CURVE EDITING THROUGH WAVELETS

By

ADIL RAUF

2001

HAMDARD UNIVERSITY

جامعه همدرد

Hamdard Institute of Information Technology
Hamdard University

CURVE EDITING THROUGH WAVELETS

By

ADIL RAUF

2001

Hamdard Institute of Information Technology
Hamdard University

HAMDARD UNIVERSITY OF INFORMATION TECHNOLOGY

HAMDARD UNIVERSITY

PROJECT REPORT APPROVAL

Date: 30^{th}, May 2001

I hereby recommend that the project report prepared and successfully defended under my supervision entitled:

CURVE EDITING THROUGH WAVELETS

by Adil Rauf

be accepted towards partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

from Hamdard Institute of Information Technology, Hamdard University.

Qurban Ali

Dr. Qurban Ali
Professor

Director
HIIT

Shahab S.

Mr. Shahab Siddiqui
Assistant Professor

Qurban Ali

Deputy Director
HIIT

K. Abbasi

Mr. Kashif Abbasi
Assistant Professor

ABSTRACT

This report discusses techniques related to curve editing through Wavelets. Actually the latest trend is, that not to confine wavelets to signal processing but to apply it in areas other than signal processing. This report is an attempt to show application of wavelets in computer graphics area.

The usual curve editing operations are: smoothing a curve, editing the overall form of a curve while preserving its details and changing its fine details while maintaining its overall sweep. A curve is edited on different scales of resolutions using reconstruction and decomposition filters. These filters make possible to construct a multiresolution-based system. Different algorithms and mathematical relations are simulated in MATLAB, their end results are presented and are found in conformity with the expected results.

CURVE EDITING THROUGH WAVELETS

by

ADIL RAUF

A report submitted in partial fulfillment of the requirements for the degree of
Master of Science in Information Technology
at
Hamdard Institute of Information Technology
Hamdard University

Project Supervisor:
Dr. Qurban Ali

ACKNOWLEDGEMENT

I have no words to thank my respected Project Supervisor Dr. Qurban Ali, who gave me the motivation to work and the thrust to conquer all the hurdles of the project, which came to me frequently. He not only guided me technically but also boosted my research spirits high enough to tackle all tasks successfully during the work.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 CLASSICAL WAVELET THEORY

DILATION EQUATION

Before rushing into the theory and applications of wavelet analysis, a brief exposure on some of the basic concepts that make wavelet analysis such a useful tool today, is presented. Basically its initial applications were found in signal processing area, but by the passage of time its applications became wider and wider in other areas as well, such as; image processing, computer graphics, mechanical vibrations, etc. Since people are usually familiar with signals and they easily understand things related to signals, therefore let's make some foundations about wavelets related to signal processing perspective.

Mostly people ask why use wavelets? Why not fourier transform? Well, Wavelet analysis is similar to Fourier analysis in the sense that it breaks a signal down into its constituent parts for analysis. Whereas the Fourier transform breaks the signal into a series of sine waves of different frequencies, the wavelet transform breaks the signal into its "wavelets", scaled and shifted versions of the "mother wavelet". There are however some very distinct differences which compares a sine wave to wavelets. When analysing signals of a non-stationary nature, it is often beneficial to be able to acquire a correlation between the time and frequency domains of a signal. The Fourier transform, provides information about the frequency domain, however time localised information is essentially lost in the process. The problem with this is the inability to associate features in the frequency in domain with their location in time, as an alteration in the frequency spectrum will result in changes throughout the time domain. In contrast to the fourier transform, the wavelet transform allows exceptional localisation in both the time domain via translations of the mother wavelet, and in the scale (frequency) domain via dilations. The translation and dilation operations applied to the mother wavelet are performed to calculate the wavelet coefficients, which represent the correlation between the wavelet and a localised section of the signal. The wavelet coefficients are calculated for each wavelet segment, giving a time-scale function relating the wavelets correlation to the signal.

The dilation function of the wavelet transform can be represented as a tree of low and high pass filters, with each step transforming the low pass filter. The original signal is successively decomposed into components of lower resolution, while the high frequency components are not analysed any further. The maximum number of dilations that can be performed is dependent on the input size of the data to be analysed. The versatility and power of the wavelet transform can be significantly increased by using its generalised form, Wavelet Packet Analysis. Unlike the wavelet transform which only decomposes the low frequency components, wavelet packet utilises both the low frequency components and the high frequency components, in this project we shall use the high frequency components for editing the curve's overall sweep and the low frequency components for editing a curve's characters in sections 2.2.1 and 2.2.2.

Now let's come to the mathematics of wavelets. The dilation equations have been studied widely only recently. To dilate is to spread out, so that dilation means expansion. The basis function $\phi(x)$ is a dilated (horizontally) version of $\phi(2x)$. It has the same height, but is stretched out over twice the horizontal scale of x, where x is a non-dimensional independent variable that may represent time or length, depending on the application. In a dilation equation $\phi(x)$ is expressed as a finite series of terms, each of which involves (for example) $\phi(2x)$. Each of these $\phi(2x)$ terms is positioned at a different place on the horizontal axis by making the argument (2x-k) instead of just 2x, where k is an integer (positive or negative). The basic dilation equation has a form [1]:

$$\phi(x) = c_0\phi(2x)+ c_1\phi(2x-1)+ c_2\phi(2x-2)+ c_3\phi(2x-3) \quad (1)$$

where the c's are numerical constants (generally some positive and some negative). Except for a very few simple cases, it is not possible to solve (1) directly to find out what is the function $\phi(x)$. Instead iterative algorithm in which each new approximation $\phi_j(2x)$ is calculated from the previous approximation $\phi_{j-1}(x)$ by the scheme [1]:

$$\phi_j(x) = c_0\phi_{j-1}(2x)+ c_1\phi_{j-1}(2x-1)+ c_2\phi_{j-1}(2x-2)+ c_3\phi_{j-1}(2x-3) \qquad (2)$$

The iteration is continued until $\phi_j(x)$ becomes indistinguishable from $\phi_{j-1}(x)$.

