NTNU
Norwegian University of
Science and Technology

Image recognition performed on handwritten letters using the windowed scattering transform

Svend-Peder Oseth

Master of Science in Physics and Mathematics
Submission date: June 2017
Supervisor: Yurii Lyubarskii, IMF

Norwegian University of Science and Technology
Department of Mathematical Sciences

Abstract

The windowed scattering transform is an operator that is invariant to small translations, deformations and rotations. The transform can be used in conjunction with a classification algorithm to perform image recognition. This thesis consists of one theoretical part and one numerical part. In the theoretical part the underlying theory of the windowed scattering transform, namely Fourier analysis and wavelets, is briefly introduced. Then, the construction of the windowed scattering transform and its numerical approximation is explained in detail. The numerical part consists of examples showcasing the properties of the transform, and the transform applied in image recognition on a dataset of handwritten letters. An error rate of 10.2% was achieved, using the k-nearest neighbors algorithm for classification. The error rate is high compared to other more sophisticated image recognition procedures. Most of the errors stem from inaccurate classification on classes with few samples, and from incorrect classifications on letters that are similar in shape. Some suggestions are given on how the error rates could be improved in further work.

Sammendrag

Windowed scattering transform er en operator som er invariant til små translasjoner, deformasjoner og rotasjoner. Transformen kan bli brukt i kombinasjon med en klassifikasjonsalgoritme for å utføre bildegjenkjenning. Denne masteroppgaven inneholder en teoretisk del og en numerisk del. I den teoretiske delen forklares teorien som ligger bak windowed scattering transform, i hovedsak Fourieranalyse og wavelets. Deretter konstrueres transformen og dens numeriske approksimasjon i detalj. Den numeriske delen består av eksempler som belyser transformens egenskaper, og transformen anvendes i bildegjenkjenning på et datasett av håndskrevne bokstaver. Bokstavgjenkjenningen hadde en feilrate på 10.2%. Denne feilraten er høy sammenlignet med andre mer sofistikerte bildegjenkjenningsprosedyrer. Mesteparten av feilene stammer fra feilaktige klassifikasjoner på klasser med få bilder, og fra forveksling mellom bokstaver som er lik i form. Avslutningsvis gis noen forslag til hvordan feilraten kan forbedres i fremtidig arbeid.

Preface

This thesis marks the completion of my five-year master degree in physics and mathematics with specialization in industrial mathematics. The thesis counts for 30 ECTS credits, which corresponds to one semester worth of work.

A project on the same topic, worth 15 ECTS credits, was completed the previous semester. That project served as an introductory study for this master thesis. Professor Yurii Lyubarskii was the supervisor for both this thesis and the previous project.

The main topic for this thesis is a technique called the windowed scattering transform. Studying the technique and its underlying theory was one of the main endeavors of this thesis. Also, a considerable amount of time and effort was dedicated to implementing the technique from scratch, involving a lot of trial and error.

I would like to thank my supervisor, Professor Yurii Lyubarskii. His encouragement and patience have been invaluable. I would also like to thank Jan Gulla, he supplied the latex template for this thesis.

Svend-Peder Oseth
Trondheim, Norway
June 2017

Contents

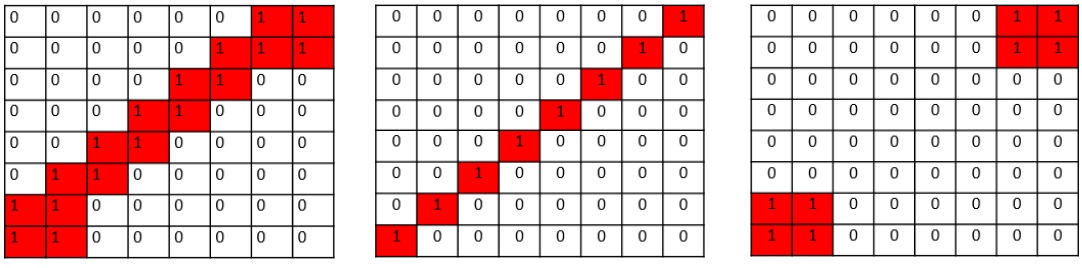
List of Figures

List of Tables

Chapter 1

Introduction

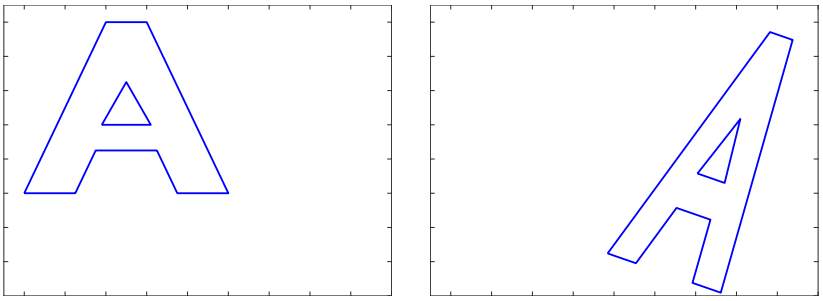
In this thesis, we will study wavelets and a technique called the windowed scattering transform. The transform will be used along with the classification algorithm k-nearest neighbors to perform image recognition on handwritten letters. Machines analyze images by looking at individual pixels. However, recognizing patterns and structures in images by comparing individual pixels only, will not yield convincing results. A simple example, shown in Figure 1.1, illustrates that a matching number of pixels between images does not imply that the images are depicting the same pattern. Both Figure 1.1a and 1.1b depict a diagonal line, while Figure 1.1c does not. When comparing Figure 1.1a with Figure 1.1b and Figure 1.1c by looking at individual pixels, they have the same number of matching pixels. More sophisticated algorithms are needed for a machine to analyze images on a larger scale in order for it to recognize patterns and structures.



(a) (b) (c)

Figure 1.1: Images (a) and (b) have the same pattern, but images (b) and (c) have an equal amount of matching pixels when compared to image (a).

For example, two images of the same object, as shown in Figure 1.2. The object in the second image is translated, deformed and rotated compared to the object in the first image. Humans can easily recognize that the two images depict the same object, but for a machine, this is a difficult task.



(a) (b)

Figure 1.2: Image (a) depicts the letter A. Image (b) depicts the same A subjected to translation, deformation and rotation.

An even more challenging task in image recognition is to recognize the same object from different angles. Figure 1.3 shows the same airplane from three different angles. Humans will recognize that the three images depict an airplane, but a machine might not. Noise, blur and light variations are other effects that could make image recognition more difficult. These effects can be analyzed using standard signal processing tools, which machines can do better than humans.

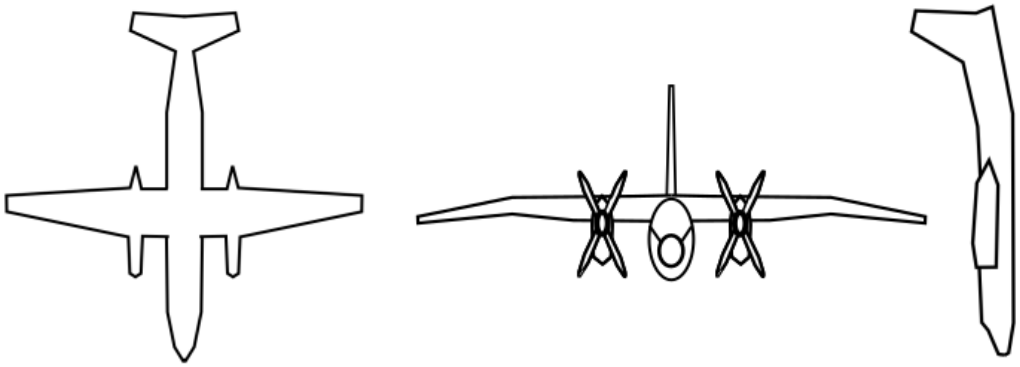


Figure 1.3: Images of the same airplane from three different angles.

A standard approach when performing image recognition is to use artificial neural networks, which is a technique developed in the field of artificial intelligence and machine learning. For a more in-depth look at artificial neural networks, see for example one of the many textbooks on machine learning [14]. In image recognition, an artificial neural network is trained by allowing the network to analyze datasets containing classified images. The network can be trained on the dataset by having it attempt to classify the images in the dataset, and have the network continuously adjust itself based on the correct classification of the image. Eventually, the network could learn the patterns which differentiate the objects from one another. This would make the network capable of classifying images of the same objects which were not in the dataset. The windowed scattering technique was developed by Mallat [13] as a straightforward alternative to artificial neural networks.

In this thesis, we restrict ourselves to identifying translated, deformed and rotated objects. More challenging tasks, like recognizing the same object from different angles will not be attempted. Noise, blur or light variations will not be considered either. Because of this, handwritten letters were chosen as the objects to classify in this thesis. The windowed scattering transform can recognize translated, deformed and rotated objects and it will be applied to a dataset of handwritten letters [9]. Letters were chosen as the objects to be classified because digits and letters are standard choices in image recognition. Choosing simple letters will allow us to analyze the windowed scattering transforms main features. Furthermore, choosing letters is beneficial because it provides several options. The task can be made easier or more difficult depending on the letters chosen. For example, recognizing letters written in different fonts is easy compared to recognizing handwritten letters.

Images are two-dimensional signals. However, in this thesis, a lot of the theory and examples will be presented in the one-dimensional case. Chapter 2 will introduce definitions and theorems from Fourier analysis that will be used in later chapters. Chapter 3 will present a more in-depth look at wavelets and the wavelet transform, which are the building blocks of the windowed scattering transform. Chapter 4 is the main theoretical part of this thesis and will define the windowed scattering transform. In Chapter 5, numerical examples showcase the windowed scattering transform and its properties. All results in Chapter 5 come from the author's implementation of the windowed scattering transform. In Chapter 6, the windowed scattering transform will be used to perform image recognition on handwritten letters. In Chapter 6 the Matlab toolbox Scatnet [15] was used to compute the windowed scattering transform, because of its improved runtime over the author's implementation. Finally, a conclusion is provided in the form of a summary, and future work is discussed.

Chapter 2

Fourier transform

In this thesis, we will restrict ourselves to one and two-dimensional functions, which corresponds to audio signals and images respectively. An example of a one-dimensional signal is an audio signal $f(t)$ measured at a fixed point in space. For this signal, the amplitude f depends on the time t. If the amplitude of a sound is too low, our eardrums will not be able to sense the fluctuating air pressure, and we cannot hear the sound. Furthermore, if the amplitude is too high, we cannot hear the sound because the high air pressure would destroy our eardrums. Limits on the amplitude range humans can sense are not the only restriction on what sound signals we can perceive. Humans are only able to perceive sounds which include frequencies between 20Hz and 20kHz [7]. These two limits correspond to two different representations of the same signal. A signal may be represented in time-space as $f(t)$, or in frequency-space where the signal is denoted $\hat{f}$ and is a function of the frequency ω. The function $\hat{f}$ is called the Fourier transform of f, and in one dimension, it is defined as follows.

Definition 2.1. The *Fourier transform* of a signal $f(t)$ is given by

$$\hat{f}(w) = \int_{-\infty}^{\infty} f(t)\, e^{-iwt}\, dt. \tag{2.1}$$

An example of a two-dimensional signal is an image $f(x, y)$. For this signal, the amplitude $f(x, y)$ depends on the position (x, y). For greyscale images f typically takes values between a lower and an upper bound, where the lower bound corresponds to black, the upper bound corresponds to white and the values in between give different tones of gray. The function $f(x, y)$ can also be represented in frequency space as $\hat{f}(u, v)$ where u and v are spatial frequencies. The two-dimensional Fourier transform is defined as follows.

Definition 2.2. The *Fourier transform* of a signal $f(x, y)$ is given by

$$\hat{f}(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)\, e^{-i(xu+yv)}\, dx\, dy. \tag{2.2}$$

In this thesis three different representations of signals will be used, the Fourier transform, the wavelet transform and the windowed scattering transform. For these representations to be meaningful, the transforms have to be stable. Stability means that a small change in the signal will produce a small change in the transform. Take a signal and add two different noise variations to it. These two signals will not be that different. If a transform lacks stability, the difference between the same transform of those two signals could be huge. Stability ensures that the same transform of two similar signals are indeed close to each other.

Definition 2.3. The *energy of a signal* f is equal to the L^2-norm of the signal squared, that is $\|f\|_2^2$.

Energy is useful because it is closely related to stability. We say that a transform preserves energy if the energy of a signal is equal to the energy of the transform. If a transform preserves energy, then the transformation is stable. It the transform is stable, then the transformation preserves the energy of the signal up to a constant. The following theorem will prove that the Fourier transform preserves energy up to a constant, and thus is stable. For the rest of this chapter, only the one-dimensional case will be discussed, but all results can be generalized to two dimensions.

Theorem 2.4. *For signals* $f, g \in L^1(\mathbb{R}) \cap L^2(\mathbb{R})$ *we have that*

$$\langle f, g \rangle = \frac{1}{2\pi} \langle \hat{f}, \hat{g} \rangle,$$

which is known as Parseval's formula. If $f = g$ *we get Plancherel's formula*

$$\|f\|_2^2 = \frac{1}{2\pi} \left\| \hat{f} \right\|_2^2. \tag{2.3}$$

Proof. See Reference [12].

The Fourier transform is a powerful tool, but it has some drawbacks. The Fourier transform gives information about what frequencies are present in a signal, but the Fourier Transform does not give any information as to at what time the different frequencies appear. A tool that can be used to solve this problem is the windowed Fourier transform.

Definition 2.5. Let the window g be any bounded function with finite support. The *Windowed Fourier transform* of a signal $f \in L^2(\mathbb{R})$ at time u and frequency ξ is given by

$$Sf(u,\xi) = \langle f(t), g(t-u)e^{it\xi} \rangle = \int_{-\infty}^{\infty} f(t)\, g^*(t-u)\, e^{-it\xi}\, dt, \tag{2.4}$$

where $g^*(t-u)$ is the complex conjugate of g.

The Fourier transform has exact frequency information, but no information as to at what time those frequencies appear. On the other hand, the windowed Fourier transform has time information, but loses exact frequency information. As per Heisenberg's uncertainty principle [12], there is a limit on the time and frequency resolution.

For different applications of the windowed Fourier transform, there are many possible choices of windows. We will now introduce Heisenberg boxes which may be used to compare windows, and in Chapter 3 the boxes will be used to compare the windowed Fourier transform and the wavelet transform.

Definition 2.6. Let $\{\zeta_\gamma\}_{\gamma\in\Gamma}$ be a famility of functions $\zeta_\gamma \in L^2(\mathbb{R})$ with $\|\zeta_\gamma\|_2 = 1$. Then for any $f \in L^2(\mathbb{R})$ we have a general time-frequency transform defined by

$$Tf = \langle f, \zeta_\gamma \rangle = \int_{-\infty}^{\infty} f(t)\zeta_\gamma^*(t)\, dt.$$

The time-frequency resolution of Tf can be represented in the time-frequency domain (t,ω) by a box whose position and width depend on the chosen function ζ_γ. This box is called a *Heisenberg box*. The expressions $|\zeta_\gamma(t)|^2$ and $\frac{1}{2\pi}|\hat{\zeta}_\gamma(t)|^2$ can be interpreted as probability density functions because

$$\int_{-\infty}^{\infty} |\zeta_\gamma(t)|^2\, dt = \|\zeta_\gamma\|_2^2 = 1 \quad \text{and} \quad \frac{1}{2\pi}\int_{-\infty}^{\infty} |\hat{\zeta}_\gamma(\omega)|^2\, \omega = \frac{1}{2\pi}\left\|\hat{\zeta}_\gamma\right\|_2^2 = \|\zeta_\gamma\|_2^2 = 1,$$

where we have used Plancherel's formula (2.3). From the probability density interpretation, we get that the position of the Heisenberg box is given by the mean values of ζ_γ and $\hat{\zeta}_\gamma$, that is

$$E_t(\gamma) = \int_{-\infty}^{\infty} t\, |\zeta_\gamma(t)|^2\, dt \quad \text{and} \quad E_\omega(\gamma) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \omega\, |\zeta_\gamma(\omega)|^2\, d\omega.$$

Also, the widths $(\sigma_t, \sigma_\omega)$ of the box is given by the variances of ζ_γ and $\hat{\zeta}_\gamma$, that is

$$\sigma_t^2(\gamma) = \int_{-\infty}^{\infty} (t - E_t)^2 \, |\zeta_\gamma(t)|^2 \, dt \quad \text{and} \quad \sigma_\omega^2(\gamma) = \frac{1}{2\pi} \int_{-\infty}^{\infty} (\omega - E_\omega)^2 \, |\zeta_\gamma(\omega)|^2 \, d\omega.$$

Remark 2.7 . Heisenberg's uncertainty principle [12] gives that $\sigma_t \sigma_\omega \geq 1/2$. Poor choice of window gives Heisenberg boxes with an area larger than $1/2$. It can be shown that using Gaussian windows, that is ζ_γ equal the Gaussian distribution, gives Heisenberg boxes with area $\sigma_t \sigma_\omega = 1/2$.

Let us compute the Heisenberg boxes of the windowed Fourier transform for a real and symmetric window g. Then the Heisenberg box of $g_{u,\xi}(t) = g(t-u)e^{i\xi t}$ is centered at $(E_t, E_\omega) = (u, \xi)$. The width $(\sigma_t, \sigma_\omega)$ of the box is given by

$$\begin{aligned} \sigma_t^2 &= \int_{-\infty}^{\infty} (t - E_t)^2 \, |g_{u,\xi}(t)|^2 \, dt = \int_{-\infty}^{\infty} t^2 \, |g(t)|^2 \, dt \quad \text{and} \\ \sigma_\omega^2 &= \int_{-\infty}^{\infty} (\omega - E_\omega)^2 \, |\hat{g}_{u,\xi}(\omega)|^2 \, d\omega = \int_{-\infty}^{\infty} \omega^2 \, |\hat{g}(\omega)|^2 \, d\omega. \end{aligned}$$

Notice that both σ_t and σ_ω do not depend on u and ξ, which implies that the time-frequency resolution is the same for all positions and frequencies. An illustration of the boxes is given in Figure 2.1. The width of the boxes is the time resolution, and the height of the boxes is the frequency resolution. Since all the Heisenberg boxes have the same size, the resolution is constant and does not depend on the chosen u and ξ.

