An Analysis of Stockwell Transforms, with Applications to Image Processing

by

John Ladan

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Applied Mathematics

Waterloo, Ontario, Canada, 2014

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Time-frequency analysis is a powerful tool for signal analysis and processing. The Fourier transform and wavelet transforms are used extensively as is the Short-Time Fourier Transform (or Gabor transform). In 1996 the Stockwell transform was introduced to maintain the phase of the Fourier transform, while also providing the progressive resolution of the wavelet transform. The discrete orthonormal Stockwell transform is a more efficient, less redundant transform with the same properties.

There has been little work on mathematical properties of the Stockwell transform, particularly how it behaves under operations such as translation and modulation. Previous results do discuss a resolution of the identity, as well as some of the function spaces that may be associated with it [2]. We extend the resolution of the identity results, and behaviour under translation, modulation, convolution and differentiation. Boundedness and continuity properties are also developed, but the function spaces associated with the transform are unrelated to the focus of this thesis.

There has been some work on image processing using the Stockwell transform and discrete orthonormal Stockwell transform. The tests were quite preliminary.

In this thesis, we explore some of the mathematics of the Stockwell transform, examining properties, and applying it to various continuous examples. The discrete orthonormal Stockwell transform is compared directly with Newland's harmonic wavelet transform, and we extend the definition to include varitions, as well as develop the discrete cosine based Stockwell transform. All of these discrete transforms are tested against current methods for image compression.

Acknowledgements

I would like to thank Dr. E.R. Vrscay for his support and patience.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Time-frequency analysis is a significant tool in signal processing. Using tools like the Fourier transform, short-time Fourier transform, and wavelets, it is possible to break a signal into simple components for analysis. Often, a signal represents some quantity varying in time, such as sound or mechanical vibrations, in which time and frequency are clear. These methods have also been used much for image processing, because an image can be represented as a two-dimensional signal.

The Fourier transform (FT) is the most established of these transforms. Its components are well understood as vibrations, and have a clear physical meaning. However, the vibrations are throughout all time, so transient behaviour such as a sequence of musical notes or edges in a picture are not well represented under this transform.

In order to capture transient behaviour, often the short-time Fourier transform (STFT) or a wavelet transform is used. The STFT involves performing the FT on smaller sections of a signal to get the spectrum locally. This has a clear relation to the concepts from the FT, but the range of frequencies that can be resolved this way is limited. Wavelet transforms measure local details in a signal at a wide range of scales. This progressive resolution catches more behaviour than the STFT, but the detail that is measured is not directly analogous to the frequency and phase of the FT.

The Stockwell transform (ST) was first introduced in 1996 as a bridge between STFT and continuous wavelet transform [24]. By adopting the progressive resolution of wavelets (using larger windows for lower frequencies), it is able to resolve a wider range of frequencies and changes in frequencies than the STFT. By using a Fourier-like basis and maintaining a phase of zero at $t = 0$ as in the STFT, Fourier based analysis could be performed locally. Since that time, the ST has been used in many applications in signal analysis [9].

A drawback of the ST is that it is highly redundant. For the discrete Stockwell transform (DST), an input of length N will return an output of size N^2. This quadratic growth in size is very restrictive, and prevents use of the DST on large signals.

Stockwell later introduced, in 2006, an orthogonal transform with the same characteristic progressive resolution and absolutely referenced phase as the ST [23]. This transform is referred to as the discrete orthonormal Stockwell transform (DOST). Because the DOST is an orthogonal transform, its output is the same size as the input, permitting it to be used on larger signals.

The DOST has been applied to texture analysis [7], medical imaging [3], and many of the same fields as the ST. More recently, Y. Wang worked with Prof. J. Orchard to apply the DOST to image processing applications [25]. The fast version of the transform they developed [28] improves computation time (the same algorithm was used elsewhere [7], but was not explicitly written). They also developed a conjugate-symmetric version [26] and tested the DOST for image compression [29] and restoration [27].

The preliminary results of Y. Wang were quite favourable, showing that the DOST might perform well compared to wavelet transforms and FT. With my supervisor, E.R. Vrscay, I performed more extensive tests for image compression. These tests are summarized in Chapter 7

Chapter 2 covers the Stockwell transform. A more general version of the ST along with inversion formulas are presented. We test the ST on some functions and invert them to test it. Properties of the ST are summarized and proven in Section 2.3. The proofs are mostly simple calculations and provided for details, but may be skipped on first reading.

The DST is defined in Chapter 3. My work has focused on ensuring higher-dimensional implementations are well-defined and invertible. Some frame properties are discussed, and a fast-synthesis operator, which I developed, is defined.

The bulk of my work was with the DOST. Chapter 4 introduces the DOST, and covers previous work, including variations of the transform. I offer a new interpretation of the DOST, which allows easy modification to obtain fast transforms for all of its variations. This interpretation is used to develop the discrete cosine Stockwell transform (DCST) in Chapter 5, which is my own work.

The DOST is quite similar to Shannon wavelets, and nearly identical to the harmonic wavelet transform (HWT) introduced by Newland [16] in the early 1990's. The primary application was ridge and phase identification [18] in transient signals, like in beam vibration and acoustic duct measurements. In Chapter 6, I compare the DOST to HWT, and show that they are identical up to a phase correction. There are more differences in the 2D versions, particularly in the DCT-based adaptations of the DOST and HWT.

The primary application I investigated was image compression. This is covered in Chapter 7, which includes my results from [13], but goes into more detail. Compression artifacts, and additional results with the HWT are also examined in that chapter.

1.1 Signals and Images

A *signal* is represented as a function from $\mathbb{R}^d$ to $\mathbb{C}$ (usually the signal is real-valued, but the theory works for general complex signals), where d is the dimension of the signal. An audio signal is an example of a one-dimensional signal, where the function value is the amplitude at that point in time. A greyscale image can be represented as a two-dimensional signal where the value of the function represents the greyscale level at a point.

To make calculations with multi-dimensional signals easier, we will adopt some specific notation. Given two vectors $x, y \in R^d$ with components x_i, y_i for $i = 1, \ldots, d$, define the following operations:

$$x \cdot y = \sum_{i=1}^{d} x_i y_i \,, \qquad x^2 = x \cdot x$$

$$(x \circ y)_i = x_i y_i \,, \text{ and} \qquad |x| = \prod_{i=1}^{d} |x_i| \,.$$

In analysis of signals, it is often assumed that the signal is over a continuous variable ($t \in \mathbb{R}^d$), however in practice a signal is sampled at discrete points. This sampling typically has a constant sample-rate T. For a continuous signal, we will use round brackets for evaluation (i.e. $f(t)$), and for discrete signals, we use square brackets, $f[n]$. As an example, if the continuous signal $f(t)$ is sampled with a sampling period T, then the discrete signal is

$$f[n] = f(Tn), \qquad t \in \mathbb{Z}.$$

The sampled signals discussed in this thesis are also sampled on a finite number of points, so $f[\cdot]$ behaves as an N-dimensional vector over $\mathbb{C}$. Furthermore, it is typically assumed that signals are elements of an inner product space, with inner products,

$$\langle h, g\rangle = \int_{\mathbb{R}^d} h(x)\overline{g(x)}\,\mathrm{d}x \qquad \text{(continuous)}$$

$$\langle x, y\rangle = \sum_{k=1}^{N} x[k]y[k] \qquad \text{(discrete)}.$$

1.2 Fourier Transform

The Fourier transform (FT) is a transform that decomposes a function into frequencies. We will use the following definition of the FT,

$$\hat{h}(\omega) = \mathcal{F}h(\omega) = \int_{\mathbb{R}^d} h(t)e^{-i2\pi\omega\cdot t}\,\mathrm{d}t. \tag{1.1}$$

This definition is convenient because the inverse transform has the same form, but with a positive exponent instead of a negative. It is also normalized, so there are fewer factors in the calculations. Furthermore, there is the benefit that this definition is used by Stockwell in his papers [24], and used by Gröchenig [12].

As in (1.1), we will denote the FT of a function with a hat. When it is convenient to treat the transform as an operator, $\mathcal{F}$ is used. Often the FT will only be applied to one variable. This partial FT will be subscripted with the index of the variable it transforms. For example,

$$\mathcal{F}_2 h(x,\omega) = \int_{\mathbb{R}^d} h(x,t)e^{-i2\pi\omega t}\,\mathrm{d}t.$$

Parseval's identity will be important in the following material. It states that the FT is a unitary form, so the inner product is preserved $\langle h, g\rangle = \langle \mathcal{F}h, \mathcal{F}g\rangle$. The shift, modulation and convolution theorems will also be used. These can be found in the introductory material in [12] or any other introductory material to the FT.

A weakness (for analysis of transient signals) of the FT is that it is a global transform, and measures the frequency response over the entire domain, i.e. $\mathbb{R}^d$. For functions with time-varying frequencies, time-frequency transforms are needed.

1.3 Time-Frequency Transforms

Time-frequency analysis is concerned with how the frequency of a function (or signal) behaves in time. Whereas the FT only provides global information, a time-frequency representation provides local frequency information, so that transient behaviour can be examined.

The most notable example of a time-frequency transform is the short-time Fourier transform (STFT), also known as the windowed Fourier transform or Gabor transform. The STFT is calculated by applying a window to the function at various times, before performing the FT [12],

$$V_g h(x,\omega) = \int_{\mathbb{R}^d} h(t)\overline{g(t-x)}e^{-i2\pi\omega\cdot t}\,\mathrm{d}t. \tag{1.2}$$

where $g(t)$ is the window function. Any window $g \neq 0$ can be used, but it is usually real-valued, positive and even-symmetric. Because the window has a fixed width, the STFT can only resolve a small range of times frequencies. Wider windows can identify lower frequencies, but narrower windows provide better time resolution.

Another transform that examines local behaviour of a function is the continuous wavelet transform. Wavelet transforms start with a mother wavelet which is scaled and translated before comparing it to the signal. Mathematically, this is

$$W_g h(x,s) = \int_{\mathbb{R}^d} h(t)s^{-d/2}g(s^{-1}(t-x))\,\mathrm{d}t, \tag{1.3}$$

where g is the mother wavelet, and s represents the scale or level of detail. The changing width of the wavelet improves resolution over a wide range of scales.

Technically, the continuous wavelet transform is a time-scale transform. If the mother wavelet has a modulation ($e^{-i2\pi\omega_0 t}$), then the scale is directly related to frequency by $\omega = \frac{\omega_0}{s}$. For example in Morlet wavelets, the mother wavelet is a modulated Gaussian, $g(x) = \pi^{-1/4}e^{-\frac{x^2}{2}}(e^{-i\omega_0 x} - e^{-\frac{\omega_0^2}{2}})$. As the wavelet is shifted, the phase of the wavelet changes with respect to the origin.

1.3.1 Common Operations

In time-frequency analysis there are some common operations of interest:

- Translation: $T_x : h(t) \mapsto h(t - x)$
- Modulation: $M_\omega : h(t) \mapsto e^{i2\pi\omega \cdot t} h(t)$
- Dilation: $D_a : h(t) \mapsto h(at)$

These operations can be composed to define the continuous transforms above. Each transform can be understood as a correlation to a set of functions, $\{\phi_{x,\omega} g : x, \omega \in \mathbb{R}^d\}$ using an inner product. As an example, the STFT can be defined as,

$$V_g h(x, \omega) = \langle h, M_\omega T_x g \rangle . \tag{1.4}$$

Each function that is correlated with the signal is often called an *atom*, though some will restrict that name to elements of a countable set of functions. The continuous wavelet transform atom is $s^{-1/2} T_x D_{1/s} g$, the FT simply uses a modulation $e^{i2\pi\omega \cdot t}$, and the STFT is uses the atom in 1.4.

1.4 Discrete Transforms

The derivation of discrete analogues to these transforms can be rather complicated. For the purposes of this thesis, we can "cheat" by simply replacing integration over a continuous domain with summation over the discrete sampling. The discrete Fourier transform (DFT) is then

$$\hat{f}[k] = \sum_{n=0}^{N} f[n] e^{-i \frac{2\pi}{N} kn}, \tag{1.5}$$

with inverse

$$f[n] = \frac{1}{N} \sum_{k=0}^{N} \hat{f}[k] e^{i \frac{2\pi}{N} kn}.$$

All of the discrete transforms can also be understood as correlating the signal with atoms from a *dictionary*. In this case, the set of atoms is finite.

It is preferable if a dictionary can perfectly reconstruct a signal, much like a basis for a vector space. However, in order to adequately discuss redundant transforms like the ST and STFT, the linear independence condition must be abandoned. Instead, the concept

of a *frame* is used. A dictionary $\{e_j : j \in J\}$ is called a frame if the following condition is satisfied:

$$A\|f\|_2^2 \leq \sum_{j \in J} |\langle f, e_j \rangle|^2 \leq B\|f\|_2^2,$$

for some constants A and B [12].

Frames are a way of generalizing the concept of a basis. A special case of a frame is an orthonormal basis, in which $A = B = 1$. The DFT has the basis $\{e_k[n] = \exp(i\frac{2\pi}{N}kn)\}$, and the discrete cosine transform (DCT) has a similar basis of cosines. There are special cases of the wavelet transform which have an orthonormal basis [5].

Every frame has a dual frame, which provides an inverse. Given a frame $\{\phi_j\}$ and its dual frame $\{\tilde{\phi}_j\}$, then

$$\sum_{j \in J} \langle f, \phi_j \rangle \tilde{\phi}_j = f = \sum_{j \in J} \left\langle f, \tilde{\phi}_j \right\rangle \phi_j.$$

The dual frame gives rise to a class of wavelets called *biorthogonal wavelets*. For details, see [5]. Biorthogonal wavelets are the family of wavelets used in the JPEG2000 standard.

Chapter 2

Stockwell Transform

The original definition of the ST comes from Stockwell's paper [24]. It is based on the continuous wavelet transform, and the STFT. Combining these two transforms, the ST was first defined for one-dimensional signals as

$$Sh(\tau,\omega) = \int_{\mathbb{R}} h(t)\frac{|\omega|}{\sqrt{2\pi}} e^{\frac{(t-\tau)^2\omega^2}{2}} e^{-i2\pi\omega t}\,\mathrm{d}t, \tag{2.1}$$

with $t, \tau, \omega \in \mathbb{R}$.

This ST is very similar to Morlet wavelets [10], which are shifted and modulated Gaussians [5]. However, because the modulation is not shifted in time, the phase of the atom does not change with respect to the origin as it does for the wavelet transform. Stockwell calls this phase "absolutely referenced", because at $t = 0$, the phase of the time-frequency atom is zero.

There have been several generalizations of the ST, extending to multiple dimensions, and using different window functions, g. The generalized definition of the ST we will use is,

$$S_g h(x,\omega) = \int_{\mathbb{R}^d} h(t)|\omega|\overline{g(\omega \circ (t-x))} e^{-i2\pi\omega\cdot t}\,\mathrm{d}t. \tag{2.2}$$

The changing window width is achieved with coordinate-wise multiplication. That is, $(\omega \circ x)_k = \omega_k x_k$. The original ST can be obtained from (2.2) by using $g(x) = \frac{1}{\sqrt{2\pi}} \exp(\frac{-x\cdot x}{2})$ as the window. This works in the one-dimensional case, as well as the multi-dimensional case.

2.1 Alternative Definitions

Some alternative definitions of the ST are useful for calculations, and examining the properties of the ST. The ST can also be described as an inner product with an atom [8], where the Stockwell atom is defined as

$$|\omega| M_\omega T_x D_\omega g,$$

using the operators as defined in Section 1.3.1. This is different from the wavelet transform, because modulation is not translated (see Section 1.3.1). It is also different from the STFT atom, because there is a dilation of the window.

The inner product definition of the ST is

$$S_g h(x, \omega) = \langle h, |\omega| M_\omega T_x D_\omega g \rangle .$$

The shift by x is reminiscent of a convolution. This may be more apparent by looking at the integral definition (2.2). The convolution form of the ST is

$$S_g h(\cdot, \omega) = |\omega| M_{-\omega} h * D_\omega g. \tag{2.3}$$

By the convolution theorem for the FT, this is equivalent to

$$(\mathcal{F}_1 S_g h)(\alpha, \omega) = \hat{h}(\alpha - \omega) \hat{g} \left(\frac{\alpha}{\omega} \right) . \tag{2.4}$$

This last definition for the ST is particularly useful for defining an efficient DST ($\mathcal{O}(N^2 \log N)$), but can also been used to examine various mathematical properties of the ST [2], compare it to wavelet transforms, and to calculate the ST of some functions as in Section 2.4.4.

The ST can also be related to the continuous wavelet transform through this definition. Recall from Section 1.3 that some wavelets are essentially a modulated window, $M_{\omega_0} g$. In this case, the modulation is on the right side of the convolution, so it is $\hat{g}$ that is shifted in (2.4) instead. This means that the two transforms are related by a factor of $e^{i2\pi\omega x}$, provided the wavelet is of that form.