Consider starting from a box function $\phi_0(x) = 1$, $0 <= x < 1$, $\phi_0(x) = 0$ elsewhere. After one iteration the box function over interval x = 0 to 1 has developed into a staircase function over the interval x = 0 to 2 as shown in figure 1. Each contribution to $\phi_1(x)$ is shown separately and coefficients c_0, c_1, c_2, c_3, has been used, defined as follows:

$$\left.\begin{array}{l} c_0 = (1+1.7432)/4,\ c_1 = (3+1.732)/4 \\ c_2 = (3-1.7432)/4,\ c_3 = -(3+1.732)/4. \end{array}\right\} \qquad (3)$$

These we shall see later generate an orthogonal D4 wavelet. The "D" stands for Daubechies who first discovered their properties.

When the iterative process is continued, the function $\phi(x)$ approaches a limiting shape as shown in figure 2. An unusual feature is the discontinuous nature of the shape. The graph has a fractal nature so that when drawn to larger scale its irregular outline remains. In order to obtain a smoother function, it is necessary to include more terms in the dilation equation (1).

The function $\phi(x)$ generated from a unit box (unit height and length) is called the scaling function and a corresponding wavelet function will be constructed from it in the next section. However, first we consider in more detail how the iterative

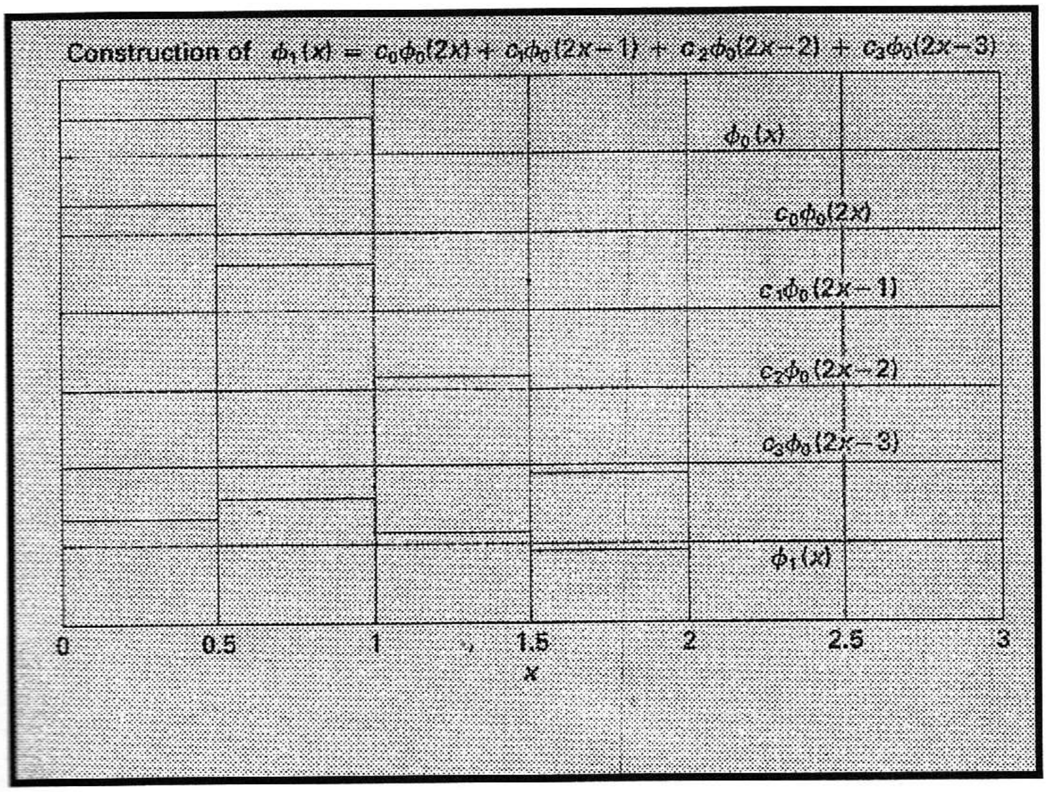
Construction of $\phi_1(x) = c_0\phi_0(2x) + c_1\phi_0(2x-1) + c_2\phi_0(2x-2) + c_3\phi_0(2x-3)$
$\phi_0(x)$
$c_0\phi_0(2x)$
$c_1\phi_0(2x-1)$
$c_2\phi_0(2x-2)$
$c_3\phi_0(2x-3)$
$\phi_1(x)$
0
0.5
1
1.5
2
2.5
3
x

Figure 1.

