

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Time Frequency Analysis of Wavelet and Fourier Transform

Karlton Wirsing

Abstract

Signal processing has long been dominated by the Fourier transform. However, there is an alternate transform that has gained popularity recently and that is the wavelet transform. The wavelet transform has a long history starting in 1910 when Alfred Haar created it as an alternative to the Fourier transform. In 1940 Norman Ricker created the first continuous wavelet and proposed the term wavelet. Work in the field has proceeded in fits and starts across many different disciplines, until the 1990's when the discrete wavelet transform was developed by Ingrid Daubechies. While the Fourier transform creates a representation of the signal in the frequency domain, the wavelet transform creates a representation of the signal in both the time and frequency domain, thereby allowing efficient access of localized information about the signal.

Keywords: time-frequency analysis, Fourier transform, wavelet transform, signal processing, vanishing moment

1. Introduction

The Fourier transform has been the basis of digital signal processing since the development of the fast Fourier transform in 1965 by Cooley and Tukey in [1]. Its use for analysis goes back much farther with the development of the Fourier transform by Jean Baptiste Joseph Fourier in 1807 as a solution to thermodynamic equations. By using the Fourier transform, we can take any signal and obtain the amplitude of the sinusoids needed to recreate it. Then we can use this information to obtain the power spectrum of the signal, or we modify the amplitudes and take the inverse Fourier transform of the signal, which then filters the signal.

A fundamental limitation of the Fourier transform is that the all properties of a signal are global in scope. Information about local features of the signal, such as changes in frequency, becomes a global property of the signal in the frequency domain. There have been various methods proposed to address this limitation; the main two are the windowed Fourier transform and wavelets.

Gabor [2] created the windowed Fourier transform in 1946. It applies a window function of a short duration to the signal and the Fourier transform is applied to the resulting data. This method is frequently used; however, there are two limitations with this method. The first is that, since the filtering window is constant, it creates problems if the feature is larger or shorter than the window. The second is that the time resolution is the same for high frequencies as it is for low frequencies. Since as frequency increases, so does the rate of change of the signal, higher frequency

signals can have more information in the same period of time as lower frequency signals, and so require a higher time resolution.

Wavelets overcome both these limitations in that the window is scaled in both time and frequency. The term wavelet was introduced by Ricker [3] in 1940 to describe the limited duration functions that he created to model seismic phenomena. The first wavelet was created earlier, in 1910, by Haar [4] as an alternative to the Fourier transform developed in 1807 by Fourier [5]. Work on the wavelet transform preceded slowly through the twentieth century until the 1980's when work on them increased dramatically with the development of the continuous wavelet transform. In the 1990's, the discrete wavelet transform and its inverse were developed, allowing filtering and compression of signals.

The wavelet transform has many more modes of operation and other options than the Fourier transform. This is one of the key problems with the use of wavelets; we can feel overwhelmed by all the options we have available with them. This chapter will go through some of these options and demonstrate their use.

2. Fourier transform

The Fourier Transform was first published in 1822 by Joseph Fourier [6]. It converts a mathematical function from the time domain to the frequency domain. This enables us to find new properties of the function that would otherwise be hidden. There are several different variations of the Fourier transform equation. In this chapter, we are using the traditional electrical engineering equation

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt \quad (1)$$

to convert $f(t)$ to the frequency domain.

The Fourier transform itself is for continuous functions. The Discrete Fourier transform was developed for astronomical observations. The goal was to calculate a trigonometric equation for the orbit of an object in the sky based on observations of its ascensions and declinations at various points in time. Most datasets consist of discrete points sampled in time. These can be converted to the frequency domain as well with the discrete Fourier transform. The computational complexity of this is $O(n^2)$.

An interesting note about the Fast Fourier Transform is that it actually predates the Fourier Transform. While the Fast Fourier Transform that we now use was published in a paper by Cooley and Tukey [1] in 1967, a functionally equivalent algorithm was found in an unpublished work by Carl Friedrich Gauss [7] that is presumed to date to 1805. A fascinating history of the Fast Fourier Transform is in [8]. Gauss was computing the discrete Fourier transform of 12 points and noted that the problem could be broken down into subproblems that could simplify the number of steps used [5].

The Fast Fourier Transform reduces the computational complexity of the Discrete Fourier Transform from $O(n^2)$ to $O(n \log_2 n)$. This enables efficient computation of time series. **Table 1** shows how the computational complexity increases for an $O(n^2)$ process versus an $O(n \log_2 n)$ process. The difference grows between the two processes until at 1 million data points, the discrete Fourier transform would require over 50,000 times the amount of time that the Fast Fourier transform would require.

The drawback with the Fourier transform is that all signal information is across the entire range of the transform. As stated in [9], "A local characteristic of the

n	$O(n^2)$	$O(n \log_2 n)$	Ratio
10	100	34	2.94
100	10,000	665	15.04
1000	1,000,000	9966	100.34
10,000	100,000,000	132,878	752.57
100,000	10,000,000,000	1,660,965	6020.60
1,000,000	1E+12	1,9931,569	50171.66

Table 1.
Computational complexity of $O(n^2)$ versus $O(n \log_2 n)$.



Figure 1.
Opening of Beethoven's 5th Symphony [11].

signal becomes a global characteristic of the transform”. As illustrated by other authors [10], the best way that this can be explained is by a score of music as shown in **Figure 1**.

The score consists of many different notes, each with a finite duration, each happening at a precise time. A Fourier transform of this signal gives you the average amplitude of the individual frequencies over the entire piece, but obscures the duration and location of the notes. The Fourier power spectrum of music often approximates that of pink ($1/f$) noise [12]. That information is not lost, since the Fourier transform is reversible, but is encoded in the phase of the Fourier transform.

3. Windowed Fourier transform

In 1946, Gabor [2] proposed the windowed Fourier transform as a way to deal with this problem. In it, a window function of a short duration is applied to the signal and the Fourier transform is taken. This is repeated at different locations in the signal. An example of the use of Hamming window function is shown in **Figure 2**.

One limitation of the windowed Fourier transform is that the window length is constant. When a signal feature is much shorter than the window, information about it can be difficult to extract, since the any local property within the time span of the window becomes a global property of the Fourier transform of the window, as noted previously. Conversely, when a signal feature is larger than the windowing function, information about it spans multiple windows, and can also be difficult to extract.

Another limitation is that the time resolution for the windowed Fourier transform is the same for high frequency signals as it is for low frequency signals. The Heisenberg uncertainty principle states that the time resolution of the window is inversely proportional to the frequency resolution. Since a high frequency signal changes much faster than a low frequency signal, it would be ideal to have a transform with better time resolution for high frequency portions of the signal, and better frequency resolution for lower frequency portions of the signal.