2.2 Inverses of the Stockwell Transform

There have been multiple inversion formulas proposed for the ST. Stockwell originally introduced a rather simple one, which translates well to a fast discrete inverse [24]. Later, Shimmel and Gallart defined an inverse which should have better time locality for filtering applications [21]. There were also various forms based on the resolution of the identity [8, 4, 2].

For completeness, they are all presented here. In Section 2.2.3, I develop the resolution of the identity for the ST with more generality by considering a second window for the inverse, as is done for STFT and wavelets in [12]. By allowing a second window in the resolution of the identity, Stockwell's inverse can be seen as a special case.

2.2.1 Original Inverse

The first inversion formula was proposed by Stockwell when he introduced the ST [24]. Using the integral definition (2.2), integrating in x, and applying Fubini's theorem, the FT of the original signal is retrieved.

$$\begin{aligned} S_g h(x,\omega) &= \int_{\mathbb{R}^d} h(t)|\omega| g(\omega \circ (t-x)) e^{-i2\pi\omega\cdot t}\,\mathrm{d}t \\ \int_{\mathbb{R}^d} S_g h(x,\omega)\,\mathrm{d}x &= \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} h(t)|\omega| g(\omega \circ (t-x)) e^{-i2\pi\omega\cdot t}\,\mathrm{d}t\,\mathrm{d}x \\ &= \int_{\mathbb{R}^d} h(t) e^{-i2\pi\omega\cdot t} \int_{\mathbb{R}^d} |\omega| g(\omega \circ (t-x))\,\mathrm{d}x\,\mathrm{d}t \\ &= \|g\|_1 \hat{h}(\omega). \end{aligned}$$

The inversion formula is then

$$h(t) = \frac{1}{\|g\|_1} \mathcal{F}^{-1}\left(\int_{\mathbb{R}} S_g h(x,\omega)\,\mathrm{d}x \right). \tag{2.5}$$

2.2.2 Alternate Inversion Formula

Another method is proposed by Schimmel et al. [21]. They consider a "time-time" representation of the time-series multiplied by the window,

$$H(\tau, t) = h(t)\frac{g(\omega(t-\tau))}{g(0)}.$$

Note that the original signal is the slice along $t = \tau$, $h(t) = H(t, t)$. The ST is equal to the FT of H on the second variable, with appropriate scaling of the window. Thus,

$$\mathcal{F}_2 H(\tau, \omega) = \frac{1}{|\omega| g(0)} S_g h(\tau, \omega),$$

and

$$h(t) = \frac{1}{g(0)} \mathcal{F}_2^{-1} \left(\frac{S_g h(t, \omega)}{|\omega|} \right) (t, t). \tag{2.6}$$

That gives another inversion formula. The authors argue that this inversion formula has better time-locality of filters, however the demonstration they give fails for continuous variables. For Stockwell's inversion formula, they used a Dirac-delta filter, and for this formula, they used $F(0, \omega) = 1$, $F(\tau, \omega) = 0$ otherwise. However, the discrete version of this transform, as tested by the authors, did have time-locality under filtering.

2.2.3 Resolution of Identity

A common formulation for inverses of the STFT and continuous wavelet transform is the resolution of the identity. The resolution of the identity is similar to Parseval's theorem [12], in that it relates the inner products of the functions and windows/wavelets to the inner product of the transforms. In this section, the theory is developed in 1D for consistency with the sources, but it can be extended into $\mathbb{R}^d$. For the STFT, the relation is [12],

$$\langle V_{g_1} f_1, V_{g_2} f_2 \rangle_{\mathcal{L}^2(\mathbb{R}^{2d})} = \langle f_1, f_2 \rangle \overline{\langle g_1, g_2 \rangle}. \tag{2.7}$$

For the continuous wavelet transform (1.3), the relation is quite similar, but with different inner products [5, 12],

$$\langle W_{g_1} f_1, W_{g_2} f_2 \rangle_{\mathcal{X}_2} = \overline{\langle g_1, g_2 \rangle_{\mathcal{H}_{-1,0}}} \langle f_1, f_2 \rangle, \tag{2.8}$$

where the inner products are defined (using parameters a and b) as

$$\langle F, G\rangle_{\mathcal{X}_a} = \int_{\mathbb{R}}\int_{\mathbb{R}} F(x,s)\overline{G(x,s)}\frac{\mathrm{d}x\,\mathrm{d}s}{s^a}, \text{ and}$$
$$\langle f, g\rangle_{\mathcal{H}_{a,b}} = \int_{\mathbb{R}} \hat{f}(\omega)\overline{\hat{g}(\omega)}|\omega+b|^a\,\mathrm{d}\omega.$$

The orthogonality relation for the ST (2.9) uses the same inner products, but with different parameters. It is worth noting that if $a = 0$, the inner product for $\mathcal{H}_{0,b}$ is equivalent to that of $\mathcal{L}^2$.

For the resolution of identity formulas to hold, it is necessary that all the functions are in the correct spaces. In (2.7), it is sufficient that the functions are all in $\mathcal{L}^2$, but for (2.8) it is also necessary that $\langle g_1, g_2\rangle < \infty$. For wavelets, the orthogonality relation and inversion formula are often presented with $g_1 = g_2 = \psi$, forcing the condition $\hat{\psi}(0) = \int_{\mathbb{R}} \psi(t)\,\mathrm{d}t = 0$.

These inner product relations lead to inversion formulas (in the weak sense). For example, the continuous wavelet transform inversion is,

$$h(t) = \frac{1}{\langle g, \gamma\rangle_Y}\int_{\mathbb{R}}\int_{\mathbb{R}} W_g h(x,s) s^{1/2} T_x D_{\frac{1}{s}}\gamma(t)\frac{\mathrm{d}x\,\mathrm{d}s}{s^2}$$

There have been several definitions for the resolution of the identity for the ST, in one dimension [8], multiple dimensions [4], and using more generalized function spaces [2]. The more general treatment using different function spaces overcomes some problems associated with permissible windows. We use the same inner product definitions, but do not worry about the associated function spaces for now. In one dimension, the resolution of the identity of the ST is [2]

$$\langle S_{g_1} f_1, S_{g_2} f_2\rangle_{\mathcal{X}_a} = \overline{\langle g_1, g_2\rangle_{\mathcal{H}_{a-2,1}}}\,\langle f_1, f_2\rangle_{\mathcal{H}_{a-1,0}}\,, \tag{2.9}$$

using the same inner product definitions as before. The results in [8] correspond to $a = 1$. For the remainder of this thesis, we will restrict ourselves to this case, and define $\mathcal{X} = \mathcal{X}_1$ and $\mathcal{H} = \mathcal{H}_{-1,1}$.

The corresponding inversion formula to (2.9) is

$$h(t) = \frac{1}{\langle g, \gamma\rangle_{\mathcal{H}}}\int_{\mathbb{R}}\int_{\mathbb{R}} S_g h(x,\omega) M_\omega T_x D_\omega \gamma(t)\,\mathrm{d}x\,\mathrm{d}\omega, \tag{2.10}$$

with the condition that $\langle g, \gamma \rangle_{\mathcal{H}} = K$ for a finite $K \neq 0$. The division by $|\omega|$ is cancelled out by the normalization factor of the Stockwell atom.

In all sources, the inversion formula (2.10) was presented using the same window/wavelet ($g = \gamma$) [8, 4, 2]. With one window, the standard Gaussian window is inadmissible, because $\hat{g}(-1) \neq 0$. If the parameter $a = 2$ is used, then any $g \in \mathcal{L}^2$ can be used [2], but weak-equality of the inverse is with respect to $\langle \cdot, \cdot \rangle_{\mathcal{H}_{-1,0}}$ instead of $\mathcal{L}^2$.

Defining the inversion formula using two windows is more consistent with the definitions in [12], and makes the transform more flexible. The weaker condition on g and γ is that either $\hat{g}(-1) = 0$ or $\hat{\gamma}(-1) = 0$.

With two windows, it can be observed that the original inversion formula (2.5) is actually (2.10) with $\gamma(t) = 1$. In that case, the FT of γ is $\hat{\gamma}(\omega) = \delta(\omega)$, so

$$\begin{aligned} h(t) &= \frac{1}{\langle g, \gamma \rangle_{\mathcal{H}}} \int_{\mathbb{R}} \int_{\mathbb{R}} S_g h(x, \omega) M_{-\omega} T_{-x} D_{\omega} \gamma(t) \, \mathrm{d}x \, \mathrm{d}\omega \\ &= \frac{1}{\int_{\mathbb{R}} \hat{g}(\omega) \delta(\omega) \frac{\mathrm{d}\omega}{|\omega+1|}} \int_{\mathbb{R}} \int_{\mathbb{R}} S_g h(x, \omega) e^{i2\pi\omega t} \, \mathrm{d}x \, \mathrm{d}\omega \\ &= \frac{1}{\hat{g}(0)} \mathcal{F}^{-} 1 \int_{R} S_g h(x, \omega) \, \mathrm{d}x, \end{aligned}$$

which is identical to (2.5).

2.3 Properties of the Stockwell Transform

There are many mathematical properties that may be of interest. We begin with the behaviour of the ST under the various operators introduced in Section 1.3.1. The results are summarized in Table 2.1.

The reflection property means the ST preserves odd/even symmetries of a function. If h is even (odd), then $S_g h$ is even (odd) about $\tau = 0$ and $\omega = 0$.

The complex conjugate and reflection properties can be combined, so that if h is real-valued, then $S_g(h)$ is conjugate symmetric in ω about $\omega = 0$. That is, $S_g h(\xi, \omega) = \overline{S_g h(\xi, -\omega)}$.

If h is real and even, then $S_g h$ is conjugate symmetric about $\omega = 0$ and $\tau = 0$. This is different from the FT, for which it would be real-valued and symmetric.

Name	Form	Relation
Shift	$S_g(T_{t_0}h)(x,\omega)$	$e^{-i2\pi\omega\cdot t_0}S_g(h)(x-x_0,\omega)$
Dilation	$S_g(D_ah)(x,\omega)$	$S_g(h)\left(ax,\frac{\omega}{a}\right)$
Reflection	$S_g(h(-t))(x,\omega)$	$S_g(h)(-x,-\omega)$
Complex Conjugate	$S_g(\bar{h})(x,\omega)$	$\overline{S_g(h)(x,-\omega)}$
Derivative	$S_g(\partial_t h)(x,\omega)$	$\left[i2\pi\omega S_g(h)+\partial_x S_g(h)\right](x,\omega)$

Table 2.1: Summary of ST properties

A significant theorem for the FT is the modulation theorem. For the ST, the modulating the function would shift the signal in frequency without re-scaling the window. This results in quite complicated behaviour, preventing a simple modulation theorem.

The convolution theorem is also lacking, because the time shift interferes with the window. The effect of convolution is complicated, and presented in Section 2.3.2.

2.3.1 Proofs

This section is dense with calculations, and can be skipped if the reader is not interested in the details. Each property can be proven with the integral definition, and the inner product definition. Only the integral derivation is shown, because using inner products would require deriving the adjoints and commutation of all of the operators as well.

Shift

$$\begin{aligned} S_g\big(h(t-t_0)\big)(x,\omega) &= |\omega|\int_{\mathbb{R}^d} h(t-t_0)g(\omega\circ(t-x))e^{-i2\pi\omega t}\,\mathrm{d}t \\ &= \int_{\mathbb{R}^d} h(s)g(\omega(s-x+t_0))e^{-i2\pi\omega(s+t_0)}\,\mathrm{d}t \\ &= e^{i2\pi\omega t_0}S_g(h)(x-t_0,\omega) \end{aligned}$$

Dilation

$$\begin{aligned} S_g\big(h(at)\big)(x,\omega) &= |\omega| \int_{\mathbb{R}^d} h(at) g(\omega \circ (t-x)) e^{-i2\pi\omega t} \,\mathrm{d}t \\ &= \int_{\mathbb{R}^d} h(s) \left|\frac{\omega}{a}\right| g\left(\omega\left(\frac{s}{a} - x\right)\right) e^{-i2\pi\omega\frac{s}{a}} \,\mathrm{d}s \\ &= \int_{\mathbb{R}^d} h(s) \left|\frac{\omega}{a}\right| g\left(\frac{\omega}{a}(s - ax)\right) e^{-i2\pi\frac{\omega}{a}s} \,\mathrm{d}s \\ &= S_g(h)\left(ax, \frac{\omega}{a}\right) \end{aligned}$$

Evenness and Oddness Properties

The proof of preservation of the even/odd is dependent on the window being an even function. We present the proof of an even function here,

$$\begin{aligned} S_g\big(h(t)\big)(x,\omega) &= \int_{\mathbb{R}^d} h(-t)|\omega| g(\omega(t-x)) e^{-i2\pi\omega t} \,\mathrm{d}t \\ &= \int_{\mathbb{R}^d} h(s)|\omega| g(-\omega(s+x)) e^{-i2\pi(-\omega)s} \,\mathrm{d}t \\ &= S_g(h)(-x,-\omega). \end{aligned}$$

For odd functions h, the proof is the same except that $h(t) = -h(-t)$ in the first step.

Complex Conjugate

Complex conjugacy requires the window function to be both real-valued and even.

$$\begin{aligned}
\overline{S_g(h)}(x,\omega) &= |\omega| \int_{\mathbb{R}^d} \overline{h(t)} g(\omega \circ (t-x)) e^{-i2\pi\omega t} \,\mathrm{d}t \\
&= |\omega| \int_{\mathbb{R}^d} h(t) \overline{g(\omega \circ (t-x)) e^{-i2\pi\omega t}} \,\mathrm{d}t \\
&= |\omega| \int_{\mathbb{R}^d} h(t) g(-\omega \circ (t-x)) e^{i2\pi\omega t} \,\mathrm{d}t \\
&= S_g(\bar{h})(x,-\omega)
\end{aligned}$$

Conjugate symmetry follows from this property as a corollary: if h is a real-valued function, then

$$S_g(h)(x,\omega) = \overline{S_g(h)(x,-\omega)}.$$

Because the conjugate symmetry is only in the ω direction, it does not combine with the reflection property to make the transform of a real-valued even function real.

Differentiation

The differentiation property can be shown most easily using an integration by parts,

$$\begin{aligned}
\frac{\partial}{\partial x}\big[S_g(h)\big] &= \frac{\partial}{\partial x} \int_{\mathbb{R}^d} h(t)|\omega| g(\omega(t-x)) e^{-i2\pi\omega t} \,\mathrm{d}t \\
&= -\int_{\mathbb{R}^d} h(t)|\omega|\omega g'(\omega(t-x)) e^{-i2\pi\omega t} \,\mathrm{d}t \\
&= \int_{\mathbb{R}^d} \big[h'(t) - i2\pi\omega h(t)\big] |\omega| g(\omega(t-x)) e^{-i2\pi\omega t} \,\mathrm{d}t \\
&= S_g(h') - i2\pi\omega S_g(h)
\end{aligned}$$

Moving the last term to the other side provides the derivative property as stated.

Rotation

It has been claimed that the ST is rotationally invariant [25], but the proof was not shown. In general, the ST is not rotationally invariant, because of the dilation of the window with

respect to frequency.

Consider a multi-variable function $h : \mathbb{R}^n \to \mathbb{C}$. We examine its behaviour under a rotation $R : \mathbb{R}^n \to \mathbb{R}^n$ of its argument. Because R is a rotation, $R^{-1} = R^T$ and it is unitary, so $\langle Rx, Ry\rangle = \langle x, y\rangle$ and $\det(R) = 1$.

$$\begin{aligned} S_g(h(Rt))(x,\omega) &= \int_{\mathbb{R}^d} h(Rt)|\omega| g\big(\omega \circ (t-x)\big) e^{i2\pi\omega\cdot t}\,\mathrm{d}t \\ &= \int_{\mathbb{R}^d} h(s)|\omega| g\big(\omega \circ (R^T s - x)\big) e^{i2\pi\omega\cdot R^T s}\,\mathrm{d}s \\ &= \int_{\mathbb{R}^d} h(s)|\omega| g\big(\omega \circ (R^T s - x)\big) e^{i2\pi R\omega\cdot s}\,\mathrm{d}s \end{aligned}$$

Some rotation property might exist if the window's argument is equivalent to $(R\omega \circ (s-x)$ or $R\omega \circ (s - Rx)$, but this is not the case, even when the window is a Gaussian. The Hadamard product is not rotationally invariant, nor is it associative or distributive with matrix-vector multiplication, so

$$\|\omega \circ (R^T s - x)\|_2^2 = \|R\big(\omega \circ (R^T s - x)\big)\|_2^2 \neq \|(R\omega) \circ (s - Rx)\|_2^2,$$

in general. An issue also arises in the scaling factor $|\omega| = |\omega_1\omega_2\cdots\omega_d|$, because $|R\omega| \neq |\omega|$ in general.

