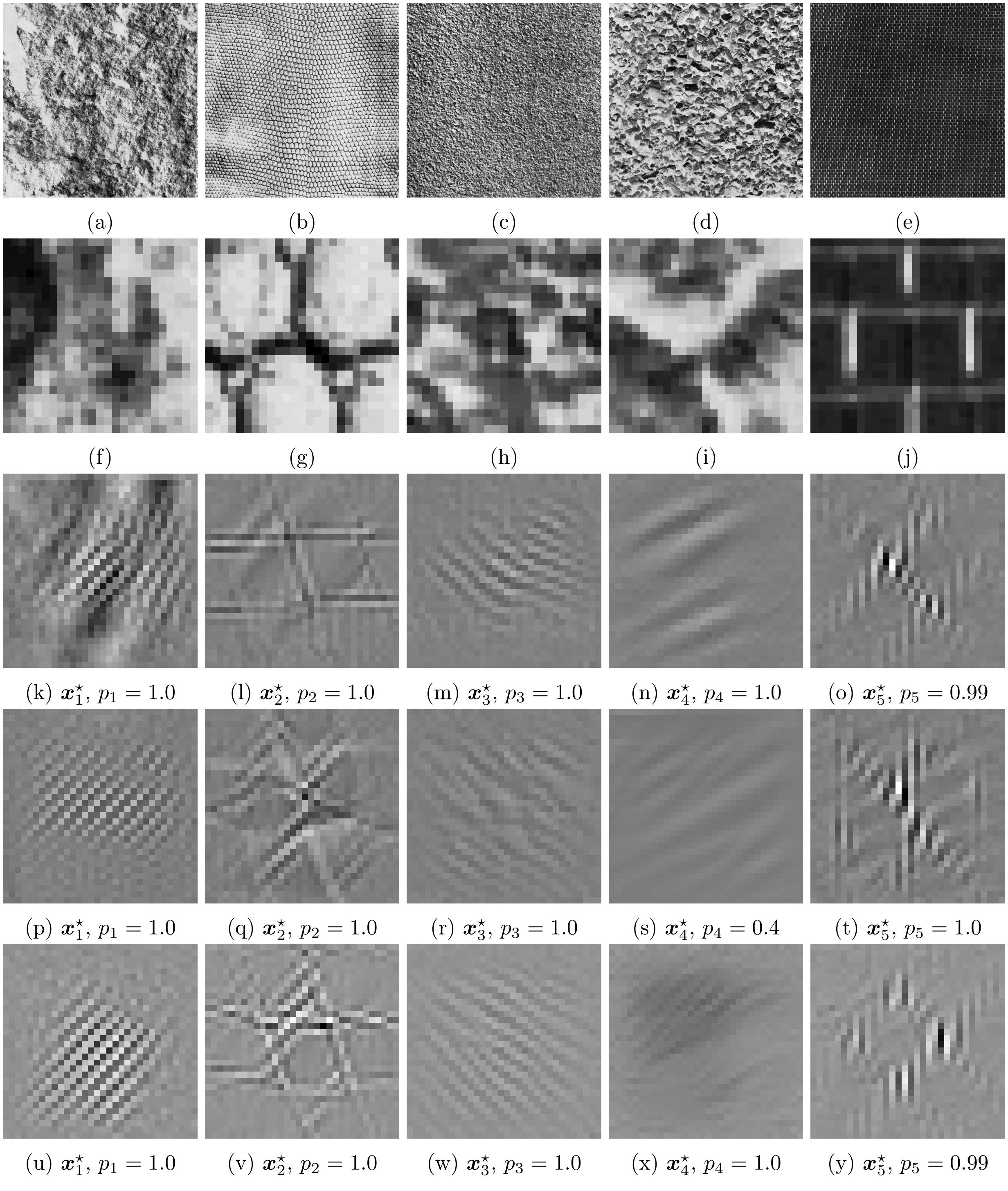
FIGURE 4.13. (a) - (d) Patches after equalization of brightness. (e) - (h) The derived features for a classifier trained patches of equalized brightness, using the same regularization parameters and loss function.


(a) (b) (c) (d)

(e) $\boldsymbol{x}_1^\star$, $p_1 = 1.0$ (f) $\boldsymbol{x}_2^\star$, $p_2 = 1.0$ (g) $\boldsymbol{x}_3^\star$, $p_3 = 1.0$ (h) $\boldsymbol{x}_4^\star$, $p_4 = 1.0$

FIGURE 4.14. (a) - (d) Patches after equalization of brightness. (e) - (h) The derived features for a classifier trained patches of equalized brightness, using the same regularization parameters and loss function.

and the optimized patches for each class exhibit certain features of that class. For example, As far as we are know, this is the first attempt at solving such an optimization problem for 2D image classification. The use of zeroth-order optimization here is prompted by the highly discontinuous nature of the gradient of the WST function, but also enables the use of non-convex constraints and opens up many interesting future directions. The exciting future direction of this work is the same as that done in the 1D setting [**74**], which is to derive explainable features that enable practitioners to gain insight into the problem they are trying to solve, while also allowing researchers to build simpler classification models after deriving these optimized inputs. In the case of the Breast MNIST dataset, for example, the optimized input for malignant tumors seems to indicate a spatial bias in the location of these tumors. If that bias were present, this feature could be exploited to build more accurate classifiers. For the case of the texture segmentation with the Brodatz datasets, specific directional filters could be applied to the texture patches rather than computing the scattering transform. This would be computationally less expensive, and could lead to similar classification results.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

(k) $\boldsymbol{x}_1^\star$, $p_1 = 1.0$ (l) $\boldsymbol{x}_2^\star$, $p_2 = 1.0$ (m) $\boldsymbol{x}_3^\star$, $p_3 = 1.0$ (n) $\boldsymbol{x}_4^\star$, $p_4 = 1.0$ (o) $\boldsymbol{x}_5^\star$, $p_5 = 0.99$

(p) $\boldsymbol{x}_1^\star$, $p_1 = 1.0$ (q) $\boldsymbol{x}_2^\star$, $p_2 = 1.0$ (r) $\boldsymbol{x}_3^\star$, $p_3 = 1.0$ (s) $\boldsymbol{x}_4^\star$, $p_4 = 0.4$ (t) $\boldsymbol{x}_5^\star$, $p_5 = 1.0$

(u) $\boldsymbol{x}_1^\star$, $p_1 = 1.0$ (v) $\boldsymbol{x}_2^\star$, $p_2 = 1.0$ (w) $\boldsymbol{x}_3^\star$, $p_3 = 1.0$ (x) $\boldsymbol{x}_4^\star$, $p_4 = 1.0$ (y) $\boldsymbol{x}_5^\star$, $p_5 = 0.99$

FIGURE 4.15. (a) - (e) Brodatz textures D2, D3, D4, D5, D6. (f) - (j) 32 × 32 patches from each of the five textures, 4 pixel zero padding. (k) - (o) Explainable features derived via Equation 4.8 from each of the five textures. (p) - (t) the same after centering each patch. (u) - (y) the same equalizing the brightness of each patch.

CHAPTER 5

Automated Morphometrics & Weight Prediction of Juvenile Chinook Salmon

In this chapter we leverage an open-source deep learning model to extract meaningful morphometric features from images of juvenile salmon. The open-source model used here is Meta's Segment Anything Model (SAM) to initially process the images, and then new method and algorithms are developed in order to derive morphometric features from these processed images. Before these methods are described some background for this project and this problem is provided.

Chinook Salmon (*Oncorhynchus tshawytscha*) populations along the California coast are struggling due to many factors including overfishing and loss of habitat. These fish are an important source of food both commercially and for recreational fisheries. Further, Chinook Salmon are important for Indigineous communities, historically serving as an vital source of food and cultural significance. Juvenile Chinook Salmon rearing and migratory habitat has been degraded, limiting the opportunity for the expression of diverse of life history strategies. (Kareiva et al. 2000 [**49**], David et al. 2016 [**27**]). There is a large ongoing effort to restore these habitats and fish populations to healthy levels including dam removal projects, habitat restoration, and novel management actions. Further, it is known that certain morphometric measurements of juvenile fish are indicative of overall fish health, such as Fulton's condition factor (K) defined as $K = 10^5 \times W \times L^{-3}$, where W is weight and L is length (Ricker 1975 [**71**]). The work described here is important for this effort in two majors ways. First, it proposes an practical platform to measure accurate morphometric data of individual fish based on a side-profile image, allowing for fast and consistent measurement in many environments. Second, it is designed specifically to minimize fish handling, keeping the fish submerged in water throughout, directly addresses a growing need for sustainable and non-invasive methods to protect fish health during measurement (Sharpe et al. 1998 [**76**]).

In 2021 Holmes and Jeffres [**42**] proposed a morphometric model for weight prediction and condition factor prediction for juvenile Chinook Salmon as manual weighing and measuring of individual juveniles is challenging and time consuming work, and is prone to user error. However, the time required to digitize each morphometric point on every fish remains a bottleneck in the data processing pipeline, and, moreover, must be done by a trained user. Here, to collect more accurate and reproducible information, as well as minimize handling of the fish, the HandsFreeFishing program has been developed, which uses an open-source deep-learning based segmentation models to automate much of the data processing pipeline. This program requires minimal user input for each individual image, while allowing researchers to measure important morphometric quantities of interest. Meta's open-source Segment Anything Model (SAM) [**52**] was used here in order to segment individual fish, then customizable algorithms were applied to extract relevant morphometric features, such as fork length and surface area, which were later eventually used to predict fish weight. There have been several previous attempts to predict fish weight using image data and deep learning. Yang et al. [**90**] (2021) built a deep convolutional neural network to predict the weight of fish through 2D image data. Rantung, Sappu and Tondok [**70**] (2021) used stereo vision, requiring several images of a fish from different angles, to predict fish surface area and volume. The approach developed here is more similar to the [**70**], as we focus on estimating well defined morphometric features which are then used to predict fish weight, though we use only one side-profile image per fish, as done in [**90**]. Lastly, another automatic morphometric model was proposed by Kristiansen et al. in 2025 [**54**], which trains a machine learning model to place landmark points of zebrafish. In this work, training data is used only to predict fish weight; all other morphometric features are computed directly from the fish segmentation and contour.

The study of this chapter focused on juvenile Chinook Salmon (fork length 27-90mm and weight 0.31g - 7.74g), but the program developed is open-source and may be easily customized to extract morphometric features from image data of other species as well.

5.1. Methods for Extracting Morphometric Features

A conceptual overview of the methodology is presented in Figure 5.1. All necessary software to reproduce these results can be found at https://github.com/briancknight/HandsFreeFishing, where installation steps and examples are outlined. It requires Python 3.10 or later, and access to Meta's open-source Segment Anything Model (SAM) [**52**].

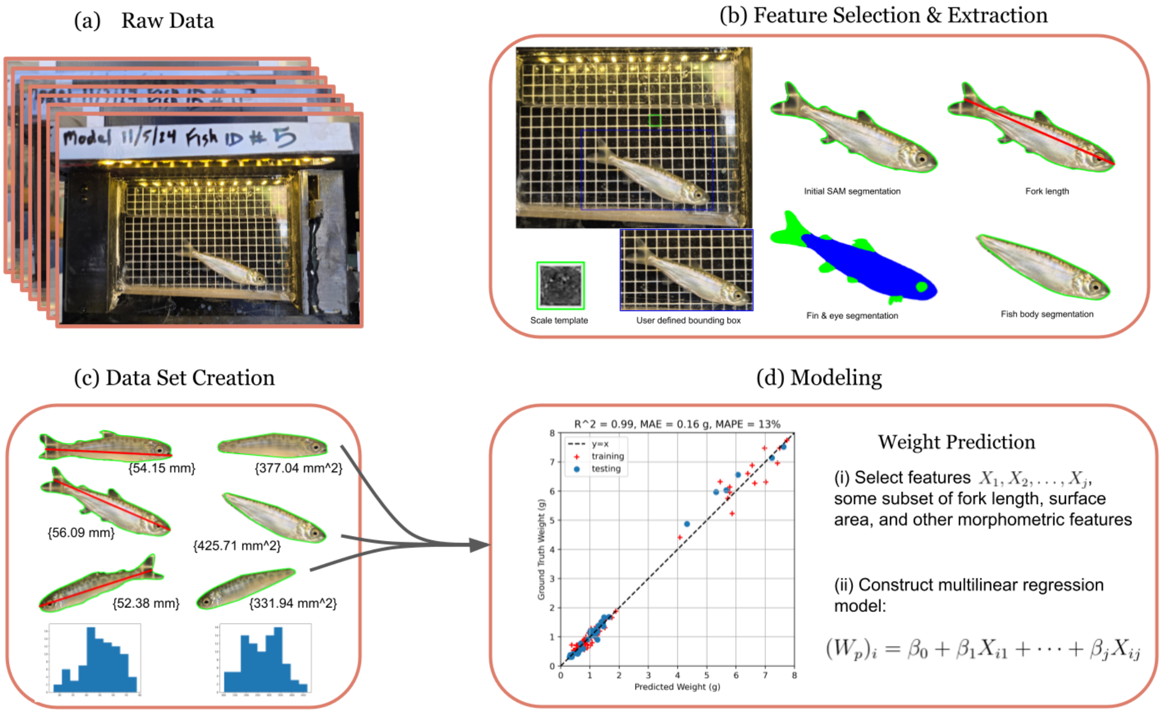
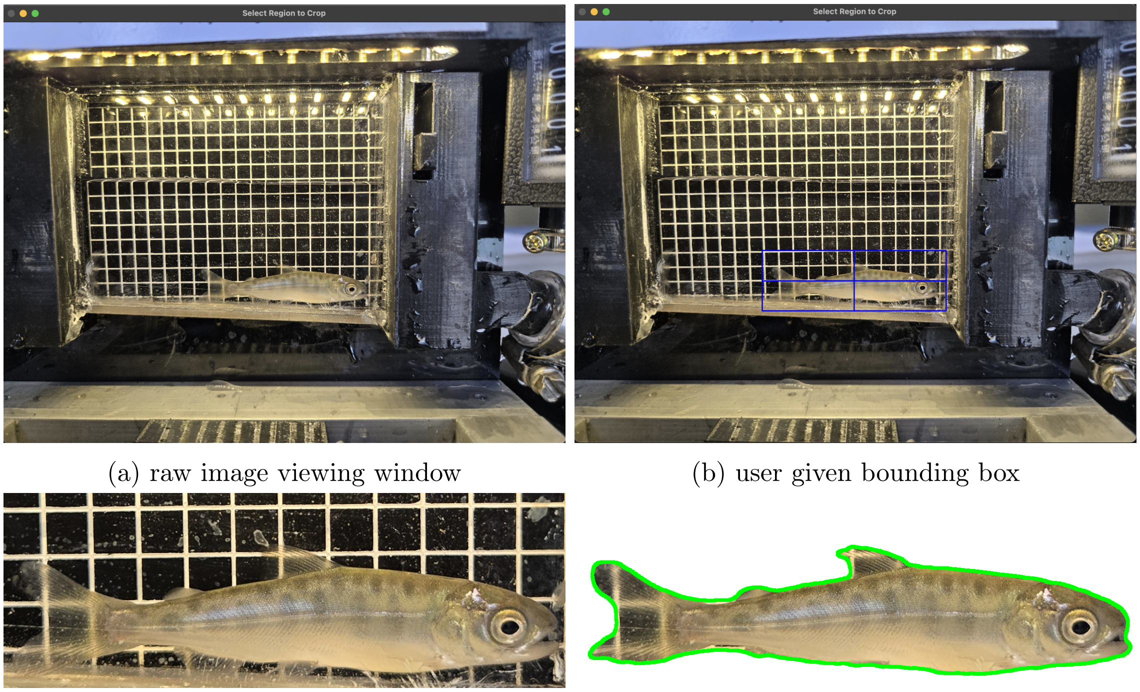

(a) Raw Data
Model 11/5/24 Fish ID # 5
(b) Feature Selection & Extraction
Initial SAM segmentation
Fork length
Scale template
User defined bounding box
Fin & eye segmentation
Fish body segmentation
(c) Data Set Creation
{54.15 mm}
{377.04 mm^2}
{56.09 mm}
{425.71 mm^2}
{52.38 mm}
{331.94 mm^2}
(d) Modeling
R^2 = 0.99, MAE = 0.16 g, MAPE = 13%
y=x
training
testing
Ground Truth Weight (g)
Predicted Weight (g)
0
1
2
3
4
5
6
7
8
Weight Prediction
(i) Select features $X_1, X_2, \ldots, X_j$, some subset of fork length, surface area, and other morphometric features
(ii) Construct multilinear regression model:
$(W_p)_i = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_j X_{ij}$

FIGURE 5.1. A conceptual overview of the proposed data processing pipeline.