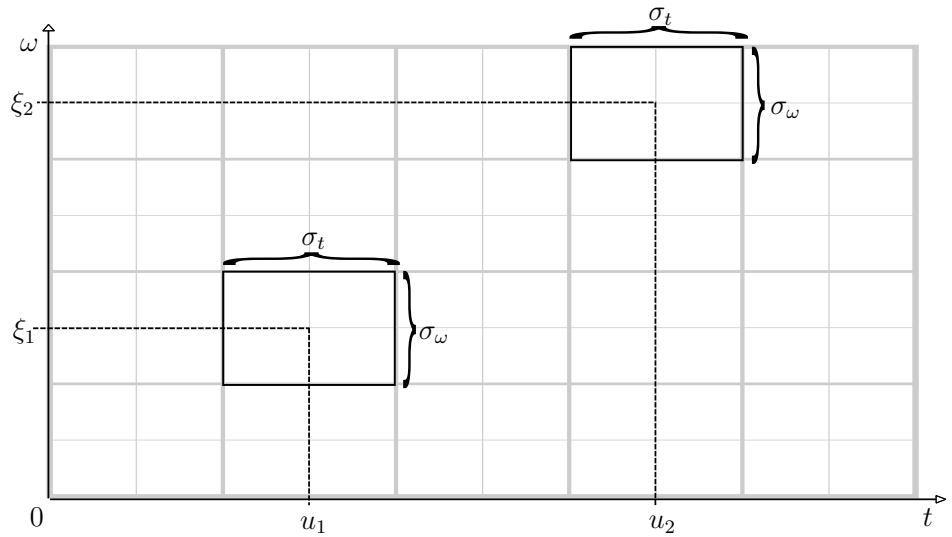


Figure 2.1: Time-frequency resolution of the windowed Fourier transform.

Chapter 3

Wavelet transform

In the previous chapter, the Fourier transform and the windowed Fourier transform were defined. These two transforms are examples of time-frequency transforms. In this chapter, we will define another time-frequency transform, namely the wavelet transform. Similarly to the windowed Fourier transform (2.4), the wavelet transform is localized both in time and frequency, and there is once again a bound on the resolution given by Heisenberg's uncertainty principle [12]. The advantage of the wavelet transform over the windowed Fourier transform is that it may offer different time-frequency resolution for different times and frequencies. The windowed Fourier transform has fixed time-frequency resolution for all times and frequencies, while the wavelet transform can have good time resolution for high frequencies and good frequency resolution for low frequencies. This property is illustrated in Figure 3.2. Wavelets and the wavelet transform will be defined in one and two dimensions in Section 3.1 and 3.2 respectively.

3.1 One-dimensional wavelets

One-dimensional wavelets are wave-like oscillations, but in contrast to sine and cosine waves whose support is infinite, wavelets decay rapidly to zero and have finite support. Sharp transitions and rapid changes are not well modeled by plane waves, but wavelets which are localized in time and frequency, are better suited to model signals that have these features.

Definition 3.1. Let ψ be the one-dimensional function $\psi \in L^1(\mathbb{R}) \cap L^2(\mathbb{R})$. We define the constant

$$C_\psi = \int_0^\infty \frac{|\hat{\psi}(w)|^2}{w}\, dw.$$

The function $\psi(t)$ is a *wavelet* if it satisfies the *admissibility condition*

$$0 < C_\psi < \infty. \tag{3.1}$$

Satisfying the admissibility condition implies that the wavelet is smooth, centered at $t = 0$ and has zero average

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

It is common to normalize wavelets such that $\|\psi\|_2 = 1$.

Example 3.1. An example of a wavelet is the *Mexican hat wavelet*. It is the second derivative of the Gaussian distribution, and is given by

$$\psi(t) = \frac{2}{\pi^{1/4}\sqrt{3\sigma}} \left(\frac{t^2}{\sigma^2} - 1\right) \exp\left(\frac{-t^2}{2\sigma}\right).$$

A possible application of the Mexican hat wavelet is in signal processing, where it may be used to detect sharp transitions and edges [12]. For $\sigma = 1$, the wavelet $-\psi(t)$ is illustrated in Figure 3.1.

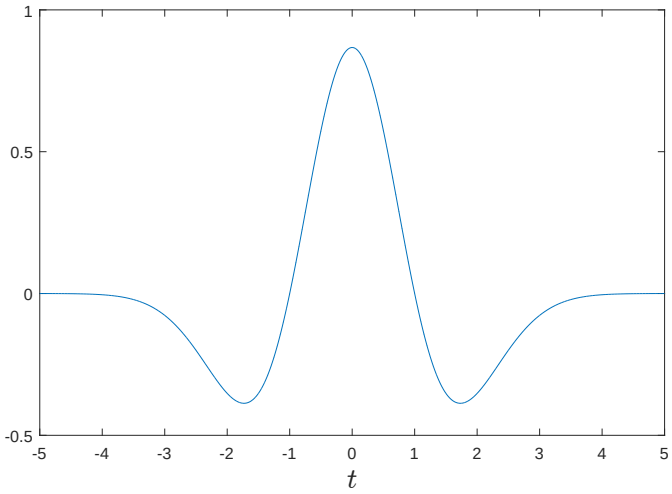


Figure 3.1: Illustration of the Mexican hat wavelet.

Before defining the wavelet transform, we introduce some notation. A wavelet that is translated by $u \in \mathbb{R}$ and scaled by $s \in \mathbb{R}_{>0}$ is denoted $\psi_{u,s}$ and is given by

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

where the constant $s^{-1/2}$ has been chosen such that $\|\psi_{u,s}\|_2 = 1$.

Definition 3.2. The *wavelet transform* of a signal $f \in L^2(\mathbb{R})$ at time $u \in \mathbb{R}$ and scale $s \in \mathbb{R}_{>0}$ is given by

$$Wf(u,s) = \langle f, \psi_{u,s} \rangle = \int_{-\infty}^{\infty} f(t) \, \frac{1}{\sqrt{s}} \, \psi^* \left(\frac{t-u}{s} \right) dt. \tag{3.2}$$

where $\psi^*(t)$ is the complex conjugate of $\psi(t)$.

Using Definition 2.6, we compute the Heisenberg boxes for a scaled and translated wavelet $\psi_{u,s}$. The box will be centered at $(E_t, E_\omega) = (u, \eta)$, where η is the center frequency of $\psi_{u,s}$. The widths of the box for $\psi_{u,s}$ are $s^2\sigma_t$ and $s^{-2}\sigma_\omega$, where σ_t and σ_ω are the widths of the box for a non-translated and non-scaled wavelet ψ. An illustration of the two boxes for two different scales s_1 and s_2 are depicted in Figure 3.2. The trade off in resolution for the wavelet transform corresponds to the chosen scale s, this is an advantage of the wavelet transform over the windowed Fourier transform (2.4). How the chosen parameters affect its resolution is more hidden for the windowed Fourier transform.

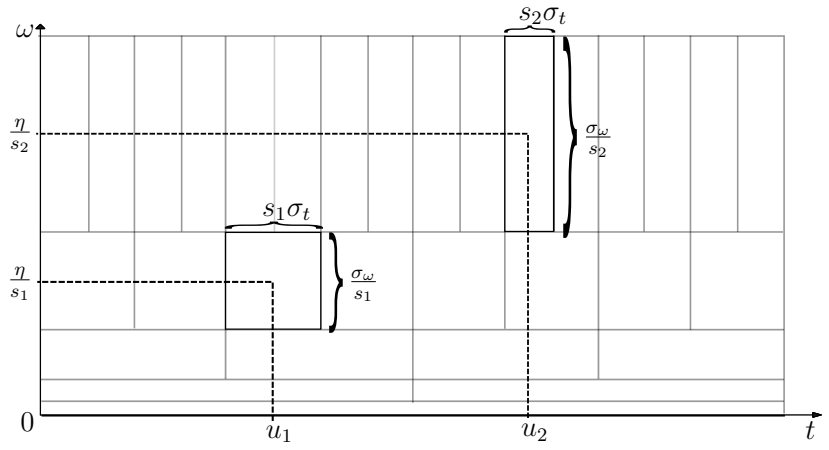


Figure 3.2: Time-frequency resolution of the wavelet transform.

The admissibility condition (3.1) ensures that the energy of a signal is equal to the energy of its wavelet transform, up to a constant. As already discussed in Chapter 2, this energy conservation property ensures that the wavelet transform is stable. The admissibility condition also ensures that a function can be recovered from its wavelet transform. Both of these properties will be proved in the next theorem.

Theorem 3.3. *For all $f \in L^2(\mathbb{R})$, if $\psi \in L^2(\mathbb{R})$ is a wavelet that satisfies the admissibility condition* (3.1)*, then*

$$\int_{-\infty}^{\infty} |f(t)|^2 \, dt = \frac{1}{C_\psi} \int_0^{\infty} \int_{-\infty}^{\infty} |Wf(u,s)|^2 \, du \, \frac{ds}{s^2} \tag{3.3}$$

and

$$f(t) = \frac{1}{C_\psi} \int_0^{\infty} \int_{-\infty}^{\infty} Wf(u,s) \, \frac{1}{\sqrt{s}} \, \psi\left(\frac{t-u}{s}\right) \, du \, \frac{ds}{s^2}. \tag{3.4}$$

Proof (3.3). The Fourier transform of $Wf(u,s)$ with respect to u is $\sqrt{s}\hat{f}(w)\hat{\psi}^*(s\omega)$. Then by applying Plancherel's formula (2.3) to the right hand side of (3.3) we get that

$$\frac{1}{C_\psi} \int_0^{\infty} \int_{-\infty}^{\infty} |Wf(u,s)|^2 \, du \, \frac{ds}{s^2} = \frac{1}{2\pi C_\psi} \int_0^{\infty} \int_{-\infty}^{\infty} |\hat{f}(w)|^2 |\hat{\psi}(s\omega)|^2 \, dw \, \frac{ds}{s},$$

where we have used that $|\psi^*(s\omega)| = |\psi(s\omega)|$. Now we apply Fubini's Theorem A.2 and substitute $s' = \omega s$

$$\frac{1}{2\pi C_\psi} \int_{-\infty}^{\infty} |\hat{f}(w)|^2 \int_0^{\infty} |\hat{\psi}(s\omega)|^2 \, \frac{ds}{s} \, dw = \frac{1}{2\pi} \int_{-\infty}^{\infty} |\hat{f}(w)|^2 d\omega \, \frac{1}{C_\psi} \int_0^{\infty} \frac{|\hat{\psi}(s')|^2}{s'} \, ds',$$

we notice that C_ψ cancel out and by applying Plancherel's formula (2.3) again, we get $||f||_2^2$, which is what we wanted to prove. □

Proof (3.4). Let $r(t)$ denote the right hand side of (3.4). First notice that the innermost integral of $r(t)$ can be written as a convolution.

$$\int_{-\infty}^{\infty} Wf(u,s) \, \frac{1}{\sqrt{s}} \, \psi\left(\frac{t-u}{s}\right) \, du = (Wf(s) * \psi_s)(t).$$

The wavelet transform, Equation (3.2), may be rewritten as a convolution.

$$Wf(u,s) = (f * \bar{\psi}_s)(u) \quad \text{with} \quad \bar{\psi}_s(u) = \frac{1}{\sqrt{s}} \, \psi^*\left(\frac{-u}{s}\right).$$

Inserting these two results into into $r(t)$ gives

$$r(t) = \frac{1}{C_\psi} \int_0^{\infty} (Wf(s) * \psi_s)(t) \, \frac{ds}{s^2} = \frac{1}{C_\psi} \int_0^{\infty} (f * \bar{\psi}_s * \psi_s)(t) \, \frac{ds}{s^2}.$$

If the Fourier transform of $r(t)$ is equal the Fourier transform of $f(t)$ then $f(t) = r(t)$. Let us compute the Fourier transform of $r(t)$.

$$\hat{r}(\omega) = \frac{1}{C_\psi} \int_0^{\infty} \hat{f}(\omega) \; \sqrt{s} \, \hat{\psi}^*(s\omega) \; \sqrt{s} \, \hat{\psi}(s\omega) \, \frac{ds}{s^2} = \frac{\hat{f}(\omega)}{C_\psi} \int_0^{\infty} |\hat{\psi}(s\omega)|^2 \, \frac{ds}{s}$$

Using the substitution $\xi = s\omega$, we recognize the last integral as the constant C_ψ and we get $\hat{r} = \hat{f}$. $\square$

We will now define the Littlewood-Paley wavelet transform. In Section 4.4 the Littlewood-Paley wavelet transform will be used instead of the standard wavelet transform (3.2) to define the windowed scattering transform.

Definition 3.4. Let $J \in \mathbb{Z}$. For a wavelet $\psi(t)$, let $\psi_j(t) = 2^{-j}\psi(2^{-j}t)$, where the constant 2^{-j} has been chosen such that $\|\psi\|_1 = \|\psi_j\|_1 = 1$. The *Littlewood-Paley wavelet transform* of a signal $f \in L^2(\mathbb{R})$ at scale 2^j is given by

$$W[j]f(t) = (f * \psi_j)(t) = \int_{-\infty}^{\infty} f(u)\, 2^{-j}\, \psi(\, 2^{-j}\, (t-u))\, du. \tag{3.5}$$

Notice that for the Littlewood-Paley wavelet transform the scales s are discrete and dyadic, $s = 2^j$ with $j \in \mathbb{Z}$, as opposed to the standard wavelet transform (3.2), where the scales were continuous. This shift from continuous scales to discrete scales is necessary when implementing the windowed scattering transform numerically (4.1) and dyadic scales is the natural choice of discretization.

The Littlewood-Paley wavelet transform also differs from the standard wavelet transform (3.2) in that it has been normalized with respect to the L^1-norm, while previously, all wavelets were normalized with respect to the L^2-norm. This change was made because it will be convenient in calculations and will among other things be used in the next proposition. The next proposition proves that if f has finite energy, then $W[j]f$ has finite energy as well.

Proposition 3.5. *If $f \in L^2(\mathbb{R})$, then $\|W[j]f\|_2 \leq \|f\|_2 \|\psi_j\|_1 = \|f\|_2 \|\psi\|_1$, which implies that $W[j]f \in L^2(\mathbb{R})$.*

Proof.

$$\|W[j]f\|_2 = \|\psi_j * f\|_2 = \left\| \int_{-\infty}^{\infty} f(x-t)\, \psi_j(t)\, dt \right\|_2.$$

Rewriting the integral as a Riemann sum gives

$$\|W[j]f\|_2 = \left\| \sum_{i=-\infty}^{\infty} \Delta\, f(x-t_i)\, \psi_j(t_i) \right\|_2,$$

where $\Delta = t_i - t_{i-1}$ and has the same value for all i. Now using Minkowski's inequality A.5, and noting that $||f(x - t_i)||_2 = ||f(x)||_2$, we get

$$\begin{aligned}\|W[j]f\|_2 &\leq \sum_{i=-\infty}^{\infty} \Delta\, \psi_j(t_i)\, \|f(x - t_i)\|_2 \\ &= \|f(x)\|_2 \sum_{i=-\infty}^{\infty} \Delta\, \psi_j(t_i) \\ &= \|f\|_2\, \|\psi_j\|_1 = \|f\|_2\, \|\psi\|_1. \quad \square\end{aligned}$$

3.2 Two-dimensional wavelets

Definition 3.6. Let ψ be the two-dimensional function $\psi \in L^1(\mathbb{R}^2) \cap L^2(\mathbb{R}^2)$. Using the same notation as for one-dimensional wavelets, we define the constant

$$C_\psi = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\omega_1, \omega_2)|^2}{\|(\omega_1, \omega_2)\|^2}\, d\omega_1\, d\omega_2.$$

Whenever the constant C_ψ is used, it should be clear from context whether it is the one- or two-dimensional version. The function $\psi(x, y)$ is a *wavelet* if it satisfies the *admissibility condition*

$$0 < C_\psi < \infty. \tag{3.6}$$

Satisfying the admissibility condition implies that the wavelet is smooth, centered at $(x, y) = (0, 0)$ and has zero average

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \psi(x, y)\, dx\, dy = 0.$$

As for the one-dimensional wavelets, two-dimensional wavelets are also normalized such that $||\psi||_2 = 1$.

For one-dimensional wavelets we considered translation and scaling. In two dimensions rotation can be considered as well. A wavelet that is translated by $(u, v) \in \mathbb{R}^2$, scaled by $s \in \mathbb{R}_{>0}$ and rotated by $\theta \in [0, 2\pi)$ is denoted $\psi_{u,v,s,\theta}$ and is given by

$$\psi_{u,v,s,\theta}(x, y) = s^{-1}\, \psi\left(r_\theta^{-1}\left(\frac{(x, y) - (u, v)}{s}\right)\right),$$

where the constant s^{-1} has been chosen such that $\|\psi_{\bar{u},s,\theta}\|_2 = 1$ and r_θ^{-1} is the inverse of the rotation matrix defined by

$$r_\theta = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}.$$

Definition 3.7. The *wavelet transform* of a signal $f \in L^2(\mathbb{R}^2)$ at position $(u, v) \in \mathbb{R}^2$, scale $s \in \mathbb{R}_{>0}$ and rotation $\theta \in [0, 2\pi)$ is given by

$$\begin{aligned} Wf(u, v, s, \theta) &= \langle f, \psi_{u,v,s,\theta} \rangle \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)\, s^{-1}\, \psi^* \left(r_\theta^{-1} \left(\frac{(x, y) - (u, v)}{s} \right) \right) dx\, dy, \end{aligned} \tag{3.7}$$

where $\psi^*(t)$ is the complex conjugate of $\psi(t)$.

Energy preservation and stability of the wavelet transform in two dimensions are proved by extending the proof of Theorem 3.3 to two dimensions. The proofs are similar except for the integral substitution, but the proofs have nonetheless been included for the reader's convenience and the sake of completeness.