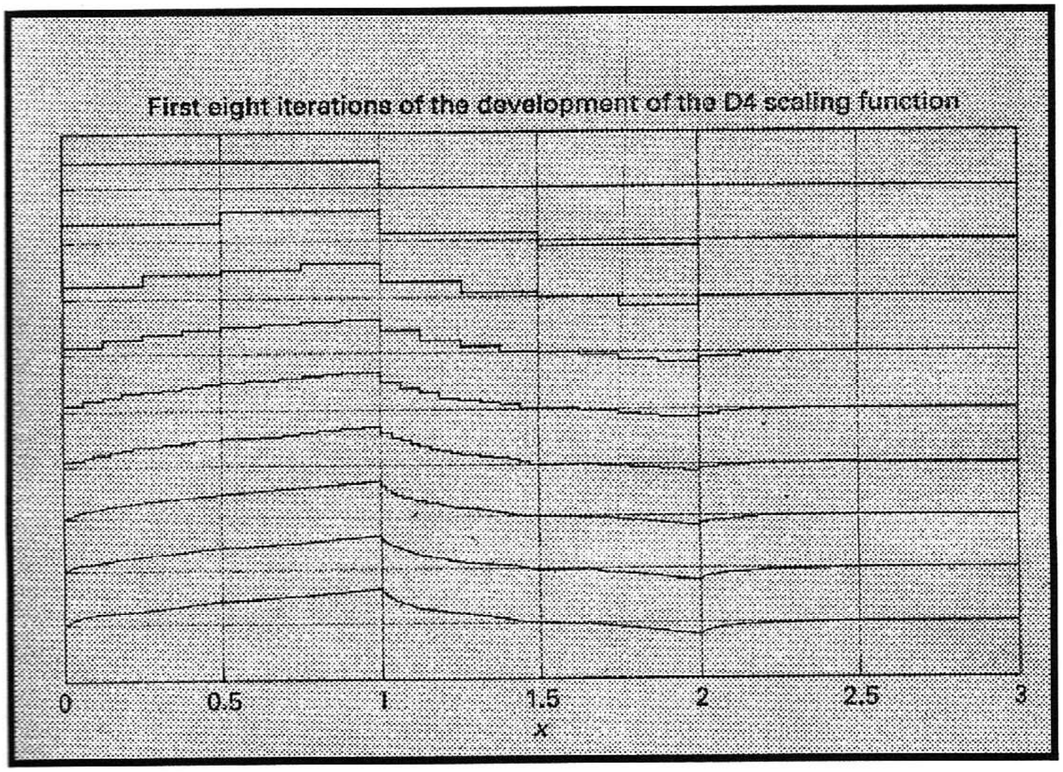
First eight iterations of the development of the D4 scaling function
0
0.5
1
1.5
2
2.5
3
x

Figure 2.

scheme used in figure 1 and 2 works. Representing the initial box function by the ordinate 1 at x = 0, the first iteration (figure 1) produces four new ordinates c_0, c_1, c_2, c_3 at x = 0, 0.5, 1, 1.5. At the second iteration, ordinate c_0 at x = 0 contributes to four new ordinates c_0^2, c_0c_1, c_0c_2, c_0c_3 at x = 0, 0.25, 0.5, 0.75; ordinate c_1 at x = 0.5 contributes to four new ordinates c_1c_0, c_1^2, c_1c_2, c_1c_3 at x = 0.5, 0.75, 1, 1.25; and so on.

After the second iteration is complete, the resulting ordinates (third row of figure 2) are:

c_0^2, c_0c_1, $c_0c_2+c_1c_0$, $c_0c_3+c_1^2$, $c_1c_2+c_2c_0$, $c_1c_3+c_2c_1$, $c_2^2+c_3c_0$, $c_2c_3+c_3c_1$, c_3c_2, c_3^2 at x = 0, 0.25, 0.5, 0.75, …, 2.25.

This calculation follows the matrix scheme [1]:

$$[\phi_2] = [c_0\ \ 0\ \ 0\ \ 0;\ c_1\ \ 0\ \ 0\ \ 0;\ c_2\ \ c_0\ \ 0\ \ 0;\ c_3\ \ c_1\ \ 0\ \ 0;\ 0\ \ c_2\ \ c_0\ \ 0;\ 0\ \ c_3\ \ c_1\ \ 0;\ 0\ \ 0\ \ c_2\ \ c_0;\ 0\ \ 0\ \ c_3\ \ c_1;\ 0\ \ 0\ \ 0\ \ c_2;\ 0\ \ 0\ \ 0\ \ c_3]\ .$$
$$[c_0;\ c_1;\ c_2;\ c_3]\ [1] = M_2\ M_1\ [1] \tag{4}$$

where,

$$M_2 = [c_0\ \ 0\ \ 0\ \ 0;\ c_1\ \ 0\ \ 0\ \ 0;\ c_2\ \ c_0\ \ 0\ \ 0;\ c_3\ \ c_1\ \ 0\ \ 0;\ 0\ \ c_2\ \ c_0\ \ 0;\ 0\ \ c_3\ \ c_1\ \ 0;\ 0\ \ 0\ \ c_2\ \ c_0;\ 0\ \ 0\ \ c_3\ \ c_1;\ 0\ \ 0\ \ 0\ \ c_2;\ 0\ \ 0\ \ 0\ \ c_3]$$
$$M_1 = [c_0;\ c_1;\ c_2;\ c_3]$$

where M_r denotes a matrix of order $(2^{r+1}+2^r-2)x(2^r+2^{r-1}-2)$ in which each column has a submatrix of the coefficients c_0, c_1, c_2, c_3 positioned two places below the submatrix to its left. The number of points on the graph increases in the sequence 1, 4, 10, 22, 46, …,$2^{r+1}+2^r-2$ so that after eight iterations it reaches 2^9+2^8-2 = 766 with each point spaced $1/2^8 = 1/256$ units apart along the horizontal axis. The graph begins at $x = 0$ and almost (but not quite) reaches $x = 3$.

1.2 DILATION OF WAVELETS

So far we have not defined a wavelet. This is described by its wavelet function W(x), which is derived from the corresponding scaling function by taking differences. For the four-coefficient scaling function defined by (1), the dilation wavelet function is:

$$W(x) = -c_3\phi(2x)+ c_2\phi(2x-1)- c_1\phi(2x-2)+ c_0\phi(2x-3) \qquad (5)$$

The same coefficients are used as for the definition of $\phi(x)$, but in reverse order and with alternate terms having their signs changed from plus to minus (the numerical values of the c's may of course be positive or negative in any particular case).

The results of making the calculation (5) for D4 scaling function in figure 3 is shown in figure 4. This is the D4 wavelet, and was used to analyse the square wave in figure 1 and 2. It retains the discontinuous, fractal nature of the scaling function and is certainly a rather surprising shape for a basis function for signal analysis.

The definition 5 may be varied and an alternative definition is [1]:

$$W(x) = c_3\phi(2x+2)-c_2\phi(2x+1)+c_1\phi(2x)-c_0\phi(2x-1) \qquad (6)$$

Let's proceed further taking the definition 5 as our basis because this fits in more conveniently with the construction of the numerical algorithms.