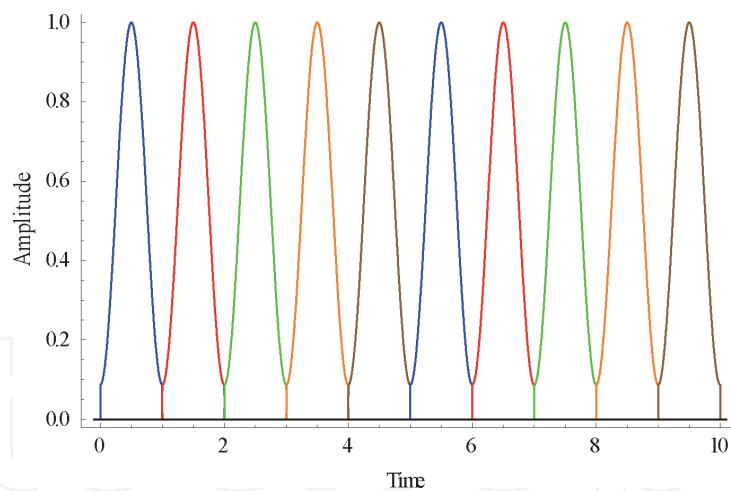


Figure 2.
Multiple Hamming window functions at successive locations in time.



Figure 3.
Musical notation for middle C and C_6 .

Going back to the music score, we can see this by looking at two different notes, Middle C, and one that is two octaves higher, called C_6 , as shown in **Figure 3**.

The frequency for middle C is 261.63 Hertz, and the frequency for C_6 is 1046.50 [13]. The frequency for C_6 is quadruple the one for Middle C, which means that for every complete cycle of the middle C note, four complete cycles of C_6 have occurred, as shown in **Figure 4**. The windowed Fourier transform would have the limitation that both notes would be treated equally, when the time resolution for C_6 needs to be 4 times that of middle C for analysis purposes.

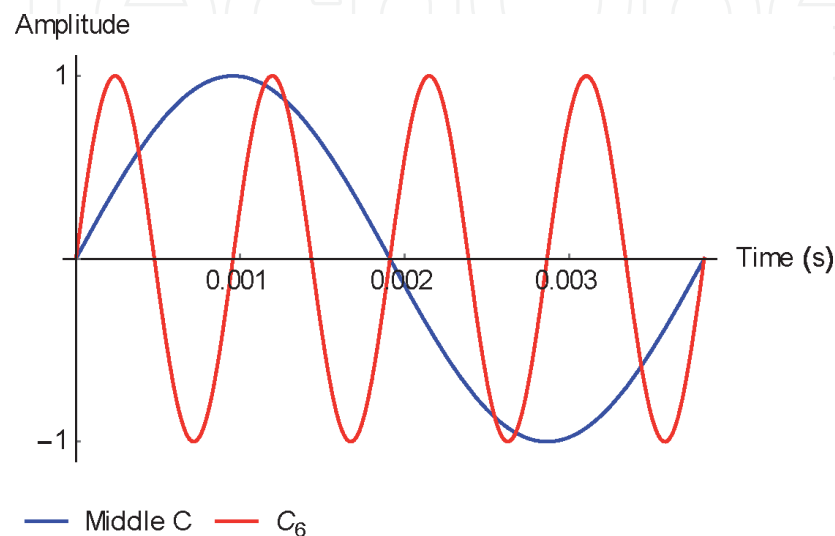


Figure 4.
Amplitude graph for Middle C and C_6 .

4. Wavelet transform

The wavelet transform overcomes the limitation of the windowed Fourier transform by scaling the bandwidth of the filter inversely to the frequency. According to [14], while each box of the windowed Fourier transform has the same bandwidth, each level of the wavelet transform has the same Q as defined as

$$Q \equiv \frac{\Delta f}{f_0}. \tag{2}$$

This gives the transform the desired time resolution for the higher frequency portions of the signal and the desired frequency resolution for lower frequency portions.

5. Continuous wavelet

The continuous wavelet has a long history spanning from the 1940’s to present. In 1940, Norman Ricker first proposed the term wavelet and various mathematical functions to model seismic waves as they traveled through the Earth’s crust in [3]. He further refined this in a series of papers [15–17]. This was the first continuous wavelet. The functions in the time domain are given by

$$\psi_{f_m}(t) = (1 - 2\pi^2 f_m^2 t^2) e^{-\pi^2 f_m^2 t^2}, \tag{3}$$

called the three-loop equation, and

$$\psi_{f_m}(t) = t e^{-2\pi^2 f_m^2 t^2}, \tag{4}$$

called the two-loop equation [18]. Graphs for both of these are in **Figure 5**.

The next development for continuous wavelets was in the 1980’s by Grossman and Morlet, and expanded on by Stephen Mallat and others [19]. The term continuous wavelet refers to the fact that it can be scaled to any time scale. Discrete wavelets can only use specific time scales, usually a power of 2.

Wavelet analysis centers around the use of a wavelet function, also called the mother function in literature, traditionally represented by the Greek letter upsilon (ψ). A key requirement is that it has finite energy, i.e.

$$\int |\psi(t)|^2 dt < \infty. \tag{5}$$

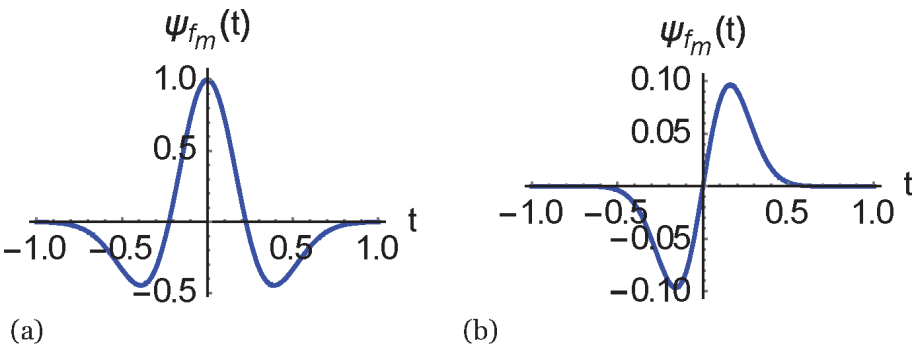


Figure 5.
Three loop (a) and two loop (b) Ricker wavelet equation with $f_m = 1$.