Theorem 3.8. *For all $f \in L^2(\mathbb{R}^2)$, if $\psi \in L^2(\mathbb{R}^2)$ satisfies the admissibility condition* (3.6)*, then*

$$\int_{-\infty}^{\infty} |f(x, y)|^2\, dx\, dy = \frac{1}{C_\psi} \int_0^{2\pi} \int_0^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |Wf(u, v, s, \theta)|^2\, du\, dv\, \frac{ds}{s^3}\, d\theta, \tag{3.8}$$

and

$$f(x, y) = \frac{1}{C_\psi} \int_0^{2\pi} \int_0^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} Wf(u, v, s, \theta)\ \psi_{u,v,s,\theta}(x, y)\, du\, dv\, \frac{ds}{s^3}\, d\theta. \tag{3.9}$$

Proof (3.8). The Fourier transform of $Wf(u, v, s, \theta)$ with respect to (u, v) is

$$s\hat{f}(\omega_1, \omega_2)\hat{\psi}^* \left(s r_\theta^{-1}(\omega_1, \omega_2) \right).$$

Then by applying Plancherel's formula (2.3) to the right hand side of (3.8) we get that

$$\frac{1}{2\pi C_\psi} \int_0^{2\pi} \int_0^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |f(\omega_1, \omega_2)|^2 \left| \hat{\psi} \left(s r_\theta^{-1}(\omega_1, \omega_2) \right) \right|^2 d\omega_1\, d\omega_2\, \frac{ds}{s}\, d\theta.$$

By applying Fubini's Theorem A.2 we get

$$\frac{1}{2\pi C_\psi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}|f(\omega_1,\omega_2)|^2\int_0^{2\pi}\int_0^{\infty}\left|\hat{\psi}\left(sr_\theta^{-1}(\omega_1,\omega_2)\right)\right|^2\,\frac{ds}{s}\,d\theta\,d\omega_1\,d\omega_2.$$

If we can get the inner double integral and the constant C_ψ to cancel each other out, we are done. Now let us take a closer look at the constant

$$C_\psi=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{|\hat{\psi}(\omega_1',\omega_2')|^2}{\omega_1'^2+\omega_2'^2}\,d\omega_1'\,d\omega_2',$$

where we will use the following substitution

$$\begin{aligned}\omega_1' &= \quad s\omega_1\cos(\theta)+s\omega_2\sin(\theta)\\ \omega_2' &= -s\omega_1\sin(\theta)+s\omega_2\cos(\theta).\end{aligned}$$

The Jacobian of this substitution is $J=s(w_1^2+w_2^2)$, resulting in

$$C_\psi=\int_0^{2\pi}\int_0^{\infty}\frac{|\hat{\psi}(sr_\theta^{-1}(\omega_1,\omega_2))|^2}{s^2(\omega_1^2+\omega_2^2)}s(\omega_1^2+\omega_2^2)\,ds\,d\theta.\quad\square$$

Proof (3.9). Let $r(x,y)$ denote the right hand side of (3.9). First notice that the inner double integral of $r(x,y)$ may be written as a convolution.

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}Wf(u,v,s,\theta)\;\psi_{u,v,s,\theta}(x,y)\,du\,dv=\left(Wf(s,\theta)*\psi_{s,\theta}\right)(x,y).$$

The wavelet transform, Equation (3.7), may be rewritten as a convolution.

$$Wf(u,v,s,\theta)=(f*\bar{\psi}_{s,\theta})(u,v)\quad\text{with}\quad\bar{\psi}_{s,\theta}(x,y)=\frac{1}{s}\psi^*\left(r_\theta^{-1}\left(\frac{-x}{s},\frac{-y}{s}\right)\right).$$

Inserting these two convolutions into $r(x,y)$ gives

$$r(x,y)=\frac{1}{C_\psi}\int_0^{2\pi}\int_0^{\infty}(f*\bar{\psi}_{s,\theta}*\psi_{s,\theta})(x,y)\,\frac{ds}{s^3}\,d\theta.$$

If the Fourier transform of $r(x,y)$ is equal the Fourier transform of $f(x,y)$, then $f(x,y)=r(x,y)$. Let us compute the Fourier transform of $r(x,y)$.

$$\begin{aligned}\hat{r}(\omega_1,\omega_2)&=\frac{1}{C_\psi}\int_0^{2\pi}\int_0^{\infty}\hat{f}(\omega_1,\omega_2)s\hat{\psi}^*(sr_\theta^{-1}(\omega_1,\omega_2))s\hat{\psi}(sr_\theta^{-1}(\omega_1,\omega_2))\,\frac{ds}{s^3}\,d\theta\\&=\frac{\hat{f}(\omega_1,\omega_2)}{C_\psi}\int_0^{2\pi}\int_0^{\infty}|\hat{\psi}(sr_\theta^{-1}(\omega_1,\omega_2))|^2\,\frac{ds}{s}\,.\end{aligned}$$

Using the same substitution as in the proof of (3.8) we get that the integral and the constant C_ψ cancel each other out and $\hat{r} = \hat{f}$. $\square$

Before defining the two-dimensional Littlewood-Paley wavelet transform, some notation will be introduced. Let $2^{\mathbb{Z}}$ be the set $\{2^j : j \in \mathbb{Z}\}$. The space of rotations in two dimensions is denoted $SO(2)$ and is defined as

$$SO(2) = \left\{ r_\theta = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} : \theta \in [0, 2\pi) \right\}.$$

Definition 3.9. Let $\lambda = 2^j r_\theta \in 2^{\mathbb{Z}} \times SO(2)$. For a two-dimensional wavelet $\psi(x, y)$, let $\psi_\lambda(x, y) = 2^{-2j}\psi(2^{-j}r^{-1}(x, y))$, where the constant 2^{-2j} have been chosen such that $\|\psi\|_1 = \|\psi_\lambda\|_1 = 1$. The *Littlewood-Paley wavelet transform* of a signal $f \in L^2(\mathbb{R}^2)$ at scale 2^j and rotation r is given by

$$\begin{aligned} W[\lambda]f(x, y) &= (f * \psi_\lambda)(x, y) \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f((u, v))\, 2^{-2j}\, \psi\left(\lambda^{-1}\left((x, y) - (u, v)\right)\right) du\, dv. \end{aligned} \tag{3.10}$$

Proposition 3.5 and its proof can be extended to two dimensions in order to show that for a signal with finite energy $f \in L^2(\mathbb{R}^2)$, the Littlewood-Paley wavelet transform of the signal also has finite energy $W[\lambda]f \in L^2(\mathbb{R}^2)$.

Chapter 4

Windowed scattering transform

In this thesis, the goal is to compare signals, and to measure the difference between them. Only one- and two-dimensional cases will be considered. The aim is to be able to recognize the similarity in the shape of two signals. The tool that will be used to achieve this is the windowed scattering transform, which will be defined in Section 4.4. Throughout this chapter, the theory will be discussed for the one-dimensional case, except for Section 4.6, which explains how to extend the windowed scattering transform to two dimensions.

We want to introduce a metric which has the property that the distance between two signals of similar shape is small, independent of position. As the next example will show, the L^p-norms do not have this property. Let $d_p(\cdot,\cdot)$ be the metric induced by the L^p-norm such that $d_p(f,g) = \|f-g\|_p$. Take for example the three one-dimensional signals f, g and h shown in Figure 4.1 and 4.2. The difference $d_p(f,g) = \|f-g\|_p$ is much smaller than the difference $d_p(f,h) = \|f-h\|_p$, which is not what we want to achieve. We want a metric $d(\cdot,\cdot)$ where $d(f,h)$ is smaller than $d(f,g)$, since f and g are more similar in shape than f and h. Two operations can be used to transform f into h, namely, translation and deformation which will be defined in Section 4.1 and 4.2 respectively.

To summarize, we want to recognize the similarity between a signal $f(t)$ and the shifted signal $f(t-c)$, and to recognize the similarity between a signal $f(t)$ and the slightly deformed signal $f(t-\tau(t))$. Figure 4.3 and 4.4 shows a shifted and a deformed signal respectively. The function $\tau(t)$ will be defined in Section 4.2. Since the standard L^p-norms do not have the desired properties, we need another metric. Consider the operator

$$\Phi : L^2(\mathbb{R}) \to \mathcal{H},$$

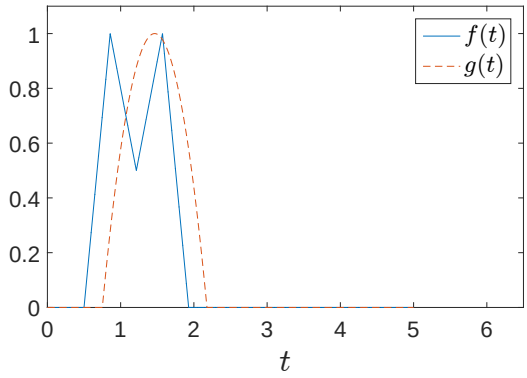


Figure 4.1: Signals f and g are different in shape, but the difference $||f - g||_p$ is small.

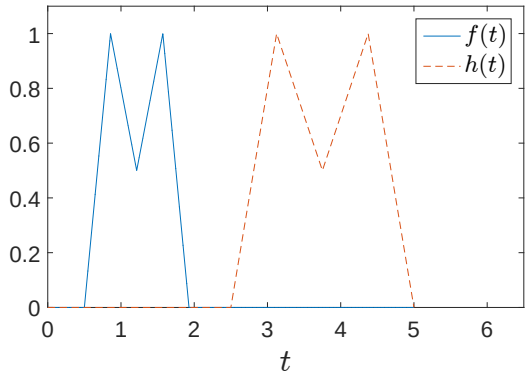


Figure 4.2: Signals f and h are similar in shape, but the difference $||f - h||_p$ is large.

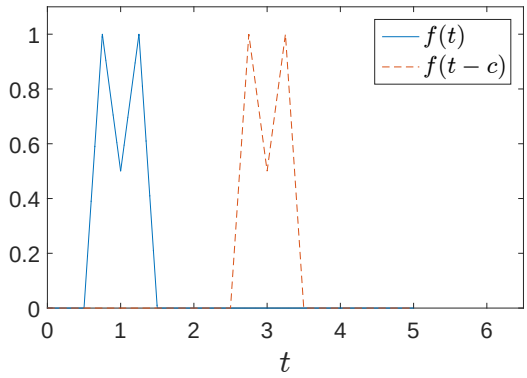


Figure 4.3: Signal f and translated signal $f(t - c)$.

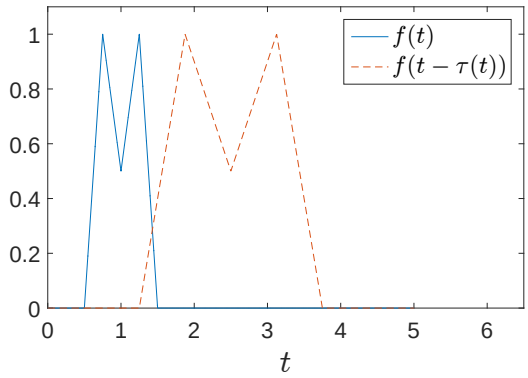


Figure 4.4: Signal f and deformed signal $f(t - \tau(t))$.

where $\mathcal{H}$ is a Hilbert space, as defined in A.3. If Φ is invariant to translation and stable to deformation, the induced metric

$$d(f, g) = \|\Phi(f) - \Phi(g)\|_{\mathcal{H}},$$

would be able to recognize the desired similarities. Furthermore, Φ should be able to distinguish different signals. Therefore we need a Φ that preserves information about all frequencies in a signal. As we will see later in this chapter, it is not very difficult to construct a Φ that is either invariant to translation or stable to deformation. The challenge is to construct a Φ that has both these properties, while retaining information about all frequencies.

4.1 Translation invariance

The translation operator $T_c : L^2(\mathbb{R}) \to L^2(\mathbb{R})$ is given by $T_c f(t) = f(t-c)$ for all constants $c \in \mathbb{R}$. Example of a translated signal can be seen in Figure 4.3.

Definition 4.1. Let f be a function in $L^2(\mathbb{R})$. An operator $\Phi : L^2(\mathbb{R}) \to \mathcal{H}$ is *invariant to translations* if

$$\forall c \in \mathbb{R}, \ \|\Phi(f) - \Phi(T_c f)\|_{\mathcal{H}} = 0.$$

Example 4.1. The *Fourier modulus operator* $f \mapsto |\hat{f}|$ is an operator that is translation invariant. The Fourier transform of a translated signal is $\widehat{T_c f}(\omega) = e^{-ic\omega}\hat{f}(\omega)$, which gives that the Fourier modulus of a translated signal is $|\widehat{T_c f}(\omega)| = |e^{-ic\omega}\hat{f}(\omega)| = |\hat{f}(\omega)|$.

In practice, we consider signals (or images) which are concentrated within given time (or space) frame. What we want is stability with respect to translations within this frame only. That is, if both a signal f and a shifted signal $T_c f$ are considered within the frame, we demand

$$\|\Phi(f) - \Phi(T_c f)\|_{\mathcal{H}} \ll 1.$$

Later, we do this relation more precise by specifying frame size and corresponding scale of the windowed scattering transform (4.1). In the rest of this thesis, when translation invariance is discussed, we really refer to translation within a given frame.

4.2 Stability to deformations

A deformation is a C^2 diffeomorphism on $\mathbb{R}$. For two intervals X and Y in $\mathbb{R}$, a function $\tau : X \to Y$ is a C^2 diffeomorphism if τ is bijective and τ and its inverse $\tau^{-1} : Y \to X$ are two times continuously differentiable. The deformation operator related to τ is denoted L_τ, and the deformation of a signal f is $L_\tau f(t) = f(t - \tau(t))$. An example of a deformed signal can be seen in Figure 4.4. Notice that the deformation of a signal resembles the translation of a signal, but for the deformation, the amount of translation is dependent on the time t. A simple example of a deformation is the dilation $\tau(t) = \epsilon t$ for $\epsilon \in \mathbb{R}\backslash\{0\}$. This deformation

will be used throughout the text as a model example. We say that the deformation $\tau(x) = \epsilon x$ is small when ϵ is small, that is $\epsilon \ll 1$.

Definition 4.2. We say that an operator Φ is *stable to the action of deformations* if Φ is Lipschitz continuous to the action of C^2 diffeomorphisms, which means that $\|\Phi(f) - \Phi(L_\tau f)\|_{\mathcal{H}}$ is bounded by the size of the deformation. In one dimension, the size of deformation over a compact set $\Omega \subset \mathbb{R}$ is given by the norm

$$\|\tau\|_\infty = \sup_{t\in\Omega} |\tau(t)| + \sup_{t\in\Omega} |\frac{d}{dt}\tau(t)|,$$

where $\sup_{t\in\Omega} |\tau(t)|$ is the maximum amplitude of the deformation and $\sup_{t\in\Omega} |\frac{d}{dt}\tau(t)|$ is the gradient of the deformation. We say that Φ is Lipschitz continuous to the action of C^2 diffeomorphism, and therefore stable to the action of deformations, if

$$\|\Phi(f) - \Phi(L_\tau f)\| \leq C \|f\|_2 \left(\sup_{t\in\Omega} |\tau(t)| + \sup_{t\in\Omega} |\frac{d}{dt}\tau(t)| \right).$$

Example 4.2. The *Fourier modulus* is not stable to deformations. Let τ be the dilation $\tau(t) = \epsilon t$. Let $f(t) = e^{i\xi t}\theta(t)$, where θ is regular, have fast decay, and $\hat{\theta}(\omega)$ is concentrated near the origin. A deformed signal can then be written $L_\tau f(t) = f(t - \tau(t)) = f((1-\epsilon)t) = f(at)$ where $a = 1 - \epsilon$. Now the Fourier transform gives

$$\widehat{L_\tau f}(\omega) = \int_{-\infty}^{\infty} \theta(at)\, e^{i\xi at}\, e^{-i\omega t}\, dt = \frac{1}{a}\, \hat{\theta}\left(\frac{\omega - a\xi}{a}\right).$$

The central frequency of $\hat{\theta}$ is ξ, while the central frequency of $\widehat{L_\tau f}(\omega)$ is $a\xi = (1-\epsilon)\xi$, as illustrated in Figure 4.5. Therefore, the difference $\| |\widehat{L_\tau f}| - |\hat{f}| \|$ is non-negligible, and proportional to $|\epsilon|\,|\xi|\,\|\theta\|$. As $|\xi|$ can be chosen arbitrarily large, the Fourier modulus is not Lipschitz continuous to the action of C^2 diffeomorphisms.

4.3 Construction of invariant operator

We want to construct an operator Φ that is invariant to translations and stable to deformations. The next theorem proves that the Littlewood-Paley wavelet transform (3.5) at scale 2^j is stable to deformations. How to retain all frequency information will be discussed in Section 4.4. The theorem is proved in one dimension, but can be extended to two dimensions.

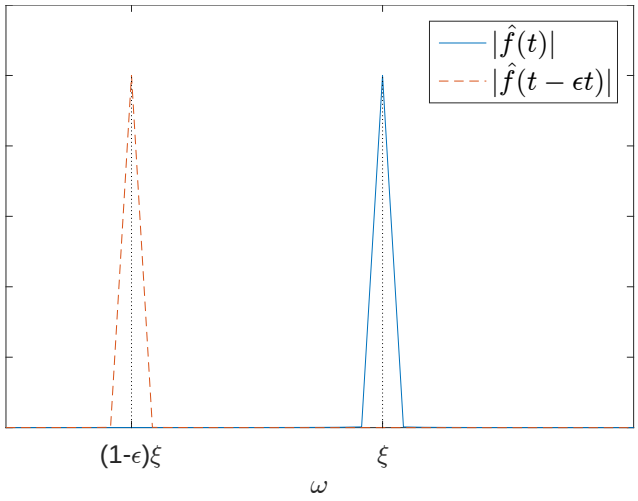


Figure 4.5: Fourier modulus is not stable to deformations.

