Application of Image Processing Techniques for Lamb Wave Characterization

A Thesis
Presented to
The Academic Faculty

by

Oliver Kotte

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in
Engineering Science and Mechanics

School of Civil and Environmental Engineering
Georgia Institute of Technology
August 2004

Application of Image Processing Techniques for Lamb Wave Characterization

Approved:

Dr. Laurence Jacobs, Chairman

Dr. Reginald DesRoches

Dr. Jianmin Qu

Date Approved: August 18, 2004

To Mother Earth, for all her trees.

Acknowledgements

I would like to thank Dr. Laurence J. Jacobs for being an outstanding advisor and friend. I am very grateful for his general support and advisement. He contributed in many ways to make my year in Atlanta a great experience. Furthermore, my thanks go to Dr. Reginald DesRoches and Dr. Jianmin Qu for their help and support.

I extend my gratitude to Marc Niethammer for his advise, great support, genuine interest in my research, fruitful discussions, and friendship.

I also extend my gratitude to my colleagues — Tobi, Hilde, The Doctor, Bo, Monica, Eak — for many memorable discussions and the good time we shared together.

Although not directly involved in this work, my thanks go to Prof. Lothar Gaul for giving me the opportunity to study at Georgia Tech. My studies were made possible by the generous financial support of the DAAD (German Academic Exchange Service).

My greatest thanks go to my wife, my family, and mother earth, for love, peace, and happiness.

Table of Contents

List of Tables

List of Figures

List of Symbols or Abbreviations

$[\quad]$	enclose index for a list of discrete values
$\lvert\quad\rvert$	magnitude
$\tilde{}$	periodic
$\lVert\quad\rVert$	square norm
Θ	vector field scaling matrix
Φ, Ψ	flow functions
α	step size scaling factor
β	diffusion weighting factor
δ	threshold value
δ_{ij}	Kronecker symbol
ϵ_{ij}	strain tensor
ε	machine precision
κ	diffusion constant
λ, μ	Lamé constants
ν	Poisson's ratio
ρ	density
σ	standard deviation
σ_{ji}	stress tensor
ϕ_{tf}	kernel function
φ	displacement potential
$\boldsymbol{\psi}$	displacement potential
ω	angular frequency
$\hat{\omega}$	reassigned angular frequency

D_+	forward difference
D_-	backward difference
E	expected value
E_d	energy density
G	measure of nonanalyticity
I	image
I^{flow}	flow due to energy movement
I^{q}	characteristic image quotient
$S(\omega)$	Fourier transform of signal $s(t)$
WV	Wigner-Ville distribution
c	diffusion function
c_g	group velocity
c_L	phase velocity of a longitudinal wave
c_T	phase velocity of a shear wave
$\mathbf{d}$	direction of particle motion
d_i	measurement distance
f	frequency
f_N	Nyquist frequency
f_s	sampling frequency
$\mathbf{f}$	body force
$h(t)$	window function
i	imaginary unit
k	wavenumber
$\mathbf{p}$	direction of wave propagation
r	reassignment displacement vector field
t	time
$\triangle t$	step size
$\hat{t}$	reassigned time
$\mathbf{u}$	displacement
v	reassignment velocity vector field
$\bar{v}$	scaled reassignment velocity vector field

z	complex variable
$\Im$	imaginary part of a complex number
$\Re$	real part of a complex number
$\mathcal{F}$	Bargmann factorization function
$\mathcal{N}$	set of negative image cells
$\mathcal{T}$	TFR

Summary

Characterization of dispersion curves in plate-like structures is possible with guided Lamb waves. In this research, experimental development of dispersion curves relies on the spectrogram, which suffers from the Heisenberg Uncertainty Principle. Reassignment is capable of localizing ill–defined dispersion curves. Unfortunately, reassignment also introduces spurious components, which reduce reassignment performance. This research develops an algorithm that provides both localization of dispersion curves and elimination of spurious components. To achieve this, an alternative formulation of reassignment called differential reassignment is modified and superimposed with nonlinear anisotropic diffusion. This study first examines reassignment and diffusion components individually. Three different versions of differential reassignment are considered, two of which are modifications explicitly derived in this research. The combined algorithm is then applied to reassign experimentally measured spectrograms, leading to a significant increase in clarity and notch detection performance.

CHAPTER 1

Introduction

Defects in plate-like structures such as aircraft skins can be located and sized using guided Lamb waves.

Previous research conducted by Hurlebaus et al. [14] and Benz [5, 13] has shown that a combination of laser based ultrasound, with a time frequency representation (TFR), is effective in locating notched defects plates. In this procedure, notch detection relies on a TFR's ability to resolve the individual modes of a plate, and to localize the generated dispersion curves. The TFR Benz selected was the reassigned spectrogram because previous research by Niethammer [20, 23] indicated that for this specific purpose, the reassigned spectrogram is superior to other TFR's, including the reassigned scalogram, the Wigner-Ville distribution, and Hilbert spectrum.

Unfortunately, the spectrogram suffers from the Heisenberg uncertainty principle, making it impossible to simultaneously have perfect resolution in both time and frequency. In an effort to enhance the spectrogram's readability, reassignment can reduce the time frequency spread, thus better localize ill-defined dispersion curves. The reassignment method was developed by Auger and Flandrin [4], who found a computationally effective way to calculate the modified moving window method developed by Kodera et al. [10]. Although reassignment localizes dispersion curves, it does introduce spurious components. A few years later, Chassande–Mottin et al. [3] published an alternative formulation of the reassignment method, called differential reassignment. While the "conventional" reassignment method works on the time-domain signal, differential reassignment can be viewed as image processing, based on a partial differential equation (PDE).

Any spurious components introduced by the reassignment method will reduce the clarity of a reassigned spectrogram. Notch detection relies on the reassigned

spectrogram, so detection quality increases with reassignment quality. The objective of this research is to improve the performance of spectrogram reassignment. This study develops an algorithm that provides both localization of dispersion curves and elimination of spurious components.

First, both differential reassignment and anisotropic diffusion are treated individually. Three different versions of differential reassignment are examined. The first one follows the defining equations of differential reassignment to move energy towards maxima of a scalar potential function. The second version uses alternative trajectories and velocities to move energy in a different fashion towards the same maxima. The third version uses the same trajectories as predicted by reassignment equations, but introduces a design parameter to favorably vary travel velocity along these trajectories. It is this third version that proves to be most effective, so it is combined with anisotropic diffusion to yield the final algorithm. A PDE is presented which fully describes image evolution caused by energy movement.

The final, combined algorithm is applied to experimentally measured spectrograms, which are then used for notch localization.

CHAPTER 2

Theoretical background

This chapter provides a brief introduction to wave propagation in elastic solids and to the signal and image processing methods used in this study. There are a number of authoritative and comprehensive books on wave propagation theory and the author recommends interested readers looking at [1, 11, 12, 24] for deeper coverage. For more information on signal processing, the reader is referred to Niethammer [20] and [23] providing a variety of signal processing methods that have the potential to assist in the interpretation of ultrasonic waveforms; for more mathematical details and derivations see for example [9, 18, 19]. For details on reassignment methods, consider [2, 3, 4, 6]. To learn more about diffusion algorithms, refer for example to [22, 25]. Note that a complete coverage of these subjects is beyond the scope of this thesis, but this chapter provides a basic review.

2.1 Wave propagation

In order to provide the reader with an at least rudimentary understanding of the materia, wave propagation in elastic solids is briefly outlined in this section. This will allow for interpretation of the information content carried by the spectrograms used in Chapter 4. The following subsections are not essential for an understanding of the algorithm derived in Chapter 3. However, this knowledge enables the reader to appreciate the significance of the achieved reassignment improvement.

2.1.1 Linear elasticity and equation of motion

In linear elasticity, the traction t_i on a plane $n_i x_i = d$ is given by

$$t_i = \sigma_{ij} n_j, \tag{2.1}$$

where σ_{ji} is the stress tensor.

The balance of linear momentum for a body with volume V and surface S can be expressed as

$$\int_S \sigma_{lk} n_k \mathrm{d}S + \int_V \rho f_l \mathrm{d}V = \int_V \rho \ddot{u}_l \mathrm{d}V, \tag{2.2}$$

with ρ representing the material mass density and f_i the body force. Gauss' theorem applied to Equation 2.2 leads to

$$\int_V (\sigma_{lk,k} + \rho f_l - \rho \ddot{u}_l) \mathrm{d}V = 0. \tag{2.3}$$

Equation 2.3 has to be fulfilled for any arbitrary volume V of the body, and therefore the stress equations of motion becomes

$$\sigma_{lk,k} + \rho f_l = \rho \ddot{u}_l. \tag{2.4}$$

It is often more efficient to have the equations of motion given solely in terms of the displacement, u_i (as opposed to Equation 2.4, which has terms of stress σ_{ij} and displacement u_i). This can be achieved by applying to Equation 2.4 Hooke's law for homogeneous, isotropic and linear elastic medium

$$\sigma_{ij} = \lambda \epsilon_{kk} \delta_{ij} + 2\mu \epsilon_{ij}, \tag{2.5}$$

where ϵ_{ij} is the strain tensor, related to the displacement u_i by

$$\epsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}), \tag{2.6}$$

and μ and λ are the Lamé constants. Plugging Equation 2.6 into Equation 2.5 and subsequently into Equation 2.4 leads to Navier's equations of motion

$$\mu u_{i,jj} + (\lambda + \mu) u_{j,ji} = \rho \ddot{u}_i \tag{2.7}$$

$$\mu \nabla^2 \mathbf{u} + (\lambda + \mu) \nabla \nabla \cdot \mathbf{u} = \rho \ddot{\mathbf{u}}. \tag{2.8}$$

Note that in this development, body forces $\mathbf{f}$ are neglected. Solving Equation 2.8, however, is difficult, because it is a coupled partial differential equation (PDE). Applying to Equation 2.8 the Helmholtz decomposition

$$\mathbf{u} = \nabla \varphi + \nabla \times \boldsymbol{\psi}, \tag{2.9}$$

provides a convenient way to uncouple these equations. Equation 2.9 represents the three components of displacement $\mathbf{u}$ with the four functions $\varphi,\ \psi_1,\ \psi_2$ and ψ_3. To guarantee the uniqueness of the solution, an additional constraint

$$\nabla \cdot \boldsymbol{\psi} = 0 \tag{2.10}$$

is introduced. Substitution of equation Equation 2.9 (Helmholtz decomposition) into the displacement equations of motion (Equation 2.8) leads to two uncoupled wave equations expressed in terms of the displacement potentials φ and $\boldsymbol{\psi}$

$$\nabla^2\varphi = \frac{1}{c_L^2}\ddot{\varphi}, \qquad \nabla^2\boldsymbol{\psi} = \frac{1}{c_T^2}\ddot{\boldsymbol{\psi}}, \tag{2.11}$$

where (as will be shown later) c_L represents the wave speed of the longitudinal wave and c_T the wave speed of the vertically and horizontally polarized (transverse) shear waves,

$$c_L^2 = \frac{\lambda + 2\mu}{\rho}, \qquad c_T^2 = \frac{\mu}{\rho}. \tag{2.12}$$

Both wave speed equations are expressed in terms of material properties, namely the density ρ and the Lamé constants μ and λ. A relationship for the material properties Young's modulus E and Poisson's ratio ν can be obtained through

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \tag{2.13}$$

$$\mu = \frac{E}{2(1+\nu)}. \tag{2.14}$$

2.1.2 Wave phenomena

Wave phenomena discussed in this section are based on the plane wave assumption, i.e. assuming a wave with constant properties (ϵ, σ, u) on a plane perpendicular to its direction of propagation $\mathbf{p}$. Equation 2.15 shows the mathematical representation of a plane wave,

$$\mathbf{u} = f(\mathbf{x} \cdot \mathbf{p} - ct)\mathbf{d}, \tag{2.15}$$

where $\mathbf{d}$ is the unit vector defining the direction of particle motion, and c is either the longitudinal wave speed c_L or the transverse wave speed c_T. By substituting Equation 2.15 into Equation 2.8, one obtains

$$(\mu - \rho c^2)\mathbf{d} + (\lambda + \mu)(\mathbf{p} \cdot \mathbf{d})\mathbf{p} = 0. \tag{2.16}$$

Since $\mathbf{p}$ are two different unit vectors, it can immediately be seen that the two possible solutions that form the basis of wave propagation are either $\mathbf{d} = \pm\mathbf{p}$ or $\mathbf{p} \cdot \mathbf{d} = 0$:

- $\mathbf{d} = \pm\mathbf{p}$ leads to $\mathbf{p} \cdot \mathbf{d} = \pm 1$ and yields with Equation 2.16, $c = c_L$ (see Equation 2.12). Since $\mathbf{d}$ and $\mathbf{p}$ are linearly dependent, this represents a particle movement in the direction of propagation — a longitudinal or P-wave.

- $\mathbf{p} \cdot \mathbf{d} = 0$ yields with Equation 2.16, $c = c_T$ (see Equation 2.12). Now the direction of motion is normal to the direction of propagation, and the wave is called a transverse wave. If a two dimensional plane of propagation is considered (for example, the (x_1, x_2)-plane), a wave with an in-plane displacement (in the (x_1, x_2)-plane) is called an SV-wave (vertically polarized), while a wave with out-of-plane displacement (in the x_3-direction) is called an SH-wave (horizontally polarized).

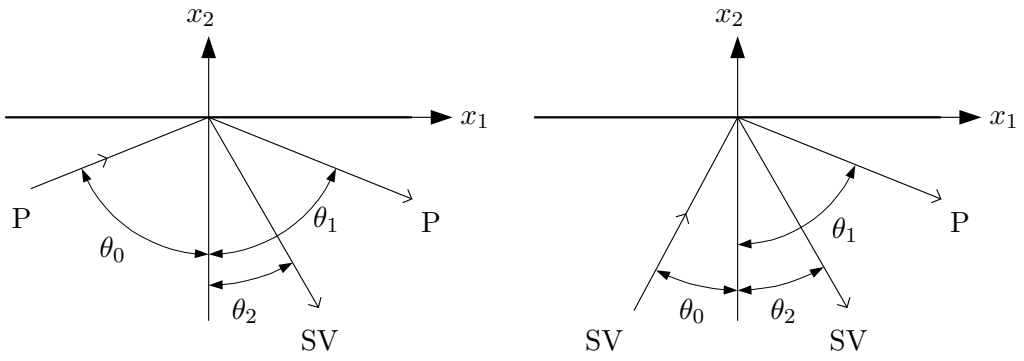
In a homogeneous, isotropic material, transverse and longitudinal wave speeds are independent of frequency — they are nondispersive.

2.1.2.1 Reflections of P and SV-waves

The wave types derived so far propagate independently in an infinite media. As soon as a finite media in the direction of propagation is considered, reflections and coupling will occur. An incident P-wave (SV-wave), which is reflected at a stress free boundary ($\sigma_{22} = 0$ and $\sigma_{21} = 0$) normally consists of both, a P-wave (SV-wave) and a SV-wave (P-wave). Figure 2.1 shows the reflections of an incident P and SV-wave. The effect of a single incident wave-type producing two different waves (after reflection from a boundary) is called mode conversion. The displacement field of a harmonic wave in the (x_1, x_2)-plane (propagating in infinite media, plane-strain case) can be expressed as

$$\mathbf{u}^{(n)} = A_n \mathbf{d}^{(n)} e^{ik_n(x_1 p_1^{(n)} + x_2 p_2^{(n)} - c_n t)}, \tag{2.17}$$

where n denotes the wave (longitudinal or transverse), $k_n = \frac{\omega}{c_n}$ is called the wavenumber of wave n and the respective wave speeds are c_n. Using these definitions, and noting that the angular frequency ω is equal for the incident and reflected waves, it



(a) Reflection of a P-wave (b) Reflection of a SV-wave

Figure 2.1: Wave reflections.

is possible to determine the relationship between the angle of the incident and the angles of the reflected waves (see Table 2.1).

Table 2.1: Angle relations for reflection on a stressfree surface

incident θ_0	reflected P θ_1	reflected SV θ_2
P	$\theta_1 = \theta_0$	$\sin\theta_2 = (c_\mathrm{T}/c_\mathrm{L})\sin\theta_0$
SV	$\sin\theta_1 = (c_\mathrm{L}/c_\mathrm{T})\sin\theta_0$	$\theta_2 = \theta_0$

Exceptions of mode conversion are the normal incidence with $\theta_0 = 0$ — in this case, the waves are reflected as themselves, and if the angle θ_0 is greater than a critical angle,

$$\theta_\mathrm{cr} = \arcsin\frac{c_\mathrm{T}}{c_\mathrm{L}}; \tag{2.18}$$

then only a SV-wave is reflected. The P-wave portion of the reflected signal degenerates into a surface wave, travelling along the surface and decreasing in amplitude with increasing depth.

2.1.2.2 Guided waves

Guided waves are waves that travel in a body (waveguide) with at least one finite and one infinite dimension. So far only single reflections have been considered, but in a wave guide multiple reflections at the surface (as shown in Figure 2.2) are possible.

As a result of mode conversion at the upper and lower boundaries, many propagating waves are reflected back and forth, resulting in an interference pattern across the body thickness that guides the waves in a certain direction. To investigate wave motion in

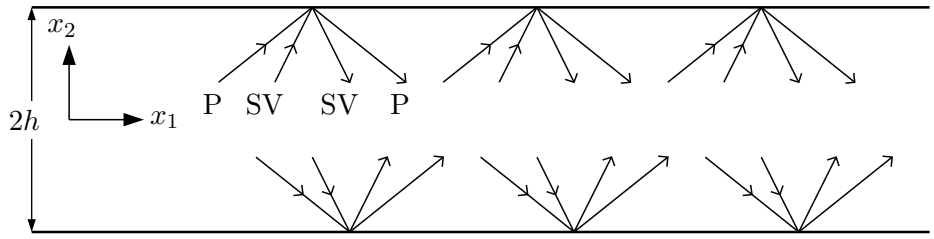


Figure 2.2: Waveguide.

an elastic wave guide, potentials in the form

$$\varphi = \Phi(x_2)e^{i(kx_1-\omega t)}, \qquad \psi_3 = \Psi(x_2)e^{i(kx_1-\omega t)}, \tag{2.19}$$

are assumed. The direction of propagation $\mathbf{p}$ is in the x_1-direction. Assuming plane strain and stress-free boundaries at $x_2 = \pm h$, one can obtain the Rayleigh-Lamb frequency equations (see Achenbach [1] for more details)

$$\frac{\tan(qh)}{\tan(ph)} = -\frac{4k^2pq}{(q^2-k^2)^2} \tag{2.20}$$

$$\frac{\tan(qh)}{\tan(ph)} = -\frac{(q^2-k^2)^2}{4k^2pq}, \tag{2.21}$$

where

$$p^2 = \frac{\omega^2}{c_L^2} - k^2, \qquad q^2 = \frac{\omega^2}{c_T^2} - k^2, \tag{2.22}$$

and $2h$ is the plate thickness. Equation 2.20 represents the symmetric Lamb modes, while Equation 2.21 provides the antisymmetric Lamb modes. (Anti)symmetric is understood to be that the displacement is (anti)symmetric to the x_1-axis and a Lamb mode is an amplitude distribution over the plate thickness that oscillates with the angular frequency ω and travels with the corresponding phase velocity $c = \frac{\omega}{k}$ obtained from an (ω, k) solution pair of the Rayleigh-Lamb spectrum (Equations 2.20 and 2.21). Note that Lamb waves are dispersive, i.e. the propagation velocity of a Lamb mode is dependent on its oscillation frequency. The Rayleigh-Lamb equations can only be solved numerically and Figure 2.3 shows a solution (dispersion curves) of the

Rayleigh-Lamb spectrum in the slowness-frequency domain. The symmetric Lamb modes are named $s_0, s_1, \ldots$ and the antisymmetric $a_0, a_1, \ldots$ starting with the mode that has the lowest angular frequency ω for a given k. The dispersion curves are

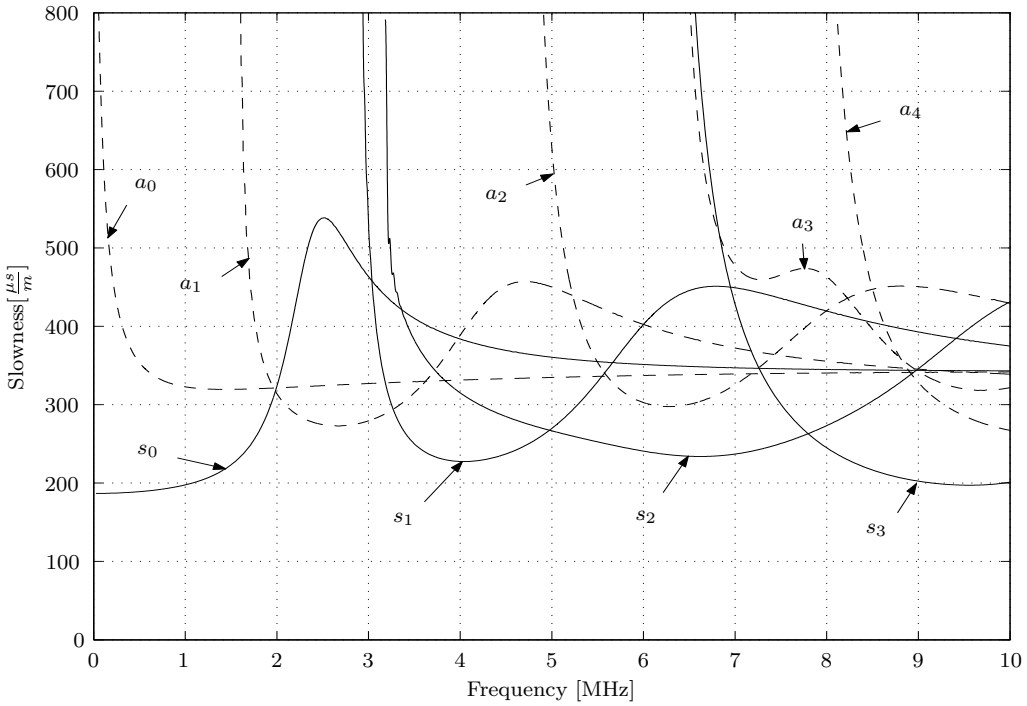


Figure 2.3: Theoretical solution in slowness-frequency domain (dispersion curves).

obtained by finding first a numerical solution in the (ω, k)-domain or (f, k)-domain respectively and then differentiating f numerically (partially with respect to k) for each of the different modes to attain the group velocity

$$c_{\mathrm{g}}(f) = \frac{2\pi \partial f}{\partial k}. \tag{2.23}$$

Group velocity as defined in Equation 2.23 describes the velocity of propagating energy and has therefore a physical meaning. In contrast, points of constant phase propagate with the phase velocity $c = \frac{\omega}{k}$. However, for nondispersive media, group and phase velocity are equal.

The energy slowness sl_e can then be obtained by the relationship

$$sl_{\mathrm{e}}(f) = \frac{1}{c_{\mathrm{g}}(f)}. \tag{2.24}$$

The expected arrival time for a specific mode at frequency f is found by

$$t(f) = \frac{sl_e(f)}{d}; \tag{2.25}$$

where d is the propagation distance source-receiver.

2.2 Signal Processing

What are the advantages of applying signal processing techniques to transient signals? It is certainly true that signal processing creates no new information — all the information is inherent in the transient signal itself. However, signal processing can provide a different view of the same information, often enabling a more accurate (and robust) interpretation.

In 1807, Fourier presented his work on the propagation of heat that included the idea that any continuous or discontinuous function of a variable can be expressed as the sum of trigonometric functions — today called the Fourier series. These results have proven to be extremely valuable, as will be demonstrated in the following sections.

2.2.1 Fourier series

The Fourier series of a T-periodic function $s(t)$ is

$$S_s(t) = \sum_{k=-\infty}^{\infty} c_k e^{ik\omega t} \tag{2.26}$$

with the coefficients c_k computed by

$$c_k = \frac{1}{T} \int_0^T s(t) e^{-ik\omega t} \mathrm{d}t \tag{2.27}$$

where $\omega = \frac{2\pi}{T}$ is called the fundamental frequency.

While Equation 2.27 is used to break down the original signal $s(t)$ into its spectral components; i.e. into components of different frequencies $\omega_k = k\omega, k \in \mathbb{Z}$, Equation 2.26 combines the different spectral components into an infinite series that represents the original signal. Clearly, the resulting Fourier series $S_s(t)$ is also T-periodic

and composed of sines and cosines with frequency ω and their harmonics[1].

For computer implementation, the discretized version of the Fourier series (DFS)

$$\tilde{s}[n] = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{S}[k] e^{\frac{-i2\pi kn}{N}} \tag{2.28}$$

can be derived [21], where

$$\tilde{S}[k] = \sum_{n=0}^{N-1} \tilde{s}[n] e^{\frac{i2\pi kn}{N}} \tag{2.29}$$

are the coefficients of the DFS and N is the sequence length. Note that in the discrete case, it is sufficient to have at most N different frequency components to completely synthesize the original sequence $s[n]$ by the DFS Synthesis Equation 2.28; this is in contrast to an infinite number required in the continuous case.

