NTNU

Norwegian University of
Science and Technology

Application of the wavelet transform for analysis of ultrasound images

Eivind Kleiven

Master of Science in Physics and Mathematics
Submission date: May 2008
Supervisor: Yurii Lyubarskii, MATH

Norwegian University of Science and Technology
Department of Mathematical Sciences

Problem Description

The project deals with analysis of medical ultrasound images. More specifically
we plan to apply the wavelet transform and other similar techniques for edge detection and edge characterisation in US images. The project will be carried out in collaboration with SINTEF Helse.

Assignment given: 14. December 2007
Supervisor: Yurii Lyubarskii, MATH

Preface

This is the master thesis for the master's degree "Master of science in physics and mathematics" within the field of specialization "Industriell matematikk" at the Norwegian university of science and technology; NTNU. The project was carried out in collaboration with SINTEF Helse.

I would like to thank professor Yurii Lyubarskii who was my supervisor for this thesis. He has been helpfull with ideas and motivation throughout the project. I also thank Thomas Langø and Sébastien Muller from SINTEF Helse for fruitful discussions and for providing images which where used in this report.

Trondheim, 16. May 2008

Abstract

In this master thesis we analyse medical ultrasound images using the wavelet transform. Mathematical theory is introduced for both one-dimensional and two-dimensional functions. Three edge detectors based on the mathematical theory introduced are given. Two of the three edge detectors are suggested by the author and one is an implementation of a known edge detector often refered to as the Canny edge detector. Our implementation will differ slightly from the original Canny edge detector since in our implementation we use the wavelet transform. All three edge detectors are applied on several images and the result is discussed.

The multiscale behavior of the wavelet transform makes it usefull for edge detection. For small scales it is sensitive to noise, but with good localisation of edges. For large scales it is not as sensitive to noise, but with poorer localisation. One problem when designing an edge detector is to find the scale that have the best trade-off between localisation and noise sensitivity. We suggest an algorithm that automatic selects this scale using information from the wavelet transform across larger scales. The result is an algorithm that works satisfactorily for a set of images that differs in amount of noise and contrast between objects in the image.

An edge detector for one-dimensional signals are given. This edge detector works very well for locating singularities and characterising Lipschitz regularity in one-dimensional signals. However, as an edge detector for images it does not function satisfactorily.

Further investigation should be done on how to use the multiscale information carried by the wavelet transform. The author are convinced that better edge detectors that are less sensitive to noise with good localisation properties can be derived using the wavelet transform across scales.

Contents

List of Figures

Chapter 1

Introduction

The use of automatic edge detection is versatile and there are much litterature about edge detectors. However, automatic edge detection is not an easy task and often a specific problem needs a specific solution. An edge detector simplify the analysis of images by reducing the amount of data to be processed, while usefull structural information is preserved. For a landscape photograph a different edge detector should be used than if the image to be analysed is an ultrasound image. Also for an ultrasound image different edge detectors should be used depending on what we are interested in finding. For example if we are looking for round objects maybe we should find an edge detector that specializes on finding round objects.

The main goal for this thesis is to use the wavelet transform for edge detection in noisy ultrasound images. We look at three edge detectors using the wavelet transform and see how well they perform on a set of images. Optimally we want an edge dector that are able to outline the border between healthy tissue and tumors based on ultrasound images. We have no foreknowledge about the shape of the tumor, also the contrast between healthy tissue and tumors may vary. Sometimes a tumor distinguish clearly from the healthy tissue while other times it is very hard to see the tumor contour.

Ultrasound images are often used in situations where the surgeon cannot observe the tumor directly. In such cases observations are performed using ultrasound and the surgeon must make decisions based on ultrasound images. For example when there are disturbing elements like blood or other tissue preventing visual observations of the tumor itself an ultrasound image may be used to locate the tumor. Ultrasound images tend to be very noisy and it can be difficult deciding what are tumor cells and what are not. Edge detectors can help the surgeon making such descisions by outlining the important structures in the ultrasound image.

Ultrasound does not affect the patient worth mentioning and it is realative easily performed compared to other methods such as MR. However, ultrasound images are very noisy which makes it difficult to seperate objects in the image. Most of the noise is due to interference between echoes of the sound waves sendt by the ultrasound probe, this is called speckle noise. Also, the electronic instruments adds some noise to the image. Therefore edge detection in ultrasound images are not an easy task.

In general the intensity changes in ultrasound images due to noise have a shorter duration than the ones corresponding to important objects. One might say that the noise are small structures in the image. The word structure has no precise meaning here, but it is to be understood intuitively. For example we may have an image of a chessboard lying on top av a table in which we would like to locate the chessboard. There will be high intensity changes between each square on the chessboard as well as in the transition between the chessboard and the table. In this case the chessboard is a larger structure than the texture of the chessboard. The wavelet transform is a multiscale transform and for large scales it is mainly sensitive to intensity changes caused by large structures, while at smaller scales it is also sensitive to intensity changes caused by smaller structures. This multiscale behavior makes the wavelet transform usefull for edge detection.

This thesis has three main chapters: Chapter 2, Chapter 3 and Chapter 4. Chapter 2 introduce the mathematical theory that will be the basis for the final edge detectors described in Chapter 4. Lipshictz regularity is introduced and related to edges. We see how Lipschitz regularity may be characterized from the decay of the wavelet transform. Therefore, the wavelet transform can be used both to locate edges and to characterize edges. We will look at theory for both one-dimension and two-dimensions.

Chapter 3 is a survey of some articles concerning reconstruction of signals from samples of the wavelet transform at local extremum positions. Also we glance at how the wavelet transform may be used in an edge detection similar to the one given by Canny[3].

In Chapter 4 we give a 1-D multiscale edge detector which both locate edges and characterize Lipschitz regularity for 1-D signals. The 2-D single scale edge detector inspired by the edge detector given by Canny is implemented. One disadvantage with this edge detector is that a fixed set of input parameters will only fit a small class of images. Lastly we suggest a 2-D multiscale edge detector such that a fixed set of input parameters will fit a bigger class of images. All of the discussed edge detectors are tested on a set of images. Thomas Langø and Sébastien Muller at SINTEF Helse have kindly supplied us with both real ultrasound images and phantom images with simulated noise.

Chapter 2

Matemathical approach

When designing an edge detection algorithm one should be familiar with the theory one want to apply in the algorithm. In this report we will use the wavelet transform for edge detection and therefore start by introducing wavelets.

An image may be represented as a set of one-dimensional functions or as a two-dimensional function. In the one-dimensional case each function correspond to some stripe of the image while in the two-dimensional case one function correspond to the whole image. Both representations have it's advantages and disadvantages when it comes to edge detection. The one-dimensional representation tend to have algorithms that are easier to implement than in the two-dimensional case, but in general it gives poorer detection. In the two-dimensional case one can combine image information in both x-direction and y-direction when detecting edges. In the one-dimensional case we only analyse one direction at a time. In Chapter 4 an edge detector for each representation is given.

We start by introducing the wavelet transform in one-dimension so the number of parameters don't obscure the ideas. Then, in the next section, we extend the theory into two-dimensions.

2.1 1-D analysis

2.1.1 Some notations

In this subsection we introduce some notations and definitions we will use throughout this report.

For any function $g(t)$ write

$$g_s(t) = \frac{1}{\sqrt{s}} g(\frac{-t}{s}) \text{ and } g_{u,s}(t) = g_s(u-t)$$

The parameter s dilates the function while u is a translation parameter. The factor $\frac{1}{\sqrt{s}}$ preserves the energy when dilating the function.

In this report a *1-D signal* is any function from the space $L^2(\mathbb{R})$ specified in Definition 2.1.

Definition 2.1. $L^2(\mathbb{R})$ is the space of functions

$$f : \mathbb{R} \mapsto \mathbb{C}$$

such that

$$\int_{-\infty}^{\infty} |f(t)|^2 dt < \infty \tag{2.1}$$

with the inner product and norm

$$< f, g > = \int_{-\infty}^{\infty} f(t)\overline{g(t)}dt \tag{2.2}$$

$$\|f\| = \left(\int_{-\infty}^{\infty} |f(t)|^2 dt \right)^{\frac{1}{2}} \tag{2.3}$$

Definition 2.2. *The energy of a signal* f is given by

$$E = \|f\|^2 = \int_{-\infty}^{\infty} |f(t)|^2 dt \tag{2.4}$$

2.1.2 The wavelet transform

A well known tool in signal processing is the Fourier transform which gives global information about the frequency spectrum $\widehat{f}$ of a 1-D signal f.

Definition 2.3. Let $f \in L^2(\mathbb{R})$, then

$$\widehat{f}(\omega) = \int_{-\infty}^{\infty} f(t) e^{-2i\pi\omega t} dt \tag{2.5}$$

is called *the Fourier transform of* f.

From now $\widehat{f}$ will always denote the Fouriere transform of f. We will use the word *frequency domain* when we talk about $\widehat{f}(\omega)$ and *time domain* about $f(t)$

In edge detection it is not enough to be able to say if an edge exists somewhere or not, we need to find the position of the edge as well. The wavelet transform is a *multiscale transform* which is suitable for edge detection. In this subsection we see that the wavelet transform contains localized information of f in both time and frequency domain.

We define a wavelet in terms of *the admissibility condition* (2.6):

Definition 2.4. A wavelet is a function $\psi(t) \in L^2(\mathbb{R})$ satisfying

$$0 < \int_0^\infty \frac{|\hat{\psi}(\omega)|^2}{\omega} d\omega < \infty \tag{2.6}$$

This definition ensures that dilation of $\widehat{\psi}$ will cover the whole frequency domain and that f may be reconstructed from it's wavelet transform. These assertions will be stated more precisely later in this subsection.

For continuous $\widehat{\psi}$ the admissibility condition implies that $\hat{\psi}(0) = 0$.

$$\widehat{\psi}(0) = \int_{-\infty}^{\infty} \psi(t)\, dt = 0$$

So a wavelet has zero average and therefore must have some oscillations and might be associated with a wave. We also want ψ and $\widehat{\psi}$ to be well localized, i.e they should both converge to zero quite rapid, hence the word "wavelet". It should be mentioned that Heisenberg's uncertainty principle A.2 gives a lower bound on how well localized ψ and $\widehat{\psi}$ can be. Better localisation in time will give worser localisation in frequency.

Definition 2.5. A *wavelet atom* is a dilated and translated wavelet ψ denoted by

$$\psi_{u,s}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right)$$

The wavelet transform will be defined in terms of wavelet atoms and in that context ψ is refered to as the mother wavelet. The factor $\frac{1}{\sqrt{s}}$ is a normalization factor such that a wavelet atom and it's mother wavelet has the same energy.

$$\|\psi_{u,s}\|_2^2 = \int_{-\infty}^{\infty} |\psi_{u,s}(t)|^2 dt = \|\psi\|_2^2$$

The parameter u translates the wavelet by a distance u while the parameter s dilates the wavelet proportional to s.

Definition 2.6. Let $f \in L^2(\mathbb{R})$ and $\{\psi_{u,s}\}_{u,s\in\mathbb{R},s>0}$ be a family of wavelet atoms, then *the wavelet transform of f with respect to ψ* is defined as

$$Wf(s,u) =< f, \psi_{u,s} >= \int_{-\infty}^{\infty} f(t)\frac{1}{\sqrt{s}}\psi\left(\frac{t-u}{s}\right)dt \tag{2.7}$$

The wavelet transform of f is the inner product between a dilated and shifted mother wavelet ψ and f. Let us take a closer look at what the parameters u and s mean for $Wf(u,s)$, but first we remember Parseval's equality.

Parseval's equality 2.1. *If $f, g \in L^2(\mathbb{R})$ then*

$$\int_{-\infty}^{\infty} f(t)\,\overline{g(t)}dt = \int_{-\infty}^{\infty} \widehat{f}(\omega)\,\overline{\widehat{g}(\omega)}d\omega$$

Using Parseval's equality we get

$$Wf(u,s) = \int_{-\infty}^{\infty} \widehat{f}(\omega)\,\overline{\widehat{\psi_{u,s}}(\omega)}d\omega$$

If for some (u,s) $\psi_{u,s}(t)$ is neglectible for all $t \notin (a,b)$ then $Wf(u,s)$ contains information about $f(t)$ only when $t \in (a,b)$. The behavior of $f(t)$ outside the intervall does not affect $Wf(u,s)$.

Also, if for some (u,s) $\widehat{\psi_{u,s}}(\omega)$ is neglectible for all $\omega \notin (\eta,\xi)$ then $Wf(u,s)$ contains information about $\hat{f}(\omega)$ only when $\omega \in (\eta,\xi)$. The behavior of $\hat{f}(\omega)$ outside the intervall does not affect $Wf(u,s)$.

We would like Wf to contain information about $f(t)$ for all t and $\widehat{f}(\omega)$ for all ω. It is clear that $\{\psi_{u,s}\}_{u\in\mathbb{R}}$ covers the whole time domain. As promised earlier in this subsection we now show that $\{\widehat{\psi_{u,s}}\}_{s\in\mathbb{R}^+}$ covers the whole frequency domain. First we notice the following properties:

Property 2.2. *Let $f \in \mathrm{L}^2(\mathbb{R})$ then*

(i) $\widehat{f_s}(\omega) = -\sqrt{s}\widehat{f}(-s\omega)$

(ii) $\widehat{f_{u,s}}(\omega) = \sqrt{s}e^{-2i\pi u\omega}\widehat{f}(s\omega)$

Proof.

$$\widehat{f}_{u,s}(\omega) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{s}} f(\frac{x-u}{s}) e^{-2i\pi x\omega} dx \tag{2.8}$$

$$= \int_{-\infty}^{\infty} \frac{1}{\sqrt{s}} f(\tau) e^{-2i\pi s\omega\tau} e^{-2i\pi\omega u} s d\tau \tag{2.9}$$

$$= \sqrt{s} e^{-2i\pi u\omega} \widehat{f}(s\omega) \tag{2.10}$$

The other statement is proved using the same procedure. □

From Property 2.2 we know $|\widehat{\psi_{u,s}}(\omega)| = |\sqrt{s}\widehat{\psi}(s\omega)|$. If the frequency domain is covered by dilations of $\widehat{\psi}$ then for each $\omega \in \mathbb{R}$ there must exist some $s > 0$ such that $|\widehat{\psi}(s\omega)| > 0$. This can be formulated as follows. For each $\omega \in \mathbb{R}$ we have

$$0 < \int_0^{\infty} \frac{|\psi(s\omega)|^2}{s} ds < \infty \tag{2.11}$$

Using substitution $\tau = s\omega$ this is exactly the same as the admissibillity condition (2.6) and therefore satisfied by every wavelet.

So now we know that Wf contains information about f from the whole time domain and frequency domain. Actually, since dilation and translation are done continuously it might be that Wf contains too much information. In the next chapter we look into some algorithms reconstructing an approximation of f using only the value of Wf at extremum points or zero crossing positions.

We notice that in the frequency-domain the parameter s dilates the mother wavelet proportional to s and in the time-domain proportional to s^{-1}. This means that if we decrease s then we get a better localization in time, but a worser localization in frequency. It is this property which makes the wavelet transform usefull for multiscale edge detection. For large s the transform is not very sensitive to noise, on the other hand the detected edge might be far away from the real edge. By using information for several values of s we might better detect edges in noisy images.

Let us look at another property of the wavelet transform.

Definition 2.7. $f \in L^2(\mathbb{R})$ is said to have n *vanishing moments* if

$$\int_{-\infty}^{\infty} t^k f(t) dt = 0 \text{ for } k = 0, \ldots, n-1$$

If a wavelet ψ has $(n+1)$ vanishing moments the corresponding wavelet transform is zero for any polynomial of degree n. In the next subsection we see that this property makes it possible to characterize the regularity of f.

It is time to introduce the reconstruction formula. Theorem 2.3 states that any $f \in L^2(\mathbb{R})$ can be reconstructed from Wf and that Wf preserves energy.

Theorem 2.3. *Let $\psi \in \mathrm{L}^2(\mathbb{R})$ be such that*

$$C_\psi = \int_0^\infty \frac{|\widehat{\psi}(\omega)|^2}{\omega} d\omega < \infty \tag{2.12}$$

Then, for each $f \in \mathrm{L}^2(\mathbb{R})$

$$f(t) = \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^\infty Wf(u,s) \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right) du \frac{ds}{s^2}$$

and

$$\|f\|^2 = \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^\infty |Wf(u,s)|^2 du \frac{ds}{s^2} \tag{2.13}$$

[6, p. 81]

Definition 2.8. A *smoothing function* is any function $\theta(t) \geq 0$ whose integral is equal to one and that converges to zero at infinity.

Definition 2.9. A function θ is said to have *fast decay* if

$$\lim_{t \to \pm\infty} |t|^k \theta(t) \to 0 \text{ for each } k \in \mathbb{N}$$

Theorem 2.4. *A wavelet ψ with fast decay has n vanishing moments if and only if there exists a θ with a fast decay such that*

$$\psi(t) = (-1)^n \frac{d^n \theta(t)}{dt^n} \tag{2.14}$$

[6, p. 167]

We end this subsection with an example of two wavelets ψ_1 and ψ_2 having 1 and 2 vanishing moments respectively. In chapter 4 ψ_1 is used in an one-dimensional edge detection algorithm.

Example 2.1. *Define ψ_1 and ψ_2 as the first- and second-order derivative of the normalized Gaussian function θ.*

$$\theta(t) = \frac{1}{\sqrt{\pi}} e^{-t^2}$$
$$\psi_1(t) = (-1)\frac{d}{dt}\theta(t) = \frac{2}{\sqrt{\pi}} t e^{-t^2}$$
$$\psi_2(t) = \frac{d^2}{dt^2}\theta(t) = \frac{2}{\sqrt{\pi}} e^{-t^2}(t^2 - 1)$$

Since θ is a smoothing function with fast decay then from Theorem 2.4 we know that ψ_1 and ψ_2 has 1 and 2 vanishing moments respectively. Let us evaluate $\widehat{\psi_1}$ and $\widehat{\psi_2}$. Since θ is a Gaussian function then $\widehat{\theta}$ is also Gaussian.

$$\widehat{\theta}(\omega) = e^{-\pi^2\omega^2} \tag{2.15}$$

For the interested reader an evaluation of (2.15) is given in [2, p. 159]. Using Proposition A.4 we get

$$\widehat{\psi_1}(\omega) = \widehat{(-1)\theta^{(1)}}(\omega) = -2i\pi\omega e^{-\pi^2\omega^2}$$
$$\widehat{\psi_2}(\omega) = \widehat{\theta^{(2)}}(\omega) = -4\pi^2\omega^2 e^{-\pi^2\omega^2}$$

Figure 2.1 shows θ, ψ_1 and ψ_2 togheter with their Fourier transforms.

Let us verify that ψ_1 and ψ_2 fulfil (2.6) and thus are wavelets.

$$\begin{aligned}\int_0^\infty \frac{|\widehat{\psi_1}(\omega)|^2}{\omega} d\omega &= \int_0^\infty 4\pi^2\omega e^{-2\pi^2\omega^2} d\omega \\ &= \lim_{a\to\infty} \big[-e^{-2\pi^2\omega^2} \big]_0^a \\ &= \lim_{a\to\infty} \Big(-e^{-2\pi^2 a^2} - (-e^0) \Big) = 1\end{aligned}$$

Using substitution $\tau = \omega^2$ and integration by parts we verify ψ_2.

$$\begin{aligned}\int_0^\infty \frac{|\widehat{\psi_2}(\omega)|^2}{\omega} d\omega &= \int_0^\infty 16\pi^4\omega^3 e^{-2\pi^2\omega^2} d\omega \\ &= \int_0^\infty 8\pi^4\tau e^{-2\pi^2\tau} d\tau \\ &= \lim_{a\to\infty} \big[-4\pi^2\tau e^{-2\pi^2\tau} \big]_0^a + \int_0^\infty 4\pi^2 e^{-2\pi^2\tau} d\tau \\ &= 0 + \lim_{a\to\infty} \big[-2e^{-2\pi^2\tau} \big]_0^a = 2\end{aligned}$$

So by Definition 2.4 ψ_1 and ψ_2 are wavelets.

From Figure 2.1(d) and (f) we see that $\widehat{\psi_1}$ and $\widehat{\psi_2}$ attenuate frequencies outside some intervalls (a, b) and (c, d) of the frequency domain. The Fourier transform of two wavelet atoms $\psi_{2_{s_1,u}}$ and $\psi_{2_{s_2,u}}$ is plotted in Figure 2.2 for scales $s_1 = 0.8$ and $s_2 = 1$ in colors green and red respectively. The size of the intervalls (a, b) and (c, d) depends on s. As s increases the intervalls gets more compact and closer to zero. As s decreases the intervalls get bigger and cover more of the frequency domain. This means that for large s the wavelet transform get contribution from f at low frequencies and for small s at high frequencies. It should be mentioned that by high/low frequencies we mean large/small value of ω in the Fourier transform.

Next we see how the wavelet transform acts as an differential operator on the smoothed signal. Use $\tau = \frac{t-u}{s}$

$$Wf(u,s) = \int_{-\infty}^{\infty} f(t)\frac{1}{\sqrt{s}}(-1)^n \frac{d^n}{d\tau^n}\theta(\tau)\,dt \tag{2.16}$$

$$\frac{d^n}{d\tau^n}\theta(\tau) = \frac{d^n}{du^n}\theta(\tau)\left(\frac{du}{d\tau}\right)^n = (-1)^n s^n \frac{d^n}{du^n}\theta(\tau) \tag{2.17}$$

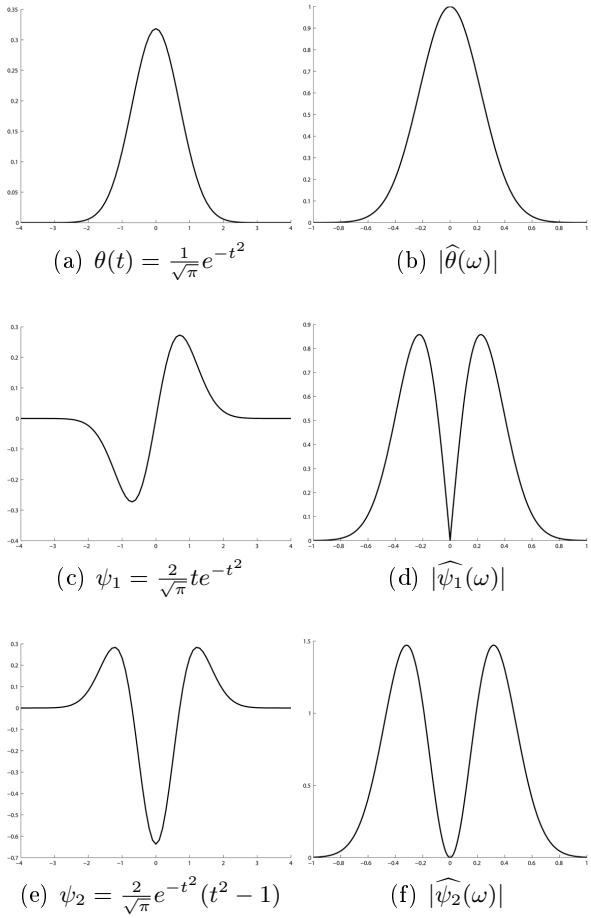
Substitute the right hand side of equation (2.17) into (2.16):

$$\begin{aligned} Wf(u,s) &= \int_{-\infty}^{\infty} f(t)(-1)^n s^{n-\frac{1}{2}} \frac{d^n}{du^n}\theta\left(\frac{t-u}{s}\right)dt \\ &= s^{n-\frac{1}{2}} \frac{d^n}{du^n}\int_{-\infty}^{\infty} f(t)\theta_s(u-t)dt \end{aligned}$$

So, we can write the wavelet transform as the derivative of the convolution with the smoothing function.

$$Wf(u,s) = s^{n-\frac{1}{2}} \frac{d^n}{du^n}(f \star \theta_s)(u) \tag{2.18}$$

Thus, depending on which wavelet we use Wf *is proportional to the first-order or second-order derivative of* $(f \star \theta_s)$.