Theorem 4.3. *Let $f \in L^2(R)$ be a bounded signal with finite support, that is $supp(f) \subset [-N, N]$ and let a deformation be $\tau(x) = \epsilon x$, with $\epsilon \ll 1$. Then the Littlewood-Paley wavelet transform $W[j] : L^2(R) \to L^2(R)$ of f is stable to the action of deformations, that is*

$$\|W[j]L_\tau f - W[j]f\|_2 \leq C(N)\|f\|_2 \left(\sup_{x\in\Omega} |\tau(x)| + \sup_{x\in\Omega} |\tau'(x)| \right).$$

Proof. We see that

$$W[j]f(t) = \int_{-\infty}^{\infty} f(u)\psi_j(t-u)du$$

and that

$$\begin{aligned} W[j]L_\tau f(t) &= \int_{-\infty}^{\infty} f\left((1-\epsilon)u\right)\ \psi_j(t-u)du \\ &= \frac{1}{1-\epsilon}\int_{-\infty}^{\infty} f(v)\psi_j\left(t - \frac{v}{1-\epsilon}\right)dv. \end{aligned}$$

Using the two previous results, and by rewriting the integral as a Riemann sum, we get

$$\begin{aligned} \|W[j]L_\tau f - W[j]f\|_2 &= \left\| \int_{-\infty}^{\infty} f(u)\left[\frac{1}{1-\epsilon}\psi_j\left(t - \frac{u}{1-\epsilon}\right) - \psi_j(t-u)\right]du \right\|_2 \\ &= \left\| \sum_{i=-\infty}^{\infty} \Delta f(u_i)\left[\frac{1}{1-\epsilon}\psi_j\left(t - \frac{u_i}{1-\epsilon}\right) - \psi_j(t-u_i)\right] \right\|_2 \\ &\leq \sum_{i=-\infty}^{\infty} \Delta|f(u_i)| \left\| \frac{1}{1-\epsilon}\psi_j\left(t - \frac{u_i}{1-\epsilon}\right) - \psi_j(t-u_i) \right\|_2, \end{aligned}$$

where $\Delta = u_i - u_{i-1}$ and the inequality follows from Minkowski's inequality A.5. Now we estimate the norm

$$\begin{aligned}
&\left\| \frac{1}{1-\epsilon}\psi_j\left(t - \frac{u_i}{1-\epsilon}\right) - \psi_j(t-u_i)\right\|_2 \\
&= \left\| \psi_j\left(t - \frac{u_i}{1-\epsilon}\right) - \psi_j(t-u_i) + \left(\frac{1}{1-\epsilon} - 1\right)\psi_j\left(t - \frac{u_i}{1-\epsilon}\right)\right\|_2 \\
&\leq \left\| \psi_j\left(t - \frac{u_i}{1-\epsilon}\right) - \psi_j(t-u_i)\right\|_2 + \frac{|\epsilon|}{|1-\epsilon|}\|\psi_j\|_2,
\end{aligned}$$

where the inequality again follows from Minkowski's inequality A.5. Now using the first order Taylor series expansion A.8, that is $f(x) - f(a) = (x-a)f'(a)$, we approximate

$$\left\| \psi_j\left(t - \frac{u_i}{1-\epsilon}\right) - \psi_j(t-u_i)\right\|_2 \approx \left\| u_i\left(1 - \frac{1}{1-\epsilon}\right)\psi_j'(t-u_i)\right\|_2 \leq N\frac{|\epsilon|}{|1-\epsilon|}\left\|\psi_j'\right\|_2,$$

where u_i is bounded by N, which is the size of the support of f. Combining all previous estimates results in

$$\begin{aligned}
\|W[j]L_\tau f - W[j]f\|_2 &\leq \sum_{i=-\infty}^{\infty} \Delta|f(u_i)| \cdot \frac{|\epsilon|}{|1-\epsilon|}\left[N\left\|\psi_j'\right\|_2 + \|\psi_j\|_2\right] \\
&= \|f\|_1 \frac{|\epsilon|}{|1-\epsilon|}\left[N\left\|\psi_j'\right\|_2 + \|\psi_j\|_2\right].
\end{aligned}$$

Using the fact that f is bounded and has finite support, Holders inequality A.6 gives that

$$\|f \cdot 1\|_1 \leq \|1\|_2\|f\|_2 = \left(\int_{-N}^{N} dt\right)^{\frac{1}{2}} \|f\|_2 = \sqrt{2N}\|f\|_2$$

which inserted into the above equation gives that

$$\begin{aligned}
\|W[j]L_\tau f - W[j]f\|_2 &\leq \sqrt{2N}\|f\|_2 \frac{|\epsilon|}{|1-\epsilon|}\left[N\left\|\psi_j'\right\|_2 + \|\psi_j\|_2\right] \\
&= C(N)\ \|f\|_2\ \frac{|\epsilon|}{|1-\epsilon|} \approx C(N)\ \|f\|_2\ |\epsilon|. \quad \square
\end{aligned}$$

Neither the Fourier modulus nor the wavelet transform have all the properties we want. Example 4.1 and 4.2 showed that the Fourier modulus is translation invariant, but not stable to deformations. The wavelet transform is stable to deformations, but not translation invariant. Translating a signal will also translate

the wavelet transform of the signal,

$$W[j]T_c f(t) = \int_{-\infty}^{\infty} T_c f(u)\, \psi_j(t-u)\, du = \int_{-\infty}^{\infty} f(u-c)\, \psi_j(t-u)\, du$$
$$= \int_{-\infty}^{\infty} f(v)\, \psi_j(t-c-v)\, dv = W[j]f(t-c).$$

A different operator is needed. Now let $U[j]$ be an operator which is defined on $L^2(\mathbb{R})$, and that commutes with translations, that is $T_c U[j]f = U[j]T_c f$. Since $U[j]$ commutes with translations, we see that $\int U[j]f(x)dx$ is translation invariant if the integral is well defined. The wavelet transform $W[j]f = f*\psi_j$ does in fact commute with translations due to how convolutions behave. However, $\int W[j]f(t)\, dt = 0$, as will be shown by the next computation.

$$\int_{-\infty}^{\infty} W[j]f(t)\, dt = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(u)\psi_j(t-u)\, du\, dt$$
$$= \int_{-\infty}^{\infty} f(u) \int_{-\infty}^{\infty} \psi_j(t-u)\, dt\, du = 0,$$

where we have applied Fubini's theorem A.2 and the fact that wavelets have zero average $\int \psi(t)\, dt = 0$. As a matter of fact, it can be shown than any linear transformation of $W[j]f$, which is translation invariant, will be zero [13].

Following Mallat [13] we will use a modulus operation in order to introduce non-linearity in a simple way. The operator $U[j]f = |f * \psi_j|$ is non-linear and stable to deformations. Furthermore, the integral $\int U[j]f(t)dt = \int |(f * \psi_j)(t)|\, dt$ is invariant with respect to translations. However, as the next example shows, the modulus maps the frequencies of $W[j]f$ to lower frequencies.

Example 4.3. Let $f(x) = e^{i\xi_1 t} + e^{i\xi_2 t}$ where ξ_1 and ξ_2 are positive and sits in the frequency band covered by $\hat{\psi}_j$. Then

$$U[j]f(t) = |(f * \psi_j)(t)| = |\int_{-\infty}^{\infty} \left(e^{i\xi_1(t-u)} + e^{i\xi_2(t-u)}\right) \psi_j(u)\, du|$$
$$= |e^{i\xi_1 t} \int_{-\infty}^{\infty} e^{-i\xi_1 u}\, \psi_j(u)\, du + e^{i\xi_2 t} \int_{-\infty}^{\infty} e^{-i\xi_2 u}\, \psi_j(u)\, du|$$
$$= |\hat{\psi}(2^{-j}\xi_1)\, e^{i\xi_1 t} + \hat{\psi}(2^{-j}\xi_2)\, e^{i\xi_2 t}| = |\hat{\psi}(2^{-j}\xi_1) + \hat{\psi}(2^{-j}\xi_2)\, e^{i(\xi_2-\xi_1)t}|,$$

which means that $|(f * \psi_j)(t)|$ oscillates at frequency $|\xi_2 - \xi_1|$. We see that the frequencies of $W[j]f$ have been shifted to lower frequencies since $|\xi_2 - \xi_1|$ is lower than $|\xi_1|$ and $|\xi_2|$.

Let $\mathbb{1}$ be a function that is equal to one for all t. Notice that the integration of $U[j]f$ can be written as a convolution with $\mathbb{1}$, that is $\int U[j]f(t)dt = U[j]f * \mathbb{1}$. More generally, any convolution with $\mathbb{1}$ gives the one-dimensional integral of that function. The Fourier transform of $\mathbb{1}$ is the Dirac delta function $\delta(\omega)$. This means that convolving a signal with $\mathbb{1}$ is the same as filtering the signal with a low-pass filter. Let ϕ_J be a low-pass filter. Since $U[j]f$ filtered with the low-pass filter $\mathbb{1}$ is translation invariant, then $U[j]f$ filtered with a low-pass filter $(f * \phi_J)(u) = \int f(t)\phi_J(u-t)dt$ will also be translation invariant. The reasoning behind our shift from $\mathbb{1}$ to ϕ_J is that the filter $\mathbb{1}$ is not well suited for numerical computations. How to choose the low-pass filter ϕ_J will be explained in Section 4.4.

Filtering a signal with a low-pass filter will cause loss of high-frequency information. In order to regain those frequencies, the wavelet transform $W[j]$ defined in (3.5) will be applied to $U[j]f(t)$. How to make sure that all frequencies are covered will be explained in Section 4.4. The modulus will also be applied again, $|W[j']U[j]f| = |U[j]f * \psi_{j'}| = U[j']U[j]f$, such that the integration $\int U[j']U[j]f(t)\phi_J(u-t)dt$ will be translation invariant. Let us illustrate this translation invariance by an example.

Example 4.4. Let $f(x) = e^{i\xi_1 t} + ae^{i\xi_2 t}$, where $a < 1$, and ξ_1 and ξ_2 are in the frequency band covered by $\hat{\psi}_j$. This signal is the same as the one used in Example 4.3 except for the introduction of the scalar a. By using the computation from Example 4.3 we get

$$(U[j]f * \psi_{j'})(t) = \int_{-\infty}^{\infty} |\hat{\psi}(2^{-j}\xi_1) + a\,\hat{\psi}(2^{-j}\xi_2)\,e^{i(\xi_2-\xi_1)u}|\,\psi_{j'}(t-u)\,du.$$

Now if $|\xi_2 - \xi_1|$ is in the support of $\psi_{j'}$, and $|\xi_2 - \xi_1| \ll 2^j$, we get that

$$\begin{aligned}(U[j]f * \psi_{j'})(t) &= a\,\hat{\psi}(2^{-j}\xi_2)\int_{-\infty}^{\infty} e^{i(\xi_2-\xi_1)u}\,\psi_{j'}(t-u)\,du \\ &= -a\,e^{i(\xi_2-\xi_1)t}\,\hat{\psi}(2^{-j}\xi_2)\,\hat{\psi}_{j'}(\xi_2-\xi_1),\end{aligned}$$

which gives that $U[j']U[j]f(t)$ is equal to the constant $|\hat{\psi}(2^{-j}\xi_2)|\,|\hat{\psi}_{j'}(\xi_2-\xi_1)|$, and constants are translation invariant.

4.4 Scattering transforms

In the previous section we defined the operator $U[j]f = |f * \psi_j|$. This operator is stable to deformations, and the integral $\int U[j]f(t)\phi_J(u-t)dt = \int |(f * \psi_j)(t)|\phi_J(u-t)dt$ is translation invariant. For a fixed j, $U[j]f$ will only have information about frequencies that are covered by the frequency band of the corresponding wavelet ψ_j. We need to cover all the frequencies of f or else we are not able to distinguish f and a signal f^- which lacks some of the frequencies of f. Together the elements of the set $\{U[j]f : j \in \mathbb{Z}\}$ cover all frequencies, but each element is not translation invariant. The elements of the set $\{\int U[j]f(t)\phi_J(u-t)dt : j \in \mathbb{Z}\}$ are translation invariant, but due to the low-pass filtering, high frequency information is lost.

Now we repeatedly apply $U[j_k]$ to $U[j]f$ for some $j_k \in \mathbb{Z}$. As shown in Example 4.3, each time an $U[j_k]$ is applied the frequencies are shifted lower because of the modulus operator. By applying $U[j_k]$ a sufficient number of times, the frequencies of the original signal f are shifted so low that they are inside the frequency band of the low-pass filter ϕ_J when ϕ_J is applied to $U[j_k]U[j_{k-1}]\dots U[j_2]U[j_1]U[j]f$.

In order to cover all frequencies, we need to systematically apply $U[j]$ iteratively and record with low-pass filter ϕ_J. This procedure is known as the windowed scattering transform. The notation will be same as the one used by Mallat [13]. First, we define paths and the scattering propagator $U[p]$.

Definition 4.4. A *path* is an ordered sequence $p = (j_1, j_2, \dots, j_m)$ where $j_k \in \mathbb{Z}$. The empty path is denoted $p = \emptyset$.

Definition 4.5. A *scattering propagator* is a path ordered product of operators defined by

$$U[p] = U[j_m]\dots U[j_2]U[j_1],$$

which gives that the scattering propagator applied to $f \in L^2(\mathbb{R})$ is

$$U[p]f = ||f * \psi_{j_1}| * \psi_{j_2}| \cdots | * \psi_{j_m}|.$$

Notice that the scattering propagator with the empty set as its path is the identity $U[\emptyset] = Id$.

Two paths $p = (j_1, \dots, j_m)$ and $p' = (j'_1, \dots, j'_m)$ can be concatenated by $p + p' = (j_1, \dots, j_m, j'_1, \dots, j'_m)$. Note that $p + j = (j_1, \dots, j_m, j)$. Furthermore, note that $U[p + p'] = U[p']U[p]$, which follows from the definition of $U[p]$.

Proposition 4.6. *Let $\mathcal{P}_\infty$ be the set of all finite paths. For each finite path $p \in \mathcal{P}_\infty$, the operator $U[p]$ is well defined on $L^2(\mathbb{R})$.*

Proof. In Proposition 3.5 it was proved that $\|W[j]f\|_2 \leq \|f\|_2 \|\psi_j\|_1$, which gives that $\|U[j]f\|_2 \leq \|f\|_2 \|\psi_j\|_1$. By iteratively applying this result to $\|U[p]f\|_2$ we get that

$$\|U[p]f\|_2 \leq \|f\|_2 \|\psi_{j_1}\|_1 \|\psi_{j_2}\|_1 \cdots \|\psi_{j_m}\|_1 = \|f\|_2 \|\psi_j\|_1^m.$$

Since $\|\psi_j\|_1 = 1$ for all j, $\|U[p]f\|_2$ is finite as well. □

The scattering propagator is a cascade of convolutions and moduli, all of which are stable to the action of deformation, and the scattering propagator is therefore also stable to deformations. Next, we define the scattering transform, which is also invariant to translations.

Definition 4.7. For all $p \in \mathcal{P}_\infty$, the *scattering transform* $\bar{S}$ of $f \in L^1(\mathbb{R})$ is a function defined on $\mathcal{P}_\infty$ by the relation

$$\bar{S}f(p) = \frac{1}{\mu_p} \int_{-\infty}^{\infty} U[p]f(t)\, dt, \quad \text{with} \quad \mu_p = \int_{-\infty}^{\infty} U[p]\delta(t)\, dt,$$

where μ_p is a normalization constant.

The path variable p in the scattering transform plays the role of the frequency variable ω in the Fourier transform (2.1). Convolving a signal with a wavelet ψ_j filter the signal according to the frequency band covered by $\hat{\psi}_j$. This indicates that p is related to frequencies. Next, we define the windowed scattering transform, which is analogous to the windowed Fourier transform (2.4).

Definition 4.8. Let $J \in \mathbb{Z}$, and fix a low-pass filter $\phi_J(t) = 2^{-J}\phi(2^{-J}t)$. Let $\Lambda_J = \{j \in \mathbb{Z} : j < J\}$, and $\mathcal{P}_J$ be the set of all finite paths $p = (j_1, \ldots, j_m)$ with $j_k \in \Lambda_J$. Then a *windowed scattering transform* of $f \in L^2(\mathbb{R})$ is defined for all $p \in \mathcal{P}_\mathcal{J}$ by

$$S_J[p]f(t) = (U[p]f * \phi_J)(t) = \int_{-\infty}^{\infty} U[p]f(u)\, \phi_J(t-u)\, du.$$

We see that the path variable p in the windowed scattering transform plays the role of the frequency variable ω and the low-pass filter ϕ_J plays the role of the window function g in the windowed Fourier transform (2.4). The parameter J needs to be chosen such that the maximum translation between two signals we want to recognize similarity in, is smaller than 2^J. In practice, J is chosen such

that 2^J is larger than the size of the frame our signals are defined on. This ensures that all translations are covered.

The transform $\bar{S}[p]$ can be seen as a convolution with the low-pass filter $\mathbb{1}$, hence its translation invariance. However, a convolution with $\mathbb{1}$ cannot be implemented numerically. Therefore, $S_J[p]$ has to be implemented instead. Notice that as J goes to infinity, the limit of the windowed scattering transform $S_J[p]$ is proportional to the scattering transform $\bar{S}(p)$. We use that $\phi(t)$ is continuous at $t = 0$ and get that for all $t \in \mathbb{R}$

$$\lim_{J\to\infty} 2^J \, S_J[p]f(t) = \phi(0) \int_{-\infty}^{\infty} U[p]f(u) \, du = \phi(0) \, \mu_p \, \bar{S}f(p).$$

Computing $S_J[p]f(t)$ will only cover one frequency path, but we need to cover as many frequencies as possible. Therefore, following the notation of Mallat [13], we are going to compute the windowed scattering transform of $\mathcal{P}_J$.

Definition 4.9. Let $\mathcal{P}_J$ be the set of all finite paths $p = (j_1, \ldots, j_m)$ with $j_k \in \Lambda_J$. The *windowed scattering transform* of $f \in L^2(\mathbb{R})$ is defined for all $J \in \mathbb{Z}$ by

$$S_J[\mathcal{P}_J]f = \{S_J[p]f\}_{p\in\mathcal{P}_J} \, , \tag{4.1}$$

which is an infinite family of functions.

The windowed scattering transform will be computed by iteratively applying the one-step propagator U_J.

Definition 4.10. The *one-step propagator* is defined for all $J \in \mathbb{Z}$ by

$$U_J f = \{A_J f, (U[j]f)_{j\in\Lambda_j}\},$$

where $A_J f = f * \phi_J$.