Since the DFS is not "aware" of the sampling frequency of the sequence $s[n]$, the index k for $\tilde{S}[k]$ has to be converted from the normalized frequency $f = \frac{k}{N} \in [0, 1]$ to the real frequency f. The corresponding frequency for a given $\tilde{S}[k]$ is

$$f = \frac{k}{N} f_\mathrm{s} \tag{2.30}$$

with f_s being the sampling frequency of the sequence $s[n]$. One should keep in mind that whenever continuous signals are discretized, frequencies with $f \leq \frac{f_\mathrm{s}}{2} = f_N$, where f_N is the Nyquist frequency, can only be unambiguously identified if the signal itself is limited to a frequency band below the Nyquist frequency. Otherwise aliasing can occur for higher frequencies; i.e. it is possible that a higher frequency appears as a lower frequency in the sampled domain, thus creating spurious information.

As previously stated, the Fourier series applies only to periodic functions. But what happens if the function $s(t)$ is not periodic? The Fourier transform brings relief.

2.2.2 Fourier transform

In contrast to the periodic Fourier series, the Fourier transform represents the limiting case of the series for $T \to \infty$. It allows for the representation of an aperiodic function $s(t)$ by the Fourier integral

$$s(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\omega t} S(\omega) \mathrm{d}\omega, \tag{2.31}$$

[1] since $e^{it} = \cos t + i \sin t$

where the coefficients $S(\omega)$ are defined as

$$S(\omega) = \int_{-\infty}^{\infty} e^{-i\omega t} s(t) \mathrm{d}t \tag{2.32}$$

and $S(\omega)$ is called the Fourier transform of $s(t)$. While the Fourier series represents a signal in terms of a fundamental angular frequency ω_0 and its harmonics, the Fourier transform uses a continuous angular frequency variable ω, which is related to the frequency f by

$$\omega = 2\pi f. \tag{2.33}$$

As for the Fourier series, there also exists a Fourier transform that operates on sequences. The corresponding equations are

$$s[n] = \int_{0}^{2\pi} S(\omega) e^{i\omega n} \mathrm{d}\omega \tag{2.34}$$

for synthesis and

$$S(\omega) = \frac{1}{2\pi} \sum_{n=-\infty}^{\infty} s[n] e^{-i\omega n} \tag{2.35}$$

for analysis.

The Fourier transform $S(w)$ of an input sequence $s[n]$ is thus a continuous function. It can be shown [21] that the DFS coefficients $\tilde{S}[k]$ of the sequence $\tilde{s}[n]$ are samples of the Fourier transform of $s[n]$. From this point of view, the discrete Fourier series is also called the discrete Fourier transform (DFT).

The frequency for a given $S(\omega)$ is calculated by Equation 2.33. If the signal length is a power of 2, the calculation time of the DFT can be lowered significantly by the fast Fourier transform algorithm proposed by Cooley and Tukey (see for example [21]). Since the Fourier transform and the DFT are usually complex valued (where $|S(\omega)|$, $|\tilde{S}[k]|$ are the magnitudes and $atan\left(\frac{\Im(S(\omega))}{\Re(S(\omega))}\right)$, $atan\left(\frac{\Im(\tilde{S}[k])}{\Re(\tilde{S}[k])}\right)$ the phase angles for a frequency), they are frequently presented (visualized) by the energy density spectrum which represents the energy distribution in the frequency domain and is calculated by

$$E_{\mathrm{d}} = |S(\omega)|^2. \tag{2.36}$$

Its discrete counterpart is

$$\tilde{E}_{\mathrm{d}}[k] = \frac{1}{N}\tilde{S}[k]\overline{\tilde{S}[k]}, \tag{2.37}$$

where $\overline{\tilde{S}[k]}$ is the conjugate complex of $\tilde{S}[k]$.

2.2.3 Time frequency representations (TFRs)

Unfortunately, a transient (time domain) signal, together with its Fourier transformed spectrum, still does not provide enough information for applications that require an understanding of how a signal's frequency changes *as a function of time*. Note that Fourier methods are limited to stationary signals – signals that have the same frequency content for all times. In contrast, nonstationary signals require signal processing methods that can quantitatively resolve changes in frequency content, as a function of time. To remedy this deficiency, an amazingly large number of TFRs have been developed that are capable of analyzing nonstationary signals. Many of these TFRs are subsumed in the general framework of Cohen's class [8, 7]. The spectrogram, subject of this research, also belongs to this important class.

2.2.4 Short-time Fourier transform (STFT)

The STFT

$$S_{\mathrm{stft}}(\omega, t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i\omega\tau}s(\tau)h(\tau - t)\mathrm{d}t, \tag{2.38}$$

where $h(t)$ is a window function, is based on the Fourier transform. Instead of considering a transform of the entire signal at once, the signal is chopped into a series of small overlapping pieces, and each of these pieces is windowed and then individually Fourier transformed. Its energy density spectrum

$$E_{\mathrm{d}}(\omega, t) = |S_{\mathrm{stft}}(\omega, t)|^2 \tag{2.39}$$

is called a spectrogram.

2.2.5 Heisenberg uncertainty principle

So far it seems as if TFRs can perfectly represent the time-frequency behavior of a transient signal. In theory, it appears that one measurement is enough to develop

a representation that can quantify changes in the frequency content of a signal as a function of time. Unfortunately, TFRs like the spectrogram suffer from what is known as the Heisenberg uncertainty[2], meaning (in the case of signal processing) that it is not possible to have a perfect resolution in time and frequency simultaneously.

Stating the uncertainty principle in equation form requires a set of preliminary definitions. The square norm $\|s(t)\|$ of a function $s(t)$ is defined as

$$\|s(t)\| = \left(\int_{-\infty}^{\infty} |s(t)|^2 \mathrm{d}t \right)^{\frac{1}{2}}. \tag{2.40}$$

The normalized function $s_n(t)$ is then given by

$$s_n(t) = \frac{s(t)}{\|s(t)\|}. \tag{2.41}$$

Since the square norm of a normalized function is equal to one, the squared magnitude is regarded as a probability density function enabling one to calculate the mean time of a function $s(t)$ by

$$E[t] = \int_{-\infty}^{\infty} t|s_n(t)|^2 \mathrm{d}t \tag{2.42}$$

and the mean angular frequency by

$$E[\omega] = \int_{-\infty}^{\infty} \omega|S_n(\omega)|^2 \mathrm{d}\omega, \tag{2.43}$$

where $S_n(\omega)$ is the normalized Fourier transform of the function $s(t)$. The variances for t and ω are then

$$\sigma_t^2 = \int_{-\infty}^{\infty} (t - E[t])^2 |s_n(t)|^2 \mathrm{d}t \tag{2.44}$$

and

$$\sigma_\omega^2 = \int_{-\infty}^{\infty} (\omega - E[\omega])^2 |S_n(\omega)|^2 \mathrm{d}\omega. \tag{2.45}$$

The uncertainty principle limits the possible resolutions by the inequality

$$\sigma_t^2 \sigma_\omega^2 \geq \frac{1}{4}. \tag{2.46}$$

Therefore, the standard deviation in time and frequency cannot be varied independently, but they are related to each other.

[2]In fact the term uncertainty is misleading. There is no element of chance or probability as far as signal processing is concerned; instead it is a completely deterministic phenomenon.

2.3 Image Processing

The distinction between signal and image processing techniques used in this research is not well–defined. Since the conventional reassignment method uses specific signal information to relocate time–frequency particles, it rather belongs to the class of signal processing techniques. Differential reassignment is both, depending on the point of view — it is an alternate formulation of the same reassignment signal processing method, and thus could be categorized accordingly. On the other hand, differential reassignment can be viewed as PDE-based processing of images. In this research, differential reassignment is viewed as an image processing technique, since it is not limited to images obtained from a time domain signal. In fact, all examples in Chapter 3 will use arbitrary images to demonstrate differential reassignment.

2.3.1 Reassignment

The representation of a TFR (like the spectrogram) is unsatisfactory in certain applications because a TFR cannot simultaneously have perfect resolution in both time and frequency (due to the Heisenberg uncertainty principle).

It is possible to improve the time-frequency resolution of a spectrogram with an algorithm that is referred to as reassignment. In the following sections, two formulations of the reassignment method are presented.

2.3.1.1 Conventional Reassignment

The basic idea, called the modified moving window method, was first proposed by Kodera et al. [10]. It remained somewhat unnoticed until an efficient computation method called reassignment was developed by Auger and Flandrin [4].

In reassignment, values are moved away from their point of computation to their center of gravity, thus localizing the diffuse information of the time–frequency representation. The reassignment method is not restricted to a specific TFR, but it can be applied to any time–frequency shift invariant distribution of Cohen's class. This

class of TFRs can be written according to [8, 4, 17] as

$$\mathcal{T}(x,t,\omega) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \phi_{\mathrm{tf}}(u,\Omega) WV(x,t-u,\omega-\Omega)\mathrm{d}u\frac{\mathrm{d}\Omega}{2\pi}, \tag{2.47}$$

where

$$WV(x,t,\omega) = \int_{-\infty}^{\infty} x(t+\frac{\tau}{2})\overline{x(t-\frac{\tau}{2})}e^{-i\omega\tau}\mathrm{d}\tau \tag{2.48}$$

is the Wigner–Ville distribution and ϕ_{tf} is called the kernel of the transformation. In the case of the spectrogram, the kernel amounts to

$$\phi_{\mathrm{tf}}(u,\Omega) = WV(h,u,\Omega), \tag{2.49}$$

where $h(t)$ is a normalized window.

Equation 2.47 is thus a weighting of the Wigner–Ville distribution by the kernel function ϕ_{tf} in a small neighborhood of (t,ω). The size of the neighborhood is determined by the support of ϕ_{tf}.

The center of gravity for a specific point in the frequency–time domain is calculated by

$$\hat{t}(x,t,\omega) = t - \frac{\iint u\cdot\phi_{\mathrm{tf}}(u,\Omega)WV(x,t-u,\omega-\Omega)\mathrm{d}u\frac{\mathrm{d}\Omega}{2\pi}}{\iint \phi_{\mathrm{tf}}(u,\Omega)WV(x,t-u,\omega-\Omega)\mathrm{d}u\frac{\mathrm{d}\Omega}{2\pi}} \tag{2.50}$$

and

$$\hat{\omega}(x,t,\omega) = \omega - \frac{\iint \Omega\cdot\phi_{\mathrm{tf}}(u,\Omega)WV(x,t-u,\omega-\Omega)\mathrm{d}u\frac{\mathrm{d}\Omega}{2\pi}}{\iint \phi_{\mathrm{tf}}(u,\Omega)WV(x,t-u,\omega-\Omega)\mathrm{d}u\frac{\mathrm{d}\Omega}{2\pi}}. \tag{2.51}$$

Auger and Flandrin [4] show that the reassigned coordinates $\hat{t}$ and $\hat{\omega}$ for the spectrogram can also be calculated by

$$\hat{t} = t - \Re\left(\frac{STFT_{\mathcal{T}h}(x,t,\omega)\cdot\overline{STFT_h(x,t,\omega)}}{|STFT_h(x,t,\omega)|^2}\right) \tag{2.52}$$

and

$$\hat{\omega} = \omega - \Im\left(\frac{STFT_{\mathcal{D}h}(x,t,\omega)\cdot\overline{STFT_h(x,t,\omega)}}{|STFT_h(x,t,\omega)|^2}\right) \tag{2.53}$$

for the STFT, where $STFT_h(x,t,\omega)$ is the short time Fourier transform of the signal x using a normalized window function $h(t)$; $STFT_{\mathcal{T}h}(x,t,\omega)$ and $STFT_{\mathcal{D}h}(x,t,\omega)$ are the short time Fourier transforms with $t\cdot h(t)$, $\frac{\mathrm{d}h(t)}{\mathrm{d}t}$ as their respective window functions.

The reassignment algorithm relocates each time-frequency point from the coordinates (t, ω) to a new, reassigned position $(\hat{t}(t,\omega), \hat{\omega}(t,\omega))$. This relocation can be expressed by the displacement vector field

$$r(t,\omega) = \begin{pmatrix} \hat{t}(t,\omega) - t \\ \hat{\omega}(t,\omega) - \omega \end{pmatrix} \tag{2.54}$$

defined by Equations 2.52 and 2.53.

If several points of the spectrogram are moved to the same point in the reassigned representation, their amplitudes are added up, thus conserving the energy. Calculating the reassignment of a spectrogram produces a much more distinct time-frequency resolution when compared to a non-reassigned spectrogram.

2.3.1.2 Differential Reassignment

As discussed in the previous section, conventional reassignment relocates time-frequency points away from their original (un-reassigned) location to their center of gravity. This action can be expressed as a displacement vector field, see Equation 2.54. Rather than using a displacement vector field, differential reassignment uses a velocity vector field that describes the motion of time-frequency particles. This approach can be viewed as PDE-based image processing of the spectrogram, and is based on a link between the reassignment vector field, see Equation 2.54, and a scalar potential.

First, let the STFT be denoted by $F(t,\omega)$. Then, by using the complex variable $z = \omega + it$ and its complex conjugate z^*, the STFT $F(t,\omega)$ can be rewritten as Bargmann factorization

$$F(t,\omega) = \mathcal{F}(z, z^*) e^{\frac{-|z|^2}{4}}. \tag{2.55}$$

Based on this factorization, Chassande-Motin et al. [3] deduce that

$$\mathbf{v}(t,\omega) = \nabla \log |F| - 2 \begin{pmatrix} \Im\{\partial_{z^*} \log \mathcal{F}\} \\ \Re\{\partial_{z^*} \log \mathcal{F}\} \end{pmatrix} \tag{2.56}$$

is the desired velocity field, holding over the set $\mathbb{R}^2 \setminus \{(t,\omega) | F(t,\omega) = 0\}$. An interpretation of this result is that given Equation 2.55, the Bargmann factorization of the STFT, its reassignment vector field can be decomposed into two terms. The first

term is the gradient of the scalar potential $\log|F|$, and the second term is a measure of the nonanalyticity of $\mathcal{F}$.

The use of a Gaussian window of unit variance, whose isocontours are circles in a Wigner representation, leads to the Cauchy equations $\partial_{z^*}\mathcal{F} = 0$, and thus to

$$\mathbf{v}(t,\omega) = \nabla \log|F|. \tag{2.57}$$

The reassignment vector field is then identical to the gradient of the scalar potential, $\log|F|$, for this case. As a result, the reassignment vectors will indicate the direction of steepest ascend of the STFT modulus. Additionally for this case, the phase is entirely determined by the modulus, and vice–versa. This means that the corresponding spectrogram — defined as the squared modulus of the STFT — carries as much information as the complex–valued STFT itself.

What happens if a window besides a Gaussian window of unit variance is used? In the case of an arbitrary window, the reassignment vectors follow the steepest descent direction of $-\log|F|$, modified by

$$G(t,\omega) = 2\begin{pmatrix} \Im\{\partial_{z^*} logF\} \\ \Re\{\partial_{z^*} logF\} \end{pmatrix}. \tag{2.58}$$

When a Gaussian window of arbitrary variance σ,

$$h(t) = \frac{1}{\sqrt[4]{\pi}\sqrt{\sigma}} e^{-\frac{t^2}{2\sigma^2}}, \tag{2.59}$$

is chosen, then the additional term (2.58) can be written analytically

$$G(t,\omega) = -(\sigma - \frac{1}{\sigma})\tilde{\mathbf{r}}\begin{pmatrix} \frac{t}{\sigma} \\ \sigma\omega \end{pmatrix}, \tag{2.60}$$

where $\tilde{\mathbf{r}}$ denotes the reassignment vector obtained with a unit variance Gaussian window. Note that $G(t,\omega)$ vanishes when a Gaussian window of unit variance $\sigma = 1$ is used.

Because Equation 2.56 is a link between the reassignment vector field of Equation 2.54 and a scalar potential $\log|F|$, it makes sense to look at the system which is governed by this scalar potential. From this perspective, the reassignment vector field is the velocity field that controls the motion of each time-frequency contribution $F(t,\omega)$ considered as a particle, with (t,ω) as its starting (un-reassigned) position.

Differential Reassignment is then defined by the motion equations

$$\begin{aligned} t(0) &= t_0 \\ \omega(0) &= \omega_0 \\ \frac{dt(s)}{ds} &= \hat{t}(t(s), \omega(s)) - t(s) \\ \frac{d\omega(s)}{ds} &= \hat{\omega}(t(s), \omega(s)) - \omega(s). \end{aligned} \tag{2.61}$$

In the unit variance Gaussian case, each particle converges to some maximum of $\log |F|$, thus localizing diffuse information by forming peaks and ridges.

2.3.2 Diffusion

The two following subsections present the mathematical background of both isotropic and anisotropic diffusion. As will be described in more detail, the anisotropic diffusion preserves edges, while isotropic diffusion does not.

2.3.2.1 Linear isotropic diffusion

The isotropic diffusion equation is identical to the heat equation, thus representing both the diffusion of mass and the flow of heat. In fact, an image subject to diffusion can be viewed as a heat profile subject to heat flow. In the spectrogram case, this heat profile is the energy density spectrum. Isotropic diffusion then follows the 2D linear partial differential heat equation,

$$\frac{\partial I(x,y,t)}{\partial t} = \frac{\partial^2 I(x,y,t)}{\partial x^2} + \frac{\partial^2 I(x,y,t)}{\partial y^2}. \tag{2.62}$$

As the PDE is first order in time, only the initial temperature distribution and boundary conditions have to be specified. In this research, the STFT modulus serves as the initial distribution, while zero Neumann boundary conditions ensure that heat is not diffused across image boundaries.

Since isotropic diffusion follows the heat equation, edges, which represent a sudden change in profile, are diffused — this is an undesirable feature for this research. While noise should be diffused, edges should be preserved, enabling the formation of sharp ridges. This demand necessitates the consideration of anisotropic diffusion.

2.3.2.2 Nonlinear anisotropic diffusion

Nonlinear anisotropic diffusion was introduced by Perona and Malik [22]. This diffusion process encourages intraregion smoothing and inhibits interregion smoothing, thus preserving edges. Mathematically, the 2D process is defined as

$$\frac{\partial I(x,y,t)}{\partial t} = \nabla \cdot [c(x,y,t) \cdot \nabla I(x,y,t)]. \tag{2.63}$$

In this research, $I(x,y,t)$ is the squared STFT modulus. The diffusion function $c(x,y,t)$ is a monotonically decreasing function of the image gradient magnitude

$$c(x,y,t) = f(|\nabla I(x,y,t)|). \tag{2.64}$$

This allows for locally adaptive diffusion strengths: edges are selectively smoothed or enhanced based on an evaluation of the diffusion function. Although any monotonically decreasing continuous function of ∇I would suffice as a diffusion function, the two functions

$$c_1(x,y,t) = e^{-(\frac{|\nabla I(x,y,t)|}{\kappa})^2} \tag{2.65}$$

$$c_2(x,y,t) = \frac{1}{1 + (\frac{|\nabla I(x,y,t)|}{\kappa})^{1+\alpha}}, \alpha > 0 \tag{2.66}$$

have been proposed by [22] and are shown in Figure 2.4(a).

The parameter κ is referred to as the diffusion constant or the flow constant. Obviously, the behavior of the diffusion process depends on κ. To clarify the effect of κ, and the diffusion function, on the diffusion process, it is helpful to define the flow function

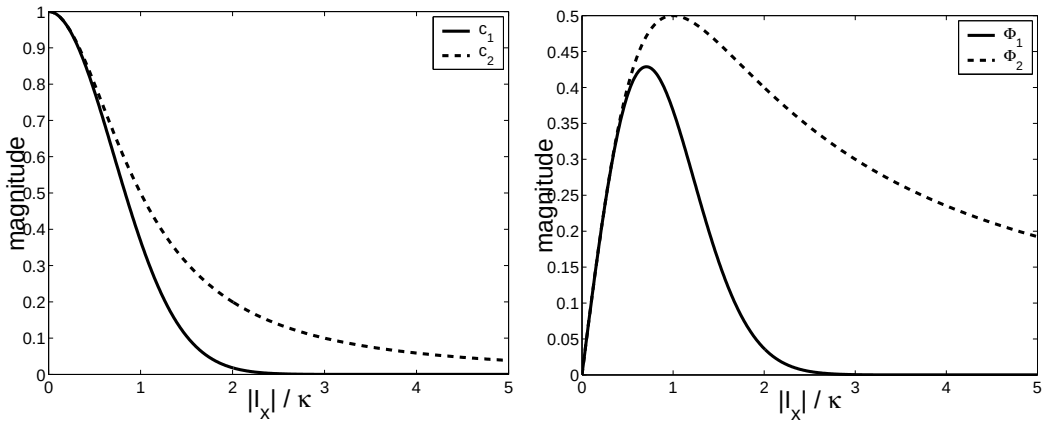
$$\Phi(x,y,t) = c(x,y,t) \cdot \nabla I(x,y,t). \tag{2.67}$$

Equation (2.63) can then be rewritten as

$$\frac{\partial I(x,y,t)}{\partial t} = \nabla \cdot \Phi(x,y,t). \tag{2.68}$$

This formulation will also be useful for developing a discrete implementation of the diffusion filter. The flow functions, Φ_1 and Φ_2, corresponding to the diffusion functions, c_1 and c_2, are plotted in Figure 2.4(b).

Notice that flow increases with the gradient to the point where $|\nabla I(x,y,t)| \simeq \kappa$, then decreases to zero. This behavior implies that the diffusion process maintains



(a) Diffusion functions

(b) Flow functions

Figure 2.4: Diffusion and flow functions.

homogeneous regions, where $|\nabla I(x, y, t)| \ll \kappa$, since little flow is generated. Similarly, edges are preserved because the flow is small in regions where $|\nabla I(x, y, t)| \gg \kappa$.

The greatest flow is produced when the image gradient magnitude is close to the value of κ — by choosing κ to correspond to gradient magnitudes produced by noise, the diffusion process can be used to reduce noise in images. Assuming an image contains no discontinuities, object edges can be enhanced by choosing a value of κ slightly less than the gradient magnitude of the edges.

CHAPTER 3

Derivation of the algorithm

One goal of this thesis is to improve the differential reassignment algorithm, proposed by Chassande-Matin et. al. [6, 3], for spectrogram reassignment. This chapter describes how the implementation of the improved algorithm's components – reassignment and diffusion – are individually derived and implemented using MATLAB.

The first section deals with differential reassignment. Three different versions have been developed and coded. The first one is a direct implementation of the differential reassignment velocity vector field, Equation 2.57. The second version uses different trajectories and velocities to move energy towards the maxima of the differential reassignment scalar potential field, $\log I^0$. This approach leads to an effective computational algorithm, as well as an advantageous localization evolution with time. Unfortunately, due to the use of different trajectories, this version lacks the well-established mathematical foundation provided by differential reassignment theory. The third version uses different velocities, but does follow differential reassignment trajectories, and is thus consistent with the theory. Examples are used to demonstrate the correct implementation, and to allow for comparison of the different versions.

The implementation of nonlinear anisotropic diffusion is elucidated, and its capabilities being demonstrated by an example in the second section.

The third section describes the combination of differential reassignment with anisotropic diffusion.

The final section presents the applied noise removal method, and an example is used to demonstrate the efficiency of the proposed approach.

3.1 Differential reassignment algorithm

The theoretical background of the differential reassignment algorithm was provided by Section 2.3.1.2. Several approaches to implement the differential flow Equation 2.61 of time-frequency particles have been considered in this research. In the following subsections, three alternative strategies are presented. The first version exactly follows the differential reassignment vector field Equation 2.57, the second uses different trajectories and velocities, and the third uses the same trajectories, but different velocities.

All alternatives, and thus this research, use Equation 2.57 in a fundamentally different way than proposed by Chassande–Mottin et. al [3]. It is essential for the reader to understand the basic differences in its interpretation and usage.

Chassande–Mottin and co–workers derived Equation 2.57 based on the STFT. Their variable, F, denotes the complex–valued STFT. Equation 2.57 is exclusively valid for STFT's obtained with a unit variance Gaussian window, whose isocontours are circles in a Wigner representation. Modification terms are needed if any other window is used. Chassande-Mottin et al. are concerned with deriving the continuous time trajectory of a time-frequency particle.

This research uses the *idea* of a logarithmic scalar potential to manipulate images. The STFT, denoted F in Equation 2.57 becomes the abstract, two dimensional matrix I of positive real cell values — an image. Equation 2.57 is valid if the distance between image cells in both dimensions are equal — for convenience, $\triangle x = \triangle y = 1$. Scaling terms are needed if unequal cell distances are used. This research is concerned with image evolution through iterative addition and subtraction of cell values. Hence, this study takes a grid-based Eulerian approach, rather than a particle-based Langrangian.

Derivation of the individual algorithms is performed using arbitrary images. Once the algorithm is developed, Chapter 4 applies the combined algorithm to spectrograms. It is important to understand that the algorithm treats these spectrograms as an abstract two-dimensional matrix of positive real cell values with $\triangle x = \triangle y = 1$.

