A FULLY ADAPTIVE 2D EMPIRICAL WAVELET TRANSFORM USING WATERSHEDS

A Thesis
Presented to the
Faculty of
San Diego State University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Applied Mathematics

by
Basile Hurat
Spring 2020

ABSTRACT OF THE THESIS

A Fully Adaptive 2D Empirical Wavelet Transform Using Watersheds
by
Basile Hurat
Master of Science in Applied Mathematics
San Diego State University, 2020

In recent work, there have been numerous approaches to 2D empirical wavelet transforms (EWT), which create 2D filter banks based on the information present in an image. These implementations of the EWT have shown promising results regarding image processing and computer vision tasks but their construction makes assumptions about the geometry of the spectrum. In this thesis, the author proposes a 2D empirical wavelet transform that creates filters whose shapes are fully adaptive to the information present in the signal. This is done by the application of the watershed transform, which defines a contour which separates select markers by the path of highest separation. The author then demonstrates the usefulness of such a transform by applying it to a texture segmentation computer vision problem.

TABLE OF CONTENTS

PAGE

ABSTRACT v
LIST OF TABLES viii
LIST OF FIGURES ix
ACKNOWLEDGMENTS x

CHAPTER

1 INTRODUCTION 1
2 BACKGROUND 3
2.1 Scale-Space Theory 3
2.1.1 1D-Discretization 3
2.1.2 2D-Discretization 4
2.2 Watershed Transform 5
2.3 1D Empirical Wavelet Transform 7
2.4 Boundary Detection for 1D EWT 8
3 EMPIRICAL WATERSHED WAVELET TRANSFORM 11
3.1 Detecting Mode Locations 11
3.2 Boundary Selection 12
3.3 Empirical Watershed Wavelet Filters 12
3.4 Defining EWWT 14
4 EXPERIMENTS 18
4.1 Review of 2D Empirical Wavelet Transforms 18
4.1.1 Separable Methods 18
4.1.2 Voronoi EWT 18
4.2 Comparative Results 19
4.3 Advantages of 2D Mode Detection 20
5 APPLICATIONS 26
5.1 Algorithm 26
5.2 Parameters 27
5.3 Experiments 28

5.3.1 Dataset 28

5.3.2 Benchmarks 29

5.3.3 Results 30

5.4 Final Comparisons 33

6 CONCLUSION 34

BIBLIOGRAPHY 36

CHAPTER 1
INTRODUCTION

The study of signals has long been interested in localizing frequency information. This study has extended during the analysis of 2D signals, where 2D frequencies, or patterns, have great importance in various applications in image processing and computer vision, especially in regards to texture analysis. The practice of decomposing 2D signals into their amplitude-modulated frequency modulated (AM-FM) modes is well studied and well used. However, it is also a challenging problem, with the Gabor-Heisenberg uncertainty principle acting as an obstacle for any method. Early on, tools such as the short time Fourier transform, Gabor filter banks, and wavelets all employ low-pass and pass-band filter banks in the image's Fourier spectrum to extract frequency features. But these were rigid methods, often needing to be tuned to the specific problem at task. As time went on, many applications demanded more adaptive methods, whose goals are to extract the frequency features in ways that adapt to the information present in the signal. As such, there has been much focus recently on such techniques.

Wavelet packets [6] are one of the earliest attempts at adaptive methods, which takes the approach of wavelets and tries to build a filter bank to decompose the signal. However, instead of building the same filter bank at every application, it uses entropy conditions using the basis pursuit algorithm to decide whether to merge filters. Despite this, the locations for these filters still follow a pattern, and are therefore prescripted.

Another approach, which is purely algorithmic, is the empirical mode decomposition (EMD)[16]. Instead of attempting to build a filter bank to decompose the signal, this method, which is widely used, attempts to detect the harmonic modes that compose the signal using an iterative algorithm. However, it is very hard to model mathematically and lacks mathematical theory. Furthermore, it has proven quite difficult to implement in 2D.

Variational Mode Decomposition (VMD) [8] [9] is another adaptive method, which attempts to create a filter bank of Gaussians using optimization techniques. However, it requires an input of the number of modes to fit and assumes that the modes are indeed Gaussian.

A recent paper by S. Madhe et al. [22] proposes a method to build a two-dimensional arbitrary shape filter bank. While this method allows for the construction of arbitrarily shaped filters, it does not automatically detect what the ideal shape for those problems should be. As such, it is not particularly adaptive. Furthermore, it is limited in making two filters—a low-approximation coefficient and a band-pass coefficient—which is insufficient for many applications.

Yet another adaptive method is the empirical wavelet transform (EWT) [11], which builds a set of filters around detected modes that are similar in construction to Littlewood-Paley wavelets. There are many 2D implementations of the EWT. The first, proposed by Gilles in [14], were the Tensor, Ridgelet, and Curvelet implementations, but the filter banks built with these methods are still restricted in their shape, being either built around rectangular or angular sector grids in the Fourier domain. Later, Ames in [1] proposed a more flexible 2D method using Voronoi diagrams. This approach detects and separates modes using voronoi tilings. While these shapes are more arbitrary than grids, Voronoi tilings are still limited in shape, as they are constructed with straight lines at midpoints, and this is not always ideal.

In this paper, we will start from a similar approach to the Ames in [1] and try to create an fully adaptive 2D approach to the empirical wavelet transform. Instead of applying Voronoi tilings, whose shapes are still somewhat limited, we use the watershed transform to find the lowest boundary curve for each mode. From this boundary, we construct a filter bank in a way similar to the empirical wavelet transform. This paper also applies this new implementation of the EWT to the task of unsupervised texture segmentation, as recent papers by Y. Huang et al. [17] [18] have demonstrated the use of EWT methods in texture segmentation.

Chapter 2 will focus on background information, reviewing the concepts of scale-space theory and the watershed transform, as well as the 1D empirical wavelet transform. Chapter 3 then goes through the definition the empirical watershed wavelet transform (EWWT). Chapter 4 shows the results of the empirical watershed wavelet transform and how it compares to previous methods. Chapter 5 uses the EWWT to extract features to perform an unsupervised texture segmentation task. Chapter 6 draws conclusions and reflects on future works related to the EWWT.

CHAPTER 2
BACKGROUND

The implementation we propose in this thesis requires the use of multiple ideas and tools. As such, we begin by discussing these concepts.

2.1 Scale-Space Theory

We begin with a review of scale-space theory, which is a representation of functions at various scales. Let us consider a function $f(x)$ and a kernel $K(x;t) = \frac{1}{\sqrt{2\pi t}}e^{-x^2/2t}$. We define the continuous scale-space representation of f, described by Lindeberg in [21], as

$$L(x,t) = T_t[f](x) = K(x;t) * f(x) \tag{2.1}$$

if it satisfies the following axioms:

- Linearity,
- Shift invariance,
- Semi-group structure,
- Kernel space invariance, positivity and normalization,
- Non-creation of local extrema.

This can be interpreted as a continuous smoothing of the function, where t represents the degree of smoothness. As such, as t increases, we observe a smoother and smoother representation of our function $f(x)$, removing more detailed patterns. It has been proven in [21] that the Gaussian kernel is the only kernel that satisfies these axioms in the continuous case.

2.1.1 1D-Discretization

When we turn to the discrete case, we are not guaranteed to preserve the scale-space axioms. Instead, Lindeberg [21] proposes a different set of conditions:

- Use linear and shift invariant transformations of the original signal,
- For a scale parameter t, the original signal can be recovered at $t = 0$, while higher values of t correspond to coarser levels of the signal,
- All signals are real-valued functions.

Given this, we can define the discrete scale-space representation

$$L(n,t) = \sum_{n=-\infty}^{\infty} K(n;t) f(m-n) \tag{2.2}$$

Unlike the continuous scale-space, where the Gaussian kernel is the only kernel satisfying the axioms, there are multiple kernels which sastify the axioms of the discrete scale-space. However, the main one of concern is the discrete Gaussian kernel, which was formulated by Lindeberg [21]

$$T(n;t) = e^{-\alpha t} I_n(\alpha t) \tag{2.3}$$

where $\alpha \in \mathbb{R}$ and I_n are the modified Bessel functions of integer order n. This kernel not only satisfies the discrete axioms, but also other important properties, namely semi-group structure, symmetry, and normalization. Furthermore, in practice, rather than the sum going from $(-\infty, \infty)$, we instead pick a bound M based on the level of accuracy we wish to meet. Shown in Figure 2.1 is the result of a 1D scale-space representation.

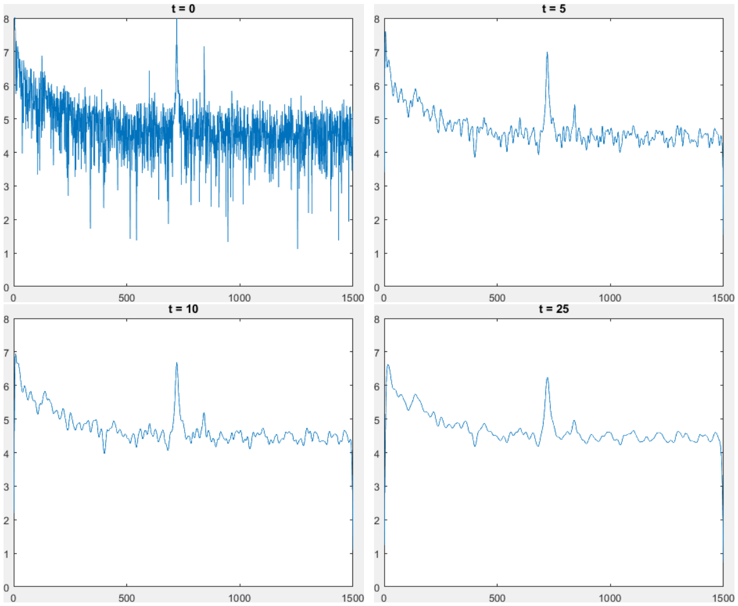


Figure 2.1. 1D Scale-Space

2.1.2 2D-Discretization

As we plan to work with sampled images, we require a 2D scale-space theory. While Lindeberg showed in [21] that this does not exist in the continuous case, this is still possible in the discrete case, which is sufficient for the application to images. When considering a discrete 2D scale-space, Lindeberg cites a yet another set of axioms:

- Semi-Group property,
- Symmetry,
- Continuity,
- Reduction of extrema.

Along with these axioms, he proposes the 2D discrete Gaussian kernel, which is constructed using the one dimensional discrete Gaussian kernel, as it is separable. Thus, the 2D scale-space representation of a signal can be written as

$$L(x, y; t) = \sum_{m=-\infty}^{\infty} T(m; t) \sum_{n=-\infty}^{\infty} T(n; t) f(x - m, y - n). \tag{2.4}$$

Shown in Figure 2.2 is the result of a 2D scale-space representation.