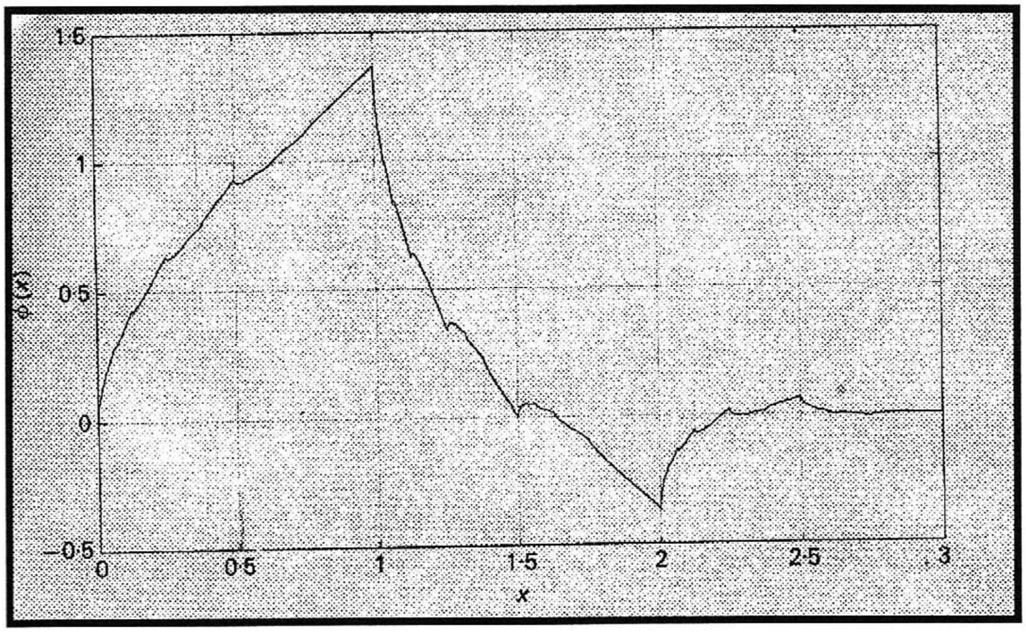

Figure 3. Scaling function for the D4 wavelet calculated for 3 x 2^{12} = 12288 points.

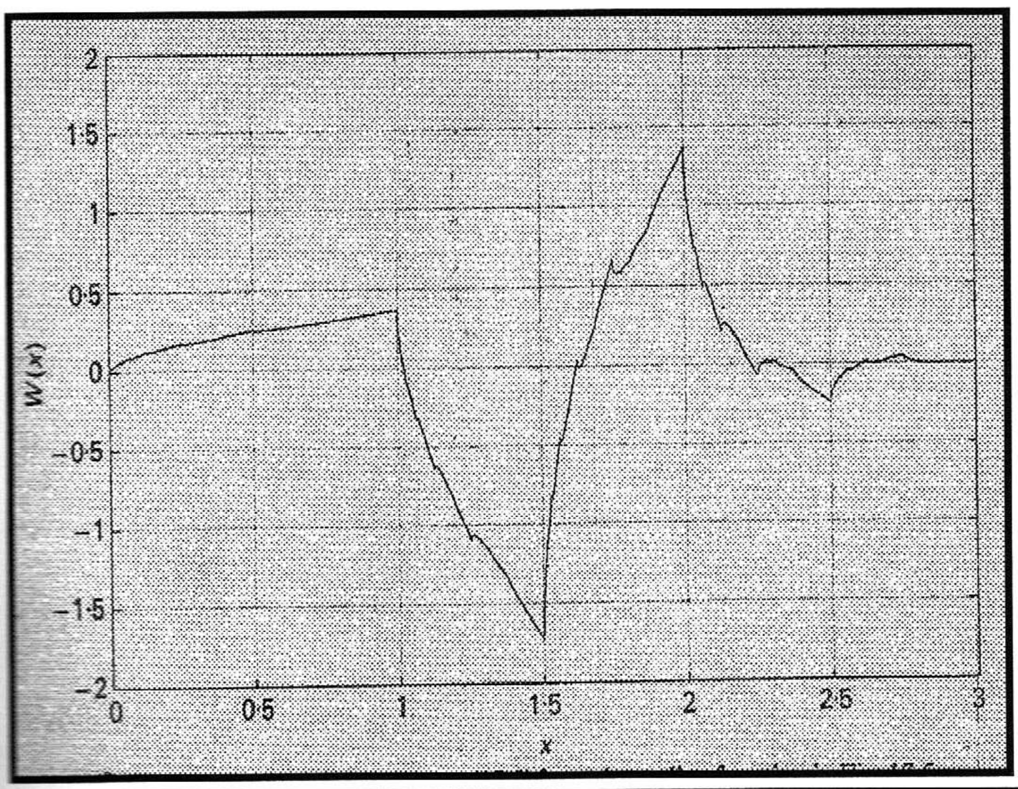

Figure 4. D4 wavelet according to 5 from the scaling function in figure 3.