3.1.1 First version

3.1.1.1 Implementation

The partial differential equation which describes the movement of energy particles is based on the transport equation

$$I_t + v^x I_x + v^y I_y = 0. \tag{3.1}$$

The particle velocity v is given by the reassignment vector Equation 2.57, plugging this equation into Equation 3.1 yields

$$\frac{\partial I}{\partial t} + \frac{\partial \log |I^0|}{\partial x} \cdot \frac{\partial I}{\partial x} + \frac{\partial \log |I^0|}{\partial y} \cdot \frac{\partial I}{\partial y} = 0. \tag{3.2}$$

The initial image I^0 is assumed to be non-negative, which is in particular true for all spectrograms, so the modulus in Equation 3.2 can be omitted. Hence, the PDE

$$\frac{\partial I}{\partial t} + \frac{\partial \log I^0}{\partial x} \cdot \frac{\partial I}{\partial x} + \frac{\partial \log I^0}{\partial y} \cdot \frac{\partial I}{\partial y} = 0 \tag{3.3}$$

describes the numerical reassignment flow of Equation 2.61. To implement this partial differential equation, its first derivatives have to be approximated.

There are a number of ways to produce numerical algorithms for differentiation, such as invoking the limit process used in the analytical definition. Given the function $f(x)$, its first derivative can be two-point approximated either by the forward difference

$$f'(x) \simeq \frac{f(x+h) - f(x)}{h} \tag{3.4}$$

or by the backward difference

$$f'(x) \simeq \frac{f(x) - f(x-h)}{h} \tag{3.5}$$

or by the central difference

$$f'(x) \simeq \frac{f(x+h) - f(x-h)}{2h}. \tag{3.6}$$

For the first derivatives appearing in Equation 3.3, a central difference approximation following Equation 3.6 is only suitable for derivatives of the scalar potential $\log I^0$. This is because the symmetric two-point central difference approximation

uses only the two adjacent points — the central point, for which the derivative is computed, is left out of the computation. The goal of the reassignment algorithm is to produce steep ridges, consequentially sharp edges occur. A central difference (essentially an average of a forward and backward difference) would smooth out these sudden changes in the first derivative. Furthermore, ridges having a width of a single cell would have no impact on the first derivative on top of the ridge (calculated perpendicular to the ridge) because the ridge itself is then ignored by the two-point central approximation. Hence, either the forward (Equation 3.4) or backward (Equation 3.5) difference must be used for first derivative approximation. This raises the obvious problem of a correct choice between the two alternatives, and leads to the use of an upwind method to compute the derivatives of the actual image surface.

Because reassignment vectors always point toward the maxima of the STFT modulus, it is known that information — mass, time-frequency particles — only travel "up the hill" of the scalar potential. This knowledge of the direction of information travel can be used to decide whether a forward or backward difference is more appropriate. It only makes sense, and it is necessary for stability, to compute the first derivative exclusively in the direction of information flow. If the flow is from left to right, then the backward difference has to be used; if it is from right to left, only the forward difference yields satisfactory results. This makes sure that only the scalar potential on the side of information flow into the point of interest is used for the approximation, not the potential on the other side.

To make a decision in favor for one of the two possible approximations, the velocity vector field v has to be considered. With D_+ indicating the forward difference (Equation 3.4) and D_- indicating the backward difference (Equation 3.5), the following two cases are possible, presented for the x-direction:

- $v_x < 0$. This corresponds to a wave travelling to the left, hence the forward difference D_+ has to be chosen.
- $v_x \geq 0$. This corresponds to a wave travelling to the right, hence the backward difference D_- has to be chosen.

These derivative approximations can be used to implement the energy particle

movement PDE, Equation 3.3. Since this research concerns a two-dimensional scalar equation, dimensional splitting can be applied. Movement of energy particles is calculated alternating between x- and y-dimensions, using the Donor-Cell Upwind Method for Advection, compare LeVeque [16].

3.1.1.2 Demonstration

This section presents a two-dimensional example using the algorithm described in the previous section. A quadratic image with a resolution of 100x100 cells is chosen. The analytic, initial image follows the Gaussian function

$$I(x,y) = \frac{1}{\pi^4\sqrt{\sigma}} \cdot e^{-\frac{(x-50)^2}{2\sigma^2}} + \frac{1}{\pi^4\sqrt{\sigma}} \cdot e^{-\frac{(y-50)^2}{2\sigma^2}} \tag{3.7}$$

with standard deviation $\sigma = 10$ and is shown in Figure 3.1. Figure 3.2 plots the scalar potential function $\log I(x,y)$ calculated from the initial image. The sequence of Figures 3.3(a) to 3.3(f) shows the evolution of the image after 500, 1,000, 2,000, 5,000, 10,000 and 100,000 iterations, respectively.

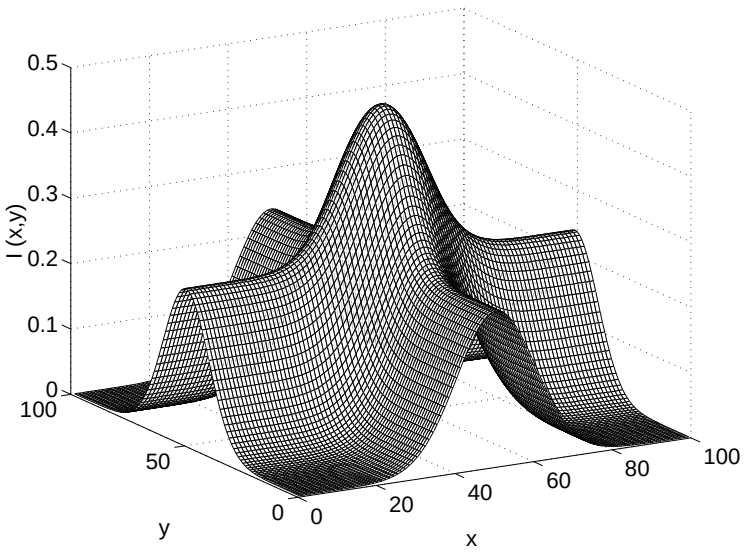


Figure 3.1: Image.

First, consider Figure 3.2. Recall that reassignment vectors point up the gradient of this function, which remains constant with time, they do not follow the steepest

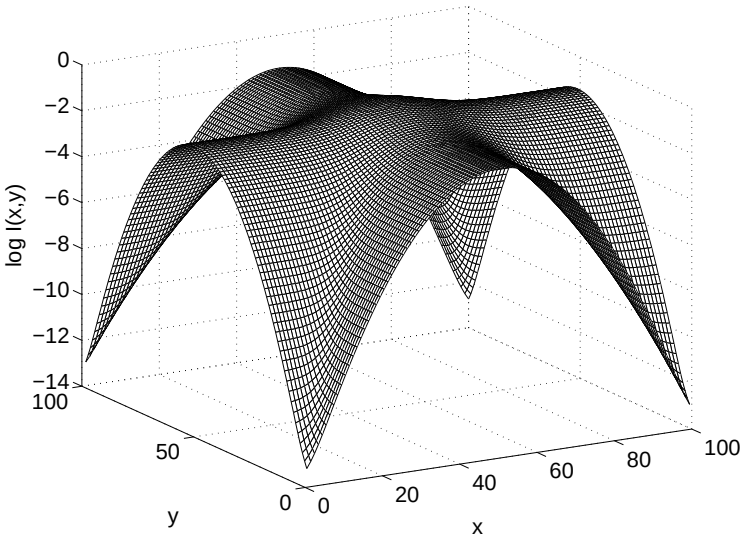
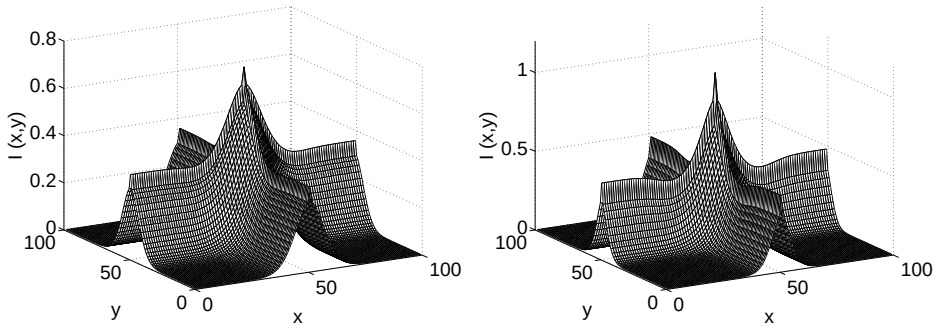


Figure 3.2: Scalar potential function.

ascend of the actual image. Note that the scalar potential surface amplifies derivatives in low energy regions. Hence, reassignment vector magnitude is far greater in low mass regions than in high mass regions. Thus, mass that is further away from a ridge moves faster towards the ridge than mass that is closer to the ridge.

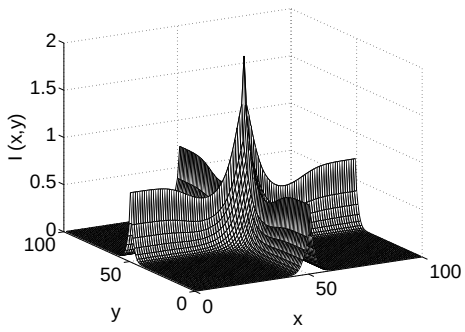
A peak lies at the intersection of the two ridges, so the top of both ridges is not flat, but ascends toward the central peak. Hence, mass that is already located on top of a ridge keeps on moving toward that peak. Figure 3.4 plots the derivative of the scalar potential function, i.e. Figure 3.2, along a ridge. As can be easily seen, mass on either side of the center is moved toward it, but unfortunately the velocities differ along the ridge. Mass that is located about 13 cells away from the center approaches it fastest. Consequently, at this distance from the center, mass on top of a ridge travels toward the center faster than it is being "resupplied" from the far ridge ends. This explains the inevitable formation and deepening of dips in these ridge regions, which are not present in the original image. The formation of dips can be followed by examining the sequence of Figures 3.3(a) to 3.3(f).

Image evolution will eventually approach its steady state. The ridges disappear as their energy has been relocated to the central peak. This steady state is consistent

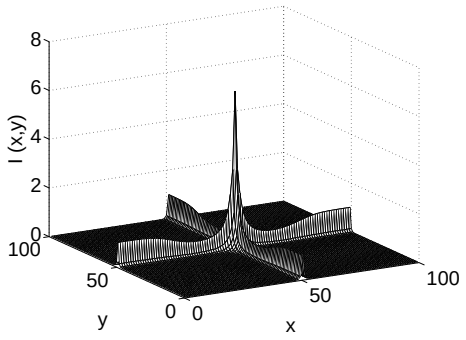


(a) 500 Iterations

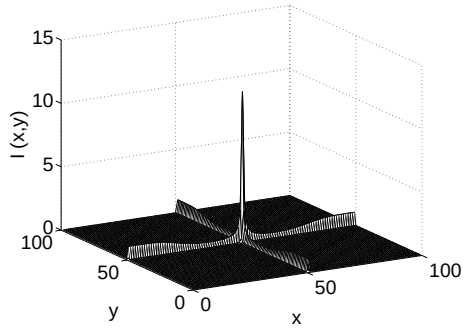
(b) 1,000 Iterations



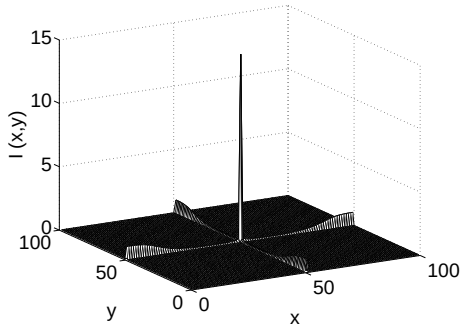
(c) 2,000 Iterations



(d) 5,000 Iterations



(e) 10,000 Iterations



(f) 100,000 Iterations

Figure 3.3: Image evolution of the first differential reassignment version.

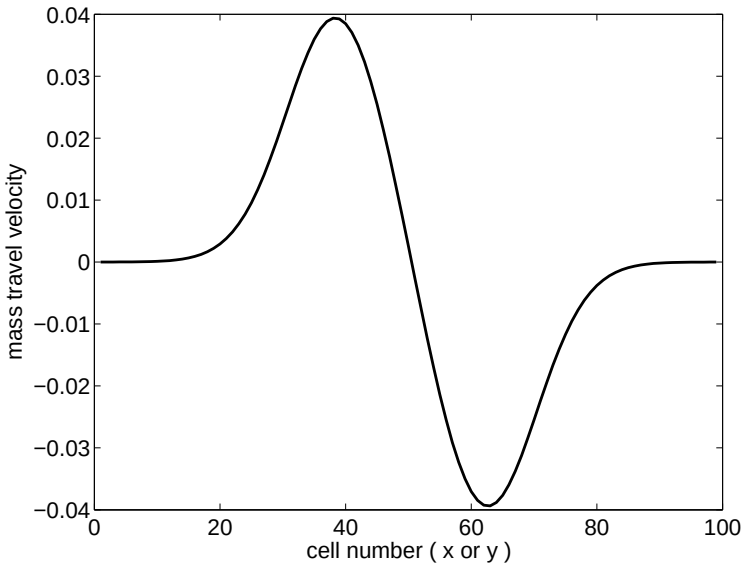


Figure 3.4: Energy travel velocity along a ridge.

with the scalar potential Figure 3.2. Close observation reveals that all trajectories finally end in the image center, after following the potential's steepest ascent.

Of course, this image evolution is highly unsatisfactory; information should be localized along smooth ridges. At certain evolution stages, such as the stage of Figures 3.3(c) or 3.3(d), the reassigned image might be considered acceptable if dip and peak formation is tolerated. However, later stages, including the steady state, are not acceptable.

Depending on their individual shape, different ridges of the same image are reassigned with different speeds. Thus, ridges that were already created in early iterations have to remain in a satisfactory, stable steady state, to allow information localization in other image regions.

Essentially, unsatisfactory image evolution results from inhomogeneous mass movement along a ridge. Countering this movement should lead to a satisfactory, stable steady state. To achieve this, an opposing force is introduced. In the steady state, undesired mass movement will be balanced out by this opposing force. Nonlinear anisotropic diffusion is capable of providing this opposition force.

3.1.2 Second version

3.1.2.1 Implementation

The second version (developed in this research) uses different trajectories and velocities from those predicted by differential reassignment theory. Since energy movement still converges towards the maxima of the scalar potential function $\log I^0$, energy is localized as desired. However, this process is no longer "differential reassignment," since it is not based upon differential reassignment theory.

Instead of following the transport Equation 3.1, image evolution follows

$$\frac{\partial}{\partial t} I + v = 0, \tag{3.8}$$

where v is the reassignment velocity vector field, based on Equation 2.57. Thus, energy particles are still moved toward the maxima of the scalar potential $\log I^0$. However, since the reassignment transport Equation 3.1 is replaced with Equation 3.8, the trajectories of energy travel are not necessarily identical to the differential reassignment trajectories. The algorithm is comprised of five steps, which are described next.

In the first step, mass flow from neighboring cells into the actual cell is computed according to the reassignment vector field. In the one-dimensional case, each cell has two neighbors, so two flows must be computed. The flow from the left neighbor into the cell is labelled Φ^{west}, while the flow from the right neighbor into the cell is labelled Φ^{east}. In the two-dimensional case, the flow from either four or eight neighboring cells can be computed, depending on if the four diagonally adjacent cells are considered neighbors or not. In this research, neighborship is defined as only vertically or horizontally adjacent, following the 4-neighborhood scheme. The four flows from neighboring cells into the actual cell are labelled Φ^{west}, Φ^{east}, Φ^{north}, and Φ^{south}, corresponding to the left, right, upper, and lower neighbors, respectively. The flow follows the reassignment vector Equation 2.57. Hence,

$$\Phi^{\text{west}} = \frac{\log I^0(x, y) - \log I^0(x - h, y)}{h} \tag{3.9}$$

and

$$\Phi^{\text{north}} = \frac{\log I^0(x, y) - \log I^0(x, y - h)}{h} \tag{3.10}$$

are the horizontal and vertical backward differences of $\log I$, respectively. On the other hand,

$$\Phi^{\text{east}} = \frac{\log I^0(x,y) - \log I^0(x+h,y)}{h} \tag{3.11}$$

and

$$\Phi^{\text{south}} = \frac{\log I^0(x,y) - \log I^0(x,y+h)}{h} \tag{3.12}$$

are the respective negative (inverse direction, since positive flow is defined into the actual cell) horizontal and vertical forward differences of $\log I^0$. Note that since the vector field is constant throughout the entire algorithm, this first step can be computed before the iterations start.

In the second step, all image cells containing a zero are detected, and the flow out of these cells into their neighbors is set to zero. While the algorithm presented in the previous Section 3.1.1 lets low energy image cells slowly approach, but not reach zero, later stages of this algorithm set these cells exactly to zero. This avoids the declining step size problem. A zero cell represents a global minimum of zero mass, so mass cannot flow out of the cell. Because a flow Φ^i is only dependent on the gradient of the scalar potential function, the flow out of a zero cell must be actively set to zero. Let (x_0, y_0) denote a zero cell, then all flows *out* of the zero cell are eliminated by

$$\Phi^i(x_0,y_0) = \max(\Phi^i(x_0,y_0), 0). \tag{3.13}$$

To avoid mass creation, these flows have to be eliminated in the neighboring cells, too. Since flows out of a zero cell are incoming for its neighbors, their corresponding flows are set to zero by

$$\begin{aligned}
\Phi^{\text{west}}(x_0+h,y_0) &= \min(\Phi^{\text{west}}(x_0+h,y_0), 0) \\
\Phi^{\text{east}}(x_0-h,y_0) &= \min(\Phi^{\text{east}}(x_0-h,y_0), 0) \\
\Phi^{\text{north}}(x_0,y_0+h) &= \min(\Phi^{\text{north}}(x_0,y_0+h), 0) \\
\Phi^{\text{south}}(x_0,y_0-h) &= \min(\Phi^{\text{south}}(x_0,y_0-h), 0).
\end{aligned} \tag{3.14}$$

It may not be intuitively clear why mass flow *into* a zero cell is possible and must be allowed for the algorithm to function properly. The reason is that reassignment vectors follow the steepest ascent of the scalar potential function Equation 2.57 (based on the *initial* image), they do *not* point up the actual image surface. It may well

happen that with ongoing iterations, the image evolves in such a way that mass travels "down the hill" of the actual image surface, while still travelling "up the hill" of the scalar potential function. In short, mass flow *into* a zero cell must be allowed, and mass flow *out* of a zero cell must be prohibited. Section 3.1.2.2 contains an example illustrating the phenomenon of mass flow into zero cells.

In the third step, the optimal step size is determined. The optimal step size is the largest step size that is smaller or equal to the largest allowable step size, that still does not produce negative image cells. Let n be the iteration number and $\triangle t$ the step size, then

$$I^{n+1} = I^n + \triangle t \cdot I^{\text{flow}}, \tag{3.15}$$

where

$$I^{\text{flow}} = \Phi^{\text{west}} + \Phi^{\text{east}} + \Phi^{\text{north}} + \Phi^{\text{south}} \tag{3.16}$$

is a matrix representing the movement of energy particles due to the flow between neighboring cells. Note that the conservation constraints

$$\sum_x \sum_y \left(\Phi^{\text{west}} + \Phi^{\text{east}}\right) = \sum_x \sum_y \left(\Phi^{\text{north}} + \Phi^{\text{south}}\right) = 0, \tag{3.17}$$

and therefore

$$\sum_x \sum_y I^{\text{flow}} = 0, \tag{3.18}$$

are satisfied. Equation 3.17 states that neither the horizontal nor the vertical flow creates or destroys mass, both are independently conservative. Accordingly, the sum I^{flow} of both flow directions — and with it the algorithm — must be conservative, too, as expressed by Equation 3.18. Let $\triangle t_{\max}$ denote the largest allowable step size, and consider the element-wise quotients

$$I^{\text{q}}(x,y) = \frac{I^n(x,y)}{\triangle t_{\max} \cdot I^{\text{flow}}(x,y)}, \tag{3.19}$$

where

$$I^{\text{q}}(x_0,y_0) = 0 \tag{3.20}$$

for zero mass, zero flow cells. These quotients are the quotients of every single (non-negative) image value divided by the amount that would be added to it in the actual iteration step, according to Equation 3.15, assuming maximum step size. These quotients can be used to determine the optimal step size, as

- $I^{\mathrm{q}}(x,y) = 0$ resembles a zero mass cell (by definition due to Equation 3.20). Since the flow is also zero, there is nothing to worry about.

- $I^{\mathrm{q}}(x,y) > 0$ indicates that mass is added to this cell (both the actual cell value and the flow added to it are positive). Again, there is nothing to worry about.

- $I^{\mathrm{q}}(x,y) < -1$ indicates a cell that loses mass, but stays positive (the actual positive cell value is greater than the absolute value of the negative flow added to it). Once again, there is nothing to worry about.

- $I^{\mathrm{q}}(x,y) = -1$ resembles a cell that will be set to zero in this iteration step (the actual cell value is positive, the added flow has the same absolute value, but negative sign). Since this boundary case is allowed, there is nothing to worry about.

- $-1 < I^{\mathrm{q}}(x,y) < 0$ indicates a cell that will become negative (the actual positive cell value is less than the absolute value of the negative flow added to it). This condition is not allowed, since each cell has to have a non-negative value at all times.

The goal is to find the optimal scaling factor $0 < \alpha \leq 1$ that is multiplied by the largest allowable step size $\triangle t_{max}$ to yield the optimal step size

$$\triangle t_{\mathrm{opt}} = \alpha \cdot \triangle t_{\mathrm{max}}. \tag{3.21}$$

Let

$$\mathcal{N} = \{I^{\mathrm{q}}(x,y) | -1 < I^{\mathrm{q}}(x,y) < 0\} \tag{3.22}$$

be the set of all quotients I^q belonging to image cells that would become negative if the largest allowable step size $\triangle t_{\mathrm{max}}$ is used. The optimal scaling factor is then given by

$$\alpha = |\min \mathcal{N}|. \tag{3.23}$$

This choice of α ensures that the iteration does not produce negative-valued image cells. If

$$\mathcal{N} = \emptyset, \tag{3.24}$$

then the optimal step size equals the largest allowable step size, corresponding to

$$\alpha = 1. \tag{3.25}$$

Note that if and only if $\alpha < 1$, at least one nonzero cell will become a zero cell after completion of the iteration step. This is because α shortens the step size just enough to let the most "critical" cell(s) become zero. If this reassignment algorithm is superimposed by a diffusion algorithm, then the scaling factor α must be kept in mind to adjust the diffusion step size accordingly.

In the forth and final step, the actual iteration

$$I^{n+1} = I^n + \triangle t_{\text{opt}} \cdot I^{\text{flow}} \tag{3.26}$$

is performed, its continuous time equivalent being

$$\frac{\partial}{\partial t} I(x, y, t) \approx \Phi^{\text{west}} + \Phi^{\text{east}} + \Phi^{\text{north}} + \Phi^{\text{south}}. \tag{3.27}$$

As will be demonstrated in Section 3.1.2.2, the algorithm converges to a final, reassigned, stable steady state. This encourages stopping the iterative algorithm as soon as the steady state, or a state satisfactorily close to it, is reached.

Further, it is recognized that in images with large, low mass regions, most early iteration steps have a tiny step size, and serve primarily to create new zero cells in these regions. Thus, the algorithm can be accelerated by setting all low mass image cells below a certain limit δ to zero,

$$I(x, y) = 0 \quad \forall \quad I(x, y) \leq \delta. \tag{3.28}$$

Of course, mass is being destroyed by this action, constituting an obvious violation of the conservation requirement. However, if the limit δ is chosen to be small enough, the loss of mass might well be tolerable. A very small δ is even necessary, since machine precision does not scale zero cells exactly to zero, but usually leaves them with tiny values.

Furthermore, the δ parameter is important for noise removal purposes. Section 3.4 describes its usage, defining a lower boundary for the parameter.