The energy of the wavelet function is usually one. Functions such as sine and cosine cannot be used as analyzing functions, because they violate this condition by having infinite energy. There is an implicit requirement that, while it has finite energy, it must have some energy, so the integration of the function must be greater than zero.

The second requirement is known as the admissibility condition, which states that the Fourier transform of the wavelet function cannot have a zero-frequency component, i.e.

$$C_g = \int \frac{|\hat{\psi}(f)|^2}{|f|} df < \infty \quad (6)$$

This can only be satisfied if $\hat{\psi}(f) = 0$, however this requirement is not absolute. The Gabor wavelet is a complex wavelet that violates the admissibility condition. The Morlet wavelet is a real valued wavelet that has a small but greater than zero value for the zero-frequency value for its Fourier transform.

A third condition is usually that the wavelet function must have zero mean, which means that it must oscillate, hence be a wavelet. Mathematically this is [20]

$$\int \psi(t) dt = 0. \quad (7)$$

Another condition is that the wavelet function has effective support. While the wavelet functions for the continuous wavelets are usually mathematical functions that extend to infinity, effective support means that the wavelet functions are effectively zero outside of a certain range. Since the continuous wavelet functions asymptotically approach 0 as x goes to either ∞ or $-\infty$, the choice of the boundary of this range is a bit arbitrary and can vary from paper to paper.

6. Continuous wavelet transform

Morlet and Grossman formalized the continuous wavelet transform in 1984 in [21]. For the continuous wavelet transform, the wavelet function itself is shifted in time and is scaled to do the wavelet transforms [22] as the following equation illustrates:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right), a, b \in \mathbb{R}, a \neq 0. \quad (8)$$

The continuous wavelet transform is defined as the integration of the function to be analyzed with the complex conjugate of the wavelet function:

$$\text{CWT}\{f(t), a, b\} = \int_{-\infty}^{\infty} f(t) \psi_{a,b}^*(t) dt \quad (9)$$

In some papers such as [22], you will see the definition of the continuous wavelet transform without the complex conjugate definition. Since most wavelet functions are real valued and not complex, both definitions are equivalent, since the complex conjugate of a real number is equal to that number. The difference only comes up when the wavelet function is complex, such as the Gabor wavelet.

An alternate formula for the continuous wavelet transform is

$$W_n(s) = \sum_{n'=0}^{N-1} x_{n'} \psi^* \left[\frac{(n' - n)\delta t}{s} \right], \tag{10}$$

where $W_n(s)$ is the transformed sequence, $x_{n'}$ is the original sequence, and ψ^* is the complex conjugate of the analyzing wavelet function, n represents the time shift or dilation, and s represents the scale. Usually the time shift is calculated over the total number of data points of the function, and s goes over the scales that are being analyzed to give a two-dimensional picture of the data [23].

7. Discrete wavelets

The first discrete wavelet was created in 1910 by Alfred Haar as an alternative to the Fourier transform. This consists of two functions as shown in **Figure 6**, one a scaling function and a wavelet function. The scaling function is the unit step function and the wavelet function consists of offsets from that.

One of the drawbacks of the continuous wavelet transform is that it creates a lot of redundant data, since the coefficients between the scales are highly correlated. Ingrid Daubechies developed the theory of discrete wavelets in 1988, which generates compact data by eliminating the redundancy. Daubechies created an entire family of wavelet functions with the Haar wavelet forming the first level of the Daubechies wavelet.

The wavelet function for discrete wavelets is modified to

$$\psi_{j,k}(t) = \frac{1}{\sqrt{s_0^j}} \psi \left(\frac{t - k\tau_0 s_0^j}{s_0^j} \right), j, k \in \mathbb{Z}, s_0 > 1, \tag{11}$$

where s_0 is the scale of the wavelet, usually 2 [20]. This condition as well as the condition that j and k are integers restricts the wavelet to only certain scales. The wavelet function has the properties of finite energy, oscillation, and the admissibility condition of the continuous wavelets, as well as the properties of compact support, vanishing moments, and orthogonality.

Compact support means that the wavelet function is defined by a series of coefficients over a finite region, and is zero at all other places. This contrasts with the continuous wavelets, which, as mentioned, are mathematical functions and have effective support in which the function continues to infinity, but is effectively zero outside of a finite range.

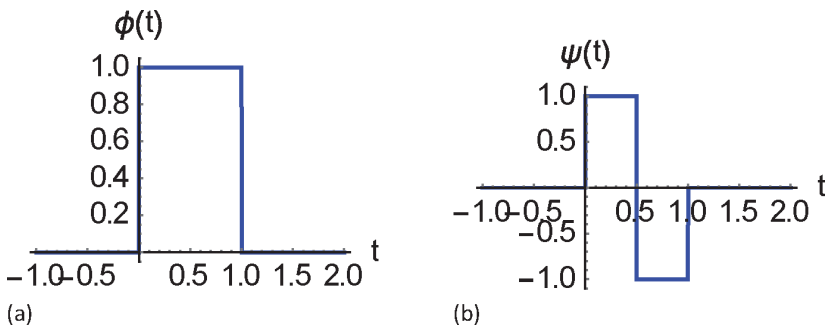


Figure 6.
Scaling (a) and wavelet (b) functions for Haar wavelet.

Vanishing moments are obtained when the following condition defined mathematically as

$$\int x^k \psi(x) dx = 0 \quad (12)$$

holds true for all integers $0 \leq k < N$, where N is the number of vanishing moments of the function [24]. This property is useful for analyzing functions that have an additive polynomial trend function given by

$$f(x) = g(x) + N(x). \quad (13)$$

Here, $g(x)$ is the function to be analyzed and $N(x)$ is the polynomial trend function (also termed a nuisance function in Economics).

The orthogonality condition removes the redundancy of the continuous wavelet transform. As stated earlier, the discrete wavelet transform can only be used at certain scales, most often a power of 2. Mathematically it is stated as

$$\int \psi_{j,k}(t) \psi_{m,n}^*(t) dt = \begin{cases} 1 & \text{if } j = m \text{ and } k = n \\ 0 & \text{otherwise} \end{cases}. \quad (14)$$

An orthogonal basis ensures that the signal is represented in the most compact way possible. However, by removing all the redundant information, this also removes information to handle shift variance. The exact same function sampled at two different places can yield very different results. In order to deal with this, some discrete wavelet transforms retain some of this redundant information.