(a) $\theta(t) = \frac{1}{\sqrt{\pi}} e^{-t^2}$ (b) $|\widehat{\theta}(\omega)|$

(c) $\psi_1 = \frac{2}{\sqrt{\pi}} t e^{-t^2}$ (d) $|\widehat{\psi_1}(\omega)|$

(e) $\psi_2 = \frac{2}{\sqrt{\pi}} e^{-t^2}(t^2 - 1)$ (f) $|\widehat{\psi_2}(\omega)|$

Figure 2.1: Smoothing function θ with corresponding wavelets togheter with their Fourier transforms.

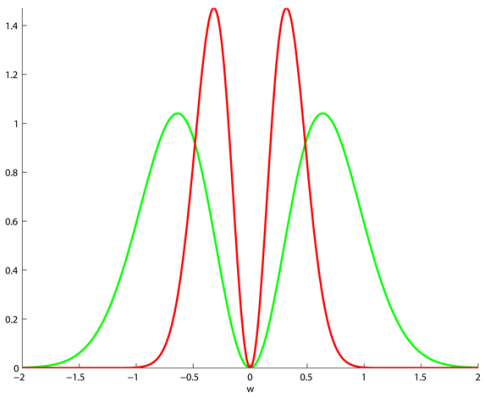


Figure 2.2: The Fourier transform of two wavelet atoms; $\psi_{2_{s_1,u}}$ and $\psi_{2_{s_2,u}}$, for scales $s_1 = 0.8$ and $s_2 = 1$ plotted with colors green and red respectively.

2.1.3 Lipschitz regularity

In this subsection we introduce Lipschitz regularity and see why it makes sense to use Lipschitz regularity as a measure of local regularity of functions.

The regularity of a function f reflects how smooth f is. If $f(t)$ has many derivatives, then it is a smooth signal. Let $f \in C^n(\mathbb{R})$ then the larger n the smoother we say f is. If f is discontinuous at t_0 then it's derivative at t_0 does not exist. This discontinuous appears as large frequency components in $\widehat{f}$. The regularity α of f is related to the decay of $\widehat{f}(\omega)$ and given as the largest α such that

$$\int_{-\infty}^{\infty} |\widehat{f}(\omega)|(1+|\omega|^{\alpha})d\omega < \infty \tag{2.19}$$

If (2.19) holds for some $n \leq \alpha < n+1$ then $\widehat{f}(\omega), (2i\pi\omega)^n\widehat{f}(\omega) \in L^1(\mathbb{R})$ which means that their inverse Fourier transform exists in $L^1(\mathbb{R})$. Assuming $f \in C^n(\mathbb{R}) \cap L^1(\mathbb{R})$ and all the derivatives $f^{(k)}$, $k = 1, \ldots, n$ are in $L^1(\mathbb{R})$ then from Proposition A.4 we know that $\widehat{f^{(n)}}(\omega) = (2i\pi\omega)^n\widehat{f}(\omega)$. Thus if $\alpha = n$ in (2.19) we know that f is n times continuously differentiable. We say that (2.19) gives a global regularity condition over the whole real line, but one can not determine wether the f is locally more regular at a point t_0 from this condition. The Fourier transform is not well adapted to measure local regularity of functions.

As discussed earlier $Wf(u,s)$ is a multiscale transform and for smaller s $|\widehat{\psi_{u,s}}(\omega)| \neq 0$ for larger ω. In addition, for smaller s $\psi(t)$ gets a better localization in time domain. Thus, one might wonder if the decay of $|Wf(u,s)|$ as s decrease can be used to measure local regularity of f. This is the case and later we see how local regularity of f can be measured from the decay of $|Wf(u,s)|$ through s.

To get a precise definition of what we mean by regularity of functions we introduce *Lipschitz regularity*.

Definition 2.10. Let $n \geq 0$ be an integer $\alpha > 0$ and $n \leq \alpha < n+1$. A signal $f(t)$ is said to be *Lipschitz α at the point t_0* if there exists a constant C > 0 and a polynomial $P_{t_0}(t) = \sum_{k=0}^{n} a_k(t-t_0)^k$ of degree n such that for all t in a neigbourhood of t_0

$$|f(t) - P_{t_0}(t)| \leq C|t-t_0|^{\alpha} \tag{2.20}$$

Definition 2.11. A signal $f(t)$ is *uniformly Lipschitz α over (a,b)* if it is Lipschitz α at the point t_0 for every $t_0 \in (a,b)$ and the constant in (2.20)

may be chosen independent of t_0.

Definition 2.12. *The Lipschitz regularity of $f(t)$ at t_0 (or over (a,b))* is the supremum of all α such that $f(t)$ is Lipschitz α at t_0 (or over (a,b)).

One can prove that if f is Lipshictz α over $\mathbb{R}$ then (2.19) is satisfied. Later in this subsection we relate the Lipschitz regularity of f at t_0 to the number of derivatives of f at t_0.

Definition 2.13. A signal f is said to be singular at a point t_0 if there exists an integer $n > 0$ and $\epsilon > 0$ such that f is Lipschitz n at t whenever $t \in \{t_0 - \epsilon, t_0 + \epsilon\} \backslash \{t_0\}$ and f is not Lipschitz n at t_0.

If $f(t)$ is Lipschitz α_0 at t_0 then it is also Lipschitz α at t_0 for all $\alpha < \alpha_0$.

Let us try to be more intuitive on what it means when a function is Lipschitz α. Assume f is singular at t_0 with $0 < \alpha < 1$. then there exists a polynomial of degree zero, $P_{t_0}(t) \equiv C_{t_0}$, such that for some $h > 0$ we have

$$|f(t) - C_{t_0}| \leq C_1 |t - t_0|^\alpha, \ \textit{whenever } t \in (t_0 - h, t_0 + h) \tag{2.21}$$

Using a new variable $\tau = t - t_0$ and defining $g(\tau) = f(\tau + t_0) - C_{t_0}$ we get

$$|g(\tau)| \leq C_1 |\tau|^\alpha \ \textit{whenever } \tau \in (-h, h) \tag{2.22}$$

As we can se in Figure 2.3 $|\tau|^\alpha$ is changing very fast near zero when α is small. Thus $f(t)$ may change very fast near t_0 when α is small.

Let us see what it means when f is Lipscithz α for $\alpha > 1$. First assume there exists a $h > 0$ such that $f(t)$ has exactly $n + 1$ derivatives for all $t \in (t_0 - h, t_0 + h)$. Then we can write the Taylor polynomial of degree n for $f(t)$ about $t = t_0$.

$$P_{t_0}(t) = \sum_{k=0}^{n} \frac{f^{(k)}(t_0)}{k!} (t - t_0)^k$$

Taylor's formula gives an error estimate [5]

$$\begin{aligned} |f(t) - P_{t_0}(t)| &\leq \frac{\sup_{u \in (t_0 - h, t_0 + h)} |f^{(n+1)}(u)|}{(n+1)!} |t - t_0|^{n+1} \\ &= C_{t_0,n} |t - t_0|^{n+1}, \forall t \in (t_0 - h, t_0 + h) \end{aligned}$$

So the function $f(t)$ is at least Lipschitz $n + 1$ at $t = t_0$.

It can be helpful to relate Lipschitz regularity to the number of derivatives. In general Lipschitz regularity tells us how many derivatives a function has

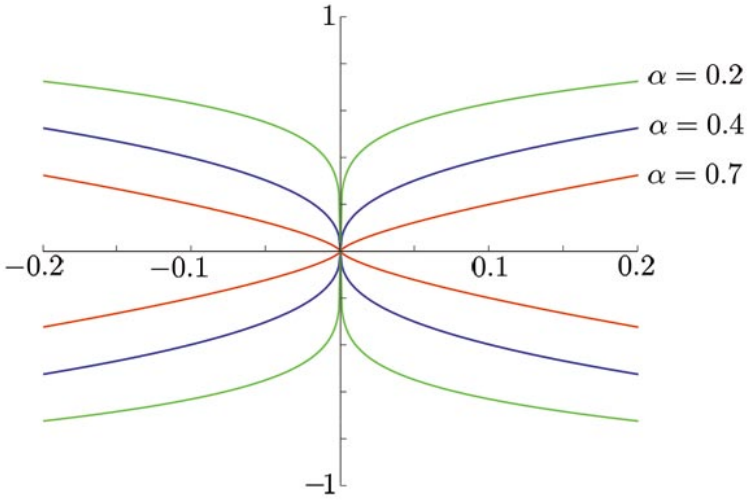


Figure 2.3: $f(\tau) = \pm|\tau|^{\alpha}$ when τ is near zero

at t_0 and what type of singular point the last derivative has at t_0. Denote by $[\alpha]$ the nearest integer of α rounded down and $\{\alpha\} = \alpha - [\alpha]$. Then if $f(t)$ is Lipschitz α at t_0 then $f(t)$ has $[\alpha]$ derivatives at t_0 and $f^{([\alpha])}(t)$ is Lipschitz $\{\alpha\}$ at t_0.

2.1.4 The wavelet transform and Lipschitz regularity

Now we will see how the decay of the wavelet transform modulus through scales is related to the local regularity of the signal being transformed. With some conditions on the wavelet both the uniform Lipschitz regularity and the Lipschitz regularity at a point can be revealed.

Assuming ψ has n vanishing moments and n derivatives having fast decay the following theorem relates the decay of the wavelet transform modulus to the Lipschitz regularity at a point.

Theorem 2.5. *If $f \in \mathrm{L}^2(\mathbb{R})$ is Lipschitz $\alpha \leq n$ at u_0, then there exists a constant A such that for every $(u, s) \in \mathbb{R} \times \mathbb{R}^+$*

$$|Wf(u,s)| \leq A s^{\alpha+\frac{1}{2}} \left(1 + \left|\frac{u-u_0}{s}\right|^{\alpha}\right) \tag{2.23}$$

Conversly, if $\alpha < n$ is not an integer and there exists a constant A and

$\alpha' < \alpha$ such that for every $(u,s) \in \mathbb{R} \times \mathbb{R}^+$

$$|Wf(u,s)| \leq As^{\alpha+\frac{1}{2}} \left(1 + \left|\frac{u-u_0}{s}\right|^{\alpha'}\right) \tag{2.24}$$

then f is Lipschitz α at u_0.
[6, p. 171]

Assuming f is uniformly Lipschitz α on $[a,b]$ the term $\left|\frac{u-u_0}{s}\right|^{\alpha}$ in theorem 2.5 is zero for every $u \in (a,b)$. This is true because for every $u \in [a,b]$ we can choose $u_0 = u$. The next theorem states the same as theorem 2.5 on an intervall $[a,b]$ instead of a point u_0.

Theorem 2.6. *If $f \in \mathrm{L}^2(\mathbb{R})$ is uniformly Lipschitz $\alpha \leq n$ over $[a,b]$, then there exists $A > 0$ such that*

$$|Wf(u,s)| \leq As^{\alpha+\frac{1}{2}}, \forall (u,s) \in [a,b] \times \mathbb{R}^+ \tag{2.25}$$

Conversly, suppose that f is bounded and that Wf(u,s) satisfies (2.25) for an $\alpha < n$ that is not an integer. Then f is uniformly Lipschitz α on (a',b') whenever $a' < a$ and $b' < b$.
[6, p. 169]

If f is Lipschitz α at u_0 then by Theorem 2.5 we know

$$\log|Wf(s,u_0)| \leq \log A + (\alpha + \frac{1}{2}) \log s \tag{2.26}$$

Conversly, if inequality (2.26) is satisfied then f is Lipschitz α at u_0.

Definition 2.14. *The cone of influence of u_0* is every point (u,s) in the scale-space plane such that $|u - u_0| \leq s$.

Let f be Lipschitz $\alpha \leq n$ at u_0 where $n > 0$ is an integer. Then (2.23) states that $|Wf(u,s)| = O(s^{\alpha+\frac{1}{2}})$ inside the cone of influence of u_0. Conversly if $|Wf(u,s)| = O(s^{\alpha+\frac{1}{2}})$ inside the cone of influence of u_0 then (2.24) states that f is Lipschitz $\alpha < n$ at u_0.

The decay of the wavelet transform inside the cone of influence of u characterize the Lipschitz regularity of f at u. If for some $s_0 > 0$ $|Wf(\cdot,s)|$ has one local maximum point inside the cone of influence for $s_0 > s > 0$ then the decay of $|Wf(u,s)|$ inside the cone of influence is bounded above by the decay of $|Wf(u_1(s),s)|$ where $u_1(s)$ is the local maximum point at scale s. The curve $(u_1(s),s)$ in the scale-space plane is called *a maximum*

line. The Lipschitz regularity α of f at t_0 can be measured from the decay of $|Wf(u,s)|$ along the maximum line inside the cone of influence of t_0.

Theorem 2.7. *Let n be a positive integer. Let $\psi(t)$ be a wavelet with compact support, n vanishing moments and be n times continuously differentiable. Let $f(t) \in \mathrm{L}^1([a,b])$.*

- *If there exists a scale $s_0 > 0$ such that for all scales $s < s_0$ and $u \in [a,b]$, $Wf(s,u)$ has no modulus maximum points, then for every $\alpha < n$, $f(t)$ is uniformly Lipschitz α on (a,b)*

[8, section 4]

Theorem 2.7 reassures us that if a function is singular at t_0 with $\alpha < n$ then every neighbourhood of t_0 contains at least one modulus maximum point for small s. This is true because if there exist a neighbourhood of t_0 not containing a modulus maximum point for fine scales, then Theorem 2.7 states that it cannot contain any points where the function is Lipschitz α for a fixed $\alpha < n$. All singular points of f can be located by following the maximum lines as the scale approach zero. However the theorem does not imply that if there exists at least one modulus maximum in every neighbourhood of t_0, then $f(t)$ is singular at t_0, i.e a maximum line does not necessarily point towards a singularity.

Using theory from this section singularities can be located and characterized using the wavelet transform. All singularities are located by following the maximum lines, then characterized by the decay of $|Wf(u,s)|$ through scales. Since $|Wf(u_0,s)|$ decays as $s^{\alpha+\frac{1}{2}}$ when f is Lipschitz α at u_0 then $\alpha = a - \frac{1}{2}$ where a is the slope of $\log|Wf(u_0,s)|$ versus $\log s$.

2.1.5 Lipschitz regularity and edgepoints

We may say that the edges of an images are the set of points having sharp intensity change. In images of physical objects the edges should consist of curves and the points having sharp intensity change should line up accordingly. In 1-D signals we define edge points in terms of Lipschitz regularity.

Definition 2.15. For $f \in L^2(\mathbb{R})$ x_0 is called an *edge point of f* if f is singular at x_0 with $0 \leq \alpha \leq 1$.

Then every edge point of f can be located and characterized by analysing the decay of the wavelet transform through scales.

Let us visualize some constructed edge points. Let the Lipschitz regularity

of f at u_0 be α. We plot f in an image plot letting $f(t)$ correspond to the brightness of the image at position t resulting in an image consisting of one pixel row. Figure 2.4 depicts five such image plots for different α. Each image is drawn using tall but narrow pixels to better see the edges.

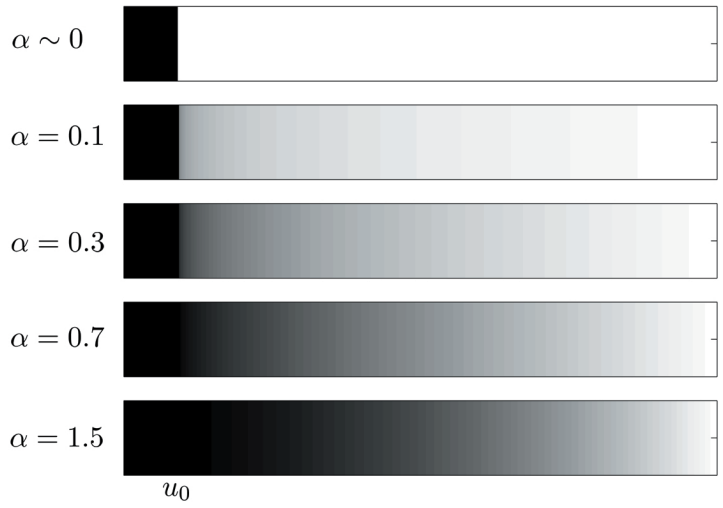


Figure 2.4: Edges in images that are Lipschitz α at u_0.

Figure 2.4 shows that an edge point corresponding to a large α is less visible than an edge corresponding to a small α. When α is small f gives a good representation of a jump at u_0 corresponding to a sharp edge in its image plot.

In chapter 4 we give an edge detector which detectes edge points for each row and column of an image by following maximalines and we calculate The Lipschitz regularity for some edge points.

2.2 2-D analysis

2.2.1 Definitions, notation and properties

In this section we introduce some notations and definitions that are used later in this report.

For any 2-D function $g(x, y)$ write

$$g_s(x, y) = \frac{1}{s^2} g(\frac{-x}{s}, \frac{-y}{s}) \quad \text{and} \quad g_{u,v,s}(x, y) = g_s(u - x, v - y)$$

s is a scaling parameter which dilates the function. If we want different scaling parameter for each direction, then we write

$$g_{(s_x,s_y)}(x,y) = \frac{1}{s_x s_y} g(\frac{-x}{s_x}, \frac{-y}{s_y}) \quad \text{and} \quad g_{u,v,(s_x,s_y)}(x,y) = g_{(s_x,s_y)}(u-x, v-y)$$

The factor $\frac{1}{s^2}$ and $\frac{1}{s_x s_y}$ keeps the energy unchange when dilating the function. The parameters u and v translates the function in the x- and y-direction respectively.

Throughout the text when we talk about 2-D signals we will mean functions from the space $L^2\left(\mathbb{R}^2\right)$ given by the following definition:

Definition 2.16. $L^2\left(\mathbb{R}^2\right)$ is the space of functions

$$f : \mathbb{R}^2 \mapsto \mathbb{C}$$

such that

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |f(x,y)|^2 dxdy < \infty$$

with the inner product and norm

$$< f, g >_2 = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |f(x,y)\overline{g(x,y)}|^2 dxdy$$

$$\|f\|_2 = \left(\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |f(x,y)|^2 dxdy\right)^{\frac{1}{2}}$$

This space contains all 2-D functions with finite energy. Any image can be represented with a function from $L^2\left(\mathbb{R}^2\right)$.

The 2-D Fourier transformation is given in Definition 2.17.

Definition 2.17. Let $f \in L^2\left(\mathbb{R}^2\right)$. Then

$$\hat{f}(\omega_x, \omega_y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y) e^{-2i\pi(x\omega_x + y\omega_y)} dxdy$$

is called the *2-D Fourier transform of f*

Definition 2.18. For $f, g \in L^2\left(\mathbb{R}^2\right)$ the convolution $(f \star g)$ is given by

$$(f \star g)(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(u,v) g(x-u, y-v) dudv$$

Property 2.8.

$$\widehat{(f \star g)}(\omega_x, \omega_y) = \widehat{f}(\omega_x, \omega_y)\widehat{g}(\omega_x, \omega_y) \tag{2.27}$$

Proof.

$$\widehat{(f \star g)}(\omega_x, \omega_y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(u,v)g(x-u,y-v)dudv\right] e^{-2i\pi(x\omega_x+y\omega_y)}dxdy \tag{2.28}$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(u,v)g(\tau_x,\tau_y)e^{-2i\pi((\tau_x+u)\omega_x+(\tau_y+v)\omega_y)}dudvd\tau_x d\tau_y \tag{2.29}$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(u,v)e^{-2i\pi(u\omega_x+v\omega_y)}dudv \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(\tau_x,\tau_y)e^{-2i\pi(\tau_x\omega_x+\tau_y\omega_y)}d\tau_x d\tau_y \tag{2.30}$$

$$= \widehat{f}(\omega_x, \omega_y)\hat{g}(\omega_x, \omega_y) \tag{2.31}$$

□

Property 2.8 say that a convolution in the time-domain corresponds to a product in the Fourier-domain.

Property 2.9. *Let* $f(x,y) \in \mathrm{L}^2\left(\mathbb{R}^2\right) \cap L^1(\mathbb{R}^2)$ *such that* $\frac{\partial^n f(x,y)}{\partial x^n}, \frac{\partial^n f(x,y)}{\partial y^n} \in L^1(\mathbb{R}^2)$ *for* $n = 1, .., N$. *Then for each* $n = 1, ..N$

(i) $\widehat{\frac{\partial^n f}{\partial x^n}}(\omega_x, \omega_y) = (2i\pi\omega_x)^n \hat{f}(\omega_x, \omega_y)$

(ii) $\widehat{\frac{\partial^n f}{\partial y^n}}(\omega_x, \omega_y) = (2i\pi\omega_y)^n \hat{f}(\omega_x, \omega_y)$

Property 2.10. *Let* $\widehat{f}(\omega_x, \omega_y) \in \mathrm{L}^2\left(\mathbb{R}^2\right) \cap L^1(\mathbb{R}^2)$ *such that* $\frac{\partial^n \widehat{f}(\omega_x,\omega_y)}{\partial \omega_x^n}, \frac{\partial^n \widehat{f}(\omega_x,\omega_y)}{\partial \omega_y^n} \in L^1(\mathbb{R}^2)$ *for* $n = 1, .., N$. *Then for each* $n = 1, ..N$

(i) $\frac{\partial^n \widehat{f}}{\partial \omega_x^n}(\omega_x, \omega_y) = [\widehat{(-2i\pi x)^n f}](\omega_x, \omega_y)$

(ii) $\frac{\partial^n \widehat{f}}{\partial \omega_y^n}(\omega_x, \omega_y) = [\widehat{(-2i\pi y)^n f}](\omega_x, \omega_y)$

2.2.2 The wavelet transform

The 2-D wavelet transform use dilated and transleted 2-D wavelets. Since a 2-D function may be dilated both in the x-direction and y-direction the wavelet transform may be defined with two scaling parameters, one for each direction. However for simplicity we define it using one scaling parameter. In Appendix A.4 we look at the wavelet transform using two scaling parameter. In both cases a reconstruction formula is given.

2-D wavelet transform with one scaling parameter

Here we define the 2-D wavelet transform in terms of one scaling parameter.

Definition 2.19. Let $\Psi = \{\psi^k\}_{k=1}^n$ be a set of functions $\psi^k(x,y) \in L^2\left(\mathbb{R}^2\right) \cap L^1(\mathbb{R}^2)$. Ψ is called a 2-D wavelet if there exists a constant $K_\Psi \in \mathbb{R}$ such that

$$0 < K_\Psi = \int_0^\infty \sum_{k=1}^n \frac{|\widehat{\psi^k}(s\omega_x, s\omega_y)|^2}{s} ds < \infty \tag{2.32}$$

In this case (2.32) are called *the 2-D admissibillity condition.* A 2-D wavelet is a set of functions such that their dilations by s togheter cover the Fourier plane and translations by (u_x, u_y) cover $\mathbb{R}^2$. As in one-dimension we will see that this definition of a 2-D wavelet give rise to a reconstruction formula.

