Robust Algorithms for Analysis of Traveling Wave Motions of the Tectorial Membrane

by

Roberto Daniel Filizzola Ortiz

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2020

Author . . . .
Department of Electrical Engineering and Computer Science
May 12, 2020

Certified by . . . .
Dennis M. Freeman
Professor of Electrical Engineering
Thesis Supervisor

Accepted by . . . .
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Robust Algorithms for Analysis of Traveling Wave Motions of the Tectorial Membrane

by

Roberto Daniel Filizzola Ortiz

Submitted to the Department of Electrical Engineering and Computer Science
on May 12, 2020, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

The tectorial membrane is a gelatinous matrix in the cochlea that is essential for the amplification and decomposition of sound in mammalian hearing. Material properties of the tectorial membrane have been studied in vitro by measuring the decay and speed of artificially generated waves of motion in excised cochlea.

Optical systems for measuring subnanometer motion are essential in these experiments. Analysis of data from these systems typically requires human intervention, which limits the objectivity and precision of the results. To overcome these limitations, this thesis focuses on the development of robust algorithms for analysing traveling wave motion data with minimal human intervention. First, we analyse a general purpose framework to estimate wave motions along a parametric path that is selected by a user. Although not fully automatic, this method is more flexible, faster, and less prone to error than previous methods. Second, we present a new gradient-based, fully automatic algorithm for estimating wave motions. Although it achieves high accuracy in synthetic data, systematic errors result when it is used to analyze images from some physiological experiments. Finally, we expand the traditional tectorial membrane model by including effects of wave reflection. This model improves the accuracy of wave estimates in experimental data and also provides convincing fits to data that were previously dismissed because the motions did not demonstrate monotonic decay. The new model demonstrates that the non-monotonicity is due to interference between the forward traveling wave and its reflection.

Thesis Supervisor: Dennis M. Freeman
Title: Professor of Electrical Engineering

Acknowledgments

I would like to acknowledge Prof. Dennis Freeman for all the support and guidance through all my years at MIT. He has been an incredible mentor through his patience and enthusiasm for teaching. I would also like to acknowledge Scott Page and Jon Sellon for all the help they provided me while working as a UROP and MEng student. Finally, I would like to acknowledge my family, especially my parents, for all their unconditional love and inspiring me to achieve my best version.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

1.1.1 Mammalian Cochlea

The mammalian cochlea is a remarkable sensor able to detect motions smaller than the diameter of a hydrogen atom and discriminate faint sounds from a large range of frequencies [4, 7].

When sound enters the cochlea, waves travel through the basilar membrane (BM) and are transduced into neural signals by sensory receptors, including both the inner and outer hair cells [4]. The inner hair cells (IHC) are primarily responsible for converting mechanical energy into electric signals that travel through the auditory nerve [3]. The outer hair cells (OHC) are thought to participate in the amplification of sound [7]. Both IHC and OHC are stimulated by motions of microscopic hair bundles. Tips of OHC bundles are embedded in a gelatinous matrix called the tectorial membrane (TM). IHC bundles are thought to be stimulated by fluid motions within the subtectorial space. These interactions induce a mechanical feedback loop between the TM, the outer hair cells and the BM that results in the amplification and decomposition of sound.

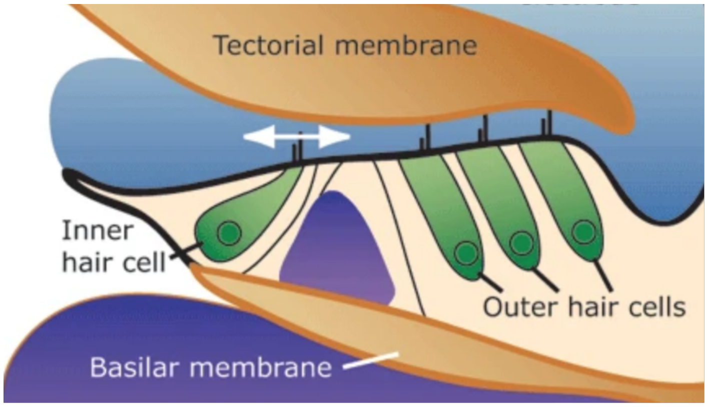


Figure 1-1: Diagram of cross-sectional view of the cochlea. Adapted from [9].

1.1.2 The Tectorial Membrane

Experimental results suggest the TM is a key component in the cochlear amplification of sound. Mouse models with genetically mutated TMs have exhibited loss of sensitivity in the BM [3, 4]. Due to its strategic location within the cochlea, it is believed the TM plays an important role in hair cell stimulation. To study this membrane structure and properties, both computational and signal processing techniques are required.

1.1.3 Isolated TM Preparation

Material properties of the TM have been studied using excised mammalian cochleae in isolation *in vitro* [4]. This is achieved by measuring the TM's response to a pure tone stimulus. The experimental setup consists of placing the TM from a mouse cochlea between two parallel-aligned supports and applying a sinusoidal motion to one support. This setup is illustrated in Figure 1-2. This creates longitudinal and radial sinusoidal waves that travel along the longitudinal direction of the TM and decay exponentially. These waves are analysed by measuring their decay constants and speeds [4, 7].

This method has been successfully used to study the role of different TM proteins. For instance, Ghaffari, et. al showed that the TM of Tecb$^{-/-}$ mice mutant, which

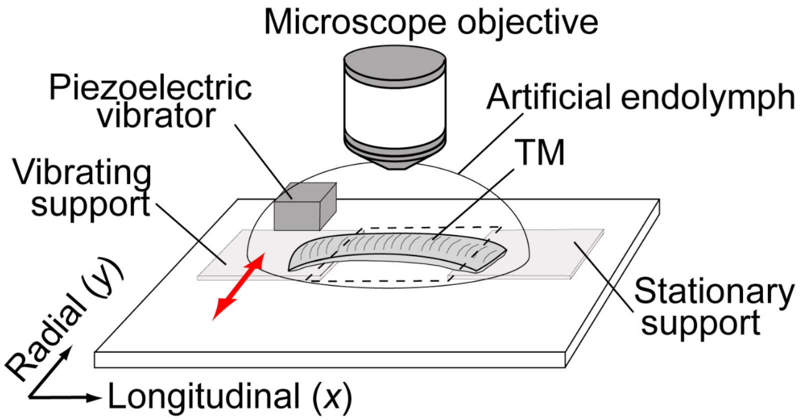


Figure 1-2: Diagram of experimental optical setup. Adapted from [7].

exhibit loss in cochlear sensitivity and sharped tuning, are significantly stiffer than the TM of wild type mice. [5]. Furthermore, this method has been utilized to study the TM properties under different frequencies. For example, Sellon, et. al demonstrated that wave speeds depend on stiffness at low frequencies and on viscosity at high frequencies. [10].

These use cases illustrate the flexible capabilities and importance of this experimental setup. For this reason, it is critical to develop robust algorithms to measure the wave decay and speed accurately.

1.2 Existing Optical Systems and Methods

To compute the decay constant and speed, the existing algorithms rely on a set of eight images of the TM, taken at evenly distributed phases. The displacements at each pixel and from frame to frame are then computed using optical flow methods. These displacements are fitted to radial and longitudinal sine waves that propagate longitudinally. The decay constants and wave speeds derive from the amplitude and phase of the best sinusoidal fit.

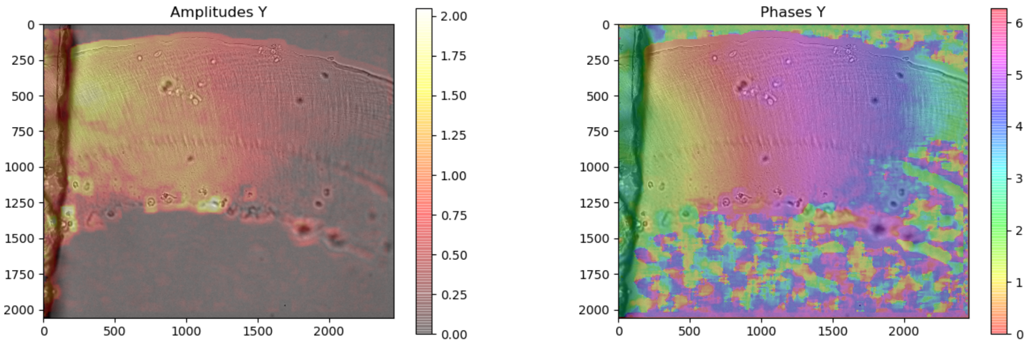


Figure 1-3: Output of the motion detection software. Amplitude and phase plots in the radial direction are shown respectively.