Consider again the matrix scheme for developing $\phi(x)$ by iteration starting from a box function of unit height over the interval x = 0 to 1. Suppose that we have taken the iteration to a stage where $\phi(x)$ is fully developed and that we want to generate W(x) from this $\phi(x)$ by using 5. For simplicity, imagine that only one iteration leads to the final $\phi(x)$ so that this is represented by just four ordinates c_0, c_1, c_2, c_3 at x = 0, 0.5, 1, 1.5 as in the upper view of figure 4. According to 5, these four ordinates generate ten new ordinates spaced 0.25 apart. The term $-c_3\phi(2x)$ in 5 gives $-c_3c_0$, $-c_3c_1$, $-c_3c_2$, $-c_3^2$; the term $c_2\phi(2x-1)$ gives c_2c_0, c_2c_1, c_2^2, c_2c_3 moved two places to the right; and so on for the other terms, so that the new ordinates (now for the wavelet) are

$$-c_3c_0,\ -c_3c_1,\ -c_3c_2+c_2c_0,\ -c_3^2+c_2c_1,\ c_2^2-c_1c_0,\ c_2c_3-c_1^2,\ -c_1c_2+c_0^2, -c_1c_3+c_0c_1,\ c_0c_2,\ c_0c_3.$$

These are generated by the matrix scheme

$$[W_2] = [-c_3\ 0\ 0\ 0;\ 0\ -c_3\ 0\ 0;\ c_2\ 0\ -c_3\ 0;\ 0\ c_2\ 0\ -c_3;\ -c_1\ 0\ c_2\ 0;$$
$$0\ -c_1\ 0\ c_2;\ c_0\ 0\ -c_1\ 0;\ 0\ c_0\ 0\ -c_1;\ 0\ 0\ c_0\ 0;\ 0\ 0\ 0\ c_0].$$
$$[c_0;\ c_1;\ c_2;\ c_3][1] \qquad (7)$$

or alternatively, by

$$[W_2] = [c_0\ 0\ 0\ 0;\ c_1\ 0\ 0\ 0;\ c_2\ c_0\ 0\ 0;\ c_3\ c_1\ 0\ 0;\ 0\ c_2\ c_0\ 0;$$
$$0\ c_3\ c_1\ 0;\ 0\ 0\ c_2\ c_0;\ 0\ 0\ c_3\ c_1;\ 0\ 0\ 0\ c_2;\ 0\ 0\ 0\ c_3].$$
$$[-c_3;\ c_2;\ -c_1;\ c_0]\ [1]. \qquad (8).$$

Notice the zeros between the diagonal lines of coefficients in 7. If the wavelet were calculated after first completing two iterations of the scaling function, there would be three zeros between each diagonal line. This is because the ordinates after two iterations of the scaling function are spaced x = 0.25 apart. As a result of the 2x in the terms on the right-hand side of 5, the ordinates of the wavelet function will be $0.25/2 = 0.125$ apart. However, the translations required by 5 are $x = 0.5(\phi(2x-1))$ is translated 0.5 with respect to $\phi(2x)$, and so on).

Similarly for W_3, we have

$$W_3 = M_3\, M_2\, [-c_3;\ c_2;\ -c_1;\ c_0] \times [1]. \tag{10}$$

Where M_3 is a matrix of order 22x10 with 10 submatrices $[c_0\ \ c_1\ \ c_2\ \ c_3]^t$ each arranged two places below its left-hand neighbour.

1.3 THEORY OF WAVELETS WITH COMPUTER GRAPHICS PERSPECTIVE

Suppose we are given a one-dimensional "image" with a resolution of four pixels, having values

$$[\ 9\ \ 7\ \ 3\ \ 5\ ]$$

We can represent this image in a wavelet transform by first taking the average of the pixels together, pairwise, to get the new lower resolution image with pixel values

$$[\ 8\ \ 4\ ]$$

Clearly, some information has been lost in this averaging process. To recover the original four-pixel values from the two averaged values, we need to store some detail coefficients, which capture the missing information. We will choose 1 for the first detail coefficient, since the average we computed is 1 less than 9 and 1 more than 7.

This single number allows us to recover the first two pixels of our original four-pixel image. Similarly, the second detail coefficient is –1, since 4+(-1) = 3 and 4-(-1) = 5.

Thus, we have decomposed the original image into a lower resolution (two-pixel) version and a pair of detail coefficients. Repeating this process recursively on the averages gives the full decomposition:

Level	Resolution	Averages	Detail Coefficients
2	4	[9 7 3 5]	
1	2	[8 4]	[1 -1]
0	1	[6]	[2]

Finally, in the end we define the wavelet transform (also called wavelet decomposition) of the original four-pixel image to be the single coefficient representing the overall average of the original image, followed by the detail coefficients in order of increasing resolution. Thus the wavelet transform of the original four-pixel is given by

$$[\ 6\ \ 2\ \ 1\ \ -1\ ]$$

1.4 THEORY OF MULTIRESOLUTION CURVES

Consider coordinates of points C^n, expressed as a column vector of samples $[c_1^n, \ldots, c_m^n)]^T$. Conventionally the samples c_j^n are called the curve's control points in two-dimensions.

In order to create a low-resolution version C^{n-1} of C^n with a fewer number of samples the standard approach is to use some form of filtering and down sampling on the samples of C^n. This process can be expressed as a matrix equation

$$C^{n-1} = A^n C^n$$

Since C^{n-1} contains fewer samples than C^n, it is clear that some amount of detail is lost in this filtering process. If A^n is appropriately chosen, it is possible to capture the lost detail as another D^{n-1}, computed by [2]:

$$D^{n-1} = B^n C^n$$

The pair of matrices A^n and B^n are called decomposition filters. The process of splitting a signal C^n into a low-resolution version C^{n-1} and detail D^{n-1} is called decomposition.

If A^n and B^n are chosen correctly, then the original signal C^n can be recovered from C^{n-1} and D^{n-1} by using another pair of matrices P^n and Q^n, called reconstruction filter, as shown [2]:

$$C^n = P^n C^{n-1} + Q^n D^{n-1}$$

Recovering C^n from D^{n-1} is called reconstruction.

1.5 FILTER BANKS

The procedure for splitting C^n into a low-resolution part C^{n-1} and a detail part D^{n-1} can be applied recursively to the new signal C^{n-1}. Thus, the original coordinates can be expressed as a hierarchy of lower-resolution coordinates $C^0,\dots,C^{n-1}$ and details $D^0,\dots,D^{n-1}$, as shown below. This recursive process is known as filter bank.