3.1.2.2 Demonstration

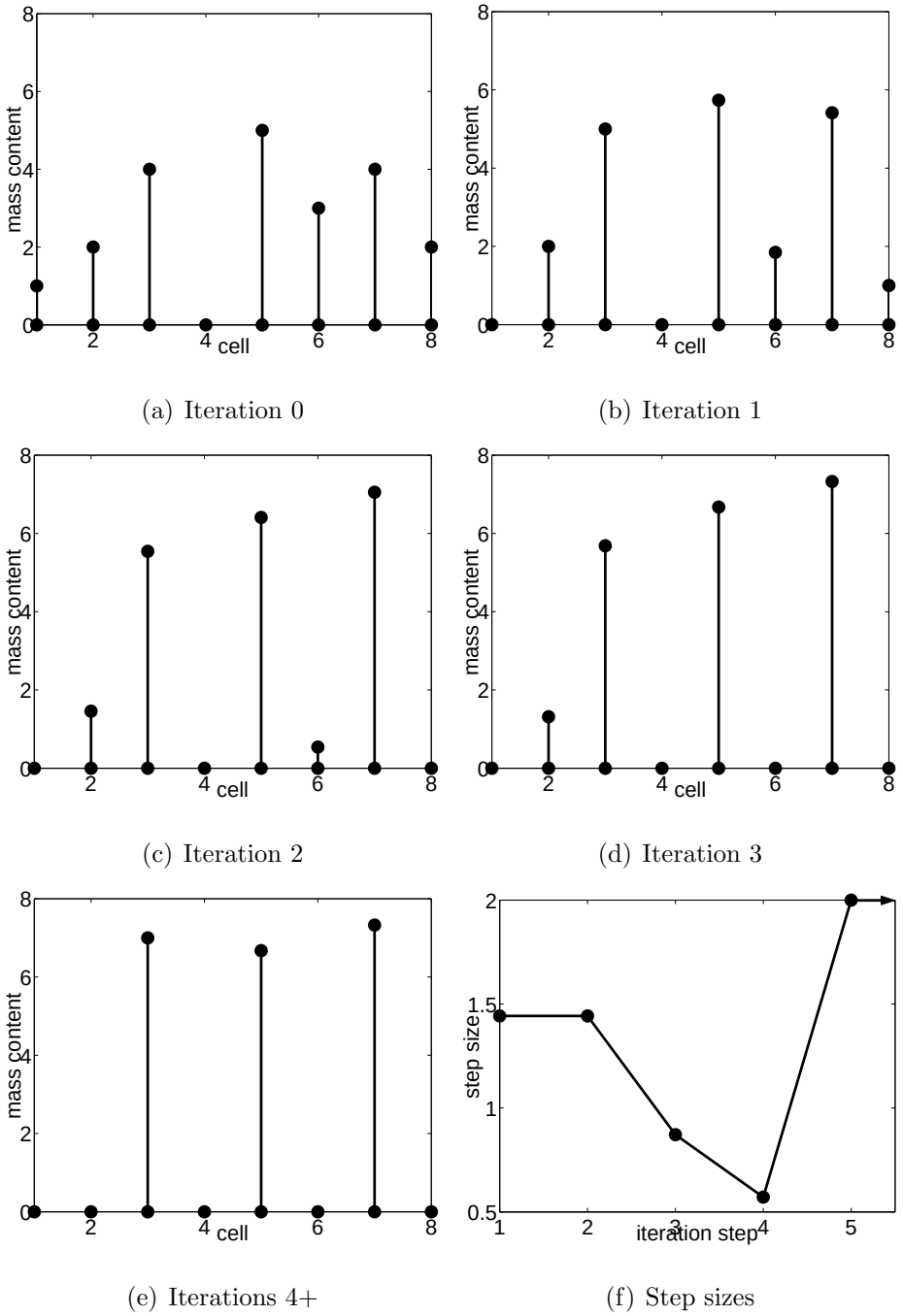
This subsection demonstrates the correct implementation and effectiveness of the algorithm described in Section 3.1.2. The algorithm is first applied in its one-dimensional form, and then in its two-dimensional form.

Select an array of eight cells as the one-dimensional example. The maximum step size is $\triangle t_{\max} = 2$. Table 3.1 and Figures 3.5(a) to 3.5(f) show the original array together with the reassigned array and all intermediate iteration steps, along with the step sizes.

Table 3.1: Demonstration of 1-D differential reassignment

iteration	step size	cell 1	cell 2	cell 3	cell 4	cell 5	cell 6	cell 7	cell 8
0	-	1	2	4	0	5	3	4	2
1	1.4	0	2	5	0	5.7	1.9	5.4	1
2	1.4	0	1.5	5.5	0	6.4	0.6	7	0
3	0.9	0	1.3	5.7	0	6.7	0	7.3	0
4	0.6	0	0	7	0	6.7	0	7.3	0
5	2	0	0	7	0	6.7	0	7.3	0
...	...	...	...	...	...	...	...	...	...

Note that the sum of array values remains 21 throughout the algorithm, confirming its conservative nature. A closer look at the different iteration steps reveals that until the fourth iteration step, the step size is smaller than the maximum step size, and each iteration produces a new zero cell. The original array consists of two "mountains," the left with a single peak, and the right with two peaks of different heights. The left "mountain" has an initial total mass of $1+2+4 = 7$ units, which is entirely reassigned to a single peak. The right "mountain" has an initial total mass of $5+3+4+2 = 14$ units, which is unequally reassigned to two peaks of different height. The entire mass content of cell 8 is reassigned to the right peak of cell 7. Since cell 6 lies in between the two peaks, its mass content is entirely, but unequally reassigned to the peaks on either side. Also note that the final, reassigned state is a stable steady state. As soon as it is reached in the fourth iteration step, the step size is set to its maximum, but the reassigned state does not change with subsequent iterations. In addition, note that since four different cells have to become zero, cells for the entire mass to



(a) Iteration 0

(b) Iteration 1

(c) Iteration 2

(d) Iteration 3

(e) Iterations 4+

(f) Step sizes

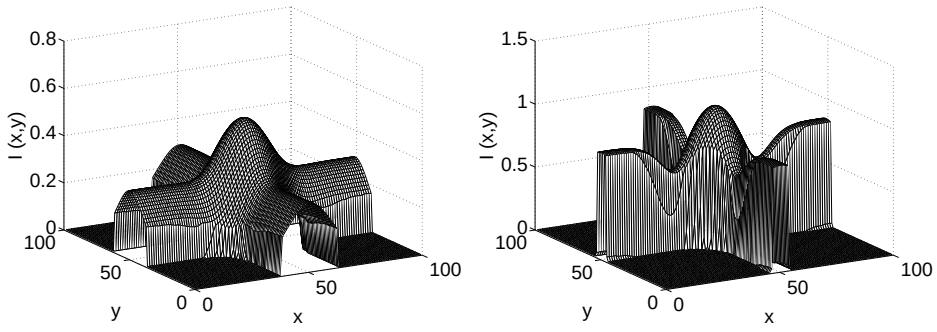
Figure 3.5: 1-D example of the second differential reassignment version.

be allocated to the three peaks, and since the steady state is reached in the fourth iteration step, the number of iterations is minimum.

The image shown in Figure 3.1 is again used as a two-dimensional example. The scalar potential function is displayed in Figure 3.2. The algorithm uses a maximum step size of $\triangle t_{max} = 5$. For acceleration purposes, cell values below $\delta = 10^{-3}$ are set to zero. The sequence of Figures 3.6(a) to 3.6(f) show the evolution of the image after 200, 500, 1,000, 1,200, 1,500 and 300,000 iterations, respectively.

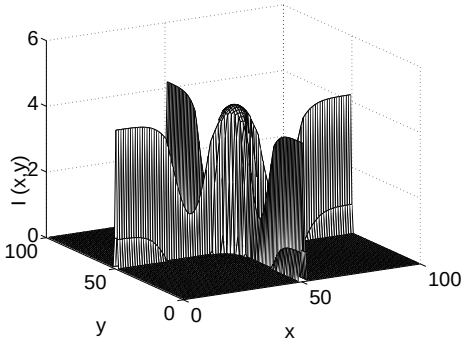
Compare this image evolution to the image evolution obtained with the first version, depicted in Figures 3.3(a) to 3.3(f). In this second version, ridges are much more pressed together, thus creating much steeper ridge edges than provided for by the first version. This is an advantage, because anisotropic diffusion will be superimposed to this reassignment movement. Steep ridge edges make it easier for the diffusion algorithm to only diffuse energy along ridges, while preserving the edges. However, the second version intensifies dip formation along ridges. In a combined algorithm, the diffusion part will be responsible to counter this undesired evolution. Luckily, steep ridge edges support the necessary diffusion performance to oppose dip formation along ridges. Again, intermediate evolution stages, such as Figures 3.6(b) or 3.6(c), can be considered satisfactory if dip formation is tolerated. Especially Figure 3.6(c) also provides narrower ridges, and thus a better energy localization than is obtained using the first version.

The image sequence of Figures 3.6(a) to 3.6(f) also illustrates the phenomenon of energy movement "down the hill" of the actual image surface. Dip formation, caused by inhomogeneous movement of energy along a ridge toward the central peak, eventually leads to Figure 3.6(d). From here on, the dips have become so deep that the ridges themselves start to disappear. This process begins with the creation of a first "zero cell" in the middle of a ridge. Recall that energy travel follows the steepest ascent of the constant scalar potential function, it does not follow the steepest ascent of the actual image. Thus, along a ridge, movement is always toward the center, a fact that can be obtained from Figures 3.2 and 3.4. This means that along the far ridge ends, energy moves "down the hill into the dip" of the actual image surface, a condition already addressed in Section 3.1.2. Note that eventually, energy from the

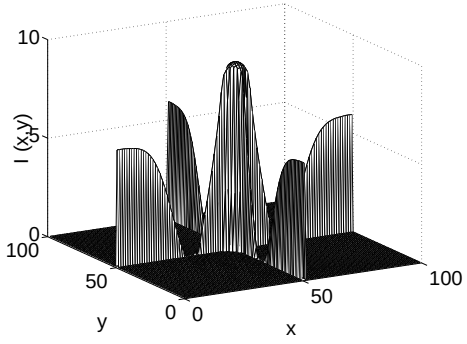


(a) 200 Iterations

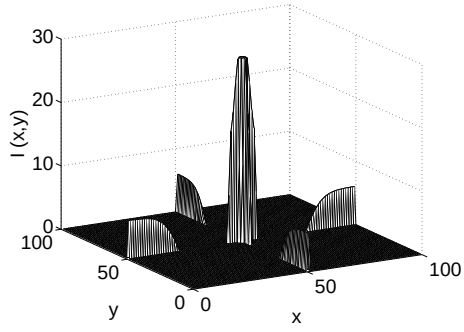
(b) 500 Iterations



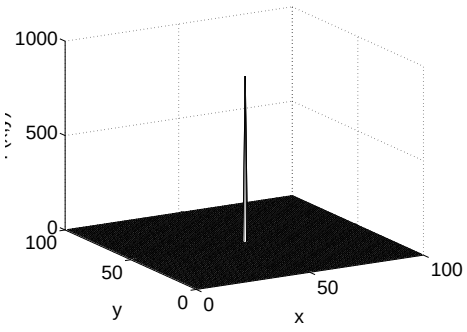
(c) 1,000 Iterations



(d) 1,200 Iterations



(e) 5,000 Iterations



(f) 300,000 Iterations

Figure 3.6: Image evolution of the second differential reassignment version.

far ridge ends has to reach the image center, according to the steepest ascent of the scalar potential function. Inevitably, this energy has to cross the "zero cells" which now divide the two ridge fractions, illustrating why energy flow *into* zero cells had to be allowed in Section 3.1.2. Technically, those cells depicted as "zero cells" in this paragraph are not real zero cells. Because energy flow into zero cells was allowed, these cells form a chain of very low energy cells delivering energy from the ridge ends to the image center, as can be seen in Figure 3.6(e).

3.1.3 Third version

3.1.3.1 Implementation

So far, two alternate forms to implement differential reassignment have been developed. The first algorithm (presented in Section 3.1.1) is the direct implementation of the mathematical representation of differential reassignment. The second algorithm (presented in Section 3.1.2) changes both trajectories and velocity of energy travel. Strictly speaking, this algorithm no longer represents differential reassignment, so it lacks the mathematical foundation present in differential reassignment theory.

This third version follows the reassignment vector field Equation 2.57, and thus guarantees that energy is moved along differential reassignment trajectories. Hence, this algorithm works on the well-established mathematical foundation provided by differential reassignment theory. It introduces a design parameter, Θ, which manipulates the velocity of energy travel along differential reassignment trajectories. A performance improvement can be achieved by exploiting the degree of freedom provided by Θ. In the following, Θ is introduced theoretically.

An implementation of differential reassignment discretizes the transport equation, Equation 3.1.

Equal scaling of v in both dimensions ensures that energy movement still follows the same trajectories, but with different velocities. This is expressed by

$$I_t + \Theta v^x \cdot I_x + \Theta v^y \cdot I_y = 0, \tag{3.29}$$

where $\Theta = \Theta(x, y)$ is a constant matrix scaling v, introducing a degree of freedom.

Hence, differential reassignment now occurs with the new velocities

$$\bar{v}^x = \Theta v^x \tag{3.30}$$

$$\bar{v}^y = \Theta v^y. \tag{3.31}$$

Because the transport equation is a two-dimensional scalar equation, dimensional splitting

$$I_t + \bar{v}^x I_x = 0 \tag{3.32}$$

$$I_t + \bar{v}^y I_y = 0 \tag{3.33}$$

can be applied in order to use the Donor-Cell Upwind Method for Advection, compare LeVeque [16]. The image is then updated with the following procedure, formulated for x-direction movement. First, energy loss is subtracted from donor cells

$$\tilde{I}^{n+1}(x,y) = I^n(x,y) - |\bar{v}^x(x,y)| \cdot \triangle t \cdot \triangle x \cdot I^n(x,y). \tag{3.34}$$

Then, if a cell receives energy from its left neighbor, so if $\bar{v}^x(x-1,y) > 0$, then this energy is added to the actual cell by

$$\hat{I}^{n+1}(x,y) = \tilde{I}^{n+1}(x,y) + |\bar{v}^x(x-1,y)| \cdot \triangle t \cdot \triangle x \cdot I^n(x-1,y). \tag{3.35}$$

Otherwise,

$$\hat{I}^{n+1}(x,y) = \tilde{I}^{n+1}(x,y). \tag{3.36}$$

Finally, if a cell receives energy from its right neighbor, so if $\bar{v}^x(x+1,y) < 0$, then this energy is added to the actual cell by

$$I^{n+1}(x,y) = \hat{I}^n(x,y) + |\bar{v}^x(x+1,y)| \cdot \triangle t \cdot \triangle x \cdot I^n(x+1,y). \tag{3.37}$$

Otherwise,

$$I^{n+1}(x,y) = \hat{I}^{n+1}(x,y). \tag{3.38}$$

Using of this scheme ensures that no energy is created or destroyed, the resulting algorithm is fully conservative. The algorithm alternates between x-dimension and y-dimension energy movement. The equations governing y-dimension energy movement are analogous to those for the x-dimension.

As $\triangle x = \triangle y = 1$, for stability, the step size has to hold the inequality

$$\triangle t \leq \frac{1}{2 \cdot \max(\bar{v}^x(\cdot,\cdot), \bar{v}^y(\cdot,\cdot))}. \tag{3.39}$$

Good results have been obtained with the design parameter

$$\Theta(x,y) = 1 - \frac{I^0(x,y)}{\max(I^0(\cdot,\cdot))}. \tag{3.40}$$

This choice ensures that Θ helps to further attenuate reassignment flow in high energy regions, namely along ridges.

Even better results have been achieved setting the design parameter to

$$\Theta(x,y) = e^{-\frac{I^0(x,y)}{\max(I^0(\cdot,\cdot))}}. \tag{3.41}$$

This choice has the same effect as the previous one, but uses the exponential function to reach the desired attenuation.

Note that the choice

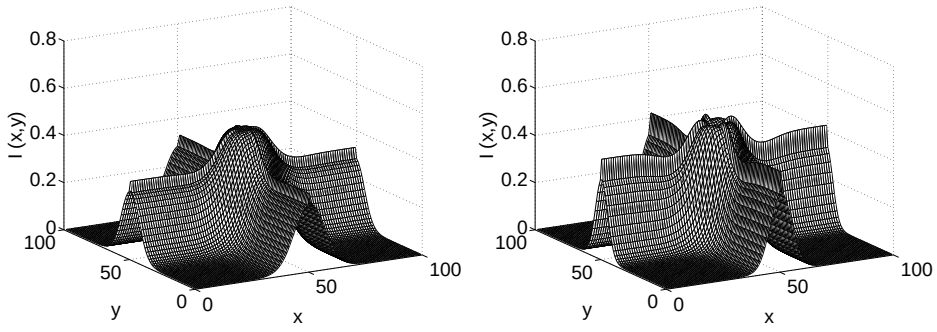
$$\Theta(x,y) = 1 \quad \forall \quad (x,y) \tag{3.42}$$

leads back to the original first version.

It is interesting to note the following. The differential reassignment vector field v depends on an image *gradient*. It is dependent upon the constant scalar potential $\log I^0$. The log-function attenuates the *gradient* in high energy regions, and amplifies it in low energy regions. Using Equation 3.40, the manipulated vector field $\bar{v}$ additionally depends on the initial image *surface*. Velocities in high energy regions are further attenuated, and velocities in low energy regions are further amplified, this time dependent on the constant initial image *surface*.

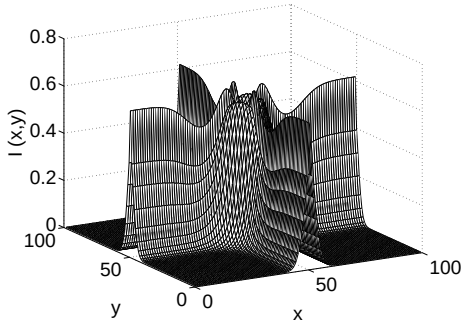
3.1.3.2 Demonstration

Once again, the two-dimensional image shown in Figure 3.1 is used for demonstration purposes — its scalar potential is presented in Figure 3.2. First, the design parameter Θ is chosen according to Equation 3.41. The sequence of Figures 3.7(a) to 3.7(f) shows image evolution after 500, 1,000, 2,000, 5,000, 10,000 and 100,000 iterations, respectively. Comparison with the image evolution of the two earlier versions reveals that this version constitutes a good trade-off. Ridges are more localized

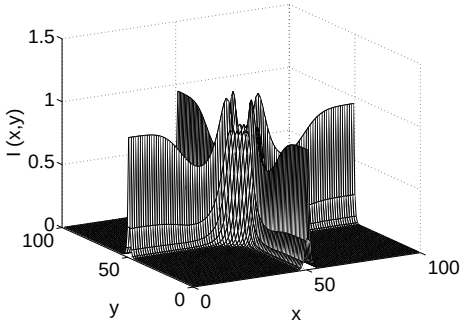


(a) 500 Iterations

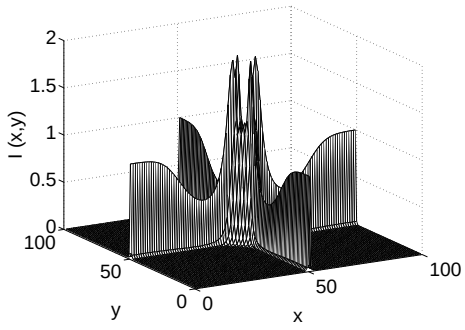
(b) 1,000 Iterations



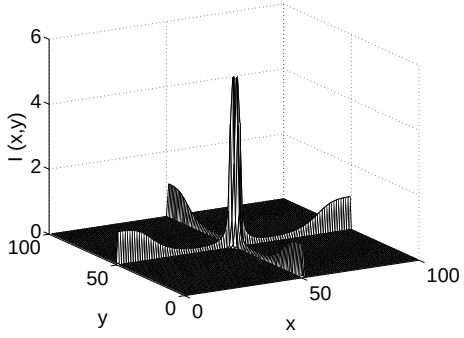
(c) 2,000 Iterations



(d) 5,000 Iterations



(e) 10,000 Iterations

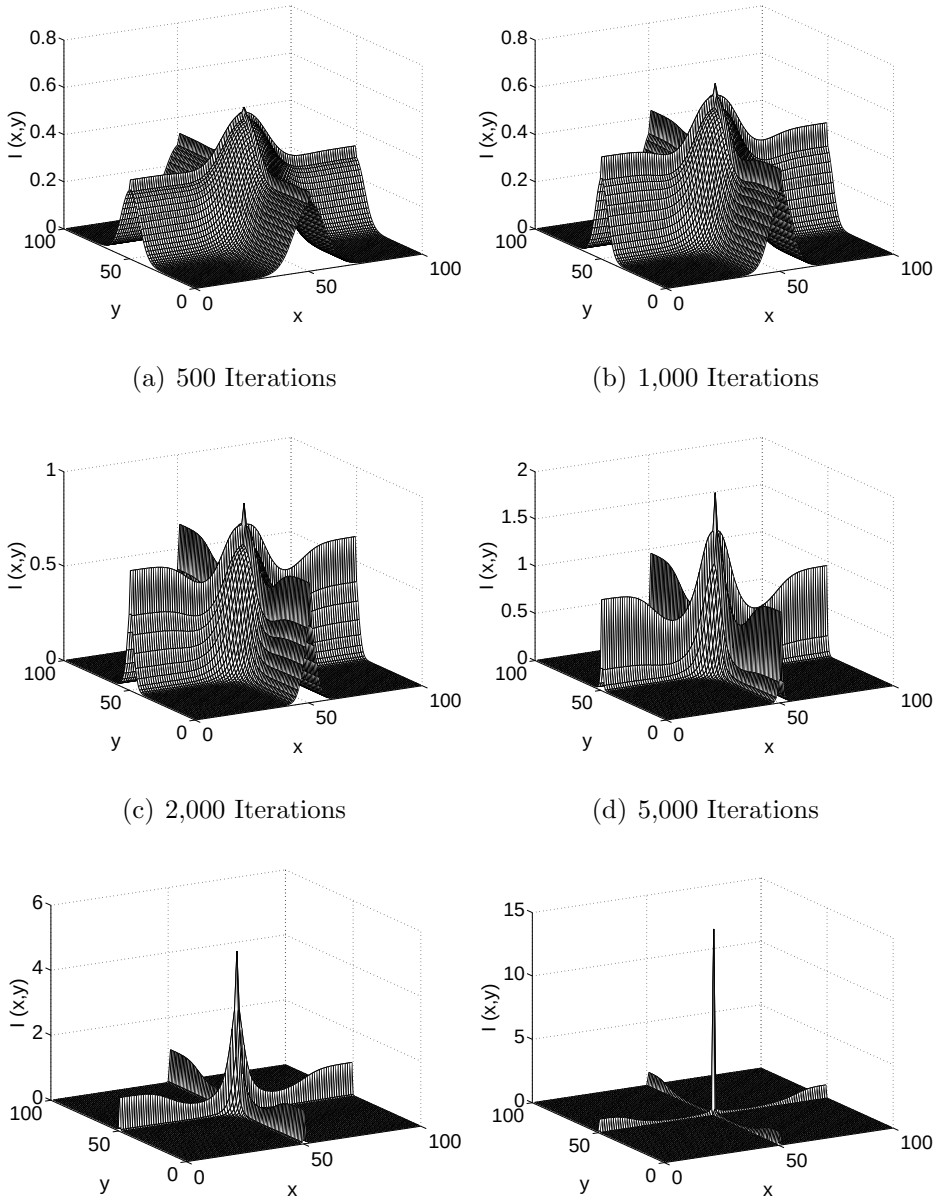


(f) 100,000 Iterations

Figure 3.7: Image evolution of the third differential reassignment version (1).

as in the first version, and dip formation is weaker as in the second version. The intermediate evolution stage shown in Figure 3.7(c) is especially satisfactory. However, at the intersection of the two ridges, energy is not as well localized as in the previous versions.

Now, consider the choice of the design parameter Θ according to Equation 3.42. The sequence of Figures 3.8(a) to 3.8(f) shows image evolution after 500, $1,000$, $2,000$, $5,000$, $10,000$ and $100,000$ iterations, respectively. Comparing this image evolution to all previous image evolutions shows that by using this latest version: ridges as well as ridge intersections are well localized; ridge edges are therefore steep; and dip formation is comparatively weak. Hence, this particular version will be used in Chapter 4 for the differential reassignment portion of the combined algorithm.



(a) 500 Iterations

(b) 1,000 Iterations

(c) 2,000 Iterations

(d) 5,000 Iterations

(e) 10,000 Iterations

(f) 100,000 Iterations

Figure 3.8: Image evolution of the third differential reassignment version (2).

3.2 Diffusion algorithm

As was demonstrated in Section 3.1.2.2, differential reassignment alone produces dips along reassigned ridges that were not present in the original image. It was also shown that these dips are the result of an ascent toward a peak that is located on a ridge. Since it cannot be assumed that any ridge will be perfectly flat on its top, dip (and with it peak) formation is generally very likely to occur. Notice that the ridge edges, especially in Figures 3.6(a) and 3.6(b), are very sharp compared to the smooth contours on top of the ridges. Thus, dip formation can be countered by an anisotropic diffusion algorithm which preserves the edges, but levels out the top of ridges.

Section 2.3.2.2 described the theoretical background of the anisotropic diffusion algorithm which is used in this research. The following subsections outline the implementation of the diffusion algorithm, and demonstrate its characteristics by the use of an example.

3.2.1 Implementation

Although not obvious from Equation 2.63, the discrete implementation of nonlinear anisotropic diffusion is fairly straightforward. The algorithm is first derived in its one–dimensional form, which is then extended to two dimensional images.