.

1.2.1 Motion Detection

The first step of the existing pipeline consists of the motion detection software. Given a set of eight TM images, this tool:

1. Computes the motion from frame to frame using the method developed by Timoner and Freeman [11]. This algorithm utilizes the constant brightness assumption [6] to compute the displacement of a region of interest with gradients in space and time. This algorithm is also optimized for sinusoidal motions.

2. For each pixel, the software tool computes the best sine fit with respect to the displacements computed at the previous step. From that, it computes the amplitude and phase in the x and y direction. The output plots of this software are illustrated in figure 1-3.

1.2.2 Estimating Wave Decay and Speed

In the ideal model of the TM, the induced radial wave motion at position x along the longitudinal axis, at time t and with angular frequency ω is defined by the following steady-state displacement formula:

$$D(x,\omega,t) = \Re[Be^{-\frac{x}{\sigma}}e^{-j\frac{x}{\lambda}}e^{j\omega t}]$$

where B is a complex constant and σ and λ are positive constants. The decay and speed are then defined as σ and $\omega\lambda$ respectively.

The amplitudes and phases output by the motion detection software are used to estimate σ and λ. The log of the amplitude decays linearly with a slope of $1/\sigma$, and the phase decreases linearly with a slope of $1/\lambda$.

1.2.3 Problems with Existing System

The existing procedure for estimating the wave decay and speed mostly relies on human intervention. In past experiments, the longitudinal trajectory was manually chosen by either specifying a horizontal line within the TM or manually selecting equally spaced region of interests. Specifying a horizontal line is not general enoguh since many TM samples are curved. Selecting region of interests is error-prone since these are arbitrarily chosen. Furthermore, this method covers fewer data points, resulting is less robust estimates. Overcoming these limitations requires a careful examination of the data at the cost of the researchers time.

1.3 Thesis Goals

The goal of this thesis is to develop robust algorithms for analysing wave motions traveling through the tectorial membrane that require minimum human intervention. These methods should allow researchers to spend more time analysing rather than collecting data.

First, we will present a general framework for analysing waves estimates through a parametric path. This framework is more general and more reliable than previous methods used in TM research. We will also compare two different methods for estimating wave decay and speed given amplitudes and phases in a defined longitudinal trajectory.

Second, we will propose a new fully automatic, gradient-based method for estimating wave motion properties. This algorithm does not require specifying a longitudinal trajectory, utilizing the entire TM image and motion.

Finally, we will analyse the effects of wave reflection in TM experiments. We will show how reflection can distort previous waves measurements and how incorporating it in our model of the TM achieves more robust wave estimates.

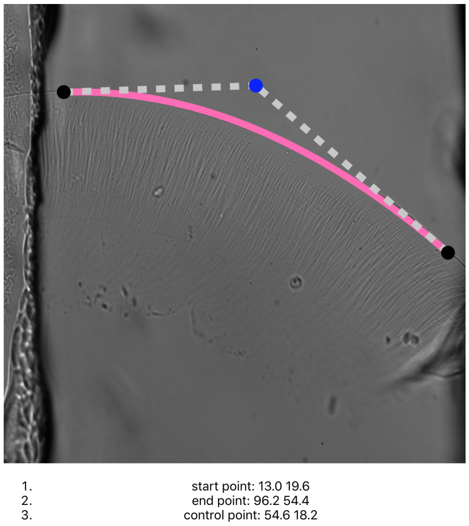
Chapter 2

Analyzing Wave Motions along a Parameterized Path

This chapter focuses on the development of a general purpose method to estimate wave motions from a sequence of video images, with special attention to ease of use, as well as accuracy and robustness. The method allows for parametric description of the trajectory of the wave, which is more flexible than previous straight line specifications while also being faster and less prone to error than manual selection of individual regions.

2.1 Parameterized Curved Selection

This section describes the framework for selecting a parameterized curve as the longitudinal axis to analyze wave motions. First, we introduce the concept of Bezier curves and the GUI built to manually select such curves in TM images. Second, we present a simple algorithm for sampling equally spaced points from a parameterized curve. Finally, we explain how to convert the sampled xy amplitudes and phases to obtain the longitudinal and radial displacements.



1. start point: 13.0 19.6
2. end point: 96.2 54.4
3. control point: 54.6 18.2

Figure 2-1: Snapshot of GUI

2.1.1 Bezier Curves

A parametric Bezier curve of degree n is a curve defined by $n + 1$ points $p_0, ..., p_n$, represented by the following parametric equation

$$p(t) = \sum_{k=0}^{n} \binom{n}{k} t^k (1 - t)^{n-k} p_k \text{ for } 0 \leq t \leq 1$$

In our experiments, we utilized a 2D quadratic Bezier curve, meaning that there are exactly three control points represented by xy coordinates.

2.1.2 GUI for Parameterized Curve Selection

After uploading the TM image, the user selects the three points that define the Bezier curve. As the three points are dragged by the user, the curve adjusts its position and shape. It also displays the relative position of each of the three control points. A snapshot of the GUI is found in figure 2-1. It was implemented in javascript using the React.js library.

2.1.3 Dividing the Curve Equally

After the user selected the curve, the next step is to divide such curve in equal parts. Although points in a Bezier curve have a convenient representation, the closed form is not as trivial. Thus, we opted for the following algorithm to approximates these points:

1. Compute $\hat{s}$, the estimated arc-length of the entire curve.
2. For $i = 0, ..., \lfloor \hat{s} \rfloor$:
    - Compute $\hat{t_i}$, the approximation of t_i such that the arc-length from $p(0)$ to $p(t_i)$ equals i.
3. Output $p(\hat{t_i})$ for every $i = 0, ..., \lfloor \hat{s} \rfloor$

Note that if step 1 and 2 output good approximations, the arc-length between $p(\hat{t_i})$ and $p(\hat{t_{i+1}})$ is close to 1. Next, we will describe step 1 and 2 of this algorithm in more detail.

Estimating arc-length

To approximate the arc-length of a curve, we first compute the intermediate points $p(i/k)$ for $i = 0, ..., k$ and for some predetermined constant k. Then, the estimated arc-length is

$$\sum_{i=0}^{k-1} \left\| p\left(\frac{i+1}{k}\right) - p\left(\frac{i}{k}\right) \right\|$$

where $\| \cdot \|$ is the l2-norm. Note that as k gets bigger, the estimates converge to the actual arc-length while the runtime increases. For our purposes, we set $k = 100$. This algorithm can also be extended to compute the arc-length between any two points alongside the curve.

Finding points in the curve

The second step involves taking as an input a target arc-length s and outputting t such that the arc-length between $p(0)$ and $p(t)$ is approximately s. This can be achieved by using binary search and the previous algorithm for estimating arc-lengths.

Here is a more detailed description. We first initialize two variables $t_{start} \leftarrow 0$ and $t_{end} \leftarrow 1$. Then, we proceed with a binary search with the following recursive algorithm. On input (t_{start}, t_{end}, s):

1. Define $t_{mid} \leftarrow \frac{t_{start}+t_{end}}{2}$. Estimate arc length $\hat{s}$ between t_{start} and t_{mid} using previously described algorithm.

2. If $|\hat{s} - s| < \epsilon$ for some small pre-determined ϵ, output t_{mid}.

3. If $\hat{s} > s + \epsilon$, recurse on (t_{start}, t_{mid}, s).

4. Otherwise, recurse on $(t_{mid}, t_{end}, s - \hat{s})$.

2.1.4 Obtaining Radial and Longitudinal Displacements

The final step involves computing the radial and longitudinal displacements. After dividing the curve in points $p_0, p_1, ..., p_m$, we proceed to sample m longitudinal and radial amplitude and phases. For every $i = 0, .., m-1$, we set the longitudinal unit vector as $\hat{l} = \frac{\vec{p}_{i+1}-\vec{p}_i}{\|\vec{p}_{i+1}-\vec{p}_i\|}$ and the radial unit vector $\hat{r}$ as the 90 degrees rotation of $\hat{l}$. Recall that the motion estimation algorithm already outputs the xy amplitudes A_x and A_y and xy phases ϕ_x and ϕ_y at each pixel. To obtain the new amplitudes and phases we just simply change the basis of the following vector

$$A_x e^{j\phi_x}\hat{\imath} + A_y e^{j\phi_y}\hat{\jmath}$$

and compute its magnitude and angle to obtain the equivalent vector

$$A_l e^{j\phi_l}\hat{l} + A_r e^{j\phi_r}\hat{r}$$

where A_l and ϕ_l are the longitudinal amplitude and phase and A_r and ϕ_r are the radial amplitude and phase.