Since the original coordinates C^n can be recovered from the sequence C^0, D^0, $D^1, \ldots, D^{n-1}$, this sequence can be thought of as a transform of the original signal, known as wavelet transform. The recursive process can be represented as shown below:

$$
\begin{tikzcd}
C^n \arrow[r, "A^n"] \arrow[dr, "B^n"'] & C^{n-1} \arrow[r, "A^{n-1}"] \arrow[dr, "B^{n-1}"'] & C^{n-2} \cdots \arrow[r, "A^1"] \arrow[dr, "B^1"'] & C^0 \\
& D^{n-1} & D^{n-2} & D^0
\end{tikzcd}
$$

1.6 ADVANTAGES OF WAVELET TRANFORM

Storing the coordinate's wavelet transform, rather than the coordinates itself, has a number of advantages.

One advantage of the wavelet transform is that often a large number of the detail coefficients turn out to be very small in magnitude. So, truncating, or removing these small coefficients from the representation introduces only small errors in the reconstructed coordinates. The size of the transform C^0, D^0, $D^1, \ldots, D^{n-1}$ is the same as that of the original coordinates C^n, so no extra storage is required. If the A^j, B^j, P^j, Q^j are constructed to be sparse, then the filter bank operation can be performed very quickly.

1.7 CONSTRUCTION OF "P" AND "Q" FILTERS FOR SPLINE WAVELETS

Basis function for spline wavelet is derived from Cox-deboor recursive formula [10]:

$N_1(x) = 1$ for $0 <= x < 1$, (11)

0 otherwise.

$$N_m(x) = [x/(m-1)].\ N_{m-1}(x)+[(m-x)/(m-1)].\ N_{m-1}(x-1) \quad (12)$$

Where (11) and (12) are basis functions.

In terms of two-scale relation, $\phi(x)$ can be given as [19]:

$$N_m(x) = \phi(x) = \sum_{k=0}^{m} p_k\ N_m(2x-k),\ \text{(scaling function)} \quad (13)$$

where, $p_k = 2^{-m+1} \begin{pmatrix} m \\ k \end{pmatrix}$, (reconstruction "p" coefficient) (14)

where, p_k = reconstruction filter

$m-1$ = degree of $N_m(x)$

k = translation integer

Now, wavelet scaling function in terms of two-scale relation, can be given as [19]:

$$\psi_m(x) = \sum_{k=0}^{3m-2} q_k\ N_m(2x-k)\ ,\ \text{(wavelet scaling function)} \quad (15)$$

$$q_k = (-1)^k\ 2^{1-m} \sum_{r=0}^{m} \begin{pmatrix} m \\ k \end{pmatrix} N_{2m}(k+1-r),\ \text{(reconstruction "q" coefficient)} \quad (16)$$

Since we have used cubic polynomials to represent curves and interpolated points via cubic polynomial so we shall calculate cubic basis function and cubic wavelet functions first, then we shall go for their respective filters. So lets apply the above relations for cubic polynomials. Firstly, we calculate $\phi(x)$ the basis function:

Degree of cubic spline polynomial = $m-1 = 3$, so $m = 4$

From cox-deboor recursive relation (11) and (12), we have:

$$N_4(x) = \phi(x) = \begin{cases} (1/6)\ x^3 & \text{for } 0 <= x < 1 \\ (1/6)\ (-3x^3+12x^2-12x+4) & \text{for } 1 <= x < 2 \end{cases}$$

$(1/2)\ (3x^3-24x^2+60x-44)$ for $2 <= x < 3$

$(1/6)\ (4-x)^3$ for $3 <= x < 4$

0 otherwise.

from (13) we have:

$$N_4(x) = \phi(x) = \sum_{k=0}^{4} p_k\ N_4(2x-k), \text{ (scaling function)}$$

on solving we have:

$$\phi(x) = p_0(2x-0)+p_1(2x-1)+p_2(2x-2)+p_3(2x-3)+p_4(2x-4) \tag{17}$$

from (14) we have:

$$p_0 = 2^{-4+1} \binom{4}{0.} = 1/8$$

Similarly,

$p_1 = 4/8$, $p_2 = 6/8$, $p_3 = 4/8$, $p_4 = 1/8$.

so (17) becomes:

$$\phi(x) = (1/8)(2x-0)+(4/8)(2x-1)+(6/8)(2x-2)+(4/8)(2x-3)+(1/8)(2x-4)$$

Secondly, we calculate $\psi(x)$ the wavelet function:

Degree of cubic spline polynomial = m-1 = 3, so m = 4

From (15) we have:

$$\psi_m(x) = \sum_{k=0}^{3m-2} q_k\ N_m(2x-k)\ , \text{ (wavelet scaling function)}$$

$$\psi_4(x) = \sum_{k=0}^{10} q_k\ N_4(2x-k)$$

On solving we have:

$$\begin{aligned}\psi_4(x) = {} & q_0N_4(2x-0) + q_1N_4(2x-1) + q_2\,N_4(2x-2) + q_3N_4(2x-3) + \\ & q_4N_4(2x-4) + q_5N_4(2x-5) + q_6N_4(2x-6) + q_7N_4(2x-7) + \\ & q_8N_4(2x-8) + q_9N_4(2x-9) + q_{10}N_4(2x-10) \end{aligned} \tag{18}$$

From (16) we have:

$$q_k = (-1)^k\, 2^{1-m} \sum_{r=0}^{m} \binom{m}{r} N_{2m}(k+1-r)$$

$$q_k = (-1)^k\, 2^{-3} \sum_{r=0}^{4} \binom{4}{r} N_8(k+1-r)$$

$$q_0 = (-1)^0 (1/8) [N_8(1)+N_8(0)+N_8(-1) +N_8(-2) +N_8(-3)]$$

$$q_0 = (1/7) [N_8(1)+ 0 + 0 + 0 + 0]$$

Calculating recursively, [10]:

$$N_8(x) = (x/7) N_7(x)+[(8-x)/7] N_7(x-1)$$

$$N_8(1) = (1/7) N_7(1)+N_7(0)$$

$$N_7(1) = (1/6) N_6(1)+N_6(0) = (1/6)(1/120) = 1/6!$$

Therefore,

$$N_8(1) = (1/7)(1/6!)$$

So,

$$q_0 = (1/8) (1/7!) = 1/8!$$

Similarly,

$q_1 = -124/8!$, $q_2 = 1677/8!$, $q_3 = -7904/8!$,

$q_4 = 18482/8!$, $q_5 = -24264/8!$, $q_6 = 18482/8!$,

$q_7 = -7904/8!$, $q_8 = 1677/8!$, $q_9 = -124/8!$,

$q_{10} = 1/8!$.