In only one dimension x, the gradient and divergence expressions in Equation 2.63 reduce to derivatives,

$$\frac{\partial}{\partial t}I(x,t) = \frac{\partial}{\partial x}\left(c(x,t)\cdot\frac{\partial}{\partial x}I(x,t)\right). \tag{3.43}$$

Substituting discrete approximations for the derivatives,

$$\frac{\partial}{\partial t}I(x,t) \approx \frac{1}{\triangle x}\left\{c(x,t)\cdot\frac{1}{\triangle x}\left[I\left(x+\frac{\triangle x}{2},t\right)-I\left(x-\frac{\triangle x}{2},t\right)\right]\right\} \tag{3.44}$$

is obtained, which can be further approximated by

$$\begin{aligned}\frac{\partial}{\partial t}I(x,t) \approx{}& \frac{1}{(\triangle x)^2}\left[c\left(x+\frac{\triangle x}{2},t\right)\cdot\left(I(x+\triangle x,t)-I(x,t)\right)\right]\\ &-\frac{1}{(\triangle x)^2}\left[c\left(x-\frac{\triangle x}{2},t\right)\cdot\left(I(x,t)-I(x-\triangle x,t)\right)\right].\end{aligned} \tag{3.45}$$

Introduction of the flow functions

$$\begin{aligned}\Psi^{\text{west}}(x,\triangle x,t) &= \frac{1}{(\triangle x)^2}\left[-c\left(x-\frac{\triangle x}{2},t\right)\cdot(I(x,t)-I(x-\triangle x,t))\right]\\ \Psi^{\text{east}}(x,\triangle x,t) &= \frac{1}{(\triangle x)^2}\left[c\left(x+\frac{\triangle x}{2},t\right)\cdot(I(x+\triangle x,t)-I(x,t))\right],\end{aligned} \tag{3.46}$$

reduces Equation 3.45 to

$$\frac{\partial}{\partial t}I(x,t) \approx \Psi^{\text{west}}(x,\triangle x,t)+\Psi^{\text{east}}(x,\triangle x,t). \tag{3.47}$$

To calculate the flow functions in Equation 3.46, the terms $c\left(x-\frac{\triangle x}{2},t\right)$ and $c\left(x+\frac{\triangle x}{2},t\right)$ have to be computed. This can be done easily by substituting the discrete approximation of the gradient into Equation 2.64, yielding

$$\begin{aligned}c\left(x-\frac{\triangle x}{2},t\right) &= f\left(\frac{1}{\triangle x}\left|I(x,t)-I(x-\triangle x,t)\right|\right)\\ c\left(x+\frac{\triangle x}{2},t\right) &= f\left(\frac{1}{\triangle x}\left|I(x+\triangle x,t)-I(x,t)\right|\right).\end{aligned} \tag{3.48}$$

Equation 3.47 indicates that, for each iteration, the value of each cell in a 1D array changes according to the flow contributed from its nearest neighbors. Positive flow directions is defined as flow into the actual cell. An expression for the discrete implementation of 1D diffusion can be derived by determining the cell values after each iteration,

$$I^{n+1}(x) = I^n(x)+\triangle t\cdot\left(\Psi^{\text{west}}+\Psi^{\text{east}}\right). \tag{3.49}$$

This iteration scheme is stable as long as $0 \le \triangle t \le \frac{1}{3}$.

The 1D discrete formulation of the diffusion process is easily extended to the 2D case by

$$\frac{\partial}{\partial t}I(x,y,t) = \frac{\partial}{\partial x}\left(c(x,y,t)\cdot\frac{\partial}{\partial x}I(x,y,t)\right)+\frac{\partial}{\partial y}\left(c(x,y,t)\cdot\frac{\partial}{\partial y}I(x,y,t)\right), \tag{3.50}$$

and the approximation

$$\begin{aligned}\frac{\partial}{\partial t}I(x,y,t) \approx\ & \frac{1}{(\triangle x)^2}\left[c\left(x+\frac{\triangle x}{2},y,t\right)\cdot(I(x+\triangle x,y,t)-I(x,y,t))\right]\\ & -\frac{1}{(\triangle x)^2}\left[c\left(x-\frac{\triangle x}{2},y,t\right)\cdot(I(x,y,t)-I(x-\triangle x,y,t))\right]\\ & +\frac{1}{(\triangle y)^2}\left[c\left(x,y+\frac{\triangle y}{2},t\right)\cdot(I(x,y+\triangle y,t)-I(x,y,t))\right]\\ & -\frac{1}{(\triangle y)^2}\left[c\left(x,y-\frac{\triangle y}{2},t\right)\cdot(I(x,y,t)-I(x,y-\triangle y,t))\right].\end{aligned} \tag{3.51}$$

Definition of the flow functions

$$\begin{aligned}
\Psi^{\text{west}} &= \frac{1}{(\triangle x)^2}\left[-c\left(x-\frac{\triangle x}{2},y,t\right)\cdot(I(x,y,t)-I(x-\triangle x,y,t))\right] \\
\Psi^{\text{east}} &= \frac{1}{(\triangle x)^2}\left[c\left(x+\frac{\triangle x}{2},y,t\right)\cdot(I(x+\triangle x,y,t)-I(x,y,t))\right] \\
\Psi^{\text{north}} &= \frac{1}{(\triangle y)^2}\left[-c\left(x,y-\frac{\triangle y}{2},t\right)\cdot(I(x,y,t)-I(x,y-\triangle y,t))\right] \\
\Psi^{\text{south}} &= \frac{1}{(\triangle y)^2}\left[c\left(x,y+\frac{\triangle y}{2},t\right)\cdot(I(x,y+\triangle y,t)-I(x,y,t))\right]
\end{aligned} \tag{3.52}$$

reduces Equation 3.51 to

$$\frac{\partial}{\partial t}I(x,y,t) \approx \Psi^{\text{west}} + \Psi^{\text{east}} + \Psi^{\text{north}} + \Psi^{\text{south}}. \tag{3.53}$$

An expression for the discrete implementation of 2D diffusion is, analogous to the 1D expression, given by

$$I^{n+1}(x,y) = I^{n}(x,y) + \triangle t \cdot \left(\Psi^{\text{west}} + \Psi^{\text{east}} + \Psi^{\text{north}} + \Psi^{\text{south}}\right). \tag{3.54}$$

The 2D diffusion process consists of updating each cell in an image by an amount equal to the flow contributed by its four nearest neighbors, with positive flow direction into the actual cell. This scheme is stable as long as $0 \leq \triangle t \leq 0.2$.

Note that both 1–D and 2–D diffusion algorithms are conservative, mass is only moved around image cells, it is not created or destroyed.

3.2.2 Demonstration

As was described in Section 2.3.2.2, nonlinear anisotropic diffusion is capable of diffusing noise while preserving edges. Section 3.1.2.2 revealed that differential reassignment produces ridges having sharp edges, which suffer from undesired dips in certain regions close to a peak.

To demonstrate the correct implementation and capability of the anisotropic diffusion algorithm, a two–dimensional image with a resolution of 100x100 was chosen. This image is zero except for the central horizontal and vertical lines, which follow

$$\begin{aligned}
I(x,50) &= 3 + \frac{1}{2}\cos\left(\frac{\pi}{10}\cdot x\right) \\
I(50,y) &= 3 + \frac{1}{2}\cos\left(\frac{\pi}{10}\cdot y\right).
\end{aligned} \tag{3.55}$$

On top of that, random noise with a maximum magnitude of 1 was added. The resulting image is shown in Figure 3.9 and serves as initial image for demonstration purposes. It consists of two ridges with dips, and is in addition to that suspect to random noise.

The anisotropic diffusion algorithm uses $\kappa = 0.5$ and a step size of $\triangle t = 0.2$ as parameters. After 200 iterations, the image presented in Figure 3.10 is obtained. Note that anisotropic diffusion has an excellent performance with regard to three different aspects. First, noise is diffused. Second, sharp edges are preserved. Third, dips on the ridges are smoothed out. Note also that the mass of $2,738$ units is preserved throughout the algorithm. The resulting image would be a highly satisfactory result of a combined algorithm, as the flat ridges are exactly located.

If an isotropic diffusion algorithm with the same step size was used, the highly unsatisfactory image of Figure 3.11 would have been obtained after 200 iterations. Notice the apparent similarity between this image and Figure 3.1. The shape of

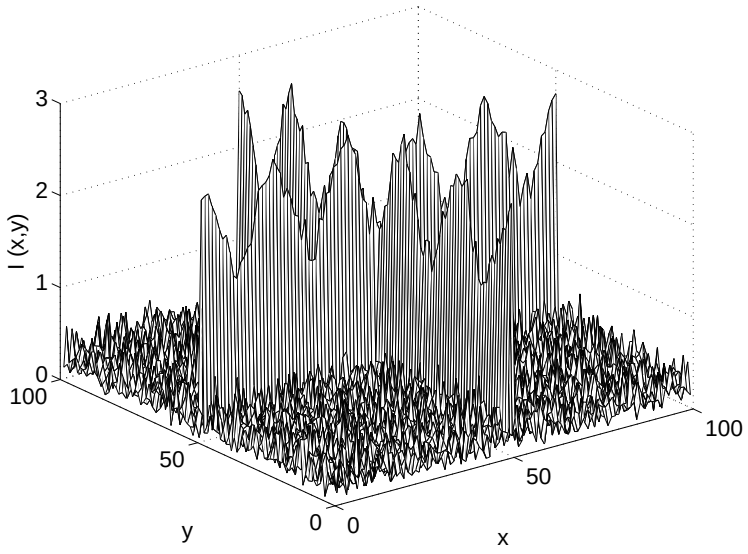


Figure 3.9: Starting image for the demonstration of diffusion algorithms.

Figure 3.11 is also obtained if a plain cross of perfectly flat ridges is used as original image for the isotropic diffusion algorithm. But this plain cross of perfectly flat ridges is the desired result of the combined algorithm, if in turn Figure 3.11 was taken as

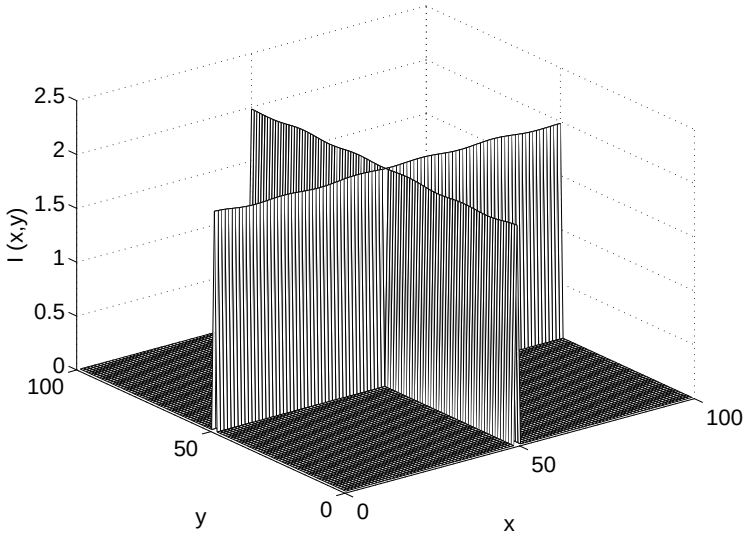


Figure 3.10: Demonstration of nonlinear anisotropic diffusion.

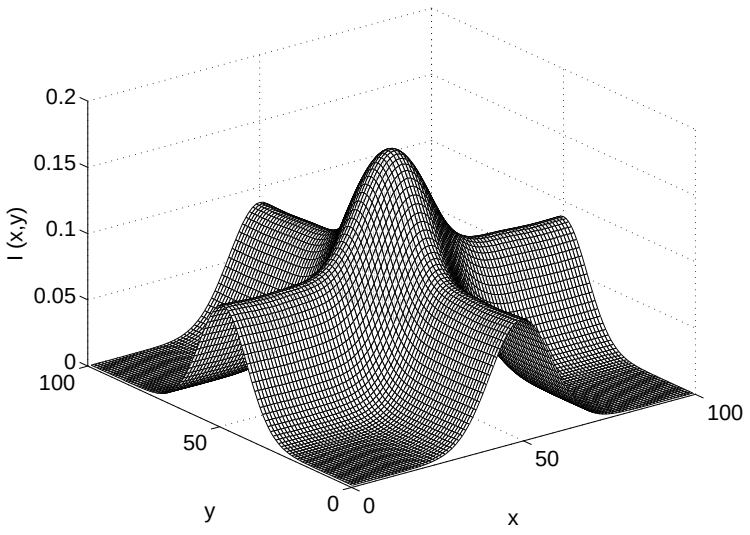


Figure 3.11: Demonstration of isotropic diffusion.

initial image. Thus, an alternative interpretation of the goal of this research is to find an image J based on an image I, such that isotropic diffusion of J leads back to I. Of course, a unique inverse operation of isotropic diffusion cannot exist, since diffusion increases the entropy of the image. For example, isotropic diffusion of a plain cross of perfectly flat ridges and of Figure 3.9 both lead to optically identical images, namely Figure 3.11. But of the two 'inverse operations', only the flat ridge cross is a desirable result of a combined algorithm.

3.3 Combined algorithm

Reassignment and diffusion algorithms are now combined in order to improve reassignment performance. While the theoretical background of both algorithms is elucidated in Sections 2.3.1.2 and 2.3.2.2, and the individual algorithms are developed in Sections 3.1 and 3.2, this section describes their combination.

The combined algorithm simply consists of the carefully weighted addition of reassignment and diffusion movements. To balance reassignment with diffusion, a weighting factor

$$\beta = \frac{\triangle t_{\text{reassignment}}}{\triangle t_{\text{diffusion}}} \tag{3.56}$$

is introduced, reflecting the relation between reassignment and diffusion step sizes. Finding a suitable value for β involves an iterative trial-and-error approach. If the reassignment part is too strong, β has to be increased. If the image is diffused too much, then β has to be reduced.

Image evolution is then expressed by a single partial differential equation. In Chapter 3.1, three different versions of differential reassignment are presented. The resulting PDE depends on the reassignment version that is incorporated into the combined algorithm. If the first reassignment version is used, the PDE governing energy movement is

$$\frac{\partial I}{\partial t} + \frac{\partial \log I^0}{\partial x} \cdot \frac{\partial I}{\partial x} + \frac{\partial \log I^0}{\partial y} \cdot \frac{\partial I}{\partial y} - \beta \cdot \left[\frac{\partial}{\partial x} \left(c_1 \cdot \frac{\partial I}{\partial x} \right) + \frac{\partial}{\partial y} \left(c_1 \cdot \frac{\partial I}{\partial y} \right) \right] = 0. \tag{3.57}$$

For the second version, the PDE reduces to

$$\frac{\partial I}{\partial t} + \frac{\partial \log I^0}{\partial x} + \frac{\partial \log I^0}{\partial y} - \beta \cdot \left[\frac{\partial}{\partial x} \left(c_1 \cdot \frac{\partial I}{\partial x} \right) + \frac{\partial}{\partial y} \left(c_1 \cdot \frac{\partial I}{\partial y} \right) \right] = 0. \tag{3.58}$$

The third version leads to

$$\frac{\partial I}{\partial t}+\Theta\cdot\frac{\partial \log I^0}{\partial x}\cdot\frac{\partial I}{\partial x}+\Theta\cdot\frac{\partial \log I^0}{\partial y}\cdot\frac{\partial I}{\partial y}-\beta\cdot\left[\frac{\partial}{\partial x}\left(c_1\cdot\frac{\partial I}{\partial x}\right)+\frac{\partial}{\partial y}\left(c_1\cdot\frac{\partial I}{\partial y}\right)\right]=0. \quad (3.59)$$

Due to the observations made in Section 3.1, only the third version is used in subsequent chapters. In particular, the choice of Θ according to Equation 3.42 has proven to be most advantageous. This choice of Θ, together with the diffusion function c_1 according to Equation 2.65, is plugged into Equation 3.59 to yield the PDE

$$\begin{aligned}&\frac{\partial I}{\partial t}+e^{-\frac{I^0}{\max(I^0)}}\cdot\frac{\partial \log I^0}{\partial x}\cdot\frac{\partial I}{\partial x}+e^{-\frac{I^0}{\max(I^0)}}\cdot\frac{\partial \log I^0}{\partial y}\cdot\frac{\partial I}{\partial y}\\&-\beta\cdot\left[\frac{\partial}{\partial x}\left(e^{-\left(\frac{1}{\kappa}\cdot|\frac{\partial I}{\partial x}|\right)^2}\cdot\frac{\partial I}{\partial x}\right)+\frac{\partial}{\partial y}\left(e^{-\left(\frac{1}{\kappa}\cdot|\frac{\partial I}{\partial y}|\right)^2}\cdot\frac{\partial I}{\partial y}\right)\right]=0. \quad (3.60)\end{aligned}$$

With the un-reassigned spectrogram as initial condition and Neumann boundary conditions, this PDE fully describes the image evolution of the final algorithm.

Demonstration of the combined algorithm is delayed to Section 4.2.2, where the algorithm is applied to reassign experimentally measured spectrograms.

3.4 Noise treatment

So far, only smooth, noise-free images have been considered. By averaging data from multiple experiments, a spectrogram with a high signal-to-noise ratio (SNR) is obtained. However, a high SNR cannot always be guaranteed, making noise removal considerations a necessity.

As demonstrated in Section 3.2.2, nonlinear anisotropic diffusion is capable of diffusing noise when its parameter κ approximately matches the highest noise magnitude. On the other hand, to ensure energy diffusion along a ridge, κ should approximately resemble the highest derivative magnitudes occurring along a ridge. How can those two requirements be met at the same time?

Superimposing reassignment with *two* anisotropic diffusion algorithms having individual parameters κ_1 and κ_2, where one algorithm diffuses noise, and the other ridge tops, is not a good idea. This is because the constant differential reassignment vector field is calculated based on the initial image, subject to noise. Noise produces

a lot of small, local maxima. Since reassignment vectors all point toward local maxima, the resulting vector field obviously favors spurious peaks and does not lead to a satisfactory image evolution.

Chassande-Mottin remarked in [6] that all attempts to smooth out a noisy reassignment vector field had failed. Accepting this, an alternative way to solve the problem of noise is needed. If the reassignment vector field cannot be smoothed out, the only logical solution is to obtain a smooth vector field in the first place. Thus, noise has to be removed from the image before it serves as a basis for vector field computation. The noise removal algorithm developed in this research consists of two parts.

First, an anisotropic diffusion algorithm with a diffusion parameter κ slightly above the maximum noise magnitude is applied to the image. Noise is being diffused stronger than the information content of the image. Once the scalar potential $\log I$ of the image is sufficiently smooth in medium to high mass regions, diffusion iterations stop.

Second, image cells below a tiny threshold δ are set to zero. The reason for this is that the scalar potential $\log I$, the basis for reassignment vector field computation, is highly sensitive to even tiny derivatives in low energy regions. Even thorough diffusion of noise in areas with (almost) zero energy content still leaves tiny derivatives, which are then amplified by the scalar potential function, resulting in undesired peak and ridge formation. If δ is chosen appropriately, the low energy content of those cells capable of forming spurious peaks is destroyed beforehand. Since those high entropy regions have very little information content left (information that was only slightly present has been most certainly diffused), a destruction of their energy content is justified. Usually, the destructed mass only accounts for a small fraction of the total image energy. Exact energy conservation can still be achieved by scaling the image such that the tiny destroyed energy fraction is distributed across the image, proportional to its cell values.

Noise diffusion still leaves tiny surface variations in medium to high mass regions. Due to its logarithmic nature, the scalar potential function is fortunately comparatively insensitive to tiny irregularities in these regions. Reassignment vectors may be

bent a little, but anisotropic diffusion smoothes out the top of a ridge anyway. Hence, small directional errors are smeared out, especially those occurring on, or close to, a ridge top.

Finally, the effectiveness of the noise removal algorithm is demonstrated. For this purpose, the image of Figure 3.1 is again chosen, but this time random noise with a maximal magnitude of 0.025 is added. The resulting image is shown in Figure 3.4. Noise removal with $\kappa = 0.028$ (slightly above the noise magnitude) leads after 20 iterations with maximum step size $\triangle t = 0.2$ back to an image which is optically indistinguishable from Figure 3.1. Also, the resulting scalar potential field is optically indistinguishable from Figure 3.2. Even subsequent iterations show no noticeable differences to the evolution of the noise-free starting image. Consequentially, noise has been successfully removed without altering the information content significantly.

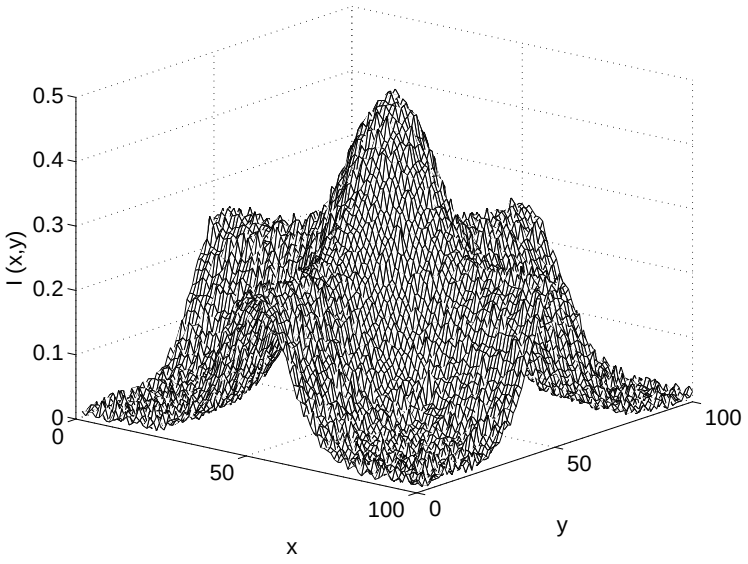


Figure 3.12: Noisy initial image.

CHAPTER 4

Application of the algorithm

So far, this research has only considered synthetically generated images. The current chapter applies the algorithm developed in Chapter 3 to experimentally measured spectrograms. The first section briefly introduces the experimental and signal processing steps required to obtain such a spectrogram. Then, both conventional and modified differential reassignment algorithms are applied to an experimentally measured spectrogram, and are compared to each other.

4.1 Experimental background

This section briefly outlines the necessary steps taken to obtain an experimentally measured spectrogram. More details on the experimental procedure are available in [5, 15].

Basic components needed to perform the experiments include the specimen itself, a source system, a detection system, and additional instrumentation to analyze the data.

Two different specimens are considered in this research. Both are plane, polished aluminum 3003 plates with dimensions 305 mm x 610 mm x 0.99 mm. One plate lacks any kind of defects and is referred to as "perfect." The other plate contains a notch of depth $0.5^{+0.1}_{-0.1}$ mm and width $1.6^{+0.2}_{-0.2}$ mm, and is referred to as "notched." The notched plate is shown in Figure 4.1.

A Q-switched, Nd:YAG pulse laser serves as an ultrasonic source. A laser is used because it provides a non-contact point source, exciting a broad frequency bandwidth. However, a laser source is not suitable for in-situ measurements; ongoing research [15] is developing a procedure that uses a contact piezoelectric element as a source.

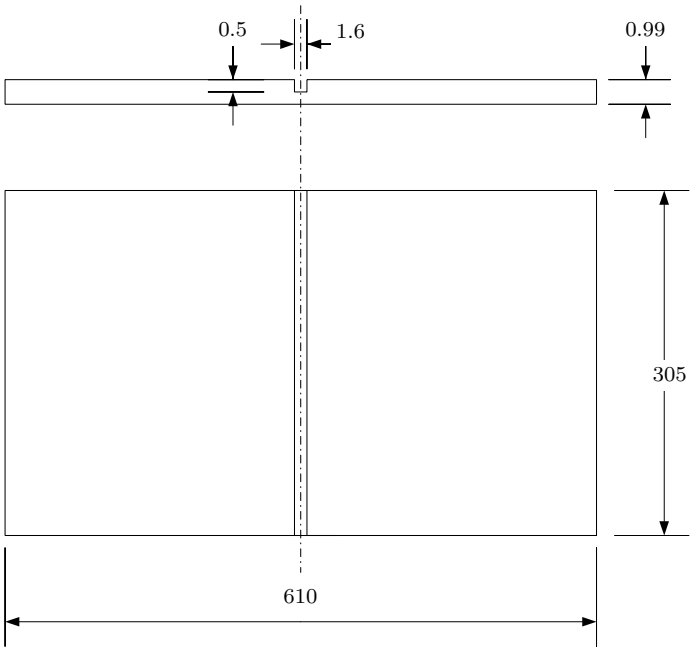


Figure 4.1: Sketch of the specimen.

A dual probe heterodyne interferometer serves as the detection system. After emission by the source, the laser beam is split in two. The object beam is reflected off the specimen surface, where Lamb waves cause an internal frequency shift in the beam (Doppler effect). The other beam serves as reference. Both beams finally hit the photodiodes of the detection system, converting light intensity into voltage signals.

The instrumentation takes the output voltage from the detection system and demodulates it to an absolute frequency shift. The demodulated signal is then low-pass filtered and discretized using an oscilloscope with a sampling frequency of 100 MHz. Here, multiple measurements are averaged to increase the signal-to-noise-ratio (SNR) of the measured signal. The oscilloscope is then connected to a personal computer.

Now, the signal processing techniques described in Chapter 2 are applied to the discretized time domain signal. The time domain signal is first high-pass filtered with a third order Butterworth filter having a cutoff frequency of 200 kHz. The energy density spectrum of a STFT finally yields a spectrogram which consists of several mode dispersion curves, characterizing the Lamb waves on the specimen's surface (see Section 2.1).

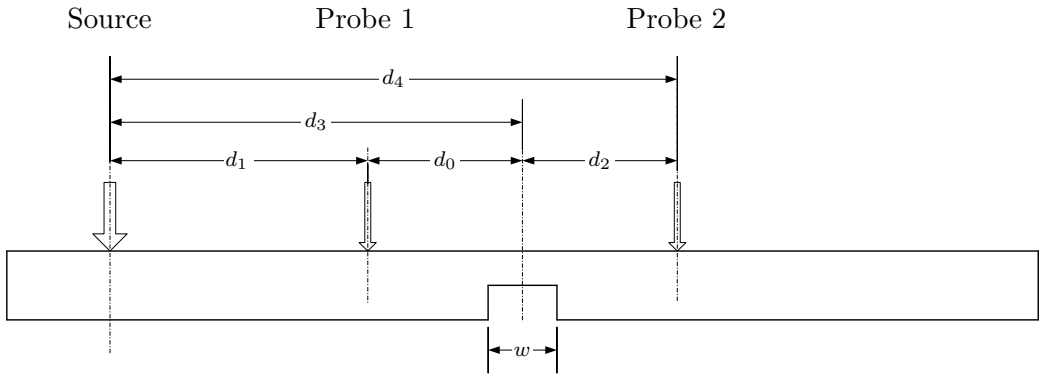


Figure 4.2: Dimensions of experimental setup.