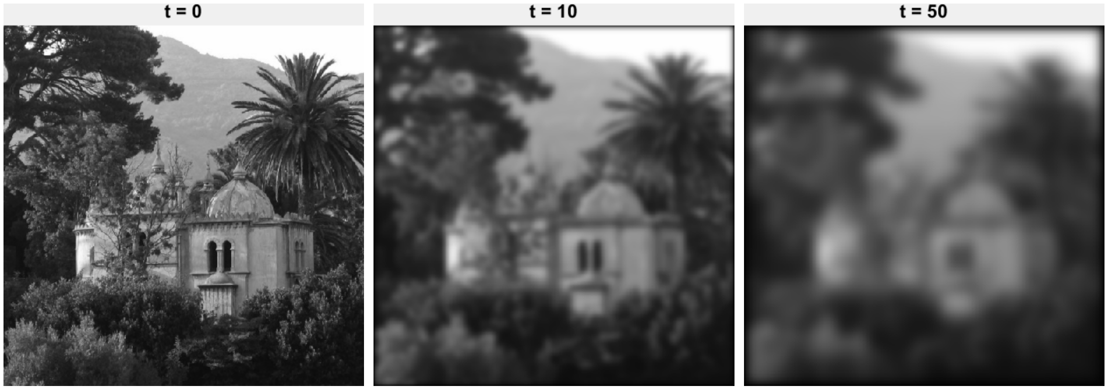


Figure 2.2. 2D Scale-Space

2.2 Watershed Transform

The Watershed transform is an algorithmic approach to contour generation for image segmentation. First proposed by S. Beucher and C. Lantuejoul in [4], we imagine a grayscale image as a topographic landscape with the pixel intensity representing the height at that point, segment it into its catchment basins, and define the contour as the watershed lines separating them. We define it thusly.

Given a set of local minima $\{x_i\}$, we uniformly flood the topographic landscape with water, starting from the lowest minima. As the water level rises to reach the height of new minima, we flood from those minima as well and fill their catchment basins uniformly. Lastly, where two bodies of water would flow into one another, we construct a dam. This process is shown in Figure 2.3. We define the contour Γ on that dam, and this contour defines a segmentation along the highest path between the regions.

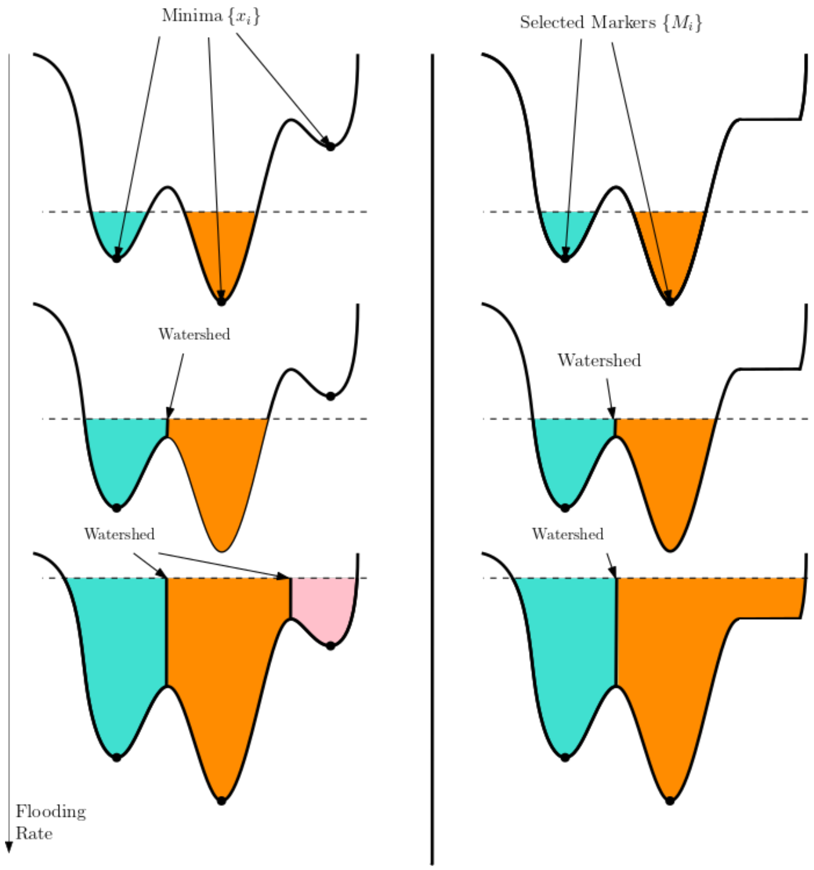


Figure 2.3. Example of Flooding for Watershed Generation Before and After Imposing Minima

Later, F. Meyer and S. Beucher discuss an approach by allowing the flooding to happen from a set of select markers $\{M_i\}$ rather than all local minima by imposing minima through morphological reconstruction. It does so by forcing the selected markers $\{M_i\}$ to be minima by setting those points to $M_i = -\infty$. It then takes the catchment basins of the minima $x_i \notin \{M_i\}$ and fills them up to the point that these regions contain no minima. The watershed contour Γ is then constructed as usual. This morphological reconstruction allows the segmentation contour Γ of an image to be the line of highest separation around select markers, which fixes the oversegmentation problem in noisy images and creates a more flexible segmentation method. F.Meyer later revisited the problem and defined it in a mathematical framework through the introduction of the topographic distance metric and the details of this metric can be found in his paper [26].

2.3 1D Empirical Wavelet Transform

The EWT, proposed by Gilles in [11], is an adaptive data driven method to build a filter bank that separates signals into their harmonic modes. Modes are defined as amplitude modulated-frequency modulated (AM-FM) components of signals and can be described by intrinsic mode functions (IMFs). IMFs are assumed to have a finite or rapidly decaying support, and as such, the goal of the EWT is to construct an adaptive filter bank of band-pass filters in the Fourier domain and then independently filter out each mode. The benefit of this approach is that it is rooted in wavelet theory and has a large body of mathematical theory to support it.

The idea is to create a bank of N filters similar to wavelets, where the first filter is a low-pass filter, and the remaining $N-1$ are band-pass filters whose pass bands are 'well chosen' to capture the relevant modes in the signal.

We construct these band-pass filters by around a set of boundaries which delineate the expected support of the harmonic modes $\{\omega_n\}$ where $n \in [0, N]$ and whose boundaries $\omega_n \in [0, \pi]$ and, as is convention, define $\omega_0 = 0$ and $\omega_N = \pi$. This set of boundaries is shown in Figure 2.4, with the shaded regions defining the transition area of width of overlapping filters.

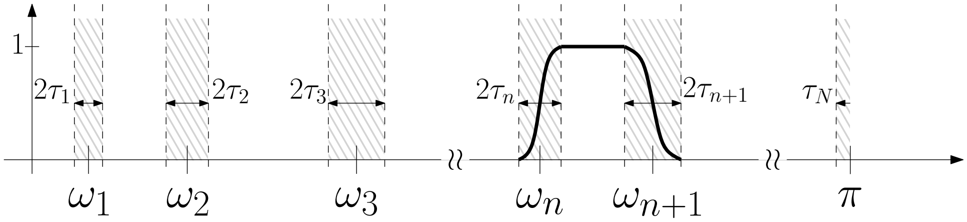


Figure 2.4. Construction of Empirical Wavelet

From these boundaries and transition width, we construct filters using the same ideas as the Littlewood-Paley wavelets in [11]. If we define our transition width τ_n to be proportional to the boundary, $\tau_n = \gamma\omega_n$, we can then define the empirical scaling function and empirical wavelets in the Fourier domain with

$$\hat{\phi}_n(\omega) = \begin{cases} 1 & \text{if } |\omega| \leq (1-\gamma)\omega_n \\ \cos\left[\frac{\pi}{2}\beta\left(\frac{1}{2\gamma\omega_n}(|\omega| - (1-\gamma)\omega_n)\right)\right] & \text{if } (1-\gamma)\omega_n \leq |\omega| \leq (1+\gamma)\omega_n \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

and

$$\hat{\psi}_n(\omega) = \begin{cases} 1 & \text{if } |\omega| \leq (1-\gamma)\omega_n \\ \cos\left[\frac{\pi}{2}\beta\big(\frac{1}{2\gamma\omega_{n+1}}(|\omega| - (1-\gamma)\omega_{n+1})\big)\right] & \text{if } (1-\gamma)\omega_{n+1} \leq |\omega| \leq (1+\gamma)\omega_{n+1} \\ \sin\left[\frac{\pi}{2}\beta\big(\frac{1}{2\gamma\omega_n}(|\omega| - (1-\gamma)\omega_n)\big)\right] & \text{if } (1-\gamma)\omega_n \leq |\omega| \leq (1+\gamma)\omega_n \\ 0 & \text{otherwise,} \end{cases} \tag{2.6}$$

where $\beta(x)$ is an arbitrary $\mathcal{C}^k$ function such that

$$\beta(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 & \text{if } x \geq 1 \end{cases} \quad \text{and } \beta(x) + \beta(1-x) = 1 \ \ \forall x \in [0,1]. \tag{2.7}$$

One such function that satisfies this is

$$\beta(x) = x^4(35 - 84x + 70x^2 - 20x^3). \tag{2.8}$$

In [11], Gilles shows that one can choose γ such that a tight frame is constructed. With that, we define the empirical wavelet transform by defining the approximation coefficient $\mathcal{W}_t^{\epsilon}(0,t)$ as the inner product with the scaling function,

$$\mathcal{W}_t^{\epsilon}(0,t) = \langle f, \phi_1 \rangle = \int f(\tau)\overline{\phi_1(\tau - t)}d\tau = \left(\hat{f}(\omega)\overline{\hat{\phi}_1(\omega)}\right)^{\vee} \tag{2.9}$$

and we define the detail coefficients $\mathcal{W}_t^{\epsilon}(n,t)$ as the inner product of f with the empirical wavelet functions,

$$\mathcal{W}_t^{\epsilon}(n,t) = \langle f, \psi_n \rangle = \int f(\tau)\overline{\psi_n(\tau - t)}d\tau = \left(\hat{f}(\omega)\overline{\hat{\psi}_n(\omega)}\right)^{\vee} \tag{2.10}$$

We also define the reconstruction as

$$f(t) = \mathcal{W}_f^{\epsilon}(0,t) * \phi_1(t) + \sum_{n=1}^{N} \mathcal{W}_t^{\epsilon}(n,t) * \psi_n(t) = \left(\widehat{\mathcal{W}_t^{\epsilon}}(0,\omega)\hat{\phi}_1(\omega) + \sum_{n=1}^{N} \widehat{\mathcal{W}_f^{\epsilon}}(n,\omega)\hat{\psi}_n(\omega)\right)^{\vee}. \tag{2.11}$$

2D methods will be discussed later in Chapter 4, where we compare them to the EWWT, but for now, we continue our focus on the 1D EWT transform by discussing boundary selection.

2.4 Boundary Detection for 1D EWT

Clearly, the boundary selection is quite important in this construction, as we want to separate each mode as completely as possible. Recall that modes are considered to have finite support in the Fourier domain. As such, modes often have corresponding maxima. Therefore, in the paper that introduced the 1D empirical wavelet transform [11], Gilles chose these boundaries by first detecting N peaks of the magnitude spectrum of the signal and then choosing the boundaries based on the midpoints between these peaks. In another paper [14], however, Gilles proposed to choose boundaries instead at the lowest minimum between these peaks and showed that it led to a better separation of modes. Later, in [12], Gilles et. al. propose the method that we will focus on.

In that paper [12], the authors use scale-space representations of their signal alongside histogram separation techniques to detect meaningful boundaries. They begin constructing a fine-to-coarse representation of the signal using discrete Gaussian kernel. Then, at each discretized step, $t = 0, t_1, \ldots, t_n$, they detect minima. Because scale-space kernels have the property of not introducing new extrema, this leads to minima disappearing as we advance in the scale-space, allowing the authors to track the change in presence of minima and map the persistence of each minimum. The persistence through scale-space of an extremum can be defined as the value t in which the extremum disappears during the fine-to-coarse representation since the continuous smoothing implies that extrema will eventually disappear as the signal flattens.