So, (18) becomes:

$$\begin{aligned}\psi_4(x) = {} & (1/8!)N_4(2x-0) + (-124/8!) N_4(2x-1) + (1677/8!) N_4(2x-2) + \\ & (-7904/8!) N_4(2x-3) + (18482/8!) N_4(2x-4) + (-24264/8!) N_4(2x-5) + \\ & (18482/8!) N_4(2x-6) + (-7904/8!) N_4(2x-7)+ (1677/8!) N_4(2x-8) + \\ & (-124/8!) N_4(2x-9) + (1/8!) N_4(2x-10)\end{aligned}$$

1.8 CONSTRUCTION OF "P" AND "Q" FILTERS FOR DAUBECHIES WAVELETS

Daubechies wavelet coefficients q_k are in even number, such as: D2,D4,D6, etc.

Consider the p_k properties [1]:

Scaling function: $\phi(x) = \sum_{k=0}^{N-1} p_k \phi(2x-k)$ (19)

Wavelets function: $\psi(x) = \sum_{k=0}^{N-1} (-1)^k p_k \phi(2x+k-N+1)$ (20)

Where, N = Even no. of wavelets coefficients

k = 0 to N-1, shifting of the $\phi(x)$ and $\psi(x)$ functions

The wavelet coefficients must satisfy the following conditions [1]:

$\sum_{k=0}^{N-1} p_k = 2$ (21)

$\sum_{k=0}^{N-1} (-1)^k k^m p_k = 0$ (22)

for m = 0, 1, 2, …, N/2-1.

$\sum_{k=0}^{N-1} p_k p_{k+2m} = 0$ (23)

for m = 1, 2, …., N/2-1.

For D2, we have: N = 2, k = 0, 1

from (21) we have:

$p_0 + p_1 = 2$

from (22) we have:

$p_0 - p_1 = 0$

from (33) we have:

$p_0^2 + p_1^2 = 2$

so,

$p_0 = p_1 = 1$.

For D4, we have: N = 4, k = 0, 1, 2, 3

from (21) we have:

$p_0 + p_1 + p_2 + p_3 = 2$

from (22) we have:

$p_0 - p_1 + p_2 - p_3 = 0$

from (23) we have:

$p_0^2 + p_1^2 + p_2^2 + p_3^2 = 2$

so,

$p_0 = (1+1.732)/4 \qquad p_1 = (3+1.732)/4$

$p_2 = (3-1.732)/4 \qquad p_4 = (1-1.732)/4.$

For D2, from (19) we have:

$\phi(x) = p_0\phi(2x-0) + p_1\phi(2x-1)$

$\phi(x) = 1\phi(2x-0) + 1\phi(2x-1)$

For D4, from(19) we have:

$\phi(x) = p_0\phi(2x-0) + p_1\phi(2x-1) + p_2\phi(2x-2) + p_3\phi(2x-3)$

$\phi(x) = ((1+1.732)/4)\phi(2x-0) + ((3+1.732)/4)\phi(2x-1) +$
$((3-1.732)/4)\phi(2x-2) + ((1-1.732)/4)\phi(2x-3)$

Now calculate q_k for D2 we have [19]:

$q_k = (-1)^k p_{-k+1}$, i.e.,

$q_1 = (-1)^1 p_{-1+1}$
$= (-1)^1 p_0 = - p_0 = - 1$

$q_0 = (-1)^0 p_{-0+1}$
$= p_1 = 1$

So from (20) we have:

$\psi(x) = q_1\phi(2x-0) - q_0\phi(2x-1)$

$\psi(x) = 1\phi(2x-0) - 1\phi(2x-1)$

Now calculate q_k for D4 we have [19]:

$q_k = (-1)^k p_{-k+1}$, i.e.,

$q_1 = (-1)^1 \, p_{-1+1}$

$= (-1)^1 \, p_0 = - \, p_0 = - \, (1+1.732)/4$

$q_0 = (-1)^0 \, p_{-0+1}$

$= \, p_1 = \, (3+1.732)/4$

$q_{-1} = (-1)^{-1} \, p_{-(-1)+1}$

$= - \, p_2 = - \, p_0 = - \, (3-1.732)/4$

$q_{-2} = (-1)^{-2} \, p_{-(-2)+1}$

$= p_3 = (1-1.732)/4$

So from (20) we have:

$\psi(x) = q_{-2}\phi(2x+2) + q_{-1}\phi(2x+1)+q_0\phi(2x+0) + q_1\phi(2x-1)$

$\psi(x) = ((1-1.732)/4) \, \phi(2x+2) - ((3-1.732)/4) \, \phi(2x+1) +$

$((3+1.732)/4) \, \phi(2x+0) - \, ((1+1.732)/4) \, \phi(2x-1)$

2. DESIGN CONCEPTS: CURVE EDITING OPERATIONS

There could be many ways a curve can be edited using wavelet transform, but the most general are discussed below because the rest are derivatives of these.

2.1 SMOOTHING

We are to solve the problem: Given a curve with average coefficients C with "m" control points, construct a best approximating curve with "m'" control points at some lower level having average coefficients C', where m'< m.