The measurement data used in this research is from experiments conducted by Benz [5], and Benz's experimental setup is depicted in Figure 4.2. This study examines a set of measurements which Benz obtained using the distances listed in Table 4.1.

Table 4.1: Measurement distances

d_0[mm]	d_1[mm]	d_2[mm]	d_3[mm]	d_4[mm]
30	70	30	100	130

4.2 Reassignment of spectrograms

This section applies both conventional and modified differential reassignment techniques (discussed in Chapters 2 and 3) to experimentally measured spectrograms. First, conventional reassignment is used to localize the dispersion curves of a measured perfect plate signal. Then, the modified differential reassignment algorithm is applied to the same measurement data. This allows for an easy comparison between the two alternative methods.

The experimentally measured time domain signal is shown in Figure 4.3 before and after high-pass filtering. The noise spike at time $t = 0$ is due to the electromagnetic discharge of the Nd:YAG source laser. This spike is spurious and is windowed out of the time domain signal before processing. Note that the time domain signal is very complicated and it is very difficult to quantitatively interpret or to resolve the

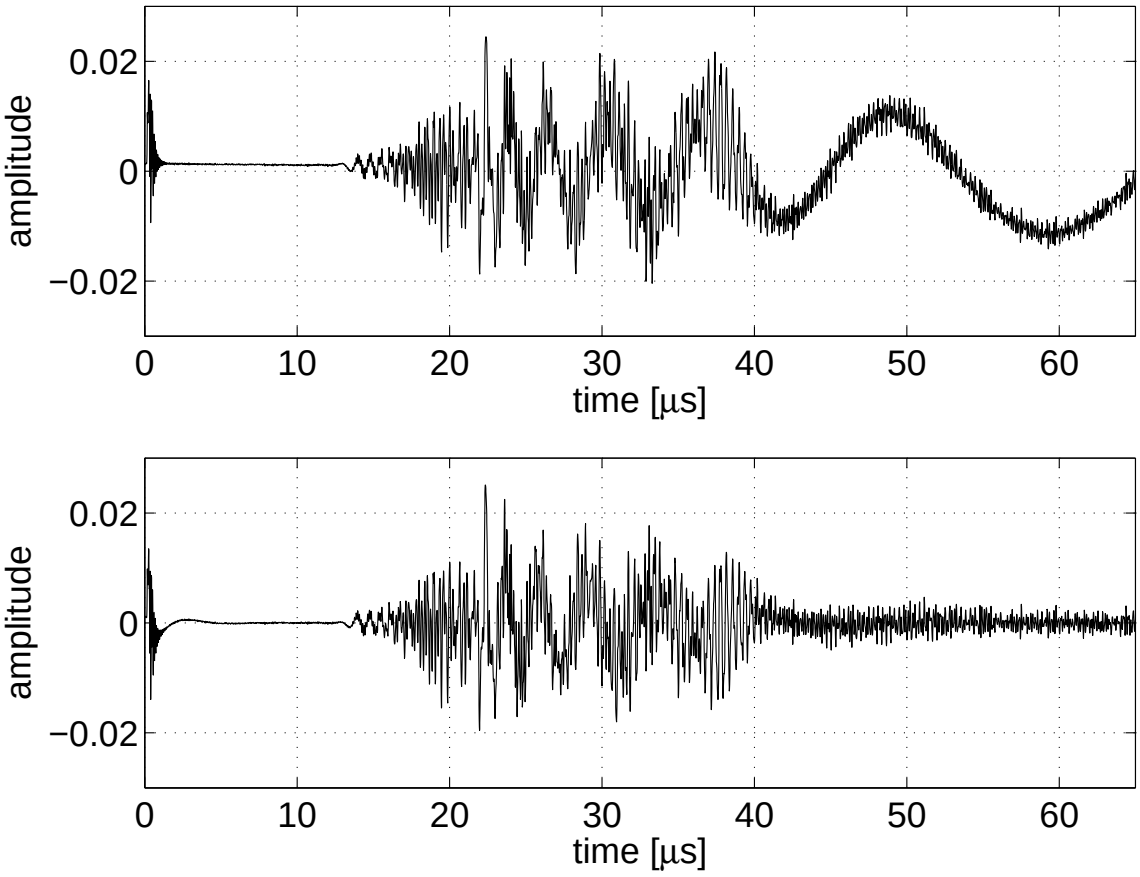


Figure 4.3: Time waveform before (top) and after (bottom) 200 kHz high-pass filtering.

contribution of individual modes.

The high-pass filtered time domain signal is used to calculate the un-reassigned spectrogram, and is shown in Figure 4.4 together with the theoretical mode solutions. Refer to Figure 2.3 to identify the theoretical modes in Figure 4.4 and following figures. A 385-pt. Hanning window is used to calculate the STFT. Due to the uncertainty principle, dispersion curves are ill-defined. Further note that the spectrogram has been converted to the slowness-frequency domain. This is done to normalize the signal with respect to propagation distance. For any given propagation distance d and time t, energy slowness is defined as

$$sl_e = \frac{t}{d}. \tag{4.1}$$

This transformation is linear in time. Hence, from an image processing point of view,

the image as such does not change, what does change is the associated vertical axis. Thus, transforming from the time-frequency to the slowness-frequency domain does not affect the accuracy of any potential image processing techniques. However, this transformation allows for comparison of signals with different propagation distances to each other, and to the same theoretical mode lines.

Reassignment techniques are applied to localize the ill-defined dispersion curves in the following subsections.

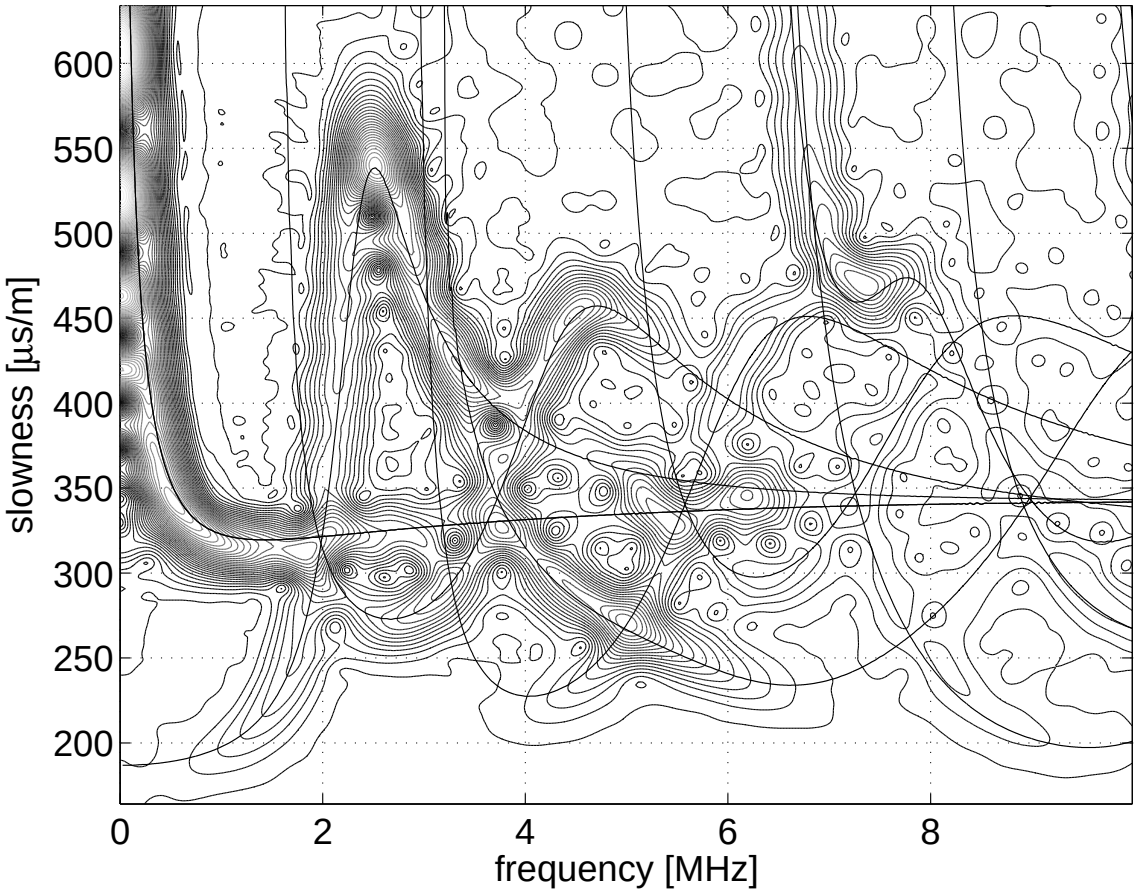


Figure 4.4: un-reassigned spectrogram.

4.2.1 Conventional reassignment

Conventional reassignment is described in Section 2.3.1.1 and is now applied to obtain a reassigned version of the experimentally measured spectrogram shown in Figure 4.4. The conventionally reassigned spectrogram is plotted in Figure 4.5. Spurious components, which are introduced by the reassignment algorithm, are clearly visible. In Figure 4.5, the lowest contour line corresponds to an energy square root of 0.001. As the energy of most spurious components is lower than the energy that is located on top of the theoretical dispersion curves, a threshold is applied to the image, and all values below the threshold are then set to zero. The reassigned spectrograms shown in Figures 4.6, 4.7 and 4.8 have been obtained with threshold values of 0.003, 0.01 and 0.03, respectively. In Figure 4.6, a significant amount of spurious components has been eliminated, although many spurious peaks still remain. Even more spurious components are eliminated in the reassigned spectrogram of Figure 4.7, without a significant loss of dispersion curve information. Applying a comparatively high threshold, as is done in Figure 4.8, eliminates most spurious components, but unfortunately, it also eliminates some "good" energy that corresponds to the theoretical dispersion curves segments — this is especially true for the higher frequencies. Thus, a more efficient procedure is needed; one that eliminates the spurious components, yet keeps the energy associated with the theoretical dispersion curves. It should be noted that applying a threshold violates the energy conservation requirement.

Figure 4.9 offers a zoomed-in view of the slowness-frequency region [1-2 MHz, 3-4 sl_e]. Ridges are far from smooth, spurious peaks are visible, and noise is generally present.

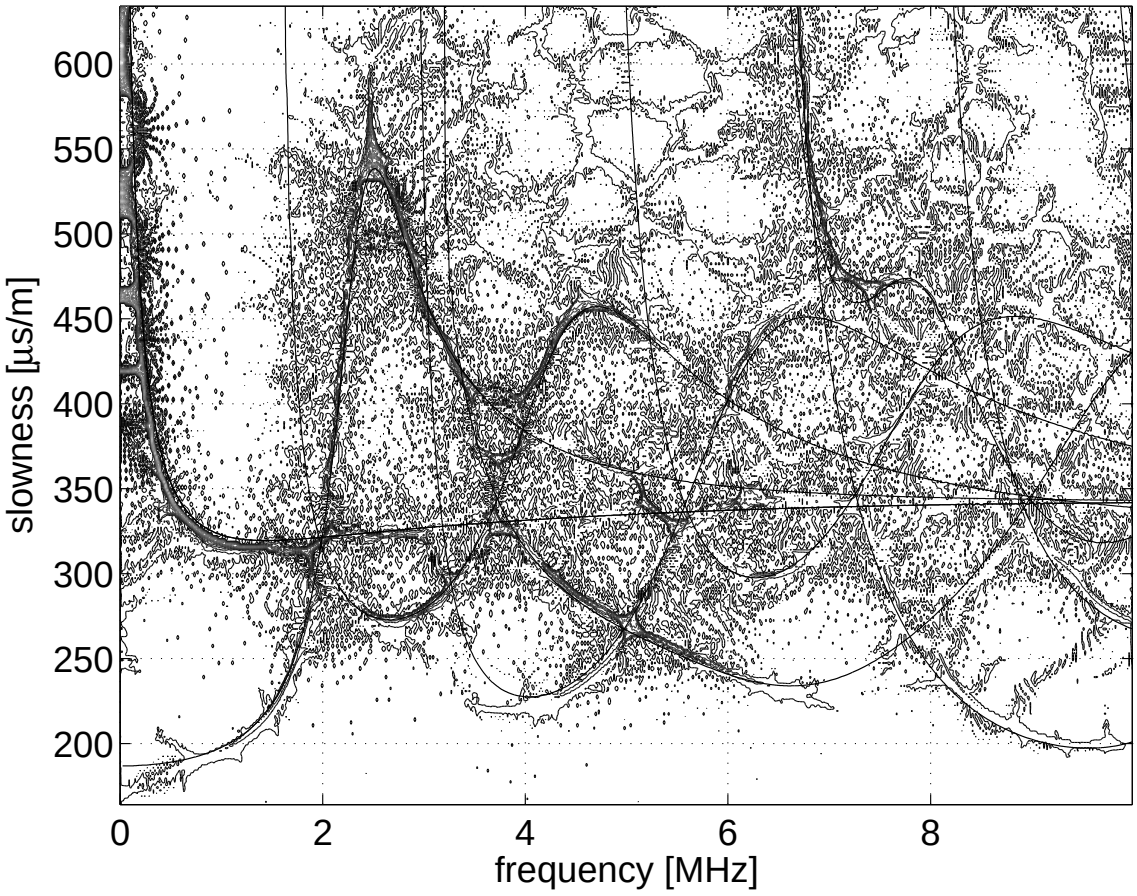


Figure 4.5: conventionally reassigned spectrogram.

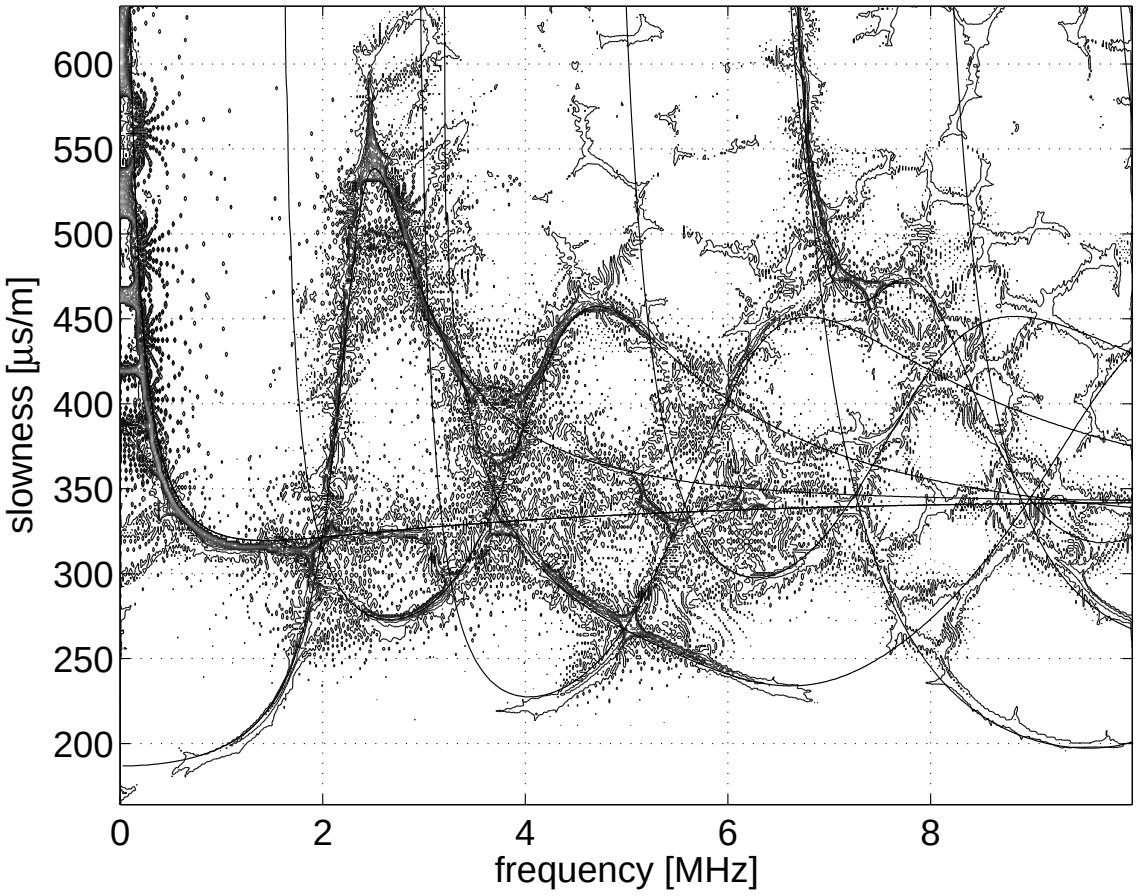


Figure 4.6: conventionally reassigned spectrogram, threshold 0.003.

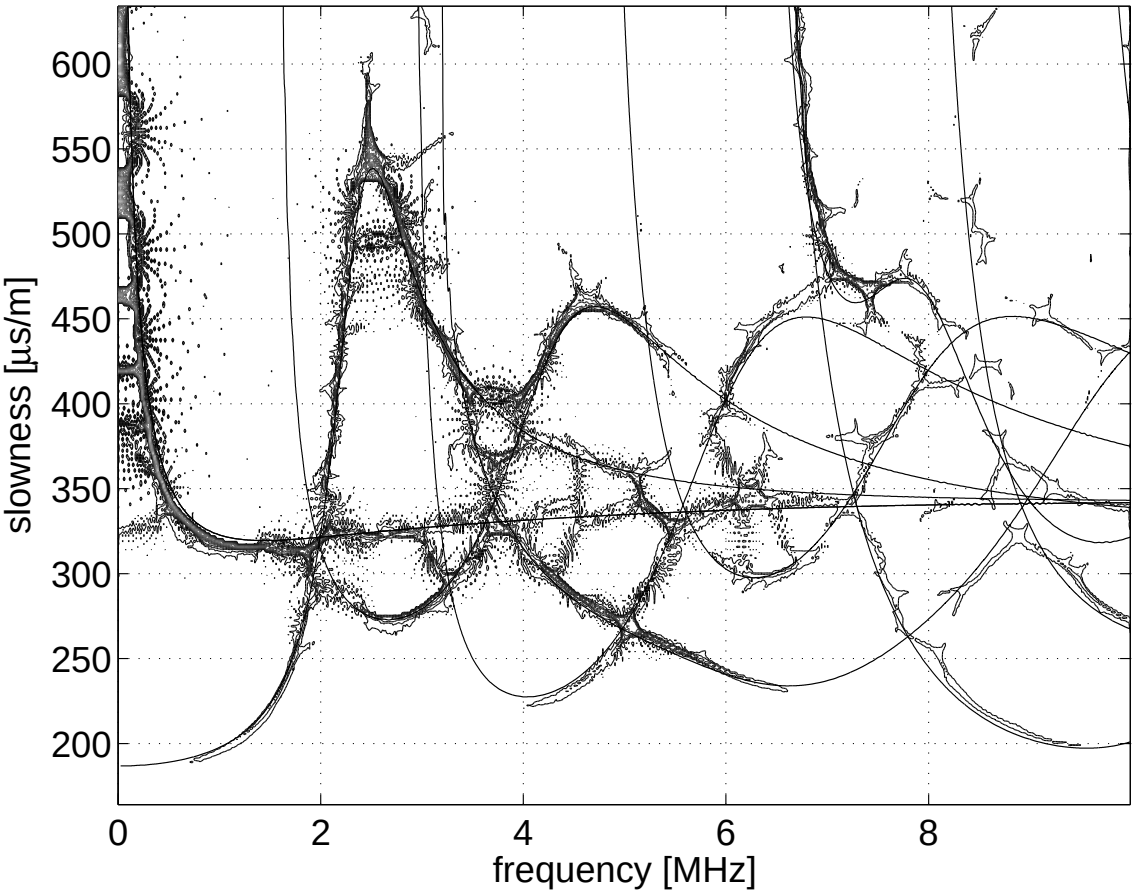


Figure 4.7: conventionally reassigned spectrogram, threshold 0.1.

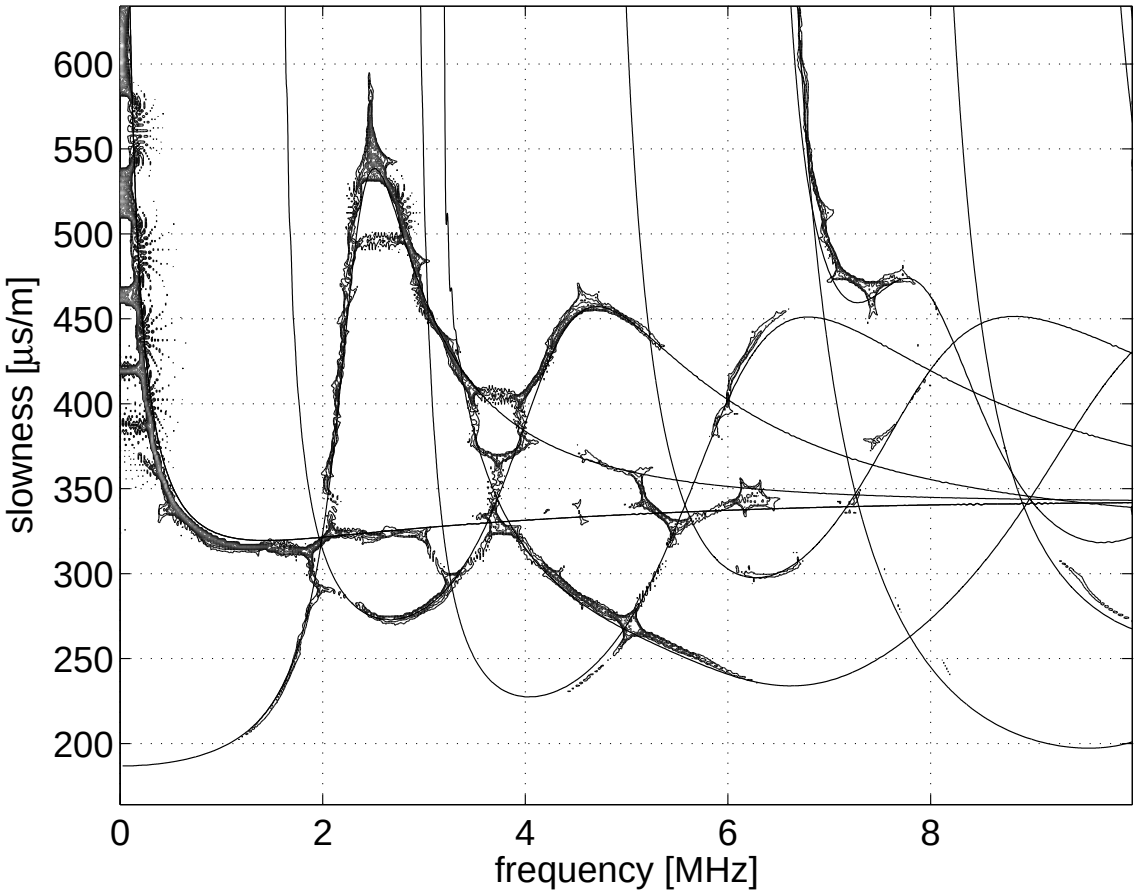


Figure 4.8: conventionally reassigned spectrogram, threshold 0.3.

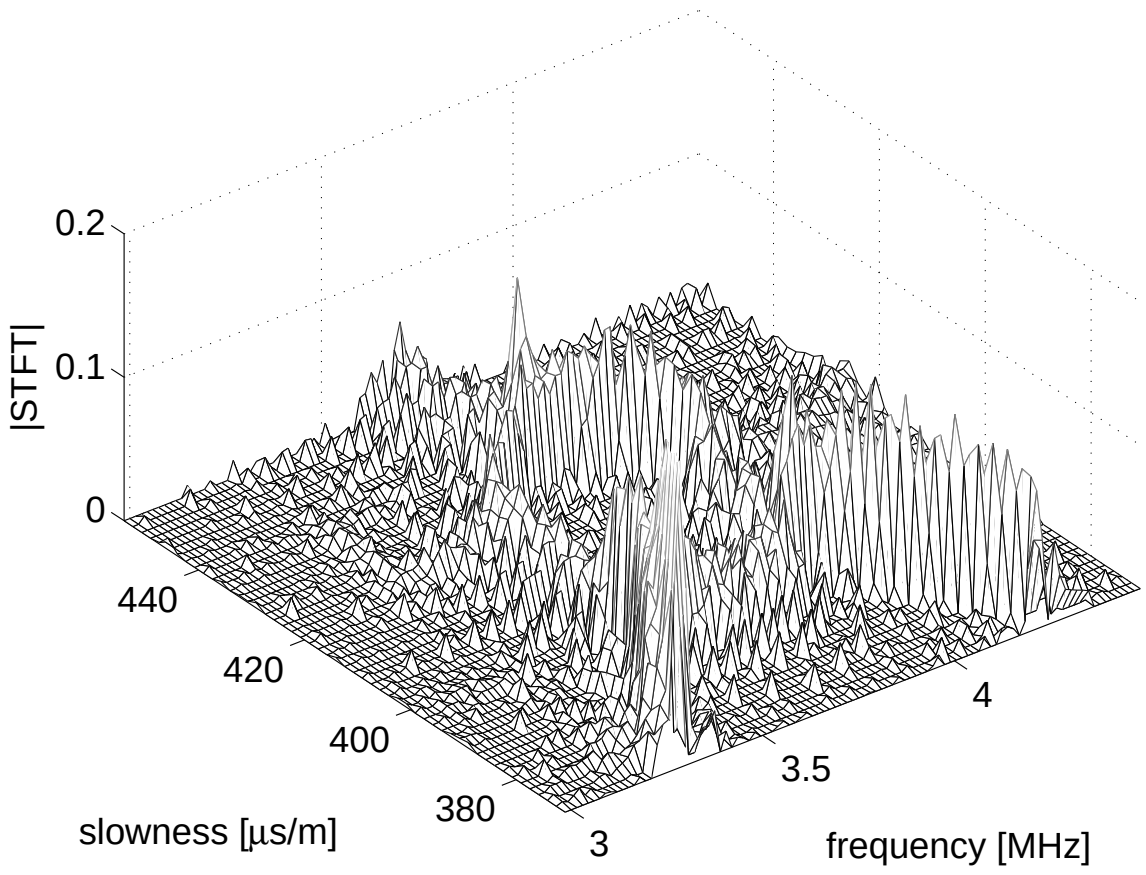


Figure 4.9: zoomed-in view on conventionally reassigned spectrogram.