2.3.2 Convolution

The convolution property is developed in [20]. It is not as immediately useful as convolution theorems for the Fourier transform or short time Fourier transform. The ST of a convolution behaves as,

$$S_g(u * v) = (S_g u) \circledast v, \tag{2.11}$$

where $\circledast$ is defined as:

$$\circledast : \mathcal{L}^2(\mathbb{R}^2) \times \mathcal{L}^2(\mathbb{R}) \to \mathcal{L}^2(R^2)(u \circledast v)(x,k) \mapsto \int u(x-s,k)v(s)e^{-iks}\,\mathrm{d}s \ ,$$

which is quite similar to a twisted convolution [12]. The proof follows:

$$\begin{aligned}
S_g(u * v) &= |\omega| \int \int u(t-s)v(s)\,\mathrm{d}s\, g(\omega(t-x))e^{-i\omega t}\,\mathrm{d}t \\
&= |\omega| \int v(s) \int u(t-s)g(\omega(t-x))e^{-i2\pi\omega t}\,\mathrm{d}t\,\mathrm{d}s \\
&= |\omega| \int v(s) \int u(z)g(\omega(z-(x-s)))e^{-i2\pi\omega(z+s)}\,\mathrm{d}z\,\mathrm{d}s \\
&= \int v(s) S_g u(x-s,\omega)e^{-i2\pi\omega s}\,\mathrm{d}s \\
&= (S_g u) \circledast v
\end{aligned}$$

2.3.3 Continuity at $\omega = 0$

From the definition of the ST (2.2), it is clear that

$$S_g h(x,0) = \int h(t)|0|g(0)e^0\,\mathrm{d}t = 0,$$

for any function h. However, the continuity of the ST at $\omega = 0$ is dependent on the function it is applied to. For example if $h(x) = 1$, then its transform is $\hat{g}(1)e^{i2\pi\omega\xi}$ for any $\omega \neq 0$.

For a large class of functions, the ST is guaranteed to be continuous as $\omega = 0$. This is shown in the following theorem.

Theorem. *If $f \in \mathcal{L}^p$ and $g \in \mathcal{L}^q$ with $\frac{1}{p} + \frac{1}{q} = 1$, and $1 < p, q < \infty$, then the ST $S_g f(\tau, \omega)$ is continuous as $\omega \to 0$.*

Proof. First, consider

$$
\begin{aligned}
\||\omega| g(\omega(t-x))\|_q &= \left[\int |\omega| g(\omega(t-x)|^q \,\mathrm{d}t\right]^{\frac{1}{q}} \\
&= \left[\int |\omega|^{q-1} |g(u)|^q \,\mathrm{d}u\right]^{\frac{1}{q}} \\
&= |\omega|^{\frac{q-1}{q}} \|g\|_q \ .
\end{aligned}
$$

Now using Hölder's inequality,

$$
\begin{aligned}
|S_g h(x,\omega)| &\leq \|h\|_p \|\omega g(\omega(t-x))\|_q \\
&= |\omega|^{\frac{q-1}{q}} \|h\|_p \|g\|_q \ .
\end{aligned}
$$

This approaches 0 as $\omega \to 0$, because $\frac{q-1}{q} > 0$. □

The above theorem also applies for the case $p = 1$ and $q = \infty$, that is $h \in \mathcal{L}^1$ and $g \in \mathcal{L}^\infty$, but not when $h \in \mathcal{L}^\infty$ and $g \in \mathcal{L}^1$.

In practice, a Gaussian function is used for the window. The Gaussian is in all L^p spaces, so as long as the signal h is also in an L^p space, it will be continuous at $\omega = 0$.

2.3.4 Limit as $\omega \to \infty$

Consider the ST as defined in (2.2). As $\omega \to \infty$, the window becomes arbitrarily narrow, while maintaining the same area. This should be familiar as one definition of the Dirac delta distribution. Thus

$$
\lim_{\omega\to\infty} S_g h(x,\omega) = \lim_{\omega\to\infty} h(x) e^{i2\pi\omega x}.
$$

While this limit does not converge, it is bounded, thus

$$
\limsup_{\omega\to\infty} |S_g h(x,\omega)| = h(x).
$$

2.4 The ST of Certain Functions

We were unable to find examples of the ST applied to functions in any papers. In this section, we apply the ST to a few simple functions, then apply the inverse formula. Apart from the Gaussian, none of these functions are in the spaces shown to work for the ST, but they are easy to work with. In the case of the constant function, when the ST is applied and inverted, there is a problematic factor of $\hat{g}(-1)$. This factor is not present for any of the other functions.

For full generality, the following transforms use a window normalized so $\|g\|_1 = 1$. To be safe, the window is scaled by $\bar{\omega}$ to separate it from the Fourier frequency.

2.4.1 Constant Function

First, consider the constant function $h(t) = 1$. Then for $\omega \neq 0$,

$$\begin{aligned} S_g h(x,\omega) &= \int_{\mathbb{R}} |\bar{\omega}| g\big(\bar{\omega}(t-x)\big) e^{-i2\pi\omega t}\,\mathrm{d}t \\ &= \frac{|\bar{\omega}|}{|\bar{\omega}|} e^{-i2\pi\omega x} \hat{g}\left(\frac{\omega}{\bar{\omega}}\right) \\ &= e^{-i2\pi\omega x} \hat{g}\left(\frac{\omega}{\bar{\omega}}\right). \end{aligned}$$

Now, substitute $\bar{\omega} = \omega$. When using a Gaussian window, $g(t) = \exp(-\pi t^2)$, there is a factor of $e^{-\pi^2}$. If the condition on g due to the resolution of the identity from [8] is considered, $\hat{g}(1) = 0$, so the ST is identically zero!

Now, applying the inverse transform as defined in (2.5),

$$\begin{aligned} &\mathcal{F}^{-1} \int_{\mathbb{R}} \hat{g}(1) e^{-i2\pi\omega x}\,\mathrm{d}x \\ =&\hat{g}(1)\mathcal{F}^{-1}\big(\delta(\omega)\big) \\ =&\hat{g}(1). \end{aligned}$$

The factor of $\hat{g}(1)$ is a problem which we have not been able to solve. It is unclear why

it appears here, and not elsewhere. It may be due to the fact that h is not in $\mathcal{L}^p$, or that its FT is a distribution.

2.4.2 Modulation Function

The problematic factor does not appear for $h(t) = e^{i2\pi\omega_0 t}$. The extra care of separating the scaling factor of $\bar{\omega}$ from the Fourier frequency ω is repeated,

$$\begin{aligned} S_g h(x,\omega) &= \int_{\mathbb{R}} e^{i2\pi\omega_0 t} |\bar{\omega}| g\big(\bar{\omega}(t-x)\big) e^{-i2\pi\omega t}\,\mathrm{d}t \\ &= \hat{g}\left(\frac{\omega-\omega_0}{\bar{\omega}}\right) e^{-i2\pi(\omega-\omega_0)x} \qquad \text{(by Fourier scaling and shift theorems)} \\ &= \hat{g}\left(1-\frac{\omega_0}{\omega}\right) e^{-i2\pi(\omega-\omega_0)x}. \end{aligned}$$

When compared against the ST of the constant function, the shift in the argument of $\hat{g}$ demonstrates the trouble in finding a modulation theorem for the ST. The behaviour of the window itself is affected by modulation. Applying the inverse,

$$\begin{aligned} &\mathcal{F}^{-1} \int_R \hat{g}\left(\frac{\omega-\omega_0}{\bar{\omega}}\right) e^{-i2\pi(\omega-\omega_0)x}\,\mathrm{d}x \\ =&\mathcal{F}^{-1}\hat{g}\left(\frac{\omega-\omega_0}{\bar{\omega}}\right)\delta(\omega-\omega_0) \\ =&\hat{g}(0)e^{i2\pi\omega_0 t}. \end{aligned}$$

This result is consistent with our inversion formulas.

2.4.3 Dirac Delta

The factor of $\hat{g}(1)$ also does not appear in the case of $\delta(t)$. Applying the ST,

$$\begin{aligned} S_g\delta(x,\omega) &= \int_{\mathbb{R}} \delta(t)|\omega|g(\omega(t-x))e^{-i2\pi\omega t}\,\mathrm{d}t \\ &= |\omega|g(-\omega x). \end{aligned}$$

The inverse, using (2.5) is,

$$\begin{aligned} &\mathcal{F}^{-1}\int_{\mathbb{R}} |\omega|g(-\omega x)\,\mathrm{d}x \\ =&\mathcal{F}^{-1}\|g\|_1 \\ =&\|g\|_1\delta(t) \end{aligned}$$

This is again consistent with the theory.

2.4.4 Gaussian

For the Gaussian function, a Gaussian window makes the calculation possible. Specifically, $h(t) = g(t) = e^{-\pi t^2}$ is used, because $\mathcal{F}g = g$. This time, an alternate formulation of the ST is used,

$$\begin{aligned} \mathcal{F}_1\big(Sh\big)(\alpha,\omega) &= \big(\mathcal{F}M_{-\omega}h(\alpha)\big)\big(|\omega|\mathcal{F}D_\omega g(\alpha)\big) \\ &= e^{-\pi(\alpha+\omega)^2}e^{-\pi\left(\frac{\alpha}{\omega}\right)^2} \\ Sh(x,\omega) &= \int_{\mathbb{R}} e^{-\pi(\alpha+\omega)^2}e^{-\pi\left(\frac{\alpha}{\omega}\right)^2}e^{i2\pi\alpha t}. \end{aligned}$$

Integrals involving quadratics in exponents can be performed with (3.323.2) in [11]. The result, after further simplification is,

$$Sh(x,\omega) = \frac{|\omega|}{\sqrt{1+\omega^2}}\exp\left(-\pi\frac{\omega^2}{1+\omega^2}(1-x^2-i2\pi\omega x)\right).$$

The modulus of this function is plotted in Figure 2.1. The function can be compared

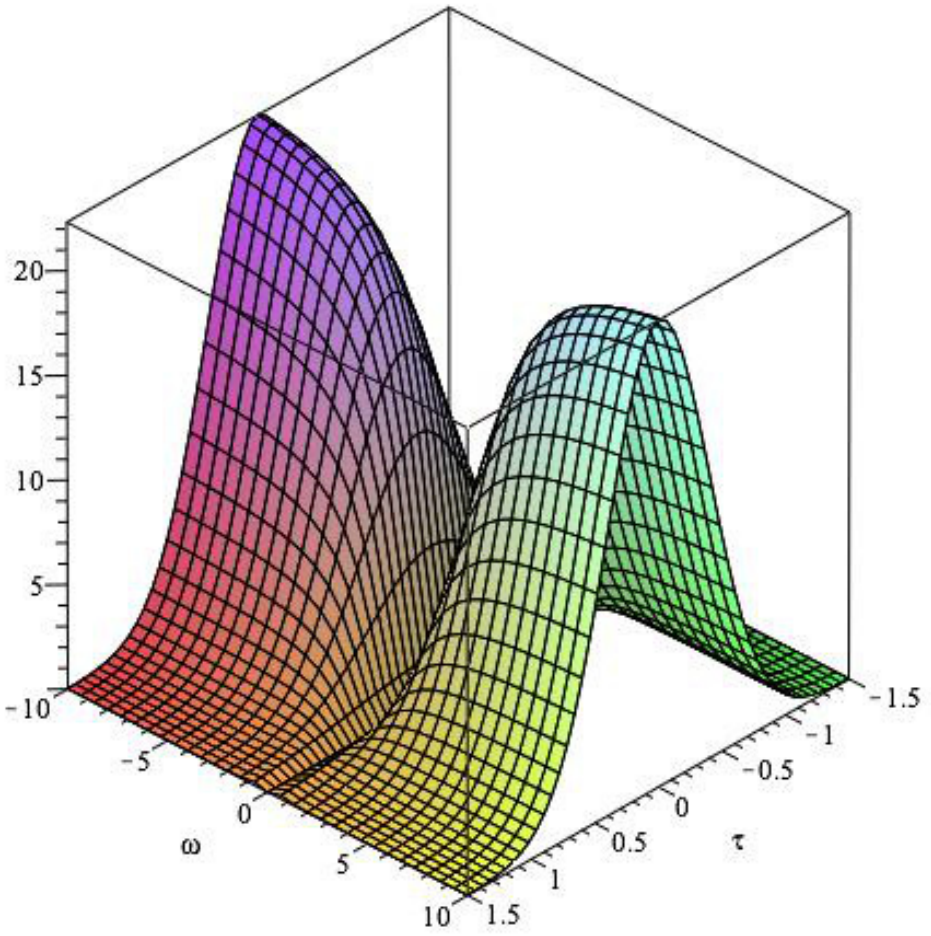


Figure 2.1: The modulus of the ST of the Gaussian $e^{-\pi t}$

against some of the ST properties from Section 2.3. Note that it is indeed continuous as $\omega \to 0$, and bounded by $e^{\pi}h(t)$ as $\omega \to 0$.

Applying the inverse (2.5), let $s = \frac{\omega^2}{1+\omega^2}$ and substitute $\tau = sx$ in the integral,

$$\begin{aligned}
&\mathcal{F}^{-1}\left[\sqrt{s}e^{-\pi s}\int_{\mathbb{R}} e^{-\frac{\pi}{s}\tau^2}e^{-i2\pi\omega\tau}\frac{\mathrm{d}\tau}{s}\right] \\
=&\mathcal{F}^{-1}\left[\frac{1}{\sqrt{s}}e^{-\frac{\pi}{s}\tau^2}\sqrt{s}e^{-\pi s\omega^2}\right] \\
=&\mathcal{F}^{-1}\left[e^{-\pi\frac{\omega^2}{1+\omega^2}(1+\omega^2)}\right] \\
=&e^{-\pi t^2}.
\end{aligned}$$

Chapter 3

Discrete Stockwell Transform

To actually compute transforms and analyze data, we need a discrete version of the Stockwell transform (ST). The discrete Stockwell transform (DST) is formulated the same way as the ST, but on a finite-dimensional vector space, with a finite number of atoms. If it is calculated as a matrix-vector product, the DST takes on the order of N^3 operations. However, the ST can also be defined using the FT (2.4). This form suggests a 'fast' DST

$$\begin{aligned} DST(h)[n,k] &= \sum_{t=0}^{N-1} \hat{g}_k[t]\hat{h}[t+k]e^{i\frac{2\pi}{N}nt} \\ DST(h)[n,0] &= \frac{1}{N}\sum_{t=0}^{N-1} h[t] \end{aligned} \tag{3.1}$$

where $\hat{g}_k$ is the DFT of the window, scaled in width according to frequency k, and $\hat{h}$ is the DFT of the signal. By refactoring the transform using the DFT, the fast Fourier transform can be used. With the FFT, this calculation is $\mathcal{O}(N^2 \log n)$.

Defining the $k = 0$ components as the mean of the signal can be justified by considering the limit $k \to 0$. For $k = 0$, the window becomes infinitely wide, so it can be treated as a constant and the DST becomes a mean. For filtering purposes, this choice works best with the inverse seen in Section 3.3.

3.1 Two-Dimensional DST

The two-dimensional DST is defined from the continuous ST in much the same way as the DST. For the most part, it is as simple as increasing the dimension of the window and DFT. However, the zero-frequency components must be defined separately in order to get an accurate inverse. This was not specified in existing work.

Starting with the Gaussian definition of the window,

$$g_{p,k}[n] = \frac{|n|}{\sqrt{2\pi}N} \exp\left(\frac{(p-n)^2k^2}{2N^2}\right)$$
$$g_{p,0}[n] = \frac{1}{N},$$

the two-dimensional window is a product of two windows, $g_{p,q,k_1,k_2}[n,m] = g_{p,k_1}[n] \circ g_{q,k_2}[m]$. The DST on an $N \times M$ signal is defined as,

$$H[p,q,k_1,k_2] = \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} \sum_{m=-\frac{M}{2}}^{\frac{M}{2}-1} h[n,m] g_{p,q,k_1,k_2}[n,m] e^{-i2\pi\left(\frac{nk_1}{N}+\frac{mk_2}{M}\right)}. \tag{3.2}$$

Now, substituting $k_2 = 0$,

$$H[p,q,k_1,0] = \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} \left(\frac{1}{M} \sum_{m=-\frac{M}{2}}^{\frac{M}{2}-1} h[n,m]\right) g_{p,k_1}[n] e^{-i\frac{2\pi}{N}nk_1}$$
$$H[p,q,k_1,0] = DST\left(\frac{1}{M} \sum_{m=-\frac{M}{2}}^{\frac{M}{2}-1} h[n,m]\right)[p,k_1].$$

The same steps can be performed with $k_2 = 0$, to complete the 2D-DST.

3.2 Implementation Details

When using this inverse and the DST for analysis and filtering applications, there are some important aspects to keep in mind.

First, the frequencies follow the same rules as for the DFT. They actually run from $-N/2$ to $N/2-1$, and $H(n,k) = H(n,N-k)$. This implies that the window function g_k in (3.1) is actually g_{N-k} for values of k above $N/2$.