2.2 Estimating Wave Decay and Speed

In this section we will describe two different methods for estimating wave decay and speed through a curve. The first method looks at the amplitudes and phases separately. The second method uses the complex overall steady-state displacements.

2.2.1 Method 1: Fitting Amplitudes and Phases

Estimating wave decay

Recall that the amplitudes of the steady-state displacements should decay exponentially alongside the longitudinal axis. To be more precise, let $A(x)$ be the amplitude of either the radial or longitudinal component at position x of the longitudinal axis. Then, in an ideal model:

$$A(x) \approx A_0 e^{\frac{-x}{\sigma}}$$

where A_0 is a constant and σ represents the decay. This is also equivalent to:

$$\ln A(x) \approx \ln A_0 - \frac{x}{\sigma}$$

which means that the log of the amplitudes decay linearly with a slope of $\frac{1}{\sigma}$. Thus, to estimate the decay we can simply apply a linear regression to the log of the amplitudes to obtain A_0 and σ.

Estimating wave speed

Recall that the phase should also change linearly. In other words, if $\phi(x) \in [0, 2\pi)$ is the phase at position x of either the longitudinal or radial direction alongside the longitudinal axis, then

$$e^{j\phi(x)} \approx e^{j(\phi_0 - \frac{jx}{\lambda})}$$

where λ represents the speed of the wave and ϕ_0 is a constant. Although the phase changes linearly, we cannot apply a normal linear regression due to phase wrapping around 0 and 2π. However, when $\phi(x)$ is not close to the boundaries, we have that:

$$\phi(x + \Delta x) - \phi(x) \approx -\frac{\Delta x}{\lambda}$$

Thus, the first differences $\Delta\phi(x) = \phi(x) - \phi(x-1)$ of the phases should be roughly equal with the exception of outliers near the boundaries. To eliminate these, we simply ignore any $\Delta\phi(x)$ above a standard deviation. After this step, the mean of the remaining $\Delta\phi(x)$ should give us a good approximation of $-\frac{1}{\lambda}$.

2.2.2 Method 2: Fitting Complex Displacements

The second method consists on fitting the complex representation of the displacement instead of the amplitude and phase separately. This can be done with a non-linear least squares regression, where the goal is to find $\sigma, \lambda, a_0, \phi_0$ that minimize:

$$\sum_x \left| A(x)e^{j\phi(x)} - e^{a_0 - \frac{x}{\sigma}} e^{j(\phi_0 - \frac{x}{\lambda})} \right|^2$$

For the non-linear least squares regression, we utilized scipy's implementation of Trust Region Reflective algorithm [1, 2, 8]. This algorithm requires an initial guess of the parameters to be estimated. These are calculated by using the method 1 of fitting amplitudes and phases separately.

2.3 Results

2.3.1 Synthetic Displacements

The first experiment consists of applying method 1 and 2 to the following function

$$f(x) = e^{-\frac{x}{\sigma} - j\frac{x}{\lambda}}$$

to estimate the values of the known σ and λ. We simulate both methods for 1000 trials, where the target σ an λ were 62.5 and 40 respectively. The values of x that

d^2	σ Met 1	σ Met 2	λ Met 1	λ Met 2
0.05	0.34 %	0.73%	3.28%	0.13 %
0.10	0.66%	1.38%	6.67 %	0.25 %
0.20	1.37%	2.74%	14.87 %	0.52 %

Table 2.1: Results of the first experiment, comparing the two different methods for estimating wave and decay with different noise levels. This table displays the mean absolute error with respect to the true values. Both methods achieve accurate σ estimates with errors below 3%. However, method 2 considerably outperform method 1 at estimating λ, with method 2 achieving almost perfect estimates and method 1 having substantial errors of around 15% for higher noise levels.

were used ranged from 0 to 200. To test for robustness, we multiply the real and imaginary parts by a normal random variable of mean 1 and variance d^2.

Table 2.1 includes the results of the experiment. For each method, we reported the ratio between the standard deviation of the estimates and the mean.

The results suggests that fitting to the complex displacements (method 2) is more robust than fitting the amplitudes and phases separately (method 1). Method 2 considerably outperforms method 1 in wave speed estimation. Although method 1 presents better wave decay estimates, method 2 still provides accurate competent estimates with errors below 3%.

Since method 2 is the better method overall, we will ignore method 1 for our following experiments.

2.3.2 Synthetic Images

For the second experiment, we generated eight synthetic images of a fake curved TM. A sample image is displayed in figure 2-2. The fake curved TM consists of a sequence of pixel columns whose brightness were modeled from a Gaussian distribution. To simulate a traveling wave motion, each column was vertically shifted with decaying amplitudes and phases. Although this motion is not perfectly radial, it still provides a good enough approximation. The parameters used were $\sigma = 1666.66$ and $\lambda = 181.18$. To test for robustness, we added both fixed pattern noise and shot noise.

The errors with respect to the true values were 3.07% for σ and 0.01% for λ. These

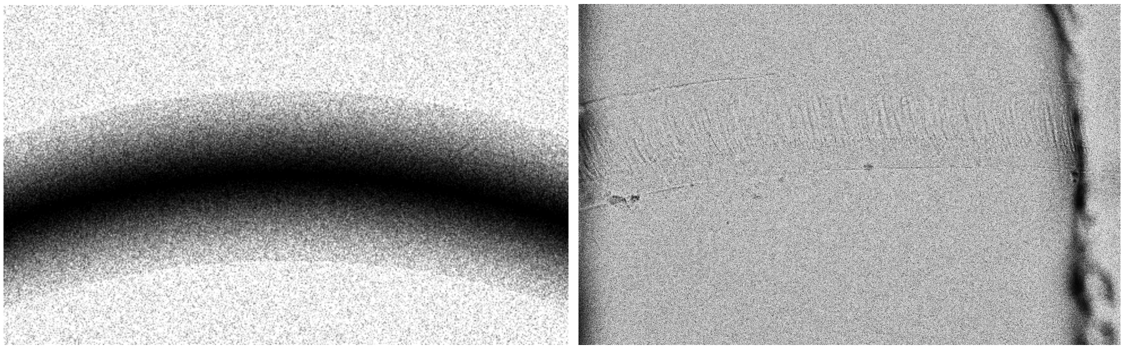


Figure 2-2: Images used in the second and third experiment. The left figure is a synthetic image replicating a curved TM. The right figure is a real TM image. Both images' columns were shifted artificially with sinosoidal motions. The left image contains shot noise and the right image fixed pattern noise. Despite the high noise levels, wave estimates were very precise in these sets of images.

results are encouraging, achieving an almost perfect estimate of λ. In addition, the error for σ remains low even when motions are not perfectly radial.

2.3.3 Real Images with Synthetic Displacements

For the third experiment, we tested our algorithm in a real TM image with synthetic displacements. This image is illustrated in figure 2-2. As in the previous experiment, each column was vertically shifted with decaying amplitudes and phases. The values of σ and λ were the same as in the previous experiment. We also added shot and fixed pattern noise to test robustness.

The results are very similar to the previous experiment, with an error of 2.55% for σ and a minor 0.25% for λ. This further suggest that this method accurately estimates the decay and speed.

2.3.4 Real Data

The last experiment consisted on a qualitatively analysis of our algorithms on real data. We used four different TM for our experiment and plotted the real part of the complex displacements versus the best fits. These plots can be found in figure 2-3. The estimate of λ looks pretty accurate in the four cases. However, σ is not a precise for two out of the four samples. This is consistent with our previous experiments where the measurements were more precise for λ than σ. In addition, in two of the samples the displacements do not exactly decay exponentially. We will address this artifact in the last chapter by bringing the concept of reflection waves.