5.1.1. Image Collection. All fish images were acquired using a point and shoot digital camera of fish inside a viewer developed to collect images of live juvenile salmon up to 90mm (Figure 5.1). The viewer held fish in water at a side profile where the back side of the viewer contained a 5x5mm grid used for standardizing a digital length in the images. The viewer contained lights above the viewing chamber to illuminate the fish to ensure fast enough shutter speeds to collect sharp, in focus images. Following image acquisition, the fish were returned to a holding bucket and ultimately returned to their rearing tank.

5.1.2. Initial Image Processing. Once aggregated, each image was manually inspected and given a digital bounding box and orientation (left facing or right facing, upside down or right-side up). The overall image quality was assessed at this stage too, to ensure any unusable images were discarded. An example of this interface is shown in Figure 5.2 (a)(b). The user interface is designed to be simple, reading each image name in from a corresponding spreadsheet or csv file.



(a) raw image viewing window

(b) user given bounding box

(c) raw image, cropped

(d) initial segmentation via SAM

FIGURE 5.2. (a) Raw image displayed by our program, waiting for a user-supplied bounding box. (b) A bounding box is supplied by the user with two clicks. Once the bounding box is confirmed, the user is asked for minimal orientation information regarding the fish. (c) The raw image (d) The initial segmentation via SAM.

5.1.3. Automated Segmentation. After initial image processing, Meta's Segment Anything Mode (SAM) [**52**] is used to extract an initial segmentation of the target fish, as shown in Figure 5.2(c)(d). Further, a smooth contour of the fish outline is computed using elliptic Fourier analysis, which yields a discretized smooth 2D curve $c = [c_0, c_1, \ldots, c_n]$ where $c_i = (x_i, y_i)$. The mesh grid from the fish viewer (in the background of each image) is used to estimate the pixel to millimeter scale. This estimate is done by template matching, finding a grid box which we know to represent a 5 by 5-millimeter square. Using the initial segmentation, contour and scale estimate, the program estimated the size location of individual fins and again applied SAM with bounding boxes computed from these estimates. Examples of successful and unsuccessful segmentations from this procedure are shown in Figure 5.4(a)(b). With individual fins segmented, they are then removed from the initial segmentation, yielding a 'no fin' segmentation (Figure 5.4(d)). Because fish fins can be in several different positions (up or down or in-between), they can influence the side-profile surface area which can affect weight prediction. As such it has been determined to remove fin segmentations [42]. In addition, the eye of the fish was segmented to measure eye diameter, a potentially important morphometric measurement.

5.1.4. Feature Extraction. For all feature extraction, the fish is assumed to be right-side up and facing left. The orientation data supplied by the user ensures each image is rotated and flipped appropriately, so that the program can process each contour in the same fashion.

5.1.4.1. *Fork Length & Surface Area Measurement .* The fork length, the distance from the tip of the jaw or snout with closed mouth to the center of the fork in the tail, is a standard measurement in fish ecology. To measure the fork length of each fish, the shape contour is first rotated so that the left-most point and right-most point lay along a horizontal line. After rotation, the new left-most point of the contour is taken to be the tip of the fish's head. To find the correct point of the caudal fin the program searched along the opposite end of the contour for two consecutive local maxima in the x-component which would represent the upper and lower lobes of the caudal fin; these contour point are called c_1^* and c_2^*. If these points are found successfully (Figure 5.3(a)-(c)), then the program now has a segment of the contour $[c_1^*, \dots c_2^*]$ within which it will find the point with minimum x-component. If it cannot find the upper and lower lobe points (Figure 5.3(d)), it assumes the caudal fin is not appropriately segmented, and instead does the following: consider only the contour points whose x-component is larger than some threshold (this gives only the contour of tail-end of the fish). Then, compute the average y-component (height) of these contour points, and find the point c^* whose height is closest to this average height. This should be near where the midpoint of the caudal fin would be given a proper segmentation.

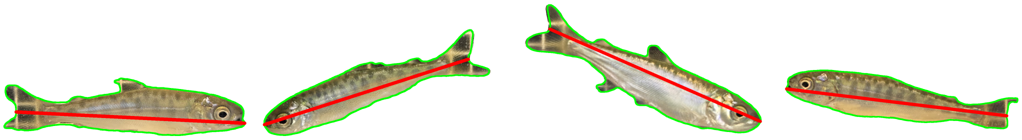

(a) 53.60 mm (b) 52.60 mm (c) 56.08 mm (d) 53.52 mm

FIGURE 5.3. Fork length predictions following the procedure outline in section 5.1.4.1.

The automated program uses similar style procedures for segmenting individual fins: caudal, adipose, dorsal, pectoral, pelvic, and anal, as well as for segmenting the eye of the fish. For high quality images, this produces good fin segmentations which can be used to measure various morphometrics researchers may be interested in. Details on the heuristics we use for each of these are included in the Appendix B, but these heuristics are customizable, which is a feature of the program. Figure 5.4 shows some examples of successful and unsuccessful fin segmentation, and corresponding 'no fin' fin segmentations. The 'no fin' segmentations are of interest because are highly correlated with the fish volume, which is directly related to fish weight via density. For more discussion on weight prediction, see Section 5.1.5. If fin segmentation is reliable, the individual surface area measurements could be used to improve a weight prediction model, or could be measured for other purposes. In this case the fin segments were removed from segmentation, as the fish can move them up or down resulting in less reliable segmentation. Additionally, because the fin density varies drastically from the density of the rest of the body it can lead to erroneous weight predictions. To remove the fins from the segmentation, masked areas were set equal to zero. When this step is performed, the resulting segmentation may have multiple connected components, or, when fin segmentation is unsuccessful, may remove too much area from the body of the fish. To mitigate these issues, two steps are performed. First, only the largest connected component of the segmentation is kept. Second, using the contour of the segmentation, a convex hull of the contour points is computed. So long as the area of the convex hull is not significantly larger than the area of the segmentation, this convex hull is kept as the final segmentation used in further processing.

For the remainder of the chapter, fish segmentation and fish contour will refer to the 'no fin' segmentation and 'no fin' contour respectively (e.g. the right most image of Figure 5.4(a)).

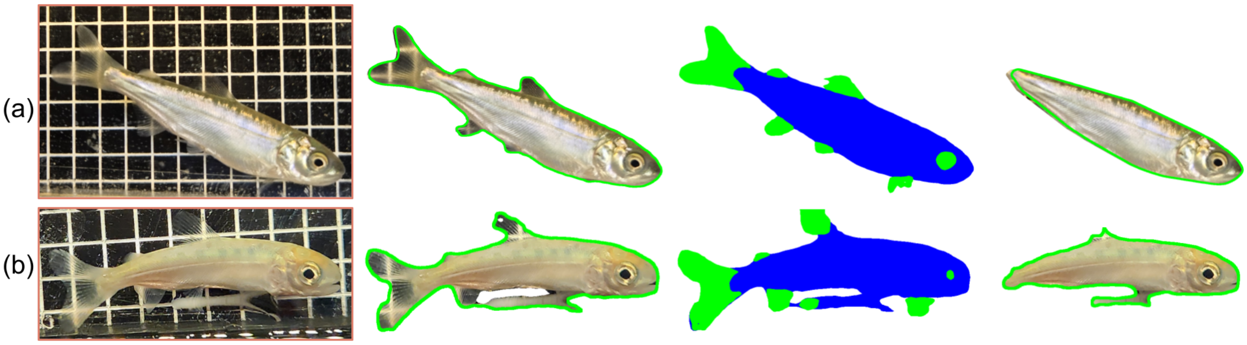


FIGURE 5.4. Two examples of automated segmentation and contour estimation after initial bounding box: (a) successful, (b) unsuccessful.

5.1.4.2. *Other morphometric features.* In addition to the fork length and surface area estimates, the program also computed eye diameter based on the eye segmentation mentioned previously, as shown in Figure 5.5(b). The estimated diameter is computed as the maximum distance between any two points along the contour of the eye segmentation.

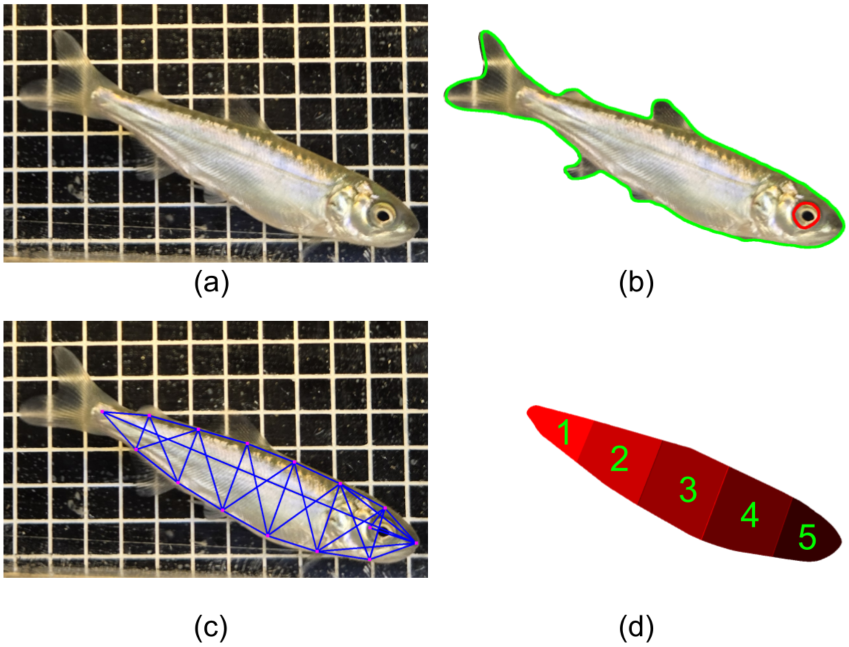


FIGURE 5.5. Other morphometric features: (a) raw image, cropped; (b) eye contour used for eye diameter; (c) landmark points & lengths; (d) partitioned SA.

To compare previously described morphometric modeling **[42]**, the program automated the placement 16 morphometric points the fish image. In contrast to previous work (Holmes and Jeffres 2021), these landmark points were not based upon fin locations, but instead placed at 6 equally spaced points in between the left and right most point of the contour, along with a point just behind the eye. A truss lattice was then constructed between these points, and the length of each of the line segments shown in Figure 5.5(c) were measured. Even without the correct anatomical placement, these length measurements provide similar data about the size and shape of the fish.

Additionally, the program can compute partitioned surface areas, which could allow a model to learn a non-constant density function across the body of the fish (Figure 5.5(d)). Lastly, we found the ellipse of best fit through the points defining our contour, and measured the major and minor semi-axes, a and b, of this ellipse, which are proximal measurements for the length and height of the fish respectively. A visualization of these quantities is shown in Figure 5.6.

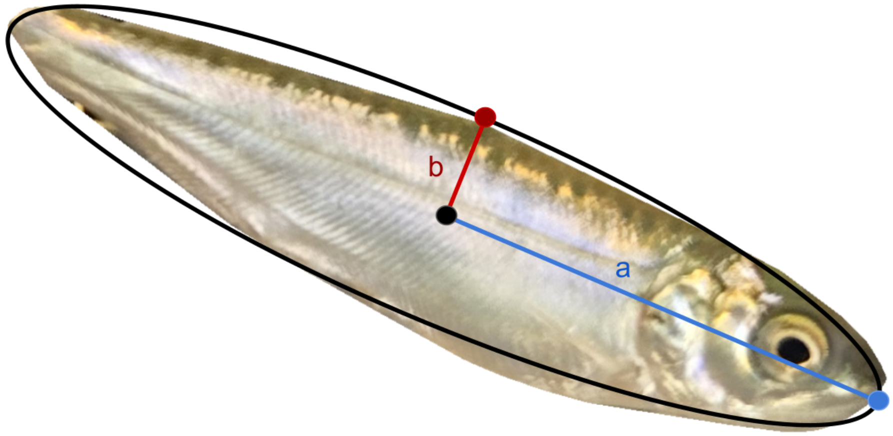


Figure 5.6. Ellipse of best fit for a given fish segmentation, with principal semi-axes labeled.

If the contour is well fit by an ellipse then the area of this ellipse, given by $A = \pi ab$, should be well approximate the surface area measured. Thus this process could also be used to automatically check the success of the segmentation.

5.1.5. Weight Prediction Procedure. The dataset consisted of 149 images and ground truth weight measurements. A curated version of this dataset was also used, consisting of 109 images where low quality image or images with unwanted segmentation artifacts were removed. The HandsFreeFishing program was employed to measure many different morphometric features. Then, based on these features and additional modeling assumptions, several linear regression and multiple linear regression (MLR) models to predict juvenile fish weight. The MLR models were all trained in Python using the Scikit-Learn package.

5.2. Explainable Weight Prediction Using Morhpological Features

For several weight prediction models, the shape of the fish body was assumed to be a 3D ellipsoid with semi-axes a, b, and c, denoting the horizontal semi-axis (length), the vertical semi-axis (height), and the depth semi-axis (girth), respectively, as depicted in Figure 5.7.

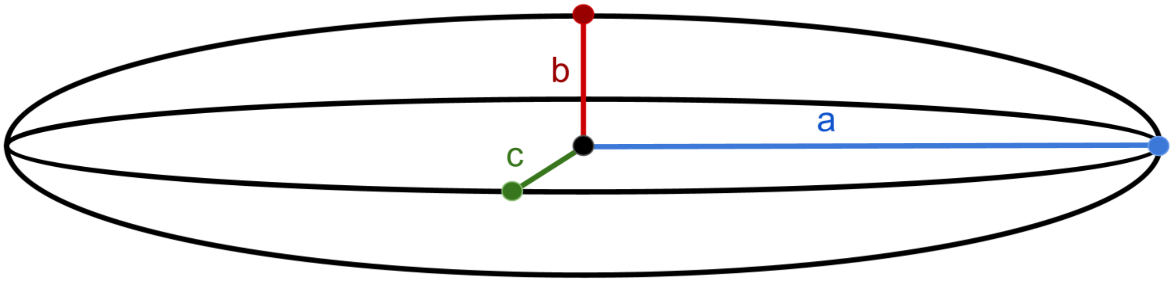


FIGURE 5.7. Ellipse of best fit for a given fish segmentation, with principal semi-axes labeled.