The window function $\hat{g}_k$ is traditionally a Gaussian in shape. For the DST to be invertible, it is necessary that $\sum_t g_k[t] = N$, but a Gaussian extends past the interval, so it's sum is too low when truncated.

The solution is to perform an N-periodic sampling of a Gaussian [1],

$$g_{p,k}[n] = \frac{|n|}{\sqrt{2\pi}N} \sum_{r=-\infty}^{\infty} e^{-\frac{(p-k+rN)^2 k^2}{2N^2}}.$$

This Poisson summation form of the window can be approximated through a "folded window" [19]. The window is typically folded over once, which does grant good accuracy for the inverse (3.3). My tests found a root mean square error of about 10^{-15} when the DST was applied to a signal followed by this inverse. The signals tested were a variety of linear and hyperbolic chirps with a maximum amplitude of 1, and Gaussian white noise.

3.3 Inverse DST

The inverse DST can be determined using the same formula as for the continuous case (2.5). Integration is replaced with summation, and the FT becomes the DFT.

$$IDST(H) = IFT(\sum_{n=0}^{N-1} H(n,k)) \tag{3.3}$$

Other inverses can be implemented which correspond to the continuous inverse formulas in Section 2.2. The discrete analogue of (2.6) did not work well in my own tests. While (3.3) had a root mean square error on the order of 10^{-15}, the time-time based inverse had an error on the order of 10^{-2}.

The resolution of the identity form (2.10) is quite flexible and corresponds to the frame interpretation of the DST. Given a window, g, and a dual window γ,

$$h = \sum_{n,k} \langle h, g_{n,k} \rangle \frac{\gamma_{n,k}}{|k|} = \sum_{n,k} S[n,k] \frac{\gamma_{n,k}}{k}, \tag{3.4}$$

where $g_{n,k}[t] = |k|g[k(t-n)]e^{i\frac{2\pi}{N}kt}$, and likewise for γ.

3.4 Example: DST of a signal

To show how the DST analyzes the time-frequency plane, it was applied to the signal with length $N = 512$,

$$h[t] = \begin{cases} \sin\left[\frac{\pi}{8}t\right] & 0 \leq t < 256 \\ \sin\left[\frac{\pi}{2}t\right] & 256 \leq t < 512\,. \end{cases} \tag{3.5}$$

The DST of the signal is shown in Figure 3.1. Time is in the horizontal direction, and frequency is in the vertical with the zero-frequency in the centre. The two frequencies are identified at the ridges of the transform. Notice that the frequency resolution is better for lower frequencies, but the time resolution is better for high frequencies. There is also a significant spike across all frequencies when the frequency of the signal changes, even though the signal is continuous.

3.5 DST Frame

The DST was analyzed as a frame in [1]. The authors proved that it is indeed a frame, and has the inverse (3.4). The highly redundant frame of N^2 atoms was shown to be reducible to about $2N$ atoms using a dyadic sampling of frequencies as in [14], rather than the linear sampling used up to this point. A reduction in redundancy was also tested using a Matching Pursuit technique.

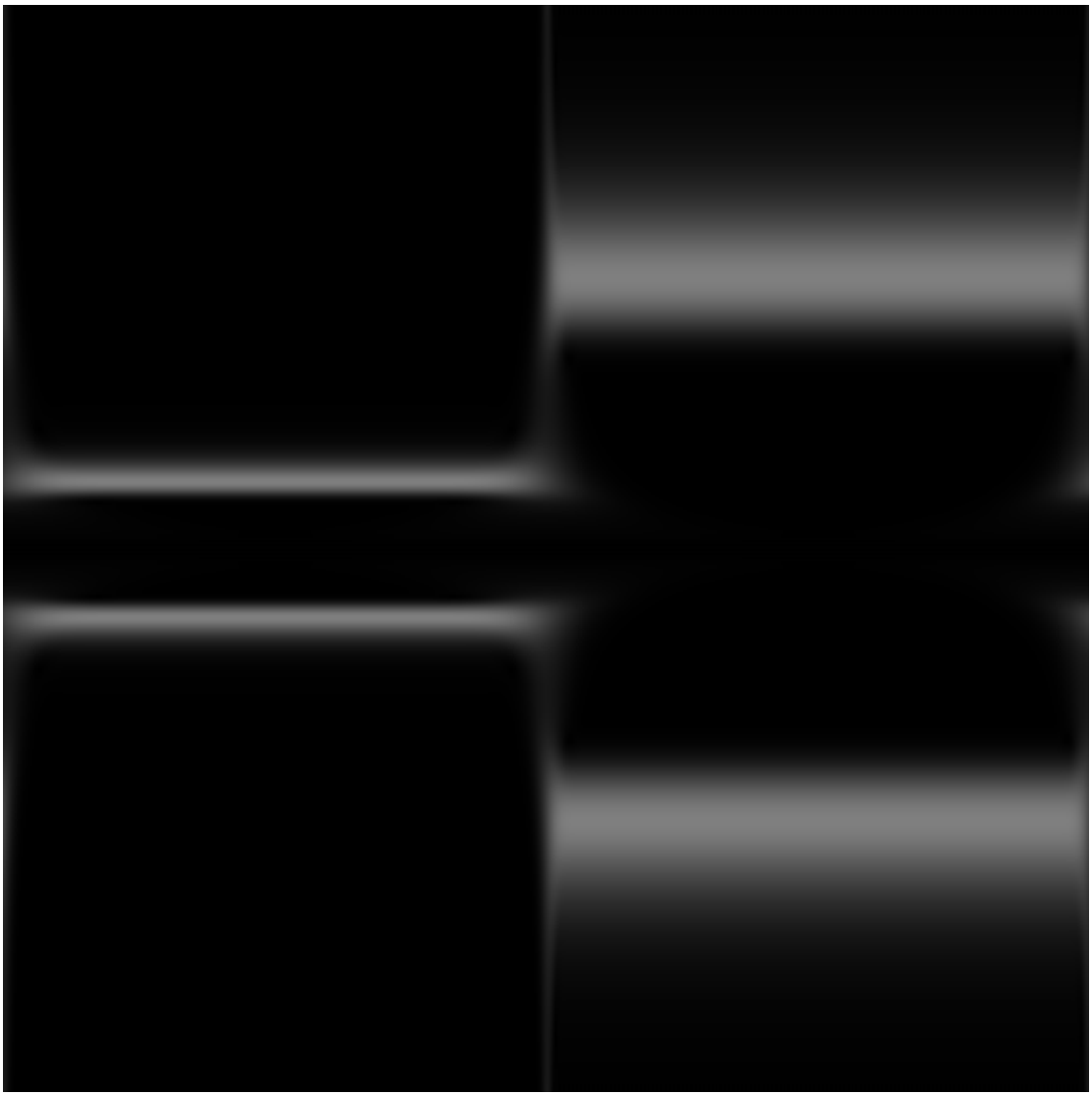


Figure 3.1: The modulus of the DST of a signal composed of two non-overlapping sine waves. Time is in the horizontal direction, and frequency is in the vertical with the zero-frequency in the centre.The two frequencies are identified at the ridges of the transform.

Chapter 4

Discrete Orthonormal Stockwell Transform

The discrete orthonormal Stockwell transform (DOST) was created to provide many of the same features as the DST, but with a non-redundant, orthonormal basis. With an orthonormal basis, the DOST transforms a signal of length N into N coefficients.

In [23], Stockwell defines the DOST with a set of basis functions,

$$D_{\nu\beta\tau}[k] = \sum_{k=0}^{N-1} h[k] \frac{e^{i\pi\tau}}{\sqrt{\beta}} \sum_{f=\nu-\beta/2}^{\nu+\beta/2-1} \exp(-i\frac{2\pi}{N}kf) \exp(+i\frac{2\pi}{\beta}\tau f).$$

The parameters ν, β, and τ represent the centre frequency, bandwidth, and centre time of the basis function, respectively. If the parameters are chosen so that the different values of ν and β do not overlap in frequency, and $\tau = 0, \ldots, \beta - 1$, then the basis is orthonormal.

This definition of the DOST does not provide much insight into what it represents, and it does not immediately present a fast version of the algorithm. Instead, we will derive the DOST as a special case of the DST in a similar way to [3].

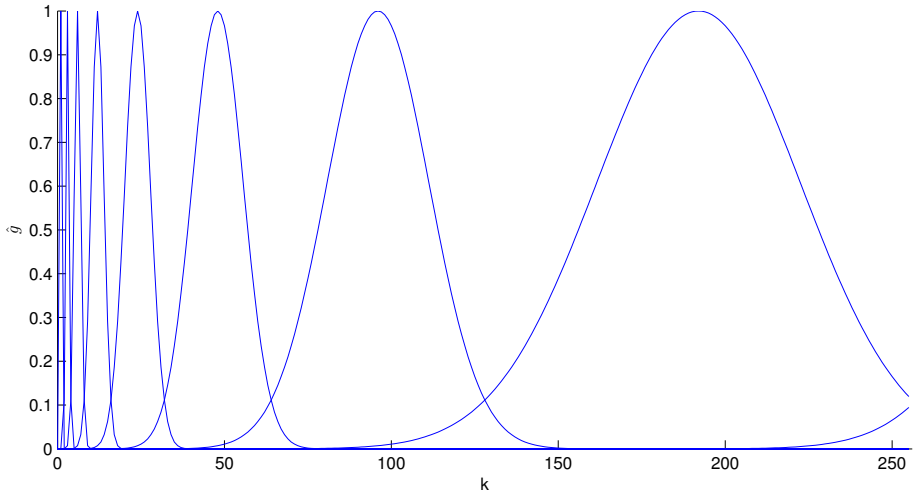


Figure 4.1: Fourier transform of DST window functions. The windows drawn correspond to the centre frequencies ν used in the DOST

4.1 Definition

To define the DOST, recall that the DST can be calculated as,

$$DST[n,k] = IFT(\hat{g}_k[t]\hat{f}[t+k]), \tag{4.1}$$

so the DST applies a window to the FT of the signal for each frequency. These windows have overlapping support, and vary in width depending on the centre frequency of the window. Typically, the window is a Gaussian, which has non-compact support. Because the windows overlap, as seen in Figure 4.1, it is impossible to make an orthonormal basis from them.

For an orthonormal DST basis, a rectangular window in frequency-space can be used instead. By only using non-overlapping windows, the windows are orthogonal and effectively partition the frequency space. The centres and widths of these windows are the ν and β in Stockwell's definition [23], respectively.

Using the inverse FT on this whole space to complete the DST would still be redundant, because when the signal is bandlimited, at most β frequencies can be orthogonal on the smaller band. These frequencies correspond to the β-dimensional FT on that band. When

a local inverse FT is applied on the band, it will be out of phase of the global inverse FT, so there will need to be a correction. The process is illustrated in Figure 4.2.

This interpretation can be confirmed by working with the sums in Stockwell's definition of the Stockwell transform ,

$$\begin{aligned} < h, D_{\nu\beta\tau} > &= \sum_{k=0}^{N-1} h[k] \frac{e^{i\pi\tau}}{\sqrt{\beta}} \sum_{f=\nu-\beta/2}^{\nu+\beta/2-1} \exp(-i\frac{2\pi}{N}kf) \exp(+i\frac{2\pi}{\beta}\tau f) \\ &= \frac{e^{i\pi\tau}}{\sqrt{\beta}} \sum_{f=\nu-\beta/2}^{\nu+\beta/2-1} \exp(+i\frac{2\pi}{\beta}\tau f) \sum_{k=0}^{N-1} h[k] \exp(-i\frac{2\pi}{N}kf) \\ &= \frac{e^{i\pi\tau}}{\sqrt{\beta}} \sum_{f=\nu-\beta/2}^{\nu+\beta/2-1} \mathcal{F}(h)[f] \exp(+i\frac{2\pi}{\beta}\tau f) \ . \end{aligned}$$

The final sum is the definition of the inverse DFT on the sub-band centred at ν with width β. Performing the calculation, the phase-correcting factor comes out as $e^{-i\pi\tau}$, and since $\tau \in \{0, ..., \beta - 1\}$, it is just the alternating sequence $(-1)^\tau$. In Figure 4.3, a basis function is shown at two adjacent time steps, and the effect of phase correction should be clear.

4.2 Details of Implementation

To ensure an orthonormal basis, the window has to be scaled by the square root of the width. When using an existing implementation of the DFT this scaling has to take into account the normalization used by the implementation. In MATLAB, the forward DFT is not scaled, but the inverse is divided by N. For the basis to be normalized, each partition must be divided by $\sqrt{\beta}$.

The order of the coefficients in the DOST can be chosen to get conjugate symmetry. For the negative frequency bands, the order of the coefficients should be reversed after the inverse FT.

The algorithm for the DOST is therefore,

```
H = FFT(h)
for each band
```

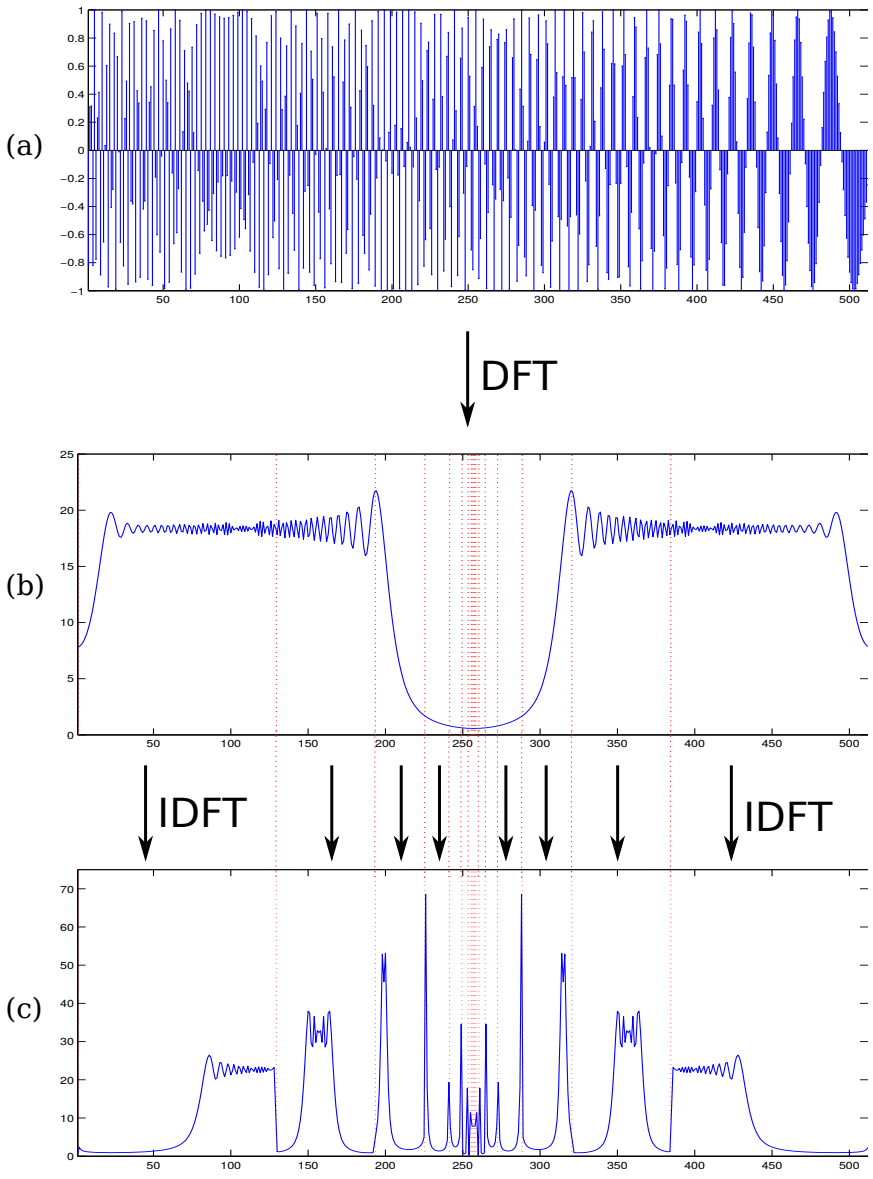


Figure 4.2: An illustration of the DOST Algorithm. Starting with the original signal (a), the DFT is applied to get (b) , then on each sub-band, an IFT (with phase correction) is applied. The final result is the DOST (c). Absolute values of the DFT and DOST are shown for clarity.