The set $U_J f$ is an infinite family of functions. The next iteration is computed by applying U_J to $U[j]f$ for each $j \in \Lambda_j$, which results in an extended infinite family of functions $\{U_J U[j]f\}_{j\in\Lambda_j}$. Next, notice that

$$U_J U[p]f = \{S_J[p]f, (U[p + j]f)_{j\in\Lambda_j}\}$$

since $A_J U[p]f = S_J[p]f$ and $U[j]U[p]f = U[p+j]$. Now, let Λ_J^m be the set of paths $p = (j_1, \ldots, j_m)$ with length m and with $j_k \in \Lambda_J$, where $\Lambda_J^0 = \emptyset$. We observe that

$$U_J U[\Lambda_J^m]f = \{S_J[\Lambda_J^m]f, U[\Lambda_J^{m+1}]f\},$$

which gives that $S_J[\mathcal{P}_J]f$ can be iteratively computed from $U[\emptyset]f = f$ since $\cup_{m\in\mathbb{N}} \Lambda_J^m = \mathcal{P}_J$. $S_J[\mathcal{P}_J]$ is an infinitely long set of functions from $L^2(\mathbb{R})$, which may be written

$$S_J[\mathcal{P}_J] = \begin{bmatrix} S_J[\emptyset]f \\ (S_J[j_1]f)_{j_1\in\Lambda_J} \\ (S_J[j_1,j_2]f)_{j_1,j_2\in\Lambda_J} \\ (S_J[j_1,j_2,j_3]f)_{j_1,j_2,j_3\in\Lambda_J} \\ \vdots \end{bmatrix} = \begin{bmatrix} f * \phi_{2^J} \\ (|f * \psi_{j_1}| * \phi_{2^J})_{j_1\in\Lambda_J} \\ (||f * \psi_{j_1}| * \psi_{j_2}| * \phi_{2^J})_{j_1,j_2\in\Lambda_J} \\ (|||f * \psi_{j_1}| * \psi_{j_2}| * \psi_{j_3}| * \phi_{2^J})_{j_1,j_2,j_3\in\Lambda_J} \\ \vdots \end{bmatrix}.$$

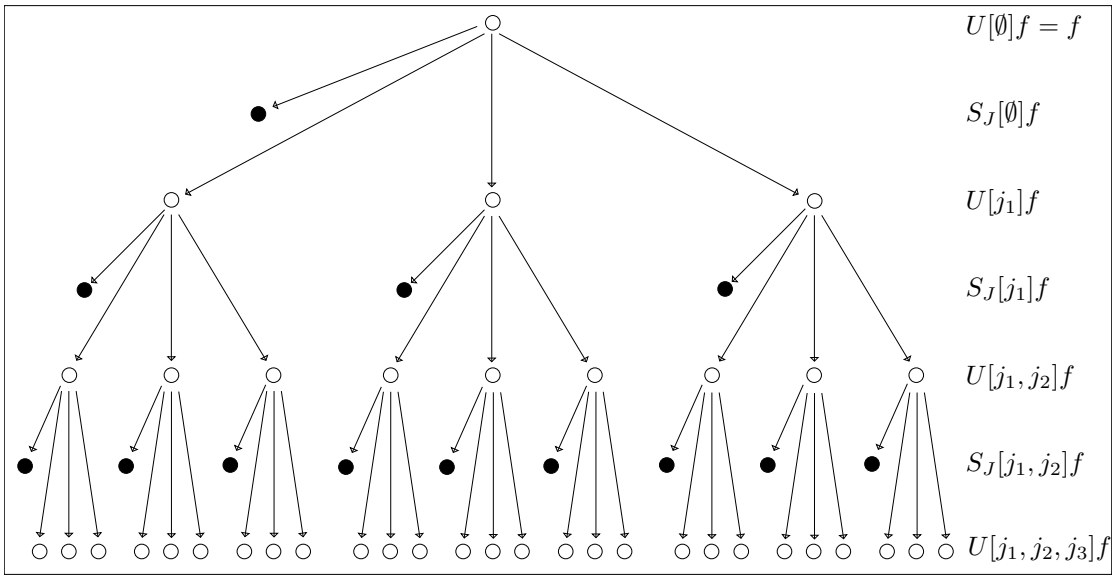


Figure 4.6: An illustration of how the one-step propagator U_J is used to compute $S_J[\mathcal{P}_J]$. The first two rows correspond to the zeroth layer, the next two rows correspond to the first layer and so on. For all layers subsequent to the first, only a few of the infinitely many nodes in the corresponding layer can be drawn. All j_k are indices from the infinite set Λ_J.

Each "row" in the system will be called a layer, where the topmost layer is denoted the 0th layer. The m'th layer will be all $S_J[p]$ with paths p of length m. All layers except the first will be infinitely long, and each layer $m+1$ will be an extension of layer m for all $m \in \mathbb{N}$. An illustration of the three first layers, that is path length $m = 2$, can be found in Figure 4.6. In order to compute a new layer, U_J will be applied to all $U[p]f$ from the previous layer. For all paths p in previous layer, this outputs a $S_J[p]$, and computes a $U[p + j_k]f$ for each $j_k \in \Lambda_J$. Those $U[p + j_k]f$

may further be used to compute the next layer. How to compute the windowed scattering transform numerically will be explained in Section 4.5.

Mallat proves [13] that the windowed scattering transform (4.1) is translation invariant and stable to deformations. The range of the windowed scattering transform is the product space generated by taking the Cartesian product, defined in A.1, of several $L^2(\mathbb{R})$-spaces. This product space is a Hilbert space, see A.4 for more details. Then we see that the norm of the windowed scattering transform (4.1) can be defined as follows.

Definition 4.11. For any set of finite paths Ω, the norm of $S_J(\Omega)$ is given by

$$\|S_J[\Omega]f\|_2 = \sum_{p\in\Omega} \|S_J[p]f\|_2.$$

The following theorem gives stability of the windowed scattering transform (4.1).

Theorem 4.12. *Let ϕ be a low-pass filter. A wavelet ψ is said to be admissible if there exists $\eta \in \mathbb{R}$ and a function ρ such that $\rho \geq 0$, with $|\hat{\rho}(\omega)| \leq |\hat{\phi}(2\omega)|$ and $\hat{\rho}(0) = 1$, such that the function*

$$\hat{\Psi}(\omega) = |\hat{\rho}(\omega - \eta)|^2 - \sum_{k=1}^{\infty} k\left(1 - |\hat{\rho}(2^{-k}(\omega - \eta))|^2\right)$$

satisfies

$$\alpha = \inf_{1\leq|\omega|\leq 2} \sum_{j=-\infty}^{\infty} \hat{\Psi}(2^{-j}\omega)|\hat{\rho}(2^{-j}\omega)|^2 > 0.$$

If the wavelet is admissible then for all $f \in L^2(\mathbb{R})$

$$\lim_{m\to\infty} \|U[\Lambda_J^m]f\|^2 = \lim_{m\to\infty} \sum_{n=m}^{\infty} \|S_J[\Lambda_J^n]f\|^2 = 0$$

and

$$\|S_J[P_J]f\| = \|f\|.$$

Proof. See Reference [13].

4.5 Numerical approximation

The windowed scattering transform (4.1) of a signal f outputs an infinitely long vector of functions from $L^2(\mathbb{R})$. In order to compute the transform (4.1) numerically, no part of the transform can be infinite. Several approximations will be made in order to achieve this. First, all functions need to be discrete rather than continuous. If we restrict the transform (4.1) to a finite number of layers, with a finite number of functions in each layer, the transform will be a finite vector of discrete signals, in other words a vector of vectors.

Now we need to determine how many layers we need, and which signals to keep in each layer. We see that increasing the number of layers increases the computational cost exponentially, and it is thus beneficial to have as few layers as possible. According to Reference [15], having more than three layers yields marginal improvements in applications. Therefore, we will typically use path length $m = 2$, which is three layers. All j_k belong to the set $\Lambda_J = \{j \in \mathbb{Z} : j < J\}$, but this set is infinite. Then, following Reference [15] we let all j_k belong to the finite set $\bar{\Lambda}_J = \{0, 1, \ldots, J-1\}$ instead of Λ_J. J is the number of scalings that are considered. With these approximations, the windowed scattering transform (4.1) will be a finite vector of discrete signals.

For m layers, the number of signals in the approximated widowed scattering transform is

$$1 + J + J^2 + \ldots + J^m = \sum_{n=0}^{m} J^n = \frac{J^{m+1} - 1}{J - 1} .$$

The number of signals can be lowered even further by considering frequency-decreasing paths. Let Ω be a path set and a subset of $\mathcal{P}_{\mathcal{J}}$. If we choose the paths in Ω to be those where $\|S_J[p]f\|$ contributes significantly to the total energy $\|S_J[\mathcal{P}_{\mathcal{J}}]f\|$, then $\|S_J[\Omega]f\|$ is an approximation of $\|S_J[\mathcal{P}_{\mathcal{J}}]f\|$. Every modulus operation in $U[p]$ shifts the frequencies of f to lower levels, which implies that the energy will be propagated towards lower frequencies. In other words, we only need to consider those paths p where the frequencies are decreasing.

Definition 4.13. A *frequency-decreasing path*, is a path $p = (j_1, \ldots, j_m)$ of length m, such that $|j_{k+1}| \geq |j_k|$ for all k.

When considering frequency decreasing paths the number og signals in the approximated widowed scattering transform is

$$1 + J + \frac{J(J+1)}{2} + \frac{J(J+1)(J+2)}{6} + \ \dots \ + \frac{\Pi_{i=0}^{m-1}(J+i)}{m!} = \sum_{n=0}^{m} \frac{(J+n-1)!}{n!(J-1)!}$$

$$= \frac{(J+m)!}{J!\, m!} \sim (2\pi m)^{-1/2} \left(\frac{eJ}{m}\right)^m,$$

where the last approximation follows from the assumption that $J \gg m$ and Stirling's formula $n! \sim \sqrt{2\pi n}(n/e)^n$ defined in A.9.

4.6 Two dimensions

By replacing how paths are defined, all the notation that was defined for one-dimensional scattering will be the same for two-dimensional scattering. First we introduce the set

$$\Gamma_J = \left\{2^j r_\theta \in 2^{\mathbb{Z}} \times SO(2) : j < J,\ \theta \in \{0, 2\pi/K, 4\pi/K, \ \dots \ , (K-1)2\pi/K\}\right\}.$$

In two dimensions a path is an ordered sequence $p = (\lambda_1, \lambda_2, \ \dots \ , \lambda_m)$ where $\lambda_k \in \Gamma_J$. Which results in the following scattering propagator

$$\begin{aligned} U[p]f &= U[\lambda_m] \dots U[\lambda_2]U[\lambda_1]f \\ &= |\,|f * \psi_{\lambda_1}| * \psi_{\lambda_2}| \cdots | * \psi_{\lambda_m}|. \end{aligned}$$

where $U[\lambda_k]f = |W[\lambda_k]f|$.

Definition 4.14. Let $\mathcal{P}_J$ be the set of all finite paths $p = (\lambda_1, \lambda_2, \ \dots \ , \lambda_m)$ with $\lambda_k \in \Gamma_J$. The *windowed scattering transform* of $f \in L^2(\mathbb{R}^2)$ is defined for all $J \in \mathbb{Z}$ by

$$S_J[\mathcal{P}_J]f = \{S_J[p]f\}_{p\in\mathcal{P}_J} = \{U[p]f * \phi_J\}_{p\in\mathcal{P}_J}\,. \tag{4.2}$$

The difference between scattering in one and two dimensions is that the paths are defined in order to account for the rotation in the two-dimensional Littlewood-Paley transform (3.10).

When discretizing scattering in two dimensions, λ_k belong to the set

$$\bar{\Gamma}_J = \left\{2^j r_\theta \in 2^{\mathbb{Z}} \times SO(2) : 0 \le j < J,\ \theta \in \{0, 2\pi/K, 4\pi/K, \ \dots \ , (K-1)2\pi/K\}\right\}.$$

The difference between Γ_J and $\bar{\Gamma}_J$ is that J is only bounded from below for $\bar{\Gamma}_J$. As for one dimensions, J is the number of scalings, while K is the number of rotations. For m layers, the number of signals in the approximated windowed scattering transform is

$$1 + (J \cdot K) + (J \cdot K)^2 + \ \ldots \ + (J \cdot K)^m = \sum_{n=0}^{m} (J \cdot K)^n = \frac{(J \cdot K)^{m+1} - 1}{(J \cdot K) - 1} \, .$$

When considering frequency decreasing paths, the number is reduced to

$$1 + JK + \frac{J(J+1)K^2}{2} + \frac{J(J+1)(J+2)K^3}{6} + \ \ldots \ + \frac{K^m \, \Pi_{i=0}^{m-1}(J+i)}{m!}$$

$$= \sum_{n=0}^{m} \frac{(J+n-1)! \, K^n}{n!(J-1)!} \sim \frac{1}{\sqrt{2\pi}} \sum_{n=0}^{m} \frac{1}{\sqrt{n}} \left(\frac{eK(J-1)}{n} \right)^n ,$$

where again the last approximation follows from the assumption that $J \gg m$ and Stirling's formula $n! \sim \sqrt{2\pi n}(n/e)^n$ defined in A.9.

Chapter 5

Numerical examples

In this chapter, we will compute the windowed scattering transform (4.1) numerically, and three different examples will be given to showcase the transform. The first two examples are one-dimensional, and will be presented in Section 5.1. The third example is two-dimensional, and will be presented in Section 5.2. In the first example, we will compute and plot the windowed scattering transform one step at a time. In the second and third examples, results showing translation invariance and stability to deformation are presented. Throughout the chapter we will use the rule of frequency decreasing paths 4.13 and path length $m = 2$. The scale 2^J is chosen to best fit each example.

5.1 One dimension

We will compute the windowed scattering transform (4.1) of f, g and h, which are shown in Figure 5.2. For this first example all the signals are contained in the time frame $[-2, 3]$. Therefore we will use $J = 3$ such that $2^J = 8$ which is larger than the size of the time frame. This implies that paths will be on the form $p = (j_1, j_2)$, where j_1 and j_2 will take values from the set $\bar{\Lambda}_3 = \{0, 1, 2\}$. This results in the path set

$$\bar{\Omega} = \{\emptyset, 1, 2, 3, (1,1), (1,2), (1,3), (2,2), (2,3), (3,3)\},$$

which contains ten elements. The structure of $S_3[\bar{\Omega}]$ applied to a signal f can be seen in Figure 5.1. In the last layer, only the outputs are computed. It is not necessary to compute the next layer of scattering propagators, because those will not be used.

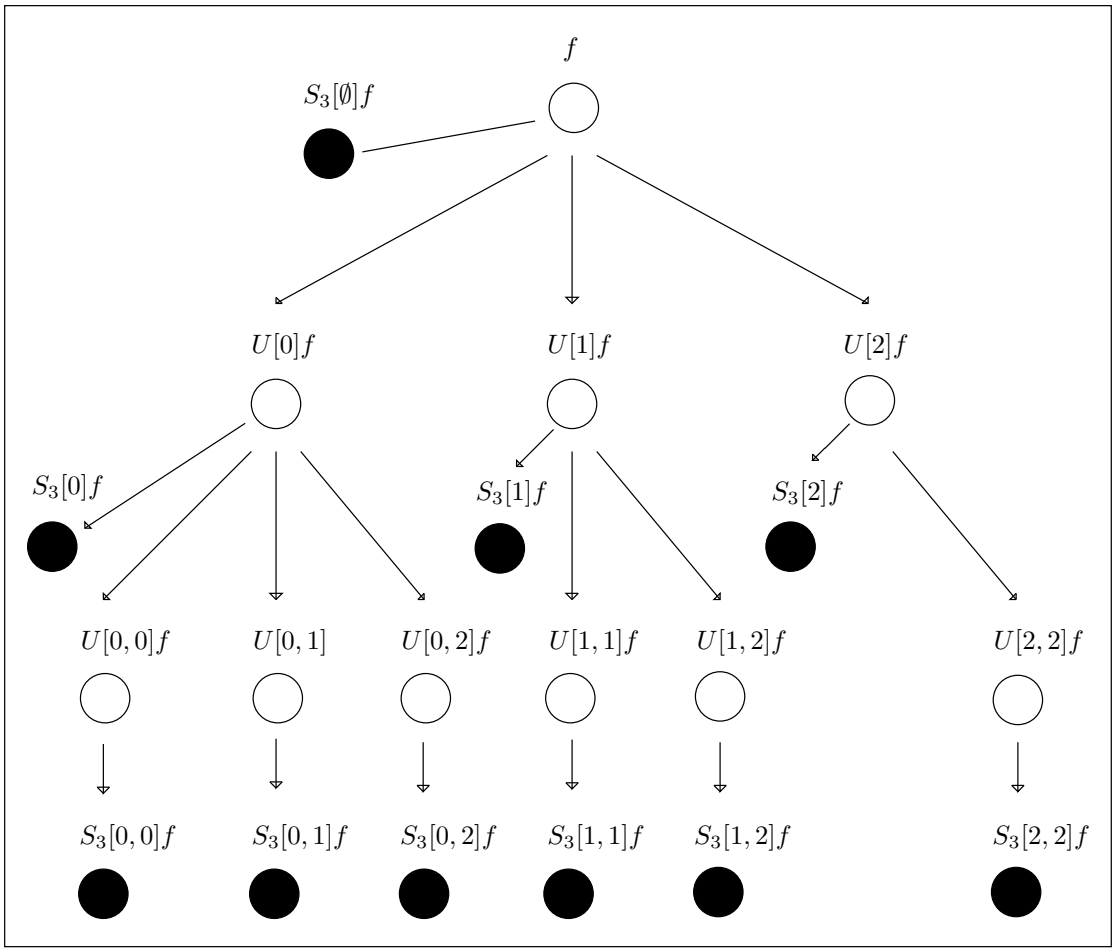


Figure 5.1: The structure of the windowed scattering transform for $m = 2$ and $J = 3$. The output of the transform is the ten black nodes. In the second layer, some paths have been removed due to the rule of frequency decreasing paths.

First we compute one of the outputs from $S_3[\bar{\Omega}]$. The operator $S_J[p] = S_3[0]$ will be applied to the three signals f, g and h. The signals are shown in Figure 5.2 and the Fourier transforms of those signals are shown in Figure 5.3. The computation of $S_J[p] = S_3[(0)]$ applied to the signals will be broken down into steps. Each step will be plotted and explained. Note that in this chapter, whenever the Fourier transform of a signal is plotted, the absolute value of that Fourier transform will be plotted instead. As the absolute value of the Fourier transforms are always symmetric, only the positive frequency axis will be displayed.

The first step is to take the wavelet transform $W[j] = W[0]$ of the signals. The wavelet that will be used is the Morlet wavelet given by

$$\psi_j(t) = 2^{-j}\,\psi(2^{-j}\,t) \text{ with } \psi(t) = \cos(2\pi t)\exp\Big(-t^2/2\Big).$$

The Morlet wavelet for $j = 0$ is shown in Figure 5.4 and its Fourier transform is shown in Figure 5.5. The wavelet transforms of f, g and h can be seen in Figure

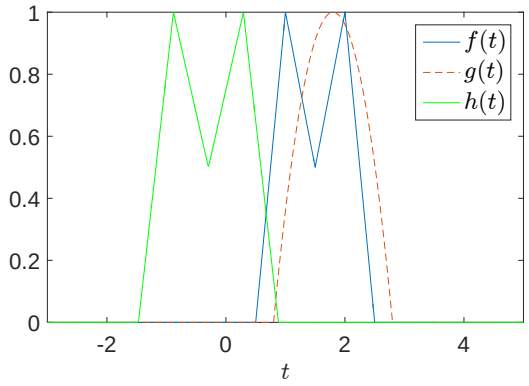


Figure 5.2: Signals f and g are similar in L^2-norm, signals f and h are similar in shape.