Definition 2.20. Let $f(x,y) \in L^2\left(\mathbb{R}^2\right) \cap L^1(\mathbb{R}^2)$ and Ψ be a 2-D wavelet. *The 2-D wavelet tranform of f with respect to* Ψ is the set of functions

$$\mathbf{W}f(u,v,s) = \{W^k f(u,v,s)\}_{k=1}^n \tag{2.33}$$

with

$$\begin{aligned} W^k f(u,v,s)\}_{k=1}^n &= < f, \psi^k_{u,v,s} > \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\frac{1}{s^2}\psi^k(\frac{x-u}{s}, \frac{y-v}{s})dxdy \end{aligned} \tag{2.34}$$

The 2-D wavelet transform of f is a set of inner products between f and $\psi^k_{u,v,s}$. Each $\psi^k_{u,v,s}$ can be constructed to have some desired properties. In Example 2.2 we look at a 2-D wavelet $\Psi(x,y) = \{\psi^1(x,y), \psi^2(x,y)\}$ such that $W^1 f(u,v,s)$ is sensitive to intensity changes in x-direction while $W^2 f(u,v,s)$ is sensive to intensity changes in y-direction. Those properties makes $\Psi(x,y)$ suited for edge detection.

Theorem 2.11 states Parseval's equality in two-dimensions.

Theorem 2.11. *If $f, g \in \mathrm{L}^2\left(\mathbb{R}^2\right)$ then*

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\overline{g(x,y)}dxdy = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \widehat{f}(\omega_x,\omega_y)\overline{\widehat{g}(\omega_x,\omega_y)}d\omega_x d\omega_y$$

From now on we will use the more compact inner product notation $< f, g >$ instead of integral signs. Parseval in two-dimensions gives

$$\mathbf{W}f(u,v,s) = \{< \widehat{f}, \widehat{\psi^k_{u,v,s}} >\}_{k=1}^n$$

If for some (u,v,s) and open set D $\psi^k_{u,v,s}(x,y)$, $k = 1,\ldots,n$ is neglectible for all $(x,y) \notin D$, then $\mathbf{W}f(u,v,s)$ contains information about $f(x,y)$ only when $(x,y) \in D$. The behavior of $f(x,y)$ outside D does not affect $\mathbf{W}f(u,v,s)$.

Also, if for some (u,v,s) and open set B $\widehat{\psi^k_{u,v,s}}(\omega_x,\omega_y)$, $k = 1,\ldots,n$, is neglectible for all $(\omega_x,\omega_y) \notin B$, then $\mathbf{W}f(u,v,s)$ contains information about $\widehat{f}(\omega_x,\omega_y)$ only when $(\omega_x,\omega_y) \in B$. The behavior of $\widehat{f}(\omega_x,\omega_y)$ outside B does not affect $\mathbf{W}f(u,v,s)$.

Next we show that $\{\widehat{\psi^k_{u,v,s}}\}_{s\in\mathbb{R}^+,k=1,\ldots,n}$ covers the Fourier plane. First we notice the following properties:

Property 2.12. *Let $\psi(x,y) \in \mathrm{L}^2\left(\mathbb{R}^2\right) \cap L^1(\mathbb{R}^2)$ then*

(i) $\widehat{\psi_{u,v,s}}(\omega_x,\omega_y) = e^{-2i\pi(u\omega_x+v\omega_y)}\hat{\psi}(s\omega_x, s\omega_y)$

(ii) $\widehat{\psi_s}(\omega_x,\omega_y) = \hat{\psi}(-s\omega_x, -s\omega_y)$

(iii) $\widehat{\psi_{u,v,(s_x,s_y)}}(\omega_x,\omega_y) = e^{-2i\pi(u\omega_x+v\omega_y)}\hat{\psi}(s_x\omega_x, s_y\omega_y)$

(iv) $\widehat{\psi_{(s_x,s_y)}}(\omega_x,\omega_y) = \hat{\psi}(-s_x\omega_x, -s_y\omega_y)$

Proof.

$$\widehat{\psi_{u,v,s}}(\omega_x,\omega_y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{1}{s^2}\psi(\frac{x-u}{s}, \frac{y-v}{s})e^{-2i\pi(x\omega_x+y\omega_y)}dxdy \tag{2.35}$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{1}{s^2}\psi(\tau_x,\tau_y)e^{-2i\pi(\tau_x(s\omega_x)+\tau_y(s\omega_y))}s^2e^{-2i\pi(u\omega_x+v\omega_y)}d\tau_x d\tau_y \tag{2.36}$$

$$= e^{-2i\pi(u\omega_x+v\omega_y)}\hat{\psi}(s\omega_x, s\omega_y) \tag{2.37}$$

The other statements are proved using the same procedure as above. □

From Property 2.12 we know $|\widehat{\psi^k_{u,v,s}}(\omega_x,\omega_y)| = |\widehat{\psi^k}(s\omega_x, s\omega_y)|$. If the Fourier plane is covered by dilations then for each (ω_x,ω_y) there must exist k such that $|\widehat{\psi^k}(s\omega_x, s\omega_y)| > 0$. for some $s > 0$. Thus, from (2.32) any 2-D wavelet covers the Fourier plane by dilations of s.

Proposition 2.13. *Let $f \in \mathrm{L}^2\left(\mathbb{R}^2\right) \cap L^1(\mathbb{R}^2)$, Ψ be a 2-D wavelet and $\mathbf{W}f(u,v,s)$ be the 2-D wavelet transform of f with respect to Ψ. Then*

$$f(x,y) = \frac{1}{K_\Psi}\int_0^\infty \sum_{k=1}^n \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} W^k f(u,v,s)\psi^k_{u,v,s}(x,y)dudv\frac{ds}{s} \tag{2.38}$$

with

$$K_\Psi = \int_0^\infty \sum_{k=1}^{n} |\hat{\psi}^k(s\omega_x, s\omega_y)|^2 \frac{ds}{s} \tag{2.39}$$

Proof. We start with the function

$$J^k(x,y,s) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} W^k f(u,v,s)\psi^k_{u,v,s}(x,y)dudv \tag{2.40}$$

Property 2.12 and Parseval gives

$$\begin{aligned} W^k f(u,v,s) &=< f, \psi^k_{u,v,s} > && (2.41)\\ &=< \hat{f}, \hat{\psi^k}_{u,v,s} > && (2.42)\\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \hat{f}(\omega_x,\omega_y)e^{2i\pi(u\omega_x+v\omega_y)}\overline{\hat{\psi}^k(s\omega_x,s\omega_y)}d\omega_x d\omega_y && (2.43)\\ &= F^{-1}[\hat{f}(\omega_x,\omega_y)\overline{\hat{\psi}^k(s\omega_x,s\omega_y)}](u,v) && (2.44) \end{aligned}$$

The we can write (A.16) as

$$\begin{aligned} J^k(x,y,s) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F^{-1}[\hat{f}(\omega_x,\omega_y)\overline{\hat{\psi}^k(s\omega_x,s\omega_y)}](u,v)\psi_s(u-x,v-y)dudv && (2.45)\\ &=< F^{-1}[\hat{f}(\omega_x,\omega_y)\overline{\hat{\psi}^k(s\omega_x,s\omega_y)}](u,v), \overline{\psi_s(u-x,v-y)} > && (2.46) \end{aligned}$$

Using Parseval and Proposition 2.12 again we get

$$J^k(x,y,s) =< \hat{f}(\omega_x,\omega_y)\overline{\hat{\psi}^k(s\omega_x,s\omega_y)}, e^{-2i\pi(x\omega_x+y\omega_y)}F^{-1}[\overline{\psi(\tau_x,\tau_y)}](s\omega_x,s\omega_y) > \tag{2.47}$$

Assuming $\psi^k(x,y)$ is real we get

$$J^k(x,y,s) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \hat{f}(\omega_x,\omega_y)e^{2i\pi(x\omega_x+y\omega_y)}|\hat{\psi}^k(s\omega_x,s\omega_y)|^2 d\omega_x d\omega_y \tag{2.48}$$

Define the function

$$g(x,y) = \int_0^\infty \sum_{k=1}^{n} J^k(x,y,s)\frac{ds}{s} \tag{2.49}$$

By interchange of integration and sum we get

$$\begin{aligned} g(x,y) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \hat{f}(\omega_x,\omega_y)e^{2i\pi(x\omega_x+y\omega_y)}\int_0^\infty \sum_{k=1}^{n}\frac{|\hat{\psi}^k(s\omega_x,s\omega_y)|^2}{s}dsd\omega_x d\omega_y && (2.50)\\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \hat{f}(\omega_x,\omega_y)e^{2i\pi(x\omega_x+y\omega_y)}K_\Psi(\omega_x,\omega_y)d\omega_x d\omega_y && (2.51)\\ & && (2.52) \end{aligned}$$

From Definition 2.19 we know that $K_\Psi(\omega_x,\omega_y)$ is constant. Then we get

$$f(x,y) = \frac{1}{K_\Psi}g(x,y)$$

□

The following proposition will be used in the next example.

Proposition 2.14. *For any function* $f, g \in \mathrm{L}^2\left(\mathbb{R}^2\right)$ *such that* $g(x,y) = \frac{\partial^n \theta(x,y)}{\partial x^n}$ *then* $< f, g_{u,v,s} >= (-1)^n s^n \frac{\partial^n (f \star \theta_s)(u,v)}{\partial u^n}$

Proof. Let $\tau_x = \frac{x-u}{s}$ and $\tau_y = \frac{y-v}{s}$ then $g(\tau_x, \tau_y) = \frac{\partial^n \theta(\tau_x,\tau_y)}{\partial \tau_x^n}$ First we notice the following

$$\begin{aligned}\frac{\partial^n \theta(\tau_x,\tau_y)}{\partial u^n} &= (\frac{\partial^n \tau_x}{\partial u^n})^n \frac{\partial^n \theta(\tau_x,\tau_y)}{\partial \tau_x^n} \\ &= (-1)^n (\frac{1}{s})^n \frac{\partial^n \theta(\tau_x,\tau_y)}{\partial \tau_x^n}\end{aligned}$$

thus we can write

$$\frac{\partial^n \theta(\tau_x,\tau_y)}{\partial \tau_x^n} = (-1)^n s^n \frac{\partial^n \theta(\tau_x,\tau_y)}{\partial u^n} \tag{2.53}$$

Use (2.53) in the inner product

$$\begin{aligned}< f, g_{u,v,s} > &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\frac{1}{s^2} g(\frac{x-u}{s}, \frac{y-v}{s})dxdy \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\frac{1}{s^2}\frac{\partial^n \theta(\tau_x,\tau_y)}{\partial \tau_x^n}dxdy \\ &= (-1)^n s^n \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\frac{1}{s^2}\frac{\partial^n \theta(\tau_x,\tau_y)}{\partial u^n}dxdy \\ &= (-1)^n s^n \frac{\partial^n \left(\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\theta_s(u-x, v-y)dxdy\right)}{\partial u^n} \\ &= (-1)^n s^n \frac{\partial^n (f \star \theta_s)(u,v)}{\partial u^n}\end{aligned}$$

□

Example 2.2. *Let us look at a 2-D wavelet we will use later. We define the 2-D wavelet* $\Psi = \{\psi^1, \psi^2\}$ *to be the partial derivatives of the normalized 2-D Gaussian* $\theta(x,y) = \frac{1}{\pi} e^{-(x^2+y^2)}$

$$\begin{aligned}\psi^1(x,y) &= \frac{\partial \theta(x,y)}{\partial x} \\ \psi^2(x,y) &= \frac{\partial \theta(x,y)}{\partial y}\end{aligned}$$

In Figure 2.5 ψ^1, ψ^2 *and* θ *are plotted.*

To verify that Ψ *is a 2-D wavelet we calculate* K_Ψ. *First we calculate the Fourier transform of* $\theta(x,y)$ *and then use Property 2.10 to get the Fourier tranform of* ψ^1 *and* ψ^1.

$$\hat{\theta}(\omega_x, \omega_y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{1}{\pi} e^{-(x^2+2i\pi\omega_x x)} e^{-(y^2+2i\pi\omega_y y)} dxdy \tag{2.54}$$

$$= \frac{1}{\pi}\int_{-\infty}^{\infty} e^{-(x^2+2i\pi\omega_x x)} \left[\int_{-\infty}^{\infty} e^{-(y^2+2i\pi\omega_y y)} dy\right] dx \tag{2.55}$$

$$= \frac{1}{\pi}\sqrt{\pi} e^{-(\pi\omega_x)^2} \sqrt{\pi} e^{-(\pi\omega_y)^2} \tag{2.56}$$

$$= e^{-(\pi\omega_x)^2 - (\pi\omega_y)^2} \tag{2.57}$$

By Property 2.10 we get

$$\widehat{\psi^1}(\omega_x, \omega_y) = (2i\pi\omega_x)e^{-(\pi\omega_x)^2-(\pi\omega_y)^2}$$
$$\widehat{\psi^2}(\omega_x, \omega_y) = (2i\pi\omega_y)e^{-(\pi\omega_x)^2-(\pi\omega_y)^2}$$

$$\begin{aligned}
K_\Psi &= \int_0^\infty \sum_{k=1}^{2} \frac{|\widehat{\psi^k}(s\omega_x, s\omega_y)|^2}{s} ds \\
&= \int_0^\infty \sum_{k=1}^{2} \frac{\widehat{\psi^k}(s\omega_x, s\omega_y)\overline{\widehat{\psi^k}(s\omega_x, s\omega_y)}}{s} ds \\
&= \int_0^\infty 4\pi^2(\omega_x^2 + \omega_y^2)s^2 e^{-s^2 2\pi^2(\omega_x^2+\omega_y^2)} ds \\
&= \left[-e^{-s^2(4\pi^2\omega_x^2+4\pi^2\omega_y^2}\right]_0^\infty \\
&= (\lim_{s\to\infty} -e^{-s^2(4\pi^2\omega_x^2+4\pi^2\omega_y^2}) - (-1) \\
&= 1
\end{aligned}$$

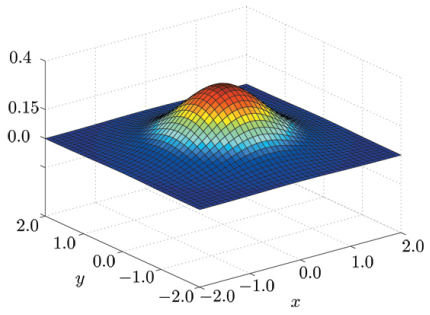
So Ψ is a 2-D wavelet and the reconstruction formula in Proposition 2.13 for the single scale 2-D wavelet transform is valid.

From Proposition 2.14 we see that $\boldsymbol{W}f(u, v, s)$ can be written as the gradient vector of the convolution $(f \star \theta_s)(u, s)$.

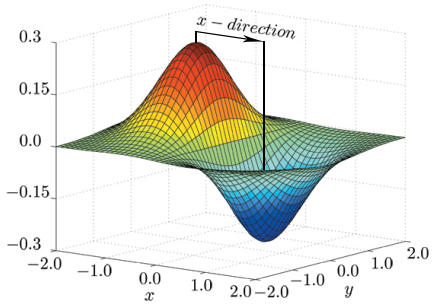
$$\begin{aligned}
\mathbf{W}f(u, v, s) &= \begin{pmatrix} W^1 f(u, v, s) \\ W^2 f(u, v, s) \end{pmatrix} \\
&= \begin{pmatrix} -s\frac{\partial}{\partial u}(f \star \theta_s)(u, v) \\ -s\frac{\partial}{\partial v}(f \star \theta_s)(u, v) \end{pmatrix} \\
&= -s\nabla(f \star \theta_s)(u, v)
\end{aligned}$$

The 2-D wavelet transform is proportional to the partial derivatives of the smoothed signal, and the parameter s dilates θ.

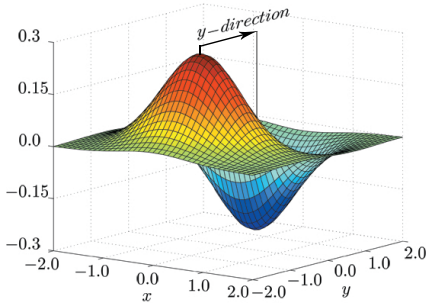
One difference between the 2-D and 1-D wavelet transform is that $\mathbf{W}f(u, s)$ has both magnitude $|\mathbf{W}f(u, s)|$ and direction $Af(u, v, s) = Arg\{\mathbf{W}f(u, s)\}$ with $0 \leq Af(u, v, s) < 2\pi$. So, $Af(u, v, s)$ gives the direction where the smoothed signal changes the most.



(a) $\theta(x,y) = \frac{1}{\pi}e^{-(x^2+y^2)}$



(b) $\psi^1(x,y) = \frac{\partial\theta(x,y)}{\partial x}$



(c) $\psi^2(x,y) = \frac{\partial\theta(x,y)}{\partial y}$

Figure 2.5: θ, the normalized 2-D Gaussian, and it's corresponding mother wavelets

2.2.3 Lipschitz regularity

In this section we introduce Lipschitz regularity in $\mathbb{R}^2$. We will se the connection between Lipschitz regularity and edges.

Definition 2.21. Lipschitz at a point Let $0 \leq \alpha \leq 1$. A function $f(x,y) \in L^2\left(\mathbb{R}^2\right)$ is said to be *Lipschitz α at a point* (x_0, y_0) if there exists a $K > 0$ such that for all $(x,y) \in \mathbb{R}^2$

$$|f(x,y) - f(x_0,y_0)| \leq K(|x-x_0|^2 + |y-y_0|^2)^{\frac{\alpha}{2}} \tag{2.58}$$

[6]

Lipschitz over an open set A function is said to be Lipschitz α over an open set $D \in \mathbb{R}^2$ if it is Lipschitz α at (x_0, y_0) for every $(x_0, y_0) \in D$.

Lipschitz regularity *The Lipschitz regularity of $f(x,y)$ at (x_0,y_0)* (or over an open set D) is the supremum of all α such that $f(x,y)$ is Lipscithz α at (x_0,y_0) (or over an open set D).

Lipschitz regularity of $f(x, y)$ at (x_0, y_0) is an estimate on how fast a function change in a neigbourhood of (x_0, y_0). If $f(x, y)$ is discontinuous at (x_0, y_0) in the x-direction, but perfectly smooth in the y-direction then the Lipschitz regularity of f at (x_0, y_0) is zero. One might say that the worst estimate is used.

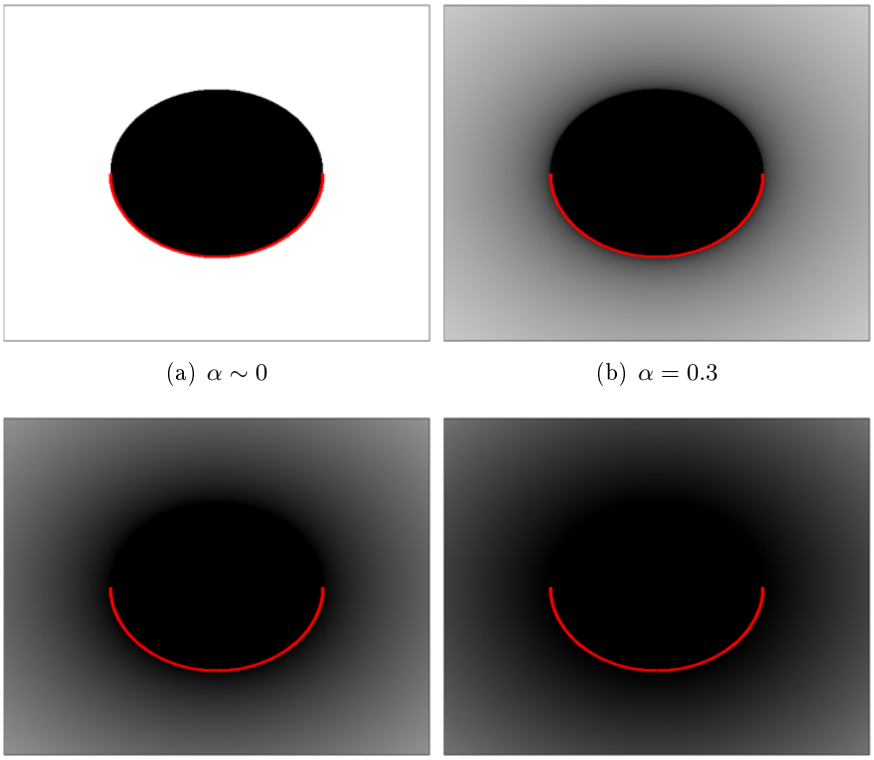
In one-dimension we saw how Lipschitz regularity was characterised by the wavelet transform. It is possible to extend Theorem 2.5 to be valid also in the two-dimensional case, i.e Lipschitz regularity for a function $f \in L^2\left(\mathbb{R}^2\right)$ may be characterised from the decay of the 2-D wavelet transform of f. However, the proof will require much work and we will not use the result in this report. Therefore we do not elaborate more on this subject.

2.2.4 Lipschitz regularity and edges

To show the connection between Lipschitz regularity and edges we construct some edges corresponding to different singular points. Let the Lipschitz regularity of f at (u, v) be α for every (u, v) on the circle $\sqrt{u^2 + v^2} = r_0$.

We plot f in an image plot letting $f(x, y)$ correspond to the brightness of the image at position (x, y). Figure 2.6 depicts four such image plots for different α. The bottom half of the circle with radius r_0 is drawn so the reader more easily can see where the singular points are located.

Figure 2.6 shows that an edge corresponding to large α is less visible than an edge corresponding to small α. When α is small f gives a good representation of a jump at (u, v) corresponding to a sharp edge in its image plot.



(a) $\alpha \sim 0$ (b) $\alpha = 0.3$

(c) $\alpha = 0.7$ (d) $\alpha \sim 1$

Figure 2.6: Images where all points on a circle are Lipschitz α. The bottom half of the circle is marked red.

Chapter 3

Survey

3.1 Introduction

This chapter is a survey of some articles concerning reconstruction of signals from samples of the wavelet transform at local extremum positions. We also see how the edge detector given by Canny in [3] might be expressed using the wavelet transform.

The survey includes the following articles: [1, p. 83-94] [7] [4] [9].

3.2 Multiscale framework

3.2.1 Multiscale edge detection

Mallat and Zhong shows how the wavelet transform is a multiscale differential operator.[7] For wavelets defined as the first and second order derivative of a smoothing function $\theta(x)$ it is shown that the wavelet transform of f is the first and second order derivative of $(f \star \theta_s)(u)$ respectively. The definition for a smoothing function is given in Definition 2.8.

$$W^a f(u,s) = -s\frac{d}{dx}(f \star \theta_s)(u) \tag{3.1}$$

$$W^b f(u,s) = s^2\frac{d^2}{dx^2}(f \star \theta_s)(u) \tag{3.2}$$

The local extrema of $W^a f(u,s)$ correspond to the zero crossings of $W^b f(u,s)$ and to inflection points of $(f \star \theta_s)(u)$.