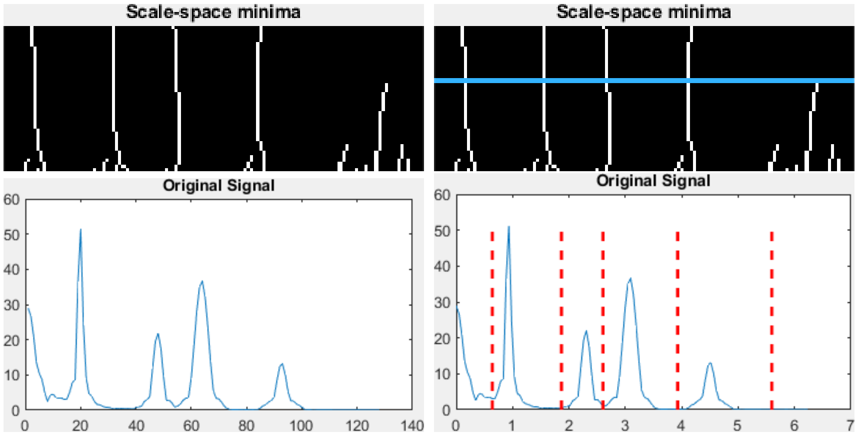


Figure 2.5. Left: Original signal and minima through scale-space, Right: Threshold detected and persistent minima found

From there, the authors construct a histogram of minimum persistence. Then, they use clustering methods to detect a threshold which separate the histogram into two halves, as shown in Figure 2.5. The minima may then be categorized into two groups, non-persistent minima, whose persistence is below the detected threshold, and persistent minima, whose persistence through the scale-space is above the threshold.

A commonly used clustering approach is Otsu's method, a method that maximizes the inter-class variance of the class probability of the histogram bins and the respective class averages.

It is worth noting that this approach acts both as an automatic mode and boundary detection, as by automatically selecting the boundaries, we have sufficient information to construct our filter bank around the modes of the signal.

CHAPTER 3
EMPIRICAL WATERSHED WAVELET TRANSFORM

To build a fully 2D empirical wavelet transform filter bank, we first must detect the locations of our modes. Then, we must find an ideal boundary to isolate them. Lastly, we must build the empirical wavelet filters. Here, since we are working with images, we assume our signal f is 2D, real and of square dimensions $\mathcal{N} \times \mathcal{N}$.

3.1 Detecting Mode Locations

Before we begin, we must define what we consider ”meaningful” modes to be. We consider that modes have a spectrum of bounded support in the Fourier domain, and thus are best separated by minimal paths between maxima. However, to prevent separation of multi-peak modes, it is not sufficient to simply consider all the maxima of the signal, and so we instead consider that a mode is meaningful if it is centered around a persistent maximum. And thus, our first step is to detect persistent maxima and the approach is a 2D extension of the method discussed in Section 2.4.

Given a signal f, we therefore begin by constructing a 2D scale-space representation of our signal's magnitude spectrum,

$$L(x, y; t) = \sum_{m=-M}^{M} T(m; t) \sum_{n=-M}^{M} T(n; t) |\hat{f}(x - m, y - n)|,$$

where $T(.; t)$ is our discrete Gaussian kernel and M is sufficiently large to minimize our error by some bound.

At each value $t = 0, t_1, t_2, \ldots, t_N$, we detect maxima of our signal at that scale to track their evolution during the fine-to-coarse representation. At the end of this fine-to-coarse representation, each maximum has a corresponding ”worm” which traces the path of the maximum through the scale-space, shown in Figure 3.1, and gives persistence of the maximum until its eventual disappearance via its final height. We can then construct a histogram of the persistence of our maxima, which we separate into two classes using Otsu's method, similar to Section 2.4.

At the end of this process, we have a set of persistent maxima of our signal $\{\mu_i\}$ where $i = 1, 2, 3, \ldots, n_P$ where n_P is the number of persistent maxima and thus

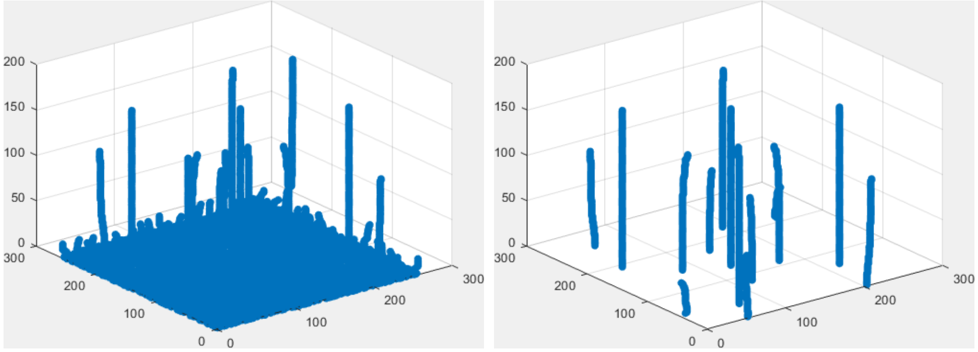


Figure 3.1. Scale-space Worms generated from maxima and detected persistent maxima

detected modes. These maxima, shown in Figure 3.2 (c), should share the Fourier spectrum's symmetry about the origin.

3.2 Boundary Selection

Given our set of persistent maxima $\{\mu_i\}$, we know where the meaningful modes are located. The goal then is to separate them from an ideal boundary contour. We begin by taking our signal f and inverting it $f_- = -|\hat{f}|$ so that our set of maxima become minima. Then, we use a 2D extension of the morphological reconstruction described in Section 2.2 to make it so that these persistent minima are the only minima present in the signal, shown in Figure 3.2 (d). From there, we perform the watershed transform, also described in Section 2.2, which finds the segmentation line Γ of highest altitude on f_- that separate $\{\mu_i\}$. This contour Γ shall be our boundary, as the path of highest altitude on f_- represents the path of lowest altitude on f, shown in Figure 3.2 (e).

3.3 Empirical Watershed Wavelet Filters

Before we can define filters, we must define our regions. Using the contour Γ, we can define regions $\{\tilde{\Omega}_i\}$ corresponding to the catchment basins of each persistent maximum. Since we are only considering real signals and the Fourier transform is symmetric with respect to the origin, we begin by pairing regions which do not include the origin. To do this, we follow the algorithm shown in Algorithm 1.

From there, since the watershed transform classifies some pixels as part of the contour Γ, rather than a region, we must assign the pixels on the watershed contour to a region. This assignment is not critical so long as it is symmetric, as these boundaries represent the lowest altitude between regions. We therefore go to each paired region Ω_k, in order of index, and assign the unassigned portions of the boundary of a region Ω_k.

```
Algorithm 1: Algorithm For Pairing Regions
    Input: A set of unpaired maxima $\{\mu_i\}$ and corresponding regions $\{\tilde{\Omega}_i\}$,
     where $i = 1, 2, \dots, n_P$
    $m_P \leftarrow \begin{cases} \frac{n_P+1}{2} \text{ if } (0,0) \in \{\mu_i\} \\ \frac{n_P}{2} \text{ otherwise} \end{cases}$
    for $k = 1, 2, \dots m_P$
      1.) $\Lambda_k = \{i | \pm \mu_k \in \tilde{\Omega}_i\}$
      2.) $\Omega_k = \bigcup_{i \in \Lambda_k} \tilde{\Omega}_i$
      3.) $\{\mu_i\} \leftarrow \{\mu_i\} - \{\mu_{\Lambda_k}\}$
    Output: Set of paired regions $\{\Omega_k\}$, where $k = 1, 2, \dots, m_P$
```

After this, each sample of our 2D signal is classified to one of our paired regions $\{\Omega_k\}$, where $k = 1, 2, 3, \dots, m_P$ where $m_P = \frac{n_P - 1}{2} + 1$ if there is a region which contains the origin, and $m_P = \frac{n_P}{2}$ if there does not. These regions are shown in Figure 3.2 (f).

Once we have defined our regions $\{\Omega_k\}$ appropriately, we want to define a transition area. However, we want to accomplish this in a way to have symmetry on bordering regions, in order to eventually guarantee a frame. As such, we consider distance transforms. Given a region Ω_k,we define $\partial\Omega_k$ to be the region's border, which lies at the sub-pixel level between border pixels within the region and the border pixels outside the region. Then, we define our distance transform , which we can see in Figure 3.2 (g) to be

$$D_k(\omega) = \begin{cases} \frac{2\pi}{N} \min\left(d(\omega, \delta\Omega_k)\right) & \text{if } \omega \in \Omega_k \\ -\frac{2\pi}{N} \min\left(d(\omega, \delta\Omega_k)\right) & \text{if } \omega \notin \Omega_k, \end{cases} \tag{3.1}$$

which is scaled such that $D_k(\omega) \in [-2\pi, 2\pi]$. In our application, we define $d(.,.)$ to be the quasi-euclidean distance,

$$d(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} (\sqrt{2} - 1)|y_2 - y_1| + |x_2 - x_1| & \text{if } |x_2 - x_1| > |y_2 - y_1| \\ (\sqrt{2} - 1)|x_2 - x_1| + |y_2 - y_1| & \text{if } |y_2 - y_1| > |x_2 - x_1|, \end{cases} \tag{3.2}$$

where $\boldsymbol{x} = (x_1, y_1)$ and $\boldsymbol{y} = (x_2, y_2)$. Then, we define the empirical watershed wavelet, shown in Figure 3.2 (h), to be

$$\hat{\varphi}_k(\omega) = \begin{cases} 1 & \text{if } D_k(\omega) > \tau \\ \cos\left(\frac{\pi}{2}\beta(\frac{\tau - D_k(\omega)}{2\tau})\right) & \text{if } D_k(\omega) \leq |\tau| \\ 0 & \text{if } D_k(\omega) < -\tau, \end{cases} \tag{3.3}$$

where β is defined as in Section 2.3 and τ is our transition area width. A full pipeline to build the empirical watershed wavelets is given in the algorithm below and visualized in Figure 3.2.

```
Algorithm 2: Algorithm For Building Empirical Watershed Wavelets
  Input: An image $f$ and transition thickness $\tau$
    1.) $\hat{f} = \mathcal{F}(f)$.
    2.) Track maxima of $|\hat{f}|$ through scale-space representation, getting lengths $L$ for each maximum,
    3.) Construct histogram and find the threshold using Otsu's method to distinguish persistent maxima $\{m_i\}$,
    4.) Invert $f_- = -|\hat{f}|$, and use morphological reconstruction to impose minima of $\hat{f}_-$ with set of persistent minima $\{m_i\}$,
    5.) Perform watershed transform to get regions $\{\Omega_i\}$ corresponding to minima,
    6.) Construct paired regions $\{\Omega_k\}$ via radial symmetry and assign watershed line to regions
    7.) Perform distance transforms $\{D_k(\omega)\}$ of each region in $\{\Omega_k\}$
    8.) Construct empirical watershed wavelet filter bank $\{\hat{\varphi}_k\}$ using $D_k$ based on transition width $\tau$.
  Output: Set of empirical watershed wavelets $\{\hat{\varphi}_k\}$.
```

3.4 Defining EWWT

Before continuing, we show that the set $\{\varphi_k\}$ forms a frame.