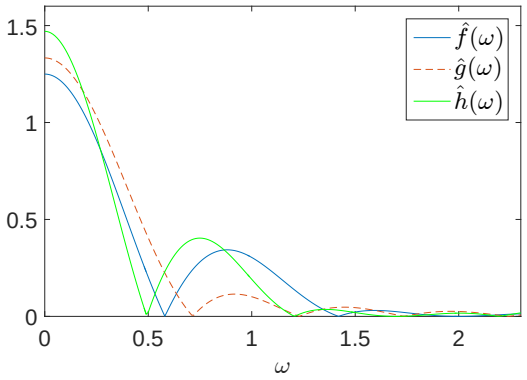


Figure 5.3: Fourier transforms of signals f, g and h.

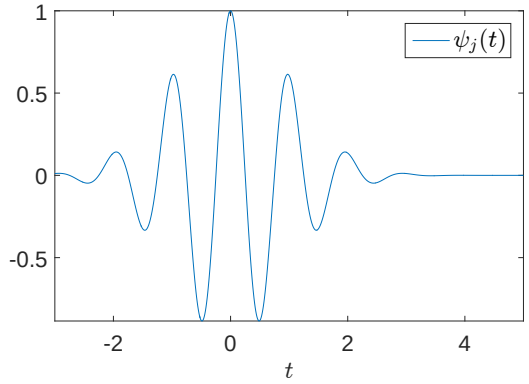


Figure 5.4: Morlet wavelet $\psi_j(t)$ for $j = 0$.

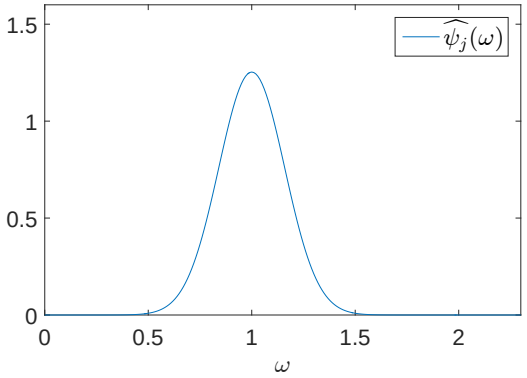


Figure 5.5: Fourier transform of the Morlet wavelet $\psi_j(t)$, with $j = 0$.

5.6 and their respective Fourier transforms can be seen in Figure 5.7. Notice that applying the wavelet transform has filtered the frequencies of the signals according to the frequency spectrum of the wavelet.

Next we take the absolute value of $W[0]f$, $W[0]g$ and $W[0]h$. That is $U[0]f = |W[0]f|$, $U[0]g = |W[0]g|$, and $U[0]h = |W[0]h|$ which are shown in Figure 5.8. Their respective Fourier Transforms are shown in Figure 5.9. As expected, the frequencies of $W[0]f$, $W[0]g$ and $W[0]h$ have been shifted to lower frequencies after applying the modulus.

Finally, the last step is to apply a low-pass filter ϕ_J. We will use the Gaussian distribution as a low pass filter, that is

$$\phi_J(t) = 2^{-J}\,\phi(2^{-J}\,t) \;\text{ with }\; \phi(t) = \frac{1}{\sqrt{2\pi}}\exp\Big(-t^2/2\Big).$$

The low pass filter ϕ_J is plotted for $J = 3$ in Figure 5.10, and its Fourier transform is plotted in Figure 5.11. Lastly, $U[0]f$, $U[0]g$ and $U[0]h$ are filtered by ϕ_3, resulting

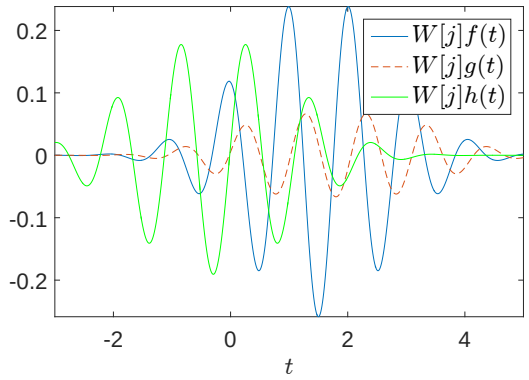


Figure 5.6: Wavelet transform $W[0]$ of signals f, g and h.

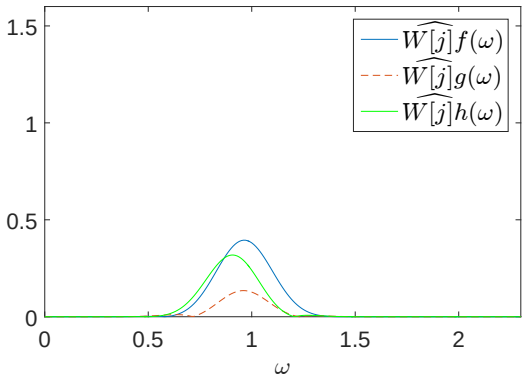


Figure 5.7: Fourier transforms of $W[0]f$, $W[0]g$ and $W[0]h$.

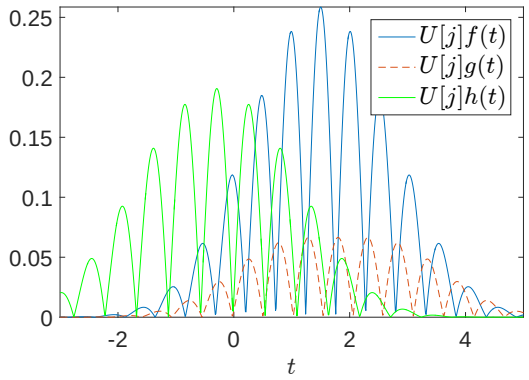


Figure 5.8: Operator $U[0]$ applied to signals f, g and h.

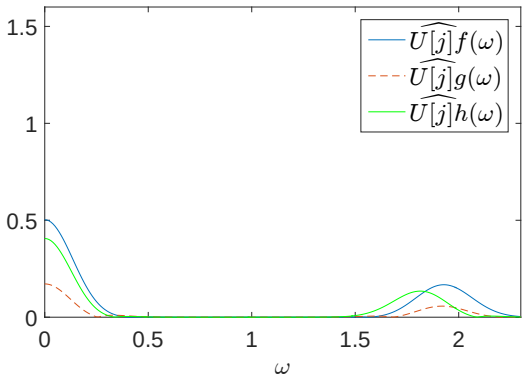


Figure 5.9: Fourier transforms of $U[0]f$, $U[0]g$ and $U[0]h$.

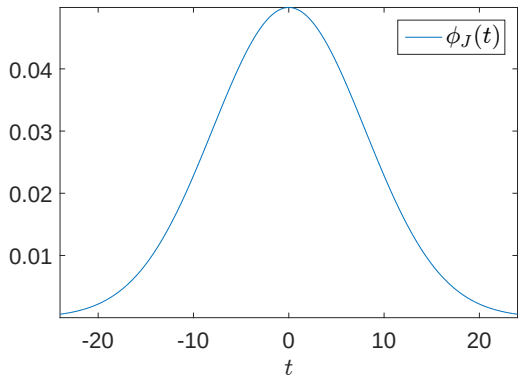


Figure 5.10: Low-pass filter $\phi_J(t)$ for $J = 3$.

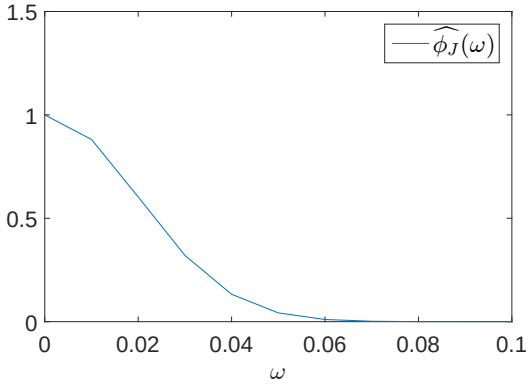


Figure 5.11: Fourier transform of the low-pass filter $\phi_J(t)$, with $J = 3$.

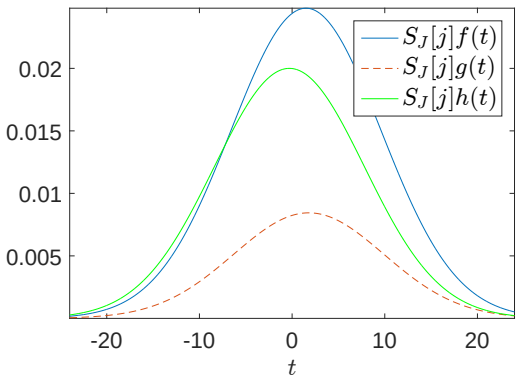


Figure 5.12: Operator $S_3[0]$ applied to signals f, g and h.

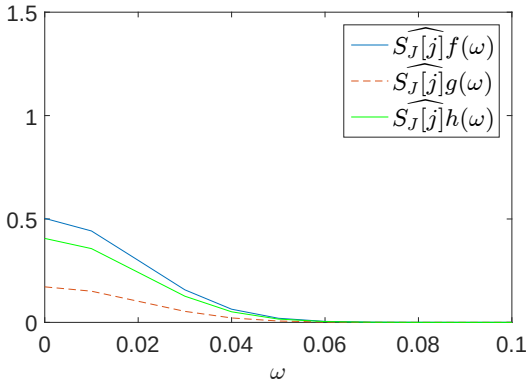


Figure 5.13: Fourier transforms of $S_3[0]f$, $S_3[0]g$ and $S_3[0]h$.

in $S_3[0]f$, $S_3[0]g$ and $S_3[0]h$ which are shown in Figure 5.12. Their respective Fourier transforms are shown in Figure 5.13. As expected, the low-pass filter has removed the high frequencies.

We started with three signals f, g and h, where f and g are close to each other in L^2-norm, while f and h are close to each other in shape. A simple calculation gives that $\|f-g\|_2 = 5.23 < \|f-h\|_2 = 13.89$. Another calculation gives that $\|S_3[0]f - S_3[0]h\|_2 = 0.22 < \|S_3[0]f - S_3[0]g\|_2 = 0.62$. For these computations the scale was $2^J = 2^3$ and the time step was 0.1. The path $p = (0)$ does capture the similarity in shape, but this path is only one path in the path set $\bar{\Omega}$. When considereing all the paths in $\bar{\Omega}$, see Figure 5.14, we get the desired result $\left\|S_3[\bar{\Omega}]f - S_3[\bar{\Omega}]h\right\|_2 = 1.94 < \left\|S_3[\bar{\Omega}]f - S_3[\bar{\Omega}]g\right\|_2 = 2.06$. We are also able to recognize similarity in shape when the rule of frequency decreasing paths is not applied. All of these results are summarized in Table 5.1.

Table 5.1: Difference between signals for different norms.

Functions	L^2-norm	Scattering - frequency decreasing	Scattering - full
$d(f,g)$	5.23	2.06	2.52
$d(f,h)$	13.89	1.94	2.47
$d(h,g)$	14.64	3.42	4.22

For the next example we once more consider m-shaped and disk-shaped signals, but several different translations and deformations will be examined. All the signals can be seen in Figure 5.15. The deformations are as follows: for the first two rows $\epsilon = 0$, for the two next rows $\epsilon = 0.1$ and for the last two rows $\epsilon = 0.15$. All m-shaped signals have been given a number from one to eight, and each disk-shaped signal have been given a number from nine to sixteen. We see that the signals are defined on a time frame of approximate length 2^5. Therefore we choose the scale $2^J = 2^5$. Then we compute $\|S_5[\Omega]f_i - S_5[\Omega]f_j\|_2$ for $i,j \in \{1,2,\ \ldots\ ,16\}$, where

Ω is the path set that corresponds to $J = 5$ and $m = 2$. The results can be seen in Figure 5.16. The square on the intersection of row i and column j shows the value of $\|S_5[\Omega]f_i - S_5[\Omega]f_j\|_2$. As expected the difference $\|S_5[\Omega]f_i - S_5[\Omega]f_j\|_2$ is small when comparing m-shaped signals with other m-shaped signals, as seen in the upper left quadrant. The difference is also small when comparing a disk-shaped signal with other disk-shaped signals, as seen in the lower right quadrant. The two remaining quadrants, in which m-shaped signals are compared to disk-shaped signals, give larger differences.

5.2 Two dimensions

For the last example we consider two-dimensional signals, and again we look at several translations and deformations. All the signals can be seen in Figure 5.17. The amount of deformation is as follows: for the first two rows $\epsilon = 0$ and for the two next rows $\epsilon = 0.1$. All box-shaped signals have been given a number from one to six, and each circle-shaped signal have been given a number from seven to twelve. We see that the space frame is of size $2^4 \times 2^4$, and we thus choose the scale $2^J = 2^4$. Then we compute $\|S_4[\Omega]f_i - S_4[\Omega]f_j\|_2$ for $i, j \in \{1, 2, \ldots, 12\}$, where Ω is the path set that corresponds to $J = 4$ and $m = 2$. The results can be seen in Figure 5.18. The square on the intersection of row i and column j shows the value of $\|S_4[\Omega]f_i - S_4[\Omega]f_j\|_2$. As expected, the difference $\|S_4[\Omega]f_i - S_4[\Omega]f_j\|_2$ is small when comparing box-shaped signals with other box-shaped signals, as seen in the upper left quadrant. The difference is also small when comparing a circle-shaped signal with other circle-shaped signals, as seen in the lower right quadrant. The two remaining quadrants, in which box-shaped signals are compared to circle-shaped signals, give larger differences.

This example does not include rotations as rotations, and thus the number of considered rotations K was equal to zero. Rotations have been excluded from this example since rotations will not be considered in Chapter 6, when the windowed scattering transform (4.2) is applied on handwritten letters.

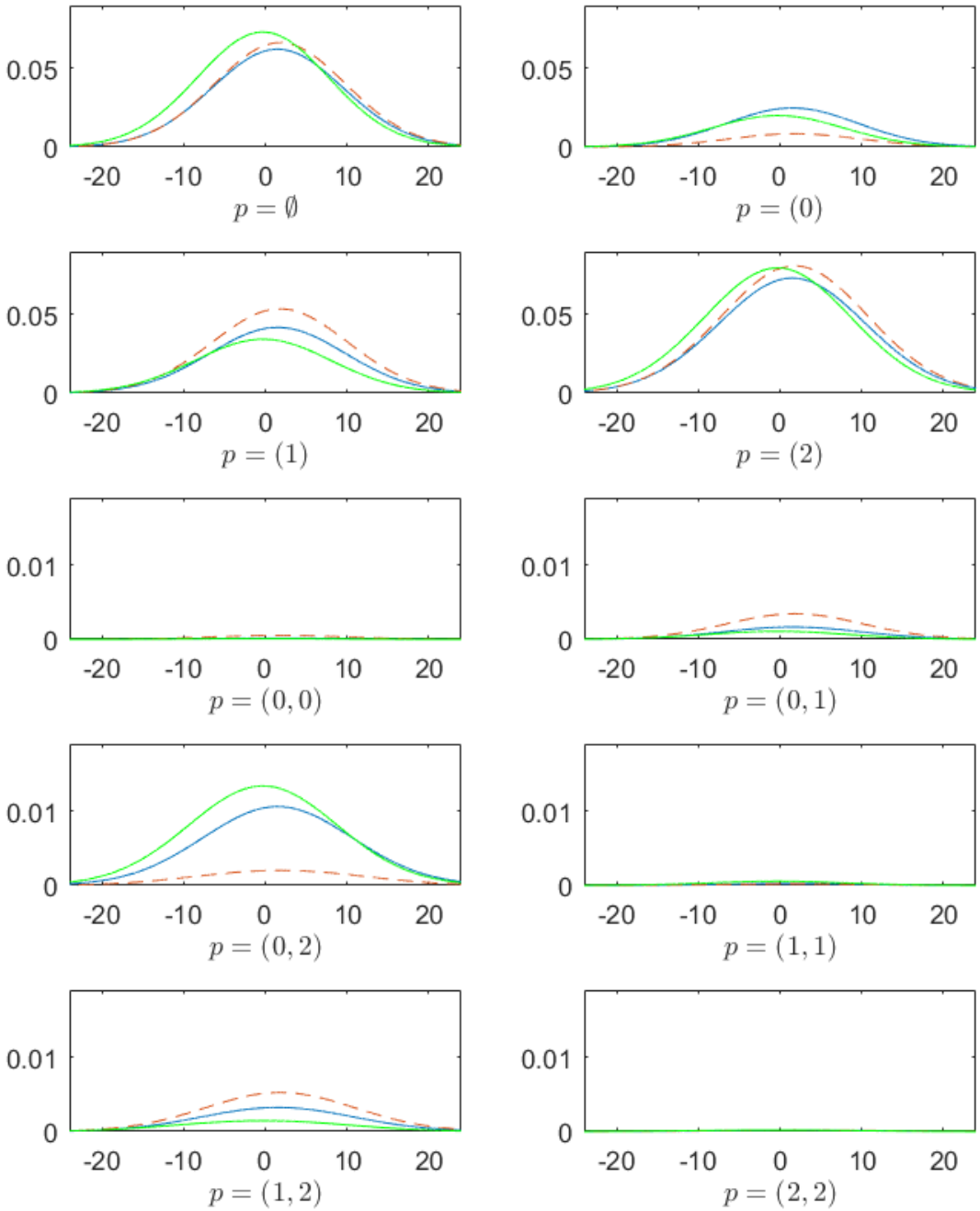


Figure 5.14: The windowed scattering transform $S_3[\bar{\Omega}]$ applied to signals f, g and h. For all paths $p \in \bar{\Omega}$, $S_3[p]f$ are plotted with blue lines, $S_3[p]g$ are plotted with red lines and $S_3[p]h$ are plotted with green lines.