J. Canny derived a optimal detector optimizing a localization and detection criteria. In 1-D the optimal detector have an approximate implementation in which edges are marked at local maximum points of the first derivative of a Gaussian convolved with f.[3] This particular case correspond to locating maxima of $W^a f(u,s)$ when $\theta(x)$ is Gaussian.

Detecting zero crossings of $W^b f(u,s)$ or extrema of $W^a f(u,s)$ are similar procedures, but the extrema approach have some advantages. With the extrema approach we can distinguish between minima and maxima. Inflection points of $(f \star \theta_s)(u)$ corresponding to minima of $|W^a f(u,s)|$ are slow variation points, whereas the maxima correspond to sharp variation points. Thus we select all sharp variation points by detecting maxima points of $|W^a f(u,s)|$. Also, the values of $|W^a f(u,s)|$ at each maxima characterize Lipschitz regularity at the inflection points. The zero crossing approach only gives position information.

J. Canny gave a 2-D approach where the image I is convolved with a symmetric 2-D Gaussian G and the zeros of the directional second order derivative $\frac{\partial^2}{\partial \mathbf{n}^2}(G \star I)$ is marked as edges. The direction is given by $\mathbf{n} = \frac{\nabla(G \star I)}{|\nabla(G \star I)|}$. A similar approach is given in terms of wavelets by Mallat and Zhong [7]. They define two wavelets as the partial derivatives of a 2-D smoothing function.

$$\psi^1(x,y) = -\frac{\partial \theta(x,y)}{\partial x} \text{ and } \psi^2(x,y) = -\frac{\partial \theta(x,y)}{\partial y}$$

The 2-D wavelet transform has two components

$$W_s^1 f(u,v) = (f \star \psi_s^1)(u,v) \text{ and } W_s^2 f(u,v) = (f \star \psi_s^2)(u,v)$$

One can show that

$$\mathbf{W_s} f(u,v) = \begin{pmatrix} W_s^1 f(u,v) \\ W_s^2 f(u,v) \end{pmatrix} = \begin{pmatrix} -s\frac{\partial}{\partial u}(f \star \theta_s)(u,v) \\ -s\frac{\partial}{\partial v}(f \star \theta_s)(u,v) \end{pmatrix} = s\nabla(f \star \theta_s)(u,v)$$

All maxima points of $|\mathbf{W_s} f|$ in the direction which $\mathbf{W_s} f$ points are marked as edges.

These two approches differs slightly. For each point (u,v) $\mathbf{W_s} f(u,v)$ gives a weighted sum of the values at some neighbouring points from the image. For the method given by J. Canny this sum will be a difference between local averages on different sides of the edge. For the latter method the sum will also be affected by differences in other directions than normal to the edge. The average parallell to the edge do not contribute to the localization of the edge.

3.2.2 Analysis of the multiscale information

As we have seen in Chapter 2 the evolution across scales of the wavelet transform characterize the local Lipschitz regularity of the signal and Lipshictz regularity were defined for functions. Mallat and Zhong extended Lipschitz regularity to distribution by saying that a distribution has a uniform Lipschitz regularity equal to α on (a, b) if and only if its primitive has a uniform Lipschitz regularity equal to $\alpha + 1$ on (a, b). For example, Dirac's delta function centered at t_0 is uniformly Lipschitz -1 in a neigbourhood of t_0 since it's primitive, a step edge, has uniformly Lipschitz regularity 0 in a neighbourhood of t_0.

It has been proved that if a signal is singular at a point t_0, there exists a sequence of wavelet transform modulus maxima that converge to t_0 when the scale decreases. Hence, we detect all the singularities from the positions of the wavelet tranform modulus maxima. Moreover, the decay of the wavelet transform is bounded by the decay of these modulus maxima, and we can thus measure the local uniform Lipschitz regularity from this decay.

Smooth variation at t_0 is modelled by a singularity smoothed with a Gaussian of variance σ^2. If the smoothing function is close to a Gaussian, then the wavelet transform at the scale s of the smoothed singularity is equal to the wavelet transform of the nonsmoothed singularity at the scale $s_0 = \sqrt{s^2 + \sigma^2}$. It is shown

$$|W_s f(x)| \leq K s_0^{\alpha - 1} \text{ with } s_0 = \sqrt{s^2 + \sigma^2} \tag{3.3}$$

If the signal is multiplied with a constant, then α_0 and σ are not affected. If the signal is smoothed by a Guassian, then α_0 is not affected, but σ increases.

3.2.3 The dyadic wavelet transform

To allow fast numerical implementations, the scales are choosen only along the dyadic sequence $\{2^j\}_{j \in \mathbb{Z}}$.

Definition 3.1. Let $\psi(x)$ be a wavelet, $f \in L^2(\mathbb{R})$ and $\mathbf{W}f$ the sequence of functions $\{W_{2^j} f(x)\}_{j \in \mathbb{Z}}$ given by

$$W_{2^j} f(x) = (f \star \psi_{2^j})(x) \tag{3.4}$$

Then $\mathbf{W}$ is called the dyadic wavelet transform operator and $\mathbf{W}f$ the *the dyadic wavelet transform of* f.

Mallat and Zong proved that if the whole frequency axis is covered by dilations; $\{2^j\}_{j \in \mathbb{Z}}$, of $\widehat{\psi}(\omega)$ then the dyadic wavelet transform is complete and

stable, i.e for any function $f \in L^2(\mathbb{R})$ $\mathbf{W}f$ characterize f uniquely and f may be reconstructed from $\mathbf{W}f$. The reconstruction wavelet $\chi(x)$ is any function whose Fourier transform satisfies

$$\sum_{j=-\infty}^{\infty} \widehat{\psi}(2^j\omega)\widehat{\chi}(2^j\omega) = 1 \tag{3.5}$$

Denote by $\mathbf{W}^{-1}$ the operator defined by

$$\mathbf{W}^{-1}\{g_j(x)\}_{j\in\mathbb{Z}}(u) = \sum_{j=-\infty}^{\infty} (g_j \star \chi_{2^j})(u) \tag{3.6}$$

Then $\mathbf{W}^{-1}$ recover the function $f(x)$ from $\mathbf{W}f$

$$f(x) = \mathbf{W}^{-1}\{W_{2^j}f\}_{j\in\mathbb{Z}} = \sum_{j=-\infty}^{\infty} (W_{2^j}f \star \chi_{2^j})(x) \tag{3.7}$$

3.2.4 discussion

The wavelet transform can be expressed as a differential operator and used as an edge detector by locating it's local extrema or zero-crossings. Also the dyadic wavelet transform is a complete and stable representation of any function $f \in L^2(\mathbb{R})$. In the next section we see how sampling of the dyadic wavelet transform at local extremum points or zero-crossings can be used to reconstruct an approximation of f.

3.3 Reconstruction from multiscale edges

Any function $f \in L^2(\mathbb{R})$ can be reconstructed from it's dyadic wavelet transform. However $\mathbf{W}f$ contains "to much information" and one would like to minimize the amount of information necessary to have a stable and complete representation of f.

In [7, sec. 5] Mallat and Zhong introduce an algorithm to reconstruct $f \in L^2(\mathbb{R})$ from the extrema of $\mathbf{W}f$. More accurate, the algorithm reconstructs an approximation of $\mathbf{W}f$ from the modulus maxima of $W_{2^j}f(x)$ from each scale j via an alternate projection iteration procedure. Then $\mathbf{W}^{-1}$ is used on the approximated dyadic wavelet transform to reconstruct an approximation of f. An alternative algorithm reconstructing f directly is given by Carmona in [1, Page 83]. Both algorithms are based on continuous-time theory while implementation takes place in the discrete-time domain.

With implementation in mind Cvetkovic and Vetterli; in [9], introduced two reconstruction algorithms based on discrete-time theory. Their reconstruction algorithms use alternate projection iteration procedure in $\ell^2(\mathbb{Z})$, one based on the wavelet zero crossings representation and the other based on the wavelet extrema representation. In this section we take a look at the above mentioned algorithms starting with the one introduced by Mallat and Zhong.

For any representation of f *the reconstruction set of* f denotes the set of functions having the same representation as f.

3.3.1 The wavelet modulus maxima representation

The decay of the wavelet transform modulus maxima through scales characterize Lipschitz regularity and thus the modulus maxima contains important information about the signal. Mallat and Zong introduced a signal representation using the position and values of the local modulus maxima of the dyadic wavelet transform of f.

Let $f \in L^2(\mathbb{R})$ and $\{x_n^j\}_{n\in\mathbb{Z}}$ be the abscissas where $|W_{2^j}f(x)|$ is locally maximum. Denote by $\mathbb{H}$ the reconstruction set of f consisting of functions from $L^2(\mathbb{R})$ having the same modulus maxima representation as f, i.e for each j and $h \in L^2(\mathbb{R}) \cap \mathbb{H}$

1. $W_{2^j}h(x_n^j) = W_{2^j}f(x_n^j)$
2. The local maxima of $|W_{2^j}h(x)|$ is located at the abscissa $\{x_n^j\}_{n\in\mathbb{Z}}$

In general $\mathbb{H} \setminus \{f\} \neq \emptyset$, i.e the wavelet modulus maxima representation does not characterize f uniquely.

Let $\mathbb{K}$ be the space of all sequences of functions $\{g_j\}_{j\in\mathbb{Z}}$ such that

$$|\{g_j\}_{j\in\mathbb{Z}}|^2 = \sum_{j=-\infty}^{\infty} \left(\|g_j\|^2 + 2^{2^j} \|\frac{dg_j}{dx}\|^2 \right) < \infty \tag{3.8}$$

Then $\mathbb{K}$ together with the above norm $|\cdot|$ is a Hilbert space. In Appendix A.1 we list some definitions for the spaces we talk about in this section.

It is shown that if there exists two constants $A_2 > 0$ and B_2 such that for all $\omega \in \mathbb{R}$

$$A_2 \leq \sum_{j=-\infty}^{\infty} |\widehat{\psi}(2^j\omega)|^2 + \sum_{j=-\infty}^{\infty} |\widehat{\frac{\psi(x)}{dx}}(2^j\omega)|^2 \leq B_2 \tag{3.9}$$

Then for any $h \in L^2(\mathbb{R})$ with the norm $|\cdot|$ in (3.8)

$$A_2\|h\|^2 \leq |\mathbf{W}h|^2 \leq B_2\|h\|^2 \tag{3.10}$$

As in last section $\mathbf{W}$ denotes the dyadic wavelet transform operator. Not every sequence from $\mathbb{K}$ is the dyadic wavelet transform of some function from $L^2(\mathbb{R})$. Denote by $\mathbb{V}$ the space of all dyadic wavelet transforms of functions from $L^2(\mathbb{R})$. From (3.10) we know that $\mathbb{V}$ is a subset of $\mathbb{K}$, $\mathbb{V} \subseteq \mathbb{K}$.

Let Γ be all sequences of functions $\{g_j\}_{j\in\mathbb{Z}} \in \mathbb{K}$ such that

$$g_j(x_n^j) = W_{2^j} f(x_n^j) \tag{3.11}$$

Then the dyadic wavelet transforms that satisfies Condition 1 are the sequence of functions that belong to

$$\Lambda = \mathbb{V} \cap \Gamma \tag{3.12}$$

The wavelet modulus maxima representation does not contain all extrema of $\mathbf{W}f$ and Condition 2 is approximated by the problem of finding $\mathbf{W}h \in \Lambda$ such that $|\mathbf{W}h|^2$ is minimum. This minimazation generally creates local modulus maxima at the positions $\{x_n^j\}_{n\in\mathbb{Z}}$. The second term; $2^{2^j}\|\frac{dh_j}{dx}\|^2$, is minimized to have as few local modulus maxima possible outside the abscissas $\{x_n^j\}_{n\in\mathbb{Z}}$.

Since $\mathbb{K}$ is a Hilbert space, $\mathbb{V} \subseteq \mathbb{K}$ and Γ an affine space the orthogonal projection of $\{g_j\}_{j\in\mathbb{Z}} \in \mathbb{K}$ on Λ is given by iterating alternate projections on $\mathbb{V}$ and Γ.

$$\mathbf{PR}_\Lambda \{g_j\}_{j\in\mathbb{Z}} = \lim_{n\to\infty} (\mathbf{PR}_\mathbb{V} \circ \mathbf{PR}_\Gamma)^n \{g_j\}_{j\in\mathbb{Z}} \tag{3.13}$$

If we start with the zero element of $\mathbb{K}$ the alternate projections converge to the element of Λ which is closest to zero, and thus whose norm $|\cdot|$ is minimum.

The orthogonal projector operator from $\mathbb{K}$ on $\mathbb{V}$ is given by

$$\mathbf{PR}_\mathbb{V} = \mathbf{W} \circ \mathbf{W}^{-1} \tag{3.14}$$

With $\mathbf{W}$ as in Definition 3.1 and $\mathbf{W}^{-1}$ as (3.6). It may be proved that $\mathbf{PR}_\mathbb{V}$ is an orthogonal projection if and only if the wavelet is symmetrical or antisymmetrical.

The orthogonal projector operator on Γ transforms any sequence $\{g_j(x)\}_{j\in\mathbb{Z}} \in \mathbb{K}$ into the closest sequence $\{h_j(x)\}_{j\in\mathbb{Z}} \in \Gamma$ with respect to the norm $|\cdot|$. This is done by chosing $\{h_j(x)\}_{j\in\mathbb{Z}}$ such that the sequence of error functions

$$\{\epsilon_j(x)\}_{j\in\mathbb{Z}} = \{h_j(x) - g_j(x)\}_{j\in\mathbb{Z}} \tag{3.15}$$

is minimum with respect to the norm $|\cdot|$.

Let x_i^j and x_{i+1}^j be two consecutive modulus maxima of $W_{2^j}f(x)$. It is shown that for $x \in (x_i^j, x_{i+1}^j)$ the solution of this minimization is

$$\epsilon_j(x) = \alpha_j^i e^{2^{-j}x} + \beta_j^i e^{-2^{-j}x} \tag{3.16}$$

Since $\{h_j(x)\}_{j\in\mathbb{Z}} \in \Gamma$ we know

$$h_j(x_i^j) = W_{2^j}f(x_i^j) \tag{3.17}$$
$$h_j(x_{i+1}^j) = W_{2^j}f(x_{i+1}^j) \tag{3.18}$$

thus the constants α_j^i and β_j^i are calculated using

$$\epsilon_j(x_i^j) = W_{2^j}f(x_i^j) - g_j(x_i^j) \tag{3.19}$$
$$\epsilon_j(x_{i+1}^j) = W_{2^j}f(x_{i+1}^j) - g_j(x_{i+1}^j) \tag{3.20}$$

The convergence of the alternate projection may be very slow and in the case where $\{\sqrt{2^j}\psi_{2^j}(x_n^j - x)\}_{(n,j)\in\mathbb{Z}^2}$ is a frame of $\mathbb{U}$ the stability depends on the value of the frame constants. Here $\mathbb{U} \subseteq L^2(\mathbb{R})$ denote the space of functions that are linear combinations of functions in the family $\{\psi_{2^j}(x_n^j - x)\}_{(j,n)\in\mathbb{Z}^2}$.

3.3.2 The discrete wavelet transform representation

In [9] Cvetkovic and Vetterli look at the zero crossings and extrema representation in $\ell^2(\mathbb{Z})$. They cover wavelet design, reconstruction and implementation. In this section we look at the two latter subjects.

The wavelet transform is defined as a linear operator

$$\mathbf{W} : \ell^2(\mathbb{Z}) \to \ell^2(\mathbf{I}) \text{ with } \mathbf{I} = \{1, 2, \ldots, J+1\} \times \mathbb{Z} \tag{3.21}$$

consisting of $J+1$ linear operators $W_j : \ell^2(\mathbb{Z}) \to \ell^2(\mathbb{Z})$, $j = 1, 2, \ldots, J+1$. Denote by $F = \mathbf{W}f$ and $F^j = W_j f$ the j'th column vector of F, $j = 1, 2, \ldots, J+1$.

The discrete wavelet transform zero crossing representation

The wavelet zero crossings representation of f is defined as:

$$E_z f = \{ZW_j f, SW_j f, j = 1, 2, \ldots, J+1\} \tag{3.22}$$

with

$$Zf = \{k : f(k)f(k-1) \leq 0\} \tag{3.23}$$

$$Sf = \{Sf(k) : Sf(k) = \sum_{j=z_{k-1}}^{z_k-1} f(j), k = 1, \ldots, |Zf| + 1\} \tag{3.24}$$

Here $|Zf|$ denote the total number of zero crossings of f, and z_k its kth zero crossing. Sf provides the sum of points between all pairs of consecutive zero crossings and is necessary to improve the stabillity of the representation. A large familiy of functions with the same zero crossings was found by Meyer. Adding Sf to the representation will not, in general, give a unique characterisation of f. However, it then seems like two functions having the same representation differ only at high frequency.

The discrete wavelet transform extrema representation

The discrete wavelet extrema representation of a signal $f \in \ell^2(\mathbb{Z})$ is define as:

$$E_e f = \{M_a W_j f, M_i W_j f, M W_j f, j = 1, 2, \ldots, J+1\} \tag{3.25}$$

with

$$M_a f = \{k : f(k+1) \leq f(k), f(k-1) \leq f(k)\} \tag{3.26}$$

$$M_i f = \{k : f(k+1) \geq f(k), f(k-1) \geq f(k)\} \tag{3.27}$$

$$Mf = \{f(k) : k \in M_i f \cup M_a f\} \tag{3.28}$$

Here Mf plays the role Sf does in $E_s f$. Actually there is a relation between these two representations

$$E_e f = E_z \Delta f \tag{3.29}$$

where Δf is the first difference of f,

$$\Delta f(n) = f(n+1) - f(n) \tag{3.30}$$

In the case where the wavelet is the first derivative of some smoothing function the modulus maxima correspond to sharp variation points in the signal while the modulus minima correspond to slow variation points. When the wavelet is the second order derivative a zero crossing corresponds either to a sharp variation point or a slow variation point of f. Using $E_e f$ we can distinguish between these two types of points, this is not possible from $E_z f$. In addition, furhter characterization of the variation points of f is possible from Mf. Therefore, for further analysis of f, the wavelet extrema representation is prefered.

As the reconstruction algorithm given by Mallat and Zhong, the following algorithm also recovers, from the representation of f, the wavelet transform $\mathbf{W}h$ for some h in the reconstruction set of f. Then h is reconstructed using the inverse wavelet transform.

The algorithm use alternating projections on the sets $\mathcal{V}$, $\mathcal{E}$ and $(\cap_{i,j}\mathcal{C}_{i,j})$ starting with any initial point $F_0 \in \ell^2(\mathbf{I})$. With

- $\mathcal{V}$ is the range of the wavelet transform;

$$\mathcal{V} = \{G : G = \mathbf{W}g, g \in \ell^2(\mathbb{Z})\} \tag{3.31}$$

- $\mathcal{E}$ is the set of all $G \in \ell^2(\mathbf{I})$ having the same value as MW_jf at all local extrema of W_jf across all scales j:

$$\mathcal{E} = \{G \in \ell^2(\mathbf{I}) : G^j(k) = W_jf(k) \forall k \in M_iW_jf \cup M_aW_jf, j = 1, 2, \ldots, J{+}1\} \tag{3.32}$$

- $\mathcal{C}_{i,j}$ is the set of all $G \in \mathcal{V} \cap \mathcal{E}$ such that $G^j(i)$ is nonincreasing/nondecreasing if $F^j(i)$ is decreasing/increasing.

The intersection $\mathcal{V} \cap \mathcal{E}$ contains all $G \in \ell^2(\mathbf{I})$ such that G is the wavelet transform of some $g \in \ell^2(\mathbf{Z})$ and its values matches with the values of $\mathbf{W}f$ at each extrema point of $\mathbf{W}f$. The intersection with the set $\mathcal{C}_{i,j}$ makes sure these matching points are the only extremum points of G.

The projections; $G_\mathcal{V} \in \mathcal{V}$, $G_\mathcal{E} \in \mathcal{E}$ and $G_{\mathcal{C}_{i,j}} \in \mathcal{C}_{i,j}$, of $G \in \ell^2(\mathbf{I})$ on $\mathcal{V}$, $\mathcal{E}$ and $\mathcal{C}_{i,j}$ respectively are given by

- $$G_\mathcal{V} = \mathbf{W}\mathbf{W}^{-1}G \tag{3.33}$$

- $$G^j_\mathcal{E}(k) = \begin{cases} F^j(k) & k \quad \text{is an extremum of} \quad F^i \\ G^j(k) & otherwise \end{cases} \tag{3.34}$$

- $$G^j_\mathcal{E}(k) = \begin{cases} F^j(k) & k \quad \text{is an extremum of} \quad F^i \\ G^j(k) & otherwise \end{cases} \tag{3.35}$$

3.3.3 discussion

We have seen how an approximation of a signal may be reconstructed from the extremum points or zero-crossings of it's wavelet transform. Really this is an irregular sampling of the wavelet transform where the set of samples

contains most of the information of the signal. Therefore the wavelet transform is suitable for compression of signals. Mallat and Zhong gives examples of image reconstruction from multiscale edges with a compression ratio over 30 [7]. The recovered images have lost some small details, but is visually of good quality.

Chapter 4

Edge detectors

4.1 Introduction

In this chapter we look into and implement some edge detectors to see how well they function on the images shown in Figure 4.1 refered to as I_1, I_2, I_3, I_4, I_5 and I_6 throughout this chapter.

The images I_1, I_2 and I_3 are constructed images where I_1 has no noise, I_2 is after addition of white noise and I_3 is after addition of simulated speckle. The edges we would like to detect are the contour round the bright areas in I_1.