2.4 Discussion

In this chapter we explored two different methods for estimating the decay and wave speed in TM experiments as well as a more user-friendly and generic framework for selecting a longitudinal axis. We obtained promising results by fitting to the complex displacements instead of looking at the amplitudes and phases separately. This method is also very precise at estimating the wave speed in both synthetic and real data. Although it is not as precise for the decay in both synthetic and real data, the results are still encouraging. Furthermore, real data presents the challenge that amplitudes are more noisy and do not exactly decay exponentially.

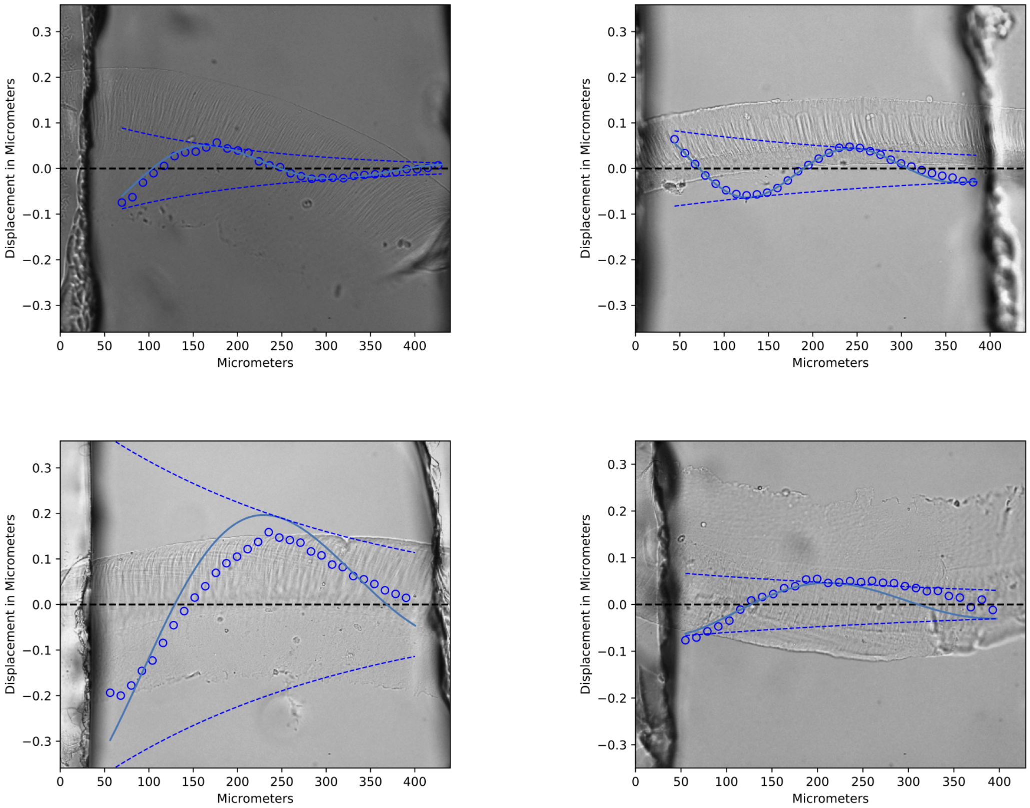


Figure 2-3: Real part of complex displacements versus best fits. The x axis represents the longitudinal axis and the y axis the displacements, both in micrometers. The dots represent the estimated displacements coming from the motion estimation algorithm while the blue solid curve represents the best fit based on our assumptions. The outer curve represents how the amplitudes decay exponentially based on the best fit of σ, the wave decay

.

Chapter 3

Automatic Gradient-Based Wave Motion Analysis

This chapter focuses on the development of an automatic method for estimating wave motions that uses the entirety of the TM image instead of a parametric path. This method first segments the TM by both brightness and motion and then robustly averages the local wave estimates to achieve a global wave decay and speed. Since this method requires minimal user inputs, it presents a more objective and robust procedure for wave motion estimation.

3.1 Gradient-Based Wave Estimation

This section describes the overall method for estimating decay and wave speed using motion gradients. For simplicity, assume that the longitudinal axis of the TM is a horizontal line and the radial axis is a vertical line. In addition, assume that the motion is purely radial. As in chapter 2, let $A(x, y)$ and $\phi(x, y)$ the amplitude and phase of the sinusoidal motion at position (x, y) in the TM. In our ideal model, the sinusoidal motion is described by:

$$U(x, y) = A(x, y)e^{j\phi(x,y)} \approx e^{a_0 - \frac{x}{\sigma}} e^{j(\phi_0 - \frac{x}{\lambda})}$$

by setting $B_0 = e^{a_0}e^{j\phi_0}$, we get the following equivalent expression:

$$U(x,y) \approx B_0 e^{-x(\frac{1}{\sigma}+\frac{j}{\lambda})}$$

Then, by taking the partial derivative in the x direction we obtain that:

$$\frac{\partial U(x,y)}{\partial x} = -B_0 \left(\frac{1}{\sigma} + \frac{j}{\lambda}\right) e^{-x(\frac{1}{\sigma}+\frac{j}{\lambda})} = -\left(\frac{1}{\sigma} + \frac{j}{\lambda}\right) U(x,y)$$

Thus, if $U_x = \frac{\partial U}{\partial x}$ we have that

$$\frac{U_x(x,y)}{U(x,y)} = -\left(\frac{1}{\sigma} + \frac{j}{\lambda}\right)$$

which is independent from x and y. This relation gives us a new tool for estimating σ and λ.

3.2 Gradient-Based Method on TM Experiments

This section presents a new algorithm for estimating the wave decay and speed in the TM experiments. It is based on the gradient method described in the previous section. Here is the high-level description of this method:

1. Estimate $U_x[x,y]$ for each pixel $p = (x,y)$ in the image.
2. Estimate local σ and λ for each pixel by computing $\frac{U_x(x,y)}{U(x,y)}$
3. Eliminate non-TM regions based on brightness and motion.
4. Robustly average local estimates of remaining pixels to obtain global decay and speed.

Next, we will explain each step in more detail.

3.2.1 Estimating Gradients

The traditional method for computing gradients in images is by convolution with linear filters. These are inspired by the continuous definition of derivatives. Recall that:

$$\frac{\partial U(x,y)}{\partial x} = \lim_{h \to 0} \frac{U(x+h,y) - U(x,y)}{h}$$

In the discrete case, we can approximate the partial derivative by setting

$$U_x[x,y] = U[x+1,y] - U[x,y]$$

where $U[x,y]$ is the discrete representation of the function U. In terms of convolution, this is equivalent to:

$$U_x[x,y] = U[x,y] * h_x[x,y]$$

where $h_x[x,y] = \delta[x+1] - \delta[x]$. However, this operation is very noise sensitive since we only use two pixels to estimate the derivatives at any point. In order to make a more robust gradient estimation, we can convolve U with a 2D Gaussian filter. In the continuous case, this is equivalent to

$$U_x(x,y) = \frac{\partial}{\partial x}(U(x,y) * G(x,y,\sigma))$$

where $G(x,y,\sigma) = g(x,\sigma)g(y,\sigma)$ and

$$g(x,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-x^2}{2\sigma^2}}$$

However, the above expression is also equivalent to:

$$U_x(x,y) = U(x,y) * \left(\frac{\partial G(x,y)}{\partial x}\right)$$

and

$$\frac{\partial G(x,y,\sigma)}{\partial x} = \frac{\partial g(x,\sigma)}{\partial x} g(y,\sigma)$$

Thus, we can estimate U_x by convolving U with $h_x[x,y] = g'[x]g[y]$ where g' represents the derivative of a Gaussian.

3.2.2 Estimating Local Wave Speed and Decay

To estimate σ and λ we just simply compute $\frac{U[x,y]}{U_x[x,y]}$ at each pixel. The real and imaginary parts of this ratio represent the estimates of $-1/\sigma$ and $-1/\lambda$ respectively. Since these estimates are subject to noise, we apply a final Gaussian blur to both the imaginary and real parts of $\frac{U[x,y]}{U_x[x,y]}$. This is useful because this ratio should be constant in the TM regions.

3.2.3 Detecting non-TM Regions

After computing the local estimates of σ and λ, the next step is to eliminate regions that are not part of the TM. The images used in our experiments usually contain the three following elements:

1. The TM, whose motion behaves as a decaying traveling wave.

2. The background, which has either a noisy motion or no movement at all.

3. Two coverslip supports, which move like a rigid body.

Next, we will describe in more detail how to detect and remove pixels that are classified as background or inside a coverslip support.