Each wavelet of the discrete wavelet family consists of two functions, a wavelet function (ψ), as in the continuous wavelet families, and also a new function called a scaling function (ϕ). In literature, these are termed the mother and father functions respectively. The scaling function has its own admissibility condition, which ensures that it has the zero-frequency component that the wavelet function does not:

$$\int \phi(x) dx = 1. \quad (15)$$

This is necessary so that a discrete wavelet transform terminates in a finite number of steps and can completely regenerate the information in the signal [20]. Otherwise, the zero-frequency component could never be captured, since no amount of scaling value can cause the wavelet filter to have a zero-frequency component.

In addition, as specified in [25] the scaling equation is defined in terms of a finite set of coefficients p_k that are defined by the following equation

$$\phi(x) = \sum_k p_k \phi(2x - k), \quad (16)$$

that adheres to the following conditions as specified in [25] as well:

$$P(z) = (1/2) \sum_k p_k z^k, \quad (17)$$

$$P(1) = 1 \text{ (Averaging Property)}, \quad (18)$$

$$|P(z)|^2 = |P(-z)|^2 = 1 \text{ for } |z| = 1, \text{ and} \quad (19)$$

$$|P(e^{it})| > 0 \text{ for } |t| \leq \pi/2 \quad (20)$$

The wavelet function is defined by

$$\psi(x) = \sum_{k \in \mathbb{Z}} (-1)^k \overline{p_{l-k}} \phi(2x - k), \tag{21}$$

where l is the length of the set of coefficients, so that the wavelet coefficients are basically the scaling coefficients in reverse order with alternating signs. These coefficients are used to implement the discrete wavelet transform as a filter bank of Finite Impulse Response (FIR) filters. Graph of the scaling and wavelet functions for Daubechies level 2 wavelet are shown in **Figure 7** and the frequency response is shown in **Figure 8**. As with the Haar wavelet, the wavelet function is a high pass filter and the scaling function is a low pass filter. Both are symmetric around $\pi/2$.

Different papers and software implementations have different coefficients for the Haar and Daubechies wavelet, depending on how they are normalized and whether the scale parameter from Eq. (8) is included is included in the filter. The coefficients for the Haar and the Daubechies level 2 wavelet are in **Tables 2** and **3** with b defined by the implementation. Mathematica uses 2 for b , which would normalize the sum of the coefficients to 1. PyWavelets uses $\sqrt{2}$ for b . In any implementation, the filter coefficients for the wavelet filter are the coefficients for the scaling filter in reverse order with every other coefficient multiplied by -1 .

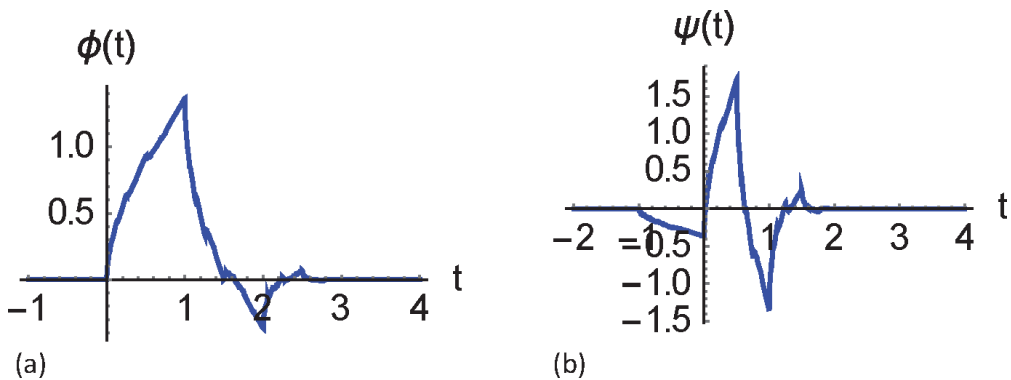


Figure 7.
Scaling (a) and Wavelet (b) functions for Daubechies level 2 wavelet.

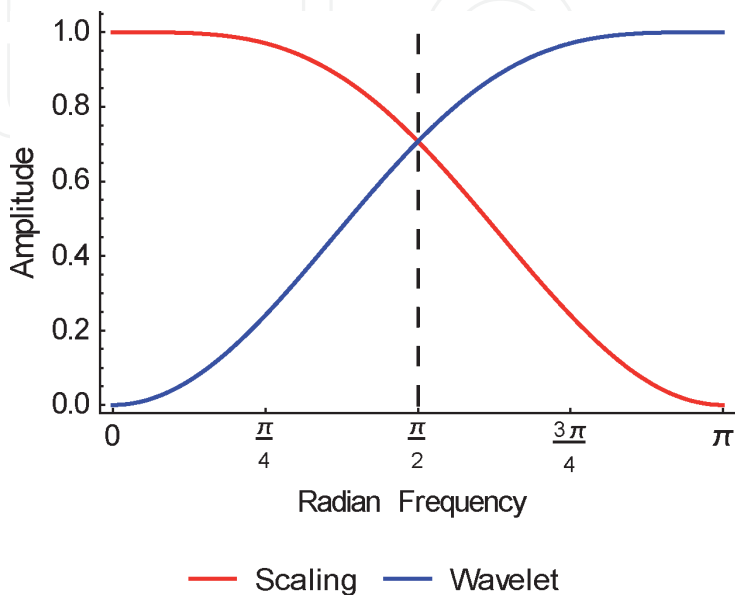


Figure 8.
Frequency response of scaling function (red) and wavelet function (blue) for Daubechies level 2 wavelet.

Scaling coefficients		Wavelet coefficients	
c_0	$1/b$	d_0	$1/b$
c_1	$1/b$	d_1	$-1/b$

Table 2.
Coefficients for Haar scaling and wavelet functions.

Scaling coefficients		Wavelet coefficients	
c_0	$\frac{1+\sqrt{3}}{4b}$	d_0	$\frac{1-\sqrt{3}}{4b}$
c_1	$\frac{3+\sqrt{3}}{4b}$	d_1	$\frac{-3+\sqrt{3}}{4b}$
c_2	$\frac{3-\sqrt{3}}{4b}$	d_2	$\frac{3+\sqrt{3}}{4b}$
c_3	$\frac{1-\sqrt{3}}{4b}$	d_3	$\frac{-1-\sqrt{3}}{4b}$

Table 3.
Coefficients for Daubechies level 2 scaling and wavelet functions.

8. Discrete wavelet transform

The class of discrete wavelet functions has many transforms available with the discrete wavelet transform in **Figure 9** the most common. Since this was the transform introduced with the Haar wavelet, it is sometimes referred to as the Haar transform [26] as well as the decimated wavelet transform [10]. Essentially, it works as a pyramid algorithm, where the number of coefficients of each lower level is roughly twice that of the preceding level, but each coefficient is influenced by half as much of the data set as the preceding level. Each level has two sets of coefficients, one is called coarse and the other is called details.