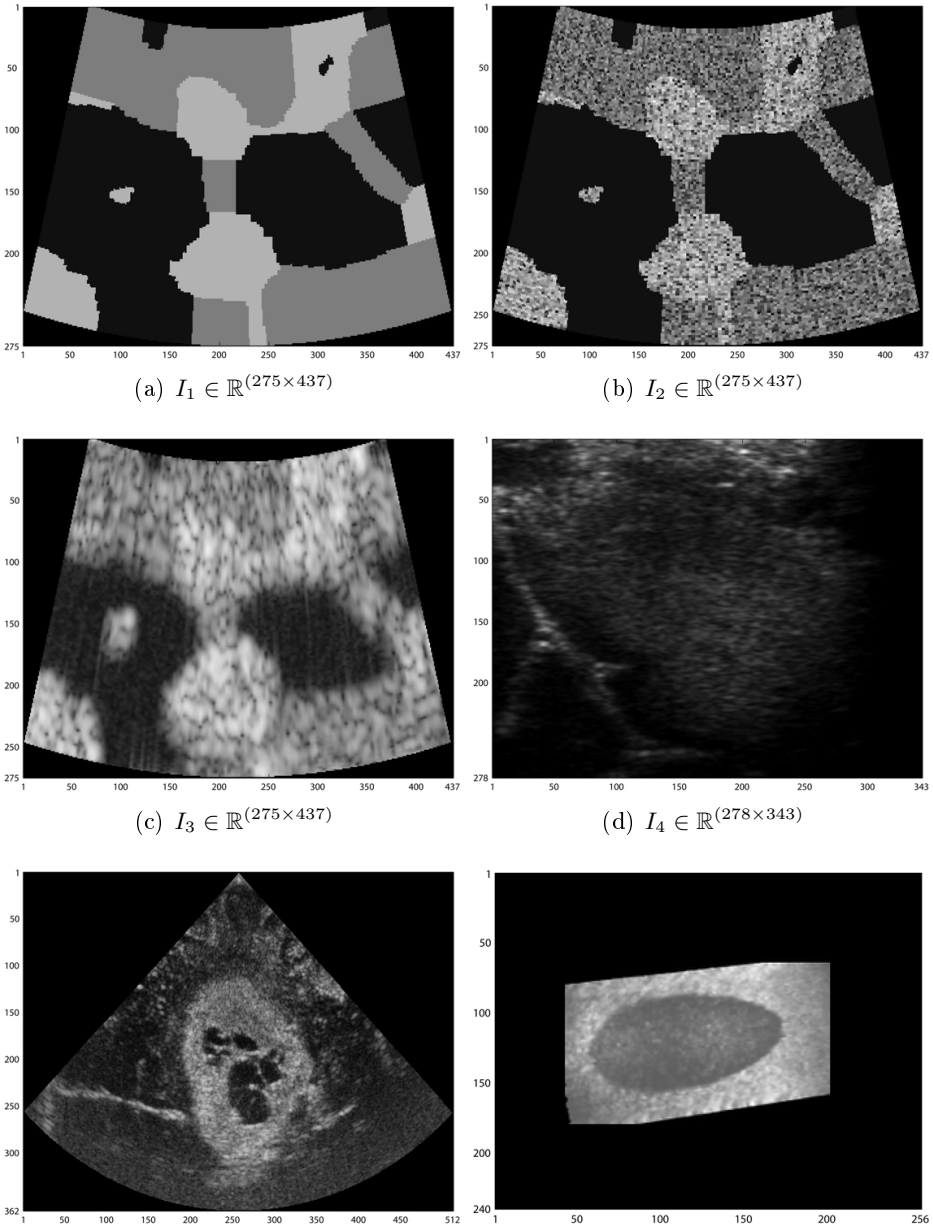
The image I_4 is an ultrasound image of a brain tumor. It is not easy to determine the contour of the tumor. The image I_5 is also an ultrasound image of a brain tumor, but in this case the contrast between the tumor and brain-cells are high. Image I_6 shows some object shaped as an egg and is included to have a variety of images to test our edge detectors for.

In Chapter 2 we looked at some theory which we would like to use in some edge detection algorithms. The introduced theory deals with continuous functions, but our algorithms are implemented on a computer. When we analyse a function f on a computer the value of f is only know for a discrete set of points $\{t_k\}_{k\in\mathbb{Z}}$ and the integral in Wf must be approximated. Since the Lipschitz regularity of f is characterized from the decay of the wavelet transform as the scale approaches zero we cannot detect Lipschitz regularity of f based on it's samples. It might seem that f is discontinuous even though it is continuous having a very sharp transition between two samples.

In practice we say that f behaves as if it is Lipschitz α at k up to its sample resolution. When we talk about discontinuity and Lipschitz regularity of samples of signals we mean that the samples behave as if the signal is

discontinuous an so on.

Denote by $\mathbb{R}^{(m\times n)}$ all matrices having m rows and n columns with elements from $\mathbb{R}$. Let $I \in \mathbb{R}^{(m\times n)}$ and denote by $I(i,j)$ the element from the i'th column and j'th row of I. Denote by $\mathcal{C} = \{I_{i\cdot}\}_{i=1}^{n}$ the family of sequences corresponding to the columns of I. Denote by $\mathcal{R} = \{I_{\cdot j}\}_{j=1}^{m}$ the family of sequences corresponding to the rows of I. All images we analyse in this chapter are greyscale images; this means that they are represented by some $I \in \mathbb{R}^{(m\times n)}$ such that the value $I(i,j)$ gives the brigthness at position (i,j) in the image. The values zero and one correspond to black and white respectively. When we use the word pixel we mean some element $I(i,j)$.



(a) $I_1 \in \mathbb{R}^{(275\times437)}$ (b) $I_2 \in \mathbb{R}^{(275\times437)}$

(c) $I_3 \in \mathbb{R}^{(275\times437)}$ (d) $I_4 \in \mathbb{R}^{(278\times343)}$

(e) $I_5 \in \mathbb{R}^{(362\times512)}$ (f) $I_6 \in \mathbb{R}^{(240\times256)}$

Figure 4.1: The six greyscale images we will use throughout this chapter.

From now $WI(i, j, s_k)$ (or $WI(i, s_k)$) with $I \in \mathbb{R}^{(m \times n)}$ means an approximation of $Wf(x, y, s_k)$ (or $Wf(x, s_k)$) where I is a sampling of $f(x, y)$ (or $f(x)$) at points (i, j) (or i). The scale $s_k \geq 1$ are integers deciding the support of the wavelet atom used to calculate $WI(i, j, s_k)$ ($WI(i, s_k)$). For example, we may determine that for $s_k = 1$ the support of the wavelet atom covers five pixels. Then the $WI(i, s_k)$ are approximated using the points $\{i-2, i-1, i, i+1, i+2\}$. As s_k increase, the support of the wavelet atom increase. For each edge detector in the following sections we will state the support for each s_k used.

When we say that two pixels are neighbours using 8-connectivity we mean that they are positioned next to each other either in orthogonal directions or diagonal directions. If two pixels are neighbours using 4-connectivity they are positioned next to each other in orthogonal directions. Both cases are shown in Table 4.1.

1	1	1
1	X	1
1	1	1

0	1	0
1	X	1
0	1	0

Table 4.1: All neighbours for pixel X are marked with 1. In the left table we use 8-connectivity and in the right table 4-connectivity.

4.2 1-D Multiscale edge detector

In this section we give a 1-D multiscale edge detector which detects singularities in 1-D signals. We see how edges in an image may be detected by applying the 1-D multiscale edge detector on each rown and column of the image.

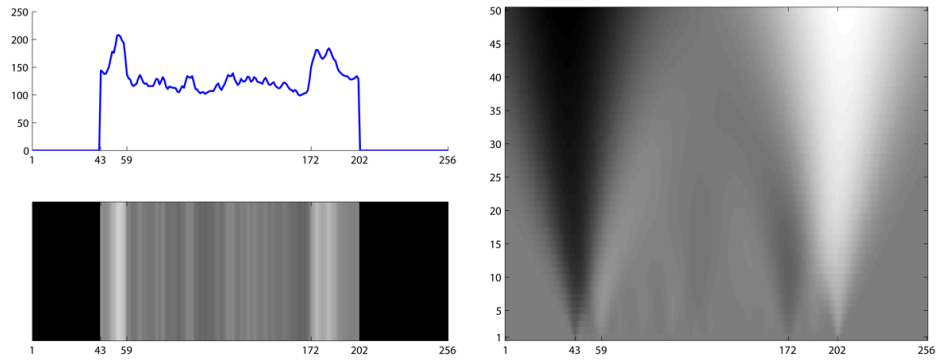
To avoid using many indices we will in this section denote by I the greyscale image $I_6 \in \mathbb{R}^{(240 \times 256)}$ shown in Figure 4.1(f). The edges of I is detected by analysing each row and column of I independently. First we look in details how row $I_{\cdot 120}$ is analysed, then the edge detector is applied for every row and column of I.

4.2.1 Application on row $I_{\cdot 120}$

Let $\psi \in L^2(\mathbb{R})$ be the first-order derivative of the normalized Gaussian function θ from Example 2.1. Denote by $WI_{\cdot 120}(i, s_k)$ the wavelet transform of $I_{\cdot 120}$ with respect to ψ given in Definition 2.6.

Since ψ has one vanishing moments Theorem 2.7 ensures that every singular point of $I_{.120}$ with $\alpha < 1$ can be located by following the maximum lines of $WI_{.120}(i, s_k)$ as s_k decrease. Also, using Theorem 2.5 $\alpha \leq 1$ may be characterised from the decay of $WI_{.120}(i(s_k), s_k)$ along the maximum line $(i(s_k), s_k)$ in the scale-space plane as s_k decrease.

We start by calculating $WI_{.120}(i, s_k)$ for scales $\{s_k = k\}_{k=1}^{50}$ and $i = 1, \ldots, 256$. The calculation is done using the wavelet toolbox in Matlab. When $s_k = 1$ the support of the wavelet atom is five pixels and as s_k gets larger the support increases. In Figure 4.2 $I_{.120}$ and $WI_{.120}(i, s_k)$ are plotted. From (b) we see that the intensity changes of $I_{.120}(k)$ at $i = 43, 59, 172, 202$ contributes with highest value of $|WI_{.120}(i, s_k)|$. The intensity increase at $i = 43$ and $i = 172$ correspond to negative values of $WI_{.120}$ while the intensity decrease at $i = 59$ and $i = 202$ correspond to positive values.



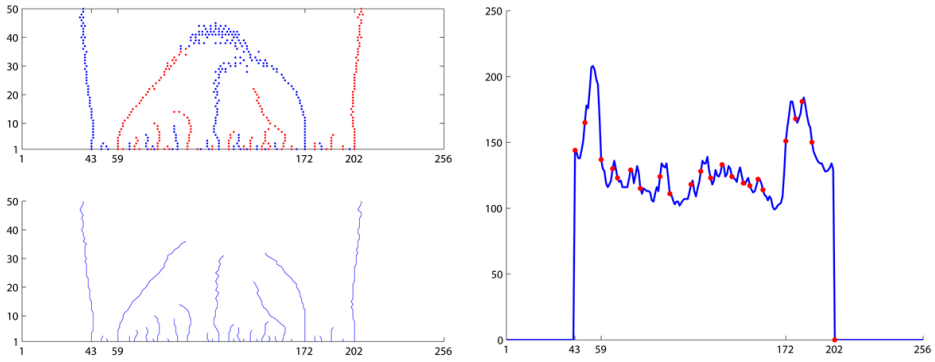
(a) (b)

Figure 4.2: (a)The uppermost graph plots $I_{.120}(i)$ as a function. Below it is plotted as an image. (b)$WI_{.120}(i, s_k)$ for $\{s_k = k\}_{k=1}^{50}$ and $i = 1, \ldots, n$. Black, white and grey correspond to values that are respectively negative, positive and zero.

Next all maximum points $\{m_{nk} = (n, s_k)\}$ of $|WI_{.120}(\cdot, s_k)|$ are located for each s_k. Then each maximum point m_{nk} is classified as either belonging to some maximum line l_p or not belonging to any maximum line. Each m_{nk} can only belong to one or none maximum line. This classification is done such that $WI_{.120}(n, s_k)$ has constant sign for all m_{nk} from the same maximum line. We have implemented both the location of maximum points and classification of maximum points in Matlab.

In Figure 4.3(a) the maximum points $\{m_{nk}\}$ are plotted in the uppermost graph. All m_{nk} such that $WI_{.120}(n, s_k) < 0$ and $WI_{.120}(n, s_k) > 0$ are plotted in blue and red respectively. Each line plotted in the lowermost graph connects all maximum points classified as belonging to the same maximum line. In (b) $I_{.120}$ is plotted and each point that a maximum line converges

towards is marked with a red dot.



(a) (b)

Figure 4.3: (a)The uppermost graph plots all maximum points (n, s_k). Blue and red dots correspond to negative and positive value of $WI_{.120}(n, s_k)$ respectively. In the lowermost graph the corresponding maximum lines are plotted. (b)Plot of $I_{.120}$. Each point which a maximum line ends at $s_k = 1$ is marked with a red dot.

Denote by u_p as $u_p = n$ such that $(n, s_1) \in l_p$, i.e u_p is the point which the maximum line l_p ends up at the lowest scale s_1. The Lipschitz regularity α_p of $I_{.120}$ at u_p are calculated using the decay of $|WI_{.120}(n, s_k)|$ along the maximum line l_p. Since $|WI_{.120}(n, s_k)|$ decays as $s_k^{\alpha_p + \frac{1}{2}}$ when $I_{.120}$ is Lipschitz α_p at u_p then $\alpha_p = a - \frac{1}{2}$ where a is the slope of $\log |Wf(n, s_j)|$ versus $\log s_k$. In Figure 4.4 a histogram of the calculated α_p for each u_p is shown. Most of the points u_p have Lipschits regularity between -0.5 and 0.5. Lipschitz regularity α_p tells us something about the behaviour of $I_{.120}$ in a neighbourhood of u_p, but not the amplitude of the intensity change. A step edge with small amplitude has the same regularity as a step edge with high amplitude. Therefore Lipschitz regularity is not usefull for determine which u_p correspond to high intensity change in $I_{.120}$. On the other hand Lipschitz regularity may be used to distinguish types of singularities. For example, from Table 4.2 we see that the points $u_p = 43$ and $u_p = 202$ behave as step edges.

The irregular signal yields many maximum lines pointing towards non-important points u_p. We are not interested in detecting the small signal fluctations since they are mainly caused by noise. In this case we are interested in detecting the edge around the dark area in Figure 4.1(f) shaped as an egg. The points of interest in $I_{.120}$ is $u_p = 43, 59, 172, 202$. As we saw in Figure 4.2(b) these are the points corresponding to the intensity change in $I_{.120}$ that contributes most to the value of $|WI_{.120}(u_p, s_k)|$. We suggest thresholding on

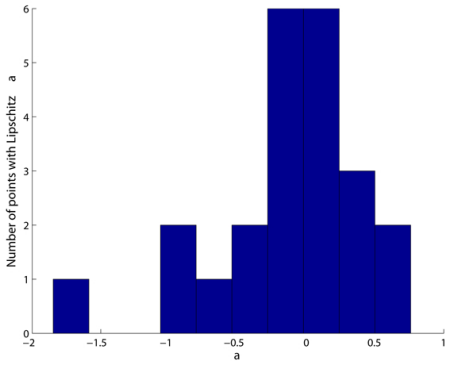
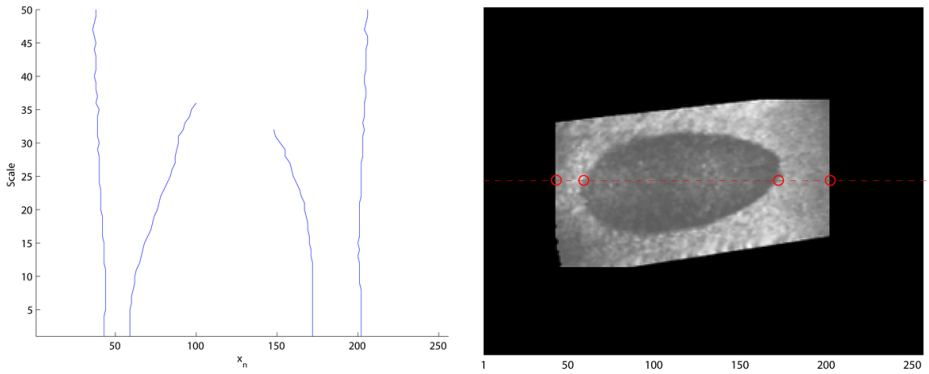


Figure 4.4: Histogram of the calculated α.

the summation of $|WI_{\cdot 120}(n, s_k)|$ over maximum lines $\{l_p\}_{p\in\mathbb{Z}}$:

$$S_{l_p} = \sum_{(n,s_k)\in l_p} |WI_{\cdot 120}(n, s_k)|$$

Denote by $\overline{S}$ the mean value of all S_{l_p}. We suggest setting the threshold $T = C\overline{S}$ where $C > 0$ is some constant. For images with little noise C should be chosen between zero and one. For images with substantial amount of noise C should be greater than one. All maximum lines l_p such that $S_{l_p} < T$ are classified as non-important, the rest as important. In Figure 4.5 (a) the important maximum lines are plotted for $C = 1.5$. In (b) the points $\{u_p\}_p$ corresponding to important maximum lines $\{l_p\}$ are marked with red circles.



(a) (b)

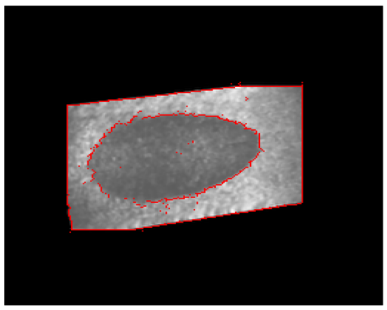
Figure 4.5: (a) The important maximum lines for I using $C = 1.5$. (b) Detected edgepoints plotted as red circles.

u_p	α_p
43	0.0134
59	0.3566
172	0.4357
202	-0.0072

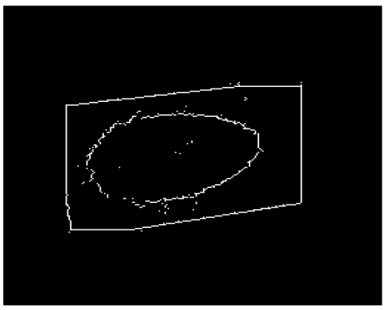
Table 4.2: The calculated α's

4.2.2 Application on images

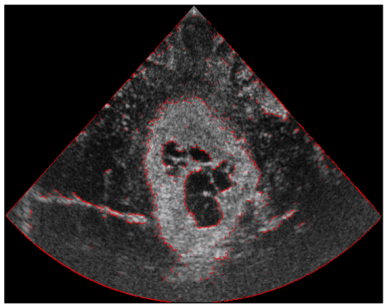
In this subsection we present the result of the 1-D multiscale edge detector applied on each row and column of some images.



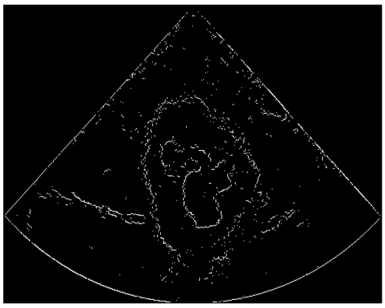
(a) I_6 with edges using $C = 0.9$



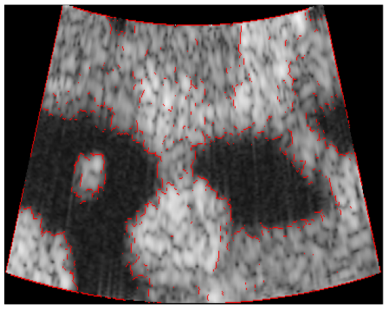
(b) Edges for I_6 using $C = 0.9$



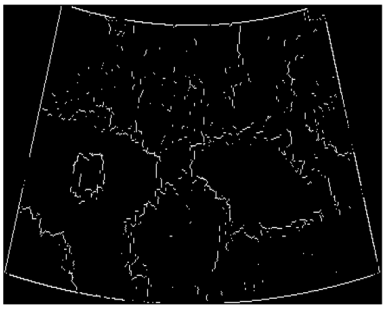
(c) I_5 with edges using $C = 6.0$



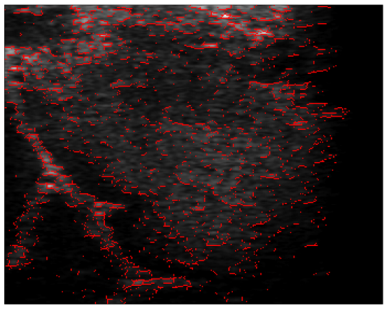
(d) Edges for I_5 using $C = 6.0$



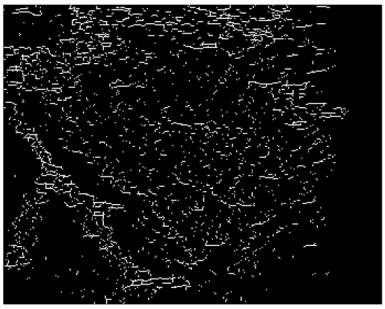
(e) I_3 with edges using $C = 2.5$



(f) Edges for I_3 using $C = 2.5$



(g) I_4 with edges using $C = 1.8$



(h) Edges for I_4 using $C = 1.8$

Figure 4.6: The 1-D multiscale edge detector applied on the images I_3, I_4, I_5 and I_6.

4.2.3 Discussion

The 1-D multiscale edge detector works well on one-dimensional signals. By connecting maximum modulus points through scales we get good localization of singular points. For each singular point Lipschitz regularity may be estimated using the value of the wavelet transform modulus along the corresponding maximum line. From both the length of the maximum line and the value of the wavelet transform we may determine if the singular point is subject to a high amplitude change or not. Both step edges $u_{p_1} = 43$ and $u_{p_2} = 202$ in $I_{\cdot 120}$ plotted in Figure 4.5(b) have high amplitude change and the corresponding maximum lines l_{p_1} and l_{p_2} are long. Lipschitz regularity is estimated to be $\alpha_{p_1} = 0.0134$ and $\alpha_{p_2} = -0.0072$. From the values $S_{l_{p_1}}$, $S_{l_{p_2}}$, α_{p_1} and α_{p_1} we determine that $I_{\cdot 120}$ has step edges of high amplitude at positions u_{p_1} and u_{p_2}. The threshold $T = C\overline{S}$ works well in the cases where the important singularities have high amplitude compared to the singularities due to noise. However, it is not possible to find one C that works optimal for each signal. When the 1-D multiscale edge detector is applied on each row and column of some image the result varies depending on the image. For an image where most of the singular points are due to noise a high value of $C > 1$ should be chosen. If the image has little noise and most of the intensity changes are due to important structures then a low value of $0 < C < 1$ should be chosen.

The image I_6 shown in Figure 4.1(f) has quite distinct jumps and the singular points corresponding to noise have low amplitude change. As we see from the Figure 4.6(b) the choice $C = 0.9$ results in output for each row and column consisting of mostly important edges. For the images I_3 and I_4 the edges are plotted in Figure 4.6(f) and (h) using $C = 2.5$ and $C = 1.8$ respectively. The result is not nearly as good since both images have much noise with vague contoures. For the image I_5 the edges are plotted in Figure 4.6(d). The hazy edges at the bottom of the object are not detected successfully. The points are spread and it is not obvious from those points how the contour of the objects looks like.

In the 1-D multiscale edge detector edges are detected by rows and columns independently. When the edge is weak in both directions it wont be detected well using this method. Instead we should combine the information from both direction to better detect weak edges. In the next section we look at a 2-D edge detector which combines information from both directions when detecting edges in an image.

4.3 2-D single scale edge detector

In this section we look at a wavelet transform edge detector inspired by the detector suggested by Canny [3]. We take advantage of the fact that the 2-D wavelet transformation acts as a differential operator on a smoothed image where the amount of smoothing is given by scale s.

We use the 2-D wavelet discussed in Example 2.2

$$\psi^1(x,y) = \frac{\partial\theta(x,y)}{\partial x} \tag{4.1}$$

$$\psi^2(x,y) = \frac{\partial\theta(x,y)}{\partial y} \tag{4.2}$$

where $\theta(x,y)$ is a normalized 2-D Gaussian.