Proposition 3.1. *The set $\{\varphi_k\}$ is a frame.*

Proof. We know that $\forall v \in \mathbb{R}^2$,

$$\sum_{k=1}^{m} |\langle v, \varphi_k \rangle|^2 = \sum_{k=1}^{m} \int |\hat{v} \cdot \hat{\varphi}_k|^2 d\omega = \sum_{k=1}^{m} \int |\hat{v}|^2 \cdot |\hat{\varphi}_k|^2 d\omega = \int |\hat{v}|^2 \cdot \sum_{k=1}^{m} |\hat{\varphi}_k|^2 d\omega.$$

But if we assume for now that (and plan to show later)

$$A^2 \leq \sum_{k=1}^{m} |\hat{\varphi}_k(\omega)|^2 \leq B^2,$$

where $0 \leq A \leq B \leq \infty$, then $\forall v \in \mathbb{R}^2$,

$$\int |\hat{v}|^2 A^2 d\omega \leq \int |\hat{v}|^2 \sum_{k=0}^{m} |\hat{\varphi}_k|^2 d\omega \leq \int |\hat{v}|^2 B^2 d\omega$$

by susbstitution. This implies

$$A^2||v||^2 \leq \sum_{k=1}^{m} |\langle v, \varphi_k \rangle|^2 \leq B^2||v||^2,$$

and implies that $\{\varphi_k\}$ is a frame.
It remains to show that such bounds A and B exist for $\sum_{k=1}^{m} |\hat{\varphi}_k(\omega)|^2$. However, $\forall \omega$, there are two cases:

- ω is not within a transition area
- ω is within a transition area

It is easy to see that if ω is not within a transition area, $\varphi_k(\omega)$ is nonzero for only one k and we have $\sum_{k=1}^{m} |\hat{\varphi}_k|^2 = 1$ by definition. It remains to show the bounds if ω is within a transition area.

The lower bound is decided when the transition area is defined between two regions. In this situation, however, we have

$$\hat{\varphi}_1^2(\omega) + \hat{\varphi}_2^2(\omega) = \left[\cos\left(\frac{\pi}{2}\beta(\frac{\tau - D_1(\omega)}{2\tau})\right)\right]^2 + \left[\cos\left(\frac{\pi}{2}\beta(\frac{\tau - D_2(\omega)}{2\tau})\right)\right]^2$$

But, by the definition of the distance transform $D_2(\omega) = -D_1(\omega)$ and therefore

$$\hat{\varphi}_1^2(\omega) + \hat{\varphi}_2^2(\omega) = \left[\cos\left(\frac{\pi}{2}\beta(\frac{\tau - D_1(\omega)}{2\tau})\right)\right]^2 + \left[\cos\left(\frac{\pi}{2}\beta(\frac{\tau + D_1(\omega)}{2\tau})\right)\right]^2.$$

Since $\frac{\tau+D_1(\omega)}{2\tau} + \frac{\tau-D_1(\omega)}{2\tau} = 1$ and $\beta(x) + \beta(1-x) = 1$,

$$\hat{\varphi}_1^2(\omega) + \hat{\varphi}_2^2(\omega) = \left[\cos\left(\frac{\pi}{2}\beta(\frac{\tau - D_1(\omega)}{2\tau})\right)\right]^2 + \left[\cos\left(\frac{\pi}{2}(1 - \beta(\frac{\tau - D_1(\omega)}{2\tau}))\right)\right]^2,$$

and since $\cos(x) = \sin(x + \frac{\pi}{2})$, this is equivalent to

$$\hat{\varphi}_1^2(\omega) + \hat{\varphi}_2^2(\omega) = \left[\cos\left(\frac{\pi}{2}\beta(\frac{\tau - D_1(\omega)}{2\tau})\right)\right]^2 + \left[\sin\left(\frac{\pi}{2}\beta(\frac{\tau - D_1(\omega)}{2\tau})\right)\right]^2 = 1.$$

Therefore, our lower bound is $A^2 = 1$.

In the situation the transition area is between r regions, we simply have $\hat{\varphi}_1^2(\omega) + \hat{\varphi}_2^2(\omega) + \hat{\varphi}_3^2(\omega) + \cdots + \hat{\varphi}_r^2 = 1 + \hat{\varphi}_3^2(\omega) + \cdots + \hat{\varphi}_r^2(\omega)$. Since $\forall k = 1, 2, \ldots, m$, $\forall \omega$, $\hat{\varphi}_k(\omega) \leq 1$, we therefore get an upper bound of $B^2 = r - 1$.

As such, $\{\varphi_k\}$ forms a frame. □

With our empirical watershed wavelets defined, we can define the empirical watershed wavelet transform.

$$\mathcal{W}_f^{\epsilon}(k,t) = \langle f, \varphi_k \rangle = \int f(\tau)\overline{\varphi_k(\tau - t)}d\tau = \left(\hat{f}(\omega)\overline{\hat{\varphi}_k(\omega)}\right)^{\vee} \qquad (3.4)$$

As for the reconstruction formula, we define our dual frame as

$$\hat{\tilde{\varphi}}_k = \frac{\hat{\varphi}_k}{\sum\limits_k |\hat{\varphi}_k|^2},$$

so that we can define our reconstruction as

$$f(t) = \sum_{K=1}^{m_P} \int \mathcal{W}_k^{\epsilon}(\tau) \tilde{\varphi}_k(\tau - t) \tau = \Big(\sum_{K=1}^{m_P} \hat{\mathcal{W}}_f^{\epsilon}(k, \omega) \hat{\tilde{\varphi}}_k(\omega) \Big)^{\vee}. \tag{3.5}$$

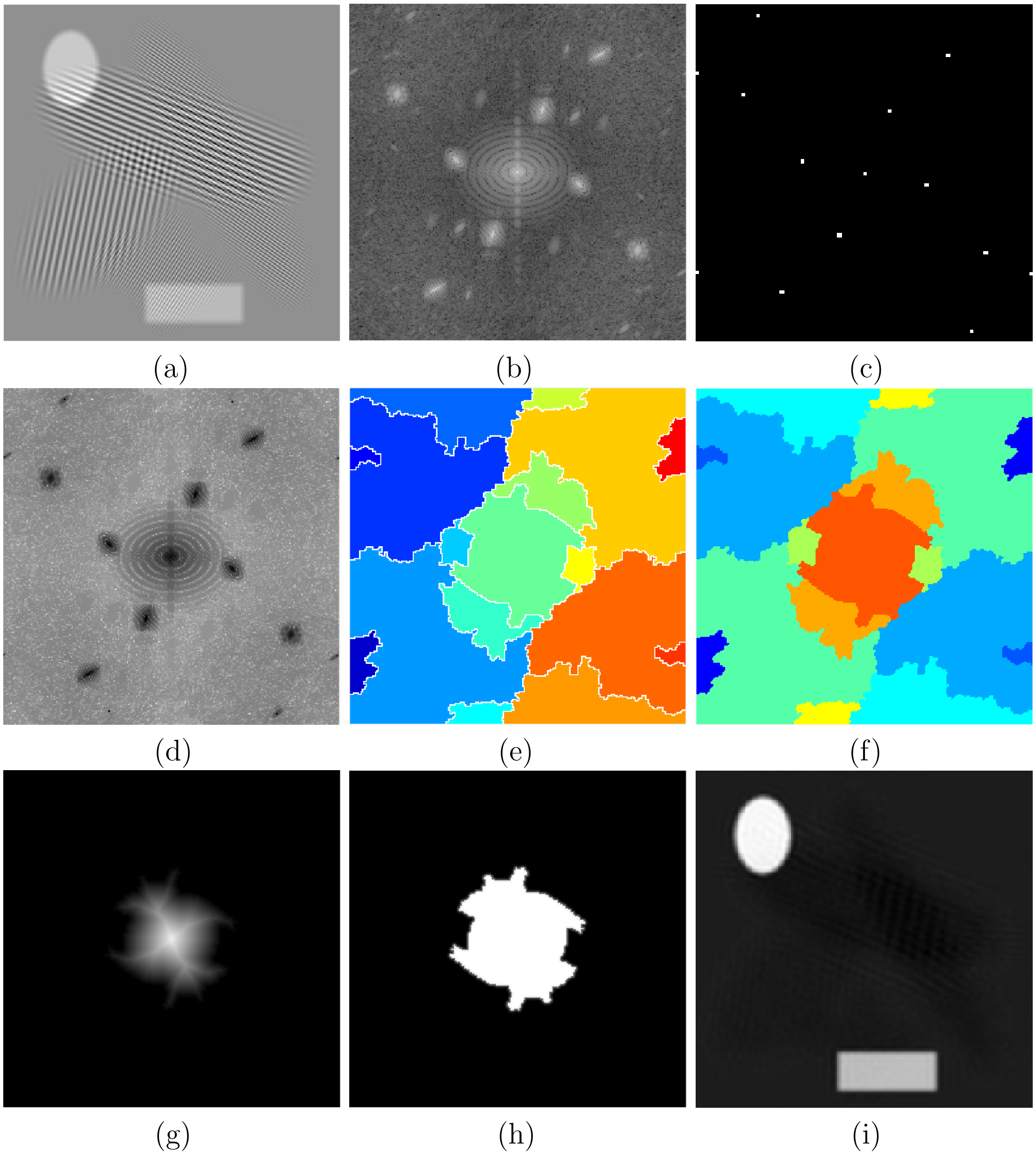


Figure 3.2. Algorithm for EWWT - (a) Original Image, (b) Magnitude Fourier Spectrum, (c) Detected Persistent Maxima, (d) Inverted Magnitude Spectrum with Imposed Minima, (e) Watershed Transform, (f) Paired Regions, (g) Distance Transform of Central Region, (h) Empirical Watershed Wavelet Filter, (i) Approximation Empirical Watershed Wavelet

CHAPTER 4

EXPERIMENTS

With the empirical watershed wavelet transform defined, it remains to see how it performs for mode decomposition. We shall compare it to previous 2D implementations of the EWT, but before discussing the comparison, we review the previous 2D empirical wavelet transforms.

4.1 Review of 2D Empirical Wavelet Transforms

As mentioned previously, there are many 2D implementations of empirical wavelet transform that have been suggested. We begin by the ones proposed by Gilles in [14], which reduce the boundary selection process to a 1D problem before constructing its filter bank.

4.1.1 Separable Methods

The first method proposed by Gilles [14] is the empirical tensor wavelet transform. We construct the empirical tensor wavelet transform by taking the image's magnitude spectrum, and performing the 1D EWT on the averaged rows and columns and then create the 2D filter bank as the product of those two.

Another useful separable method is the empirical curvelet transform, which instead tries to take advantage of the symmetry in the Fourier spectrum by segmenting the Fourier transform into angular sectors and scales. It does so by detecting the boundaries on rows and columns of the pseudo-polar Fourier transform's magnitude and building a filter bank from those boundaries. There are three options for this transform:

- Option 1: Detect angles and scales independently,
- Option 2: Detect scales, and then for each scale, detect angles separately,
- Option 3: Detect angles, and then for each angular sector, detect scales separately,

4.1.2 Voronoi EWT

The Voronoi empirical wavelet transform is the first non-separated two dimensional approach to the EWT. It is constructed similarly to the EWWT, with a few exceptions. After detecting the persistent maxima, Ames [1] constructs a voronoi tiling to define the boundary. This is done by finding the lines between the persistent maxima and then defining boundary lines that go through the midpoint of these lines and are orthogonal. Their intersections create a tiling and this tiling creates regions, which Ames uses to construct a Voronoi empirical wavelet in a similar way to the EWWT.

4.2 Comparative Results

We compare the resulting decomposition using the simulated image shown on the top left of Figure 4.1 (a). This image is a good example, as we have a clearly defined low frequency coefficient, a couple of medium frequency ellipses of differing orientations, and a couple of high frequency shapes of differing orientations. On top of this, these five modes intersect at points in the image. When looking at the Fourier spectrum of this image,in Figure 4.1 (b), we see that these modes are indeed represented at higher intensities and are fairly well separated.