4.2.2 Modified differential reassignment

The theoretical background of differential reassignment is discussed in Section 2.3.1.2. Chapter 3 derives a modified version of the differential reassignment algorithm, which is now applied to reassign the experimentally measured spectrogram shown in Figure 4.4. The parameters used in the modified differential reassignment algorithm are $\beta = 5 \cdot 10^{-4}$, $\kappa = 1.3$, and $\triangle t = 0.0158$.

Figures 4.10, 4.11, 4.12, and 4.13 show image evolution stages at t=0.0316, t=0.0791, t=0.1582, and t=0.4745, corresponding to $4,000$, $10,000$, $20,000$, and $60,000$ iterations, respectively.

This sequence of figures illustrates how the algorithm localizes dispersion curves with time. Compare Figure 4.13 to Figure 4.7. Both algorithms localize about the same dispersion curve information, but the differential version lacks spurious components. Hence, with a comparable information content between conventional and differential versions, the modified differential reassignment algorithm does increase clarity. Now, compare Figures 4.13 and 4.8. In both cases, spurious components are largely absent, although somewhat visible in the conventionally reassigned spectrogram. However, the differentially reassigned version includes significant energy that corresponds to the theoretical dispersion curves (especially in the higher frequencies) which do not appear in the conventionally reassigned spectrogram. Thus, with a comparable clarity between conventional and differential versions, the modified differential reassignment algorithm does extract more information from the same spectrogram.

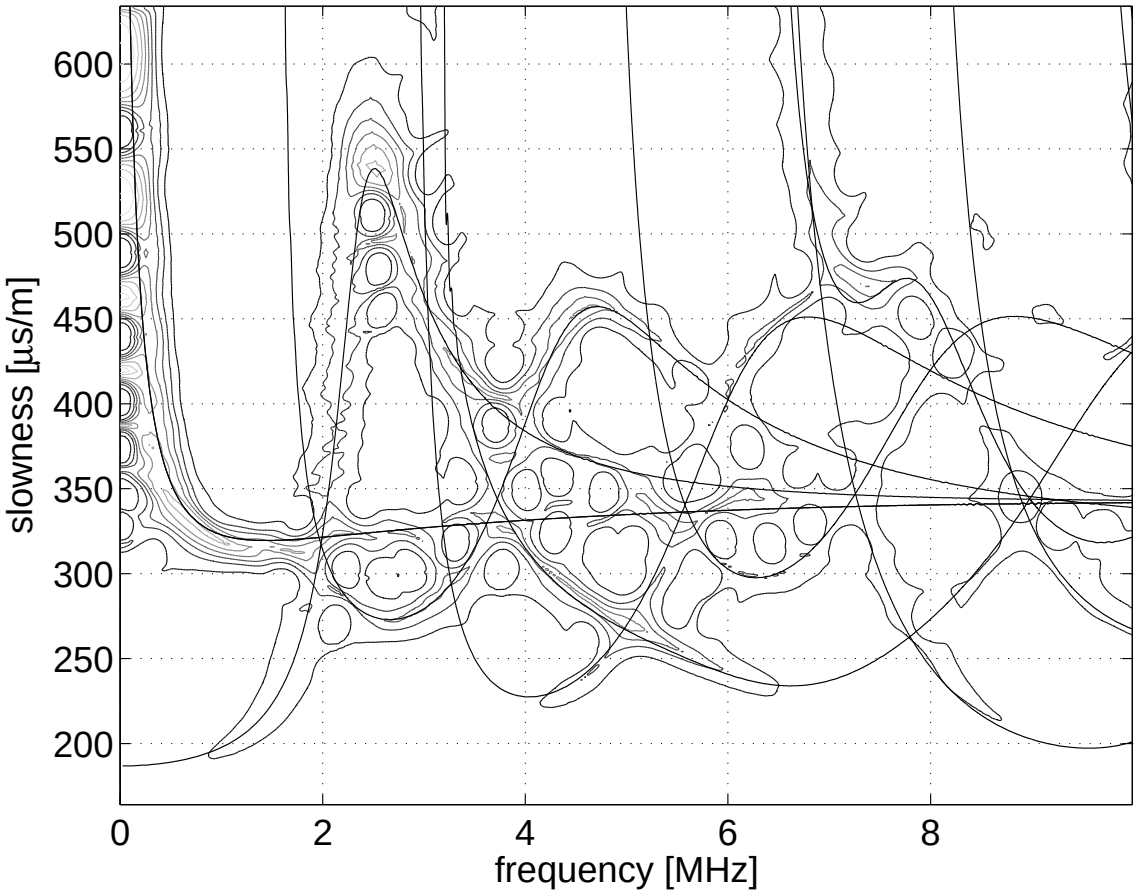


Figure 4.10: reassigned spectrogram after $4,000$ iterations.

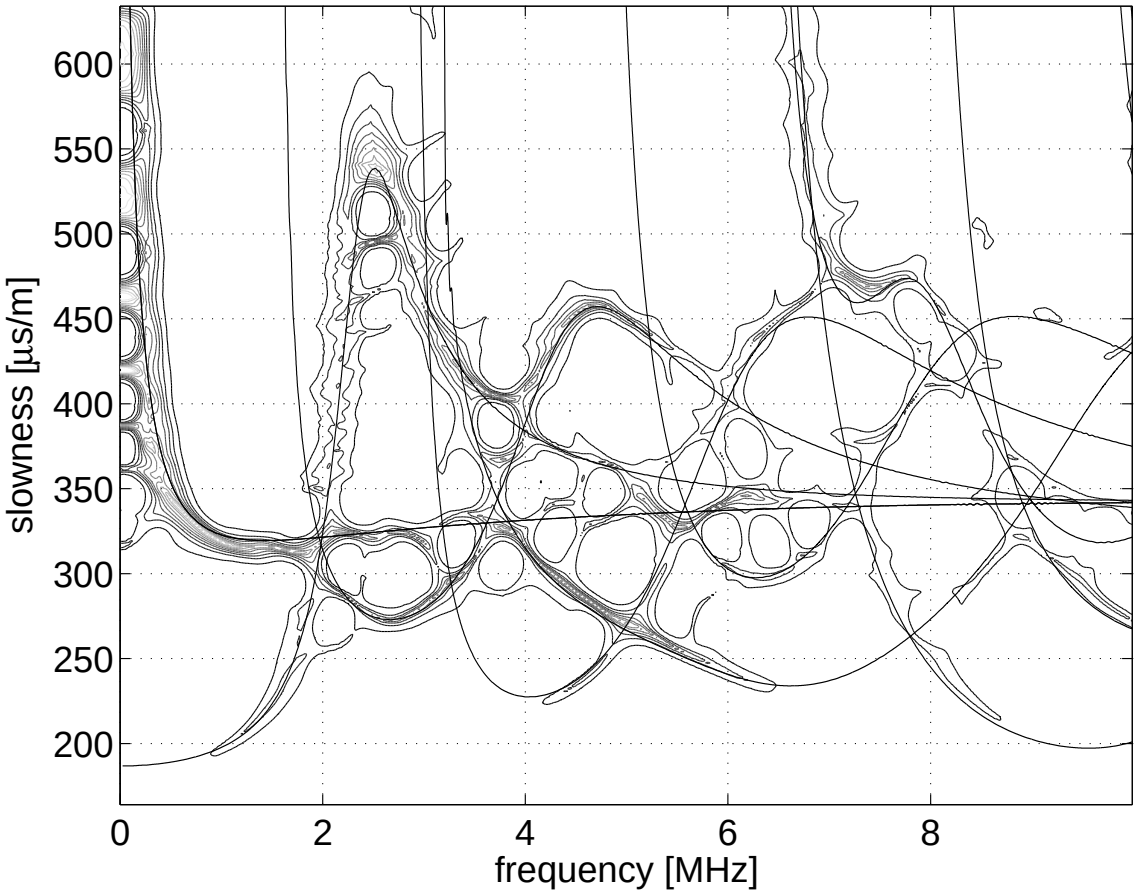


Figure 4.11: reassigned spectrogram after $10,000$ iterations.

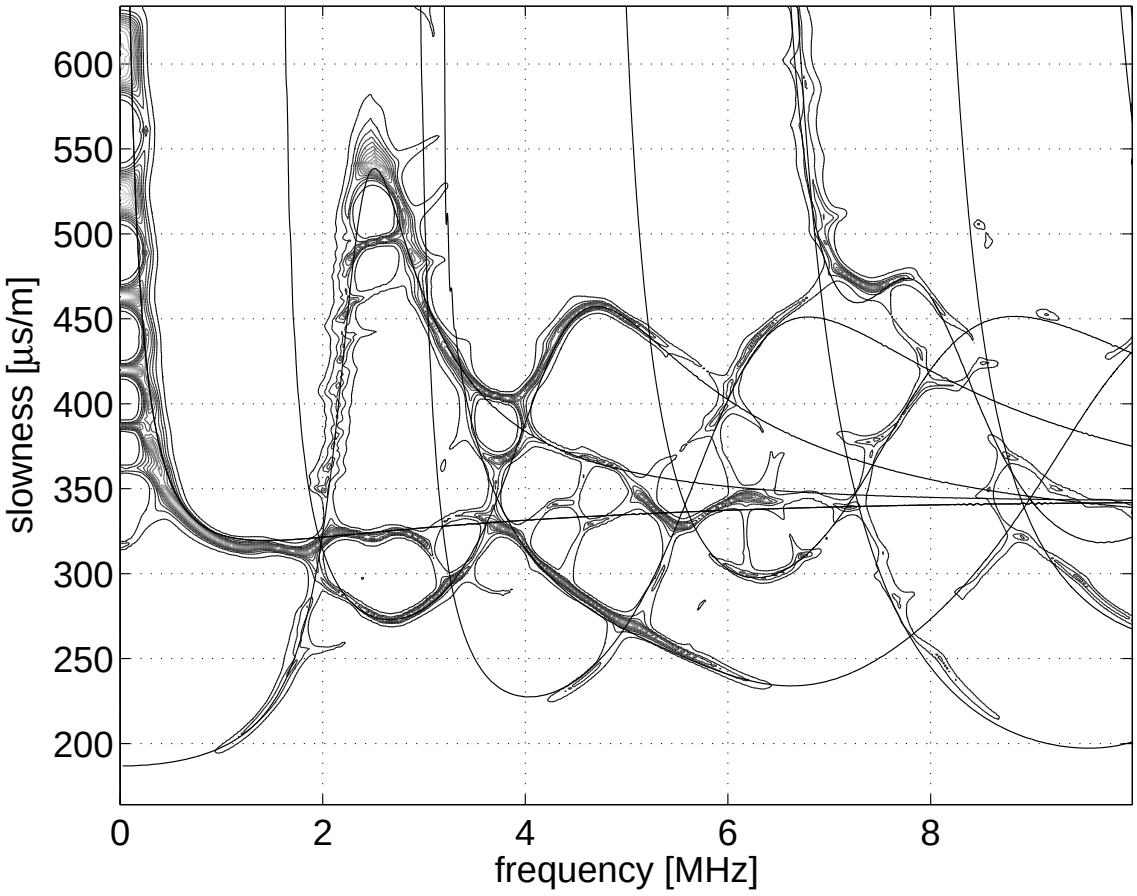


Figure 4.12: reassigned spectrogram after $20,000$ iterations.

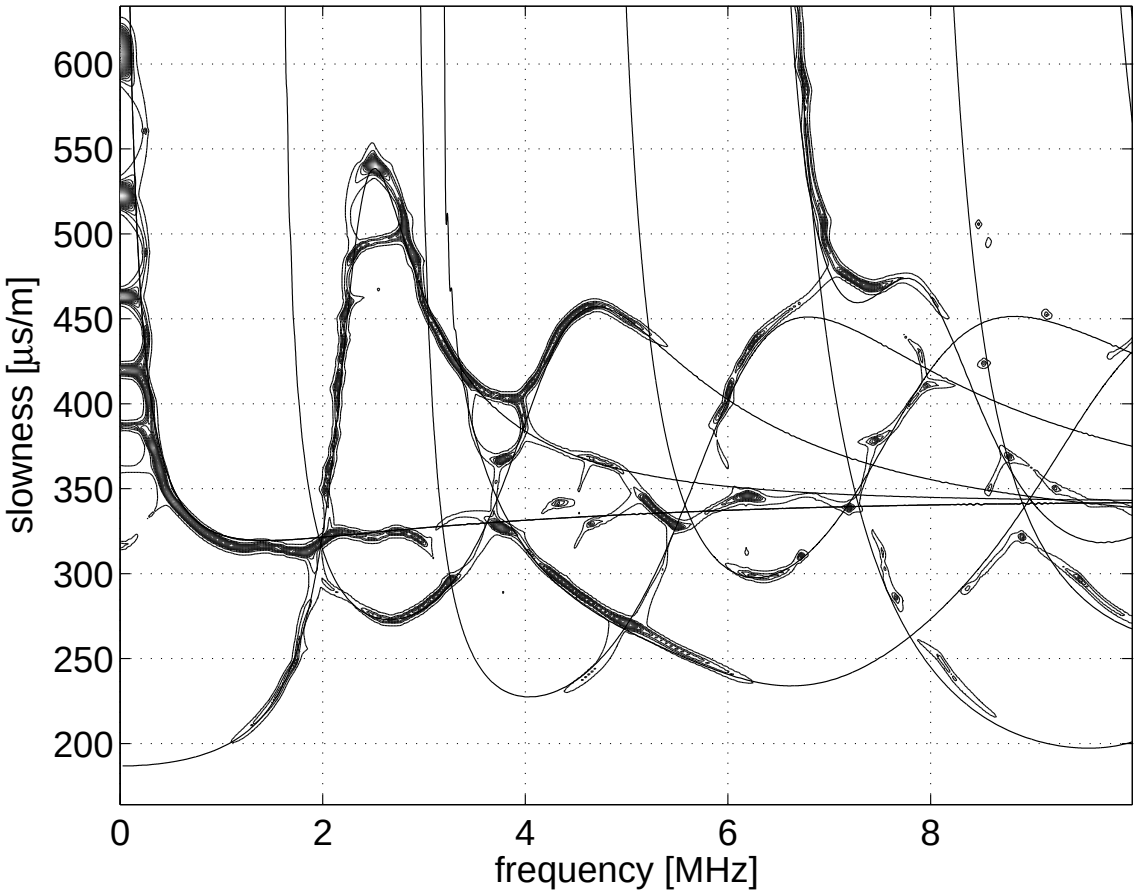


Figure 4.13: reassigned spectrogram after $60,000$ iterations.

Furthermore, differentially reassigned dispersion curves are smoother than their conventionally reassigned counterparts. This becomes even more obvious when considering the close-up views shown in Figures 4.9 and 4.14. Modified differential reassignment results in smoother ridges, as well as elimination of both spurious peaks and noise. Note that anisotropic diffusion of Figure 4.9 does not yield a result which is as localized as Figure 4.14.

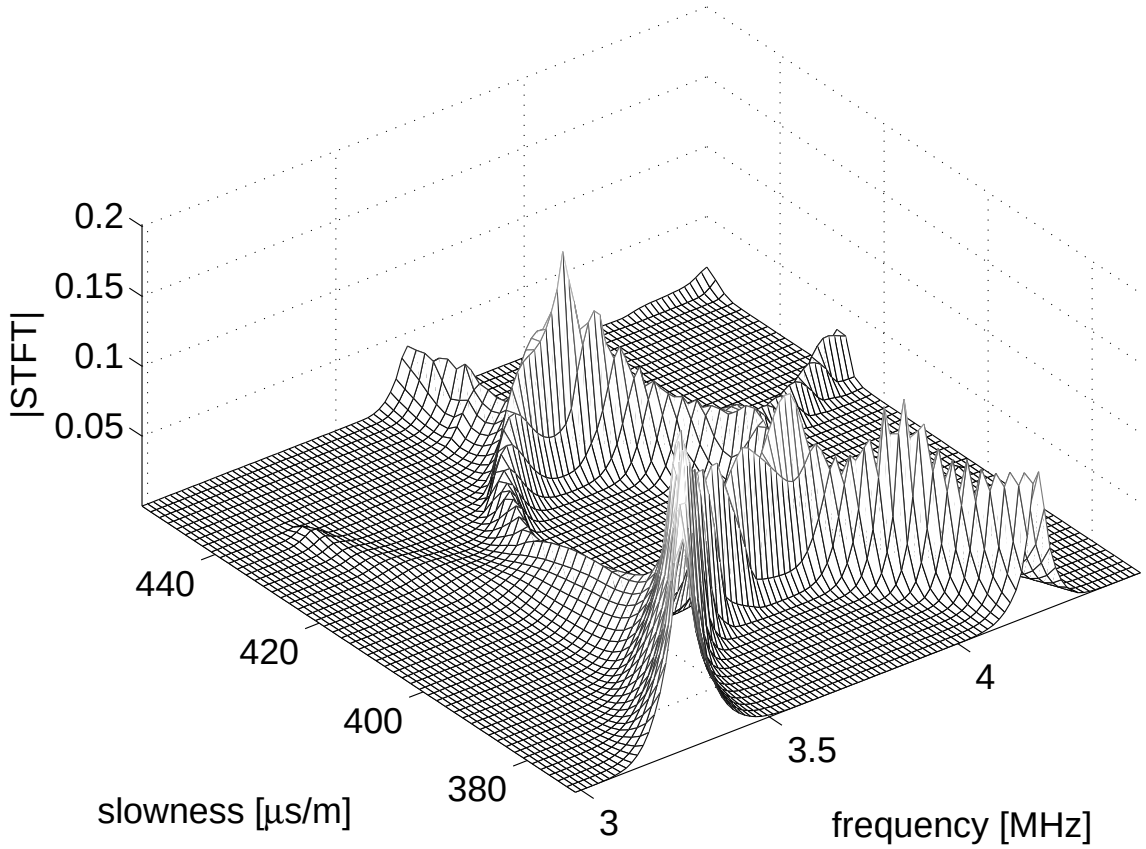


Figure 4.14: zoomed-in view on differentially reassigned spectrogram.

So far, all reassigned spectrograms were energy conservative. If energy conservation is not an issue, but a yes/no decision on measured information is desired, then a threshold δ can be applied to Figure 4.13. All values below δ are set to zero, and all values above δ are set to one. Figure 4.15 shows the obtained image for $\delta = 0.004$.

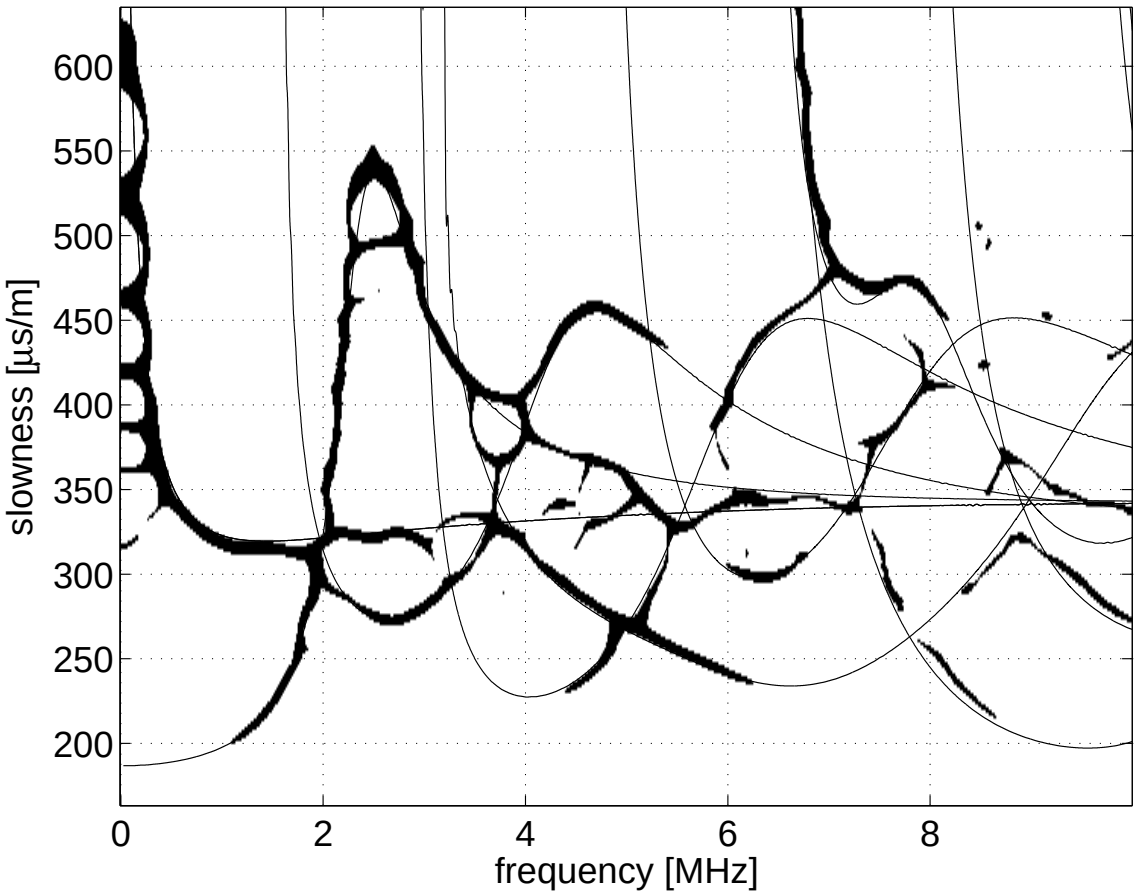


Figure 4.15: yes/no image on measured information.

CHAPTER 5

Notch localization

Previous research by Hurlebaus et al. [14] formulates an autocorrelation-technique to locate a notch in a plate. The autocorrelation procedure in [14] does not require, nor was it developed with, a formal theoretical understanding of the scattering of a Lamb wave by a notch. Later research by Benz [5, 13] refines the technique developed in [14] both by exploiting a theoretical understanding of scattering, and by using the reassigned spectrogram.

As was demonstrated in Section 4.2, the modified differential reassignment algorithm achieves an increase in clarity of the reassigned spectrogram. Hence, this chapter investigates the effects of an improved, reassigned spectrogram on notch localization performance. The first section outlines the notch localization technique developed by [5, 13, 14], using the conventionally reassigned spectrogram. The subsequent section presents the results that were obtained using the modified differential reassignment technique developed in this research.

5.1 Localization using conventional reassignment

The notch localization procedure presented in [14] is based on the reflection off the front edge of the notch. The measurement distances introduced in Figure 4.2 define notch location by its centerline. To consider the finite width of the notch, the new distances

$$d_0' = d_0 - \frac{\omega}{2} \tag{5.1}$$

$$d_3' = d_3 - \frac{\omega}{2} \tag{5.2}$$

are introduced, defining notch location by its front edge.

Autocorrelation is used to localize the notch. Assume $\Delta d_0'$ is the unknown distance from probe 1 to the notch, compare Figure 4.2. By varying systematically $\Delta d_0'$, and thus the associated propagation distance $d = d_1 + 2\Delta d_0'$, slowness-frequency representations (SFRs) are calculated for each distance. Next, autocorrelate each of these SFRs with the incident modes, i.e. with the SFR calculated with distance d_1. Obviously, this autocorrelation reaches its global maximum for a representation calculated with $\Delta d_0' = 0$, as this corresponds to the correlation of two identical signals. Reflections from both the notch and any plate edges will introduce additional, local maxima in the correlation curve at certain $\Delta d_0' > 0$. These local maxima occur when the reflected modes within the SFR coincide with the incident modes — which in turn provides a measure of the receiver-notch distance. A ratio curve is obtained by division of the autocorrelation of the notched plate by the autocorrelation of the perfect plate. This emphasizes the local maxima caused by features that exist in the notched plate but not in the perfect plate. Maxima due to reflections off the plate edges are cancelled out, while those due to reflections off the notch are amplified.