Detecting the background

We used two different strategies to eliminate the background pixels. The first one is based on motion. Since the estimated amplitudes and phases are random in this region, the estimates of σ and λ should also be random. In contrast, the estimates of σ and λ inside the TM should be roughly uniform. This is based on our assumption that the TM motion behaves like a decaying traveling wave.

To measure how similar the local estimates of σ and λ are around a certain pixel, we compute the magnitude of their gradients. More specifically, for σ we compute:

$$\nabla_\sigma[x,y] = \sqrt{\hat{\sigma}_x^{-1}[x,y]^2 + \hat{\sigma}_y^{-1}[x,y]^2}$$

where $\hat{\sigma}^{-1}$ is the local approximation of $1/\sigma$ and $\hat{\sigma}_x^{-1}$ and $\hat{\sigma}_y^{-1}$ are the approximation of the partial derivatives in x and y of $\hat{\sigma}^{-1}$ respectively. These are computed with the technique described in section 3.2.1. We also define and compute $\triangledown_\lambda$ analogously.

Notice that if $\hat{\sigma}^{-1}$ and $\hat{\lambda}^{-1}$ are uniform locally, then both $\triangledown_\sigma$ and $\triangledown_\lambda$ should be close to 0. On the other hand, if these estimates are not uniform then these are much greater than 0. Thus, we can pick a threshold and ignore all points with $\triangledown_\sigma$ and $\triangledown_\lambda$ above that certain threshold. In addition, we also eliminate regions where the amplitude is below a certain threshold. This is to account for edge cases where the background amplitude is very small but the phases are uniform.

The second strategy is based on brightness. Unlike with motion, the background's brightness is mostly uniform meaning that the magnitude of its gradient is close to 0 in this region. This is the opposite for objects like the TM where brightness changes considerably. Thus, we eliminate pixels whose gradient's magnitude is bigger than a certain threshold. The magnitude of the gradient is computed analogously as described for $\triangledown_\sigma$ and $\triangledown_\lambda$.

Detecting the coverslip supports

The previous methods for eliminating the background are not effective for detecting the two coverslip supports. Their local σ and λ estimates are uniform since the coverslip supports also move sinusoidally. In addition, their brightness is not uniform unlike the background.

However, the coverslip supports' motion is significantly different from the TM's; they move like a rigid body. More specifically, U should be uniform across the rigid body meaning that U_x should be 0. This implies that both $\hat{\sigma}^{-1}$ and $\hat{\lambda}^{-1}$ should be 0 inside the coverslip supports while attaining non-negligible values inside the TM. Thus, we pick a threshold and ignore all pixels with either $\sigma^{\hat{-1}}$ or $\hat{\lambda}^{-1}$ below that threshold to detect and eliminate pixels corresponding to the coverslip supports.

d^2	Error σ	Error λ
No noise	0.24%	0.22%
0.05	2.05 %	0.25 %
0.10	2.34 %	0.36%
0.20	5.22%	0.65 %

Table 3.1: Results of the first experiment. The first column represents the amount of noise added. The second and third columns display the average absolute error with respect to the true values of σ and λ.

3.2.4 Averaging Local Wave Estimates

The final step of the algorithm is to obtain a global σ and λ from the remaining pixels. We achieve this by:

1. Computing the mean of $\hat{\sigma}^{-1}$ and $\hat{\lambda}^{-1}$.

2. Eliminating pixels whose average is one standard deviation away from the mean.

3. Recomputing the mean of the remaining pixels.

3.3 Results

3.3.1 Synthetic Displacements

The first experiment aimed to test the method for estimating $1/\sigma$ and $1/\lambda$ locally. It consisted of generating a 250×250 2d-array satisfying the following function:

$$f(x, y) = e^{-\frac{x}{\sigma}} e^{-\frac{jx}{\lambda}}$$

where $\sigma = 1818$ and $\lambda = 222$. To test for robustness, we multiply the real and imaginary values by a normal random variable of mean 1 and variance d^2.

For estimating the gradients, we truncate our Gaussian kernel as a 25×25 box with a std of 12 in the y direction and 3 in the x direction. For estimating $1/\sigma$ and $1/\lambda$, we applied a Gaussian blur with a 25 x 25 kernel and standard deviation 12.

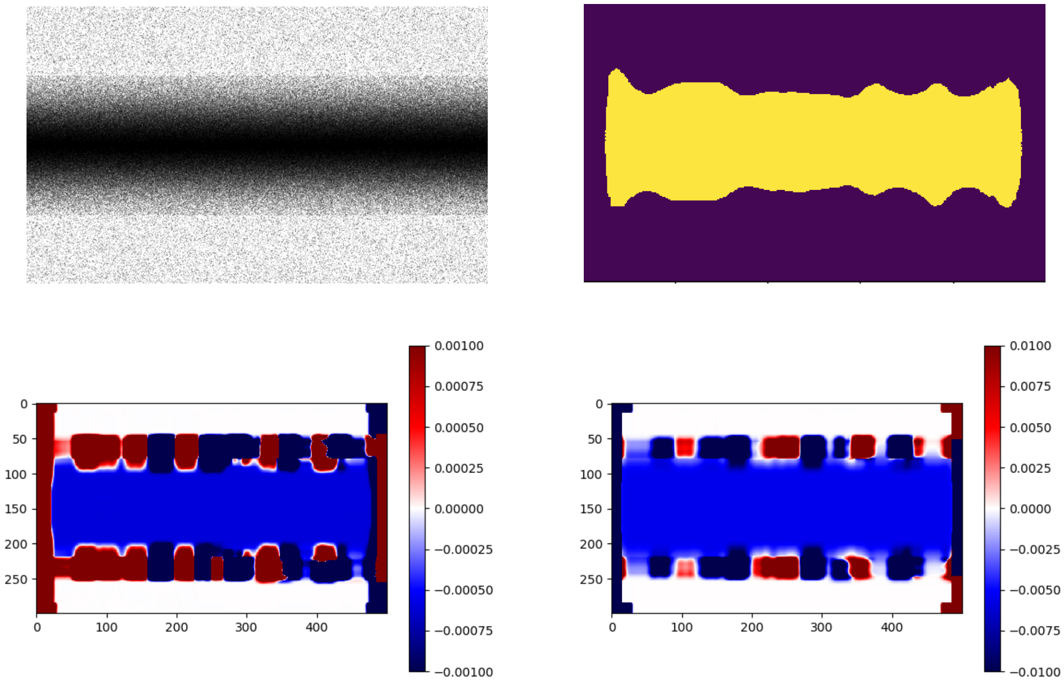


Figure 3-1: (A) Sample image with shot noise used to simulate TM experiments. (B) Results of background elimination based on motion. (C) Estimates of the real part of U_x/U or $-1/\sigma$. The true value is 6e-4. (D) Estimates of imaginary part of U_x/U or $-1/\lambda$. The true value is 5.5e-3. For both σ and λ, the algorithm estimates the right values in the interior of the simulated TM. However, there are artifacts near the edges of the TM in both cases, more prominent for σ than λ. These are ultimately ignored by the background elimination since the values of σ and λ are not uniform in these regions with artifacts.

.

The results can be found in table 3.1. We reported the average absolute error with respect to the valid answer after 10 trials.

The results are encouraging and consistent with our previous experiments. Again, λ estimates are extremely precise and robust while σ is more sensitive to noise.