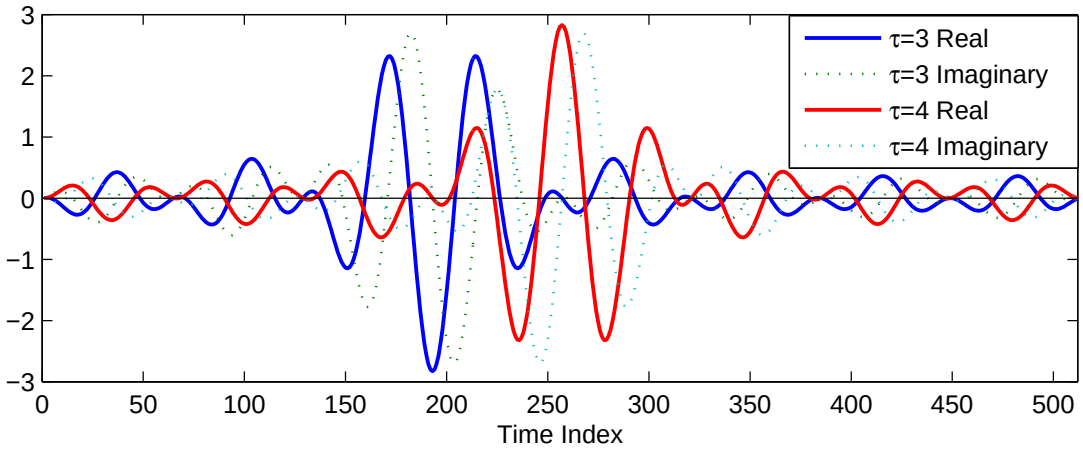


Figure 4.3: DOST basis function with parameters $\nu = 16, \beta = 8$. The phase correction of ± 1 can be seen by comparing the functions with $\tau = 3$ and $\tau = 4$

```
    S(band) = sqrt(band_width) * IFT(band) .* [1,-1,...]
    if the frequency is negative,
        S(band) = reverse( S(band) )
end for
```

The DOST in higher dimensions can be performed the same way, except with N-dimensional bands and DFT's. Another way is to perform the one-dimensional transform in each direction. Because the basis functions are made entirely of exponents, it can be separated into factors of the DOST in each dimension.

4.3 Variations on the DOST

There are a few variations of the DOST that have been introduced. For a real-valued signal, the original is conjugate symmetric about $k = 0$, which prevents $k = 0$ and $k = -N/2 - 1$ from being in conjugate symmetric pairs. Y. Wang [26] proposed making it conjugate symmetric about $k = -1/2$, so that each coefficient has a conjugate. The symmetry is done by first modulating the signal (shifting it in frequency by $1/2$), and then partitioning the frequency space symmetrically about zero.

As an example, the dyadic partitions of a signal of length 32 are,

$$\text{DOST} : [1, 8, 4, 2, 1, 1, 1, 2, 4, 8]$$
$$\text{symDOST} : [8, 4, 2, 1, 1, 1, 1, 2, 4, 8]$$

starting from the frequency $k = -N/2$.

The "generalized Fourier family transform" described by Brown [3] is an attempt to concentrate the support of the basis functions. They use the same Gaussian windows as in the DST, but cut the tails of so that the windows do not overlap. Truncating the Gaussian window allows the basis functions to be orthonormal, but inverting the transform requires dividing by this Gaussian window again. We call it the *windowed DOST*, because the only difference applying a window to each sub-band.

In Section 7.3, the windowed DOST is shown to have very poor performance compared to other methods for compression. We think this is because of the extra windowing operations, putting more significance on smaller values.

The basis functions for these variations are presented in Figure 4.4. The symmetric DOST is nearly identical to the original, but is at a slightly different frequency, which is visible when the two are shown on the same axes. The windowed DOST has a very different profile, because the window is smoother in the time-domain, causing fewer ripples.

4.4 Example: DOST of a Signal

To show how the DOST analyzes the time-frequency plane, it was applied to the signal defined in (3.5). Using the values of ν, β and τ for each component, the Heisenberg boxes can be determined. Colouring each box by the value of the coefficient gives the time-frequency plot in Figure 4.5. Because the DST is normalized differently, each coefficient in the DOST was also multiplied by $\sqrt{\beta}$ so that the two plots would look similar.

The two frequencies are still identified, but not as precisely, because there are no ridges. The frequency resolution is still better for lower frequencies, and the time resolution is better for high frequencies. There is also a significant spike across all frequencies when the frequency of the signal changes.

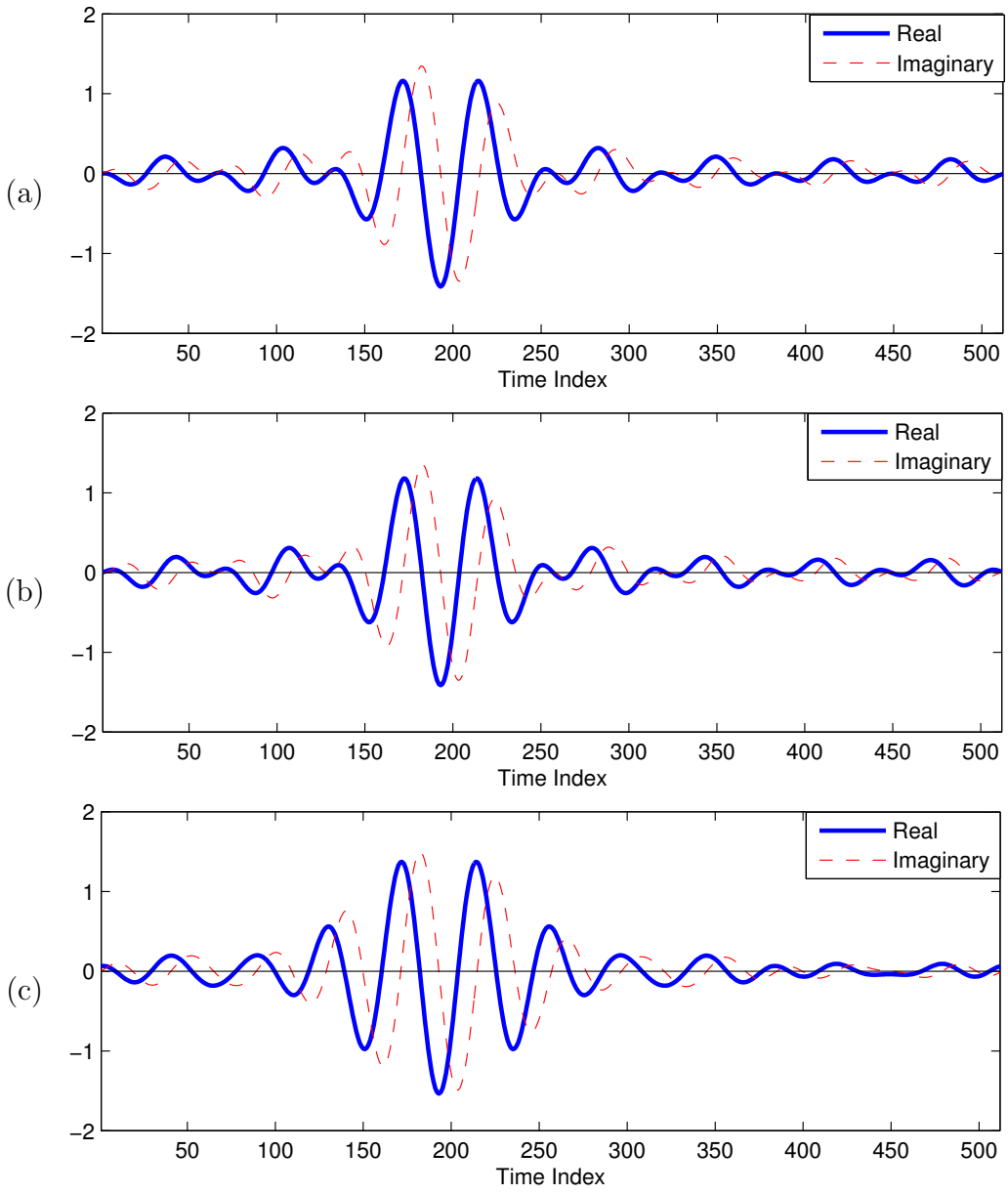


Figure 4.4: Basis functions for the DOST variations using the parameters $\nu = 16, \beta = 8, \tau = 3$. The corresponding transforms are (a) DOST, (b) Conjugate-Symmetric DOST, and (c) Windowed DOST.

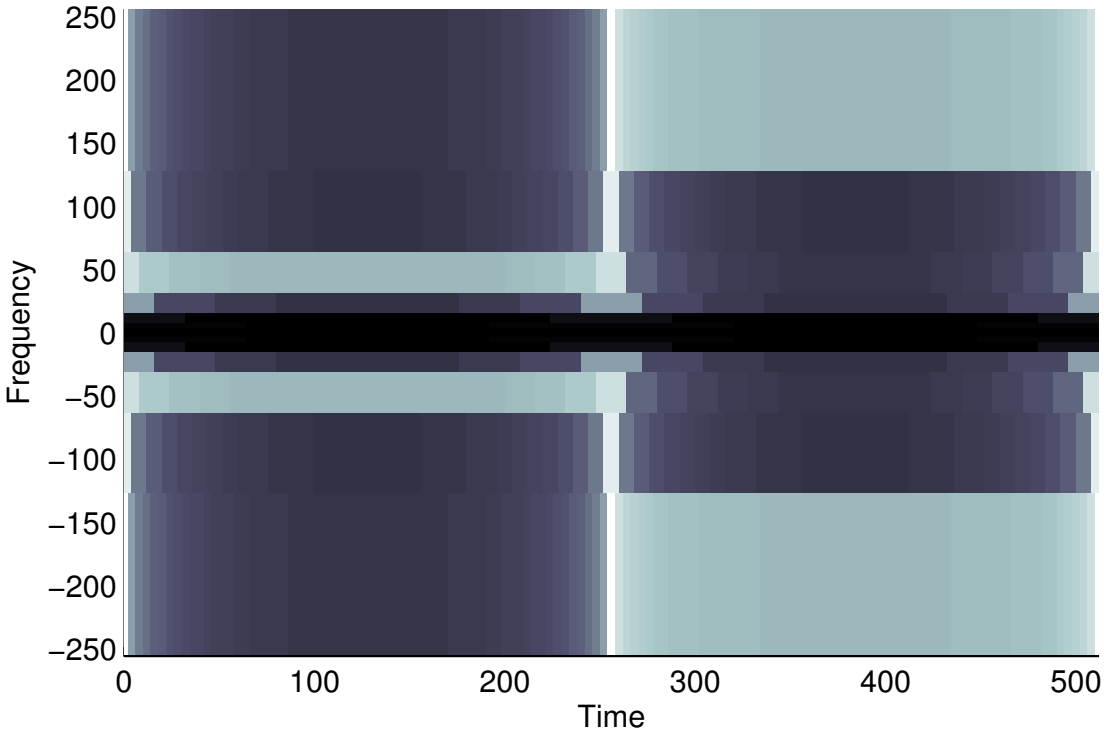


Figure 4.5: The time-frequency representation of the signal (3.5) as calculated with the DOST. The Heisenberg box of each basis function coloured by the value of its coefficient.

4.5 DOST of an Image

All variations of the DOST look very similar when applied to an image. There are slight differences in the partitioning and actual coefficient values, but the large-scale features are the same. The DOST of an image is shown in figure 4.6. The zero-frequency components are in the center of the figure, with frequency increasing to the right and down.

A miniature copy of the image is seen in each sub-band. This is due to the stronger correlation of the basis functions to edges, particularly for higher frequencies. A similar effect is seen in quad-tree representations of wavelet transforms. The image in bands with negative frequencies is reflected, because the order of coefficients is reversed to maintain conjugate-symmetry.

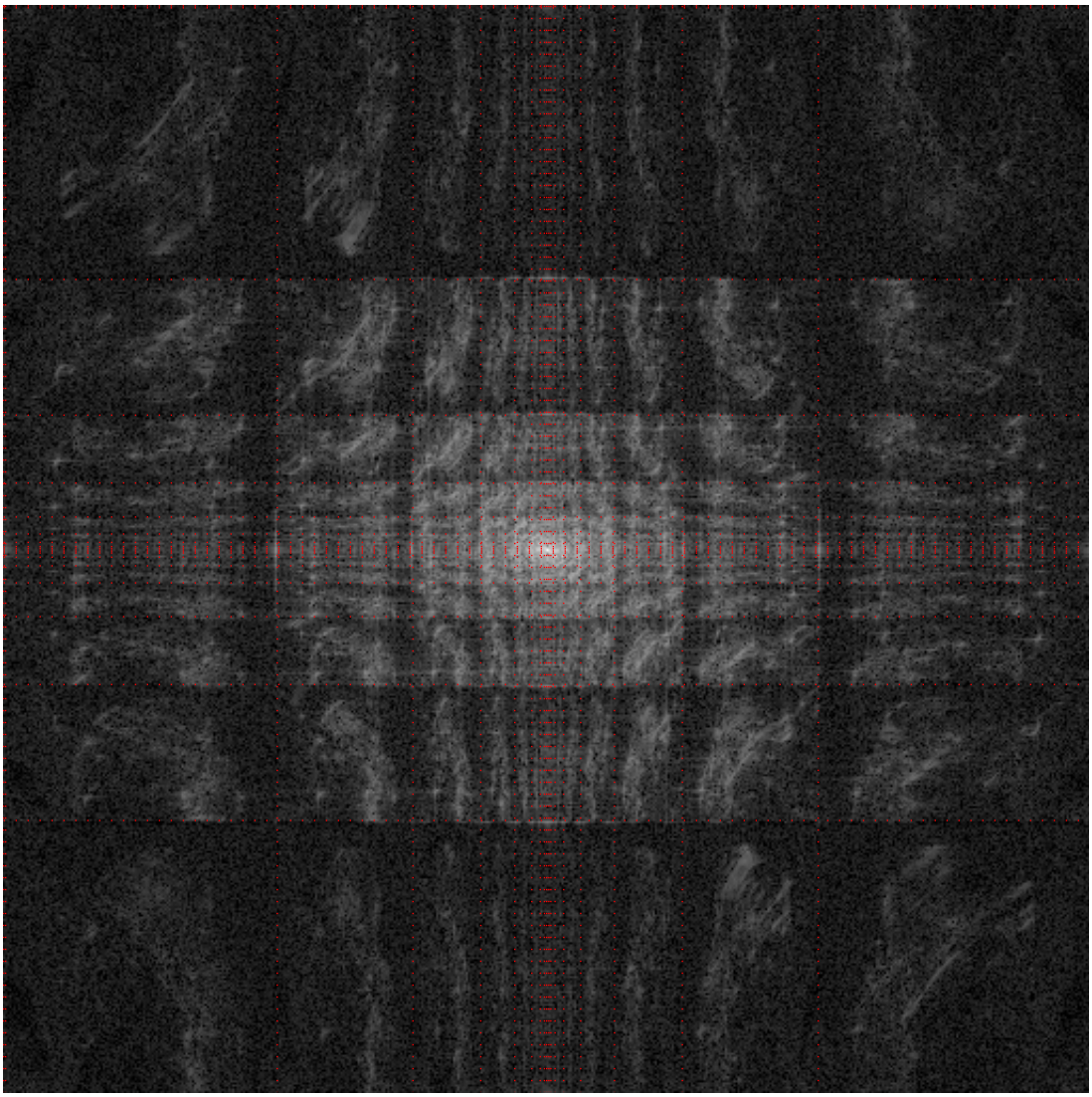


Figure 4.6: The DCST transform of the test image Lena. Intensities are absolute values on a logarithmic scale.

Chapter 5

Discrete Cosine Based DOST

The factoring of the DOST into a FT followed by a local inverse FT on each partition leads to the question: can this same process be used for any unitary transform? Yes, it can be done, because the construction is based entirely on the unitary nature of the transforms.

In the general version, a unitary transform is applied to a signal. The transformed signal is partitioned, and the inverse transform is applied on each sub-band. Because the Fourier basis functions are complex exponentials and the DFT has modulation-translation and scaling relations, this has the effect of scaling and translating a set of basis functions, much like a wavelet transform (see Section 6.1).

Because of the success that the DCT has seen in image compression (notably JPEG), as well as its real-valued nature and close relation to the FT, we applied it to this algorithm.

5.1 Details of Implementation

First, a DCT is applied to the signal. There are a few different DCT definitions depending on what symmetries are assumed for the signal [14]. We tested the DCT-I, which is most closely related to the DFT (even symmetric about $k = 0$), as well as the DCT-II used in JPEG compression, which is even-symmetric about $k = -1/2$,

$$X[k] = \sum_{n=0}^{N-1} x[n] \cos\left(\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right). \tag{5.1}$$

Unlike the DOST, the DCST is not a separable transform. For 2-dimensional images, it is necessary to use the 2D versions of the underlying transforms. The final version we settled on uses the JPEG DCT. The MATLAB signal processing toolbox implementation was used.