The volume V of an ellipsoid with these principal semi-axes is given by $V = \frac{4}{3}\pi abc$, and that the area of a 2D ellipse with principal semi-axes a and b is given by $A = \pi ab$. The surface area measured here is similar to the area of an ellipse, since the image is a 2D projection of the fish. Making the simplifying assumption that girth is directly proportional to height, it follows that $V \propto SA \cdot b$, and since volume should be directly proportional to weight, the quantities $V_1 = SA \cdot b$ and $V_2 = ab^2$ should be directly related to weight via density. Further still, if it is assumed that height is directly proportional to length, i.e. all fish have the same proportions, then the model simplifies to $V \propto b^3$, and therefore $V \propto SA^{\frac{3}{2}}$. The quantities V_1 and V_2 and $SA^{\frac{3}{2}}$ were all used in a standard 1D linear regression with the ground truth weight measurements to train Model 1, Model 4, and Model 2, respectively, of which V_1 and $SA^{\frac{3}{2}}$ gave the most accurate results. The model believed to perform the best is Model 1, which achieved an r-squared statistic of 0.99, a mean average error of 0.16g, a mean average percentage error of 12% and an AIC score of -19.98. This is chosen over Model 2 even though Model 2 achieves a better AIC score on the cleaned dataset, because Model 2 implicitly assumes the length of the fish is proportional to the height and girth, which may be a bad assumption when there is more variance in the fish condition factors, for instance.

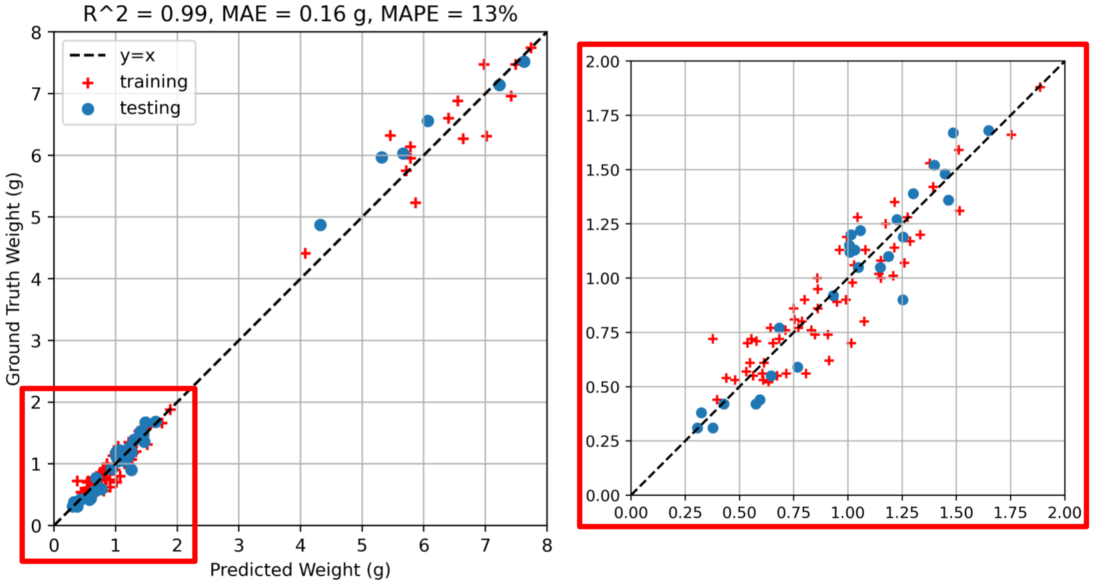


FIGURE 5.8. Multilinear regression between SA, SA^2, and ground truth weight measurements. Model statistics shown are averages from 5 fold cross validation.

Several other models were considered as well. Model 3 is based on surface area, SA, and major and minor axis terms a and b respectively. It uses these along with all second order combinations of SA, a, and b as explanatory variables, which consequently includes V_1. This model achieved results similar to using V_1 alone, but scored a worse in AIC and overall accuracy (Tables 6.1, 5.2). Model 5 uses the partitioned surface areas multiplied by the height estimate b as the explanatory variables. This model also performed well, but worse than simpler models in all categories. Finally, landmark length models, Model 6 and Model 7, are interesting to consider. They are trained using all the truss lengths computed as shown in Figure 5.5(c). Interestingly, considering these lengths cubed, which is motivated simply by volume being proportional to length cubed, the landmark lengths become much better predictors of weight. These models are summarized below:

- **Model 1**: a linear regression is performed with the quantity $V_1 = SA \times b$; this model implicitly assumes $b \propto c$.
- **Model 2**: a linear regression is performed with the quantity $SA^{\frac{3}{2}}$; this model implicitly assumes $a \propto b \propto c$.
- Model 3: a multilinear regression is performed with the quantities SA, a, b, SA^2, $SA \times a$, $SA \times b$, a^2, $a \times b$, b^2; this model implicitly assumes $a \propto b \propto c$.
- Model 4: a linear regression is performed with the quantity $V_2 = a \times b^2$; this model implicitly assumes $b \propto c$, and further that $a \times b$ approximates the SA of the side profile of the fish well.
- Model 5: a multilinear regression is performed on the quantities $v_j \times b$, where v_j is the surface area of the jth partition.
- Model 6: a multilinear regression is performed with each of the landmark lengths cubed;
- Model 7: a multilinear regression is performed with each of the landmark lengths;

All weight prediction results are summarized in Tables 6.1 and 5.2

TABLE 5.1. Cleaned dataset, 109 images

model name	explanatory variable(s)	r^2	MAE	MAPE	AIC
Model 1	$V_1 = SA \times b$	**0.99**	**0.16**	**12%**	-19.98
Model 2	$SA^{\frac{3}{2}}$	**0.99**	**0.16**	13%	**-24.76**
Model 3	SA, major, minor (ord=2)	**0.99**	**0.16**	13%	-14.40
Model 4	$V_2 = a \times b^2$	0.98	0.17	14%	12.16
Model 5	(Partitioned SA) $\times b$	0.98	0.17	13%	16.66
Model 6	(Landmark Lengths)^3	0.98	0.19	15%	52.72
Model 7	Landmark Lengths	0.84	0.50	55%	232.22

[1] SA denotes surface area.
[2] a and b denote the semi-axes of the ellipse of best fit through the segmentation contour.
[3] ord=2 denotes that all second order combinations of the variables listed were also used as regression variables.
[4] bold face denotes best performance

The r-squared value, mean average error (MAE), mean average percentage error (MAPE), and Akaike information criterion (AIC) **[3]** score is reported for each. Tables 6.1 shows results for a cleaned data set, where segmentations were manually inspected and images yielding poor segmentation results were discarded (109 images). As the goal of this model is to provide accurate weight data for unmeasured fish, this was done to achieve higher performance. On the other hand, Tables 5.2 second contains all 149 images, which is included to indicate that even with imperfect

TABLE 5.2. Full dataset, 149 images

model name	explanatory variable(s)	r^2	MAE	MAPE	AIC
Model 1	$V_1 = SA \times b$	**0.98**	**16%**	**0.13**	**9.69**
Model 2	$SA^{\frac{3}{2}}$	**0.98**	0.17	14%	13.79
Model 3	SA, major, minor (ord=2)	**0.98**	**0.16**	**0.13**	26.14
Model 4	$V_2 = a \times b^2$	0.98	0.18	14%	37.51
Model 5	(Partitioned SA) $\times b$	0.98	0.18	14%	40.35
Model 6	(Landmark Lengths)^3	0.94	0.21	19%	79l90
Model 7	Landmark Lengths	0.88	0.46	0.51	293.29

[1] SA denotes surface area.

[2] a and b denote the semi-axes of the ellipse of best fit through the segmentation contour.

[3] ord=2 denotes that all second order combinations of the variables listed were also used as regression variables.

[4] bold face denotes best performance

segmentation, accurate weight prediction is still feasible. All regression results are computed via 5-fold cross validation, using Scikit-Learn's regression model and cross validation scoring function.

5.2.1. Discussion. The HandsFreeFishing program extracted many important morphometric features from raw images of juvenile Chinook Salmon, including fork length, surface area, and eye diameter. This allows for the collection of important physical characteristics of juvenile salmon without the necessity of removing the fish from water, and the data extracted can be used for a variety of applications in fisheries ecology and management. Further, this work removes the current bottleneck in the morphometric data processing pipeline which will help expedite post processing of data and provide valuable morphometric datasets, including potential morphometrics that have not previously been measured.

Additionally, it addresses a growing concern with fish handling, as it should eventually allow a single in-water photo of an individual to be sufficient for measuring all important morphometric data. Fish handling often results in increased stress for fish [**76**]. Programs like this can help to minimize fish stress while still collecting critical metrics used to make management decisions. In addition to reduced handling and stress, additional metrics such as weight and condition factor can now be collected simultaneously in a single image. While working with threatened and endangered species these metrics can provide critical information to quantify success of various management actions while minimizing handling and stress on the fish.

For the purpose of predicting the weight of juvenile salmon, the program measured relevant features like surface area, fish height, and fork length. The actual weight of juvenile salmon were measured using a digital scale, with the juveniles ranging from 0.3 to 7.8 grams. Several linear regression models were then trained to predict weight from the measured features. The most successful models assumed that a fish body can be well approximated by an ellipsoid in 3D, who's girth is proportional to its height. From this assumption it follows that the volume of the fish body is proportional to the surface area multiplied by the height. Therefore, by predicting both surface area and the fish height, the weight can be estimated accurately. This is verified empirically here, as Model 1 obtained the best prediction results and second best AIC score.

Collecting more ground truth weight data for the predictive models to be trained on will allow for more accurate and potentially more complex models. Model 5 and Model 6, for example, suffer due to the size of the training dataset, and could potentially outperform Model 1 if trained on a larger number of fish. Other future work with this program could involve new feature extraction methods and algorithms for accurate anatomical landmark point placement. For example, no attempt has yet been made to segment the midline, which could help with orienting the fish and improve the fork length estimate. Predictive models of other morphometric measurements, such as condition factor, would be an interesting direction to pursue.

5.2.2. Conclusion. By adapting a state of the art segmentation model like Meta's Segment Anything Model [**52**], highly accurate segmentations of juvenile Chinook salmon are readily attained. Relying on this accurate segmentation data, the HandsFreeFishing program is able to extract a wide variety of morphometric features automatically. The approach taken here, and a lot of the software and methods developed, should be adaptable to other species and life stages, and the use of the HandsFreeFishing program to create a growing database of these morphometric features is an exciting prospect and highly encouraged.

CHAPTER 6

Tracking of Surface Protrusions via Multi-timepoint 3D Displacement Vector Fields

This final chapter lies at the intersection of image processing and computational developmental biology. As imaging devices become more advanced, the level of detail that can be studied in biological systems increases, enabling many interesting developmental questions to be studied quantitatively. This project began during a summer internship in 2024, and is part of a larger ongoing effort to study developmental dynamics in fruit flies (*Drosophila melanogaster*) at the Center for Computational Biology at the Flatiron Institute, with collaborators in the Department of Molecular Biology at Princeton. The specific goal of the project is to understand the dynamics of particular cells, *primordial germ cells* [**66**], developing in a fruit fly embryo, and to develop methods to quantify the certain dynamics of this development and track quantitative features of these cells through time.

The cell is the most basic unit of complex multicellular organisms like plants and animals; in many cases it remains unclear how the development and homeostasis of an organism are related to the dynamics at the cellular scale. To quantitatively probe this question, it is often useful to virtually decompose an organism into cells by 3D cellular instance segmentation. Increasingly sophisticated methods for microscopy and segmentation have enabled large-scale 3D reconstructions of complicated cell shapes in diverse contexts [**32**, **80**, **85**]. Because cells are typically segmented on the basis of images of their membranes [**47**,**80**], the difficulty of instance segmentation often depends sensitively on the geometric and topological properties of the cell surface. In many biological contexts, including during development, a key difficult case arises: a cell can be "incomplete", meaning that its membrane does not form a closed surface. Such cases occur during cell fusion, including in muscle and placental development, by which a cell fuses into an existing cell. This also occurs during cellularization, including in early insect embryogenesis and plant endosperm formation, by which cells form out of an existing cell. Here we propose a method to generate

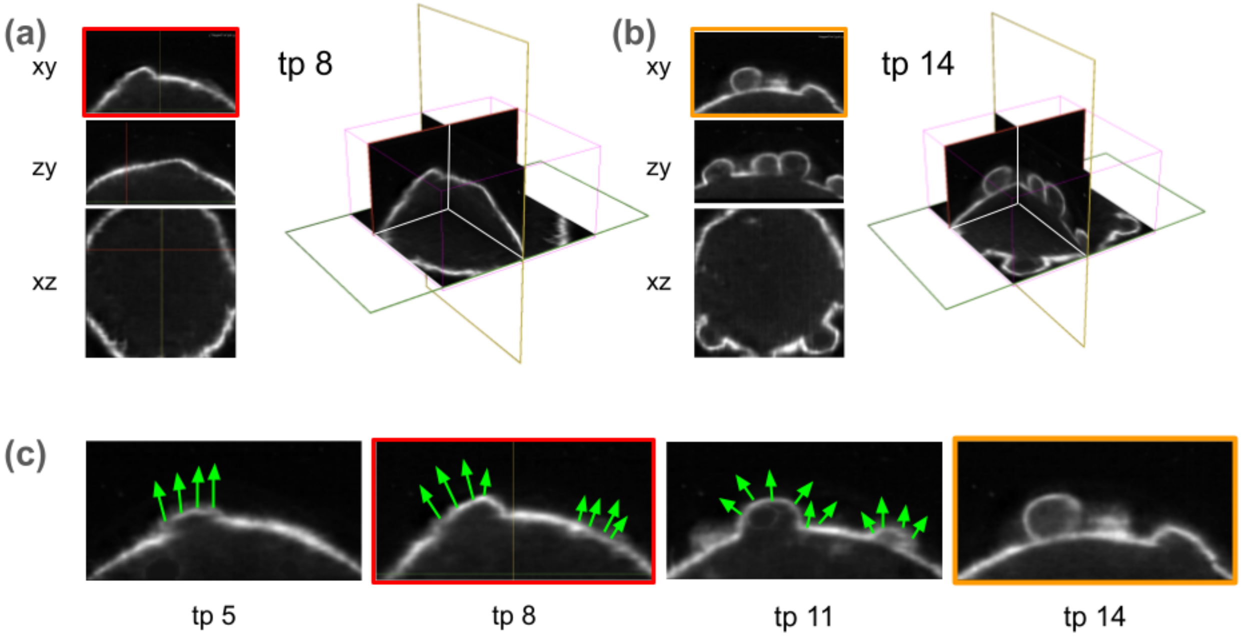


FIGURE 6.1. (a) Left: xy, zy, and xz slices of the 3D volume 8 minutes into the 9th nuclear cycle. Right: 3D visualization of these slices, (b) Same as (a), but for minute 14 during the 10th nuclear cycle. (c) Progression of xy slices from minute 5, 8, 11 and 14 minutes since 9th nuclear cycle. The green arrows are a cartoon depiction of a displacement vector field.

segmentations of incomplete cells by first inferring displacement vector fields from time-lapse images of membrane data during cellularization in early *Drosophila melanogaster* embryos.