3.3.2 Synthetic Images

For the second experiment, we created eight synthetic images with a fake TM and a background. The fake TM was generated by a sequence of pixel columns with brightness values modeled from a Gaussian distribution. The center of each column

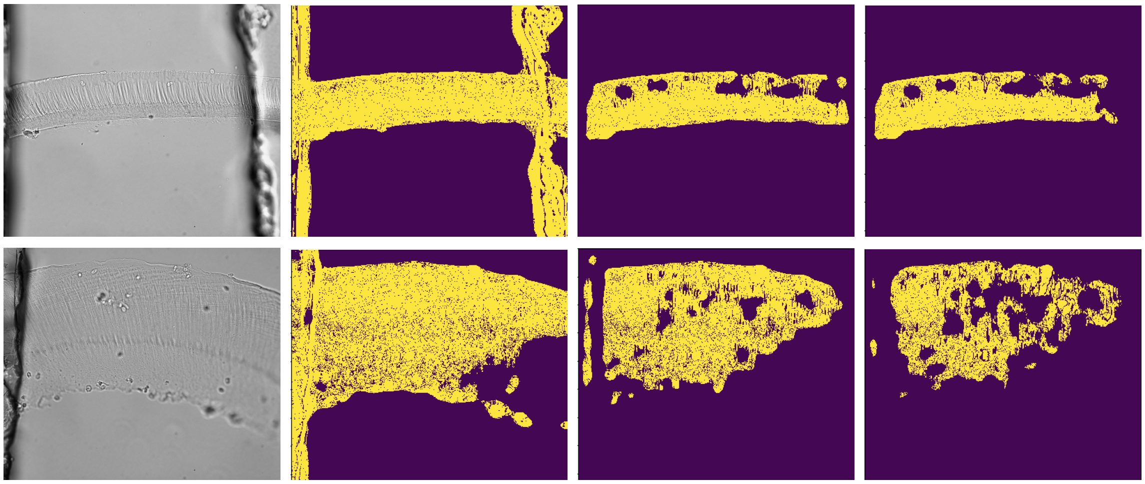


Figure 3-2: Results of TM detection for both real TM samples. From left to right: (A) TM original image. (B) TM detection by brightness only. (C) TM detection by brightness and wave motion (D) Final output, including rigid body removal. Most of the background gets eliminated in these examples when looking at changes in brightness. However, brightness alone cannot eliminate the coverslip supports. These are filtered in (C) and (D), resulting in a high precision segmentation. This is not as true for recall. However, precision is much more important than recall since σ and λ are constant across the entire TM in our ideal model and the TM generally occupies the majority of the image.

.

is dark while the edges are bright. A sample image can be found in figure 3-1. The columns move only in the vertical direction. The motion travels with $\sigma = 1666.66$ and $\lambda = 181.18$. To test for robustness, we added fixed pattern noise and shot noise.

The parameters used for estimating σ and λ locally are the same as used in the first experiment. We also performed the background detection algorithm based on motion with a threshold of 2e-4 and ignoring pixels with amplitudes smaller than 0.1.

Figure 3-1 shows the results of the background elimination as well as the local estimates of σ and λ. In all the cases, the global estimates were almost identical despite the noise levels. The errors were 0.99% for σ and 0.25% for λ, which shows that this method is robust especially for λ.

	Previous decay	New decay	Previous speed	New speed
TM Sample 1	326.55 µm	246.53 µm	4.73 m/s	4.59 m/s
TM Sample 2	274.85 µm	207.93 µm	5.75 m/s	5.57 m/s

Table 3.2: Comparison of wave decay and speed estimates between previous and current chapter's methods. The decay is in micrometers and the speed in meters per second. The speed estimates are similar while the decay estimates are not.

3.3.3 Real Data

For the last experiment, we used two real TM samples. The results of the TM detection are displayed in figure 3-2. Overall, the TM detection method prioritizes precision (number of false positives) over recall (number of false negatives). However, this trade off is still positive. The TM region, which occupies most of our images, should contain similar estimates of σ and λ while the background and coverslip supports provide much more different estimates. Thus, precision is much more important than recall since it is more crucial to minimize the number of false positives.

We compared this new method's wave decay and speed estimates with the previous chapter's. The two selected images already had fairly reliable estimates with the method described in chapter 2. This comparison can be found in table 3.2. The speed estimates are almost identical in both methods. However, this is not true for the decay. The main reason is that the amplitudes do not decay exponentially at all points in the TM. This phenomena is displayed in figure 3-3. We will address this issue in the next chapter.

3.4 Discussion

In this chapter we explored a new algorithm for estimating wave decay and speed automatically. This method uses the entire TM images instead of a parametric path. This method demonstrates a high accuracy for synthetic data. It was also very effective at estimating wave speed in real samples, but not so much for the decay. The reason is that amplitudes do not exactly decay exponentially in real TM samples. This issue will be addressed in the next chapter.

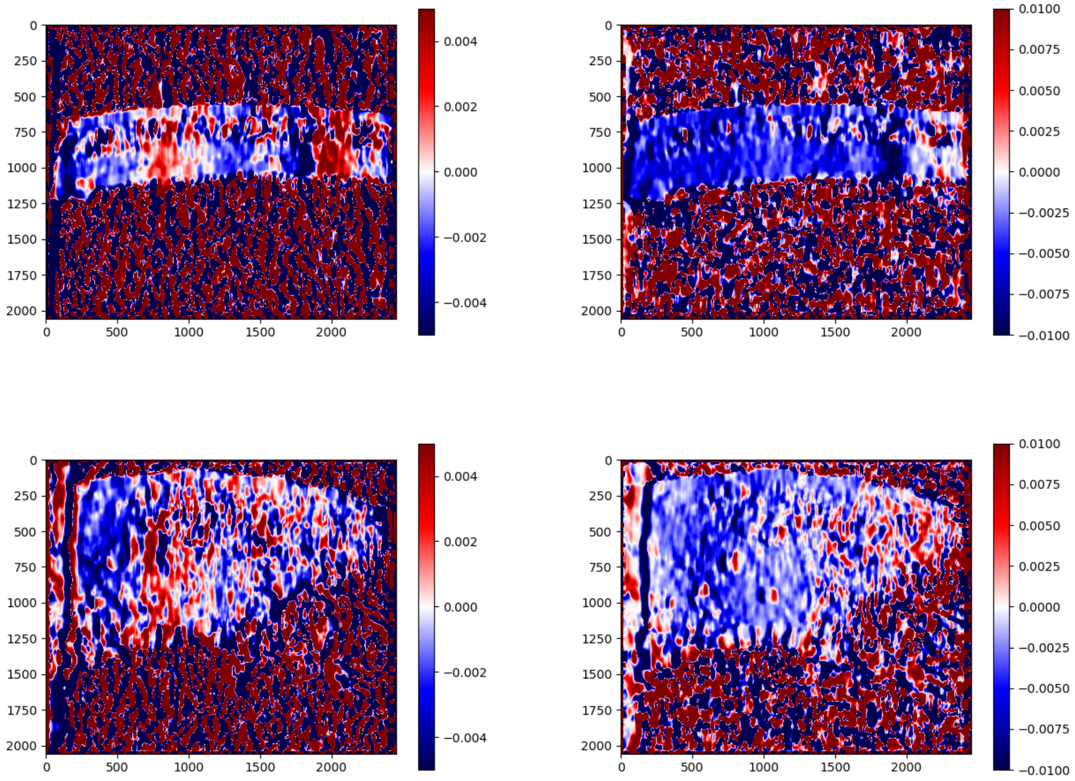


Figure 3-3: The left plots represent the local estimates of $-\frac{1}{\sigma}$ and the right plots represent the local estimates of $-\frac{1}{\lambda}$. This illustrates the challenge of estimating σ with this method. In real TMs, the amplitudes do not exactly decay exponentially. In some regions the amplitudes even increases. However, the phases decay mostly uniformly resulting in much more precise measurements.

.

Chapter 4

Wave Reflection Effect

This chapters introduces the concept of wave reflection in TM experiments. If wave reflection occurs, a reverse wave is present and it interferes with the original wave motion traveling through the TM. This chapter analyses how reverse waves affect previous decay and speed estimates and incorporates them to achieve more robust and accurate results.

4.1 Wave Reflection

In previous chapters, we observed that the amplitudes in real data were not decaying exponentially as stated in our ideal TM model. To correct for this issue, we introduce the concept of wave reflection. This has been used in previous TM studies for more robust measurements [7].

In the TM experiments, the motion wave travels from one coverslip support to another. The first coverslip support induces a forward decaying traveling wave to the TM. The second coverslip support is stationary. However, this second coverslip support can cause a wave reflection, generating a reverse wave with the same speed and decay but with an opposite direction.