Because there are no negative frequencies for the DCT, a dyadic partitioning is used with $\beta_0 = 1$, $\beta + n = 2^{n-1}$. The algorithm is then a straightforward adaptation of the DOST,

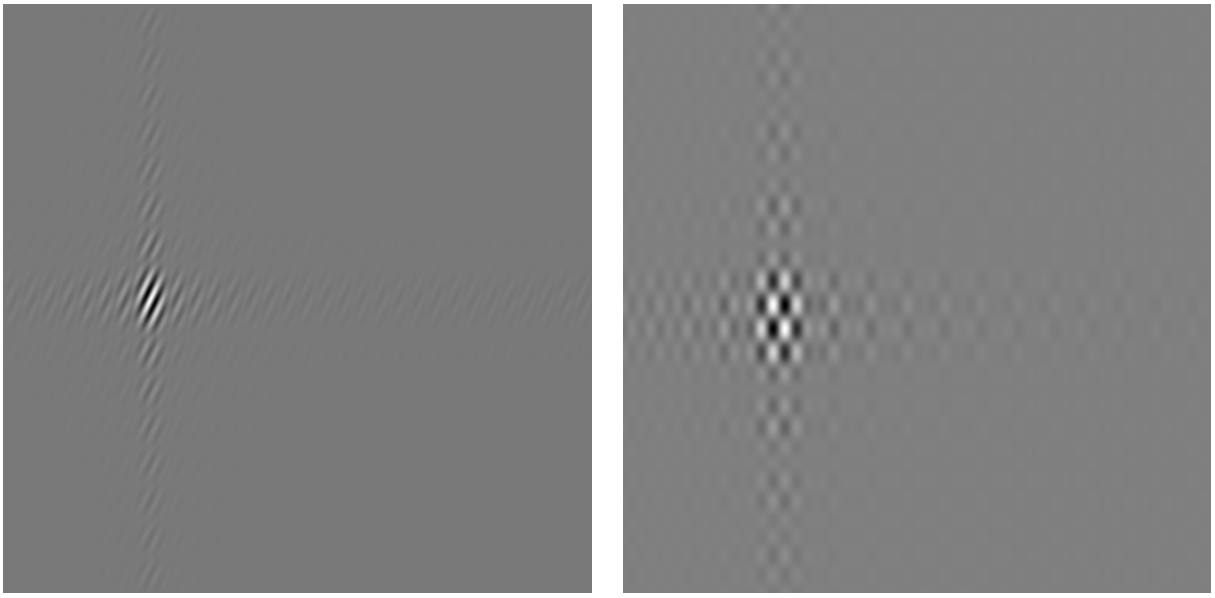
```
H = dct2(h)
x = y = 0
for bx in [1,1,2,4,8,...]
    for by in [1,1,2,4,8,...]
        H(x:x+bx-1, y:y+by-1) = idct2(H(x:x+bx-1, y:y+by-1))
    end for
end for
return H
```

5.2 Basis Elements

A typical basis element for the DCST is shown in figure 5.1 alongside a DOST basis element. The DCST has the same grating style as the DCT, but is somewhat localized in space. Because of the sharp cut-off in the window, the basis function oscillates out from its centre. This causes ringing around edges, and reduces the time-locality, but to a lesser extent than for the DOST.

5.3 Transform of an Image

The DCST of an image looks quite similar to that of the DOST, except that it is real-valued, and does not have the negative frequencies, so there is no symmetry in the coefficients. The lowest frequencies are in the upper left corner, and increase to the right and down. The same repetition of the edges of the image in each block is seen as for the DOST in figure 5.2.



(a) (b)

Figure 5.1: Basis elements of the (a) DOST and (b) DCST. Both are for band-widths 32x16 and positions 8x8.

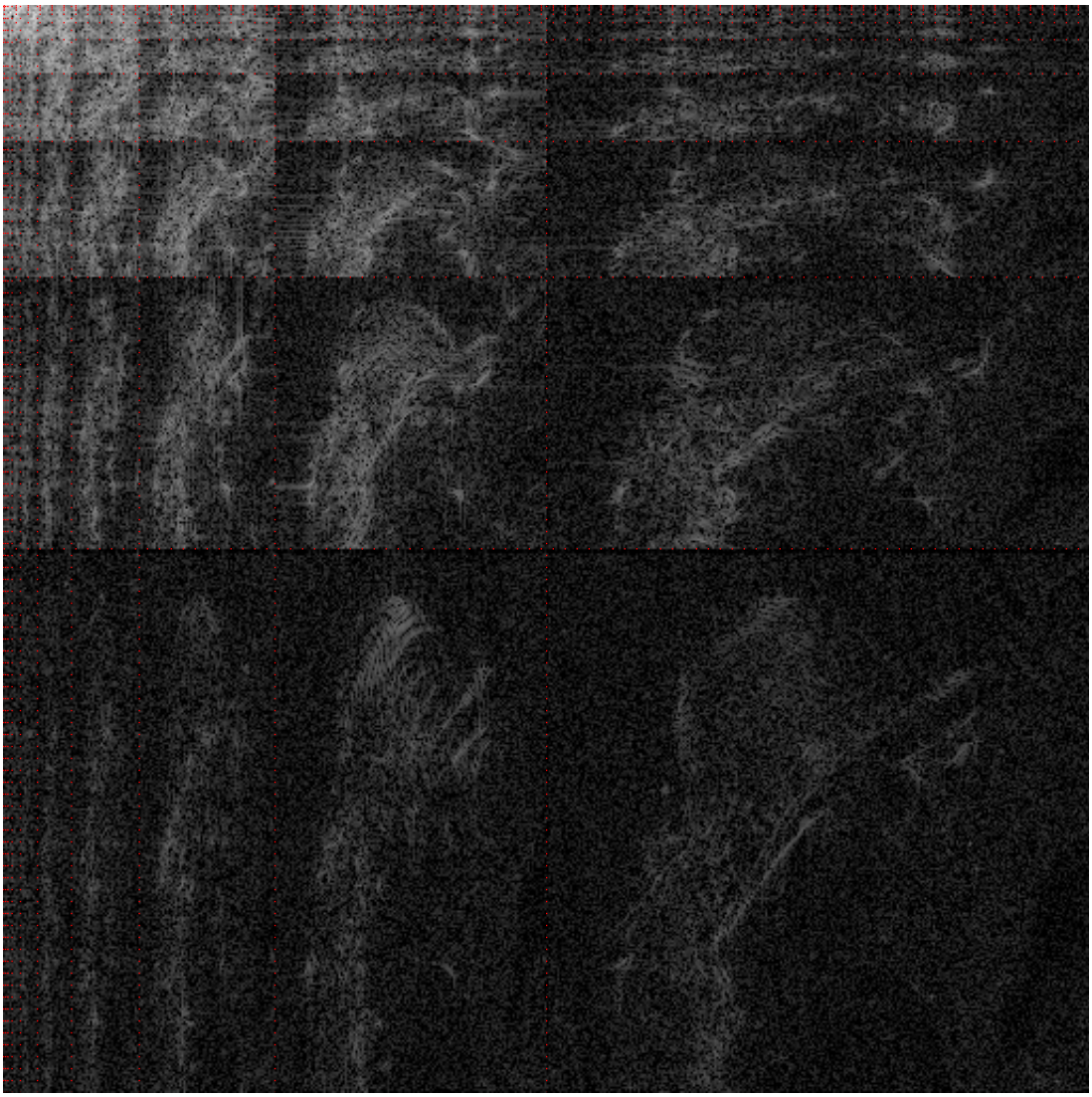


Figure 5.2: The DCST transform of the test image Lena. Intensities are absolute values on a logarithmic scale.

Chapter 6

Harmonic Wavelet Transform

The HWT was a transform introduced by Newland for vibration analysis [16]. The DOST is nearly identical to the HWT, only differing in the complex phase, but they have not been compared before. In this chapter, the similarities of the two transforms are shown by constructing them both side by side.

Due to their similarities, both the DOST and HWT would have the same applications. Harmonic wavelets have been used primarily used for ridge and phase identification of signals [17, 18].

Ridge identification is a method of finding the highest energy in time-frequency (or time-scale). The phase does not matter, so results would be largely the same. Various other wavelets (like Morlet wavelets) are also used for this application.

Phase extraction uses the complex phase of the transformed signal to determine the frequency and additional information of the signal. The frequency is approximately equal to the time-derivative of the phase. Using HWT there are other correcting terms that need to be included to deal with the phase change introduced by shifting the wavelet in time. The phase correction in the Stockwell transform eliminates this step.

A sharp change in phase also indicates a reflection. In [18], phase extraction was used on the data from bending waves generated by an impulse in a beam. The ridges indicated when the energy at a particular frequency was located at the measurement device, while the abrupt phase changes indicated when the wave was reflected at one end.

Apart from compression and denoising, there have not been many applications of HWT for two-dimensional signals. The tests that have been done for images are also less extensive than the tests we have performed.

A generalized HWT was also developed [18]. This version approaches the ST, because it allows different window conditions and widths, but does not require orthogonality. Because filtering and signal reconstruction was not an interest in [18], the invertibility of the generalized HWT was ignored.

6.1 Derivation of Basis Functions

There are a few differences in how the HWT and DOST were derived, but the result is almost identical. The DOST was developed strictly as a transform on a discrete space [23], while Newland [16] developed harmonic wavelets on a continuous domain, then adapted it to a discrete domain. To make the construction simpler and cleaner, we will follow the same path as Newland.

In the two developments, a different Fourier transform is used, along with slightly different notation. Stockwell uses the basis functions $e^{i2\pi fx}$, while Newland uses $e^{i\omega x}$. This affects how the frequency bands are defined as well. Here, the same Fourier basis as Stockwell will be used in order to be consistent with the rest of the thesis.

Both transforms define their basis functions from the same concept. We desire a partitioning of the frequency domain, so that the basis functions are as tight as possible in frequency. Box-car functions (or indicator functions on an interval) are used.

Stockwell defines the interval using two parameters: ν, the centre of the interval; and β the width of the interval. Because Newland is immediately interested in a wavelet transform, he simply chooses a primary interval of $\omega_N \in [2\pi, 4\pi)$. This is equivalent to $\nu = 3/2$, $\beta = 1$, so that the FT of the mother wavelet is

$$\hat{\psi}(\omega) = \frac{1}{\sqrt{\beta}} \text{boxcar}\left(\frac{\omega - \nu}{\beta}\right),$$

where

$$\text{boxcar}(\omega) = \begin{cases} 1 & \omega \in \left[-\frac{1}{2}, \frac{1}{2}\right) \\ 0 & \text{otherwise} \end{cases}.$$

It then follows that

$$\begin{aligned}\psi(x) &= e^{i2\pi\nu x}\sqrt{\beta}\,\mathrm{sinc}(\beta x) \\ &= \sqrt{\beta}\frac{e^{i\pi(2\nu+\beta)x} - e^{i\pi(2\nu-\beta)x}}{i2\pi\beta x}.\end{aligned}$$

The last line follows from an expansion of $\mathrm{sinc}(\cdot)$ into exponential terms, and is the form used by Newland in his construction. We will continue with the definition using the sinc function, because the modulation is explicit. The factor of $\sqrt{\beta}$ normalizes the basis functions in $\mathcal{L}^2$.

At this point, Newland uses this function (with the particular ν and β) as a mother wavelet and forms a basis through shifts and scaling. Stockwell keeps it a bit less structured, and just shifts this form. If it was the original Stockwell transform, then we should shift by τ, but it will prove more convenient to shift by τ/β, i.e.,

$$\psi(x) = e^{-i2\pi\nu\frac{\tau}{\beta}}e^{i2\pi\nu x}\sqrt{\beta}\,\mathrm{sinc}(\beta x - \tau) \tag{6.1}$$

The standard dyadic partitioning used by Stockwell [23] is now applied. We set $\beta = 2^j$, $\nu = 3\cdot 2^{j-1}$ and $\tau = k$, so

$$\psi_{j,k}(x) = e^{-i3\pi k}2^{j/2}\,\mathrm{sinc}(2^j x - k)e^{i\pi 2^j x} \quad .$$

This exact result can be obtained by scaling and shifting the mother wavelet,

$$\begin{aligned}\psi_{j,k}(x) &= 2^{j/2}\psi(2^j x - k) \\ &= 2^{j/2}\exp(i3\pi(2^j x - k))\,\mathrm{sinc}(2^j x - k) \\ &= e^{-i3\pi k}2^{j/2}\,\mathrm{sinc}(2^j x - k)\exp(i3\pi(2^j x)) \quad .\end{aligned}$$

Up to this point, the two transforms are identical. The Stockwell transform takes an additional step: a multiplication by $e^{i\pi\tau}$ "corrects" the phase of the basis function, which has been skewed by the translation in space. The final result is

$$\psi_{\nu,\beta,\tau}(x) = \sqrt{\beta}\,\mathrm{sinc}(\beta x - \tau)e^{i2\pi\nu x} \ , \text{ or}$$
$$\psi_{j,k}(x) = 2^{j/2}\,\mathrm{sinc}(2^j x - k)\exp(i3\pi(2^j x)) \ .$$

This looks very similar to a short-time Fourier transform atom, which does have an absolutely referenced phase. The difference is that the window size changes with frequency, and the frequencies are chosen non-redundantly.

Note that the "phase correction" term of $e^{i\pi\tau}$ is actually simpler, because τ is always an integer. Hence, the difference between the HWT and the DOST is a factor of -1 for odd values of τ (or k).

6.2 Algorithms

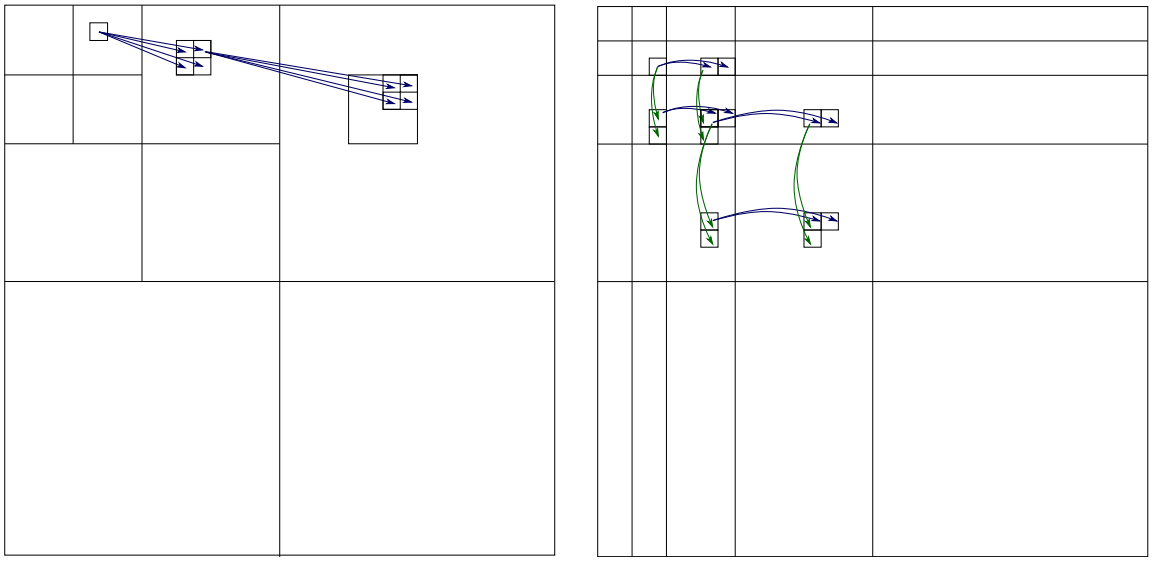
The fast algorithm for Harmonic wavelets [16] is nearly identical to the fast algorithm presented in [28] and used elsewhere, even in the order of coefficients. The only difference is the phase correction term in the DOST, which is simply a change of sign of alternating coefficients.

6.3 DCST vs. DCTWT

A couple of papers introduced the "DCTWT" or discrete cosine transform harmonic wavelet transform [15, 22]. This transform is very similar to the DCST which I proposed. In fact, for the one-dimensional version, the only difference is the type of DCT used.

6.4 Two-Dimensional Transforms

All of these transforms are treated differently when it comes to two-dimensional data. The main difference is the structure of the output data (i.e. how the frequency-space is partitioned). The partitioning, and hierarchical structure are shown in Figure 6.1. The original DOST is treated as a separable transform, as is the HWT. The result is the directed graph structure as I have shown before [13], where the horizontal and vertical frequencies are partitioned independently.



(a) (b)

Figure 6.1: Hierarchical structure of (a) wavelets, (b) DCST

The discrete cosine harmonic wavelet transform (DCHWT) is approached as wavelets, and given a quad-tree structure. Each refinement is treated as a separate filtering stage as in the analysis algorithm for orthogonal wavelets. For the DCST, the same structure as the DOST is used, but the two-dimensional transform is not separable, so each block is treated as a vector with the 2D-DCT being performed on it.

Chapter 7

Image Compression

We are interested in applications to image processing. A simple application is compression of images. Some preliminary tests were performed by Y. Wang [29], which showed that the DOST might perform well. I expanded on the tests comparing the DOST against more linear transforms and the DCST [13]. Most of this chapter covers these findings.