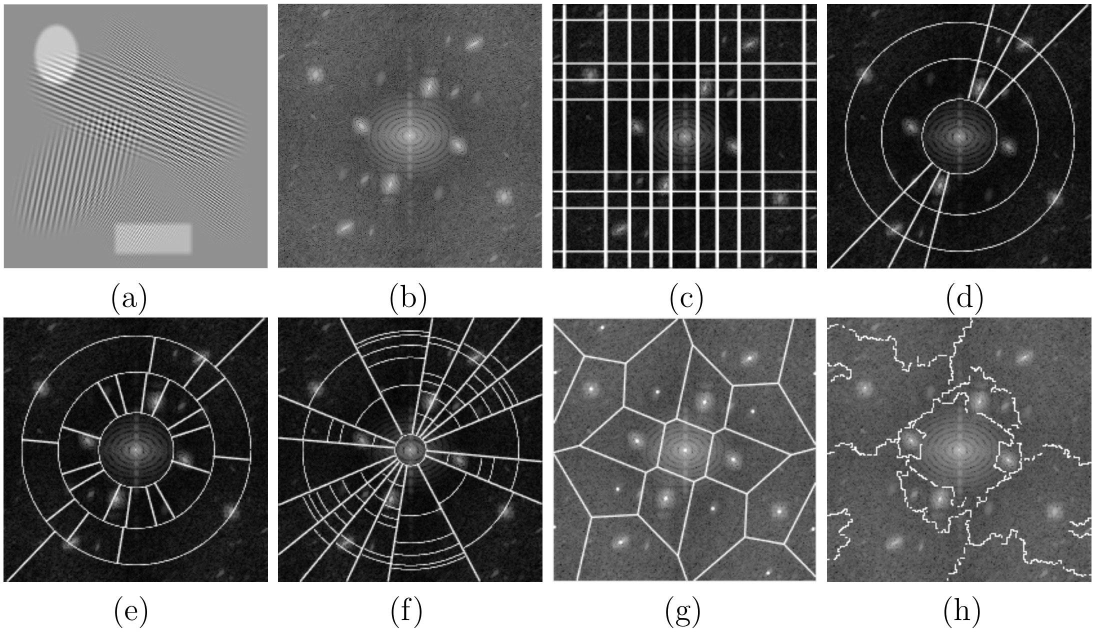


Figure 4.1. (a) Simulated image, (b) Fourier spectrum of image, (c) Boundaries of Empirical Tensor Wavelet, (d-f) Empirical Curvelets Options I, II, and III, (g) Voronoi, and (h) EWWT

We now compare the effectiveness of all six empirical wavelet transforms in separating these modes. In each case, we look to automatically detect these modes and the boundaries to separate them using scale-space methods with Otsu. The resulting boundaries for each discussed method can be seen in Figure 4.1. What we notice is that for all of these transforms, we detect more modes than are present in the signal. This is most egregious with the Tensor EWT, which decomposes our image into 30 modes and the Curvelet-3 EWT, which separates the signal into 45 modes. This is also true of the Curvelet-2 EWT, which defines 13 regions. Curvelet-1 EWT, Voronoi, and EWWT each detect only 10, which while still too many, is greatly reduced.

We also note in Figure 4.1 that the Voronoi and the Curvelet-1 EWT are bisecting some of the modes present in the signal, most notable the low frequency mode, and as we see in their empirical wavelet coefficients, this problem leads to artifacts. The result of the filtering for these latter three transforms is worth examining, and their empirical wavelet coefficients are shown alongside their corresponding wavelet filters in Figures 4.2, 4.3, and 4.4.

Figure 4.2 shows the results of curvelet-1 empirical wavelet decomposition. It does a fairly good separation of modes. However, as a result of some boundaries cutting off the edges of some modes, we see the same mode represented in multiple empirical wavelet coefficients. For example, coefficients shown in Figure 4.2 f,h,i,j all contain information regarding one of the higher frequency modes.

Meanwhile, when we look at the results of the Voronoi EWT in Figure 4.3, we see different, more concerning artifacts. While the modes present in the image are better segmented into different empirical wavelet coefficients, artifacts from the lower frequency mode are present in many of the coefficients (Figure 4.3 b,c,d) and are quite a bit more pronounced. This makes sense when looking at the decomposition, as much of that mode is bisected and included in other regions.

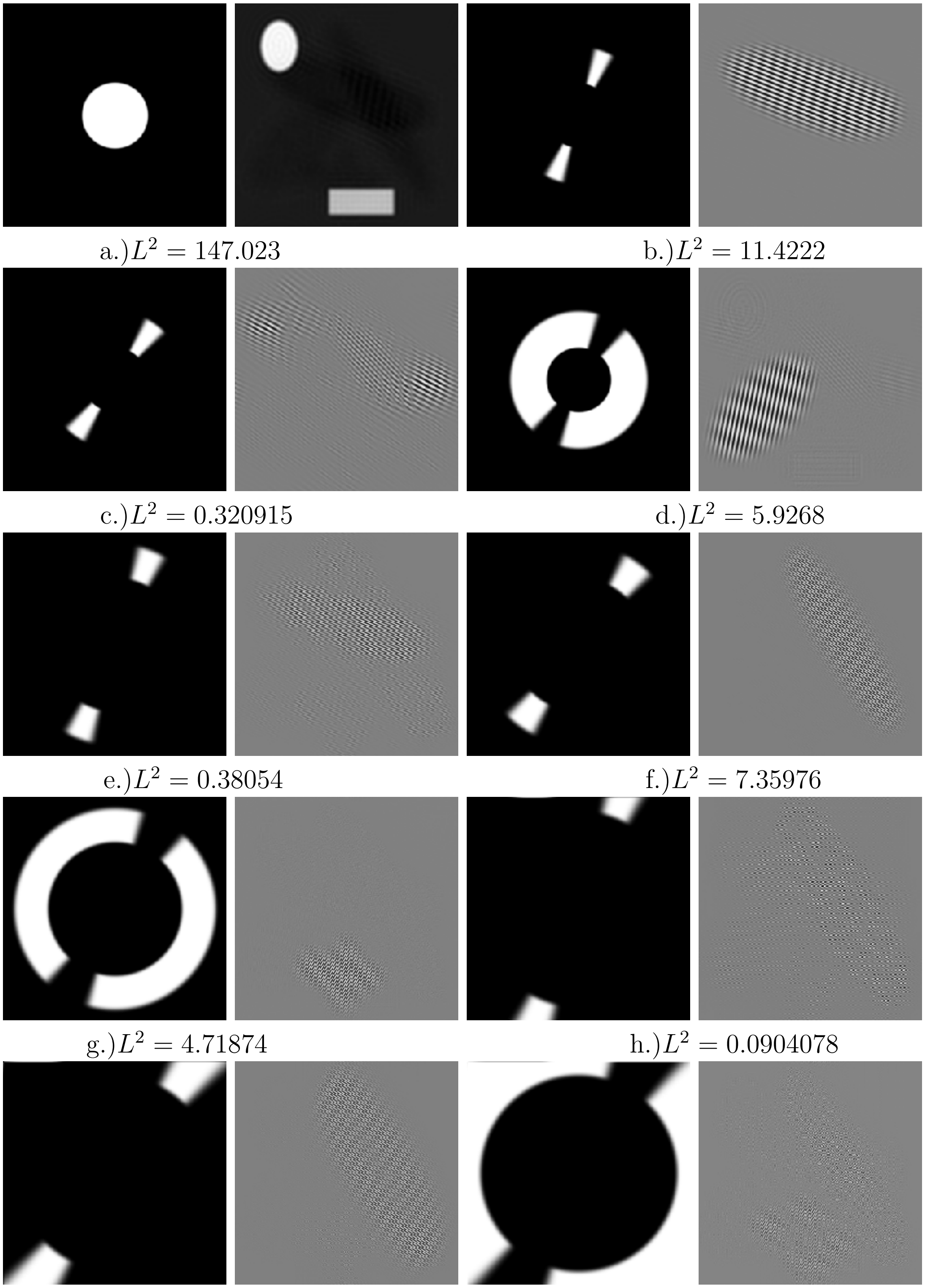
Lastly, Figure 4.4 shows the EWWT's decomposition. In this decomposition, we see a lot fewer artifacts. Some artifacts from Gibbs phenomenon appear, but we also see very clear separation of modes. In addition, we have the coefficients shown in Figure 4.4 c,i which display the intersection between the modes. The L^2 energy of each mode is also an order of magnitude separate from the coefficients which contain no modes.

4.3 Advantages of 2D Mode Detection

While the Curvelet-1 option performs quite well, it is worth considering that by detecting modes with a 2D approach, the Voronoi and EWWT both have multiple advantages over the seperable methods, which detect their boundaries using multiple 1D detection methods. For example, one could input a number of modes to detect. This would be hard to handle in the separable methods, as the number of modes is the product of two boundary types. However, the non-separable methods discussed can resolve this easily by taking the N most persistent pairs of maxima, where N is the desired modes.

Also, one could choose to remove a mode. Doing so in Curvelet and Tensor is possible by removing a boundary, but the original boundaries are not redrawn. Meanwhile, with Voronoi and EWWT one simply removes a maximum, and the boundaries can be redrawn from the new set of maxima.

Using these ideas, Figure 4.5 shows the empirical watershed wavelet decomposition of our signal if we deliberately choose to only keep the maxima that correspond to the 5 modes (Maxima corresponding to the filters a,b,d,f,g, in Figure 4.4).