More formally, the radial complex displacements alongside the longitudinal direction should satisfy the following function:

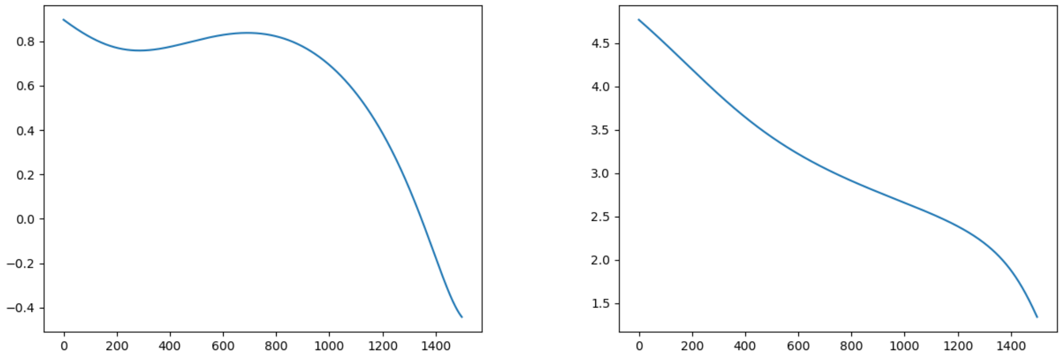


Figure 4-1: Plot of the log of the amplitudes (left) and phases (right) when a reverse wave is present. The amplitudes are more affected than the phases. The amplitudes oscillate since the forward and reverse wave interfere both constructively and destructively.

.

$$D(x,\omega,t) = (B_f e^{-\frac{x}{\sigma}-j\frac{x}{\lambda}} + B_r e^{\frac{x}{\sigma}+j\frac{x}{\lambda}})e^{jwt}$$

where B_f and B_r are complex constants, σ and λ represent the decay and speed. Notice that the sum of two sinusoidal waves is still a sinusoidal wave. However, these two waves interfere both constructively and destructively. This means that the amplitudes and phases do not decay as in our ideal model. This is especially true for amplitudes as seen in figure 4-1. The amplitude sometimes increases when the two waves interfere constructively and decay dramatically when these interfere destructively.

4.2 Fitting Displacements with a Reverse Wave

For simplicity, we will test the reflection effect theory by using the method described in chapter 2. Recall that the goal was to find a_0, ϕ_0, σ and λ that minimize:

$$\sum_{x=0}^{s} |U(x) - e^{a_0-\frac{x}{\sigma}} e^{j(\phi_0-\frac{x}{\lambda})}|^2$$

Refl. factor	σ Error	λ Error
1	63.06 %	46.76%
2	15.52 %	15.33 %
5	4.40 %	5.05 %
10	1.97 %	2.38%

Table 4.1: Error associated to the reflection effect. The first column represents the reflection factor or ratio between the amplitudes of the forward and reverse waves where the reflection occurs. The reported figures correspond to the absolute error with respect to the true values.

where $U(x) = A(x)e^{j\phi(x)}$ and $A(x)$ and $\phi(x)$ represent the amplitude and phase at position x of the sinusoidal motion alongside the longitudinal axis. We extend this model by adding two extra variables a_0^r, ϕ_0^r and the goal becomes minimizing:

$$\sum_{x=0}^{s} |U(x) - e^{a_0-\frac{x}{\sigma}} e^{j(\phi_0-\frac{x}{\lambda})} - e^{a_0^r+\frac{x}{\sigma}} e^{j(\phi_0^r+\frac{x}{\lambda})}|^2$$

4.3 Results

4.3.1 Synthetic Displacements

The first experiment consisted in generating an array of size 1500 satisfying the following function:

$$f(x) = e^{a_0-\frac{x}{\sigma}} e^{\phi_0-\frac{x}{\lambda}} + e^{a_0^r+\frac{x}{\sigma}} e^{\phi_0^r+\frac{x}{\lambda}}$$

This experiment had two different goals. The first is to predict σ and λ using the method described in chapter 2 with different reflection factors. This is to measure the possible errors in previous measurements for not including the reverse wave in the ideal model. The second goal is to test this new method's robustness to noise.

Reflection effect in previous fitting method

We applied the method described in chapter 2 to measure how wave reflection affects the estimates of σ and λ. To achieve this, we chose different ratios of $\frac{e^{a_0-\frac{x}{\sigma}}}{e^{a_0^r+\frac{x}{\sigma}}}$ or ratio

d	σ Error	λ Error
No noise	0.00 %	0.00%
0.05	0.31 %	0.04%
0.10	1.89 %	1.05 %
0.20	8.92 %	5.96 %

Table 4.2: Results after adding noise to the displacements. The metric used was average absolute error with respect to the true values.

between the amplitudes of both waves where the reflection occurs. The parameters used were $a_0 = 1, \phi_0 = 0, \sigma = 2000$ and $\lambda = 444.44$.

The results are displayed in table 4.1. The errors are significantly high when the reflection factor is close to 1, even surpassing 50% for both σ and λ estimates. This suggests that if a strong reverse wave is present, the previous measurements of σ and λ might not be reliable.

Robustness to noise

To test the new method's robustness to noise, we multiplied the real and imaginary parts of the generated displacements by a normal random variable of mean 1 and variance d^2. The parameters used to generate the displacements were $a_0 = 1, \phi = 0, \sigma = 2000, \lambda = 444.44$ with a reflection factor of 1.

The results are included in table 4.2. We ran 100 trials for each noise level and reported the average absolute error. The results suggest this new method is robust to noise. For low noise levels the errors were below 2%. The results were not as accurate for higher noise levels ($d = 0.2$). However, these are still reasonable considering the amount of noise.

4.3.2 Real Data

For the second experiment, we selected three different TM samples where the previous measurements of decay and speed seemed implausible. We estimated the decay and speed of the three samples with and without considering the reflection effect.

The results suggest that a reverse wave is present in these three samples. Figure

	No refl. decay	Refl. decay	No refl. speed	Refl. speed
TM Sample 1	298.03 µm	293.32 µm	4.82 m/s	4.76 m/s
TM Sample 2	426.12 µm	259.42 µm	3.49 m/s	3.73 m/s
TM Sample 3	250.78 µm	377.43 µm	6.75 m/s	8.29 m/s

Table 4.3: Comparison of wave decay and speed estimates with and without adding the reverse wave in our fitting model. The decay is in micrometers and the speed in meters per second. Both estimates are different. This is more noticeable for the decay, where some estimates are completely different.

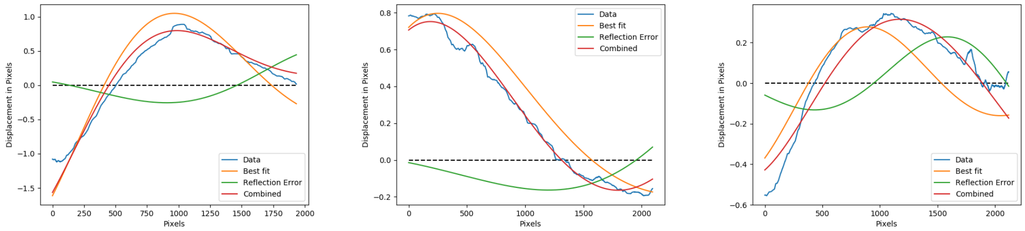


Figure 4-2: Plots of the measured displacement of the three samples and the best fits. The blue curve represents the real steady state displacements. The orange and green curve represent the forward and reverse wave fits respectively. The red curve is a combination of these two, providing a good approximation in all three cases. The reverse wave is also significant in all samples.

.

4-3 shows snapshots of the raw displacements with and without correcting for the reflection effect. Subtracting the reverse wave from the original displacements results in a wave motion similar to a decaying traveling wave. In addition, figure 4-2 displays the fitted forward and reverse wave. These two waves constructively and destructively interfere to achieve a fit similar to the original data.

Finally, table 4.3 compares how much the decay and speed changed by including the reflection effect in our model. The decay changes considerably in some samples as we expected. The speed also varies for all samples but in a lesser degree. This is consistent with the results obtained in the previous chapters.

4.4 Discussion

In this chapter we introduced the concept of wave reflection in the TM experiments. This theory was previously studied in an attempt for more robust wave measurements. By experimenting with synthetic data, we showed that not adjusting for the reflection effect might yield very different estimates. This was also true in real data. We obtained more precise and reliable fits by adding the reverse wave in our model. This suggests that reverse waves are present in many TM experiments, which could significantly affect the final wave decay and speed estimates.