Hurlebaus et al. [14] use an un-reassigned group-velocity frequency representation to localize the notch, whereas Benz [5, 13] chooses a conventionally reassigned SFR. He achieves a further improvement of localization clarity by limiting autocorrelation to the frequency band from 0-2 MHz. This frequency limitation emphasizes the a_0-mode. Reflections do not convert the a_0-mode through a wide frequency bandwidth, making it the most significant mode for notch localization. Using his improvements, Benz [5, 13] obtains the correlation curves shown in Figure 5.1 by varying $\Delta d_0'$ between 0 and 120 mm with an increment step of 0.2 mm. The maximum of the ratio curve locates the notch at $\Delta d_0' = 29$ mm, which is only 0.2 mm different from the actual notch location at 29.2 mm.

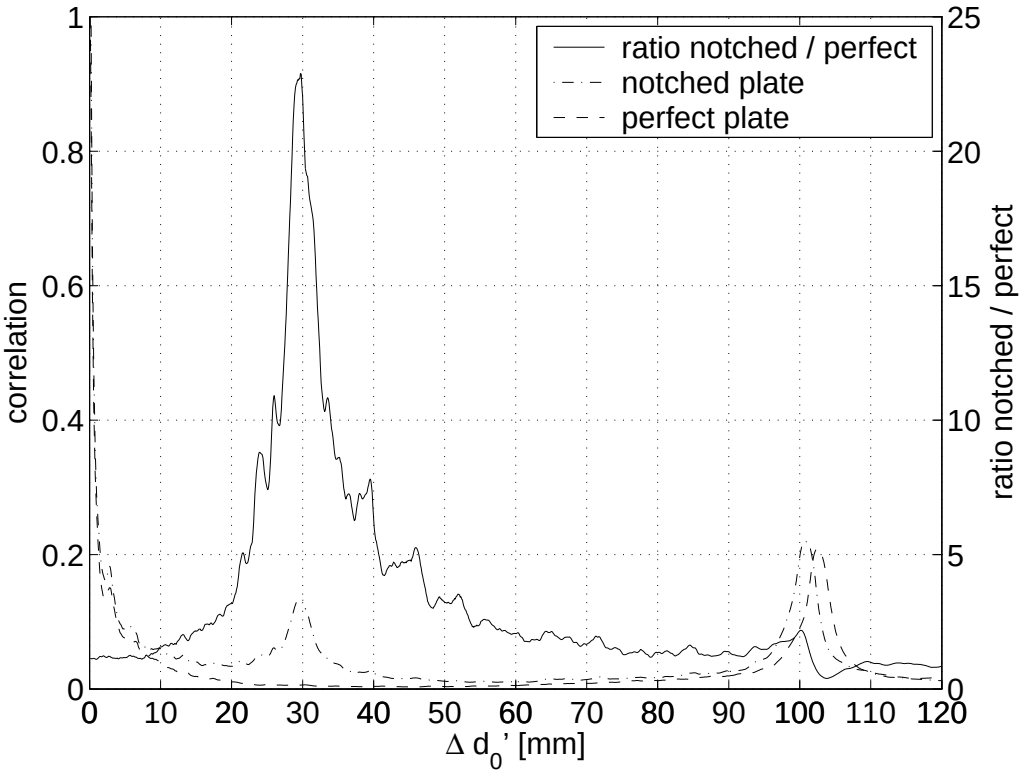


Figure 5.1: Correlation curves for conventional reassignment.

5.2 Localization using modified differential reassignment

This section describes the effects of the modified differential reassignment algorithm on notch localization performance. The SFRs of the notched and perfect plate are reassigned using the algorithm derived in Chapter 3, with parameters $\beta = 5 \cdot 10^{-4}$ and $\kappa = 5$, computing $60,000$ iterations with $\triangle t = 0.0158$. These spectrograms substitute for the conventionally reassigned spectrograms Benz [5, 13] uses to localize the notch.

Autocorrelation following the procedure described in Section 5.1 leads to the correlation curves of the notched and perfect plate — both correlations curves are displayed in Figure 5.2, using the logarithmic scale. Because modified differential reassignment

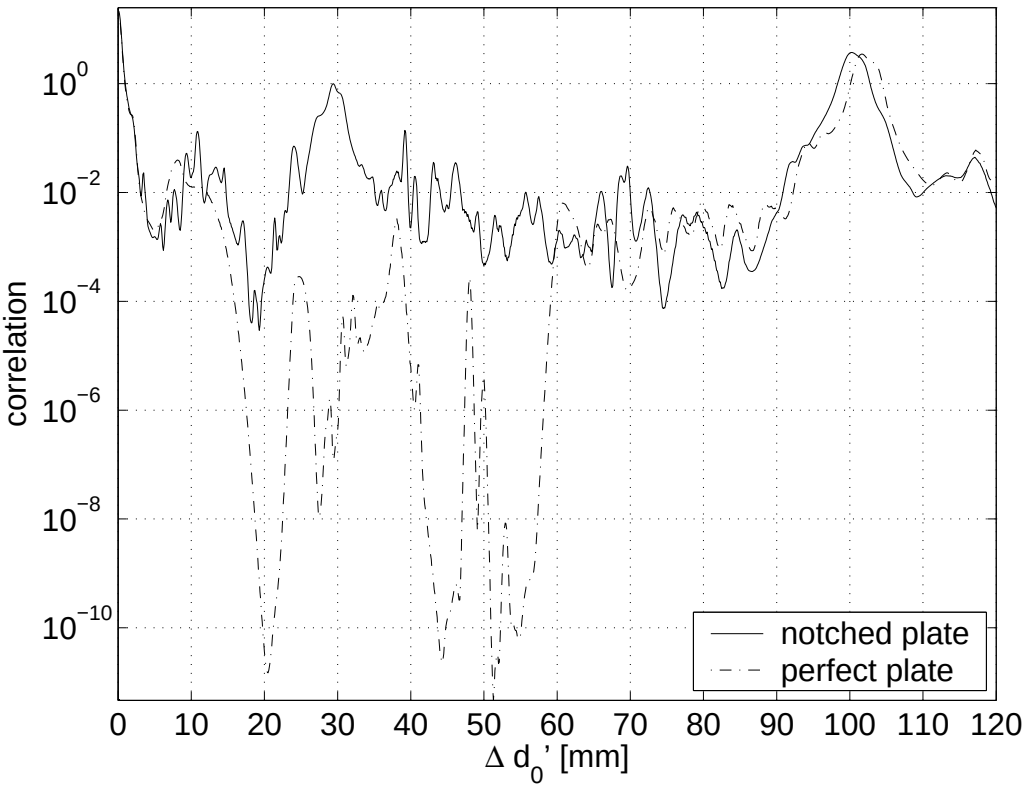


Figure 5.2: Correlation curves for the notched and perfect plate.

eliminates spurious components, the autocorrelation curve of the perfect plate is almost zero for most of the assumed distances $\Delta d_0'$ in Figure 5.2. A division of these

two curves (notched/perfect) is shown in Figure 5.3 — this ratio curve provides poor localization of the notch.

The reason for this is that small variations in the almost zero perfect plate autocorrelation curve dominate the shape of the ratio curve so strongly that the notch at $\Delta d_0' = 29.2$ mm cannot be localized. Note that the almost zero perfect plate autocorrelation curve causes the ratio curve in Figure 5.3 to have much larger values than the ratio curve in Fig 5.1.

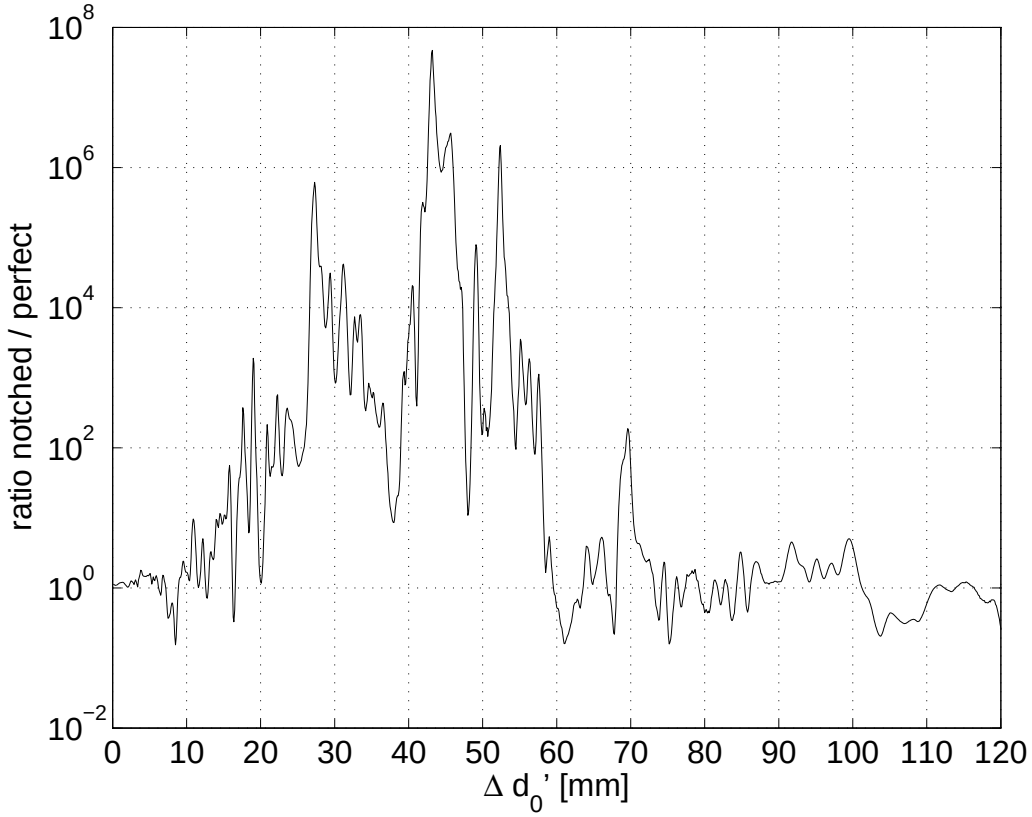


Figure 5.3: Ratio curve.

To guarantee a more accurate accounting of the effect of the notch, the ratio curve must be less sensitive to small variations in the reassigned SFR of the perfect plate curve. This is achieved by introducing a threshold $\delta = 10^{-4}$. Whenever the autocorrelation curve for the perfect plate is smaller than δ, the threshold value is used instead to obtain the ratio curve. This scheme leads to the ratio curve shown in Figure 5.4.

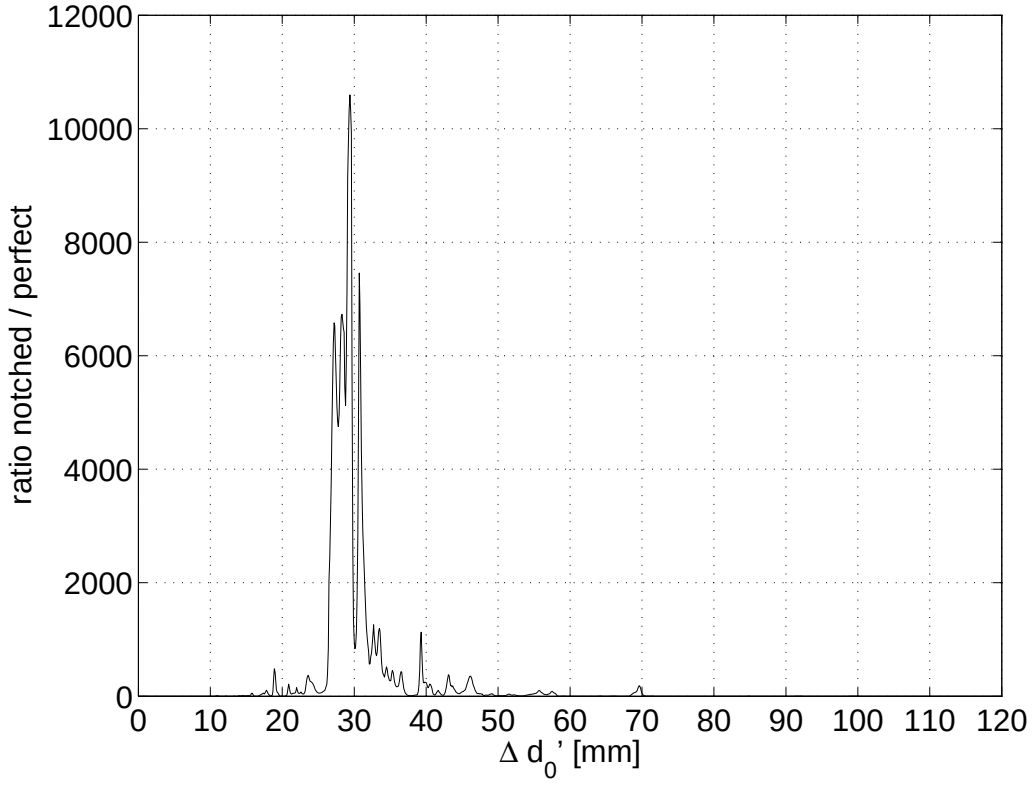


Figure 5.4: Ratio curve with threshold δ.

The notch is localized at $\Delta d_0' = 29.4$ mm, which is again 0.2 mm different from the actual notch location of 29.2 mm. More importantly, this curve is much sharper, and its maximum value is much greater, demonstrating a more unequivocal set of results; the localization results are more reliable if the ratio curve is a sharp impulse at the exact notch location. Figure 5.4 has a sharp impulse very close to the actual notch location, but also has smaller, spurious peaks in the vicinity of the notch. These spurious peaks reduce localization performance, because it is not obvious if these additional peaks localize additional notches, or not.

Consider Figure 4.13. For frequencies below approximately 330 kHz, the a_0 mode lies close to the vertical slowness axis. In this frequency region, the STFT introduces spurious, horizontal ridges, which connect the slowness axis to the a_0 mode. Note that these spurious, horizontal ridges are an effect of the STFT, not of reassignment. A wider window, having higher frequency and less time resolution, would introduce vertical ridges at other locations instead. Hence, introduction of spurious ridges

cannot be avoided by choosing another window type or length.

Recall from Section 4.2 that the time domain signal has been high-pass filtered with a cutoff frequency of 200 kHz prior to calculating the STFT.

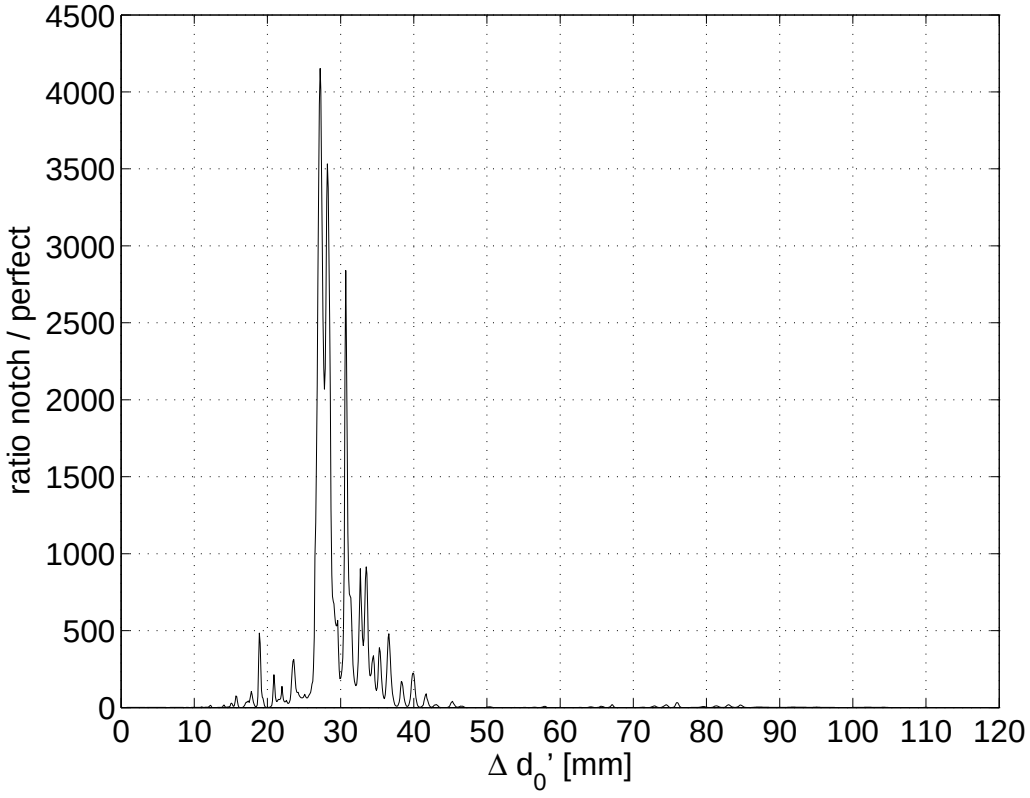


Figure 5.5: Ratio curve for frequencies below 330 kHz, with threshold δ.

As a demonstration, consider an autocorrelation just for frequencies below 330 kHz — Figure 5.5 shows this ratio curve when limiting autocorrelation to frequencies below 330 kHz. This ratio curve is inaccurate since there is a local minimum at the actual notch location of $\Delta d_0' = 29.2$, and there are three spurious peaks close to the actual notch location that correspond to the spurious peak locations in Figure 5.4. Clearly, frequencies below 330 kHz should be excluded from an automated notch localization.

Figure 5.6 shows the ratio curve obtained when limiting autocorrelation to the frequency band 330 kHz – 2 MHz, together with the ratio curve of Figure 5.1.

Both reassignment methods lead to a very accurate notch localization, each being off by only 0.2 mm. However, this research obtains a peak which is much closer

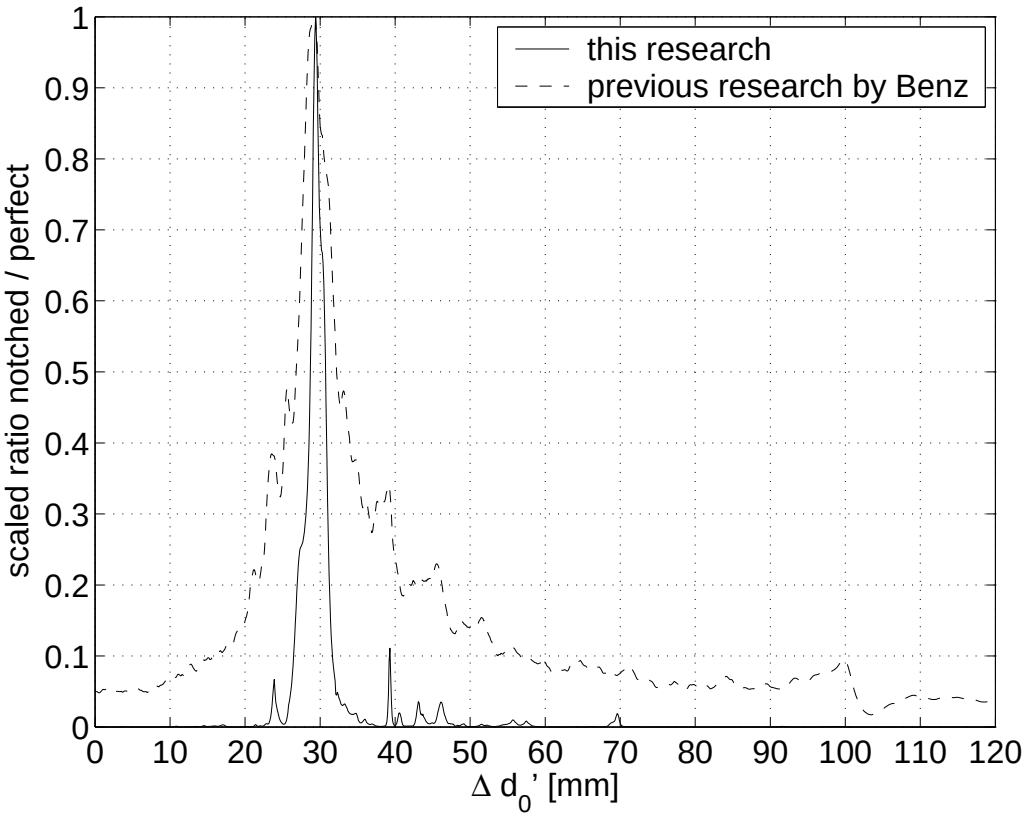


Figure 5.6: Ratio curve for the frequency band 330 kHz - 2 MHz, with threshold δ.

to a sharp impulse. Thus, notch localization performance has been significantly increased by using the modified differential reassignment algorithm, and by limiting autocorrelation to the frequency band 330 kHz - 2 MHz.

CHAPTER 6

Conclusion

This thesis develops a modification of differential reassignment. It derives a partial differential equation (PDE) which fully describes image evolution due to energy movement. Differential reassignment is then combined with anisotropic diffusion. Finally, travel velocity along reassignment trajectories is changed to allow for a more favorable evolution with time.

A further contribution of this study is the application of the derived PDE to reassign experimentally measured spectrograms for multi-mode guided waves. The modified differential reassignment technique enhances the clarity of reassigned slowness–frequency–representations (SFR). Spurious components are eliminated, and mode dispersion curves are smoothed. These results constitute a qualitative way to measure the increase in reassignment performance.

In a next step, this research uses improved, reassigned spectrograms for notch localization. For this purpose, the reassigned SFRs of a notched and perfect plate are autocorrelated. This research further refines the automated notch localization technique by excluding low frequencies from the autocorrelation process. The results achieved show an obvious improvement of notch localization performance. These results constitute a more quantitative way to measure the increase in reassignment performance.

This work makes a significant contribution by applying image processing methodologies to quantitative nondestructive evaluation. It clearly demonstrates the effectiveness of applying image processing techniques to representations of experimentally measured dispersion data. Image processing is not capable of creating new information. However, it does provide means to view the same data in a different fashion,

enabling extraction of formerly hidden information. By doing so, this thesis strengthens a link in the data processing chain to detect and localize notches in plate–like structures.

Bibliography

[1] J. D. Achenbach. *Wave propagation in elastic solids*. North-Holland, 1975.

[2] E. Chassande-Mottin, F. Auger, and P. Flandrin. Supervised time-frequency reassignment. *IEEE Int. Symp. on Time-Frequency and Time-Scale Analysis*, pages 517–520, 1996.

[3] E. Chassande-Mottin, I. Daubechies, F. Auger, and P. Flandrin. Differential reassignment. *IEEE signal processing letters*, 4(10):293–294, 1997.

[4] F. Auger and P. Flandrin. Improving the readability of time-frequency and time-scale representations by the reassignment method. *IEEE transactions on signal processing*, 43(5):1068–1089, May 1995.

[5] R. Benz. Localization of notches with lamb waves. Msc, Georgia Institute of Technology, School of Civil and Environmental Eng., 2003.

[6] E. Chassande-Mottin. *Méthodes de réallocation dans le plan temps-fréquence pour l'analyse et le traitement de signaux non stationnaires*. PhD thesis, Université de Cergy-Pontoise, 1998. in French.

[7] L. Cohen. Time-frequency distributions – a review. In *Proceedings of the IEEE*, volume 77, pages 941–981, July 1989.

[8] L. Cohen. *Time-frequency analysis*. Prentice Hall, 1995.

[9] O. Föllinger. *Laplace- und Fourier-Transformation*. Hüthig, 1993. in German.

[10] K. Kodera, R. Gendrin, and C. de Villedary. Analysis of time-varying signals with small bt values. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):64–76, February 1978.

[11] P. L. Gould. *Introduction to linear elasticity*. Springer Verlag, 1994.

[12] K. F. Graff. *Wave motion in elastic solids*. Dover publications, 1975.

[13] R. Benz, M. Niethammer, S. Hurlebaus, and L. J. Jacobs. Localization of notches with lamb waves. *Journal of the Acoustical Society of America*, 114(2):677–685, 2003.

[14] S. Hurlebaus, M. Niethammer, L. J. Jacobs, and C. Valle. Automated methodology to locate notches with lamb waves. *Acoustics Research Letters Online*, 2(4):97–102, 2001.

[15] T. Kreuzinger. Digital signal processing methods for source function extraction of piezoelectric elements. Msc, Georgia Institute of Technology, School of Civil and Environmental Eng., 2004.

[16] R.J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002.

[17] G. Longo and B. Picinbono. *Time and frequency representations of signals and systems*. Springer Verlag, 1989.

[18] K. Meyberg and P. Vachenauer. *Höhere Mathematik 2*. Springer Verlag, 1991. in German.

[19] I. N. Bronstein, K. A. Semendjajew, G. Musiol, and H. Mühlig. *Taschenbuch der Mathematik*. Verlag Harry Deutsch, 1997. in German.

[20] M. Niethammer. Application of time frequency representations to characterize ultrasonic signals. Msc, Georgia Institute of Technology, School of Civil and Environmental Eng., 2000.

[21] A. V. Oppenheim and R. W. Schafer. *Discrete-time signal processing*. Prentice Hall, 1999.

[22] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[23] M. Niethammer, L. J. Jacobs, J. Qu, and J. Jarzynski. Time-frequency representation of lamb waves. *Journal of the Acoustical Society of America*, 109(5):1841–1847, 2001.

[24] J. L. Rose. *Ultrasonic waves in solid media*. Cambridge University Press, 1999.

[25] P. J. Olver, G. Sapiro, and A. Tannenbaum. Affine invariant detection: Edge maps, anisotropic diffusion, and active contours. *Acta Applicandae Mathematicae*, 59:45–77, 1999.