In **Figure 9**, g is the scaling filter defined by the set of scaling filter coefficients and h is the wavelet filter defined by the set of wavelet filter coefficients. At each level, the detail coefficients (W) are outputs, except for the final level, where the coarse coefficients (V) are given as outputs as well. Collectively, this set of coefficients contains enough information to reconstruct the signal perfectly.

One key part of the discrete wavelet transform is the down sampling operator, which is a function that removes every other position from a sequence. An example would be the sequence $\{a, b, c, d, e, f, g, h\}$ would be $\{a, c, e, g\}$ or $\{b, d, f, h\}$, after

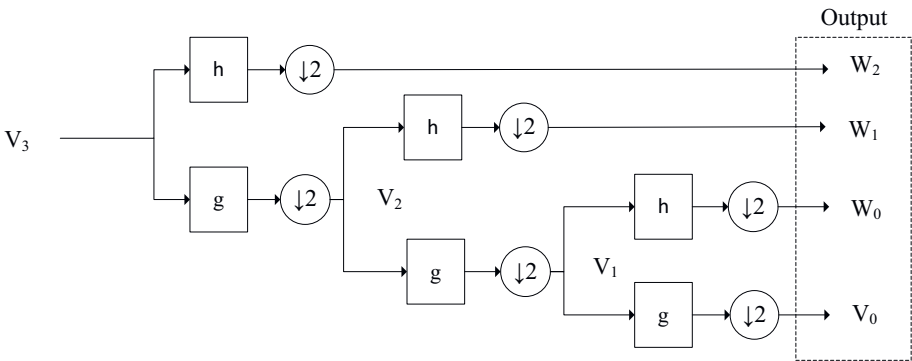


Figure 9.
Diagram of a three-level discrete wavelet transform.

the down sampling operator is applied, depending on whether the even or the odd positions are eliminated. Both are valid, however, by convention with the discrete wavelet transform, the even positions are eliminated, leaving only the odd positions. The down sampling operator is what makes the discrete wavelet transform a pyramid function and also reduces the set of coefficients to the minimum amount necessary to reconstruct the signal.

A problem with the decimation operator is aliasing. This is when different sequences map to the same sequence after the application of the operator. An example would be that the sequences {a, b, c, d, e, f, g} and {a, h, c, i, e, j, g} would both map to the sequence {a, c, d, g}. Therefore, just given the sequence {a, c, d, g}, it would be impossible to reconstruct the original. The filters of the discrete wavelets are designed to compensate for this, ensuring that the original sequence can be recovered. The combination of these filters with the down sampling operator is referred to as decimation.

The discrete wavelet transform also has an inverse transform. This process combines as described in **Figure 10** to form a perfect reconstruction of the signal, where \tilde{g} is the inverse scaling filter coefficients and \tilde{h} is the inverse wavelet filter coefficients. Just as the discrete wavelet transform had the decimation operator, the inverse transform has the upsampling operator. This takes a sequence and inserts 0 at every other position. For example, the sequence {a, c, d, g} would be {a, 0, c, 0, d, 0, g, 0} after the operator is applied.

Implementing the discrete wavelet transform as a finite impulse response filter and using decimation gives it a computational complexity of $O(n)$. As **Table 4** shows, an $O(n)$ process can be much faster than an $O(n \log_2 n)$ process such as the fast Fourier Transform. At 1 million samples, an $O(n)$ process requires almost 20 times less operations than an $O(n \log_2 n)$ process (**Table 4**).

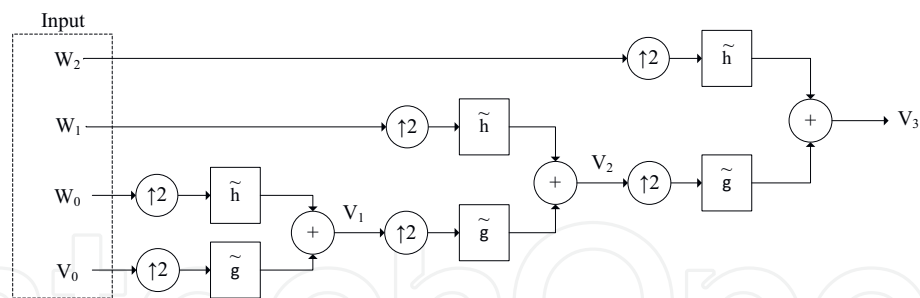


Figure 10.
Diagram of a three-level inverse discrete wavelet transform.

n	$O(n \log_2 n)$	$O(n)$	Ratio
10	34	10	3.40
100	665	100	6.65
1000	9966	1000	9.97
10,000	132,878	10,000	13.29
100,000	1,660,965	100,000	16.61
1,000,000	19,931,569	1,000,000	19.93

Table 4.
Computational complexity of $O(n \log_2 n)$ versus $O(n)$.

9. Stationary wavelet transform

Another wavelet transform for discrete wavelet functions is the stationary wavelet transform, also known as the undecimated discrete wavelet transform. Essentially the stationary wavelet transform is the discrete wavelet transform without the decimation operation for the data. Whereas the number of coefficients for each level is half that of the preceding level in the discrete wavelet transform, the number of coefficients is the same for each level in the stationary wavelet transform.

The procedure is diagrammed in **Figure 11**, where g_n is the set of the scaling filter coefficients and h_n is the set of the wavelet filter coefficients. The reason that the scaling filter and wavelet filter coefficients are different for each level is that instead of the decimation operator being applied to the wavelet data coefficients after each level, the upsampling operator is applied to the wavelet and scaling filter coefficients. The wavelet and scaling coefficients for each level are upsampled from the previous level, as shown in **Figure 12**.