We have seen that the 2-D wavelet transform of $f \in L^2\left(\mathbb{R}^2\right)$ then can be written as the gradient of $(f \star \theta_s)(u,v)$:

$$\begin{aligned}\mathbf{W}f(u,v,s) &= \begin{pmatrix} W^1 f(u,v,s) =< f, \psi^1_{u,v,s} > \\ W^2 f(u,v,s) =< f, \psi^2_{u,v,s} > \end{pmatrix} \\ &= -s\nabla(f \star \theta_s)(u,v)\end{aligned}$$

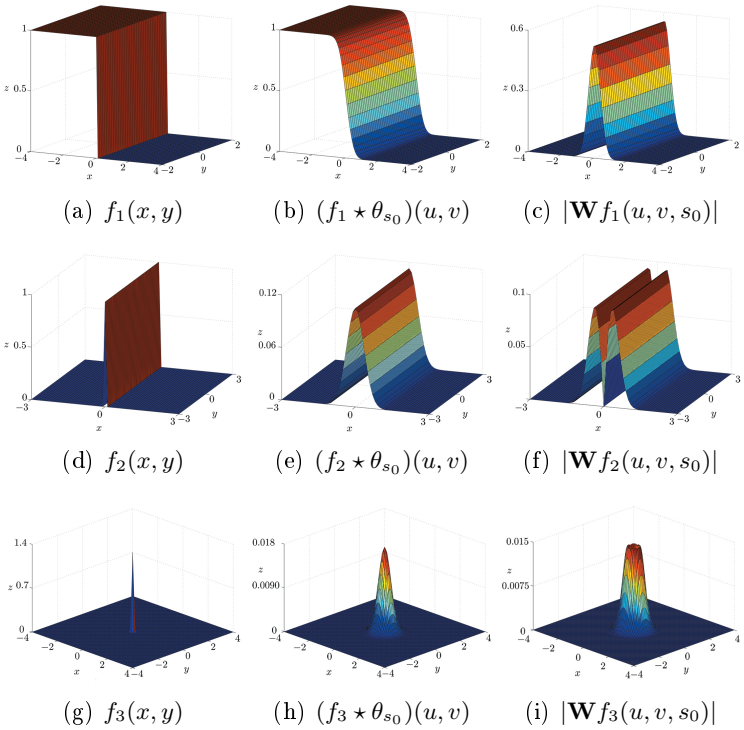
Thus $\mathbf{W}f(u,v,s)$ points in the direction where $(f \star \theta_s)(u,v)$ changes the most, i.e where it's partial derivatives are maximum. Actually, since the gradient is multiplied with $-s$ then $\mathbf{W}f(u,v,s)$ points in the direction where the smoothed function decrease most.

Three distinct edges f_1, f_2 and f_3 are depict in Figure 4.7 (a), (d) and (g). For each edge f_i, $i = 1,2,3$ the corresponding $(f_i \star \theta_{s_0})(u,v)$ and $|\mathbf{W}f_i(u,v,s_0)|$ are plotted.

The 2-D wavelet $\Psi = \{\psi^1, \psi^2\}$ is designed such that ψ^1 and ψ^2 feels edges in the x-direction and y-direction respectively.

Let us look at the step edge in the x-direction $f_1(x,y)$ depict in Figure 4.7(a). From Figure 4.7(b) we see that the gradient of $(f_1 \star \theta_{s_0})(u,v)$ is zero in the v-direction. Thus $W^2 f_1(u,v,s_0) = 0$, i.e $W^2 f_1(u,v,s_0)$ ignores step edges in the x-direction. However, the step edge is felt by $W^1 f_1(u,v,s_0)$ which has one local modulus maximum point in the u-direction for each v. The important observation is that $\mathbf{W}f(u,v,s_0)$ points in the direction normal to the step edge and $|\mathbf{W}f(u,v,s_0)|$ is local maximum in that direction for all points along the step edge.

The edge f_2 depict in Figure 4.7(d) is of short duration. This results in $(f_2 \star \theta_{s_0})(u,v)$ having a sharp intensity increase followed by a sharp intensity decrease. Then $W^1 f_2(u,v,s_0)$ has two local modulus maximum points in



(a) $f_1(x,y)$ (b) $(f_1 \star \theta_{s_0})(u,v)$ (c) $|\mathbf{W}f_1(u,v,s_0)|$

(d) $f_2(x,y)$ (e) $(f_2 \star \theta_{s_0})(u,v)$ (f) $|\mathbf{W}f_2(u,v,s_0)|$

(g) $f_3(x,y)$ (h) $(f_3 \star \theta_{s_0})(u,v)$ (i) $|\mathbf{W}f_3(u,v,s_0)|$

Figure 4.7: Edges and how they are smoothed

the u-direction for each v. In this case when (u,v) lies on either side of the edge $\mathbf{W}f_2(u,v,s_0)$ points in the direction normal to the edge such that $(f_2 \star \theta_{s_0})$ decreases most. So, for the edge of short duration $|\mathbf{W}f_2(u,v,s_0)|$ has two local maximum points, one corresponding to the increase and the other corresponding to the decrease in f_2.

The last f_3 is normalized with very small support and intend to simulate Dirac's delta function in two-dimensions. Since a convolution with Dirac's delta yields the function itself we get $(f_3 \star \theta_{s_0}) = \theta_{s_0}$. Using substitution $\tau_x = \frac{-x}{s_0}$ and $\tau_y = \frac{-y}{s_0}$ we get

$$\begin{aligned} W^1 f_3(x,y,s_0) &= -s\frac{\partial \theta_{s_0}(x,y)}{\partial x} \\ &= \frac{-s}{s_0^2}\frac{\partial \theta(\tau_x,\tau_y)}{\partial \tau_x}\frac{\partial \tau_x}{\partial x} \\ &= \frac{1}{s_0^2}\frac{\partial \theta(x,y)}{\partial x} \\ &= \frac{1}{s_0^2}\psi^1(x,y) \end{aligned} \tag{4.3}$$

The same procedure can be used to show $W^2 f_3(x,y,s_0) = \frac{1}{s_0^2}\psi^2$. Then we

get

$$|\mathbf{W}f_3(u,v,s_0)| = \sqrt{s_0^{-4}\left(\psi^1(u,v)^2 + \psi^2(u,v)^2\right)}$$
$$\leq As_0^{-2} \text{ with } A = \max_{(u,v)\in\mathbb{R}^2} \sqrt{\psi^1(u,v)^2 + \psi^2(u,v)^2} \quad (4.4)$$

Thus as the scale decreases the value of $|\mathbf{W}f_3(u,v,s_0)|$ increases.

For a greyscale image $I \in \mathbb{R}^{(m\times n)}$ we would like to detect the points (i,j) where I has sharp intensity change. Let $AI(i,j,s_k) = Arg\{\mathbf{W}I(i,j,s_k)\}$ where $-\pi < AI(i,j,s_k) \leq \pi$. Denote by *a provisional edge point of I at scale s_k* as any point (i,j) such that $|\mathbf{W}I(i,j,s_k)|$ is local maximum in the direction given by $AI(i,j,s_k)$.

The 2-D single scale edge detector in this section consists of locating all provisional edge points of I at scale s_k and then thresholding with hysteresis is employed on those points. In the following subsection we explain what we mean by thresholding with hysteresis.

4.3.1 Thresholding with hysteresis

Some of the provisional edge points of I might correspond to noise and we would like to seperate them from the ones corresponding to important intensity changes. Assuming that the noise contributes with low energy in $\mathbf{W}I(\cdot,\cdot,s_k)$ we threshold on $|\mathbf{W}I(\cdot,\cdot,s_k)|$. Suppose we have a single threshold T_1 and that an edge exists such that the mean value of $|\mathbf{W}I(\cdot,\cdot,s_k)|$ along the edge equals T_1. The value $|\mathbf{W}I(\cdot,\cdot,s_k)|$ may vary along the edge due to noise which leads to a broken edge contour. This is called streaking and is a common problem with edge detectors using a single threshold.

As suggested by Canny [3] we will employ thresholding with hysteresis to reduce the probabillity of streaking. Two thresholds are set; T_{low} and T_{high}, such that $T_{low} < T_{high}$. All provisional edge points $\{(i,j)\}$ of I such that $|\mathbf{W}I(i,j,s_k)| > T_{low}$ are classified as *candidate edge points*. Then all sets of 8-connected candidate edge points containing at least one point (l,q) such that $|\mathbf{W}I(l,q,s_k)| > T_{high}$ are classified as edge points.

If isolated weak provisional edge points were to be removed using a single threshold then weak provisional edge points connected to strong provisional edge points would also be removed. Thresholding with hysteresis allows the value along connected edge points to vary, but not lower than T_{low}. In addition at least one of the connected edge points must have value higher than T_{high}. This way the probabillity of streaking is reduced while isolated weak provisional edge points are removed.

We select T_{high} so that a given fraction $0 < P_{ne} < 1$ of $|\mathbf{W}I|$ has value greater than zero and lower than T_{high}. Then we chose T_{low} to be some fraction $0 < R < 1$ of T_{high}.

Algorithm 1 describes the 2-D single scale edge detector for an image $I \in \mathbb{R}^{(m\times n)}$ at scale s_k.

Algorithm 1 2-D single scale edge detector at scale s_k

Require: $I \in \mathbb{R}^{(m\times n)}$, $s_k \geq 1$, $0 < R < 1$ and $0 < P_{ne} < 1$
Ensure: $\{(i,j) : (i,j)$ is an edge point of $I\}$

Calculate $|\mathbf{W}I(\cdot,\cdot,s_k)|$ and $AI(\cdot,\cdot,s_k)$.
$\mathcal{P} \Leftarrow \emptyset$
for $(i,j) \in \{[1,2,\ldots,m] \times [1,2,\ldots,n]\}$ **do**
    if $|\mathbf{W}I(i,j,s_k)|$ is local maximum in direction $AI(i,j,s_k)$ **then**
        $\mathcal{P} \Leftarrow \mathcal{P} \cup \{(i,j)\}$
    end if
end for
Calculate T_{high} such that $\frac{count\left(\{(i,j) \,:\, 0<|\mathbf{W}I(i,j,s_k)|<T_{high}\}\right)}{mn} = P_{ne}$
$T_{low} \Leftarrow RT_{high}$
$\mathcal{W} = \{(i,j) \in \mathcal{P} : |\mathbf{W}I(i,j,s_k)| > T_{low}\}$
$\mathcal{S} = \{(i,j) \in \mathcal{W} : |\mathbf{W}I(i,j,s_k)| > T_{high}\}$
$\mathcal{E} \Leftarrow \mathcal{S}$
for all 8-connected set of pixels $O_q = \{(i,j)\} \in \mathcal{W}$ **do**
    if $\mathcal{S} \cap O_q \neq \emptyset$ **then**
        $\mathcal{E} \Leftarrow \mathcal{E} \cup (O_q \setminus \mathcal{S})$
    end if
end for
return $\mathcal{E}$

Implementation

We have implemented Algorithm 1 in Matlab. The Matlab routine *"conv2.m"* was used to calculate $(I \star \theta_{s_k})$ and *"grad.m"* to calculate $-s_k \nabla (I \star \theta_{s_k})$. The support of $\theta_{s_k}(i,j) = (\pi s_k^2)^{-1} e^{-(i^2+j^2)s_k^{-2}}$ includes all (i,j) such that $\theta_{s_k}(i,j) > 0.0001$. In Table 4.3 the support of θ_{s_k} for each $s_k = 1, \cdots, 10$ is listed. For example when $s_k = 4$ then $(I \star \theta_{s_k})(i,j)$ is a weighted sum of I over the $361 = (9+9+1)^2$ pixels $\{(q,l)\}$ located in the 19×19 square centered at (i,j). In the case when $(q,l) \notin \{[1,m] \times [1,n]\}$ we define $I(q,l) = 0$.

s_k	support of θ_{s_k}
1	$[-2,2] \times [-2,2]$
2	$[-5,5] \times [-5,5]$
3	$[-7,7] \times [-7,7]$
4	$[-9,9] \times [-9,9]$
5	$[-11,11] \times [-11,11]$
6	$[-12,12] \times [-12,12]$
7	$[-14,14] \times [-14,14]$
8	$[-15,15] \times [-15,15]$
9	$[-17,17] \times [-17,17]$
10	$[-18,18] \times [-18,18]$

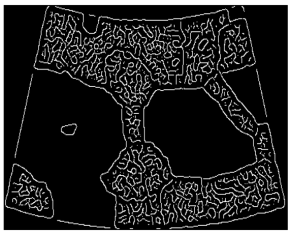
Table 4.3: The support of θ_{s_k} for $s_k = 1, \cdots, 10$

The non-maximum supression method used to locate the provisional edge points is not a standard function in Matlab, but the sub-function *"cannyFindLocalMaxima"* located in the function file *"edge.m"*. For each pixel (i,j) the method estimates values of $|\mathbf{W}I|$ between pixels using interpolation. Then these estimates are used to check if (i,j) is a local modulus maximum point in direction $AI(i,j,s_k)$.

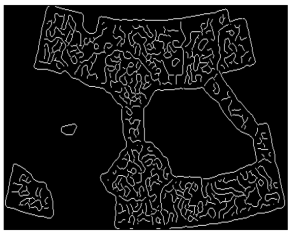
4.3.2 Result

In this subsection we look at the output of Algorithm 1 for different inputs. For each of the four images I_2, I_3, I_4 and I_5 shown in Figure 4.1 the Algorithm 1 is applied at scales $s_k = 4, 6, 8$. The parameters $P_{ne} = 0.7$ and $R = 0.3$ are used in all twelve cases.

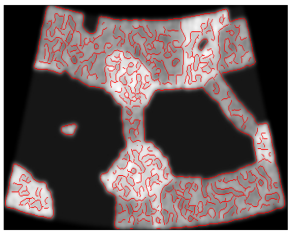
For each twelve cases we plot the provisional edge points and the edge points after thresholding. The edge points after thresholding is plotted both as an binary image and as red pixels in the smoothed image $(I \star \theta_{s_k})(i,j)$.



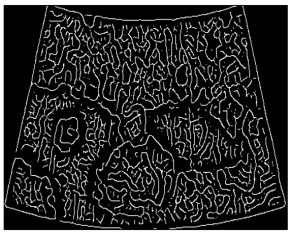
(a) Provisional edgepoints for I_2



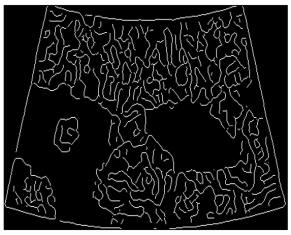
(b) Edge points for I_2



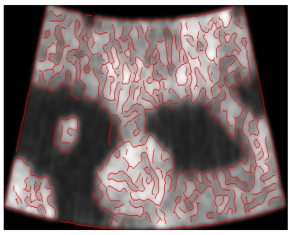
(c) $(I_2 \star \theta_4)$ where edge points are marked as red pixels.



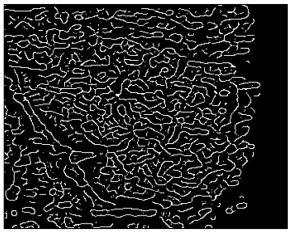
(d) Provisional edgepoints for I_3



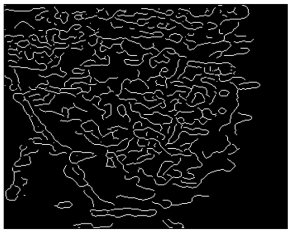
(e) Edge points for I_3



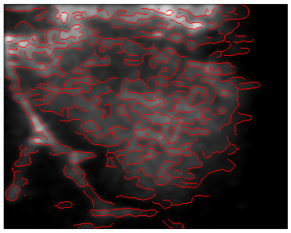
(f) $(I_3 \star \theta_4)$ where edge points are marked as red pixels.



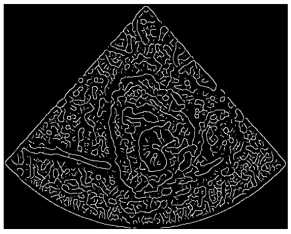
(g) Provisional edgepoints for I_4



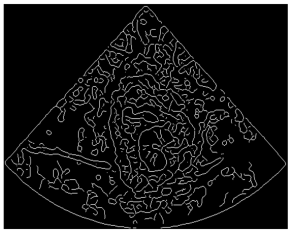
(h) Edge points for I_4



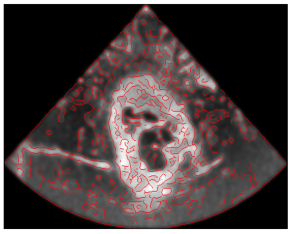
(i) $(I_4 \star \theta_4)$ where edge points are marked as red pixels.



(j) Provisional edgepoints for I_5

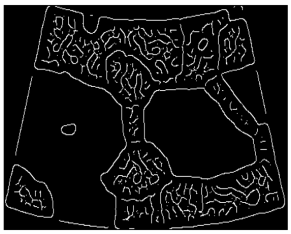


(k) Edge points for I_5

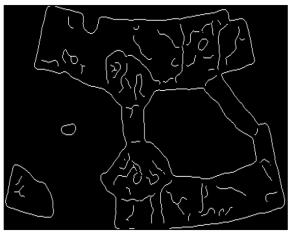


(l) $(I_5 \star \theta_4)$ where edge points are marked as red pixels.

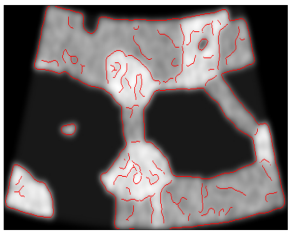
Figure 4.8: Algorithm 1 applied on the images I_2, I_3, I_4 and I_5 shown in Figure 4.1 at scale $s_k = 4$. The parameters used are $P_{ne} = 0.7$ and $R = 0.3$.



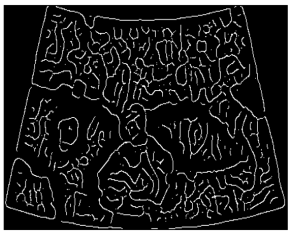
(a) Provisional edgepoints for I_2



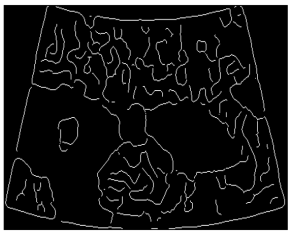
(b) Edge points for I_2



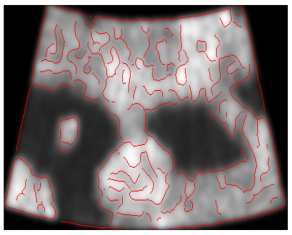
(c) $(I_2 \star \theta_6)$ where edge points are marked as red pixels.



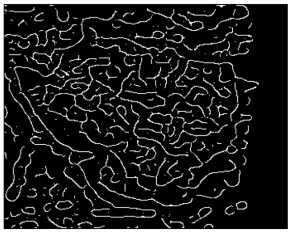
(d) Provisional edgepoints for I_3



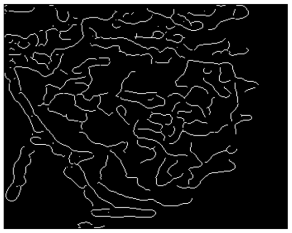
(e) Edge points for I_3



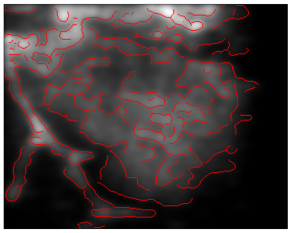
(f) $(I_3 \star \theta_6)$ where edge points are marked as red pixels.



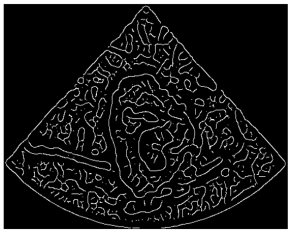
(g) Provisional edgepoints for I_4



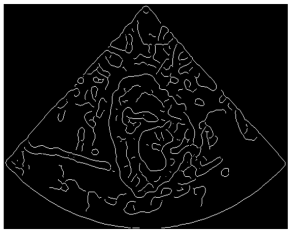
(h) Edge points for I_4



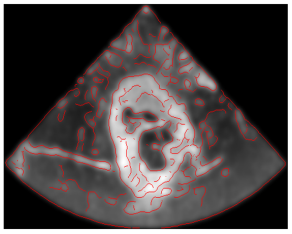
(i) $(I_4 \star \theta_6)$ where edge points are marked as red pixels.



(j) Provisional edgepoints for I_5

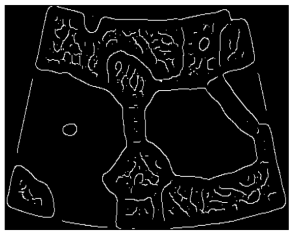


(k) Edge points for I_5

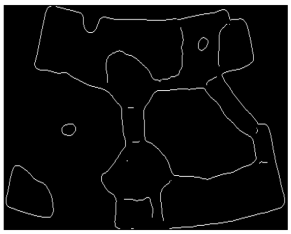


(l) $(I_5 \star \theta_6)$ where edge points are marked as red pixels.

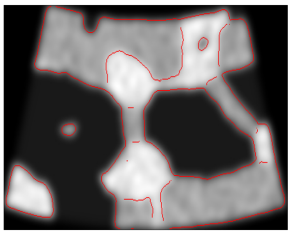
Figure 4.9: Algorithm 1 applied on the images I_2, I_3, I_4 and I_5 shown in Figure 4.1 at scale $s_k = 6$. The parameters used are $P_{ne} = 0.7$ and $R = 0.3$.



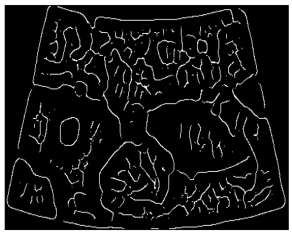
(a) Provisional edgepoints for I_2



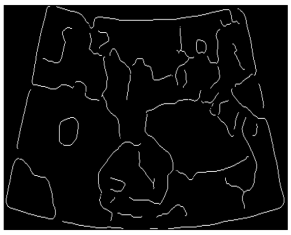
(b) Edge points for I_2



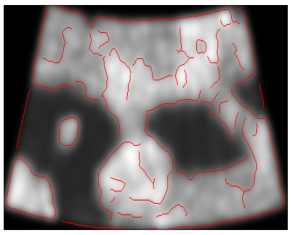
(c) $(I_2 \star \theta_8)$ where edge points are marked as red pixels.



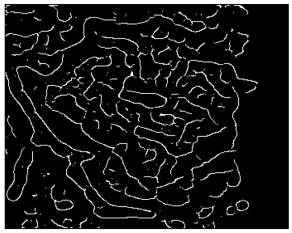
(d) Provisional edgepoints for I_3



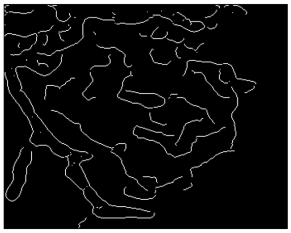
(e) Edge points for I_3



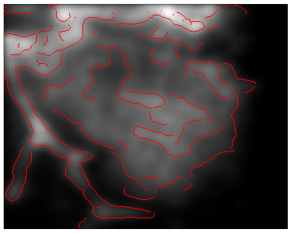
(f) $(I_3 \star \theta_8)$ where edge points are marked as red pixels.



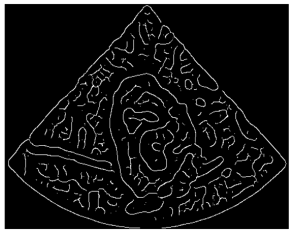
(g) Provisional edgepoints for I_4



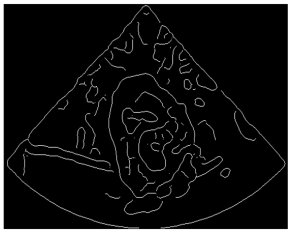
(h) Edge points for I_4



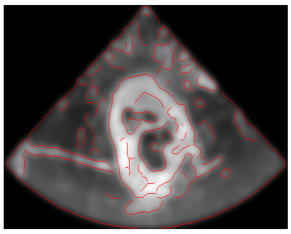
(i) $(I_4 \star \theta_8)$ where edge points are marked as red pixels.