Primordial germ cells (PGCs) are the first to cellularize during Drosophila embryogenesis [**31**, **81**]. Notably, PGCs are initially "incomplete" cells, without fully closed cell membranes, and thus standard cell segmentation and tracking procedures (e.g., [**47**, **80**]) cannot be readily applied (see Figures 6.1 and 6.2). We are interested in (1) quantifying these dynamics and (2) measuring related quantities (such as cell surface areas, volumes, germ plasm concentrations [**77**]) as early in the formation process as possible in order to study variation at the cellular level, as such variation may influence the future function of the cell and its daughters. Previous studies of PGC formation [**19**] show an interesting budding behavior along the embryo's membrane as nuclei reach the embryonic cortex during the end of the 9th nuclear cycle, although this budding is not quantified. Kilwein et al. [**51**] recently proposed several physical surface deformation models for this budding behavior in early stages. One goal of this work is to quantify these surface deformations in several light-sheet microscopy videos by quantities such as cell volume and cell surface area. To accomplish this a convolutional neural network is trained to predict a displacement vector field (DVF) that registers two volumes to one another. The CNN is trained following the approach of Sokooti et al. [**78**], where synthetic DVFs are randomly generated. Several parameters are tuned to control the spatial frequency of these DVFs, and masks localized around the membrane are used to ensure the deformations occur in reasonable locations. To ensure that these random DVFs are sufficiently smooth the coefficients of a B-spline transformation are randomized, rather than the displacement vectors themselves. These DVFs are used as ground truth data in order to train the RBS model. Additionally, a physical surface deformation model proposed in [**51**] was also used to generate more realistic synthetic deformations, which was then used to train a separate model, which we will refer to as the *physical model*.

Overall, two approached are considered for training a registration model: (1) use only random synthetic DVFs [**78**], (2) use only physically informed DVFs. Because the ground truth pairs for the RBS model can be generated freely, whereas the physical model requires simulation data, the RBS model is trained on many more displacements. To control for this, we also include the small RBS (sRBS) model which is trained on the same number of ground truth pairs as the physical simulation model. After training the registration produced by each model is observed qualitatively, and an average mesh distance (AMD) is reported to quantify the success of registration. The AMD, defined in Equation 6.3, measures how close the embryo surface of the registered volume matches that of the fixed volume. The best model is then used to produce a 4D segmentation of PGCs. Finally, these cell segmentations are used to measure cell volume in the early formation of these PGCs.

6.1. Registering 3D Volumes with via Convolutional Neural Network

We begin by describing the architecture and training of a 3D registration model, the creation of synthetic ground truth data, and information regarding the loss function and other evaluation metrics used for determining the model's accuracy on real video data.

6.1.1. Ground Truth Training Data. In order to train a CNN a substantial amount of labeled training data is necessary. In this case, a labeled data point is a triplet $(\mathcal{V}_f, \mathcal{V}_m, F)$ consisting of a fixed volume $\mathcal{V}_f$, a moving volume $\mathcal{V}_m$, and the displacement field F under which Equation 6.1 holds. Figure 6.4 shows examples of the types of synthetic displacements we consider, which are described in more detail below.

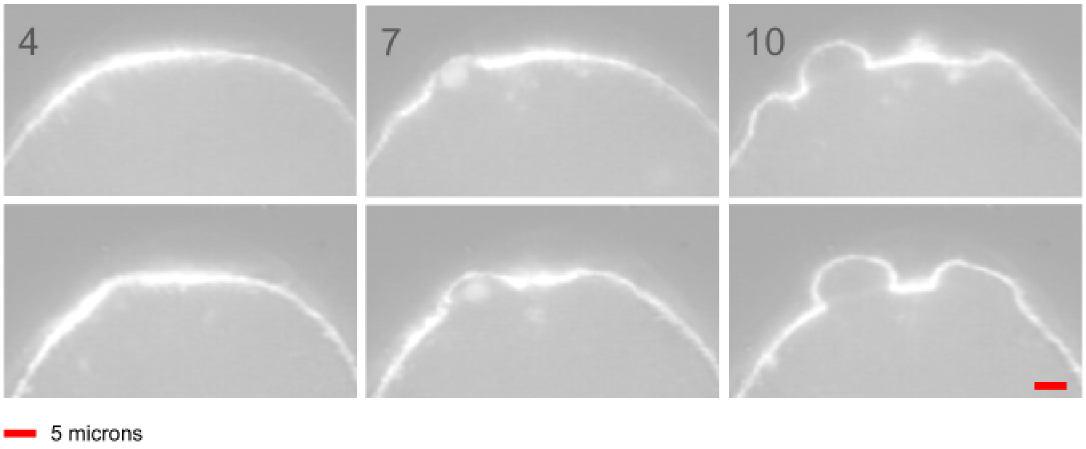


FIGURE 6.2. 2D slice of PGC budding. First and second row: two separate instances of this budding at different points on the same embryo surface.

6.1.1.1. *Synthetic training data.* In order to successfully train the RBS, we generate 4500 synthetic ground truth triplets $(\mathcal{V}_f, \mathcal{V}_m, F)$, generated using code provided by [**78**] and modified for the dataset under consideration. The pipeline is as follows: initial DVF parameters are chosen such as spatial frequency and maximum displacement magnitudes, then random values are assigned to a grid of control points of a specified spacing. If these control points are outside of a specified mask, they are set to zero, otherwise they are subsequently smoothed via a Gaussian kernel, resampled to obtain a DVF, and normalized to be within the maximum magnitude specified along each axis. For further details see [**78**]. This method has the advantage of using very little prior information, and thus allows the network to learn DVFs without substantial bias.

6.1.1.2. *A physical surface mesh model.* Additionally, we utilize a surface mesh in-plane polymerization model developed by Kilwein et al. [**51**] in order to generate physically realistic DVFs which are then used for both a finetuning task, and to train a 3D registration model from scratch. In this case, the displacement data lies along a 2D surface in 3D, and thus we require intermediate

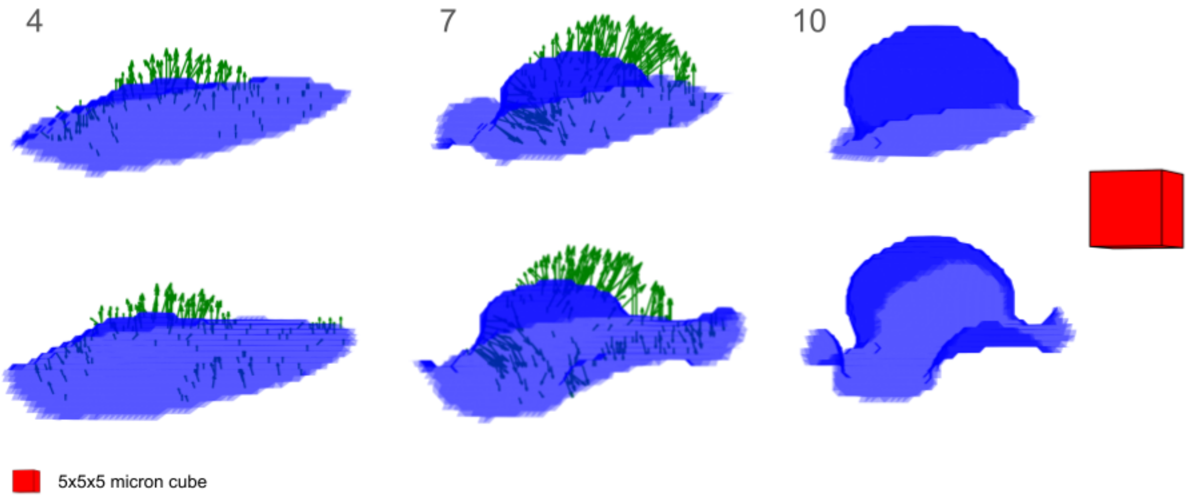


FIGURE 6.3. From left to right: two surface deformations caused by budding PGCs within the first 10 minutes of the 9th nuclear cycle. The green vectors are computed as the shortest vector between the given surface and the surface at the next time point.

steps to extrapolate this to a 3D DVF for our purposes. To this end, we first map our 2D surface coordinate to voxels in 3D, and apply maximum filtering and smoothing before finally applying these DVFs to the physical data obtained via light-sheet microscopy. The results of this approach can be seen in Figure 6.4 row 1.

6.1.2. 3D Registration Model (3DRegNet). Following the work of Sokooti et al. [**78**], we propose using a U-Net architecture [**72**] which takes as input a fixed volume and moving volume, $\mathcal{V}_f, \mathcal{V}_m : I^{m\times n\times \ell} \to \mathbb{R}$, where $I^{m\times n\times \ell}$ is a 3D grid defined on $\{(i,j,k) : 1 \leq i \leq m, 1 \leq j \leq n, 1 \leq k \leq \ell\}$, and outputs a vector field $T : I^{m\times n\times \ell} \to \mathbb{R}^3$. A successful registration would lead to

$$\mathcal{V}_m[\boldsymbol{x}] = \mathcal{V}_f[\boldsymbol{x} + T(\boldsymbol{x})], \tag{6.1}$$

although, as we predict continuous displacement vector fields, in practice this requires reinterpolating our deformed grid back to our fixed lattice grid $I^{m\times n\times \ell}$. We are focused mainly on early stages of the germ cell's development, when its membrane coincides with the outer membrane of the entire embryo. This is roughly within the first 10-12 minutes since the nuclei arrive at the

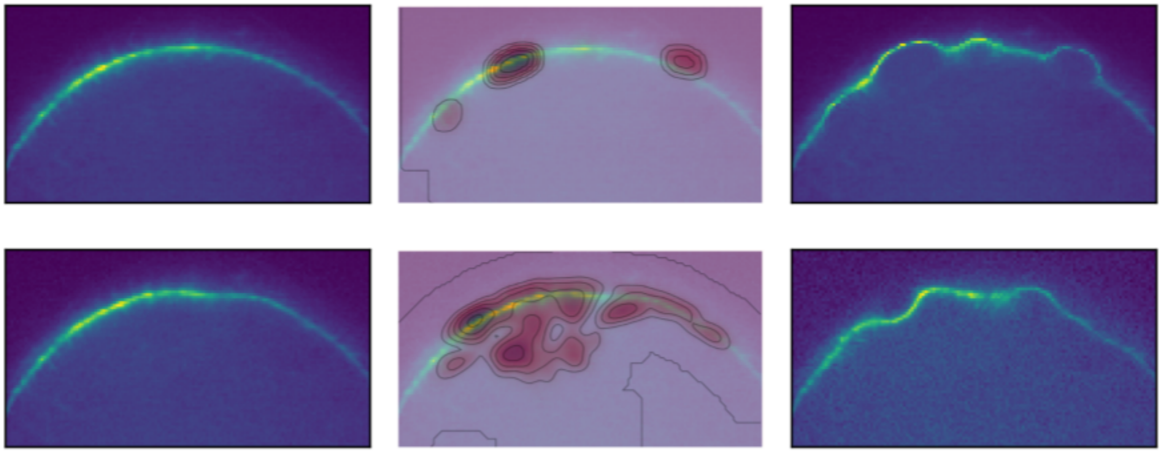


Figure 6.4. Left column: $\mathcal{V}_m$ (slice), middle column: overlayed with $\|F\|$, right column: $\mathcal{V}_f$ (slice). Top row shows physical model, bottom row RBS model.

surface, and in this stage the embryo's membrane is well described by a 2D surface. We anticipate that most useful data for successful registration of any two volumes will derive from these surface deformations, especially considering that the signal from the individual nuclei are inconsistent due to imaging methods. In fact, the nuclei were segmented a priori and then removed from this data. For this reason, we construct masks for the membrane via an intensity thresholding method and use these mask locations to inform the locations where the synthetic DVFs are concentrated. The Adam optimizer is used with a learning rate of $\alpha = 0.0001$ with bending regularization to control the smoothness of the predicted DVF. A target loss is set and a maximum of 500 epochs is run for each model.

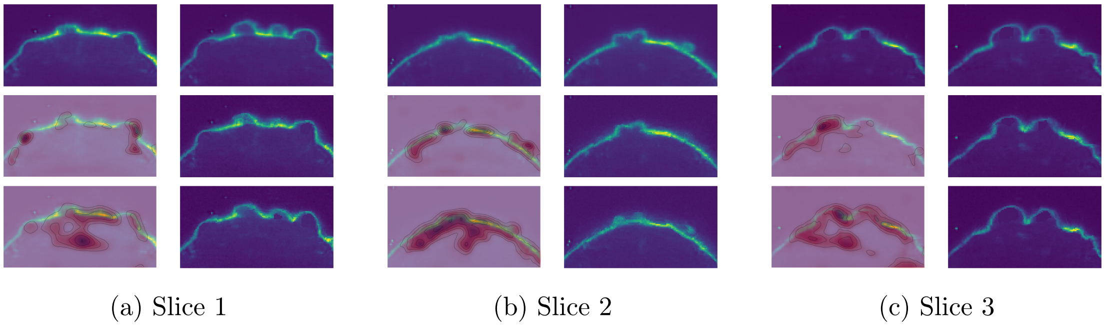
6.1.2.1. *Model architecture.* All experiments with volumetric data downsampled by a factor of 4, training a model on volumes of size $141 \times 77 \times 141$. The initial volumes are cropped to contain the portion of the embryo where PGCs begin their budding process. A standard U-Net architecture is used, based on [**78**] and augmented to fit the size of the volumes considered here. In general, the input consists of two volumes of size $141 \times 77 \times 141$, and the output is a displacement vector field of size $3 \times 141 \times 77 \times 141$.

In our training, we aim to predict the correct vector at each voxel, so $\boldsymbol{e}$ above would represent the difference between the predicted and ground truth displacement vector for a given voxel. Huber loss is used rather than a standard ℓ_2 loss as a way to penalize larger discrepancies more significantly than smaller ones, while remaining convex in $\|\boldsymbol{e}\|$. This is useful in any regression tasks where exact recovery is not expected [**64**]

The Huber loss is given by

$$L_H(\boldsymbol{e}) = \begin{cases} \frac{1}{2}\|\boldsymbol{e}\|^2, & \|\boldsymbol{e}\| \leq 1, \\ \|\boldsymbol{e}\| - \frac{1}{2}, & \text{otherwise.} \end{cases} \tag{6.2}$$

Figure 6.5 shows some examples of the registration between the moving and fixed volumes. We show only slices in the zy plane here. In (a) the RBS model seems to register the volumes better than the physical model. In (b) the opposite is true, and (c) is more ambiguous. The average mesh distance discussed in 6.2 is better suited for judging the overall accuracy of the surface registration, but it is worth pointing out that, although the RBS model generally predicts DVFs along the embryo's surface, there is often non-negligible predictions within the embryo as well. This is in contrast to the physical model's DVF predictions which do not seem to stray from this surface in these examples.


(a) Slice 1

(b) Slice 2

(c) Slice 3

FIGURE 6.5. For each (a), (b), (c), the top left image is a slice of the moving volume, and the top right is the same slice of the fixed volume. The two bottom rows show an overlayed magnitude of the predicted DVFs on the left for the physical model (middle) and the RBS model (bottom), and on the right show the same slice of the moved volumes after applying the predicted DVF.

6.2. Modeling Results

Table 5.1 gives some information about the training of the models discussed here, and also reports the average and standard deviation of AMD values for each time point across three different videos. The RBS model performs best across all three of these videos. The sRBS model is consistently worse than the RBS model, as expected, and outperforms the physical model on Video 2 and Video 3 in terms of average AMD.