Like the discrete wavelet transform, the stationary wavelet transform has an inverse transform, as shown in **Figure 13**. The difference between this and the inverse discrete wavelet transform is the absence of the upsampling operator. As with the stationary wavelet transform, the filter coefficients for the inverse stationary wavelet transform are changed instead of the data. In this case, the filters are down sampled. The retention of redundant data in the stationary wavelet

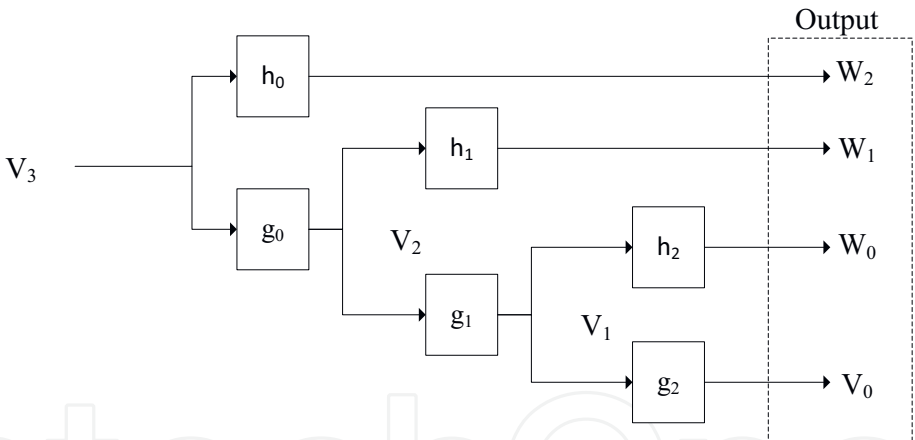


Figure 11.
Diagram of a three-level stationary wavelet transform.

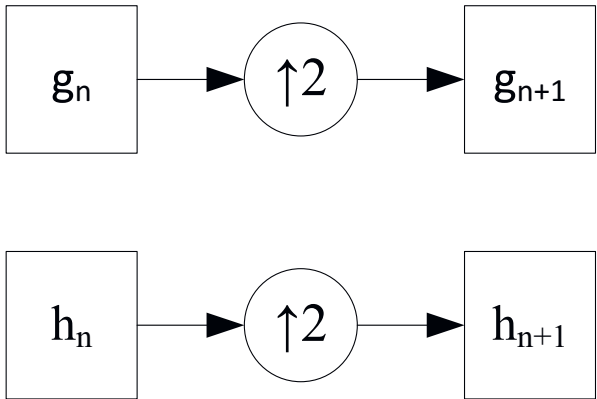


Figure 12.
Diagram of a filter upsampling for the stationary wavelet transform.

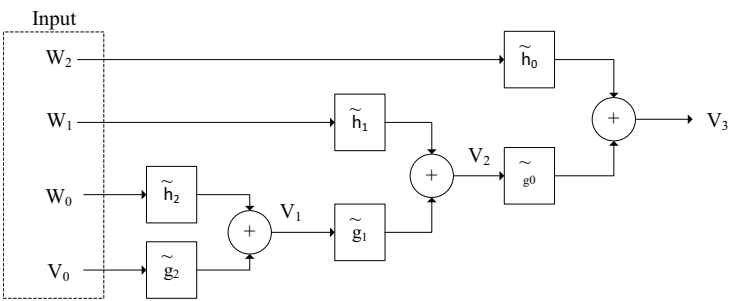


Figure 13.
Diagram of a three-level inverse stationary wavelet transform.

transform helps to make it translation invariant, which is useful for filtering applications (**Figure 13**).

Since the decimation step is not used, the stationary wavelet transform has a computational complexity of $O(n \log_2 n)$, the same as the Fast Fourier Transform. However, there is also memory complexity to consider. While the Fast Fourier Transform and the Discrete Wavelet Transform has an $O(n)$ memory complexity, the stationary wavelet transform has an $O(n \log_2 n)$ memory complexity. Therefore, the output will always be larger than the input.

10. Discrete wavelet packet transform

The two previous transforms applied the detail and the coarse filters to the data at each level. The output of the coarse filter is given as the input to the next level and the output of the detail filter at that level is included in the set of the outputs of the transform. In the final level, the output of both the detail and the coarse filters were included in the set of outputs of the transform; however, that is not the only possibility. The packet transform creates a binary tree where the detail and coarse filters are applied to each node, diagrammed in **Figure 14**. The output of the detail

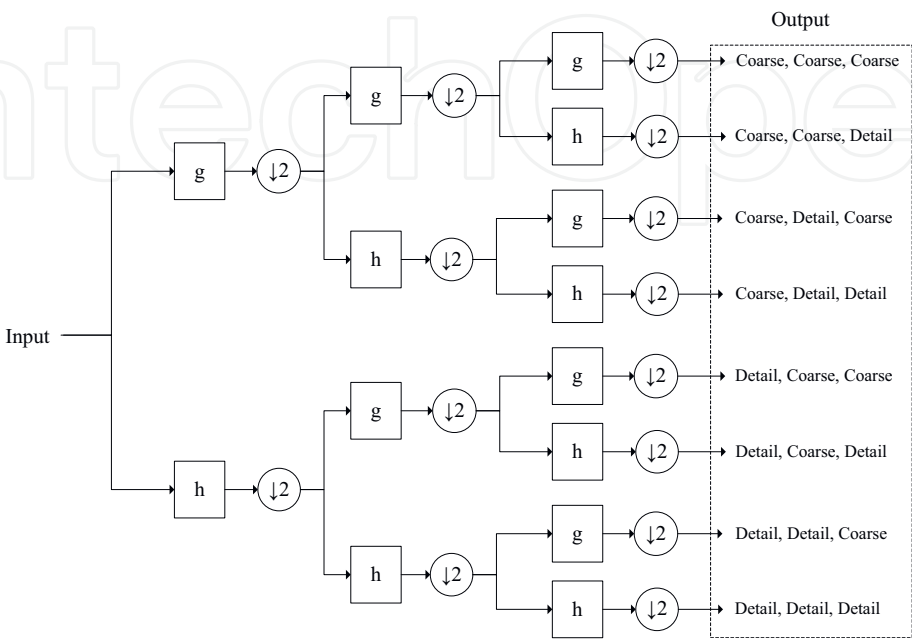


Figure 14.
Diagram of a three-level discrete wavelet packet transform.

filter becomes one child and the output of the coarse filter becomes the other. This process is repeated until the final level is reached, creating a set of output coefficients where each set is identified by the sequence of filters applied to it.

11. Stationary wavelet packet transform

The stationary wavelet packet transform is yet another transform for discrete wavelet functions. Basically, it combines the stationary wavelet transform with the wavelet packet transform, as diagrammed in **Figure 15**. Instead of the decimation operator, the filters themselves are upsampled for each level. The transform creates a binary tree, as with the discrete wavelet packet transform, where both filters for each level are applied at each node. As with the wavelet packet transform, the output from the detail filter becomes one child and the output from the scaling filter becomes the other, and the process is repeated until the final level is reached. Each set of output coefficients are also identified by the sequence of filters applied to it, with the difference that since there is no decimation applied between levels, the number of each set of output coefficients is the same as the input data. This leads to the total number of output coefficients to be 2 times the number of levels multiplied by the length of the input data. Both the discrete wavelet packet transform and the stationary wavelet packet transform have inverse transforms.