2.1.1 INTEGER LEVEL SMOOTHING

We can solve this problem by first running the decomposition algorithm, as given by [2]:

$$C^{n-1} = A^{n} C^{n}$$

Until a curve with just m' control points is reached. Here A and C are defined as before. Here the design issue is to find the values of A^n and B^n. The equation of their relation is [2]:

$$\begin{bmatrix} A^{j} \\ \hline B^{j} \end{bmatrix} = [\, P^{j} \mid Q^{j} \,]^{-1}$$

If we find A^n and B^n from the equation above then it leads to large processing times. There is a better approach and the idea is to compute C^{j-1} and D^{j-1} from C^j by solving the sparse linear system of equation [2]:

$$[\,P^j \mid Q^j\,] \left[\frac{C^{j-1}}{D^{j-1}} \right] = C^j$$

In order to solve this system for $\left[\frac{C^{j-1}}{D^{j-1}} \right]$, we first make the matrix $[\,P^j \mid Q^j\,]$ into a banded matrix simply by arranging the columns of P^j and Q^j. The resulting banded system can then be solved in linear time using LU decomposition (a numerical technique to solve matrix equations) that will be explained in the coming sections. Therefore we can compute the entire filter bank operation without ever forming and using A^j or B^j explicitly.

2.1.2 FRACTIONAL LEVEL SMOOTHING

One notable aspect of the multiresolution curve representation is its discrete nature. We can smooth a curve for each level this is known as integer level smoothing, but we have no obvious way to smooth a curve between two levels, i.e., 2.1 or 4.5, etc., this is known as fractional level smoothing.

To have a fractional level curve $f^{j+t}(u)$ for some $0 <= t <= 1$ in terms of a linear interpolation between its two nearest integer-level curves $f^j(u)$ and $f^{j+1}(u)$, as follows [2]:

$$\begin{aligned} f^{j+t}(u) &= (1\text{-}t)f^j(u) + t\, f^{j+t}(u) \\ &= (1\text{-}t)\, P^{j+1}C^j + tC^{j+1} \end{aligned}$$

These fractional-level curves allow for continuous levels of smoothing.

2.1.3 LU FACTORIZATION

An "nxn" system of linear equations can be written in matrix form as [6]:

$$Ax = b$$

Where A, x and b are matrices of valid orders. We have to find x matrix where A and b are known matrices.

First of all define [6]:

$$A = LU$$

Where, L = Lower triangular matrix, i.e., having non-zero elements on and below the diagonal and the zero elements are above the diagonal.

U = Upper triangular matrix, i.e., having non-zero elements on and above the diagonal and the zero elements are below the diagonal.

Solve in the following order, [6]:

$$LUx = b$$
$$z = L^{-1}b$$
$$x = U^{-1}z$$

Let's solve $[\, P^j \mid Q^j \,]\,[\, C^{j-1} / D^{j-1} \,] = C^j$ for our system. Apply LU decomposition on $[\, P^j \mid Q^j \,]$ and obtain L and U matrices, and follow the steps:

$$[\, P^j \mid Q^j \,]\,[\, C^{j-1} / D^{j-1} \,] = C^j$$
$$[\, P^j \mid Q^j \,] = LU$$
$$LU\,[\, C^{j-1} / D^{j-1} \,] = C^j$$
$$Z = L^{-1}C^j$$
$$[\, C^{j-1} / D^{j-1} \,] = U^{-1}Z$$

where,

$[\, C^{j-1} / D^{j-1} \,]$ is of order $(d+2^{j-1}+2^{j-1})x2$

C^{j-1} is of order $(d+2^{j-1})x2$

D^{j-1} is of order $(2^{j-1})x2$

While d, j, C and D have their usual meanings.

2.2 EDITING

The multiresolution analysis enables us to edit the curve in two very different types of ways:

- Changing a curves's overall "sweep" while maintaining its fine details, or "character".
- Changing a curve's "character" without affecting its overall "sweep".

Each one is described as below:

2.2.1 EDITING THE SWEEP OF A CURVE

Editing the sweep of a curve at an integer level of the wavelet transform is simple. Let C^n be the control points of the original curve $f^n(u)$. Let C^j be a low-resolution version of C^n, and let $\underline{C}^j$ be an edited version of C^j, given by $\underline{C}^j = C^j + \Delta C^j$. The edited version of the highest-resolution curve $\underline{C}^n = C^n + \Delta C^n$ can be computed through reconstruction [2]:

$$\begin{aligned} \underline{C}^n &= C^n + \Delta C^n \\ &= C^n + P^n P^{n-1} \dots P^{j+1} \Delta C^j \end{aligned}$$

Where,

P's are reconstruction filters, P^{j+1} at level "j+1" and P^n is at level "n"

if $\Delta c_i > 0$ then $\underline{c}_i > c_i$, that is new curve is above the given curve if $\Delta c_i < 0$ then $\underline{c}_i < c_i$, that is new curve is below the given curve

if $\Delta c_i = 0$ then no editing is done

c_i is a coefficient at ith. row in the $[C^n]^T$ matrix.

Also,

$\underline{C}^n = [\ c_0, c_1, \underline{c}_i, \ldots, c_n\ ]^T$, $C^n = [\ c_0, c_1, c_i, \ldots, c_n\ ]^T$, $\Delta C^n = [\ 0, 0, \Delta c_i, \ldots, 0\ ]^T$.

2.2.2 EDITING THE CHARACTER OF A CURVE

Second editing feature which is supported by multiresolution curves is editing the "fine details" or "character" of a curve, without affecting its overall sweep.

In this let C^n be the control points of a curve, and let $C^0, \ldots, C^{n-1}, D^0, \ldots, D^{n-1}$ denote the components of its multiresolution decomposition. Editing the character of the curve is simply a matter of replacing the existing set of detail functions $D^j, \ldots, D^{n-1}$ with some new set $\underline{D}^0, \ldots, \underline{D}^{n-1}$, and apply reconstruction to achieve higher levels, i.e., $C^j = P^j C^{j-1} + Q^j D^{j-1}$.