Table 6.1. Model Comparison

model name	training size	training loss	testing loss	AMD 1(μ)	AMD 2 (μ)	AMD 3 (μ)
RBS	2400	0.06	0.08	$\mathbf{0.48 \pm 0.09}$	$\mathbf{0.47 \pm 0.26}$	$\mathbf{0.34 \pm 0.07}$
sRBS	540	0.11	0.10	0.64 ± 0.17	0.54 ± 0.26	0.37 ± 0.10
physical	540	0.02	0.03	0.56 ± 0.28	0.74 ± 0.50	0.70 ± 0.65
unregistered	na	na	na	1.11 ± 0.82	1.21 ± 0.89	1.11 ± 0.43

[1] $\mu = 1$ microns

[2] testing and training loss are average Huber loss, L_H, of DVF errors

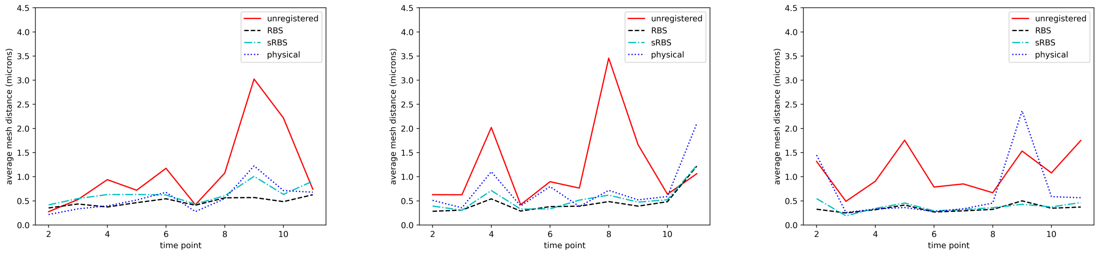
6.2.1. Average Mesh Distance.

In the model training, the error in the predicted DVF is used as the model loss, but when applying the model to the light-sheet microscopy videos capturing the embryo development, the ground truth DVF is unknown. Comparing the intensity values through mean squared error is standard, but there are two issues with this metric here: (1) a mean squared error does not give a clear picture of successful registration of two surfaces, as a small offset of the surface after registration could lead to a large MSE, (2) the intensity profile along the embryo may vary across time, but the registered volume will always contain similar intensity information to the moving time point, not the fixed, so even in the case of a perfect registration, the MSE may still be significant. For these reasons, we define the *average mesh distance* (AMD) as a way to quantify the success of the registration task.

Definition 6.2.1. *Let $S_m = (V_m, F_m, N_m)$ and $S_f = (V_f, F_f, N_f)$ represent discrete meshes, where S_m is the moving mesh and S_f the fixed mesh. Here V_m and V_f represent the vertices of their respective meshes. The average mesh distance is given by the quantity*

$$AMD(V_f, V_m) = \frac{1}{|V_f|} \sum_{v_f \in V_f} \min_{v_m \in V_m} \|v_f - v_m\|_2. \tag{6.3}$$

During nuclear cycles 9 and 10 the embryo of posterior pole being studied in this chapter can be represented as a 2D surface, as observed in the slices shown in Figure 6.1. The DVF predicted



(a) Video 1 (b) Video 2 (c) Video 3

Figure 6.6. Average mesh distance between successive timepoints from 1 to 11 from three different videos. Each compares the results of the RBS, sRBS, and physical registration models.

between two time points then aims to register these surfaces. So, by approximating this surface at each time point as a discrete surface mesh, the two meshes can then be compared via the AMD between them. AMD is particularly well suited to surfaces undergoing smooth local displacements, as we have here, and can provide better insight into the success of the surface registration than MSE. However, one draw back is that AMD requires extracting an accurate surface mesh from the volumetric data, which becomes an issue for the data here at later time points when the spatial frequency of the mesh increases and adjacent PGCs bump up against one another, so approximating the embryo as a smooth surface is no longer possible. By this stage, however, more standard cell segmentation methods function better, but this is outside the scope of this project.

6.3. Tracking Surface Budding Deformations Across Time

6.3.1. Discrete Differential Geometry Prerequisites. The segmentation method used in this chapter relies on a discrete mean curvature estimate which is computed on the 2D surface extracted from each time point. For a introductory resource into discrete differential geometry which we rely on here, see [**25**]. In the continuous setting a 2D surface can be defined as a map $\boldsymbol{f} : M \subset \mathbb{R}^2 \to \mathbb{R}^3$ with *differential* $\mathrm{d}S$ which deforms vectors in $U \subset \mathbb{R}^2$ to vectors along the surface $\boldsymbol{f}(M) \subset \mathbb{R}^3$. The metric g induced by the surface f is defined as

$$g(X, Y) = \mathrm{d}\boldsymbol{f}(X) \cdot \mathrm{d}\boldsymbol{f}(Y), \tag{6.4}$$

where X and Y are any two *tangent vectors* take from the *tangent plane* at the point $p \in M$. A vector u is normal to the surface at the point p if $\mathrm{d}\boldsymbol{f}(X) \cdot u = 0$ for any tangent vector X at a point

p. A *unit normal* N is a normal vector with length one, of which there are only two at each point p, namely N and $-N$. In this chapter the surfaces of interest are all orientable, so it is possible to choose a consistent orientation so that the normal vector defined a continuous map $N : M \to S^2$, referred to as the *Gauss map*, where $S^2 \subset \mathbb{R}^3$ denote the unit sphere. In fact, how the unit normal changes along a path on the surface is precisely the idea of local curvature that is important for capturing the surface deformations of interest here.

6.3.1.1. *Principal curvature and mean curvature.* Given a surface $\boldsymbol{f} : M \to \mathbb{R}^3$, the Gauss map $N : M \ \to S^2$ gives the unit normal to $\boldsymbol{f}$ at every point $p \in M$. The Gauss map can itself be thought of as a surface, and the change in the normal along a particular tangent vector X at the point p is given by $\mathrm{d}N(X)$. The *normal curvature* along this same tangent vector is given by:

$$\kappa_n(X) := \frac{\mathrm{d}\boldsymbol{f}(X) \cdot \mathrm{d}N(X)}{|\mathrm{d}\boldsymbol{f}(X)|^2}, \tag{6.5}$$

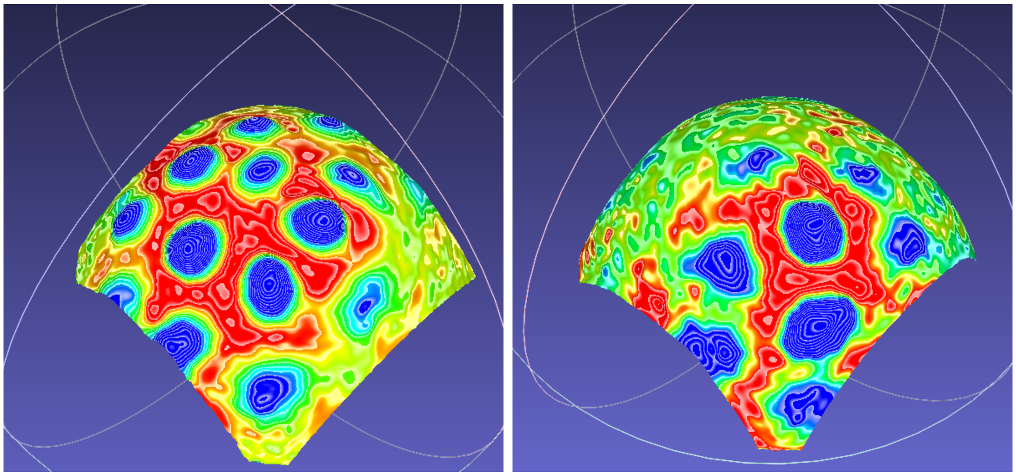
and can be thought of as the change of the unit normal along the direction $\mathrm{d}\boldsymbol{f}(X)$, where the denominator is a normalizing factor for how X is stretched by $\boldsymbol{f}$. The *principal curvatures* are the minimum and maximum values of κ, denoted by κ_1 and κ_2 which occur along the principal direction X_1 and X_2 respectively. Further, whenever $\kappa_1 \neq \kappa_2$, the principal directions are always orthogonal with respect to the induced metric g. The *mean curvature* H is given by the average of the principal curvatures:

$$H = \frac{\kappa_1 + \kappa_2}{2}, \tag{6.6}$$

and the *Gaussian curvature* K is given by

$$K = \kappa_1 \kappa_2. \tag{6.7}$$

Roughly speaking, the mean curvature measures whether there is curvature along at least one direction, whereas the Gaussian curvature measures if there is curvature along both directions; but curvature can be negative, and if $\kappa_1 = -\kappa_2$ then the mean curvature will be zero, causing an issue with this interpretation. The sign of the Gaussian curvature can detect whether a saddle structure or a quadratic structure is present, similar to a second order local extremal test in a standard calculus text.



(a) example 1

(b) example 2

Figure 6.7. Mean curvature of two embryo surfaces at early budding stages; the mean curvature data localizes the PGC budding behavior, allowing for accurate surface segmentation.

For the purpose of segmenting discrete surfaces based on mean curvature, the `scale-dependent quadratic fitting` method in `MeshLab` (`PyMeshLab`) [**17**,**18**] was utilized, which finds a quadratic surface of best fit around a given vertex and a subset of its neighbor, and assigns the mean curvature of this quadratic surface to be the vertex under consideration. This method takes a scale as an input, which controls the size of the neighborhood used around each point. The larger the neighborhood used, the smoother the discrete curvature will be. Figure 6.8 shows some examples of extracted surfaces colored by their mean curvature approximated in this way.

6.3.2. Tracking Primordial Germ Cells via surface geometry and registration. In most cell segmentation tasks, cells to be segmented are 'complete' in the sense that they have a complete, or mostly complete, cell membrane. Low-level methods rely on locating nuclei and performing watershed segmentation with seeds related to this nuclei location. This is also the basis for state of the art data-drive methods such as CellPose. In the case of primordial germ cells (PGCs), complete cell membranes are not present at early stages. Therefore new method are required that do not depend on a complete cell membrane.

To this end, several observations are made: (1) during early stages of development, the embryo can be modeled as a 2D surface, (2) by tracking this surface through time, it is possible to locate and track portions of the surface undergoing the budding process that characterizes PGCs.

6.3.2.1. *Notation.* Let K be the number of PGCs to be tracked, and $\mathcal{T}$ be the total number of time steps in consideration. Let $\{S_t = (V_t, F_t, N_t)\}_{t=1}^{\mathcal{T}}$ be the surface mesh for each time $t = 1, \ldots \mathcal{T}$, which is a triple containing the vertices V_t, faces F_t, and normal vectors N_t of the mesh respectively. Each PGC will correspond to a cluster of vertices. The cluster corresponding to cell k and time t is denoted by C_t^k, with corresponding center $c_t^k \in C_t^k$. Let T_t denote a displacement vector field acting on V_t, and $\mathcal{P}_t$ denote the projection of any point $v \in \mathbb{R}^3$ to the nearest vertex in V_t, that is, $\mathcal{P}(v) = \arg\min_{\tilde{v} \in V_t} \|v - \tilde{v}\|_2^2$. These projection operators are necessary because the predicted displacement vector fields will not perfectly register two successive surfaces. For example, for a given $v \in V_t$, $T_t(v)$ is not guaranteed to be a vertex in V_{t+1}, but $P_t(T_t(v))$ will be. The function d_t denotes the geodesic distance between two points on S_t, and the geodesic ball of radius $\epsilon > 0$ about a vertex $v \in V_t$ is denoted as $B_t(v, \epsilon)$, and defined as $B_t(v, \epsilon) = \{v' \in V_t \mid d_t(v, v') < \epsilon\}$. $H_t : V_t \to \mathbb{R}$ will denote the mean curvature of the surface at time point t, which is discussed more in the following section.

6.3.2.2. *Surface geometry.* As the nuclei reach the posterior pole, buds begin to form along the embryo membrane. In order to locate these buds, we propose segmenting the embryo membrane via a marching cube algorithm, and then performing surface smoothing to produce a surface smooth enough to provide meaningful mean curvature data. Generally, one should expect lower mean curvature along the surface where no bud is present, higher mean curvature along the buds, and negative mean curvature along the transition between these two regions. See Figure 6.8 for reference.

6.3.2.3. *Initial budding locations and surface segmentation.* Each portion of the membrane corresponding to a bud should be segmented, which can be done by using local curvature data. Initial bud locations are provided directly via the user, though an unsupervised clustering algorithm based on mean curvature data over the surface could also be used to determine initial bud locations. The surface is then automatically segmented into individual PGCs following Algorithm 1.

6.3.2.4. *Tracking.* With the initial surface segmentations just described, these segmented voxels are tracked through time by making use of 3DRegNet. The tracking procedure is simple to describe qualitatively: For each initial center c_t^i where the subscript $0 \leq t \leq \mathcal{T}$ indicates the time step, and the superscript indicates the cell index, a small geodesic ball around it is found, denoted by $B_{g_t}(c_t^i, \epsilon_2)$. Here ϵ_2 denotes a different value than is used for segmentation. This ball is then acted on by the predicted displacement vector field $F_{t \to t+1}$. Because the predicted DVF will not perfectly register

Algorithm 1 PGC segmentation

```
Require: Surface mesh $S = (V, F, N)$, cell centers $\{c^i\}_{i=1}^K \subset V$, $\epsilon_1 > 0$, geodesic distance $d_g : V \times V \to \mathbb{R}$, mean curvature $K_M : V \to \mathbb{R}$
  for $i = 1, \dots K$ do
    $V^i \leftarrow V(c^i, \epsilon_1) = \{v \in V \mid d_g(c^i, v) < \epsilon_1\}$
    $K^i \leftarrow K_M|_{V^i}$
    $thresh = \texttt{get\_thresh}\ (K^i)$
    $\tilde{C}^i = \{v \in V^i \mid K^i[v] > thresh\}$
    $C^i = \texttt{find\_largest\_cluster}(\tilde{C}^i)$
  return $\{C^i\}_{i=1}^K$
```

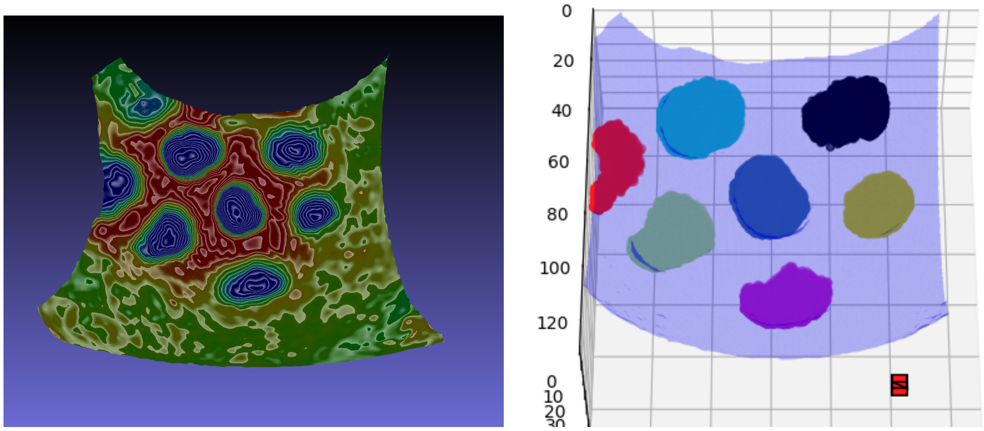
the two surfaces, the operators $\{P_j\}_{j=1}^{\mathcal{T}}$ are defined, which project any $v \in \mathbb{R}^3$ to the nearest vertex in V_j, the vertices of the mesh from time point j; concretely, $P_j(v) = \arg\min_{\tilde{v}\in V_j} \|v - v_j\|_2^2 \in V_j$. The result of this is a new set of vertices $P_j\left(F_{t\to t+1} B_{g_t}\right) \subset V_{t+1}$. The euclidean average of this set is computed and this average is again projected onto V_{t+1}, which finally yields the new cell center c_t^i. This process is described in Algorithm 2. Given a new set of cell centers, Algorithm 1 is then applied to segment the cells at time $t+1$.