Most natural images are compressed using either the JPEG standard or the JPEG2000 standard. These standards follow the same process: linear transform compression.

1. Transform the colour space (to luminance and chroma YUV)
2. Perform a linear transform on each channel
3. Quantize the transformed signal
4. Entropy encode the quantized signal

The second step is the primary concern of this paper. The goal of the transform is to put the most significant information in a small number of components. If the majority of components in the transformed image are near zero, then the quantization step rounds them down to zero. A signal comprised mostly of zeros has low entropy.

We approximate the quantization step with hard thresholding, in which all coefficients below a threshold are set to zero. This effectively quantizes the smallest values, but none of the others, giving an idea of what the image would look like after this compression. As the threshold increases, more coefficients are removed and the degradation is more

severe. Unfortunately, without fully implementing some quantization scheme, measures of information content like entropy cannot be applied.

Thresholding is also effective for signal denoising [6]. If the noise is white noise, or the spectrum of the clean image is sparse, then much of the noise can be removed by thresholding the smaller components of the transformed signal. The tests for image compression, therefore provide evidence for how well the linear transforms work for image denoising. For denoising, a soft-threshold, which subtracts the threshold value from each coefficient, works a little better.

Another view of compression is the sparse representation of a signal. Minimizing the number of coefficients needed to faithfully reconstruct an image also works to compress the image. There are many different algorithms and choices of heuristic functions which approximate this optimization problem, with many parameters that require tuning. With all the complexity, a good comparison of two dictionaries would be nigh impossible. The results would also be quite similar to the tests performed here.

7.1 Measures of Image Quality

The most commonly used metric for image/signal quality to date is the peak signal to noise ration (PSNR). It is essentially equivalent to the $\mathcal{L}^2$ metric of a signal, because it is simply the log of the root mean square. However, it has the advantage of being normalized by the maximum possible value. The PSNR is defined as,

$$PSNR(x, y) = 20 \log_{10}(MAX) - 10 \log_{10}(\frac{\|x - y\|_2}{N}), \tag{7.1}$$

and is measured in decibels (dB), hence the logarithmic scale and factors of 10. In (7.1), MAX refers to the maximum signal value. For an 8-bit channel, that would be $MAX = 255$, and for an image normalized to $[0, 1]$, $MAX = 1$.

The PSNR is actually a rather poor metric for image quality. Many different types of deformations result in the same PSNR, even though their visual quality is vastly different [30]. For example, adding a noise of ± 1 with mean zero would give the same PSNR as adding 1 to every pixel (a slight increase in brightness).

Another measure of image quality that better represents human perception is the structural similarity (SSIM) [30]. The idea of SSIM is that after normalizing for luminance and contrast, the correlation of two patches measures similarity in structure. If the same pixels

are bright (or dark) in both patches, then they should look quite similar, and will have a high correlation. The SSIM between two image patches, x and y, is calculated as follows,

$$SSIM(x,y) = \left[\frac{2\bar{x}\bar{y} + \epsilon_1}{\bar{x}^2 + \bar{y}^2 + \epsilon_1}\right]\left[\frac{2s_{xy} + \epsilon_2}{s_x^2 + s_y^2 + \epsilon_2}\right], \tag{7.2}$$

where

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i\,, \qquad s_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y}) \tag{7.3}$$

and s_x^2 is s_{xy} taken with $x = y$. The two parameters, $\epsilon_i \ll 1$ are small constants relative to the maximum intensities, which provide numerical stability and accommodate the human visual system (Weber's Law).

SSIM provides a good measure of image quality on a small patch, but is not well suited to a large image. For large images, the mean-SSIM is used: the SSIM is calculated on a local patch around each pixel of the images, and the average of all these values is taken.

The SSIM and mean-SSIM are bounded between -1 and 1. If two images, X and Y, are identical, then $SSIM(X,Y) = 1$. A value of -1 essentially means that one image is the negative of the other. For all of our tests, we would expect a value in the range $(0, 1)$.

7.2 Method

To test the quality of each transform under compression, a degradation curve is computed. This measures the quality of the image after the least significant coefficients are set to zero through hard thresholding. The operation of thresholding approximates the effect of quantization, which rounds coefficients to a set of discrete values. Thresholding does not quantize any higher coefficients, so this is an approximation to the effect and compression rate. The result is similar to a rate-distortion curve.

By Parseval's equality, we know that the L_2 norm of a vector is preserved when transformed with an orthogonal basis. Therefore the image quality (particularly PSNR) monotonically decreases as more coefficients are set to zero. In addition, the rate of degradation increases, so more data points are required to track quality as compression increases.

The test images used were the well-known 512×512, 8 bits per pixel greyscale test images *Lena*, *Boat* and *Mandrill*. Each image was compressed at many levels using each

transform. The compressed images were compared against the originals to measure image quality. The quality of each image and the average quality across all images was then plotted with respect to the fraction of coefficients retained using PSNR and mean-SSIM. The PSNR is included for convention, while the mean-SSIM is used because it is a better measure of image quality.

All tests were performed in MATLAB. The DOST and its variations as well as the DCST were computed using my own code, and the wavelet transforms were calculated using MATLAB's Wavelet Toolbox. The mean-SSIM is calculated using the `ssim_index` function for MATLAB [31]. This implementation uses a window size, and regularization parameters that should work well for the test images.

The transforms used in the first test were the DOST (original, conjugate-symmetric DOST, and windowed DOST), DCST, 8x8 block DCT (JPEG), Daubechies 15 wavelets and biorthogonal 4.4 wavelets (JPEG2000).

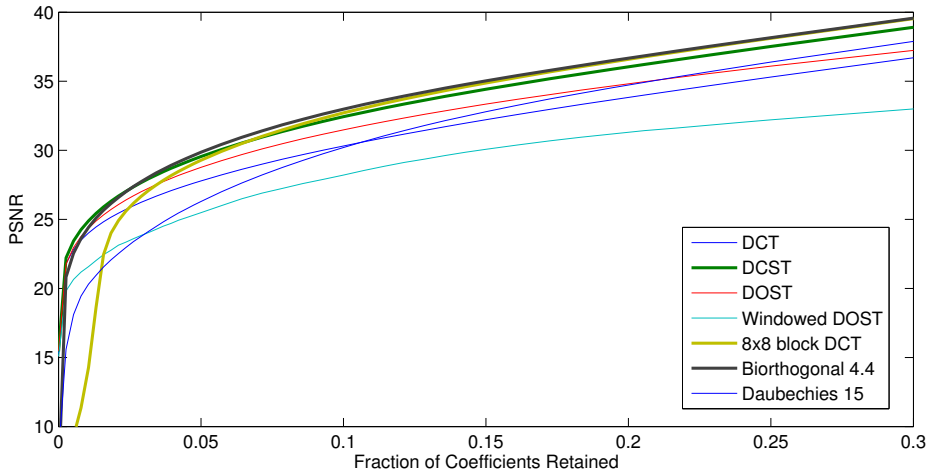
In Section 6.1, we showed the main difference between DCT-based harmonic wavelets (DC-HWT) and the DCT is how the frequency-space is partitioned. In order to examine the effect of different partitionings of frequency, the DCT based harmonic wavelet transforms were tested separately and compared against the DCST. These transforms were each implemented using the both the DCT-I and DCT-II as defined in Chapter 5, with both the wavelet-style quad-tree frequency space partitioning and the DOST-style partitioning. The same tests were performed with these transforms.

Most image compression techniques perform the transformation on sub-blocks of an image. I also implemented the transforms on sub-blocks to see how well they perform. The best block size was about 64x64 pixels. The measured quality was better, but the results did not change significantly. The *Boat* and *Mandrill* test images were compressed this way and can be seen in Figures 7.7 and 7.4.

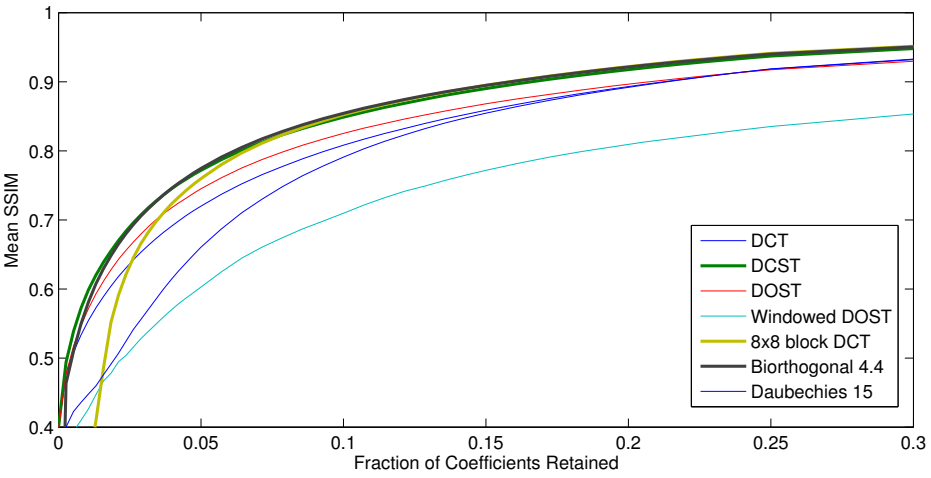
7.3 Compression Results

The average degradation curves for the transforms are shown in figure 7.1. The PSNR and SSIM both give the same ordering on quality. The biorthogonal wavelets used for lossy JPEG2000 compression perform best, followed closely by the 8x8-block DCT and the DCST.

The DOST and conjugate-symmetric DOST performed nearly identically, with the conjugate-symmetric just slightly better, so on the plots, only the DOST is shown. The



(a)



(b)

Figure 7.1: Averaged degradation curves for various transforms as more coefficients are retained. The PSNR (a), and mean-SSIM (b) have similar results.

(a) (b)

Figure 7.2: The *Boat* test image under compression. The original is shown in (a), and a compressed version using DCST in (b). The compressed version has lost 95% of its components, and has a mean-SSIM of 0.784 compared against the original

windowed DOST (or generalized Fourier family transform) is not well suited to image compression.

These results do not actually contradict the work by Y.Wang in [29]. His tests compared the DOST against Daubechies wavelets, which do not perform as well as the biorthogonal wavelets for compression of images. When compared against the better linear transforms, the DOST performs slightly worse.

There are some differences between the SSIM and PSNR for image quality in our graphs. For higher quality images, the PSNR shows more difference in quality between the transforms, but as we get above 90% of coefficients removed, the SSIM provides a larger spread, and the PSNR values become closer together. However, the overall behaviour is the same. That is, the order of transforms from best to worst does not change between mean-SSIM and PSNR. For this reason, only the mean-SSIM is shown for the rest of the analysis.

It should be noted that the image quality is quite poor at this level of compression as seen in figure 7.2. This is also the region in which the DCST starts to perform better than the others, which suggests that it is not more practical for compression.

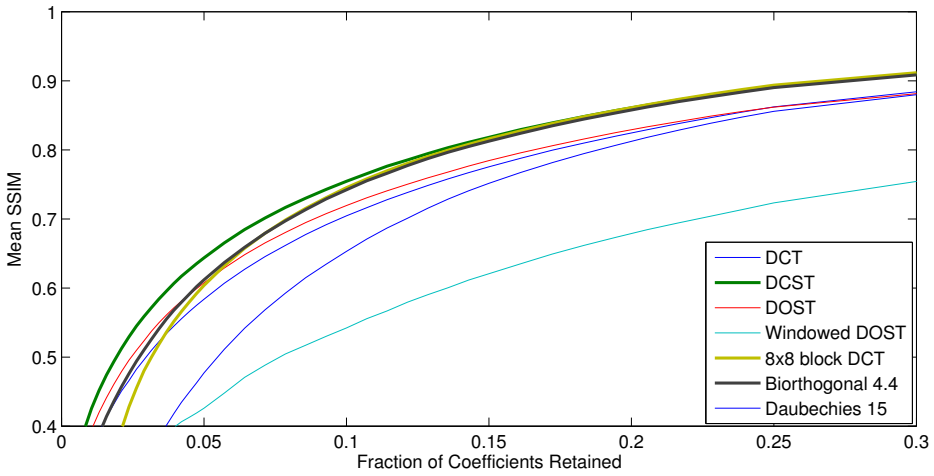


Figure 7.3: Mean-SSIM degradation curves for the *Mandrill* test image. The DCST does perform well in this case.

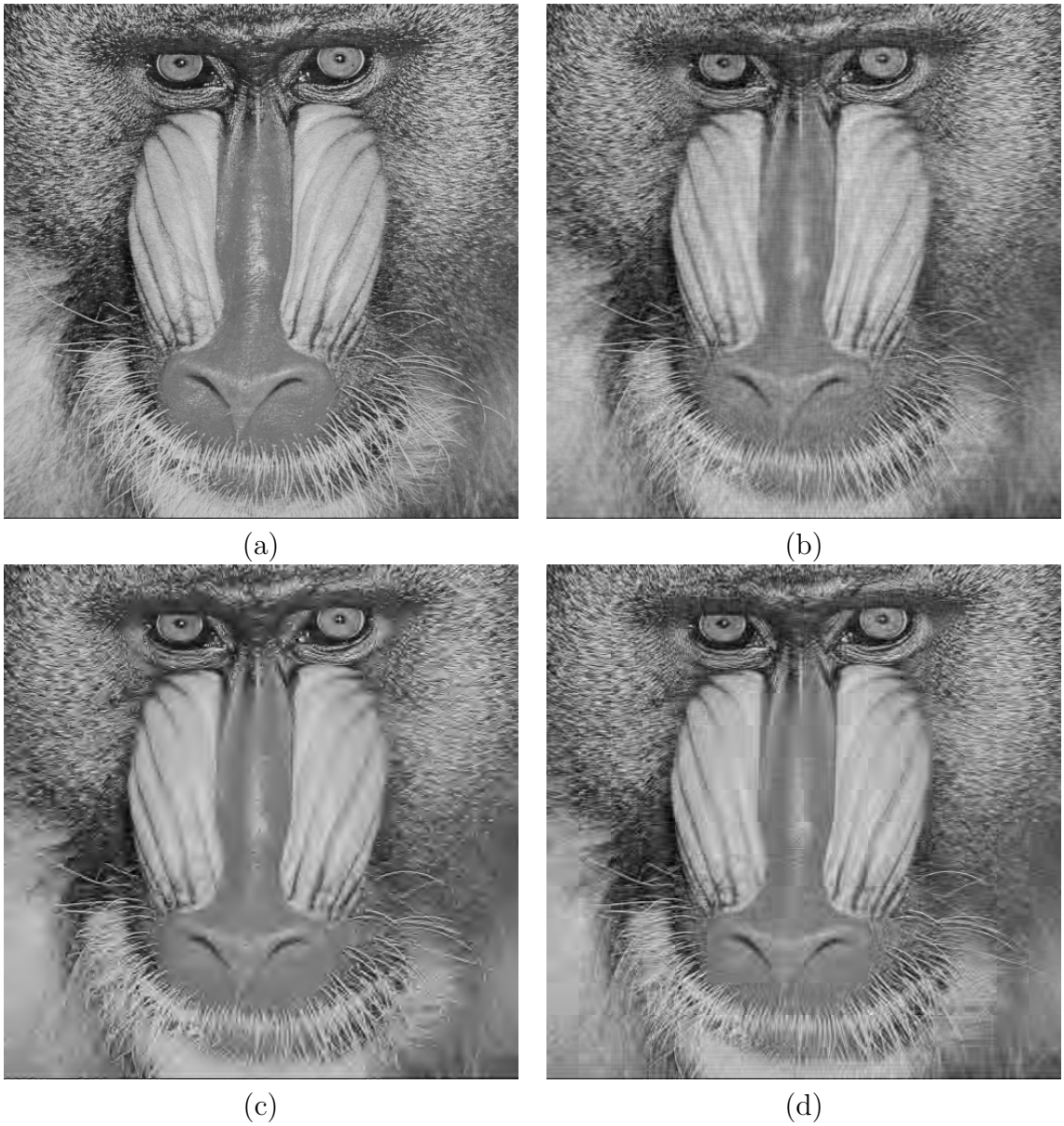
The *Mandrill* test image warrants closer inspection. The degradation curve for this image is shown in Figure 7.3. The DCST has higher mean-SSIM over a much wider range of compression rates than for the other images.

The reason might be seen by looking at the compressed images. Figure 7.4 shows the original *Mandrill* image and highly compressed copies. Compression using wavelets seems to remove much of the detail that describes texture in the image, while the DCST at least approximates that texture. These differences can even be seen when 20% of the coefficients are retained.

The *Lena* image by comparison is much smoother with less highly-textured area. For the *Lena* image, biorthogonal wavelets performed much better according to the mean-SSIM and PSNR. The corresponding degradation curve is shown in Figure 7.5. The *Boat* image had less difference between the transforms, but the degradation curves were still quite similar to those of the *Lena* image.