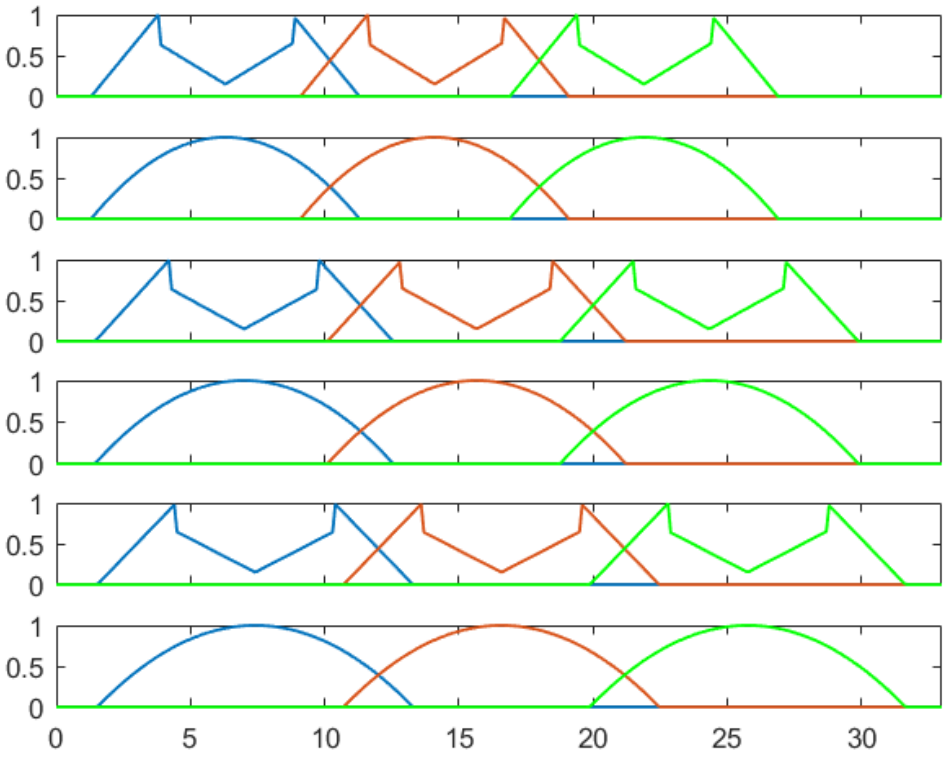


Figure 5.15: Several translated and deformed one-dimensional signals.

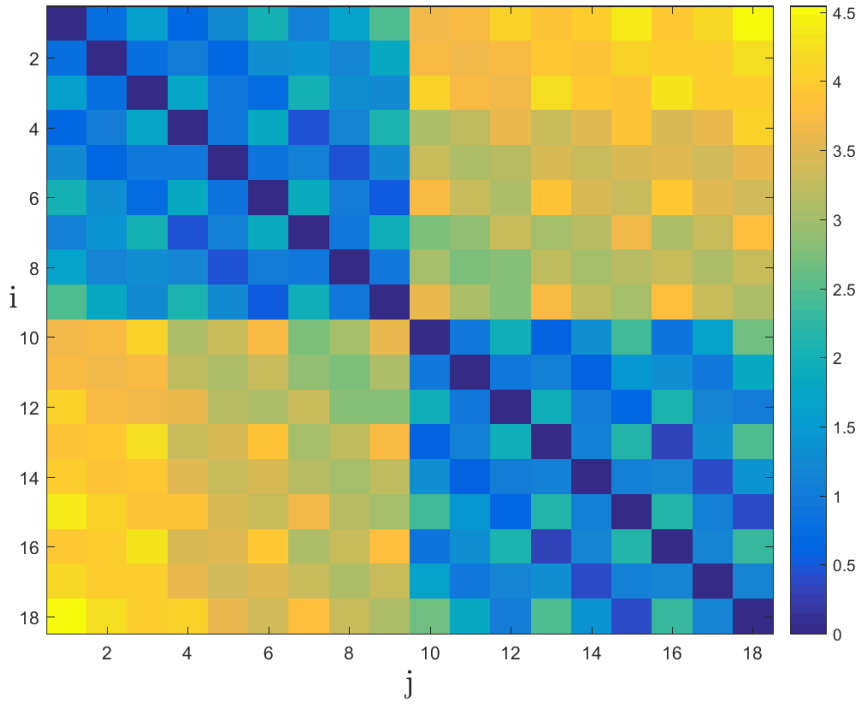


Figure 5.16: Norm of difference between several translated and deformed one-dimensional signals after applying the windowed scattering transform.

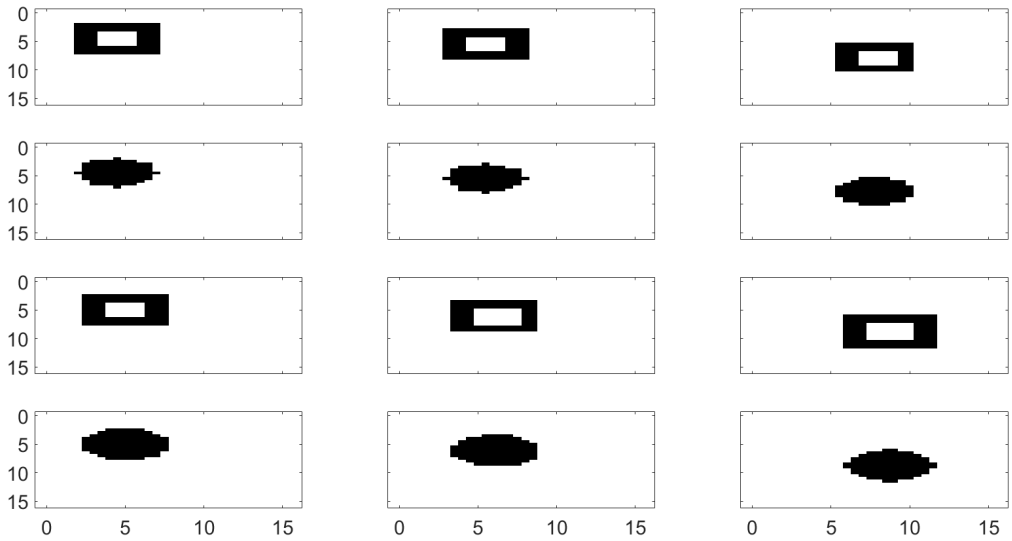


Figure 5.17: Several translated and deformed two-dimensional signals.

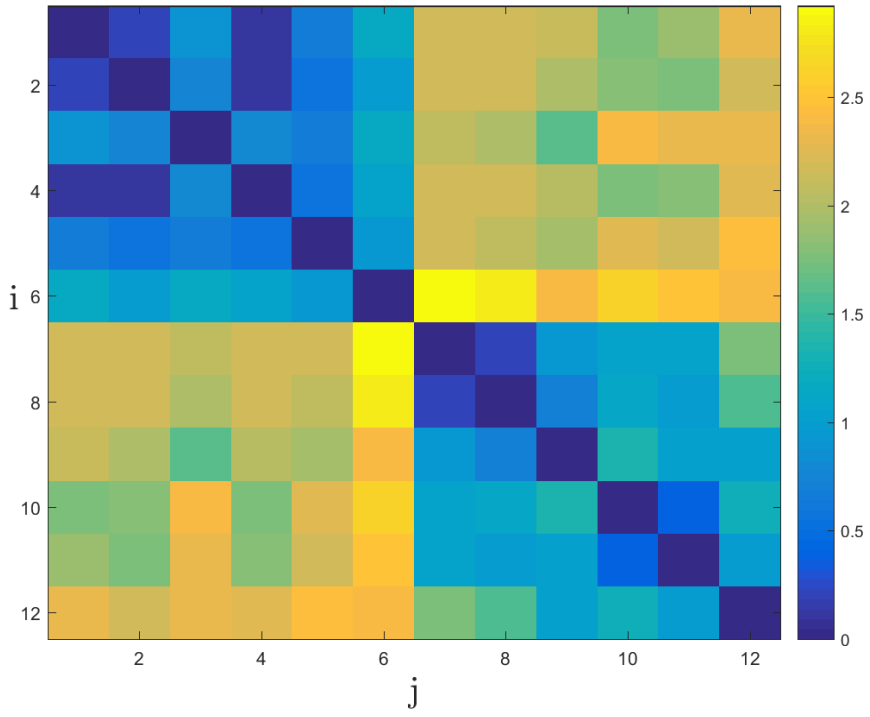


Figure 5.18: Norm of difference between several translated and deformed two-dimensional signals after applying the windowed scattering transform.

Chapter 6

Image recognition

In this chapter, image recognition will be performed on handwritten letters using the windowed scattering transform (4.2). The k-nearest-neighbours algorithm, hereby denoted KNN, will be used to classify the results. The handwritten letters come from a dataset of handwritten words collected by Rob Kassel at MIT Spoken Language Systems Group [9]. All the letters were extracted from the handwritten words, disregarding the structure imposed by the words. Capital letters were ignored, as including them would have made the problem more complicated. Therefore, the dataset consist of all letters in English alphabet, that is letters a-z. This results in a dataset consisting of 52152 letters from 26 classes. The number of images of each letter in the dataset is not equal. Some letters have more samples than other letters. Each letter is represented by a binary image of size 16x8 pixels, see Figure 6.1 for examples. The KNN algorithm will be explained in Section 6.1. Section 6.2 shows the results of the image recognition procedure and Section 6.3 discusses those results.

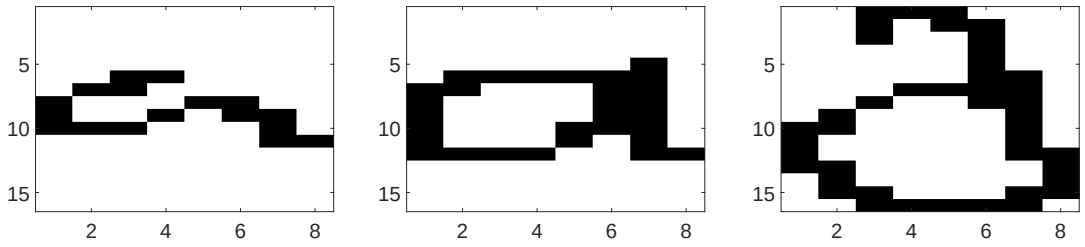


Figure 6.1: Three *a*-letters from the dataset of handwritten letters [9]. Notice that the *a*-letter in image (c) has a different font.

6.1 k-nearest neighbors algorithm

Image recognition is a classification problem, where the goal is to determine which object the images are depicting, or in other words which class the images belong to. Two standard approaches when performing classification are supervised learning and unsupervised learning. A supervised learning algorithm is defined as an algorithm that makes a model based on data where the class of the data is known, called training data, like the handwritten dataset presented earlier. The model can then be used to predict the classes of unknown data. For the model to be successful, all the training data from each class has to be sufficiently representative of the variance within its class. An unsupervised learning algorithm uses data with unknown classes. The unsupervised algorithm classifies the unknown data, and a model is made based on those classifications. An unsupervised learning algorithm essentially aims to partition data into meaningful classes. Since it is known which letter each image in the database of handwritten letters correspond to, we will use supervised learning.

The k-nearest neighbors algorithm is a supervised classification algorithm. For $k = 1$ the class of an unknown datapoint will be equal to the class of the nearest datapoint in the model. For k larger than one, the class of an unknown datapoint will be equal to whichever class has the majority among its k neighbors. In the case of a tie, either in majority or distance, there exists several possible tiebreakers. There are also several possible metrics that can be used to compute the distances. In this thesis, we will use Euclidean distance as the metric, and the closest-neighbor-rule will break ties. The closest-neighbor-rule states that the class will be equal to the class of the closest neighbor among the tied neighbors. In the rare case that the distances to different classes are perfectly equal, the class will be chosen randomly from the tied neighbors. The KNN-algorithm was chosen for thesis because it is simple and easy to interpret.

Choosing k too small will lead to overfitting, that is a small perturbation in a new datapoint might lead to a different classification. Choosing k too large will lead to underfitting and the model's rules for classifying new datapoints might become too simple. Ideally, we want some middle ground k. The simplest way to determine the best k is through trial and error.

6.2 Results

The first step of the image recognition procedure is to compute the widowed scattering transform (4.2) of all images. This new dataset will be called the dataset of scattering coefficients. The next step is then to split the dataset of scattering coefficients into a training set and a test set. According to the Pareto principle [10], 80% of the variance in a dataset comes from 20% of the samples. Therefore 20% of the images of each letter will be used as the test set. These images are chosen randomly. Then finally, after using the KNN algorithm, the error rate can be computed. The error rate is defined as the number of misclassified images in the test set divided by the number of test images. A flowchart of the image recognition procedure can be seen in Figure 6.2.

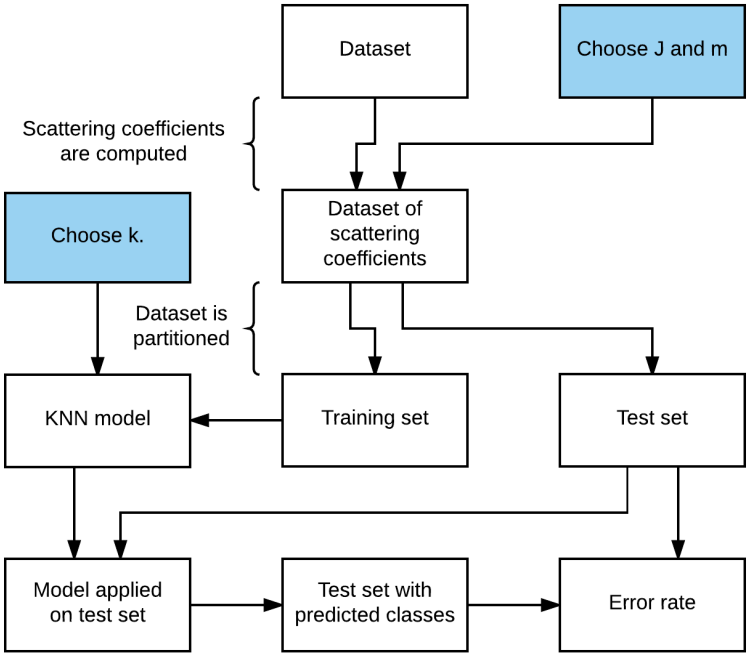


Figure 6.2: Flow chart of the image recognition procedure. Blue boxes indicates parameters that have to be chosen.

Notice that, the parameters J and m in the windowed scattering transform have to be chosen, as well as the number of considered neighbors k in the KNN algorithm. Rotation will not be considered as none of the letters in the dataset are rotated. Another benefit when not considering rotations is that computational cost is reduced. In this chapter, the error rate showed in different figures need not coincide for similar parameters. This is because the partitioning of the dataset into test and training sets is determined randomly.

Figure 6.3 shows the error rate for different values of J and k. For all these error rates $m = 2$. The same test and training set was used for all values of k and J. We observe that $k = 7$ and $J = 2$ gives the lowest error rate.

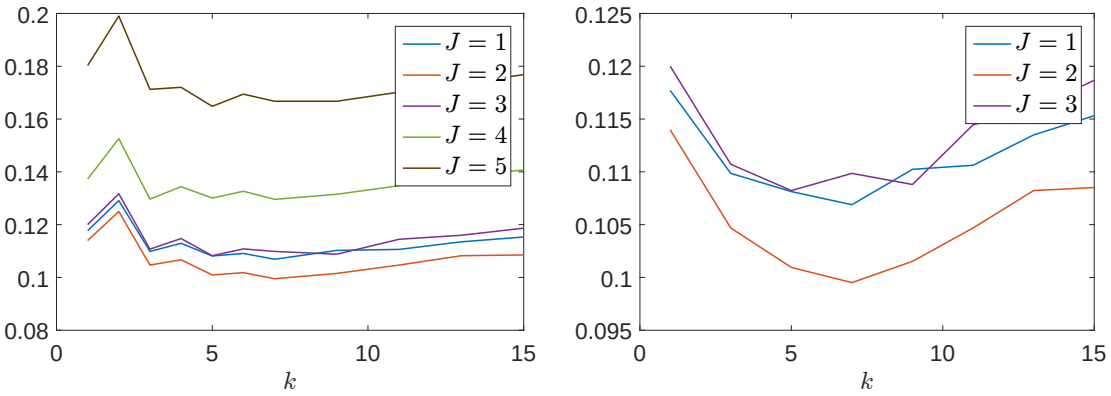


Figure 6.3: Error rates of the image recognition procedure as a function of k for several values of J. The right subfigure is the same has the left subfigure, except that some lines have been removed and even values of k have been excluded.

Figure 6.4 shows the error rate for different values of m and k. For all these error rates $J = 2$. The same test and training set was used for all values of m and J. We observe that $k = 7$ and $m = 2$ gives the lowest error rate. For all the remaining results we chose $m = 2$, $J = 2$ and $k = 7$, since those values gave the lowest error rate.

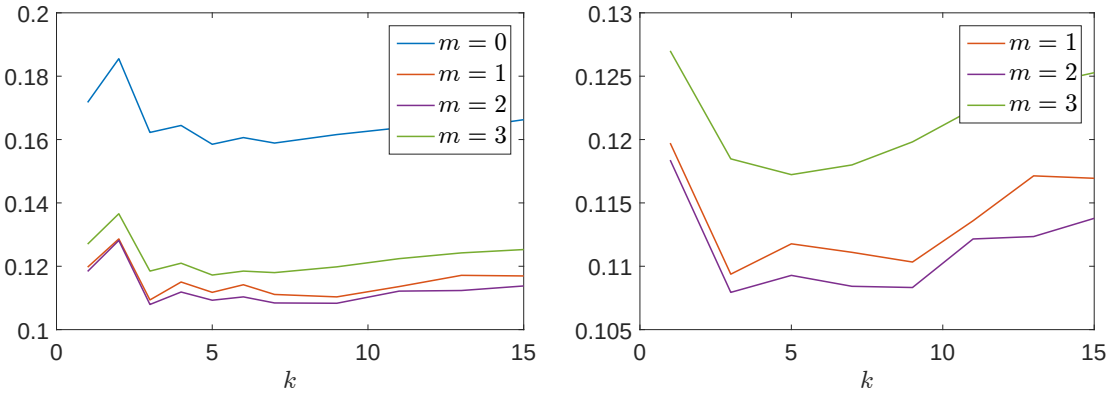


Figure 6.4: Error rates of the image recognition procedure as a function of k for several values of m. The right subfigure is the same has the left subfigure, except that some lines have been removed and even values of k have been excluded.

Figure 6.5 shows the error rates of the procedure as a function of the size of the training set. Figure 6.5 also shows the error rate of the KNN algorithm applied directly on the dataset of images without using the windowed scattering transform (4.2). For each size of training set data, the training set is a subset of all larger training sets. The same test set was used for all the different training set sizes.