Algorithm 2 PGC Tracking

```
Require: Surface meshes $S_{t_i}$, $S_{t_{i+1}}$, cell centers $\{c_t^i\}_{i=1}^K \subset V$, $\epsilon_2 > 0$, geodesic distance $d_{g_t}$, and DVF $F_{t\to t+1}$
  for $i = 1, \dots K$ do
    $\tilde{C}_{t+1}^i \leftarrow F_{t\to t+1} B_{g_t}(c_t^i, \epsilon_2)$
    $c_{t+1}^i = P_{t+1}\left(\frac{1}{|\tilde{C}_{t+1}^i|} \sum_{v\in\tilde{C}_{t+1}^i} v\right)$
  return $\{c_{t+1}^i\}_{i=1}^K$
```

6.4. Extracting physical data from 4D segmentations

After PGC segmentation and tracking, each cell has a corresponding cluster of vertices at each time point, which will be denoted by C_t^i for cell i at timepoint t. The collection of these is the tuple $(C_1^i, C_2^i, \dots, C_{\mathcal{T}}^i)$. An example of such a surface segmentation for Video 2 at timepoint 7 is shown alongside the mean curvature data in Figure 6.8. Then, for each $t = 1, \dots, \mathcal{T}$ the convex hull of the vertices is computed which gives an estimate of the cell volume over time, which we write as the vector $\boldsymbol{a}^i \in \mathbb{R}^{\mathcal{T}}$. An example of these convex hulls and the resulting volume estimates for over time for Video 3 is given in Figure 6.9(a)(b). Figure 6.9(c)(d) shows the volumes tracked for Video 1 and Video 2. We currently track the volume until the next nuclear cycle begins, when many of the cells



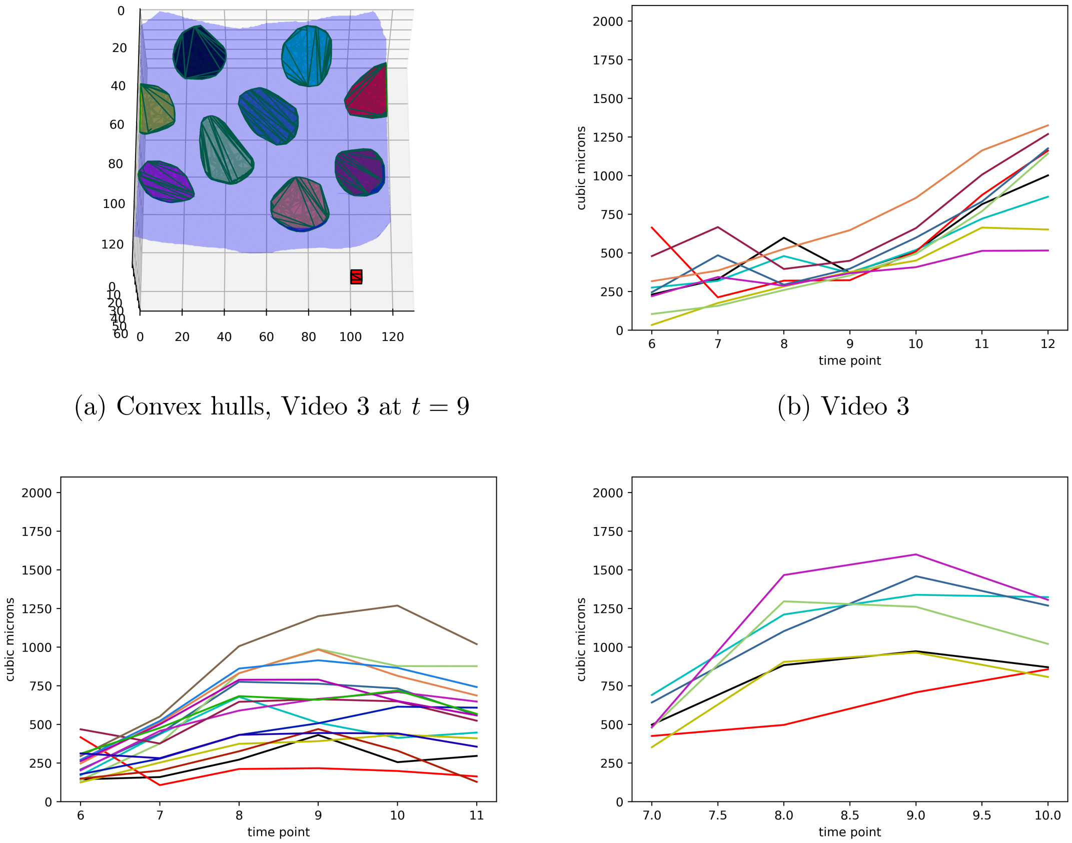
(a) mean curvature

(b) segmentation

FIGURE 6.8. Mean curvature of two embryo surfaces at early budding stages; the mean curvature data localizes the PGC budding behavior, allowing for accurate surface segmentation.

multiply and it becomes necessary to track the lineages. Future work will extend these algorithms to track the surfaces and subsequent cell volumes across these lineages as well.

6.4.1. Discussion. This work is the first attempt at quantitatively measuring cell volumes of PGCs in these early stages of development. By utilizing a CNN to predict displacement vector fields, we are able to capture the dynamics of these cells by predicting displacement vector fields which register the volumes between successive timepoints. This successful registration is leveraged to track them through time. Mean curvature data from the extracted surfaces is used to cluster the surface protrusions which characterize these PGCs. We are currently working on unsupervised clustering techniques to locate these protrusions on the surface of each time point, which will aid in further automating this tracking process. We are also working to extend Algorithm 1 and Algorithm 2 to track cells across multiple nuclear cycles.



(a) Convex hulls, Video 3 at $t = 9$

(b) Video 3

(c) Video 1

(d) Video 2

FIGURE 6.9. (a) Alpha shapes of segmented cells at minute 6 of Video 3 (b) Volumes of PGCs segmented from Video 3 from timepoint 6 to timepoint 12. (c) Cell volume estimates Video 1 (d) Video 2. x-axis is in minutes after nuclei arrive at the surface of the embryo.

CHAPTER 7

Conclusion

This dissertation developed new methods and tools to extract import features from a wide variety of image data, solving problems in spatial phase estimation, explainable texture segmentation, 3D image registration and 3D+t image segmentation. Each chapter is at the intersection of method development and real-world application, and much of the code which has been written for these projects is accessible to the interested reader to be able to join in on similar research efforts to develop robust feature extraction methods. The code for the spatial phase estimation presented in Chapter 3 and the scripts to reproduce the experiments shown in that chapter can be found at https://gitlab.com/briancknight/SSVM2025. The HandsFreeFishing package can be found at https://github.com/briancknight/HandsFreeFishing, where installation steps are outlined. The main contributions of this dissertation are summarized below, and for each we mention possible future directions.

In Chapter 3 a new spatial phase estimate is constructed based on a multi-scale local phase quality map, which used the features of the structure multivector. This estimate achieved robust single-shot spatial phase estimation in fringe pattern images such as fingerprint data or fringe projection profilometry data, and was shown to outperform existing methods on several synthetic experiments. It was used to perform fine-scale fingerprint registration on real-life fingerprint data, though future work is required to benchmark these results against state-of-the-art finescale registration method. Lastly a low rank version of the estimate was also introduced, and several of the previous experiments were repeated. These multiscale feature estimates could act as high quality inputs for deep learning models, which is an exciting direction for future work here.

Chapter 4 extended the results of Saito [**74**] in developing explainable and class revealing features in image classification problems. This was accomplished by solving an optimization problem to find inputs to a classifier which maximize the probability of belonging to a particular class. In order to extract interpretable feature vectors, the wavelet scattering transform was first applied to the images to be classified, and a multiclass logistic regression classifier was then trained on these features. Due to the nature of the scattering transform coefficients standard first and second order optimization schemes were ill fit, and so zeroth-order optimization was used in this case and found to be successful when the appropriate regularization terms were applied. With this setup, an explainable texture segmentation method is developed based on classification of local patches. This approach is used on the Brodatz texture database, with some preliminary results using the MNIST and Breast MNIST datasets. Future work will consider how the monogenic wavelet scattering network [**15**] compares to the standard scattering transform network discussed here and how regularization parameters might be chosen more systematically.

In Chapter 5, the HandsFreeFishing program is introduced. This program is built to extract important morphometric features from images of juvenile Chinook salmon. It is based off of automated segmentations computed by an open-source deep learning segmentation model. From this segmentation, it is possible to measure morphometric quantities such as fork length, eye-diameter, and fish height. Then, with a small amount of ground truth weight data, these features were used to train a weight prediction model. When modeling the 3D fish body as an ellipsoid, the morphometric features measured by the HandsFreeFishing program were used to estimate the volume of the fish body, and a standard 1D regression model was then able to predict fish weight with high accuracy. Compared to previous models, the one introduced in this chapter is very simple; this fact exemplified the importance of extracting relevant and robust features from data. As far as we know, all of the methods and algorithms developed to measure such quantities are new, and the hope is that they will aid researchers as they actively collect more images of juvenile Chinook and other fish species. Moving forward with this project, we hope to extend the capabilities to different species of fish, and gather more ground truth data to be able to train alternate weight prediction models and compare their performances.

Finally, Chapter 6 presents a method to register timepoints of 3D videos of fruit fly (*Drosophila melanogaster*) embryo development. This is accomplished by using a convolutional neural network, where simulated deformations are used as ground truth training data. The data used was light-sheet microscopy data of fruit fly embryos developing at early stages, during the 9th and 10th nuclear cycle, when nuclei first begin to reach the surface of the embryo. In these early stages the surface of the embryo can be modeled as a 2D surface, and thus this network was used to study the dynamics of a surface captured in a 3D+t dataset. Because certain deformation patterns along the surface are of interest (they correspond to early formation of cells in fruit flies), the predicted deformations from the CNN are used along with local mean curvature data of the extracted 2D surface to locate and segment regions undergoing these patterns. These features are then studied in the context of the physical and biological problem at hand, and used to measure cell volumes at earlier stages than previous methods are able to achieve. Currently we are working to extend the algorithms presented for segmentation and tracking across nuclear cycles.

APPENDIX A

Orientation Estimation with the Structure Mulitvector

A.1. Construction of the SMV

Two-dimensional signals can, of course, vary in two or more orientations in any given local patch, and further work has been done using hypercomplex signal processing that can deal with these cases. One extension, the *structure multivector* (SMV), was introduced along with the monogenic signal in the PhD dissertation of Felsberg [**33**]. It is designed to deal with signals of the form

$$f(\boldsymbol{x}) = f_1(\boldsymbol{n}(\boldsymbol{x}) \cdot \boldsymbol{x}) + f_2(\boldsymbol{n}(\boldsymbol{x})^{\perp} \cdot \boldsymbol{x}), \tag{A.1}$$

These are intrinsically 2D (i2D) signals with two orientations in each local patch that are orthogonal to one another.

The features of the SMV will essentially be that of two monogenic signals, and to accommodate these additional features the SMV lives in a larger dimensional Clifford algebra, $C\ell_{3,0}$ which subsumes the quaternions. For more information on Clifford algebras see [**29**], [**33**]. In general we denote the product $e_{i_1}e_{i_2}\cdots e_{i_n} := e_{i_1 i_2 \cdots i_n}$.

We consider an image of the form $\mathbf{f} : \mathbb{R}^2 \to e_3\mathbb{R}$,

$$\mathbf{f}(xe_1 + ye_2) = f(x, y)e_3.$$

Note $f = \mathbf{f}e_3 = e_3\mathbf{f}$.

The corresponding structure multivector (SMV) is given by

$$\begin{aligned} M_S(\boldsymbol{x}) &= \left[\mathbf{f}(\boldsymbol{x}) + (h_2^1 * \mathbf{f})(\boldsymbol{x})\right] + e_3\left[(h_2^2 * \mathbf{f})(\boldsymbol{x}) + (h_2^3 * \mathbf{f})(\boldsymbol{x})\right] \\ &= M_0 + M_1e_1 + M_2e_2 + M_3e_3 + M_{23}e_{23} + M_{31}e_{31} + M_{12}e_{12} \end{aligned}$$

The explicit definitions of these functions are given below, where $\boldsymbol{x} = xe_1 + ye_2$:

$$M_1 = \frac{x}{2\pi|\boldsymbol{x}|^3} * f(\boldsymbol{x}), \qquad M_2 = \frac{y}{2\pi|\boldsymbol{x}|^3} * f(\boldsymbol{x}), \qquad M_3 = f(\boldsymbol{x}),$$
$$M_{23} = \frac{3(3x^2y - y^3)}{2\pi|\boldsymbol{x}|^5} * f(\boldsymbol{x}), \qquad M_{31} = \frac{3(3xy^2 - x^3)}{2\pi|\boldsymbol{x}|^5} * f(\boldsymbol{x}),$$
$$M_0 = \frac{-2(x^2 - y^2)}{2\pi|\boldsymbol{x}|^4} * f(\boldsymbol{x}), \qquad M_{12} = \frac{-4xy}{2\pi|\boldsymbol{x}|^4} * f(\boldsymbol{x}).$$

The Fourier transforms of the operators h_2^k are given below:

$$\begin{aligned} H_2^1(\boldsymbol{u}) &= \frac{\boldsymbol{u}}{|\boldsymbol{u}|} I_2^{-1} = \frac{-ue_2 + ve_1}{|\boldsymbol{u}|} \\ H_2^2(\boldsymbol{u}) &= \frac{e_1\boldsymbol{u}e_1\boldsymbol{u}}{\boldsymbol{u}^2} = \frac{(u^2 - v^2) + 2uve_{12}}{\boldsymbol{u}^2} \\ H_2^3(\boldsymbol{u}) &= \frac{\boldsymbol{u}e_1\boldsymbol{u}e_1\boldsymbol{u}}{|\boldsymbol{u}|^3} I_2^{-1} = \frac{(3uv^2 - u^3)e_{23} - (3u^2v - v^3)e_{31}}{|\boldsymbol{u}|^3}, \end{aligned}$$

where $I_2 = e_{12}$ and $I_2^{-1} = e_{21}$ as $e_{12}e_{21} = 1$ by definition. I_2 acts an imaginary unit here, as $I_2^2 = -1$. Note h_2^1 is simply the Riesz transform, and h_2^3 is the composition of h_2^2 and the Riesz transform, so the crux of this extension is in understanding H_2^2. First, it responds only to even signals. Second, any two perpendicular vectors $\boldsymbol{n}$ and $\boldsymbol{n}^\perp$ are antiparallel after action by H_2^2, which means that an even signal according to the (A.1) will yield a response to H_2^2 whose argument is precisely twice that of the main orientation of $\boldsymbol{n}$.