(j) Provisional edgepoints for I_5



(k) Edge points for I_5



(l) $(I_5 \star \theta_8)$ where edge points are marked as red pixels.

Figure 4.10: Algorithm 1 applied on the images I_2, I_3, I_4 and I_5 shown in Figure 4.1 at scales $s_k = 8$. The parameters used are $P_{ne} = 0.7$ and $R = 0.3$.

4.3.3 Discussion

In general, most of the edges detected for small s_k are dominated by noise. For example from Figure 4.8(e) it is very hard to get an impression of the important structures in the image. Most of the detected edges follows the small structures caused by simulated speckle noise. However, in Figure 4.10(e) the edges follows larger and, in this case more, important structures. This behavior was expected since larger s_k means that the noise is smoothed away more. The cost of this smoothing is worser edge localisation.

For which scale s_k the edge detection should be performed depends on the image analysed. If an image has high contrast structures and small amount of noise, then small s_k should be used to get best localisation of edges. The image I_5 is such an image. In comparison, the image I_3 has low contrast and quite heavily noise, therefore larger s_k should be used.

In all four cases the thresholding only removed edges located at low contrast areas of the image. However, for some images and scales there are still detected edges which should be removed. For example image I_5 has quite high contrast edges, but as we see in Figure 4.9(k) edges in low contrast areas are detected.

In the next section we suggest a method using information across scales to determine at which scale s_k we should detect the provisional edge points. In addition we suggest a thresholding method rewarding both length and high contrast edges.

4.4 2-D Multiscale edge detector

In this section we suggest an edge detector which estimates the smallest scale s_c for which the provisional edge points at s_c are not dominated by noise. Then thresholding on the provisional edge points of I at s_c is employed. An alternative to the thresholding with hysteresis is suggested.

Let I_1, I_2, I_3, I_4, I_5 and I_6 be the images from last section shown in Figure 4.1.

4.4.1 Edge curves

Using 8-connectivity the provisional edge points of an image I at scale s_k are connected to form a set of edge curves; $\{e_{qk}\}_{q=1}^{N_{ec}(s_k)}$ where $N_{ec}(s_k)$ is the number of edge curves at scale s_k.

All but two points along an edge curve must have exactly two neighbours belonging to the same edge curve, i.e an edge curve has exactly two endpoints.

Let us define a function giving the smallest angle between two vectors with angles α and β:

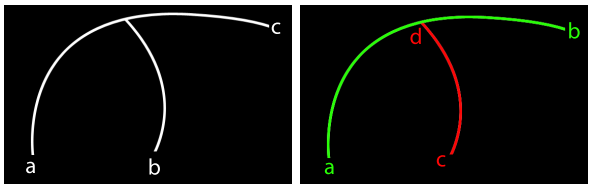
Definition 4.1. Let $-\pi < \alpha, \beta <= \pi$. Then the distance $d(\alpha, \beta)$ is given by

$$d(\alpha,\beta) = \begin{cases} |\alpha| + |\beta| & \text{if} \quad \alpha\beta < 0 \quad \text{and} \quad |\alpha| + |\beta| <= \pi \\ 2\pi - |\alpha| - |\beta| & \text{if} \quad \alpha\beta < 0 \quad \text{and} \quad |\alpha| + |\beta| > \pi \\ |\alpha - \beta| & \textit{otherwise} \end{cases} \tag{4.5}$$

Let $d(\alpha, \beta)$ be the distance function from Definition 4.1. Each provisional edge point (i, j) at scale s_k can only belong to the edge curve e_{qk} containing it's neighbour (p, l) of shortest distance $d\left(A(i, j, s_k), A(p, l, s_k)\right)$ and (p, l) cannot have any neighbour (a, b) from another curve e_n such that

$$d\left(A(p, l, s_k), A(a, b, s_k)\right) < d\left(A(i, j, s_k), A(p, l, s_k)\right)$$

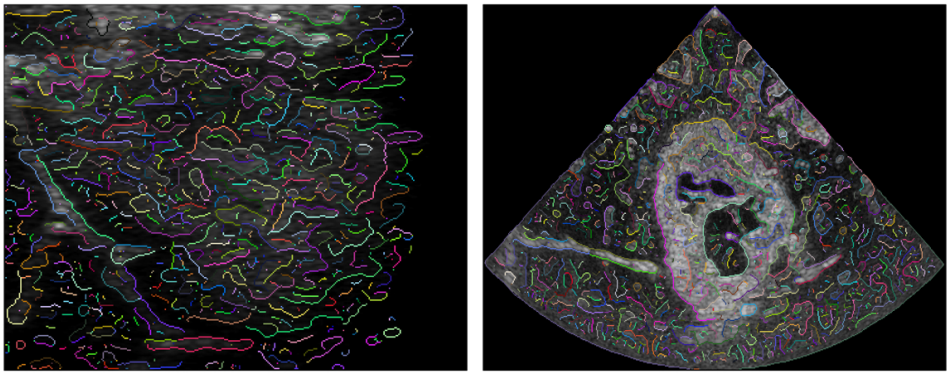
This criteria makes sure that a 8-connected set of provisional edge points having more than two endpoints are seperetad into two or more edge curves such that the gradient direction changes minimum along the edge curves.



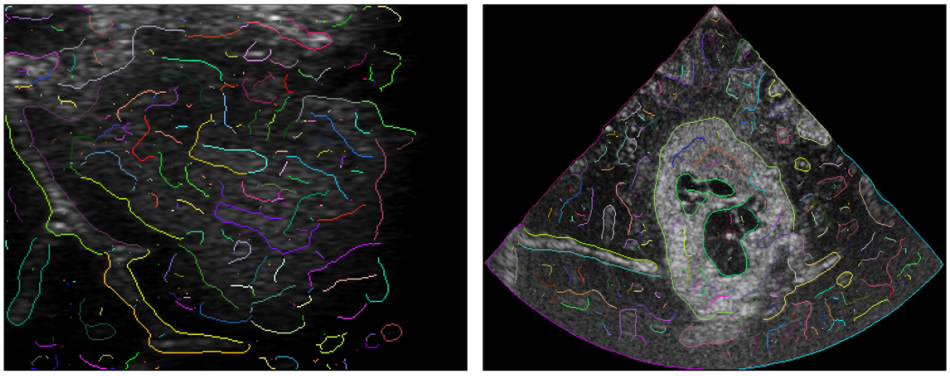
(a) Connected set having three endpoints (b) Corresponding edge curves

Figure 4.11: (a) A connected set of provisional edge points with three endpoints a,b and c. (b) The corresponding set of edge curves. Each edge curve is drawn in one color.

In Figure 4.12 all edge curves at scales $s_k = 4, 8$ for the images I_3 and I_4 are plotted such that the color does no change along an edge curve.



(a) $s_k = 4$ (b) $s_k = 4$

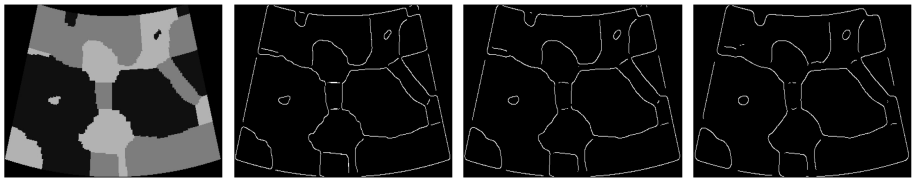


(c) $s_k = 8$ (d) $s_k = 8$

Figure 4.12: The images show all edge curves from scale $s_k = 8$ and $s_k = 4$. The edge curves are plotted on top of the corresponding image such that the color does not change along an edge curve.

4.4.2 Deciding which scale to use for final edge detection

As the scale s_k decreases $\mathbf{W}I(i, j, s_k)$ gets more sensitive to intensity changes in I of short duration and if I has many small structures then I has many provisional edge points. From Figure 4.8, Figure 4.9 and Figure 4.10 we can see that the number of provisional edge points increase as s_k decrease. Specially in noisy image the increase is quite large. For comparison all provisional edge points of an image without noise at scales $s_k = 8, 6, 4$ are plotted in Figure 4.13. We can see that the number of points is almost constant across the three scales.



(a) I_1 (b) $s_k = 4$ (c) $s_k = 6$ (d) $s_k = 8$

Figure 4.13: Algorithm 1 applied on an image whithout noise at scales $s_k = 4, 6, 8$

Denote by $N_{ep}I(s_k)$ the number of provisional edge point of I at scale s_k and $N_{ec}I(s_k)$ the number of edge curves of I at scale s_k. As we see from Figure 4.12 $N_{ec}I_3(4) > N_{ec}I_3(8)$ and $N_{ec}I_4(4) > N_{ec}I_4(8)$, i.e there are more edge curves at scale $s_k = 4$ than $s_k = 8$.

Let us see if the increase of $N_{ep}I(s_k)$ or $N_{ec}I(s_k)$ as s_k decrease is usefull for deciding at which scale s_k thresholding on the provisional edge points should be done.

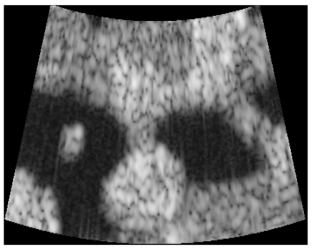
In Figure 4.14 (d) - (i) $N_{ep}I_i(s_k)$ and $N_{ec}I_i(s_k)$ $(i = 2, 3, 4)$ and $s_k = 1, \ldots, 10$ are plotted. For all three images $N_{ec}I_i(s_k)$ has a distinct increase in growth starting at some s_k. This is not the case for $N_{ep}I_i(s)$ which has a more evenly increase as the scale decrease.

For a noisy image an edge curve at large scale will at some lower scale be divided into shorter edge curves. This means that at the scale in which the edge curves are dominated by noise we will see an increase in $N_{ec}I_i(s_k)$ reflecting both the addition of new edge curves due to smaller structures/noise and the dividing of edge curves into shorter edge curves due to noise. On the other hand, the increase of $N_{ep}I_i(s_k)$ only reflects the addition of new provisional edge points due to smaller structures/noise. When $N_{ec}I_i(s_k)$ increase by one then $N_{ep}I_i(s_k)$ increase by n where n is the number of points along the added edge curve. Thus $|\frac{d}{ds_k}N_{ep}I_i(s_k)| > |\frac{d}{ds_k}N_{ec}I_i(s_k)|$ as shown in Figure 4.14 (j) - (l). Based on these observations we suggest choosing the smallest scale s_c such that for some chosen $T_{N_{ec}} > 0$ we have $(-1)\frac{d}{ds_k}N_{ec}I_i(s_k) < T_{N_{ec}}$ for

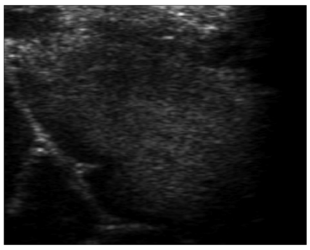
$s_k \geq s_c$. Then employ thresholding on the provisional edge points of I_i at s_c.

In the next subsection we suggest thresholding using edge curves.

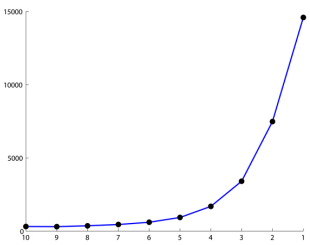
(a) $I_4(n,m)$



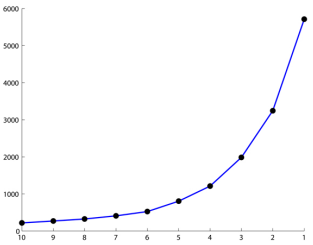
(b) $I_2(n,m)$



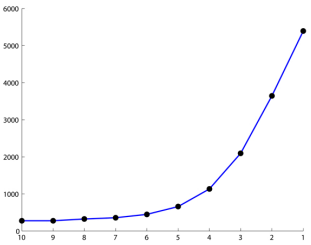
(c) $I_3(n,m)$



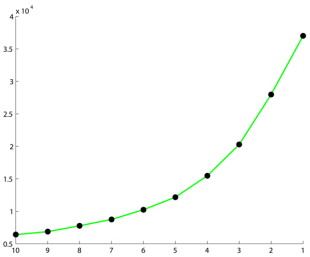
(d) $N_{ec}I_1(s_k)$, $s_k = 10,\ldots,1$



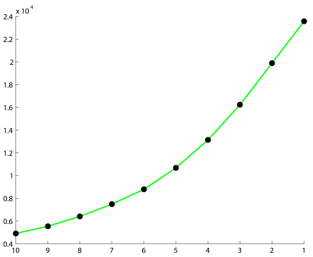
(e) $N_{ec}I_2(s_k)$, $s_k = 10,\ldots,1$



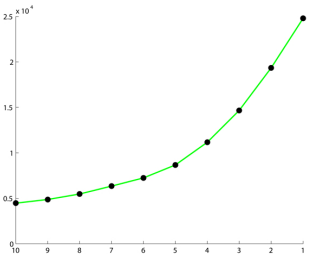
(f) $N_{ec}I_3(s_k)$, $s_k = 10,\ldots,1$



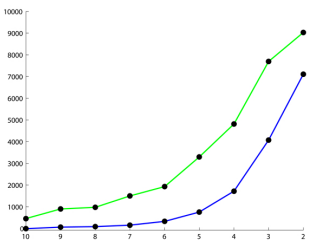
(g) $N_{ep}I_1(s_k)$, $s_k = 10,\ldots,1$



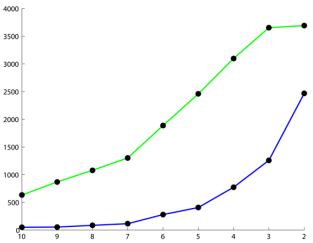
(h) $N_{ep}I_2(s_k)$, $s_k = 10,\ldots,1$



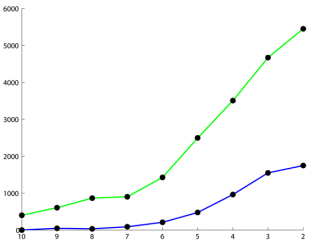
(i) $N_{ep}I_3(s_k)$, $s_k = 10,\ldots,1$



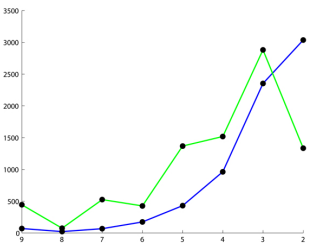
(j) $\frac{d}{ds_k}N_{ep}I_1(s_k)$ plotted in green. $\frac{d}{ds}N_{ec}I_1(s)$ plotted in blue.



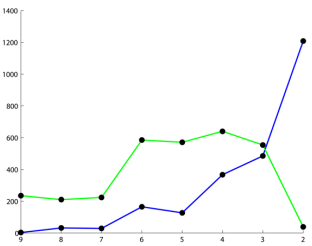
(k) $\frac{d}{ds_k}N_{ep}I_2(s_k)$ plotted in green. $\frac{d}{ds}N_{ec}I_2(s)$ plotted in blue.



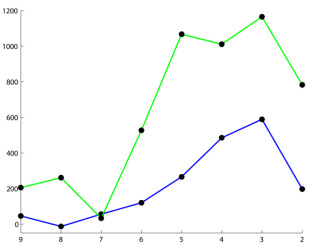
(l) $\frac{d}{ds_k}N_{ep}I_3(s_k)$ plotted in green. $\frac{d}{ds}N_{ec}I_3(s)$ plotted in blue.



(m) $\frac{d^2}{ds_k^2}N_{ep}I_1(s_k)$



(n) $\frac{d^2}{ds_k^2}N_{ep}I_2(s_k)$



(o) $\frac{d^2}{ds_k^2}N_{ep}I_3(s_k)$

Figure 4.14: For each image (a), (b) and (c) the corresponding number of edge curves and edge points VS s_k are plotted below.

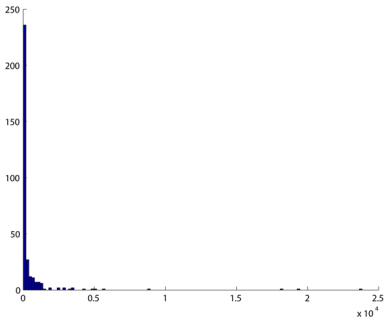


Figure 4.15: Histogram of $\{S_{qk}\}_{i=1}^{N_{ec}I_2(s_k)}$ for a fixed $s_k = 8$. The height of each bar shows the number of edge curves having values S_{qk} in the intervall along the horisontal axis covered by the bar.

4.4.3 Thresholding

In this subsection we suggest a thresholding method which rewards both long edge curves and high intensity changes across the edge. In addition the threshold should also adapt to the image analysed.

Denote by S_{qk} the sum of $|\mathbf{W}I(\cdot,\cdot,s_k)|$ over edge curve e_{qk}:

$$S_{qk} = \sum_{(i,j)\in e_{qk}} |\mathbf{W}I(i,j,s_k)|$$

Figure 4.15 shows a histogram of $\{S_{qk}\}_{i=1}^{N_{ec}I_2(s_k)}$ with $s_k = 8$.

As we see, most of the S_{qk} are located in the bins close to zero and are most likely a result of noise. We want to determine a threshold $T_k > 0$ such that if $S_{qk} < T_k$ then e_{qk} is removed from the family of edge curves.

There are several methods for choosing T_k. For example T_k can be chosen such that the number of edge curves with $S_{qk} < T_k$ are a given fraction $0 < P_{ec} < 1$ of the total number of edge curves $N_{ec}I(s_k)$. This means that we assume $P_{ec}N_{ec}I(s_k)$ number of edge curves are non-important while $(1 - P_{ec})\, N_{ec}I(s_k)$ are important. Then the quality of the output will depend on the image since a fixed P_{ec} will not fit every image. For an image with one large structure and little noise $\{S_{qk}\}_{q=1}^{N_{ec}I(s_k)}$ are concentrated around some value. In this case it is a better suggestion to say that most of the edge curves are important edges. A fixed P_{ec} will only fit one small class of images.

We suggest chosing T_k depending on which scale s_k the thresholding is done.

Denote by $\overline{S_k}$ as the mean value of $\{S_{qk}\}_{q=1}^{N_{ec}I(s_k)}$

$$\overline{S_k} = \frac{1}{N_{ec}I(s_k)} \sum_{q=1}^{N_{ec}I(s_k)} S_{qk}$$

As discussed in the previous subsection assume s_c is the smallest scale such that for some chosen $T_{N_{ec}} > 0$ we have $(-1)\frac{d}{ds_k}N_{ec}I_i(s_k) < T_{N_{ec}}$ for $s_k \geq s_c$.

If s_c is large this means that the image is subjected to noise of significant structure size and only edge curves e_{qk} with S_{qk} that have a significant higher value than $\overline{S_k}$ should be trusted to be important edges. If s_k is small, then the image have little noise and edge curves e_{qk} with S_{qk} close to $\overline{S_k}$ should be trusted to be important edges. We suggest chosing $T_k = C_k\overline{S_k}$ for some fixed C_k. A set $\{C_k\}_k$ should be choosen such that the thresholding works optimal on a given test set of ultrasound images.

Algorithm 2 describes the edge detector for an image I.

Algorithm 2 Edge Detector

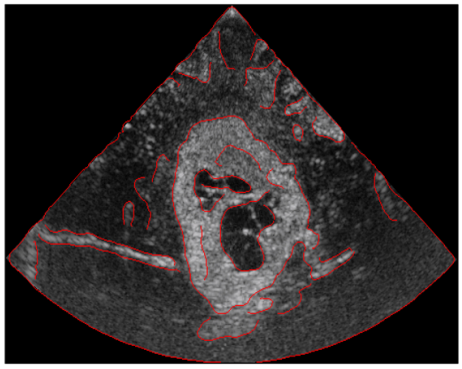
Require: $I \in \mathbb{R}^{(m\times n)}$, $\{s_k\}_{k=1,\ldots,N}$ with $s_k < s_{k+1}$, $T_{N_{ec}} > 0$ and $\{C_k\}_{k=1,\ldots,N}$

Ensure: $\{(i,j) : (i,j)$ is an edge point of $I\}$

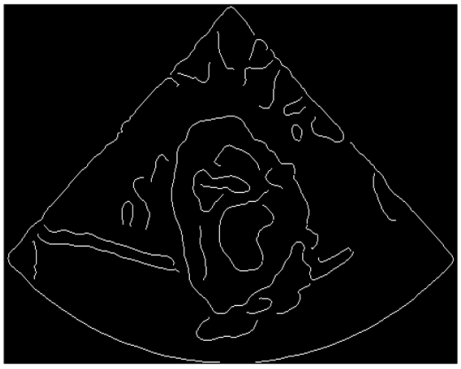
for $k = N, \ldots, 1$ **do**
  Calculate $|\mathbf{W}I(\cdot,\cdot,s_k)|$ and $AI(\cdot,\cdot,s_k)$.
  $\mathcal{P}_k \Leftarrow \emptyset$
  for $(i,j) \in \{[1,2,\ldots,m] \times [1,2,\ldots,n]\}$ **do**
    if $|\mathbf{W}f(i,j,s_k)|$ is local maximum in direction $AI(i,j,s_k)$ **then**
      $\mathcal{P}_k \Leftarrow \mathcal{P}_k \cup \{(i,j)\}$
    end if
  end for
  Generate $\{e_{qk}\}_{q=1}^{N_{ec}I(s_k)}$ from the set $\mathcal{P}_k$
  if $k = 1$ **then**
    Skip to the next iteration in the FOR-loop
  else if $k = 2$ **then**
    $\frac{d}{ds_k}N_{ec}I(s_{k-1}) \Leftarrow \frac{N_{ec}(s_{k-1})-N_{ec}(s_k)}{s_{k-1}-s_k}$
  else
    $\frac{d}{ds_k}N_{ec}f(s_{k-1}) \Leftarrow \frac{N_{ec}(s_{k-2})-N_{ec}(s_k)}{s_{k-2}-s_k}$
  end if
  if $(-1)\frac{d}{ds_k}N_{ec}f(s_{k-1}) > T_{N_{ec}}$ **then**
    Jump out of FOR-loop
  end if
end for
$c \Leftarrow k-1$
Calculate $\{S_{qc}\}_{q=1}^{N_{ec}f(s_c)}$ and $\overline{S_c}$
$T_c \Leftarrow C_c\overline{S_c}$
$\mathcal{E} \Leftarrow \emptyset$
for all q such that $S_{qc} > T_c$ **do**
  $\mathcal{E} \Leftarrow \mathcal{E} \cup \{(i,j) : (i,j) \in e_{qc}\}$
end for
return $\mathcal{E}$

4.4.4 Result

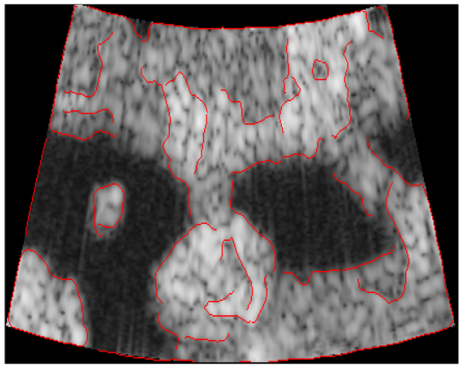
Below are some results after using the edge detector on five different images using $T_{N_{ec}} = 100$, $C = [0, 0.1, 0.6, 1.3, 1.5, 1.8, 2.2, 2.5, 2.5, 2.5]$ and $s_k = k$ for $k = 1, \ldots, 10$.