a.)$L^2 = 147.023$

b.)$L^2 = 11.4222$

c.)$L^2 = 0.320915$

d.)$L^2 = 5.9268$

e.)$L^2 = 0.38054$

f.)$L^2 = 7.35976$

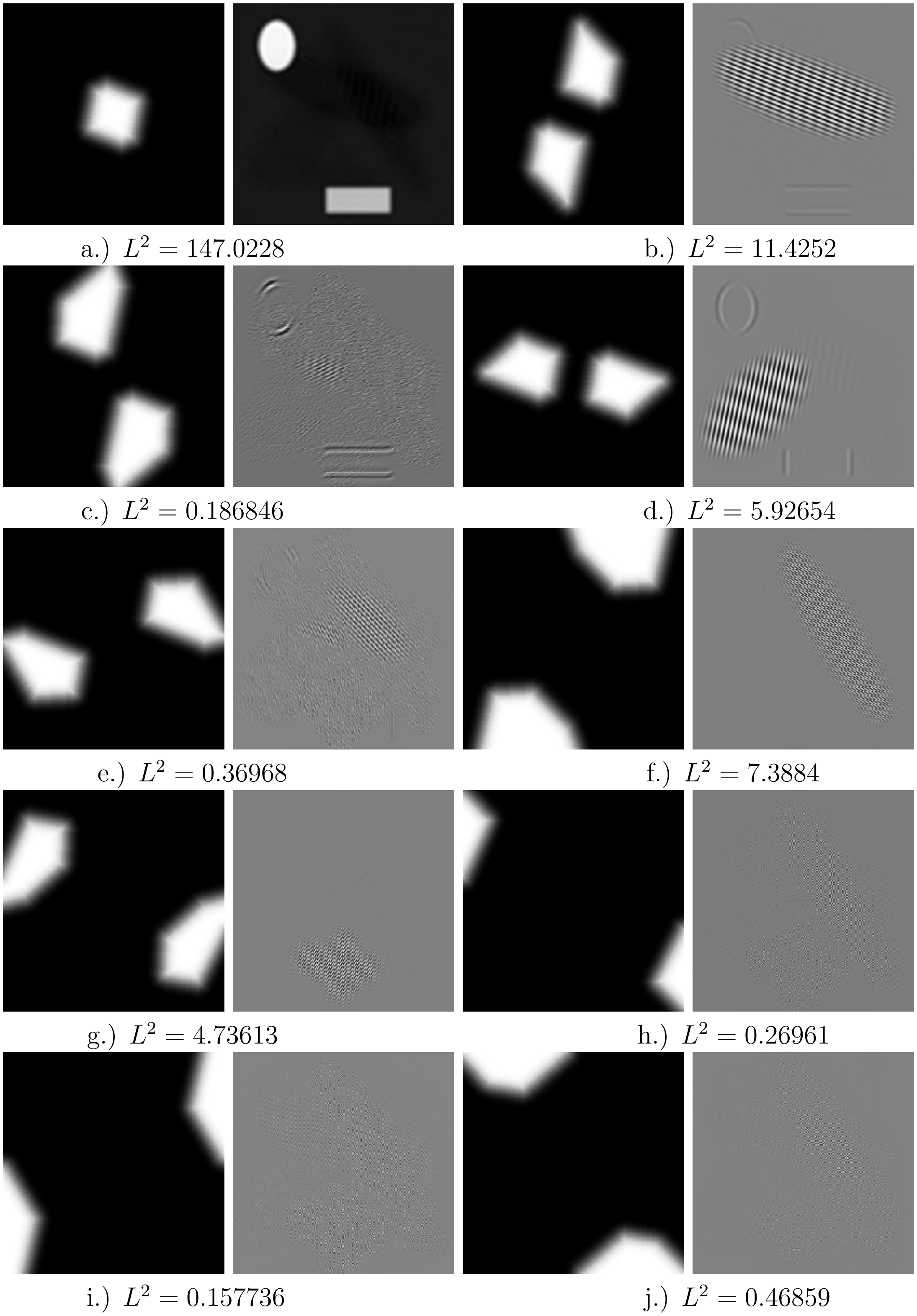
g.)$L^2 = 4.71874$

h.)$L^2 = 0.0904078$

i.)$L^2 = 1.3394$

j.)$L^2 = 0.59255$

Figure 4.2. Curvelet-1 EWT



a.) $L^2 = 147.0228$

b.) $L^2 = 11.4252$

c.) $L^2 = 0.186846$

d.) $L^2 = 5.92654$

e.) $L^2 = 0.36968$

f.) $L^2 = 7.3884$

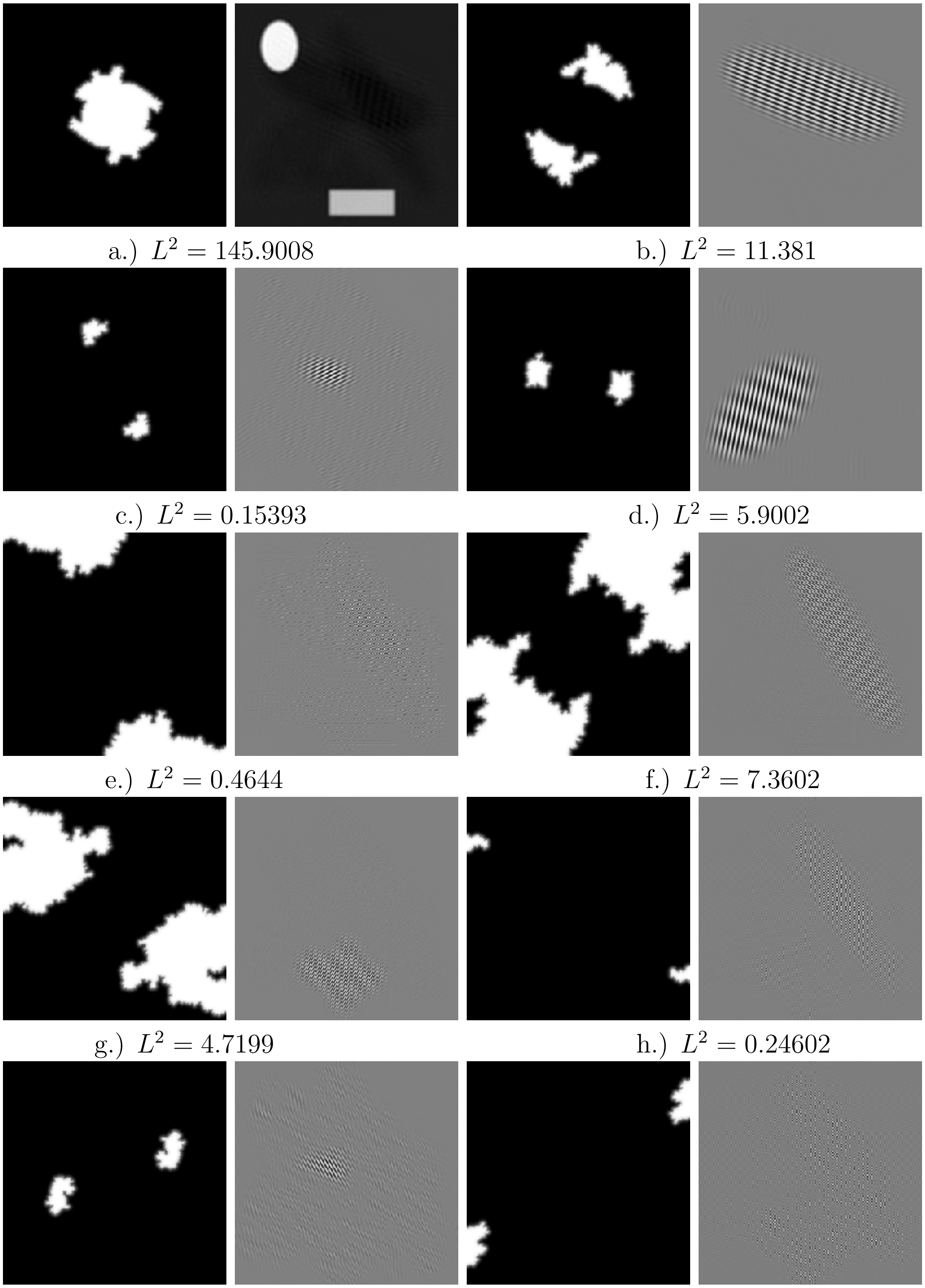
g.) $L^2 = 4.73613$

h.) $L^2 = 0.26961$

i.) $L^2 = 0.157736$

j.) $L^2 = 0.46859$

Figure 4.3. Voronoi EWT



a.) $L^2 = 145.9008$

b.) $L^2 = 11.381$

c.) $L^2 = 0.15393$

d.) $L^2 = 5.9002$

e.) $L^2 = 0.4644$

f.) $L^2 = 7.3602$

g.) $L^2 = 4.7199$

h.) $L^2 = 0.24602$

i.) $L^2 = 0.15393$

j.) $L^2 = 0.14717$

Figure 4.4. EWWT

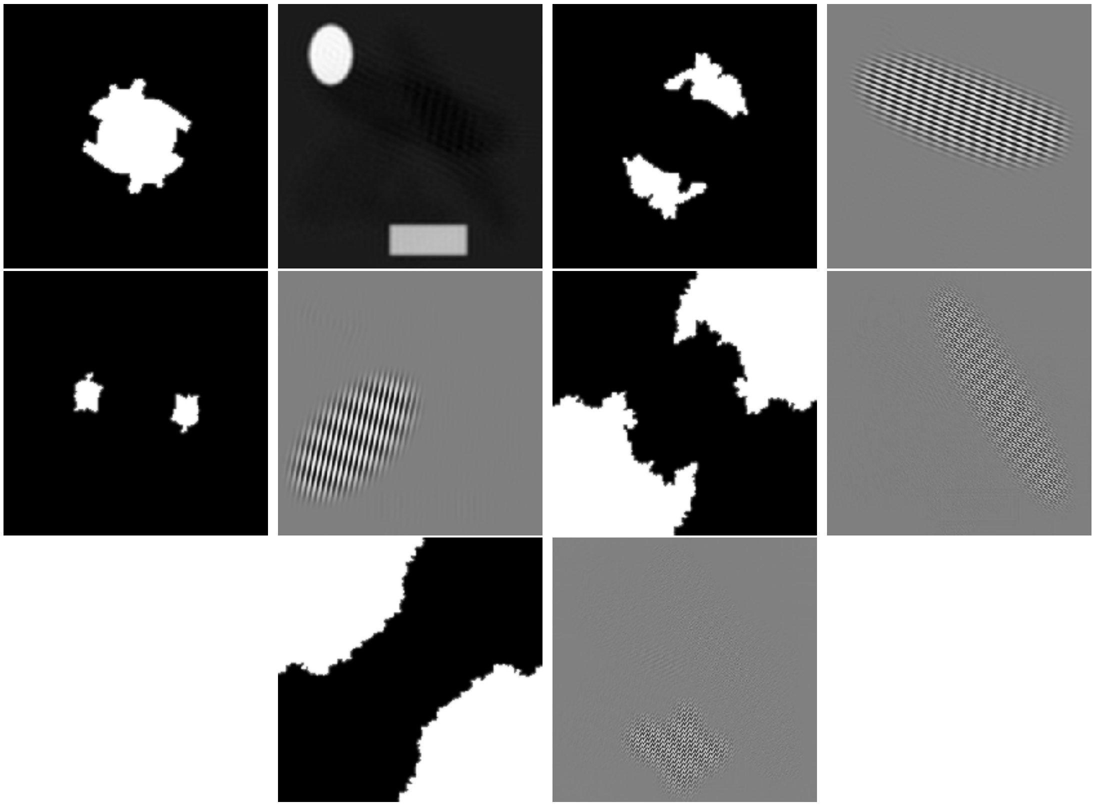


Figure 4.5. Empirical Watershed Wavelet Transform Of Signal With Selected Maxima

CHAPTER 5
APPLICATIONS

To see the effectiveness of the EWWT, we consider an application in unsupervised texture segmentation. Unsupervised texture segmentation is a computer vision problem in which we wish to do a pixelwise classification of textures. This is challenging, as textures are by their very nature nonlocal and have a lot of variety in their pixelwise intensities. Instead, textures are often categorized by their combination of frequencies and orientations. As such, many approaches have looked to construct feature vectors based on the influence of AM-FM modes. Y. Huang et al. showed in their recent papers [17] and [18] that the use of empirical wavelet transforms, along with some postprocessing, make excellent features for texture segmentation. As such, it is worth looking into the effectiveness of the EWWT as a feature for texture segmentation. In this paper, we focus specifically on the unsupervised case, where we are given no apriori information on the images or present textures.

5.1 Algorithm

While wavelet features have seen success in unsupervised texture segmentation, they are not usable on their own and must be post-processed to be effective features. Given a wavelet transform $\tilde{f}(x) : \omega \to \mathbb{R}^K$, we define a post-processing operator $\xi(\tilde{f})_k(x) = W(E[\tilde{f}(x)]_k$, where W is a sliding window filter of a given size centered at x. The operator E is then defined to be either energy $E(u) = (u_1^2, u_2^2, \dots, u_K^2)$ or entropy $E(u) = \big[- u_1 \log(u_1), -u_2 \log(u_2), \dots, -u_k \log(u_K)\big]$, where $u \in \mathbb{R}^K$.

We also choose to pre-process the images. In order to extract strictly textural information, we first perform a cartoon-texture decomposition, and choose to only perform the EWWT on the texture component of the image. We do this by using a method based on the Rudin-Osher-Fatemi (ROF) [30][5] model.