Specifically, we can calculate the orientation $\boldsymbol{n}$ given a signal of the form $f(\boldsymbol{x}) = A\cos(\boldsymbol{n} \cdot \boldsymbol{x}) + B\cos\big(\boldsymbol{n}^\perp \cdot \boldsymbol{x}\big)$ directly from this response. To handle odd structures, we finally take the Riesz transform of h_2^2 to yield h_3^2. The product of the Riesz response and the response of the third order harmonic estimates this same orientation, but is better suited for odd structures, hence the average of these two arguments provides a robust orientation estimate of the structure multivector, as given in Felsberg's dissertation [**33**]:

$$\theta_e = \frac{1}{4} \arg\left[(M_0 + M_{12}I_2)^2 + (M_1 + M_2I_2)(M_{31} - M_{23}I_2)\right]. \tag{A.2}$$

In [**33**] he shows that the extended signal model provides a more robust orientation estimator than that of the monogenic signal. This is further confirmed in [**62**]. In addition to these facts, we show that: 1) the feature set of the SMV is robust even to i2D signals which violate the orthogonality constraint; and 2) if one of the local i1D signal dominates the local energy, then we can estimate the corresponding orientation well even in the case of large deviation from this constraint. See Appendix A.2 for details.

With this orientation estimate it is then possible to construct a pair of angular filters that decompose a signal f into two i1D signals, which will then yield two local amplitudes, two local orientations, and two local phases that can be used for further processing.

Explicitly, these filters are given by:

$$\begin{aligned} \mathcal{W}_1 f &= \frac{M_3 + \cos(2\theta_e)M_0 + \sin(2\theta_e)M_{12}}{2} \\ \mathcal{W}_2 f &= \frac{M_3 - \cos(2\theta_e)M_0 - \sin(2\theta_e)M_{12}}{2} \end{aligned}$$

and their Riesz transforms, projected onto $-e_2$:

$$\begin{aligned} \mathcal{W}_3 f &= \frac{3(\cos(\theta_e)M_1 + \sin(\theta_2)M_2) + \cos(3\theta_e)M_{31} - \sin(3\theta_e)M_{23}}{4} I_2 \\ \mathcal{W}_4 f &= \frac{3(-\sin(\theta_e)M_1 + \cos(\theta_2)M_2) + \sin(3\theta_e)M_{31} + \cos(3\theta_e)M_{23}}{4} I_2. \end{aligned}$$

Then our two complex i1D signals are given by

$$F_1(\boldsymbol{x}) = \mathcal{W}_1 f + \mathcal{W}_3 f, \quad F_2(\boldsymbol{x}) = \mathcal{W}_2 f + \mathcal{W}_4 f$$

Recall our orientation θ_e is a function of the spatial variable $\boldsymbol{x}$ and is implicit in the above signals; see Eq. (3.4) (or Eq. (A.2)).

The full feature set of the SMV then is given by this local orientation estimate and:

$$A_i(\boldsymbol{x}) = |F_i(\boldsymbol{x})|, \quad \phi_i(\boldsymbol{x}) = \arg|F_i(\boldsymbol{x})|,$$

for $i = 1, 2$. At each location $\boldsymbol{x}$, we choose the main signal by selecting the pair with the largest local amplitude. This selection is given by the dominance index $d(\boldsymbol{x}) = \arg\max_{1,2}\{A_1(\boldsymbol{x}), A_2(\boldsymbol{x})\}$, so that we have a major and minor IAP representation given by:

$$\begin{aligned} A(\boldsymbol{x}) &= A_{d(\boldsymbol{x})}, & \Phi(\boldsymbol{x}) &= \phi_{d(\boldsymbol{x})}, \\ a(\boldsymbol{x}) &= A_{3-d(\boldsymbol{x})}, & \phi(\boldsymbol{x}) &= \phi_{3-d(\boldsymbol{x})}. \end{aligned}$$

Here the capital A and Φ denote the dominant, or major, local i1D signal.

A.2. Orientation Estimation with the SMV

Our signal model (3.2) (also (A.1)) assumes the sum of two orthogonal sinusoidal modes, and we would like to know how this orientation estimate behaves if they are not perfectly orthogonal. We will restrict ourselves to the case where they have equal frequency so that they are inseparable even in the multiscale phase model.

Suppose $\boldsymbol{n} = \cos(\theta)e_1 + \sin(\theta)e_2$, $\boldsymbol{n}_\epsilon^\perp = \sin(\theta+\epsilon)e_1 - \cos(\theta+\epsilon)e_2$, and

$$f(\boldsymbol{x}) = A\cos(\boldsymbol{n}\cdot\boldsymbol{x}) + B\cos\Big(\boldsymbol{n}_\epsilon^\perp\Big).$$

We have the following:

$$\begin{aligned}
M_1 + M_2 I_2 &= A(e_1\boldsymbol{n})\sin(\boldsymbol{n}\cdot\boldsymbol{x}) + Be_1\boldsymbol{n}_\epsilon^\perp \sin\Big(\boldsymbol{n}_\epsilon^\perp\cdot\boldsymbol{x}\Big)\\
M_0 + M_{12} I_2 &= A(e_1\boldsymbol{n})^2\cos(\boldsymbol{n}\cdot\boldsymbol{x}) + B(e_1\boldsymbol{n}_\epsilon^\perp)^2 \cos\Big(\boldsymbol{n}_\epsilon^\perp\cdot\boldsymbol{x}\Big)\\
M_{31} - M_{23} I_2 &= A(e_1\boldsymbol{n})^3\sin(\boldsymbol{n}\cdot\boldsymbol{x}) + B(e_1\boldsymbol{n}_\epsilon^\perp)^3 \sin\Big(\boldsymbol{n}_\epsilon^\perp\cdot\boldsymbol{x}\Big),
\end{aligned}$$

where

$$\begin{aligned}
e_1\boldsymbol{n}_\epsilon^\perp &= -I_2\exp(\theta I_2)\exp(\epsilon I_2)\\
&= -I_2\exp(\epsilon I_2)(e_1\boldsymbol{n})\\
(e_1\boldsymbol{n}_\epsilon^\perp)^2 &= -\exp(2\theta I_2)\exp(2\epsilon I_2)\\
&= -\exp(2\epsilon I_2)(e_1\boldsymbol{n})^2\\
(e_1\boldsymbol{n}_\epsilon^\perp)^3 &= I_2\exp(3\theta I_2)\exp(3\epsilon I_2)\\
&= I_2\exp(3\epsilon I_2)(e_1\boldsymbol{n})^3.
\end{aligned}$$

Analyzing the even response first, we have

$$\begin{aligned}
(M_0 + M_{12}I_2)^2 = (e_1\boldsymbol{n})^4 &\\
&[A^2\cos^2(\boldsymbol{n}\cdot\boldsymbol{x}) + B^2\exp(4\epsilon I_2)\cos^2(\boldsymbol{n}_\epsilon^\perp\cdot\boldsymbol{x})\\
&+ 2AB\exp(2\epsilon I_2)\cos(\boldsymbol{n}\cdot\boldsymbol{x})\cos\Big(\boldsymbol{n}_\epsilon^\perp\cdot\boldsymbol{x}\Big)].
\end{aligned}$$

For the product of odd responses we have:

$$\begin{aligned}(M_1 + M_2I_2)(M_{31} - M_{23}I_2) = (e_1\boldsymbol{n})^4[&A^2 \sin^2(\boldsymbol{n}\cdot\boldsymbol{x}) \\ &+ B^2 \exp(4\epsilon I_2)\sin^2(\boldsymbol{n}_\epsilon^\perp \cdot \boldsymbol{x}) + AB(\exp(3\epsilon I_2) \\ &+ \exp(\epsilon I_2))\sin(\boldsymbol{n}\cdot\boldsymbol{x})\sin\Big(\boldsymbol{n}_\epsilon^\perp \cdot \boldsymbol{x}\Big)].\end{aligned}$$

In general for any real number a we have: $\exp(ai) + \exp(3ai) = 2\exp(2ai)\cos(a)$, our odd response becomes

$$\begin{aligned}(M_1 + M_2I_2)(M_{31} - M_{23}I_2) = (e_1\boldsymbol{n})^4[&A^2 + B^2 \exp(4\epsilon I_2) \\ &+ 2AB\exp(2\epsilon I_2)[\cos(\boldsymbol{n}\cdot\boldsymbol{x})\cos\Big(\boldsymbol{n}_\epsilon^\perp \cdot \boldsymbol{x}\Big)+ \\ &\cos(\epsilon)\sin(\boldsymbol{n}\cdot\boldsymbol{x})\sin\Big(\boldsymbol{n}_\epsilon^\perp \cdot \boldsymbol{x}\Big)]].\end{aligned}$$

Therefore, letting $M = \cos(\boldsymbol{n}\cdot\boldsymbol{x})\cos\Big(\boldsymbol{n}_\epsilon^\perp \cdot \boldsymbol{x}\Big)+\cos(\epsilon)\sin(\boldsymbol{n}\cdot\boldsymbol{x})\sin\Big(\boldsymbol{n}_\epsilon^\perp \cdot \boldsymbol{x}\Big)$ for brevity, our estimate is give by:

$$\begin{aligned}\Theta(\epsilon) &= \theta + \frac{1}{4}\arg[A^2 + B^2\exp(4\epsilon I_2) + 2AB\exp(2\epsilon I_2)\cdot M] \\ &= \theta + \frac{1}{4}\tan^{-1}\left[\frac{B^2\sin(4\epsilon) + 2AB\sin(2\epsilon)\cdot M}{A^2 + B^2\cos(4\epsilon) + 2AB\cos(2\epsilon)\cdot M}\right].\end{aligned}$$

We see that as $\frac{B}{A} \to 0$ it follows that $\Theta(\epsilon) \to \theta$.

Additionally, in the case $A = B$, the argument in the artan function becomes

$$\begin{aligned}\frac{\sin(4\epsilon) + 2\sin(2\epsilon)\cdot M}{1 + \cos(4\epsilon) + 2\cos(2\epsilon)\cdot M} &= \frac{2\sin(2\epsilon)[\cos(2\epsilon) + M]}{2\cos(2\epsilon)[\cos(2\epsilon) + M]} \\ &= \tan(2\epsilon).\end{aligned}$$

Thus

$$\Theta(\epsilon) = \theta + \frac{\epsilon}{2}.$$

for $|\epsilon| < \pi/4$.

APPENDIX B

Fin and Eye Segmentation Heuristics

This section provides additional details in regarding Chatper 5. Here, the heuristics used to segment individual fins and the eyball of the fish are discussed. Once the initial fish contour is extracted and rotated, the left most point of the contour lies close to the snout of the fish, and the right most point lies somewhere along the caudal fin. Let $c = [c_0, c_1, \dots c_n]$ where $c_i = (x_i, y_i)$ for $0 \leq i \leq n$ represent this rotated contour. The fork length is then computed as outlined in 5.1.4. For the following heuristics, the quantity $x_{max} = \max_{0 \leq j \leq n} x_j$ will be used frequently. This represents the largest x value and thus the right most point of the contour. The average x value $\overline{x} = \frac{1}{n} \sum_{0 \leq j \leq n} x_j$ will also be utilized. Note that these countour points are extracted from the fish segmentation, and thus small y values are higher in the image, whereas large y-values are lower in the image. Small x values correspond to the left side of the image whereas large x values correspond to the right side of the image.

(1) (*Dorsal Fin*) To estimate a bounding box for the dorsal fin, all points of the contour c for which $0.5x_{max} \leq x_i \leq 0.6x_{max}$ are found. From these points, the largest y-value is computed, and this is taken to approximate the top most point of the dorsal fin. After this point is located, a bounding box with a width of $\frac{1}{7}$ the fork length and height of $\frac{2}{21}$ the fork length is centered about this point and given to the segmentation model, which produced the dorsal fin segmentation.

(2) (*Adipose Fin*) To estimate a bounding box for the dorsal fin, all points of the contour c for which $0.7x_{max} \leq x_i \leq 0.8x_{max}$ are found. From these points, the smallest y-value is computed, and this is taken to approximate the top most point of the adipose fin. After this point is located, a square bounding box with a width of $\frac{1}{15}$ the fork length and height of $\approx \frac{1}{30}$ the fork length is centered about this point and given to the segmentation model, which produced the adipose fin segmentation.

(3) (*Caudal Fin*) To estimate a bounding box for the caudal fin, first all points of the contour $c = [c_0, c_1, \dots c_n]$ for which $x_i > \overline{x}$ were computed. These are the contour points along the

back half of the fish where the caudal fin is. Next, the minimum and maximum y-values along this subset of the contour were computed. A ratio of $\frac{2}{9}$ is provided, which estimated that caudal fin will contained in the last $\frac{2}{9}$ of the fork length. Finally, a bounding box which is vertically between the minimum and maximum y values, and horizontally between the right most point of the contour and the last $\frac{2}{9}$ of the contour was given to the segmentation model, which produced the caudal fin segmentation.

(4) (*Anal Fin*) To estimate a bounding box for the dorsal fin, all points of the contour c for which $0.6x_{max} \leq x_i \leq 0.7x_{max}$ are found. From these points, the largest y-value is computed, and this is taken to approximate the bottom most point of the anal fin. After this point is located, a square bounding box with a width and height of $\frac{1}{10}$ the fork length is centered about this point and given to the segmentation model, which produced the adipose fin segmentation.

(5) (*Pelvic Fin*) To estimate a bounding box for the dorsal fin, all points of the contour c for which $0.5x_{max} \leq x_i \leq 0.6x_{max}$ are found. From these points, the largest y-value is computed, and this is taken to approximate the bottom most point of the pelvic fin. After this point is located, a bounding box with a width of $\frac{1}{15}$ the fork length and height of $\approx \frac{1}{30}$ the fork length is centered about this point and given to the segmentation model, which produced the pelvic fin segmentation.

(6) (*Pectoral Fin*) To estimate a bounding box for the caudal fin, first all points of the contour $c = [c_0, c_1, \ldots c_n]$ for which $0.2\overline{x} < x_i \leq 0.4\overline{x}$ were computed. From these points, the maximum y value was computed. After this point is located, a square bounding box with a width and height of $\frac{1}{15}$ the fork length is centered about this point and given to the segmentation model, which produced the pectoral fin segmentation.

(7) (*Eyeball*) To estimate a bounding box for the caudal fin, first all points of the contour $c = [c_0, c_1, \ldots c_n]$ for which $0.1\overline{x} < x_i \leq 0.25\overline{x}$ were computed. From these points, the mean y value was computed to estimate the midpoint which would be around eye level. After this point is located, a square bounding box with a width and height of $\frac{1}{20}$ the fork length is centered about this point and given to the segmentation model, which produced the pectoral fin segmentation.

These heuristics are included for the sake of completeness, but also to emphasize that these procedures are easily altered and customizable to the particular interests of the researcher. For the purposes of this manuscript, the main goal was to disclude fins whenever present, and accurate fin segmentation was not of critical importance. Hence these heuristics are not optimized, and could be improved if the main task was to estimate caudal fin sizes, for example.

Bibliography

[1] Texture classification by cellular neural network and genetic learning. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 2-Conference B: Computer Vision & Image Processing.(Cat. No. 94CH3440-5)*, pages 381–383. IEEE, 1994.

[2] Edward H Adelson. Checkershadow illusion, 1995.

[3] H Akaike. Information theory as an extension of the maximum likelihood principle. á in: Petrov, bn and csaki, f. In *Second International Symposium on Information Theory. Akademiai Kiado, Budapest, pp. 276Á281*, 1973.

[4] Akram Aldroubi, Carlos Cabrelli, and Ursula M. Molter. Wavelets on irregular grids with arbitrary dilation matrices and frame atoms for $L^2(\mathbb{R}^d)$. *Applied and Computational Harmonic Analysis*, 17(2):119–140, 2004. Special Issue: Frames in Harmonic Analysis, Part II.