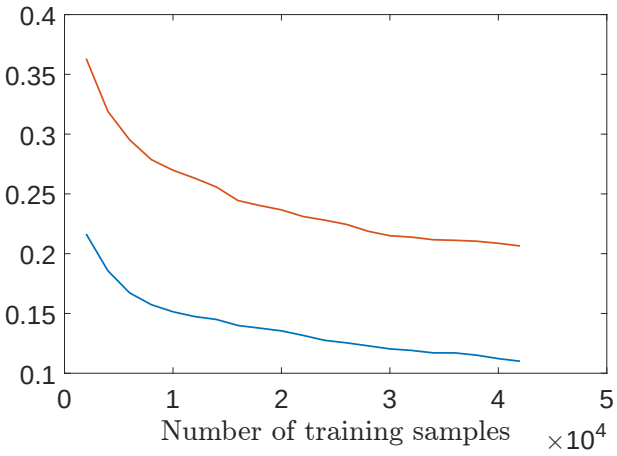


Figure 6.5: Error rates of the image recognition procedure as a function of the number of images in the training set. The blue line is KNN applied on the scattering coefficients. The red line is KNN applied directly on the images.

Figure 6.6 shows a confusion matrix. The procedure was performed on all the letters, but for the sake of printing, only the parts of interest from the confusion matrix has been included. The summarized confusion matrix consists of those letters that were difficult to classify correctly. The columns and the rows of the matrix are numbered by letters. The square on the intersection of column x and row y contains two numbers. The upper number shows how many times x has been classified as y, and the lower numbers shows the percentage of such classifications among the total number of test images. The diagonal (green) show the number of correct classifications for each letter. The bottommost row (blue) shows the percentages of correct and incorrect classifications for each letter. The rightmost column (blue) show how many percents of the classifications for one letter that was correct. For example, out of all the a-letter classifications that were made, the top right square shows how many percent of those classifications were correct. In the bottom right corner (yellow) the total error rate is shown.

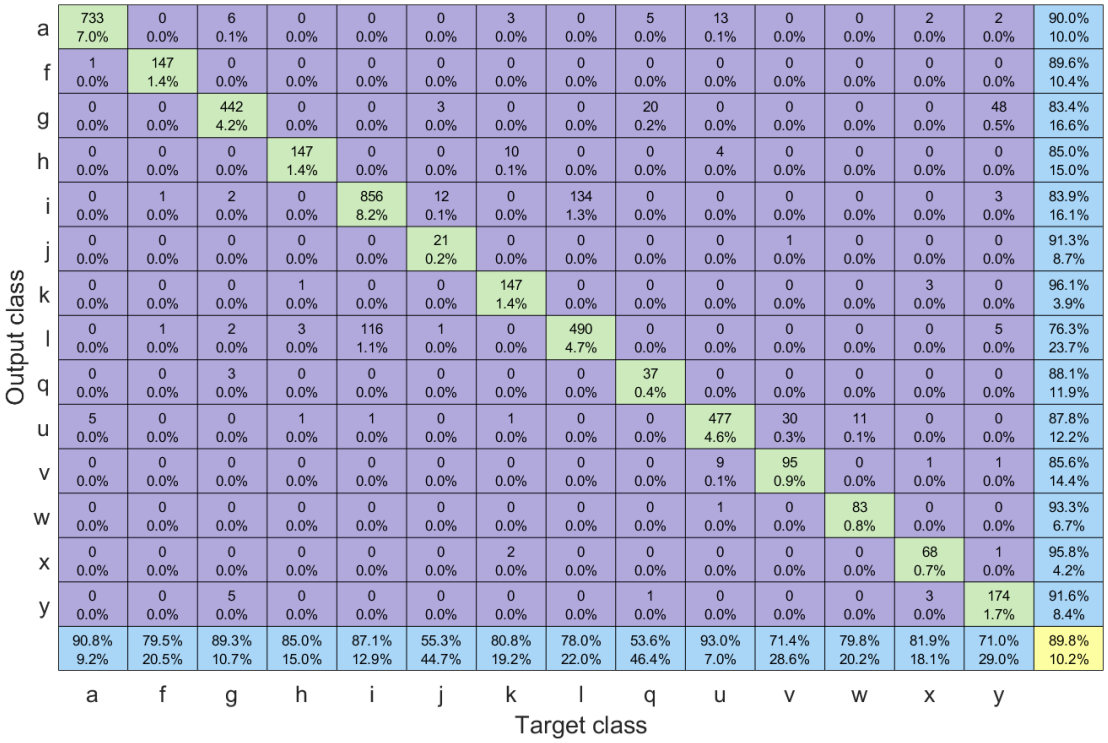


Figure 6.6: Confusion matrix displaying the number of correct and incorrect classifications for some letters.

6.3 Discussion

In this section, we will first discuss how the different parameters affected the error rates. Then the largest contributors to the error rates are discussed. Finally, the error rates achieved in this thesis is compared with results from other research.

From Figure 6.4 we see that choosing $m = 2$ layers gives the lowest error rate. Choosing $m = 2$ yields a small improvement over choosing $m = 1$, but at the cost of substantially increasing the runtime of the procedure. The expectation was that larger m would give lower error rates. It was not expected that $m = 1$ and $m = 2$ would be so close. It is also surprising that choosing $m = 3$ gives such high error rates. For increasing m, the number of scattering coefficients increases exponentially, this could result in a weaker KNN-model as the dimension grows to large. It would be interesting to investigate why $m = 3$ did not yield a better error rate.

From both Figure 6.3 and 6.4 we see that odd k are better than even k. When k is even, the tie break rules is employed more often, which results in poorer accuracy. Testing has shown that $k = 7$ was often the best choice, but choosing $k = 3$, $k = 5$ or $k = 7$ gives similar error rates subject to variations in the partitioning

of the test and training set. There are other options for KNN besides choosing k. Several metrics can be chosen, and some of them might yield better results, but only Euclidean distance have been used in this thesis. When the number of samples available to each class varies, it is possible to apply a weighting in the KNN model. This was briefly attempted, but did not improve the error rates.

The scale J could be chosen according to the image size of 16x8 pixels, but as seen in Figure 6.3, choosing scale $J = 2$ unexpectedly gave the lowest error rate. Scale $J = 2$ gives translation invariance within a frame of size $2^J = 4$ pixels. We learn that most of the variation within one class can be captured in a frame of size 4x4 pixels. The increased translation invariance when $J > 2$ makes it more difficult to distinguish the classes. Choosing $J = 1$ gives translation invariance within a frame of 2x2 pixels, but this small frame is too small to capture the variance within one class of letters and gives a poorer error rate than choosing $J = 2$.

A significant portion of the errors come from letters that are similar, like g and y, and especially i and l. Similar letters will always pose a problem. A possible solution could be to increase the number of training samples for the letters that are similar. This solution is not guaranteed to improve the error rates, and for this database, all the letters have been employed already. From the confusion matrix, Figure 6.6, we see that i-letters was classified as l-letters 116 times, and l-letters was classified as i-letters 134 times. To better identify these two letters, it is possible to investigate if another value of J or m could better distinguish these two letters. However, switching parameters would lead to overall poorer error rates.

A possible solution to similar letters is to apply the windowed scattering transform more than once. First, for a set of parameters the scattering coefficients are computed for all samples. Then, whenever a letter is classified as an i or an l, apply the windowed scattering transform with a different set of parameters. These parameters would be fine-tuned to distinguish i and l. Applying the windowed scattering transform once more should be beneficial when two letters are easily confused with each other, but this solution does not work if a letter is easily confused with several other letters. Instead of applying the windowed scattering transform more than once, another option is to use other algorithms. For example, to better distinguish i and l one could check the connectivity of letters. l-letters are connected, while i-letters are not.

Some letters like f, h, j, k, q, v, w and x are represented only by a few images in the dataset. From the confusion matrix, Figure 6.6, we see that these letters that have few samples have poor error rates. However, a dataset that contains

equally many images from each class will not necessarily give better error rates. The dataset needs to have a large amount of images for each letter. This can be seen in Figure 6.5, which shows that having few training samples results in poor error rates.

For some letters in the dataset there are multiple fonts, which makes it more difficult to get accurate classifications. For example, there are two kinds of *a*-letters in the set, *a* and a, see Figure 6.1 for examples. However, from the confusion matrix 6.6 we see that the class of *a*-letters have an error rate of 9.2% which is below the total error rate of 10.2%. Therefore these different kinds of fonts do not have a huge impact on the error rate. This discovery was a bit surprising as it was expected that the different fonts would lead to poorer error rates. For few samples, if both fonts are not represented in the test and training set, the different fonts could increase the error rate, but the chance for this to happen is negligible

Classification on handwritten letters is closely related to classification of handwritten digits, and in order to compare with similar procedures, we compare error rates with results from classification of handwritten digits. The aim was to do something new by doing classification on handwritten letters, but in retrospect, it would have been better to do classification on handwritten digits. Then the comparisons of error rates would be more meaningful.

In previous research [4], the windowed scattering transform was applied on the MNIST database of handwritten digits [11]. Two different classification algorithms were used, support vector machines and principal component analysis. When a support vector machine was used for classification, an error rate of 0.70% was achieved, whereas the principal component analysis classification achieved an error rate of 0.72%. Reference [11] lists the error rates of several artificial neural networks, many of which acquired error rates below 1% on the MNIST dataset.

In Reference [2] an error rate of 1.6% was achieved on the MNIST dataset using KNN and some preprocessing of the images. In Figure 6.5 we see that applying KNN directly on the images yields an error rate of 20.6%. The error rate of 10.2% achieved in this thesis is better than the error rate when only KNN is applied to the images, but it is not better than KNN with preprocessing or the windowed scattering transform with another classifier. In this thesis, KNN was used in its simplest form. The author expects that a more sophisticated classification algorithm or a more advanced variant of KNN would yield better results. It would be interesting to investigate how the error rate is affected by choice of classification algorithm, but it is not in the scope of this thesis.

The error rate of 10.2% in this thesis was achieved using the simple classifier KNN, and was run on a home computer, which is not comparable to results from more advanced research. Other research [2, 4, 11] have achieved error rates that are vastly better than the one produced in this thesis. It is harder to classify letters than to classify digits because there are more classes, that is 26 compared to 10 classes. The images in the MNIST dataset are relatively evenly distributed with respect to the classes, as opposed to in the database of handwritten letters [9], in which some classes contain significantly more samples. Combined with the fact that the MNIST database has more images, poorer error rates are to be expected on the database of handwritten letters [9]. However, fewer samples and the additional classes only account for some of the difference in error rates. The windowed scattering transform may be thought of as a pre-conditioning that we can apply a classification algorithm on. Refinement and additional steps, both in preprocessing and classification that is seen in other research, are necessary in order to produce the best error rates. By refining and adding additional steps to the straightforward approach in this thesis, comparable error rates should be attainable.

Chapter 7

Conclusion and future work

In this thesis, the windowed scattering transform and its underlying theory were defined in detail. Examples of the capabilities of the transform were given. The transform was used to perform image recognition on handwritten letters, and the k-nearest neighbors algorithm was used for classification. An error rate of 10.2% was achieved. The error rate is high compared to other research, where error rates below 1% were obtained on a dataset of handwritten digits using the windowed scattering transform [4] and using artificial neural networks [11]. It was discovered that the scale 2^J should not be chosen based on the size of the images. The scale should rather be chosen according to the size of the variation within each class of letters. Furthermore, the number of images in the training set substantially impacts the error rates. Therefore, as many training samples as possible should be used in order to improve error rates. There were two main challenges which were the cause of the high error rates. There were more inaccurate classifications on classes with few samples. Also, it was challenging to distinguish letters like i and l, which are very similar. To better distinguish similar letters, an interesting solution could be to combine the windowed scattering transforms with other techniques.

For future work, it would be interesting to improve the error rates by combining the windowed scattering transform with other techniques. It would also be interesting to investigate other classification algorithms besides the k-nearest neighbors algorithm. With the aim of finding the classification algorithms best suited to be used in combination with the windowed scattering transform. Another option is to investigate the k-nearest neighbors algorithm further, in order to see if another metric would yield improved error rates. Results showed that $m = 3$ gave poorer error rates than $m = 2$, which is unexpected and should be studied further.

Bibliography

[1] Milton Abramowitz and Irene A Stegun. *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, volume 55. Courier Corporation, 1964.

[2] U. R. Babu, A. K. Chintha, and Y. Venkateswarlu. Handwritten digit recognition using structural, statistical features and k-nearest neighbor classifier. *International Journal of Information Engineering and Electronic Business*, 6(1):62, 2014.

[3] Adam Bowers and Nigel J. Kalton. *An introductory course in functional analysis*. Springer, 2014.

[4] Joan Bruna and Stéphane Mallat. Classification with scattering operators. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1561–1566. IEEE.

[5] John B. Conway. *A course in functional analysis*, volume 96. Springer Science & Business Media, 2013.

[6] Mats Ehrnstrom. Lecture notes in tma4145 - linear methods. 2014.

[7] Glenn Elert. Frequency range of human hearing. *The Physics Factbook*, 2003.

[8] Joel Hass, Maurice D. Weir, and George B. Thomas. *University calculus*. Pearson Addison-Wesley Boston, 2007.

[9] Rob Kassel. Ocr dataset. Accessed: 7 June 2017. `http://ai.stanford.edu/~btaskar/ocr/`.

[10] Ankunda R. Kiremire. The application of the pareto principle in software engineering. *Consulted January*, 13:2016, 2011.

[11] Yann LeCun et al. Mnist handwritten digit database. Accessed: 1 June 2017. `http://yann.lecun.com/exdb/mnist/`.

[12] Stéphane Mallat. *A wavelet tour of signal processing: the sparse way*. Academic press, 2008.

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

[14] Tom M. Mitchell. *Machine learning*. McGraw-Hill series in computer science, Artificial intelligence. McGraw-Hill, New York, 1997.

[15] Laurent Sifre et al. Scatnet : a matlab toolbox for scattering networks. 2013.

[16] Terence Tao. *An introduction to measure theory*, volume 126. American Mathematical Soc., 2011.

Appendix A

Results from mathematical analysis

Definition A.1. [6] The *Cartesian product* of two sets A and B, is the set of ordered pairs (a, b) of elements $a \in A$ and $b \in B$, such that $A \times B = \{(a, b) : a \in A, b \in B\}$.

Theorem A.2. [16] (Fubini's Theorem). *Let (X, B_X, μ_X) and (Y, B_Y, μ_Y) be complete σ-finite measure spaces, and let $f : X \times Y \to C$ be absolutely integrable with respect to the closure of $B_X \times B_Y$. Then*

1. *For μ_X-almost every $x \in X$, the function $y \to f(x, y)$ is absolutely integrable with respect to μ_Y, and in particular $\int_Y f(x, y) d\mu_Y(y)$ exists. Furthermore, the (μ_X-almost everywhere defined) map $x \to \int_Y f(x, y) d\mu_Y(y)$ is absolutely integrable with respect to μ_X.*

2. *For μ_Y-almost every $y \in Y$, the function $x \to f(x, y)$ is absolutely integrable with respect to μ_X, and in particular $\int_X f(x, y) d\mu_X(x)$ exists. Furthermore, the (μ_Y-almost everywhere defined) map $y \to \int_X f(x, y) d\mu_X(x)$ is absolutely integrable with respect to μ_Y.*

3. *We have*

$$\begin{aligned}\int_{X \times Y} f(x, y)\, \overline{B_X \times B_Y}(x, y) &= \int_X \left(\int_Y f(x, y)\, d\mu_Y(y) \right) d\mu_X(x) \\ &= \int_Y \left(\int_X f(x, y)\, d\mu_X(x) \right) d\mu_Y(y).\end{aligned}$$

Definition A.3. [5] A *Hilbert space* is a vector space $\mathcal{H}$ over $\mathbb{R}$ or $\mathbb{C}$ together with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ such that relative to the metric $d(x, y) = ||x - y||$ induced by the norm, $\mathcal{H}$ is a complete metric space.

Proposition A.4. [5] *Let $\mathcal{H}_1, \mathcal{H}_2, \ldots$ be Hilbert spaces, let*

$$\mathcal{H} = \{(h_n)_{n=1}^{\infty} : h_n \in \mathcal{H}_n \, \forall n \text{ and } \sum_{n=1}^{\infty} ||h_n||^2 < \infty\}.$$

For $h = (h_n)$ and $g = (g_n)$ in $\mathcal{H}$, define

$$\langle h, g \rangle_{\mathcal{H}} = \sum_{n=1}^{\infty} \langle h_n, g_n \rangle_{\mathcal{H}_n}.$$

Then $\langle \cdot \, , \cdot \rangle$ is an inner product on $\mathcal{H}$ and the norm relative to this inner product is $||h|| = (\sum_{n=1}^{\infty} ||h_n||^2)^{1/2}$. With this inner product $\mathcal{H}$ is a Hilbert space.

Theorem A.5. [3] (Minkowski's inequality). *If x and y are elements of an inner product space, then $||x + y|| \leq ||x|| + ||y||$.*

Theorem A.6. [3] (Hölder's inequality). *Let (X, μ) be a positive measure space. Suppose $1 \leq p < \infty$ and let $1/p + 1/q = 1$. If $f \in L_p(\mu)$ and $g \in L_q(\mu)$, then $fg \in L_1(\mu)$ and $\|fg\|_1 \leq \|f\|_p \|g\|_q$.*

Theorem A.7. [8] (Taylor's Theorem). *If f and its first n derivatives $f', f'', \ldots, f^{(n)}$ are continuous on the closed interval $[a, x]$, and $f^{(n)}$ is differentiable on the open interval (a, x), then there exist a number c between a and x such that*

$$\begin{aligned} f(x) =& f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \ \ldots \\ &+ \frac{f^{(n)}(a)}{n!}(x-a)^n + \frac{f^{(n+1)}(c)}{(n+1)!}(x-a)^{n+1}. \end{aligned}$$

Remark A.8 . A function f can be approximated to order $m < n$ using Taylor's Theorem A.7. We get the approximation

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \ \ldots \ + \frac{f^{(m)}(a)}{m!}(x-a)^m.$$

The approximation is valid for x sufficiently close to a, because then the discarded terms are small compared to the approximation, that is

$$\frac{f^{(m+1)}(a)}{(m+1)!}(x-a)^{m+1} + \ \ldots \ + \frac{f^{(n+1)}(c)}{(n+1)!}(x-a)^{n+1} \ll f(a) + \ \ldots \ + \frac{f^{(m)}(a)}{m!}(x-a)^m.$$

In order to get the best approximation, the order m should be chosen as large as possible. However, often the aim is to simplify as much as possible and m is chosen small, even though smaller m's gives poorer approximations.

Definition A.9. [1] Stirling's approximation. Let θ be a number such that $0 < \theta < 1$. Then for all $x > 0$ we have that $x! = \sqrt{2\pi} \, x^{x+\frac{1}{2}} \, e^{-x} \, (1 + o(1))$.