The ROF model minimizes the total variation based functional to remove the details of the image and create a cartoon version of the image. Meyer [27] then showed that the difference between the input image and its cartoon version does not contain all the details, and therefore he introduced a space of oscilatting functions, denoted G, to model these textures and he modified the ROF model to account for this. If $\mathcal{I}$ is our input image and $\hat{u}$ and $\hat{v}$ denote the cartoon and texture components, respectively, then the cartoon texture decomposition model consists in solving

$$(\hat{u}, \hat{v}) = \underset{u \in BV, v \in G}{\operatorname{argmin}} ||u||_{TV} + \mu ||v||_G, \text{subject to } \mathcal{I} = u + v, \tag{5.1}$$

where the space BV is the space of functions of bounded variation and is characterized by the semi-norm $||u||_{TV} = ||\nabla u||_1$. There are many discrete formulations of Equation 5.1, but we choose to use the formulation given by Aujol in [3],

$$(\hat{u}, \hat{v}) = \underset{u \in BV, v \in G_\mu}{\operatorname{argmin}} ||u||_{TV} + J^*\left(\frac{v}{\mu}\right) + \frac{\lambda}{2}||\mathcal{I} - (u + v)||_2^2, \tag{5.2}$$

where $G_\mu = \{v \in G \text{ s.t. } ||v||_G \leq \mu\}$ and J^* is the characteristic funtion over G_1. This formulation has the advantage of being relatively simple to solve numerically. In this paper, we use the algoirhtm described in [13] to preprocess all our images. As there has been limited literature on the choice of parameters μ and λ, we opt to have $\mu = \frac{\mathcal{N}}{2}$ (where $\mathcal{N}$ is the image size), and $\lambda = \frac{\omega_1^r}{2}$, where ω_1^r is the first radius detected using empirical curvelet as described in Section 4.1.1. An example of this decomposition can be seen Figure 5.1.

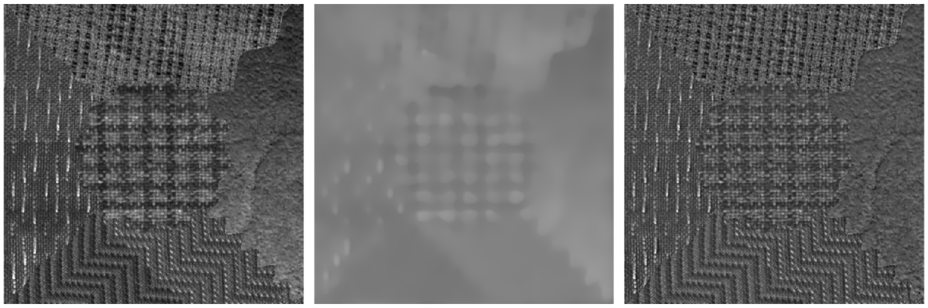


Figure 5.1. An Image And Its Cartoon And Texture Components

Once we have the empirical watershed wavelet features extracted from the texture component of the image, we perform a pixel-wise clustering in order to obtain the final segmentation. This paper focuses on the use of two algorithms. The first is the classic k-means algorithm [15] , which is a centroid-based algorithm that is widely used in literature. The second we shall consider is the Nystrom method [10][31], which is a more recent clustering algorithm that relies on the computation of eigenvectors of the graph Laplacian of an affinity matrix . Both these clustering algorithms require the input of the number of expected classes, and so we will provide this to each algorithm.

5.2 Parameters

Here, we present the parameters we will consider when testing the effectiveness of the EWWT for clustering. We will compare the EWWT with the Curvelet-1 empirical wavelet, as Y. Huang et al. showed in [17] that, of the previous EWT implementations, it performs the best on average. We will look at multiple options for our post-processing and clustering algorithm. For feature post-processing, we consider both energy and entropy. Since these are calculated on a window, we also consider the effects of the window size. We consider both k-means and Nystrom clustering and for our clustering metric, we consider multiple distances: cityblock, Euclidean, cosine, and correlation. We will also look at the effect that taking the log of the Fourier spectrum has on the clustering accuracy.

We will also be looking at many parameters for the EWWT to see their effects on the classification. We will look at the effect that our choice of transition width τ has. Since we saw in Section 4.2 that the EWWT's mode detection sometimes detects higher frequency modes that are not significant, we consider the effect of ignoring all detected maxima near the edges of the image.

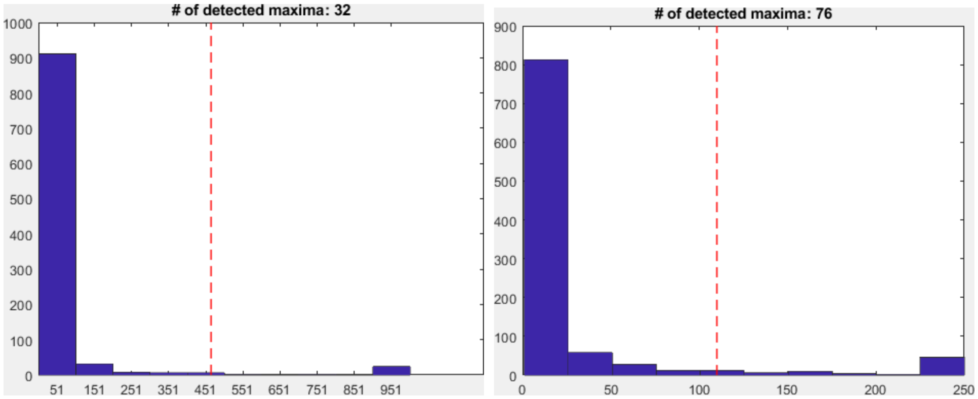


Figure 5.2. Detected Maxima From Persistence Histogram Using Full Scale-Space And Using Truncated Scale-Space Representation

We also want to consider the effect the threshold placement, which detects the final mode number. To do this, we simply end the fine-to-coarse representation early, which collapses all the maxima which persist beyond that point into the same histogram bar, as shown in Figure 5.2. By reducing the scale-space's step-size t_1 while keeping the total number of steps constant, we can emulate this scenario. As such, we explore the effects that modifying t_1 has on our segmentation results.

5.3 Experiments

In this section, we present the results of the algorithm described above. We look at each of the parameters described.

5.3.1 Dataset

To explore the efficacy of the EWWT with unsupervised texture segmentation, we use the Outex [28] dataset. This dataset comes with 100 composite images, which are generated by stitching together five textures from a database of 12 unique textures with multiple possible orientations. The size of each image is 512 × 512 pixels with 256 gray levels. There are five textures, and this information is given in to the algorithms.

5.3.2 Benchmarks

We use multiple benchmarks to judge the quality of the segmentation of each experiment. There are multiple ways to judge the partitioning of an image, and we focus on region-based interpretations, which are based on the regions made of segmented pixels.

Going forward, we consider gray-scale textured images. We will denote the original image as $\mathcal{I}$ and $\mathcal{N}^2$ will denote the number of pixels in the image. We assume that we have the ground-truth $\mathcal{P}_G = \{R'_1, \dots, R'_N\}$ of each image's segmentation for comparison and we label the obtained segmentation $\mathcal{P}_S = \{R_1, \dots, R_N\}$, where N is the number of clusters. Given a region R, $|R|$ denotes that region's cardinality. All the metrics used provide a number between 0 and 100%, where 100% represents a perfect segmentation. Note that all of these metrics are available in the Supervised Evaluation of Image Segmentation Methods (SEISM) toolbox via the *eval_segm* function (https://github.com/jponttuset/seism).

Normalized Variation of Information: Introduced in [25] , the variation of information measures the information difference between two clusterings. The author uses entropy $H(\mathcal{P}_S)$ and mutual information $I(\mathcal{P}_S, \mathcal{P}_G)$ of the segmentations to define the Variation of Information,

$$VoI(\mathcal{P}_S, \mathcal{P}_G) = H(\mathcal{P}_S) + H(\mathcal{P}_G) - 2I(\mathcal{P}_S, \mathcal{P}_G). \tag{5.3}$$

This quantity can then be normalized by $\log N$ to get the normalized Variation of Information (NVOI).

Swapped Directional Hamming Distance: This metric evaluates the similarity between regions from two segmentations by finding the region R with the maximum overlap for each region of the ground-truth segmentation, R'. The directional Hamming distance (DHD) [29] is defined as

$$D_H(\mathcal{P}_S \to \mathcal{P}_G) = \mathcal{N}^2 - \sum_{R' \in \mathcal{P}_G} \max_{R \in \mathcal{P}_S} |R' \cap R|. \tag{5.4}$$

and the swapped directional Hamming distance (SDHD) simply corresponds to $D_H(\mathcal{P}_G \to \mathcal{P}_S)$.

van Dongen Distance: The van Dogen (VD) distance [7] is simply a symmetric extension of the directional Hamming distance:

$$d_{vD}(\mathcal{P}_S, \mathcal{P}_G) = D_H(\mathcal{P}_S \to \mathcal{P}_G) + D_H(\mathcal{P}_G \to \mathcal{P}_S). \tag{5.5}$$

Swapped Segmentation Covering: The measure of the overlap between two regions, which can be used to assess pixelwise classification, is defined in [23] as

$$\mathcal{O}(R, R') = \frac{|R \cap R'|}{|R \cup R'|}. \tag{5.6}$$

In [2] , the author defines the covering of a segmentation $\mathcal{P}_S$ by a segmentation $\mathcal{P}_G$ as

$$C(\mathcal{P}_G \to \mathcal{P}_S) = \frac{1}{\mathcal{N}^2} \sum_{R \in \mathcal{P}_S} |R| \cdot \max_{R' \in \mathcal{P}_G} \mathcal{O}(R, R'). \tag{5.7}$$

The swapped segmentation covering (SSC) is defined by $C(\mathcal{P}_S \to \mathcal{P}_G)$.

Bipartite Graph Matching: The Bipartite Graph Matching (BGM) metric [19] [20] aims to use a bijective matching method to measure the maximum overlap between regions of two segmentation methods. The partitions $\mathcal{P}_S$ and $\mathcal{P}_G$ are seen as a common set of nodes, $\{R_1, \dots R_N\} \cup \{R'_1, \dots, R'_N\}$ on a graph and edges are inserted between each pair of nodes and assigned a weight. Given such a graph, the sum of the weights, denoted w, can be used to define the maximum-weight bipartite graph matching measure,

$$BGM(\mathcal{P}_S, \mathcal{P}_G) = 1 - \frac{w}{\mathcal{N}^2}. \tag{5.8}$$

Bidirectional Consistency Error: The last metric we will use was proposed in [24] , in which the author defined a measure that does not tolerate refinement to oversegmentation. This measure is bidirectional consistency error (BCE) and is defined as

$$BCE(\mathcal{P}_S, \mathcal{P}_G) = 1 - \frac{1}{\mathcal{N}} \sum_{R \in \mathcal{P}_S, R' \in \mathcal{P}_G} |R \cap R'| \min\Big(\frac{|R \cap R'|}{|R|}, \frac{|R \cap R'|}{|R'|}\Big). \tag{5.9}$$

5.3.3 Results

In this section, we present the results of the unsupervised texture segmentation using the Outex dataset. We explore all the parameters and options discussed in Section 5.2. To compare results, we consider the average of all six metrics discussed above.

Influence of Scale-Space Parameter t_1: We begin by looking into the effect that the choice of threshold for persistent maxima. As a reminder, this is done through prematurely stopping the scale-space representation. At the same time, we look at the application of the log of the spectrum before performing our fine-to-coarse representation and how that effects the results. To do this, we apply the k-means algorithm with the cityblock distance. We use energy as our post-processing method with a window size of 19. Also, for the EWWT parameters we have $\tau = 1$ and do not exclude edges.

The results, shown in Table 5.1 show two things. First, the application of the log does not improve results on average. Second, lower thresholds which detect more persistent maxima lead to better segmentation. We continue the testing using the Watershed without applying the log and the threshold $t_1 = 0.1$.