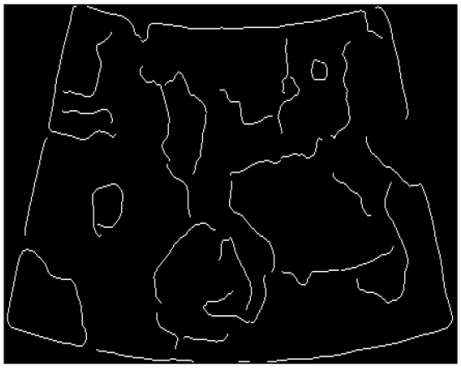
(a) Image I_5 with edges plotted as red pixels.



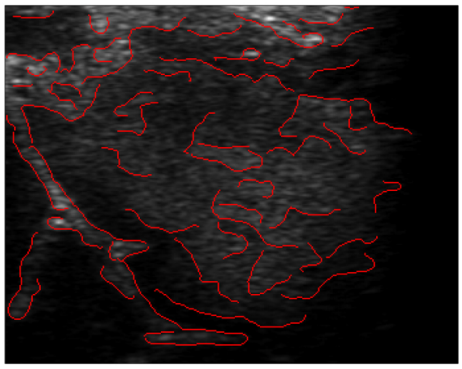
(b) Edges for image I_5. $s_c = 7$, $\overline{S_c} = 272.6$, $C_c = 2.2$ and $P_{ec} = 0.9251$



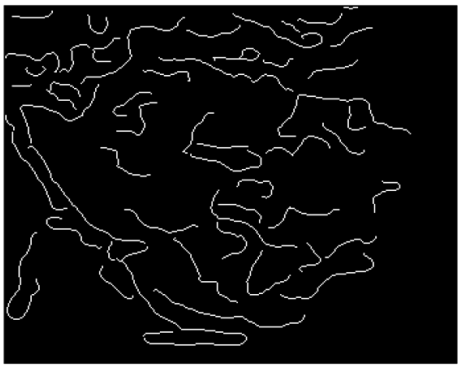
(c) Image I_3 with edges plotted as red pixels.



(d) Edges for image I_3. $s_c = 7$, $\overline{S_c} = 438.8$, $C_c = 2.2$ and $P_{ec} = 0.9171$

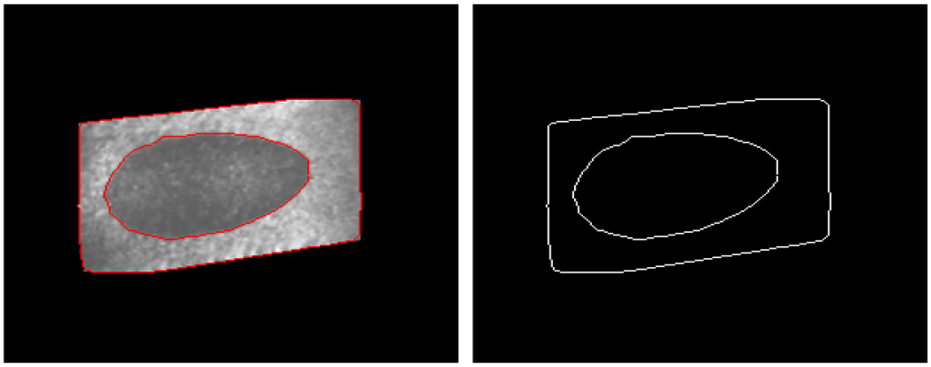


(e) Image I_4 with edges plotted as red pixels.



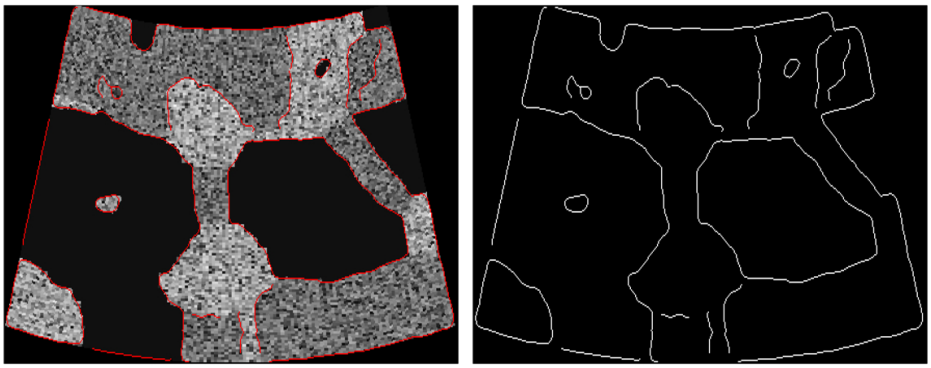
(f) Edges for image I_4. $s_c = 6$, $\overline{S_c} = 105.3$, $C_c = 1.8$ and $P_{ec} = 0.8597$

Figure 4.16: The edge detector used on different images. The same thresholding parameters was used for each image



(a) Image I_6 with edges plotted as red pixels.

(b) Edges for image I_6. $s_c = 4$, $\overline{S_c} = 396.7$, $C_c = 1.3$ and $P_{ec} = 0.9862$



(c) Image I_2 with edges plotted as red pixels.

(d) Edges for image I_2. $s_c = 6$, $\overline{S_c} = 366.7$, $C_c = 1.8$ and $P_{ec} = 0.9592$

Figure 4.17: The edge detector used on different images. The same thresholding parameters was used for each image

4.4.5 Discussion

As we see in Figure 4.16 and Figure 4.17 the result is acceptable for the quite different images I_2, I_3, I_4, I_5 and I_6. The image I_6 has distinct intensity changes and little noise. The 2-D multiscale edge detector therefore detects edges at the small scale $s_c = 4$. Only 1.38 percent of the intially detecte edge curves was not removed in the thresholding procedure. For the image I_3 having not as distinct edges and more noise than I_6 the edge detector uses the larger scale $s_c = 7$ and 8.29 percent of the initially detected edge curves was not removed in the thresholding procedure. It seems like thresholding on $\frac{d}{ds_k}N_{ec}(s_k)$ works well for determine the scale s_c. For all images the important edges are retained after thresholding on S_{qc} while keeping a low number of edges corresponding to low contrast areas. For the image I_5 our thresholding method results in fewer low contrast edges than after thresholding with hyseterias using $P_{ne} = 0.7$ and $R = 0.3$ at $s_k = 8$. The parameters P_{ne}, s_k and R could be adjusted to give better result for image I_5, but then result might not be very good for the other images.

Chapter 5

Conclusion

The wavelet transform of an image acts as a differential operator on the smoothed image and the amount of smoothing increases as the scale s increases. Therefore small s should be used for best localisation of edges. However, for small s the wavelet transform is sensitive to noise.

The 1-D multiscale edge detector works very well for locating singularities and characterising Lipschitz regularity in 1-D signals. However, as an edge detector for images the 1-D multiscale edge detector does not function satisfactory. The result is poor escpecially for noisy images containg low contrast edges. Also edges that are weak in both x-direction and y-direction wont be detected well. For images with quite distinct edges and little noise the 1-D multiscale edge detector yields edge points where most of the edge points correspond to important intensity changes in the image. For most of the images the edge points did not lie nicely aligned as curves, but they were spread with high consentration near high intensity areas in the image. Lipschitz regularity is not suited for determine which singular points are corresponding to high intensity changes. For example a step edge of low amplitude have the same Lipschitz regularity as a step edge of high amplitude.

The 2-D single scale edge detector and the 2-D multiscale edge detector differs in how thresholding is performed and that the 2-D multiscale edge detector automatically choose the scale to use when detecting the edges. The threshold $T_k = C_k\overline{S_k}$ has an overall better performance for the images in this report than the threshold with hysteresis.

The parameters N, $T_{N_{ec}}$ and $\{C_k\}_{k=1,\ldots,N}$ in the 2-D multiscale edge detector may be chosen to fit a large class of images. After the parameters are chosen the 2-D multiscale edge detector only takes as input the image to be analysed. Thus the 2-D multiscale edge detector is an user-friendly edge detector. In comparison for fixed parameters P_{ne}, s_k and R the 2-D single scale edge

detector will not function as satisfactory for a large class of images as the 2-D multiscale edge detector will.

Since the 2-D multiscale edge detector does calculations for several scales it will use longer time detecting edges than the 2-D single scale edge detector. However the running time for the 2-D multiscale edge detector may be shortened by choosing a smarter order which the calculations in Algorithm 2 is performed. For example if $(-1)\frac{d}{ds_k}N_{ec}I$ is small for some s_j then we can skip to scale s_{j+n} for some $n > 1$ and check the value at that scale instead of s_{j+1}. If $(-1)\frac{d}{ds_k}N_{ec}I$ has to high value at s_{j+n} then we go back and check some scale between s_j and s_{j+n}.

The wavelet transform is also well adapted for other image analysis. For example image compression and noise removal can be done effectively using the wavelet transform. Mallat and Zhong gives examples of image reconstruction from multiscale edges with a compression ratio over 30 [7]. The recovered images have lost some small details, but is visually of good quality. This means that most of the important information of an image can be represented by it's edges.

5.1 Further work

To derive edge detectors that are less sensitive to noise with good localisation properties further investigation on how to use the multiscale information carried by the wavelet transform should be performed. For example in the 1-D single scale edge detector we connect maximum points through scales. Maybe this can be done in the two-dimensional case as well by connecting provisional edge points through scales. Another suggestion is to connect edge curves through scales as well. Neither of these suggestions are easy to implement and more details on how points or edge curves should be connected must be derived.

Appendix A

Appendix

A.1 Some definitions

Definition A.1. *vector space* $\mathbb{V}$ is a set that is closed under finite vector addition and scalar multiplication.

Definition A.2. A *metric space* is a set $\mathbb{S}$ with a metric g that, for every two points $x, y \in S$, gives the distance between them as a nonnegative real number $g(x, y)$. A metric space must also satisfy

1. $g(x, y) = 0$ if and only if $x = y$
2. $g(x, y) = g(y, x)$
3. The triangle inequality $g(x, y) + g(y, z) >= g(x, z)$

Definition A.3. A *Cauchy sequence* is a sequence $a_1, a_2, \ldots$ such that the metric $d(a_m, a_n)$ satisfies

$$\lim_{\min m,n \to \infty} d(a_m, a_n) = 0 \tag{A.1}$$

Definition A.4. A *complete metric space* is a metric space in which every Cauchy sequence is convergent.

Definition A.5. A *Hilbert space* is a vector space $\mathbb{H}$ with an inner product $< f, g >$ such that the norm defined by

$$\|f\| = \sqrt{< f, f >} \tag{A.2}$$

turns $\mathbb{H}$ into a complete metric space.

Definition A.6. Let $\mathbb{V}$ be a vector space over a field K, and let $\mathbb{A}$ be a nonempty set. Now define addition $p + a$ in $\mathbb{A}$ for any vector $a \in \mathbb{V}$ and element $p \in A$ subject to the conditions

1. p+0=p
2. (p+a)+b=p+(a+b)
3. For any q in A, there exists a unique vector a in V such that q=p+a

Here, $a, b \in \mathbb{V}$. Then $\mathbb{A}$ is an *affine space* and K is called the coefficient field.

Proposition A.1. *Any closed subspace V of a Hilbert space H is itself a Hilbert space.*

A.2 Heisenberg's uncertainty principle

Let $f(t)$ be a function in $L^2(\mathbb{R})$ and denote

$$u = \frac{1}{\|f\|^2}\int_{-\infty}^{\infty} t|f(t)|^2 dt \tag{A.3}$$

$$\xi = \frac{1}{\|f\|^2}\int_{-\infty}^{\infty} \omega|\hat{f}(\omega)|^2 d\omega \tag{A.4}$$

Since $\int_{-\infty}^{\infty} \frac{|f(t)|^2}{\|f\|^2} dt = 1$ the integrand can be interpret as a probability density function with u as it's expected value. Using the same reasoning we interpret ξ as the the expected value for $\frac{|\hat{f}(t)|^2}{\|f\|^2}$. Denote the variances around u and ξ to be

$$\sigma_t^2 = \frac{1}{\|f\|^2}\int_{-\infty}^{\infty} (u-t)^2|f(t)|^2 dt \tag{A.5}$$

$$\sigma_\omega^2 = \frac{1}{\|f\|^2}\int_{-\infty}^{\infty} (\xi-\omega)^2|\hat{f}(\omega)|^2 d\omega \tag{A.6}$$

σ_t and σ_ω is a measure of how much $f(t)$ spreads around u and ξ respectively. We say that $f(t)$ is well localized in time if σ_t is small enough and well localized in frequency if σ_ω is small enough.

Heisenberg's uncertainty principle gives a lower bound on how well a function from $L^2(\mathbb{R})$ can be localized in time and frequency.

Heisenberg's uncertainty principle A.2. *With σ_t^2 and σ_ω^2 as above for $f \in \mathrm{L}^2(\mathbb{R})$ the following inequality is satisfied*

$$\sigma_t^2 \sigma_\omega^2 \geq \frac{1}{4} \tag{A.7}$$

This inequality is an equality if and only if there exists $(u, \xi, a, b) \in \mathbb{R}^2 \times \mathbb{C}^2$ with $b > 0$ such that

$$f(t) = a \exp^{i\xi t - b(t-u)^2} \tag{A.8}$$

[6, p. 31]

A Heisenberg box is a graphical view of the limitations given by Heisenberg's uncertainty principle. Figure A.1 shows the Heisenberg box for a function f with u, ξ, σ_t^2 and σ_ω^2 as above.

A.3 One-dimensional definitions and properties

Definition A.7. For $f, g \in L^2(\mathbb{R})$ the convolution denoted by $(f \star g)$ is given by

$$(f \star g)(x) = \int_{-\infty}^{\infty} f(u) g(x-u) du \tag{A.9}$$

Property A.3. *For $f, g \in \mathrm{L}^2(\mathbb{R})$ we have*

$$\widehat{(f \star g)}(\omega) = \hat{f}(\omega)\hat{g}(\omega) \tag{A.10}$$

[2, P. 313]

Property A.3 says that a convolution in the time-domain corresponds to a product in the Fourier-domain.

Proposition A.4. *If $f \in C^n(\mathbb{R}) \cap L^1(\mathbb{R})$ and if all derivatives $f^{(k)}$, $k = 1, \ldots, n$ are in $L^1(\mathbb{R})$, then*

$$\widehat{f^{(k)}}(\omega) = (2i\pi\omega)^k \widehat{f}(\omega) \text{ for } k = 1, \ldots, n$$

[2, P. 157]

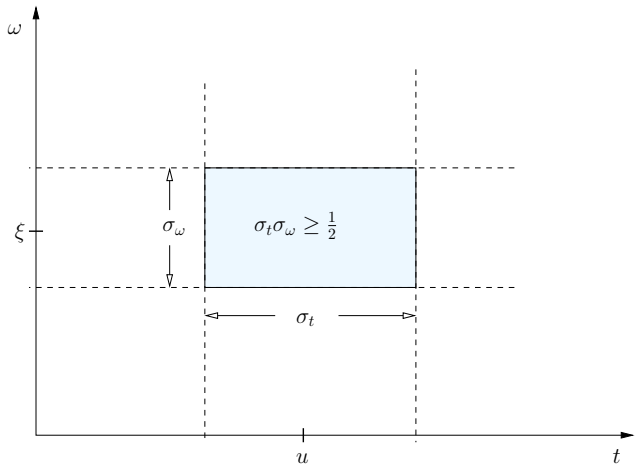


Figure A.1: The Heisenberg box for a function f.

A.4 2-D wavelet transform with two scaling parameters

As in one-dimension we define the 2-D wavelet in such a way that dilations of the 2-D wavelet cover the Fourier plane and a reconstruction formula exists.

Definition A.8. Let $\Psi = \{\psi^k\}_{k=1}^n$ be a set of functions $\psi^k(x,y) \in L^2\left(\mathbb{R}^2\right) \cap L^1(\mathbb{R}^2)$. Ψ is called a 2-D wavelet if there exists constants $A > 0, B < \infty \in \mathbb{R}$ such that

$$A \leq C_\Psi = \int_0^\infty \int_0^\infty \sum_{k=1}^n \frac{|\hat{\psi}(\omega_x, \omega_y)|^2}{\omega_x \omega_y} d\omega_x d\omega_y \leq B \tag{A.11}$$

In this case (A.11) are called *the 2-D admissibillity condition.* A 2-D wavelet is a set of functions such that their dilations by (s_x, s_y) togheter cover the Fourier plane and translations by (u_x, u_y) cover $\mathbb{R}^2$.

Definition A.9. Let $f(x,y) \in L^2\left(\mathbb{R}^2\right) \cap L^1(\mathbb{R}^2)$ and Ψ a 2-D wavelet. The 2-D wavelet tranform of f with respect Ψ is the set of functions

$$\mathbf{W}f(u,v,s_u,s_v) = \{W^k f(u,v,s_u,s_v)\}_{k=1}^n \tag{A.12}$$

with

$$W^k f(u,v,s_u,s_v)\}_{k=1}^n = < f, \psi^k_{u,v,(s_u,s_v)} > \tag{A.13}$$

Reconstruction - two scaling parameters

Proposition A.5. *Let $f \in \mathrm{L}^2\left(\mathbb{R}^2\right) \cap L^1(\mathbb{R}^2)$, Ψ be a 2-D wavelet and $\mathbf{W}f(u,v,s)$ be the 2-D wavelet transform of f with respect to Ψ. Then*

$$f(x,y) = \frac{1}{C_\Psi} \int_0^\infty \int_0^\infty \sum_{k=1}^n \left[\int_{-\infty}^\infty \int_{-\infty}^\infty W^k f(u,v,s_u,s_v) \psi^k_{u,v,(s_u,s_v)}(x,y) du dv \right] \frac{ds_u ds_v}{s_u s_v} \tag{A.14}$$

with

$$C_\Psi = \int_0^\infty \int_0^\infty \sum_{k=1}^n \frac{|\hat{\psi}(\omega_x, \omega_y)|^2}{\omega_x \omega_y} d\omega_x d\omega_y \tag{A.15}$$

Proof. We start with the function

$$J^k(x,y,s_u,s_v) = \int_{-\infty}^\infty \int_{-\infty}^\infty W^k f(u,v,s_u,s_v) \psi^k_{u,v,s_u,s_v}(x,y) du dv \tag{A.16}$$

Proposition 2.12 and Parseval gives

$$W^k f(u,v,s_u,s_v) =< f, \psi^k_{u,v,s_u,s_v} > \tag{A.17}$$

$$=< \hat{f}, \hat{\psi^k}_{u,v,s_u,s_v} > \tag{A.18}$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \hat{f}(\omega_x,\omega_y) e^{2i\pi(u\omega_x+v\omega_y)} \overline{\hat{\psi}^k(s_u\omega_x, s_v\omega_y)} d\omega_x d\omega_y \tag{A.19}$$

$$= F^{-1}[\hat{f}(\omega_x,\omega_y)\overline{\hat{\psi}^k(s_u\omega_x, s_v\omega_y)}](u,v) \tag{A.20}$$

The we can write (A.16) as

$$J^k(x,y,s_u,s_v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F^{-1}[\hat{f}(\omega_x,\omega_y)\overline{\hat{\psi}^k(s_u\omega_x, s_v\omega_y)}](u,v)\psi_{(s_u,s_v)}(u-x,v-y)dudv \tag{A.21}$$

$$=< F^{-1}[\hat{f}(\omega_x,\omega_y)\overline{\hat{\psi}^k(s_u\omega_x, s_v\omega_y)}](u,v), \overline{\psi_{(s_u,s_v)}(u-x,v-y)} > \tag{A.22}$$

Using Parseval and Proposition 2.12 again we get

$$J^k(x,y,s_u,s_v) =< \hat{f}(\omega_x,\omega_y)\overline{\hat{\psi}^k(s_u\omega_x, s_v\omega_y)}, e^{-2i\pi(x\omega_x+y\omega_y)} F^{-1}[\overline{\psi(\tau_x,\tau_y)}](s\omega_x, s\omega_y) > \tag{A.23}$$

Assuming $\psi^k(x,y)$ is real we get

$$J^k(x,y,s_u,s_v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \hat{f}(\omega_x,\omega_y) e^{2i\pi(x\omega_x+y\omega_y)} |\hat{\psi}^k(s_u\omega_x, s_v\omega_y)|^2 d\omega_x d\omega_y \tag{A.24}$$

Define the function

$$g(x,y) = \frac{1}{C_\Psi}\int_0^{\infty}\int_0^{\infty}\sum_{k=1}^{n} J^k(x,y,s_u,s_v)\frac{ds_u ds_v}{s_u s_v} \tag{A.25}$$

By interchange of integration and sum we get

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \hat{f}(\omega_x,\omega_y) e^{2i\pi(x\omega_x+y\omega_y)} \left[\int_0^{\infty}\int_0^{\infty}\sum_{k=1}^{n} \frac{|\hat{\psi}^k(s_u\omega_x, s_v\omega_y)|^2}{s_u s_v} ds_u ds_v\right] d\omega_x d\omega_y \tag{A.26}$$

$$= \frac{1}{C_\Psi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \hat{f}(\omega_x,\omega_y) e^{2i\pi(x\omega_x+y\omega_y)} \left[\int_0^{\infty}\int_0^{\infty}\sum_{k=1}^{n} \frac{|\hat{\psi}^k(\tau_x,\tau_y)|^2}{\tau_x\tau_y} d\tau_x d\tau_y\right] d\omega_x d\omega_y \tag{A.27}$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \hat{f}(\omega_x,\omega_y) e^{2i\pi(x\omega_x+y\omega_y)} d\omega_x d\omega_y \tag{A.28}$$

$$= f(x,y) \tag{A.29}$$

□

Bibliography

[1] Anestis Antoniadis and George Oppenheim, editors. *Lecture Notes in Statistics: Wavelet and Statistics.* Springer-Verlag, Berlin, 1995.

[2] C. Gasquet and P. Witomski. *Fourier Analysis and Applications.* Springer, 1998.

[3] John Canny. A Computational Approach to Edge Detection. *IEEE TPAMI*, 8(6), November 1986.

[4] Karlheinz Gröchenig. Irregular Sampling of Wavelet and Short-time Fourier Transforms. *Springer-Verlag New York, Constructive Approximation*, 9, 1993.

[5] Robert A. Adams. *A Complete Course Calculus Fourth Edition.* Ron Doleman, 1999.

[6] Stephane Mallat. *a Wavelet Tour Of Signal Processing Second Edition.* Academic Press, 1999.

[7] Stephane Mallat and Sifen Zhong. Characterization of Signals from Multiscale Edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(7), July 1992.

[8] Stephane Mallat and Wen Liang Hwang. Singularity Detection and Processing with Wavelets. *IEEE Transactions On Information Theory*, 38(2), March 1992.

[9] Zoran Cvetkovic and Martin Vetterli. Discrete-time Wavelet Extrema Representation: Design and Consistent Reconstruction. *IEEE Transactions on Signal Processing*, 43(3), March 1995.