The wavelet packet transform introduces many more possibilities for use, some of which are discussed here. Depending on the application, you can do different combinations of the scaling and wavelet filters. Computational complexity depends on the filter combinations selected. If it is taken to the maximum level with the maximum filter combinations, then the discrete wavelet packet transform has a complexity of $O(n \log_2 n)$ and the stationary wavelet packet transform has a complexity of $O(n^2)$.

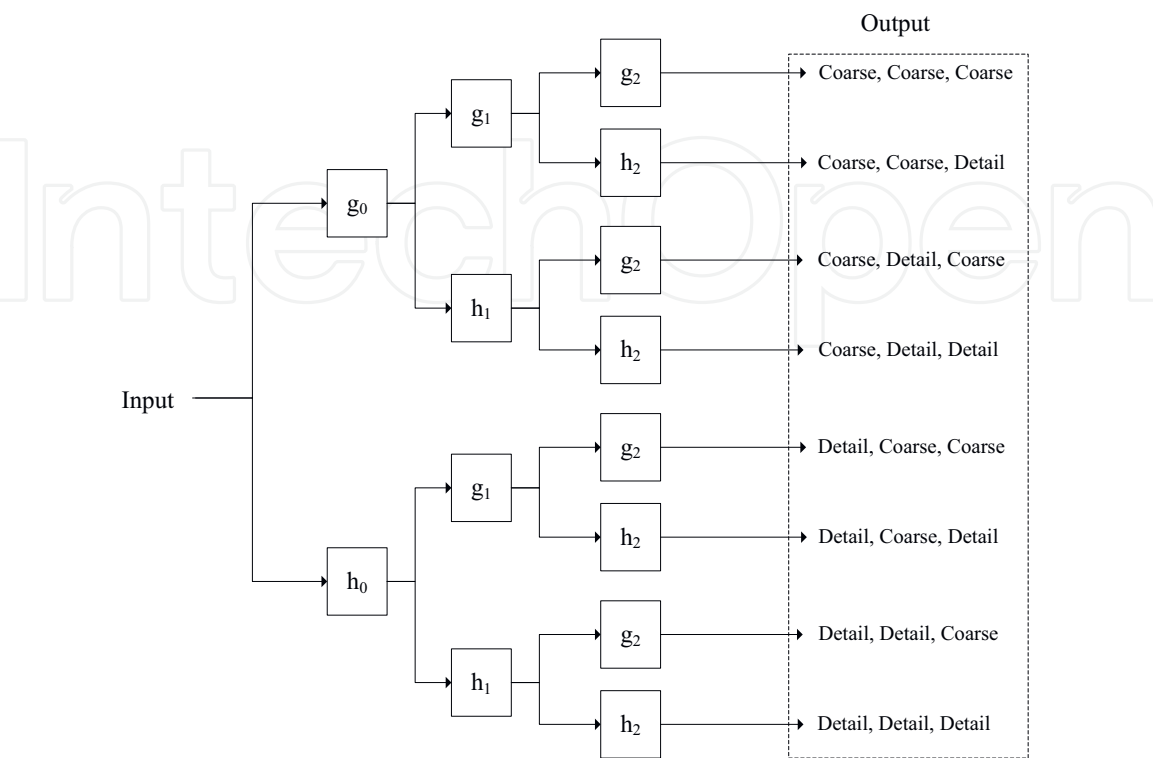


Figure 15.
Diagram of a three-level stationary wavelet packet transform.

12. Conclusions

The Fast Fourier Transform has been listed as one of the top algorithms of the 20th century [27]. Its development has been instrumental to digital signal processing. However, recently a new algorithm, the wavelet transform, has started to have a significant impact on digital signal processing. The wavelet transform improves on the Fourier Transform in that it can analyze a signal by time and frequency simultaneously, thereby easily recovering localized signal information. This is key to many applications, including fractal and multifractal analysis, compression, and filtering.

The wavelet transform introduces many possibilities for use and this chapter has only touched the surface of it. Different wavelets can be used and the transform itself can be customized to fit the application as shown with the wavelet packet transform. Future research will be to determine the proper combination of features for various applications. In addition, there are other possibilities, such as the lifting wavelet transform, which wasn't covered in this chapter. Only orthogonal wavelets that use the same set of wavelets for the forward and inverse transform were covered in this chapter. Biorthogonal wavelets that use different wavelets for the forward and inverse transforms are also available.

The key to wavelet compression and filtering is the sparse signal representation generated by the wavelet transform. The wavelet transform can reduce a signal to minimal set of coefficients. Coefficients that are near zero can be rounded to zero, reducing the size of the signal. In addition, fractional parts of the coefficients can be rounded, also reducing signal size. One of the first uses of this was to compress fingerprints for the FBI [28]. As stated in [29], in the 1990's the FBI had 25 million cards, each containing 10 fingerprints. Digitized, each card contained 10 megabytes of information, for a total of 250 terabytes. Using the two-dimensional discrete wavelet packet transform gives a compression ratio of 20 to 1, enabling the archive to be stored on approximately 12.5 terabytes, while still being able to search and match unknown fingerprints against the ones in the archive. The recently developed JPEG format at the time was based on using the discrete cosine transform on blocks of the image, which left unacceptable artifacts in the image.

JPEG 2000 was developed using the two-dimensional wavelet transform to be the successor to JPEG, although it hasn't caught on. JPEG 2000 allows both lossy and lossless compression. It also doesn't have the lossy artifact generation that the JPEG format has as mentioned previously. Both lossy and lossless compression use the discrete wavelet transform, the difference is that the lossless one uses a wavelet transform that is reversible, while the lossy one uses a wavelet transform that introduces quantization noise that making it irreversible.

Compressed sensing deals with the fact that we that we can obtain a vast amount of information and a lot of it can be discarded and still retain what is relevant. As stated in [24], "singularities and irregular structures often carry the most important information in signals." This is due to the fact that they represent changes to one or more of the properties of the signal. An example of this would be the edges in an image. Compressed sensing removes the redundant, unnecessary information from a signal and analyzes the remaining part of the signal. This is an ideal application for the wavelet transform.