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
log	87.40	87.29	86.05	86.05	85.95	83.87	80.94	76.51	73.55	71.96
no log	**87.95**	87.30	86.09	86.09	85.32	0.85.17	83.54	80.24	76.46	73.44

Table 5.1. Influence of scale-space stepsize

Influence of Post-Processing Method and Window Size: Next we study the effects of the post-processing method. We apply the algorithm with the k-means clustering equipped with the cityblock distance. For the EWWT parameters, we have $\tau = 1$ and do not exclude edges. We also look at multiple window sizes in both cases. There are no theoretical results that dictate how this must be chosen, and often it depends on the textures present in the problem. As such, this is chosen empirically. The results on the Outex dataset is shown in Table 5.2. We see that the L^2 energy is a better post-processing method than the entropy, and that the ideal window is 19 pixels wide. We therefore continue our testing using the energy as our post-processing option and a window size of 19 pixels.

	Energy					Entropy				
	15	17	19	21	23	15	17	19	21	23
EWWT	87.14	87.25	**87.95**	87.69	87.88	84.71	85.63	85.34	84.84	85.97

Table 5.2. Influence of postprocessing method and window size

Influence of Clustering Method: We now look at our clustering method, and what we notice is from Table 5.3 is that kMeans outperforms the Nystrom clustering method. As such, we use kMeans clustering going forward.

Clustering Method	kMeans	Nystrom
	87.95	86.54

Table 5.3. Influence of clustering method

Influence of Clustering Distance: From there, we consider multiple distance metrics for the k-means clustering. Once again, we consider the transition width $\tau = 1$ and we do not exclude edges. For our distance metric, we consider the cityblock, euclidean, cosine, and correlation distances. Upon examination of the results shown in Table 5.4, we see that cityblock distance performs best, which could possibly be explained by the fact that the cityblock distance is less sensitive to outliers. Going forward, we use the cityblock distance metric.

Distance	Cityblock	Euclidean	Correlation	Cosine
EWWT	**87.95**	80.00	83.98	84.21

Table 5.4. Influence of clustering distance metric

Influence of Transition Width τ: We then consider which transition width gives the best results. This transition width is a balance, as too narrow a transition results in Gibbs phenomena in the wavelet coefficients, while too wide a transition leads to a mixing of modes. We see from the results in Table 5.5 that $\tau = 0.1$ performs best and as such, we use it going forward.

τ	0	0.05	.1	.2	.3	.5
EWWT	87.22	87.51	**87.95**	87.78	86.42	87.21

Table 5.5. Influence of trasition width τ

Influence of Excluding Edges: Lastly, we consider the influence of excluding maxima at the edges of the image. The motivation behind this is that the mode detection method of the EWWT sometimes detects modes that are at the edges of the image and do not contain relevant information, as seen in previous sections. When we exclude any persistent maxima that are close to the edge, we find from the results in Table 5.6 that excluding all maxima within 2 pixels of the images edge improves the clustering.

Pixels	0	1	2	3	5
EWWT	87.95	88.00	**88.37**	87.16	87.31

Table 5.6. Influence of edge exclusion

5.4 Final Comparisons

Having tested the effect that each parameter has on the segmentation, we now compare the empirical watershed wavelet transform to the Curvelet-1 EWT, which was shown by Huang et al. in [17] to be the best performer out of previous wavelet methods for the task of unsupervised texture segmentation. We use the best performer for each parameter, meaning that we use k-means with cityblock, a post-process with energy and a 19 × 19 pixel window. For the EWWT, the thresholding is made smaller by setting $t_1 = 0.1$, the transition width τ is set to $\tau = 0.1$, and the any persistent maximum within two pixels of the image edge are excluded.

Method	Curvelet-1	EWWT
Result	86.41 (7.09)	**88.37 (7.50)**

Table 5.7. Final comparative results

What we see in Table 5.7 is that the EWWT outperforms the Curvelet-1 on this dataset. This is consistent with some of the advantages that we discussed in Chapter 4. However, it is not without its drawbacks. The EWWT is a slower algorithm and constructs larger feature vectors than the EWT-Curvelet1. As such, both the feature detection and the pixel-wise classification steps take longer. However, the added flexibility of the EWWT, both in boundary shape and in mode detection, ultimately lead to a more accurate classification of textures.

CHAPTER 6
CONCLUSION

In this thesis, we looked to build a 2D empirical wavelet transform with fully adaptive boundaries based on the watershed transform of the Fourier spectrum. To do this, we used the 2D scale-space strategy proposed by Ames in [1] to detect what we define as persistent maxima, which give us the position of harmonic modes in the Fourier domain. We then used these persistent maxima as markers for a watershed transform, which defined catchment basins that formed regions. These regions are then used as supports to define a filter bank with transition areas similar in construction to the 1D empirical wavelet transform. We proved that this filter bank is a frame and defined the empirical watershed wavelet transform. We then explored the efficacy of the empirical watershed wavelet transform at separating harmonic modes and compared its results with previous implementations of the 2D empirical wavelet transform and highlighted some of the strengths offered by this method. We also explored an application of the EWWT as a crucial part of the feature extraction for an unsupervised texture segmentation problem, in which we used unsupervised machine learning techniques along with the empirical watershed wavelet transform and some data pre-processing to do a pixel-wise classification of textures. We showed that on the Outex dataset, the EWWT offers improved results over previous 2D empirical wavelet implementations.

In terms of future work, there are multiple avenues we can focus on. First, the EWWT has been shown to form a frame, but not a tight frame. Further work could be done to redefine the construction of the filters in such a way to guarantee a tight frame. Another direction that future work could take is in regards to harmonic mode detection. While the use of 2D scale-space theory to detect persistent maxima works, we have seen in this paper that it has flaws, and other approaches to mode detection could be studied. One consideration could be an entropy based approach similar to wavelet packets, as maxima can be easily removed and the boundaries can be redefined by re-applying the watershed transform on a new set of markers. More work is also needed in regards to the application of the EWWT to various machine learning problems. In terms of unsupervised texture segmentation alone, there is room for wider testing over multiple datasets to see if the results found in this thesis apply more broadly. There is also the examination of other texture related computer vision problems, such as the supervised case, or texture classification problems.

BIBLIOGRAPHY

[1] J. Ames, *2D Filtering Based on Voronoi Partitioning*, (2018).

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, *Contour Detection and Hierarchical Image Segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 33 (2011), pp. 898–918.

[3] J.-F. Aujol, G. Aubert, L. BLanc-Feraud, and A. Chambolle, *Image decomposition into a bounded variation component and an oscillating component*, Journal of Mathematical Imaging and Vision, 22 (2005), pp. 71–88.

[4] C. Beucher and S. Lantuejoul, *Use of Watershed In Contour Detection*, (1979).

[5] A. Chambolle, *An algorithm for total variation minimization and applications*, Journal of Mathematical Imaging and Vision, 20 (2004), pp. 89–97.

[6] R. R. Coifman and M. V. Wickerhauser, *Entropy-based algorithms for best basis selection*, IEEE Transactions on Information Theory, 38 (1992), pp. 713–718.

[7] S. Dongen, *Performance criteria for graph clustering and markov cluster experiments*, technical report, NLD, 2000.

[8] K. Dragomiretskiy and D. Zosso, *Variational mode decomposition*, IEEE Transactions on Signal Processing, 62 (2014), pp. 531–544.

[9] K. Dragomiretskiy and D. Zosso, *Two-Dimensional Variational Mode Decomposition*, (2015), pp. 197–208.

[10] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, *Spectral Grouping Using the Nyström Method*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 26 (2004), pp. 214–225.

[11] J. Gilles, *Empirical wavelet transform*, IEEE Transactions on Signal Processing, 61 (2013), pp. 3999–4010.

[12] J. Gilles and K. Heal, *A parameterless scale-space approach to find meaningful modes in histograms - application to image and spectrum segmentation*, International Journal of Wavelets, Multiresolution and Information Processing, 12 (2014), pp. 1450044–1–1450044–17.

[13] J. Gilles and S. Osher, *Bregman implementation of meyer's g-norm for cartoon + textures decomposition.* UCLA CAM Report 11-73, 2011.

[14] J. Gilles, G. Tran, and S. Osher, *2d empirical transforms. wavelets, ridgelets and curvelets revisited*, SIAM Journal on Imaging Sciences, 7 (2014), pp. 157–186.

[15] J. A. Hartigan and M. A. Wong, *Algorithm as 136: A k-means clustering algorithm*, Journal of the Royal Statistical Society. Series C (Applied Statistics), 28 (1979), pp. 100–108.

[16] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N. Yen, C. C. Tung, and H. H. Liu, *The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis*, Proc. Royal Soc. A: Math., Phys. Eng. Sci., 454 (1998), pp. 903–995.

[17] Y. Huang, V. De Bortoli, F. Zhou, and J. Gilles, *Review of wavelet-based unsupervised texture segmentation, advantage of adaptive wavelets*, IET Image Processing, 12 (2018), pp. 1626–1638.

[18] Y. Huang, F. Zhou, and J. Gilles, *Empirical curvelet based fully convolutional network for supervised texture image segmentation*, Neurocomputing, (2019).

[19] X. Jiang, C. Marti, C. Irniger, and H. Bunke, *Distance measures for image segmentation evaluation*, Eurasip Journal on Applied Signal Processing, (2006), pp. 1–10.

[20] S. Khuller and B. Raghavachari, *Advanced Combinatorial Algorithms*, in Algorithms and theory of computation Handbook, 1999, pp. 1–23, ch7.

[21] T. Linderberg, *Scale-Space Theory in Computer Vision*, Springer, 1994.

[22] S. P. Madhe, B. D. Patil, and R. S. Holambe, *Design of a frequency spectrum-based versatile two-dimensional arbitrary shape filter bank: application to contact lens detection*, Pattern Analysis and Applications, 23 (2020), pp. 45–58.

[23] T. Malisiewicz and A. A. Efros, *Improving spatial support for objects via multiple segmentations*, BMVC 2007 - Proceedings of the British Machine Vision Conference 2007, (2007).

[24] D. R. Martin, *An Empirical Approach to Grouping and Segmentation*, PhD thesis, EECS Department, University of California, Berkeley, Aug 2003.

[25] M. Meilă, *Comparing clusterings - An axiomatic view*, ICML 2005 - Proceedings of the 22nd International Conference on Machine Learning, (2005), pp. 577–584.

[26] F. Meyer, *Topographic Distance and Watershed Lines*, Signal Processing, 38 (1994), pp. 113–125.

[27] Y. Meyer, *Oscillating patterns in image processing and nonlinear evolution equations: the fifteenth Dean Jacqueline B. Lewis memorial lectures*, American Mathematical Society, 2001.

[28] T. Ojala, J. Viertola, S. Huovinen, M. Vision, and M. P. Unit, *Outex - New Framework for Empirical Evaluation of Texture Analysis Algorithms Timo Ojala , Topi Mäenpää , Matti Pietikäinen , Jaakko Viertola , Juha Kyllönen and Sami Huovinen*, Proc. 16th International Conference on Pattern Recognition, Quebec, Canada, (2002), pp. 701—-706.

[29] Qian Huang and B. Dom, *Quantitative methods of evaluating image segmentation*, in Proceedings., International Conference on Image Processing, vol. 3, 1995, pp. 53–56 vol.3.

[30] L. I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise*

removal algorithms, Physica D: Nonlinear Phenomena, 60 (1992), pp. 259–268.

[31] U. Von Luxburg, *A tutorial on spectral clustering*, Statistics and Computing, 17 (2007), pp. 395–416.