[5] Mathieu Andreux, Tomás Angles, Georgios Exarchakis, Roberto Leonarduzzi, Gaspar Rochette, Louis Thiry, John Zarka, Stéphane Mallat, Joakim Andén, Eugene Belilovsky, et al. Kymatio: Scattering transforms in python. *Journal of Machine Learning Research*, 21(60):1–6, 2020.

[6] Chaimae Anibou, Mohammed Nabil Saidi, and Driss Aboutajdine. Textured image segmentation via scattering transform and score fusion. In *2016 International Symposium on Signal, Image, Video and Communications (ISIVC)*, pages 24–28, 2016.

[7] Laleh Armi and Shervan Fekri Ershad. Texture image analysis and texture classification methods - A review. *CoRR*, abs/1904.06554, 2019.

[8] E. Bedrosian. A product theorem for Hilbert transforms. *Proceedings of the IEEE*, 51(5):868–869, 1963.

[9] Phil Brodatz. Textures: a photographic album for artists and designers. 1966.

[10] Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1872–1886, 2013.

[11] T. Bulow and G. Sommer. Hypercomplex signals-a novel extension of the analytic signal to the multidimensional case. *IEEE Transactions on Signal Processing*, 49(11):2844–2852, November 2001.

[12] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing*, 16(5):1190–1208, 1995.

[13] Paulo Cavalin and Luiz S. Oliveira. A review of texture classification methods and databases. In *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T)*, pages 1–8, 2017.

[14] P. Cerejeiras and U. Kähler. Monogenic signal theory. In *Operator theory*, pages 1–22. Springer, 2014.

[15] Wai Ho Chak and Naoki Saito. Monogenic wavelet scattering network for texture image classification. *JSIAM Letters*, 15:21–24, 2023.

[16] Yan Qiu Chen, Mark S Nixon, and David W Thomas. Statistical geometrical features for texture classification. *Pattern recognition*, 28(4):537–552, 1995.

[17] Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. MeshLab: an Open-Source Mesh Processing Tool. In Vittorio Scarano, Rosario De Chiara, and Ugo Erra, editors, *Eurographics Italian Chapter Conference*. The Eurographics Association, 2008.

[18] Paolo Cignoni, Alessandro Muntoni, Guido Ranzuglia, and Marco Callieri. MeshLab.

[19] Ryan M Cinalli and Ruth Lehmann. A spindle-independent cleavage pathway controls germ cell formation in drosophila. *Nat. Cell Biol.*, 15(7):839–845, July 2013.

[20] L. Cohen. *Time-frequency Analysis*. Electrical engineering signal processing. Prentice Hall PTR, 1995.

[21] Ronald R Coifman, Jacques Peyrière, and Guido Weiss. On complex analytic tools, and the holomorphic rotation methods. In *The Mathematical Heritage of Guido Weiss*, pages 145–161. Springer, 2025.

[22] Ronald R. Coifman and Stefan Steinerberger. Nonlinear Phase Unwinding of Functions. *Journal of Fourier Analysis and Applications*, 23(4):778–809, August 2017.

[23] Ronald R Coifman and Hau-Tieng Wu. On the practical use of Blaschke decomposition in nonstationary signal analysis. *arXiv preprint arXiv:2508.10861*, 2025.

[24] Andrew R Conn, Katya Scheinberg, and Luis N Vicente. *Introduction to derivative-free optimization*. SIAM, 2009.

[25] Keenan Crane. Discrete Differential Geometry: An Applied Introduction.

[26] Zhe Cui, Jianjiang Feng, Shihao Li, Jiwen Lu, and Jie Zhou. 2-D phase demodulation for deformable fingerprint registration. *IEEE Transactions on Information Forensics and Security*, 13(12):3153–3165, 2018.

[27] Aaron T David, Charles A Simenstad, Jeffery R Cordell, Jason D Toft, Christopher S Ellings, Ayesha Gray, and Hans B Berge. Wetland loss, juvenile salmon foraging performance, and density dependence in Pacific Northwest estuaries. *Estuaries and Coasts*, 39(3):767–780, 2016.

[28] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems*, 27, 2014.

[29] Richard Delanghe. Clifford Analysis: History and Perspective. *Computational Methods and Function Theory*, 1(1):107–153, September 2001.

[30] Li Deng. The MNIST database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[31] Anne Ephrussi and Ruth Lehmann. Induction of germ cell formation by oskar. *Nature*, 358(6385):387–392, 1992.

[32] Dennis Eschweiler, Richard S Smith, and Johannes Stegmaier. Robust 3d cell segmentation: extending the view of cellpose. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 191–195. IEEE, 2022.

[33] Michael Felsberg. *Low-Level Image Processing with the Structure Multivector*. PhD thesis, University of Kiel, Kiel, 2002.

[34] N. I. Fisher. *Statistical Analysis of Circular Data*. Cambridge University Press, 1993.

[35] Dennis Gabor. Theory of communication. part 1: The analysis of information. *Journal of the Institution of Electrical Engineers-part III: radio and communication engineering*, 93(26):429–441, 1946.

[36] Corrado Gini. Measurement of inequality of incomes. *The economic journal*, 31(121):124–125, 1921.

[37] Vinat Goyal and Sanjeev Sharma. Texture classification for visual data using transfer learning. *Multimedia Tools and Applications*, 82(16):24841–24864, 2023.

[38] D. Griffin and Jae Lim. Signal estimation from modified short-time Fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, 1984.

[39] S.L. Hahn. Multidimensional complex signals with single-orthant spectra. *Proceedings of the IEEE*, 80(8):1287–1300, August 1992.

[40] Stefan Held et al. Steerable wavelet frames based on the Riesz transform. *Journal of the Optical Society of America A*, 19(3), March 2010.

[41] Ludimar Hermann. Eine Erscheinung simultanen Contrastes. *Archiv für die gesamte Physiologie des Menschen und der Tiere*, 3:13–15, 1870.

[42] Eric J. Holmes and Carson A. Jeffres. Juvenile Chinook salmon weight prediction using image-based morphometrics. *North American Journal of Fisheries Management*, 41(2):446–454, 2021.

[43] M Holschneider. *Wavelets: An Analysis Tool*. Oxford University Press, 03 1995.

[44] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pages 2366–2369. IEEE, 2010.

[45] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*. John Wiley & Sons, 2013.

[46] Niall Hurley and Scott Rickard. Comparing measures of sparsity. *IEEE Transactions on Information Theory*, 55(10):4723–4741, 2009.

[47] Uriah Israel, Markus Marks, Rohit Dilip, Qilin Li, Changhua Yu, Emily Laubscher, Ahamed Iqbal, Elora Pradhan, Ada Ates, Martin Abt, et al. CellSAM: a foundation model for cell segmentation. *Nature methods*, 2025.

[48] Gerald Kaiser. *A friendly guide to wavelets*. Birkhauser Boston Inc., USA, 1994.

[49] Peter Kareiva, Michelle Marvier, and Michelle McClure. Recovery and management options for spring/summer Chinook salmon in the Columbia River Basin. *Science*, 290(5493):977–979, 2000.

[50] Mohamed Kaseb, Guillaume Mercère, Hermine Biermé, Fabrice Brémand, and Philippe Carré. Phase estimation of a 2D fringe pattern using a monogenic-based multiscale analysis. *Journal of the Optical Society of America A*, 36(11):C143, November 2019.

[51] Marcus D. Kilwein, Pearson Miller, Kwan Yin Lee, Miriam Osterfield, Alex Mogilner, Stanislav Y. Shvartsman, and Elizabeth R. Gavis. Formation of *Drosophila* germ cells requires spatial patterning of phospholipids. *Current Biology*, 35(7):1612–1621.e3, Apr 2025.

[52] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[53] Brian Knight and Naoki Saito. A structurally coherent spatial phase estimate. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 311–323. Springer, 2025.

[54] Henning H Kristiansen, Moa Metz, Lorena Silva-Garay, Fredrik Jutfelt, and Robine H J Leeuwis. HusMorph: a simple machine learning app for automated morphometric landmarking. *Conservation Physiology*, 13(1):coaf073, 10 2025.

[55] Rishikesh Kulkarni and Pramod Rastogi. Two-step phase shifting interferometry based on orientation selective monogenic filtering. *Optics Communications*, 450:208–215, 2019.

[56] Kieran Larkin et al. Natural demodulation of two-dimensional fringe patterns. i. general background of the spiral phase quadrature transform. *Journal of the Optical Society of America A*, 18(8), August 2001.

[57] Jeffrey Larson, Matt Menickelly, and Stefan M Wild. Derivative-free optimization methods. *Acta Numerica*, 28:287–404, 2019.

[58] Li Liu, Jie Chen, Paul W. Fieguth, Guoying Zhao, Rama Chellappa, and Matti Pietikäinen. A survey of recent advances in texture representation. *CoRR*, abs/1801.10324, 2018.

[59] Sijia Liu, Pin-Yu Chen, Bhavya Kailkhura, Gaoyuan Zhang, Alfred O Hero III, and Pramod K Varshney. A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. *IEEE Signal Processing Magazine*, 37(5):43–54, 2020.

[60] Dario Maio, Davide Maltoni, Raffaele Cappelli, Jim L. Wayman, and Anil K. Jain. FVC2004: Third fingerprint verification competition. In David Zhang and Anil K. Jain, editors, *Biometric Authentication*, pages 1–7, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[61] Stéphane Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, 2012.

[62] Ross Marchant and Paul Jackway. A sinusoidal image model derived from the circular harmonic vector. *Journal of Mathematical Imaging and Vision*, 57, 02 2017.

[63] Kanti V Mardia and Peter E Jupp. *Directional statistics*. John Wiley & Sons, 2009.

[64] Gregory P Meyer. An alternative probabilistic interpretation of the huber loss. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 5261–5269, 2021.

[65] Michel Nahon. *Phase Evaluation and Segmentation*. PhD thesis, Yale University, New Haven, 2000.

[66] Aleksandar Nikolic, Vladislav Volarevic, Lyle Armstrong, Majlinda Lako, and Miodrag Stojkovic. Primordial germ cells: current knowledge and perspectives. *Stem cells international*, 2016(1):1741072, 2016.

[67] Simon Prunet. A low-rank approach to image defringing. *Publications of the Astronomical Society of the Pacific*, 133(1029):114502, 2021.

[68] Tao Qian. Intrinsic mono-component decomposition of functions: An advance of Fourier theory. *Mathematical Methods in the Applied Sciences*, 33(7):880–891, May 2010.

[69] Tao Qian, Wolfgang Sprößig, and Jinxun Wang. Adaptive Fourier decomposition of functions in quaternionic Hardy spaces. *Mathematical Methods in the Applied Sciences*, 35(1):43–64, January 2012.

[70] Jotje Rantung, Frans Palobo Sappu, and Yan Tondok. Real-time measurement method for fish surface area and volume based on stereo vision. *Advances in Science, Technology and Engineering Systems Journal*, 6, 2021.

[71] William Edwin Ricker. Computation and interpretation of biological statistics of fish populations. *Fish. Res. Board Can. Bull.*, 191:1–382, 1975.

[72] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[73] Hans Sagan. *Boundary and Eigenvalue Problems in Mathematical Physics.* Dover Publications, New York, 1989. Reprint of the original 1961 edition published by John Wiley & Sons.

[74] Naoki Saito and David Weber. Explainable and class-revealing signal feature extraction via scattering transform and constrained zeroth-order optimization. In *2025 IEEE Statistical Signal Processing Workshop (SSP)*, pages 196–200, 2025.

[75] Ervin Sejdić, Igor Djurović, and Jin Jiang. Time–frequency feature representation using energy concentration: An overview of recent advances. *Digital signal processing*, 19(1):153–183, 2009.

[76] Cameron S Sharpe, Daniel A Thompson, H Lee Blankenship, and Carl B Schreck. Effects of routine handling and tagging procedures on physiological stress responses in juvenile Chinook salmon. *The Progressive Fish-Culturist*, 60(2):81–87, 03 1998.

[77] L Dennis Smith and Marilyn A Williams. Germinal plasm and determination of the primordial germ cells. In *Symp. Soc. Dev. Biol.;(United States)*, volume 3. Purdue Univ., West Lafayette, IN, 1974.

[78] Hessam Sokooti, Bob de Vos, Floris Berendsen, Mohsen Ghafoorian, Sahar Yousefi, Boudewijn P. F. Lelieveldt, Ivana Isgum, and Marius Staring. 3D convolutional neural networks image registration based on efficient supervised learning from artificial deformations, 2019.

[79] Elias M. Stein and Guido Weiss. *Introduction to Fourier analysis on Euclidean spaces.* Number 32 in Princeton mathematical series. Princeton University Press, Princeton, N.J, 1975.

[80] Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. Cellpose: a generalist algorithm for cellular segmentation. *Nature methods*, 18(1):100–106, 2021.

[81] William Sullivan and William E Theurkauf. The cytoskeleton and morphogenesis of the early drosophila embryo. *Current opinion in cell biology*, 7(1):18–22, 1995.

[82] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

[83] Andre-i Nikolaevich Tikhonov, Alexander V Goncharsky, Vâčeslav Vasilévič Stepanov, and Anatoliĭ Grigorévich Yagola. Numerical methods for the approximate solution of ill-posed problems on compact sets. In *Numerical methods for the solution of Ill-posed problems*, pages 65–79. Springer, 1995.

[84] Michael Unser and Nicolas Chenouard. A unifying parametric framework for 2d steerable wavelet transforms. *SIAM Journal on Imaging Sciences*, 6(1):102–135, 2013.

[85] Yinan Wan, Katie McDole, and Philipp J Keller. Light-sheet microscopy and its potential for understanding developmental processes. *Annual review of cell and developmental biology*, 35(1):655–681, 2019.

[86] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[87] Davis Weber. *On Interpreting Sonar Waveforms via the Scattering Transform.* PhD thesis, Univ. California, Davis, December 2021.

[88] Thomas Wiatowski and Helmut Bölcskei. Deep convolutional neural networks based on semi-discrete frames. In *2015 IEEE International Symposium on Information Theory (ISIT)*, pages 1212–1216. IEEE, 2015.

[89] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. MedMNIST v2-a large-scale lightweight benchmark for 2D and 3D biomedical image classification. *Scientific Data*, 10(1):41, 2023.

[90] Yunhan Yang, Bing Xue, Linley Jesson, Matthew Wylie, Mengjie Zhang, and Maren Wellenreuther. Deep convolutional neural networks for fish weight prediction from images. In *2021 36th International Conference on Image and Vision Computing New Zealand (IVCNZ)*, pages 1–6. IEEE, 2021.

[91] Yanlei Zhang, Damien Martins Gomes, Chen Liu, Ronald R Coifman, Michael Perlmutter, Guy Wolf, Smita Krishnaswamy, and Dhananjay Bhaskar. Bdn: Blaschke decomposition networks. *Authorea Preprints.*

[92] Z. Zhang. PRIMA: Reference Implementation for Powell's Methods with Modernization and Amelioration. available at http://www.libprima.net, DOI: 10.5281/zenodo.8052654, 2023.

[93] Changjun Zhao, Yunyun Dong, Wenhao Wu, Bangsen Tian, Jianmin Zhou, Ping Zhang, Shuo Gao, Yuechi Yu, and Lei Huang. A modification to phase estimation for distributed scatterers in InSAR data stacks. *Remote Sensing*, 15(3), 2023.

[94] Lijun Zhao, Qingsheng Li, and Bingbing Li. SAR target recognition via monogenic signal and Gaussian process model. *Mathematical Problems in Engineering*, 2022(1):3086486, 2022.

[95] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(2):301–320, 2005.