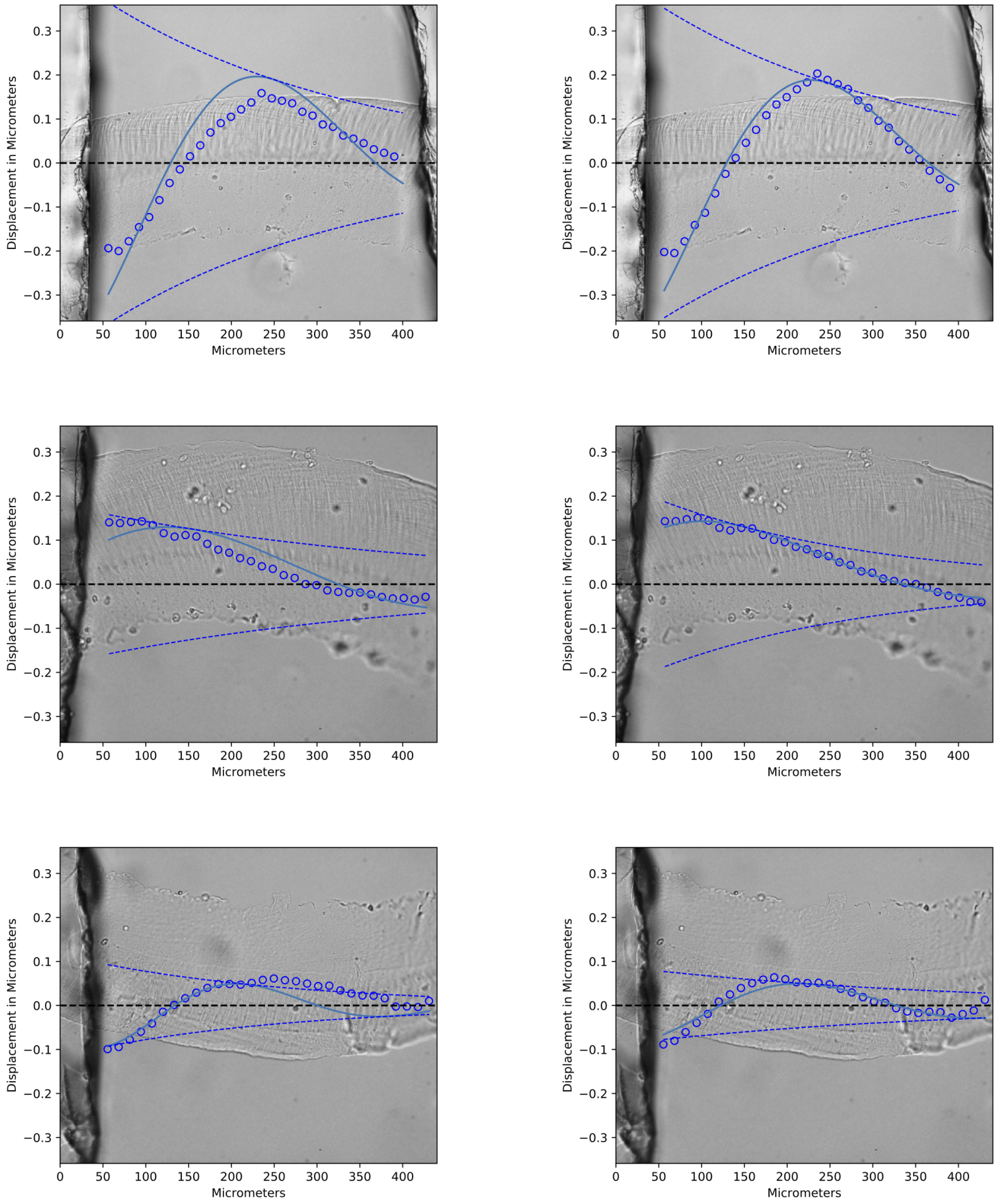


Figure 4-3: Wave snapshots of the real steady-state displacements. In the left images, the blue dots represent the displacements measured from the motion estimation algorithm. The blue curve represents the best fit with the traditional method. In the right images, the blue dots represent the original displacements minus the reverse wave in order to obtain a decaying traveling wave pattern. The blue curve represents the best forward wave fit. In all three cases, the left fits seem to have inaccuracies. However, the fits on the right plots provide a good approximation of the data validating our theory that a reverse wave is present.

.

Chapter 5

Conclusions

In this thesis we have presented several algorithms for robust estimation of TM wave motion parameters with minimum human intervention. After a brief introduction to the problem (chapter 1), the main results were organized in three sections (chapters 2 to 4).

In chapter 2, we presented a general purpose framework that allows the user to analyse wave motions along a parametrically described trajectory. This framework is more flexible, faster and less prone to error than previous methods. This chapter also compared two different algorithms for estimating wave speed and decay given the amplitudes and phases along a defined longitudinal axis. The first method involves fitting the amplitudes and phases separately to compute the wave decay and speed respectively. The second method consists in fitting the complex displacements to jointly obtain the decay and speed. The latter proved to be more robust, accurately predicting the decay and speed whenever the motion data behaved as a decaying traveling wave. However, some TM samples presented amplitudes that did not decay uniformly. For these cases, the speed estimates were still accurate while the the decay estimates were not as reliable.

In chapter 3, we proposed a new fully automatic gradient-based algorithm for wave estimation. This novel method does not require manual specification of a longitudinal axis since it computes a global wave decay and speed based on the whole TM image and motion. This method provided encouraging results in synthetic data for both decay and speed. Furthermore, this method accurately estimated the wave speed in real TM samples. However, it was not as effective for the wave decay. This is due to amplitudes in real TM samples being noisy and not decaying uniformly in certain regions.

In chapter 4, we introduced the wave reflection effect in TM experiments. We first showed that when a wave reflection is present, amplitudes and phases do not decay as in the ideal model of the TM. By analysing synthetic data we concluded that previous estimates of wave decay and speed can be significantly distorted if the reflection is large enough. Finally, we tested this theory on TM samples where wave motions did not behave as a decaying traveling wave. By including a reverse wave in our model, we achieved fits that fit well with real data. The biggest effect of the reflected wave was in changes to the decay estimates, which is consistent with our results in the previous chapters. This suggests that the reflection effect is present in many TM experiments and should be considered for more robust estimates.

Bibliography

[1] Mary Ann Branch, Thomas F Coleman, and Yuying Li. A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. *SIAM Journal on Scientific Computing*, 21(1):1–23, 1999.

[2] Richard H Byrd, Robert B Schnabel, and Gerald A Shultz. Approximate solution of the trust region problem by minimization over two-dimensional subspaces. *Mathematical programming*, 40(1-3):247–263, 1988.

[3] Simon S. Gao, Patrick D. Raphael, Rosalie Wang, Jesung Park, Anping Xia, Brian E. Applegate, and John S. Oghalai. In vivo vibrometry inside the apex of the mouse cochlea using spectral domain optical coherence tomography. *Biomed. Opt. Express*, 4(2):230–240, Feb 2013.

[4] Roozbeh Ghaffari, Alexander J. Aranyosi, and Dennis M. Freeman. Longitudinally propagating traveling waves of the mammalian tectorial membrane. *Proceedings of the National Academy of Sciences*, 104(42):16510–16515, 2007.

[5] Roozbeh Ghaffari, Alexander J Aranyosi, Guy P Richardson, and Dennis M Freeman. Tectorial membrane travelling waves underlie abnormal hearing in tectb mutant mice. *Nature communications*, 1(1):1–6, 2010.

[6] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[7] Charlsie Lemons, Jonathan B Sellon, Elisa Boatti, Daniel Filizzola, Dennis M Freeman, and Julien Meaud. Anisotropic material properties of wild-type and tectb-/- tectorial membranes. *Biophysical journal*, 116(3):573–585, 2019.

[8] Jorge J Moré. The levenberg-marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105–116. Springer, 1978.

[9] Tianying Ren. The cochlear amplifier and ca 2+ current–driven active stereocilia motion. *Nature neuroscience*, 8(2):132, 2005.

[10] Jonathan B Sellon, Shirin Farrahi, Roozbeh Ghaffari, and Dennis M Freeman. Longitudinal spread of mechanical excitation through tectorial membrane traveling waves. *Proceedings of the National Academy of Sciences*, 112(42):12968–12973, 2015.

[11] Samson J Timoner and Dennis M Freeman. Multi-image gradient-based algorithms for motion estimation. *OPTICAL ENGINEERING-BELLINGHAM-INTERNATIONAL SOCIETY FOR OPTICAL ENGINEERING-*, 40(9):2003–2016, 2001.