The discrete wavelet transform has been used for Iris recognition for biometric identification in patent US 2002O150281A1 [30]. After taking a picture of the eye, the iris is extracted from the image and then converted to polar coordinates. Using the discrete wavelet transform, the high frequency components are extracted, which are the detail coefficients as referenced in this paper. These form the characteristic vector that is used to identify an iris from the previously recorded data.

The wavelet transform can provide an efficient way to filter white noise from a signal. The procedure consists of applying one of the discrete wavelet transforms to the data and then executing a threshold algorithm that modifies the detail coefficients. After the coefficients are modified, then the inverse transform is applied; the resulting output is a representation of the signal with the noise component significantly reduced.

There are numerous packages available for experimenting with the wavelet transform. The discrete and stationary wavelet transforms are available in Mathematica, Maple, Matlab, R, and PyWavelets to name a few, with the wavelet packet transform available in Mathematica, Matlab, and PyWavelets.

The wavelet transform provides many possibilities for signal analysis depending on the application. A few potential applications were touched on here. The reader is encouraged to develop their own uses and applications for the wavelet transform.

Notes

A lot of this has been previously published under my Master's thesis [31].


Author details

Karlton Wirsing

Virginia Polytechnic and State University, Manassas, Virginia, United States

*Address all correspondence to: kwirsing@vt.edu

IntechOpen

© 2020 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Cooley JW, Tukey JW. An Algorithm for the Machine Calculation of Complex Fourier Series. *Mathematics of Computation*. 1965;**19**(90):297-301
- [2] Gabor D. Theory of communication. Part 3: Frequency compression and expansion. *Electrical Engineers - Part III: Radio and Communication Engineering, Journal of the Institution of*. 1946;**93**(26): 445-457
- [3] Ricker NH. The form and nature of seismic waves and the structure of seismograms. *Geophysics*. October 1, 1940;**5**, 1940(4):348-366
- [4] Haar A. Zur Theorie der orthogonalen Funktionensysteme. *Mathematische Annalen*. 1910;**69**(3): 331-371
- [5] Briggs WL, Henson VE. *The DFT : an owner's manual for the discrete Fourier transform*. Philadelphia: Society for Industrial and Applied Mathematics; 1995 xv, 434 p
- [6] J. B. J. Fourier, *Théorie analytique de la chaleur*. 1822.
- [7] C. F. Gauss, "Nachlass, Theoria Interpolationis Methodo Nova Tractata," in *Carl Friedrich Gauss Werke, Band 3, Königlichen Gesellschaft der Wissenschaften* Göttingen, 1866, pp. 265-330.
- [8] Heideman MT, Johnson DH, Burrus CS. Gauss and the History of the Fast Fourier Transform. *Archive for History of Exact Sciences*. 1985;**34**(3): 265-277
- [9] Hubbard BB. *The world according to wavelets : the story of a mathematical technique in the making*. Wellesley, Mass: A.K. Peters; 1998
- [10] D. L. Fugal, *Conceptual wavelets in digital signal processing : an in-depth, practical approach for the non-mathematician*, 1st ed. San Diego, Calif.: Space & Signals Technical Pub., 2009, pp. xxii, 302, 50 p.
- [11] L. v. Beethoven, "Symphonie No. 5, Op. 67," E. Pauer, Ed., ed: B. Schott's Söhne, 1808.
- [12] R. F. Voss and J. Clarke, "'1/f noise' in music: Music from 1/f noise," *The Journal of the Acoustical Society of America*, vol. 63, no. 1, pp. 258-263, 1978.
- [13] B. H. Suits. (1998, September 2, 2013). Physics of Music - Notes. Available: <http://www.phy.mtu.edu/~suits/notefreqs.html>
- [14] J. Y. Stein, Digital signal processing : a computer science perspective (Wiley series in telecommunications and signal processing). New York: Wiley, 2000, pp. xx, 859 p.
- [15] N. Ricker, "Further developments in the wavelet theory of seismogram structure," *Bulletin of the Seismological Society of America*, vol. 33, no. 3, pp. 197-228, July 1, 1943 1943.
- [16] Ricker N. Wavelet functions and their polynomials. *Geophysics*. July 1, 1944;**9**, 1944(3):314-323
- [17] Ricker N. The form and laws of propagation of seismic wavelets. *Geophysics*. January 1, 1953;**18**, 1953(1): 10-40
- [18] Hosken J. Ricker wavelets in their various guises. *First Break*. January 1988; **6**:24-33
- [19] Graps A. An introduction to wavelets. *Computational Science & Engineering, IEEE*. 1995;**2**(2):50-61
- [20] C. Valens, "A Really Friendly Guide to Wavelets," Available: <http://www.rob>

ots.ox.ac.uk/~parg/mlrg/papers/arfgtw.pdf

transform” USA Patent US 2002/0150281 A1, 2001.

[21] Grossmann A, Morlet J. Decomposition of Hardy Functions into Square Integrable Wavelets of Constant Shape. *SIAM Journal on Mathematical Analysis*. 1984;15(4):723-736

[31] K. Wirsing, “Application of Wavelets to Filtering and Analysis of Self-Similar Signals,” Master of Science, Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, 2014.

[22] Y. Yu, J. Zhou, Y. Wang, F. Li, and C. Ge, “On the computation of wavelet series transform,” in *Signal Processing Proceedings, 1998. ICSP '98. 1998 Fourth International Conference on*, 1998, pp. 313-316 vol.1.

[23] C. Torrence and G. P. Compo, “A Practical Guide to Wavelet Analysis,” *Bulletin of the American Meteorological Society*, vol. 79, no. 1, pp. 61-78, 1998/01/01 1998.

[24] Mallat S, Hwang WL. Singularity detection and processing with wavelets. *IEEE Trans. Inf. Theor.* 1992;38(2): 617-643

[25] A. Boggess and F. J. Narcowich, *A first course in wavelets with Fourier analysis*, 2nd ed. Hoboken, N.J.: John Wiley & Sons, 2009, pp. xv, 315 p.

[26] Strang G, Nguyen T. *Wavelets and filter banks*. Wellesley. MA: Wellesley-Cambridge Press; 1996. p. xxi 490 p

[27] Rockmore DN. The FFT: an algorithm the whole family can use. *Computing in Science & Engineering*. 2000;2(1):60-64

[28] Brislawn CM. The FBI Fingerprint Image Compression Specification. In: Topiwala PN, editor. *Wavelet Image and Video Compression*. Boston, MA: Springer US; 1998. pp. 271-288

[29] “Wavelets, Fingerprints, and Image Processing,” in *Discovering Wavelets*, 1999, pp. 1-21.

[30] S.-W. Cho, “Method of recognizing human iris using daubechies wavelet