Figure 7.6 shows the results comparing the DCST with DC-HWT. The primary purpose was to compare the two different partitioning methods.

The quad-tree partitioning worked better than the DOST partitioning. However, when the more effective DCT (from JPEG) is used, this difference is quite small.



(a) (b)

(c) (d)

Figure 7.4: The *Mandrill* image compressed with 5% of its coefficients remaining. The original (a) is compressed using (b) DCST (m-SSIM=0.644), (c) biorthogonal wavelets (m-SSIM=0.612), and (d) DCST on 64x64 blocks (m-SSIM=0.655).

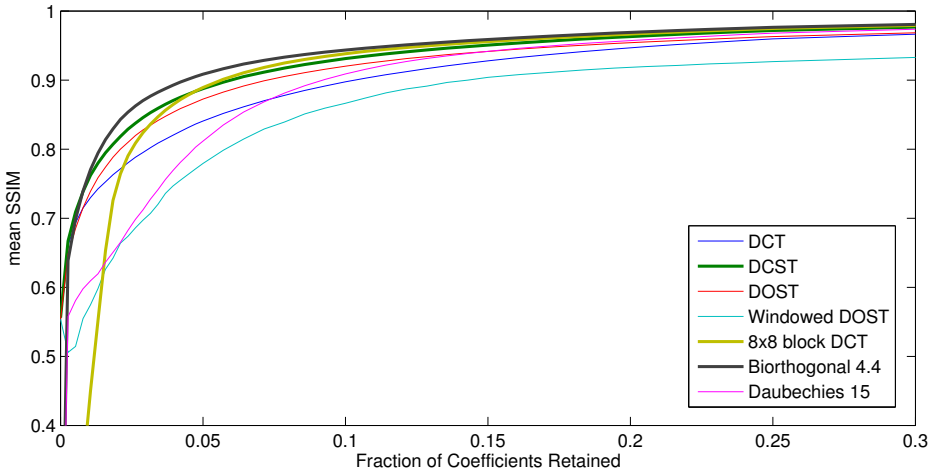


Figure 7.5: Mean-SSIM degradation curves for the *Lena* test image. The DCST does not perform as well as biorthogonal wavelets.

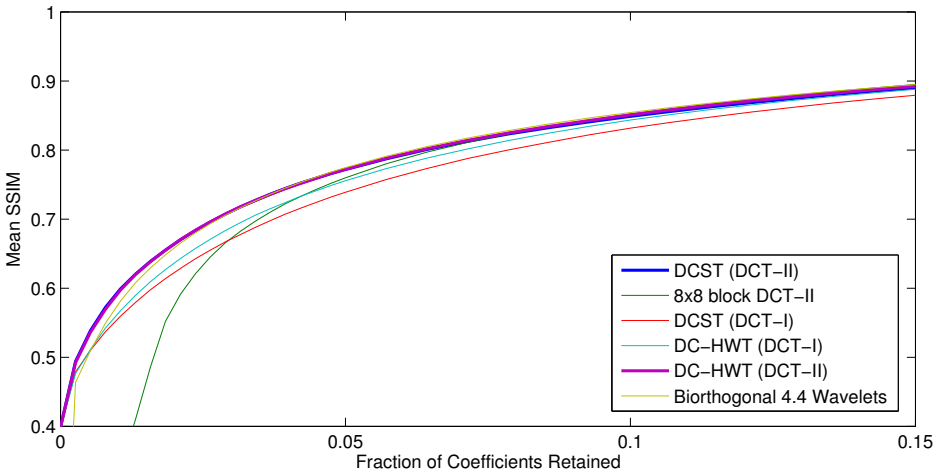


Figure 7.6: Averaged Mean-SSIM degradation curves of DCST, and DC-HWT using two types of DCT

7.4 Compression Artifacts

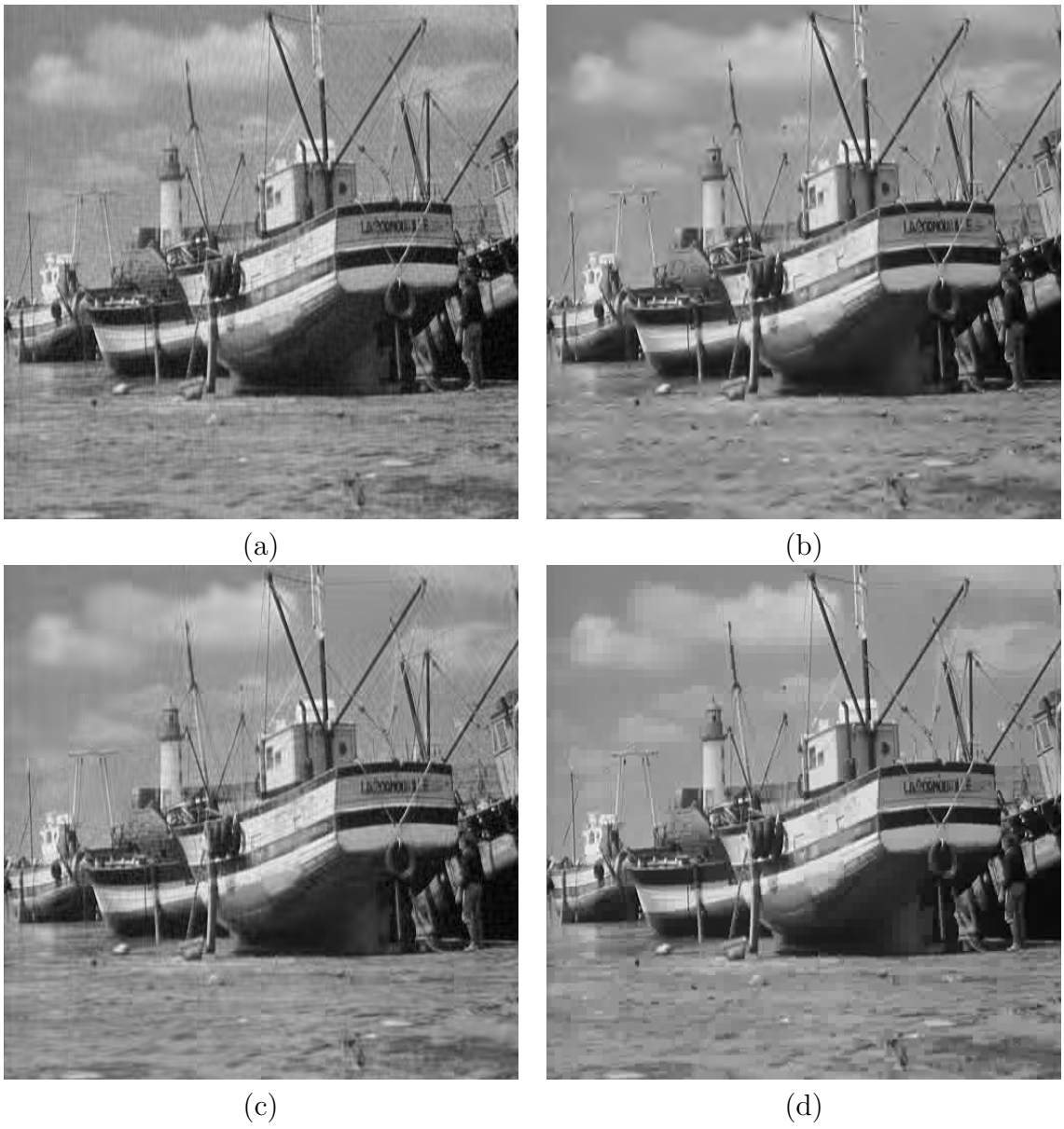
More examples of compressed images are in figure 7.7. The different effects of the transforms can be seen in these highly compressed images. The DCST has a slight blurring effect, but an interesting noise pattern which comes from ringing around edges. The wavelet transforms result in more blurriness as detail is removed. Some of the rigging is actually removed completely in in the wavelet version, while the DCST is better at retaining that kind of detail.

The transforms can also be performed on sub-blocks of the image. On these block-wise transforms the edges between blocks become apparent. This is more evident in the block-DCT transform than the block-DCST. In our testing, we found that the block-DCST performs best with larger (64x64) blocks. This prevents the ringing edges from propagating throughout the entire image.

The ringing around edges in the DOST methods is clearer in the *Lena* test image (figure 7.8), because there are fewer sharp edges interfering. The ringing from DOST artifacts is a very smooth wave coming from edges, but the DCST has a bit of checkerboarding because of the DCT basis. These patterns can also be seen in the basis functions for these transforms (figure 5.1).

The *Mandrill* image is useful to show how highly-detailed texture is affected by compression. In Figure 7.4 the blurring effect of wavelets is quite apparent. Compression using wavelets seems to remove much of the detail that describes texture in the image, while the DCST at least approximates that texture.

The 64x64 block DCST compressed image has a better mean-SSIM than the other images, but the edges between the blocks are very apparent on the nose of the *Mandrill*. The discontinuity in the ridges, and the sharp change of texture around the nostril are two harsh artifacts that are not picked up by the mean-SSIM.



(a) (b)

(c) (d)

Figure 7.7: The *Boat* image compressed with (a) DCST 5% coefficients, (b) biorthogonal wavelets 4.25% coefficients, (c) DCST on 64x64 blocks 4% coefficients, and (d) DCT on 8x8 blocks 4.9% coefficients. All have the mean-SSIM at 0.784.

(a) (b)

(c) (d)

Figure 7.8: The *Lena* test image under severe compression. The original is shown in (a), and a compressed version using (b) Biorthogonal wavelets, (c) DOST, (d) DCST. The compressed versions retain only 1% of the original image's coefficients, and have mean-SSIM's of (b) 0.765, (c) 0.734, (d) 0.759.

Chapter 8

Conclusion

This thesis summarizes most of the analysis of the Stockwell transform, discrete orthonormal Stockwell transform, and directly related transforms. The resolution of the identity of the ST was extended to work with two windows, and Stockwell's original inverse was shown to be a special case of this. The transform and its inverse were tested on several continuous functions for the first time to see how they behaved.

Mathematical properties of the ST were proven including behaviour under various operators, and boundedness and continuity. It was shown that the transform is not rotationally invariant, contrary to past claims.

The discrete versions, the DOST and DST, were defined in relation to each other, and variations of the DOST were reinterpreted in this way. This led to the creation of the DCST. Unfortunately, the DOST and DCST are nearly identical to harmonic wavelets. We were the first to outline the similarities.

More extensive tests for image compression were performed to compare the various orthonormal Stockwell transforms with current standards. These tests show that the DOST and its variations are not better than current standards.

Because the Stockwell transforms are the same as wavelet transforms up to a phase correction, the behaviour of the two should be similar for algorithms that ignore the phase of the components. Future research in this area may include analysis associated with the phase of the transform, rather than the magnitude. The quality of the ST as a dictionary for sparse representation may also be interesting, but our compression tests indicate that it likely will not perform better than other dictionaries in use.

References

[1] A. Bastari, S. Squartini, and F. Piazza. Discrete stockwell transform and reduced redundancy versions from frame theory viewpoint. *2007 Ieee International Symposium on Circuits and Systems, Vols 1-11*, pages 2319–2322, 2007.

[2] P. Boggiatto, C. Fernández, and A. Galbis. A group representation related to the stockwell transform. *Indiana University Mathematics Journal*, 58(5):2277–2296, 2009.

[3] R.A. Brown, M.L. Lauzon, and R. Frayne. A general description of linear time-frequency transforms and formulation of a fast, invertible transform that samples the continuous s-transform spectrum nonredundantly. *IEEE Transactions on Signal Processing*, 58(1):281–290, 2010.

[4] D. Damanik, A.M. Finkelshtein, A. Iosevich, V. Vougalter, Y. Wang, L. Riba, and M.W. Wong. Continuous inversion formulas for multi-dimensional stockwell transforms. *Mathematical Modelling of Natural Phenomena*, 8(1):215–229, 2013.

[5] I. Daubechies. *Ten lectures on wavelets*. Society for Industrial and Applied Mathematics, Philadelphia, PA; Philadelphia, Pa., 1992.

[6] D.L. Donoho. De-noising by soft-thresholding. *Information Theory, IEEE Transactions on*, 41(3):613–627, 1995. ID: 1.

[7] S. Drabycz, R.G. Stockwell, and J.R. Mitchell. Image texture characterization using the discrete orthonormal s-transform. *Journal of Digital Imaging*, 22(6):696–708, DEC 2009.

[8] J. Du, M.W. Wong, and H. Zhu. Continuous and discrete inversion formulas for the stockwell transform. *Integral Transforms and Special Functions*, 18(8):537–543, 2007.

[9] M. Eramian, R.A. Schincariol, R.G. Stockwell, R.P. Lowe, and L. Mansinha. Review of applications of 1d and 2d s transforms. In *Wavelet Applications IV*, volume 3078, pages 558–568, Bellingham, WA, United States. Society of Photo-Optical Instrumentation Engineers.

[10] P.C. Gibson, M.P. Lamoureux, and G.F. Margrave. Letter to the editor: Stockwell and wavelet transforms. *Journal of Fourier Analysis and Applications*, 12(6):713–721, 2006.

[11] I.S. Gradshteyn and I.M. Ryzhik. *Table of integrals, series, and products*. Academic Press, New York, 1965.

[12] K. Gröchenig. *Foundations of time-frequency analysis*. Birkhäuser, Boston, 2001.

[13] J. Ladan and E.R. Vrscay. The discrete orthonormal stockwell transform and variations, with applications to image compression. *Image Analysis and Recognition*, volume 7950 of *Lecture Notes in Computer Science*, pages 235–244. Springer Berlin Heidelberg, 2013.

[14] S.G. Mallat. *A wavelet tour of signal processing*. Academic Press, San Diego, CA, 1999.

[15] S.V. Narasimhan, M. Harish, A.R. Haripriya, and N. Basumallick. Discrete cosine harmonic wavelet transform and its application to signal compression and subband spectral estimation using modified group delay. *Signal, Image and Video Processing*, 3(1):85–99, 2009.

[16] D.E. Newland. Harmonic wavelet analysis. *Proceedings of the Royal Society of London.Series A: Mathematical and Physical Sciences*, 443(1917):203–225, October 08 1993.

[17] D.E. Newland. Harmonic and musical wavelets. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 444(1922):605–620, 1994.

[18] D.E. Newland. Ridge and phase identification in the frequency analysis of transient signals by harmonic wavelets. *Journal of Vibration and Acoustics, Transactions of the ASME*, 121(2):149–155, 1999.

[19] S.C. Pei, P.W. Wang, J.J. Ding, and C.C. Wen. Elimination of the discretization side-effect in the s transform using folded windows. *Signal Processing*, 91(6):1466–1475, 2011.

[20] R. Roopkumar. Stockwell transform for boehmians. *Integral Transforms and Special Functions*, 24(4):251–262, April 2013 2013.

[21] M. Schimmel, E. Stutzmann, and J. Gallart. Using instantaneous phase coherence for signal extraction from ambient noise data at a local to a global scale. *Geophysical Journal International*, 184(1):494–506, 2011.

[22] M. Shivamurti and S.V. Narasimhan. Analytic discrete cosine harmonic wavelet transform(adchwt) and its application to signal/image denoising. In *2010 International Conference on Signal Processing and Communications, SPCOM 2010*, 18 July 2010 through 21 July 2010 2010.

[23] R.G. Stockwell. A basis for efficient representation of the s-transform. *Digital Signal Processing: A Review Journal*, 17(1):371–393, 2007.

[24] R.G. Stockwell, L. Mansinha, and R.P. Lowe. Localization of the complex spectrum: The s transform. *IEEE Transactions on Signal Processing*, 44(4):998–1001, 1996.

[25] Y. Wang. *Efficient stockwell transform with applications to image processing*. PhD Thesis, Department of Applied Mathematics, University of Waterloo, Waterloo, Ont., 2011.

[26] Y. Wang and J. Orchard. Symmetric discrete orthonormal stockwell transform. *International Conference on Numerical Analysis and Applied Mathematics*, 1048:585–588, 2008.

[27] Y. Wang and J. Orchard. The discrete orthonormal stockwell transform for image restoration. *IEEE International Conference on Image Processing*, 2009.

[28] Y. Wang and J. Orchard. Fast discrete orthonormal stockwell transform. *Siam Journal on Scientific Computing*, 31(5):4000–4012, 2009.

[29] Y. Wang and J. Orchard. On the use of the stockwell transform for image compression. *Conference on Image Processing: Algorithms and Systems VII*, 7245:724504, 2009.

[30] Z. Wang, A.C. Bovik, H.R. Sheikh and E.P. Simoncelli: Image quality assessment: From error visibility to structural similarity, IEEE Transactions on Image Processing **13** (2004) 600–612

[31] Z. Wang, A.C. Bovik, H.R. Sheikh and E.P. Simoncelli: The SSIM Index for Image Quality Assessment `https://ece.uwaterloo.ca/~z70wang/research/ssim/`