Wavelets in Computer Graphics: Multiresolution Curve Editing

Group 13:
T M Feroz Ali (09307906)
Deepak Baby (10307019)
Harikrishnan Potty (10307926)
Sreeraj R (10307933)

28 March, 2011

1 Abstract

Multiresolution analysis and wavelets provide useful and efficient tools for representing functions at multiple levels of detail. We are planning to implement a multiresolution curve representation, based on wavelets. The main advantage of such a representation is that it conveniently supports a variety of operations like smoothing a curve, editing the overall sweep of a curve while preserving its character. We present methods to implement continuous levels of smoothing as well as direct manipulation of the sweep of the curve. These manipulations are done based on the control points, which are defined by the user and it is done in such a way that the underlying properties of the curve are preserved. The multiresolution representation requires no extra storage beyond that of the original control points, and the algorithms using the representation are both simple and fast.

2 Introduction

Wavelets can be used as tools for hierarchically decomposing functions. They allow a function to be described in terms of a coarse overall shape, plus details that range from broad to narrow. Wavelets have recently been applied to many problems in computer graphics. These graphics applications include image editing, image compression, and image querying; automatic level-of-detail control for editing and rendering curves and surfaces; surface reconstruction from contours; and fast methods for solving simulation problems in animation. Regardless of whether the function of interest is an image, a curve, or a surface, wavelets offer an elegant technique for representing the levels of detail present. There are many applications of multiresolution curves, including computer aided design, in which cross-sectional curves are frequently used in the specification of surfaces; keyframe animation, in which curves are used to control parameter interpolation; 3D modeling and animation, in which backbone curves are manipulated to specify object deformations; graphic design, in which curves are used to describe regions of constant color or texture; font design, in which curves represent the outlines of characters; and pen-and-ink illustration, in which curves are the basic elements of the finished piece. In all of these situations, the editing, smoothing, and approximation techniques using wavelets can be powerful tools.

Wavelet representation for curves allows for flexible editing, smoothing, and scan conversion. It provides

1. the ability to change the overall sweep of a curve while maintaining its fine details, or character;
2. the ability to change a curves character without affecting its overall sweep
3. the ability to edit a curve at any continuous level of detail, allowing an arbitrary portion of the curve to be affected through direct manipulation
4. continuous levels of smoothing, in which undesirable features are removed from a curve;
5. curve approximation, or fitting, within a guaranteed maximum error tolerance, for scan conversion and other applications.

It requires no extra storage beyond that of the original m control points, and the algorithms that use it are both simple and fast, typically linear in m. In this project we have implemented the .

3 Theory of multiresolution curves

Consider a discrete signal C^n, expressed as a column vector of samples $[c_1^n, ..., c_m^n]^T$. In our application, the samples c_i^n could be thought of as the control points of a curve in $\mathbb{R}^2$. Suppose we wish to create a low-resolution version C^{n-1} of C^n with a fewer number of samples m'. The standard approach for creating the m' samples of C^{n-1} is to use some form of filtering and downsampling on the m samples of C^n. This process can be expressed as a matrix equation

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

where A^n is an $m' \times m$ matrix.
Since C^{n-1} contains fewer samples than C^n, it is intuitively clear that some amountof detail is lost in this filtering process. If A^n is appropriately chosen, it is possible to capture the lost detail as another signal D^{n-1}, computed by

$$D^{n-1} = B^n C^n \tag{2}$$

where B^n is an $(m - m') \times m$ matrix, which is related to matrix A^n. The pair of matrices A^n and B^n are called analysis 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 detail Q^n called synthesis filters, as follows:

$$C^n = P^n C^{n-1} + Q^n D^{n-1} \tag{3}$$

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

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 signal can be expressed as a hierarchy

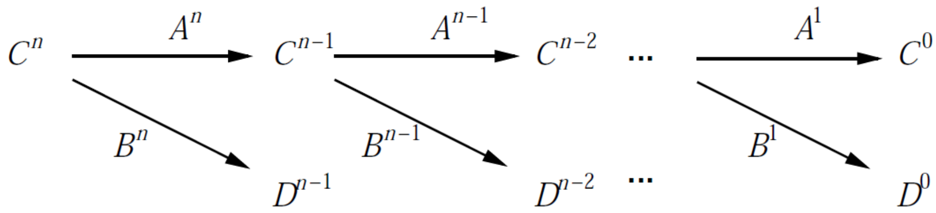

Figure 1: The Filter Bank

of lower-resolution signals $C^0, ..., C^{n-1}$ and details $D^0, ..., D^{n-1}$.This recursive process is known as a filter bank.

Since the original signal C^n can be recovered from the sequence $C^0, D^0, D^1, ..., D^{n-1}$, this sequence can be thought of as a transform of the original signal, known as a wavelet transform. Note that the total size of the transform $C^0, D^0, D^1, ..., D^{n-1}$ is the same as that of the original signal C^n, so no extra storage is required.

Hence all that is needed for performing a wavelet transform is an appropriate set of analysis and synthesis filters A^j, B^j, P^j, Q^j. To see how to construct these filters, we associate with each signal C^n a function $f^n(u)$ with $u \in [0,1]$ given by

$$f^n(u) = \Phi^n(u)C^n \tag{4}$$

where $\Phi^n(u)$ is a row matrix of basis functions $[\phi_1^n(u), ..., \phi_m^n(u)]$,called scaling functions. In our application, for example, the scaling functions are the endpoint-interpolating B-splines basis functions, in which case the function $f^n(u)$ would be an endpoint-interpolating B-spline curve.The scaling functions are required to be refinable; that is, for j in $[1, n]$ there must exist a matrix P^j such that

$$\Phi^{j-1} = \Phi^j P^j \tag{5}$$

In other words, each scaling function at level $j-1$ must be expressible as a linear combination of finer scaling functions at level j. As suggested by the notation, the refinement matrix turns out to be the same as the synthesis filter P^j.

Next, let V^j be the linear space spanned by the set of scaling functions Φ^j. The refinement condition on Φ^j implies that these linear spaces are nested: $V^0 \subset V^1 \subset \cdots \subset V^n$. Choosing an inner product for the basis functions in V^j allows us to define W^j as the orthogonal complement of V^j in V^{j+1}, that is, the space W^j whose basis functions $\Psi^j = [\psi_1^j(u), ..., \psi_{m-m'}^j(u)]$ are such that Φ^j and Ψ^j together form a basis for V^{j+1}, and every $\psi_i^j(u)$ is orthogonal to every $\phi_1^j(u)$ under the chosen inner product. The basis functions $\psi_i^j(u)$ are called wavelets.
We can now construct the synthesis filter Q^j as the matrix that satisfies

$$\Psi^{j-1} = \Phi^j Q^j \tag{6}$$

Above two equations can be expressed as a single equation by concatenating the matrices together:

$$\left[\Phi^{j-1} \quad | \quad \Psi^{j-1}\right] = \Phi^{j}\left[P^{j} \quad | \quad Q^{j}\right] \tag{7}$$

Finally, the analysis filters A^j and B^j are formed by the matrices satisfying the inverse relation:

$$\left[\Phi^{j-1} \quad | \quad \Psi^{j-1}\right] \left[\frac{A}{B}\right] = \Phi^{j} \tag{8}$$

Note that $[P^j|Q^j]$ and $[A^j|B^j]^T$ are both square matrices. Thus,

$$\left[\frac{A^j}{B^j}\right] = \left[P^j|Q^j\right]^{-1} \tag{9}$$

from which it is easy to prove a number of useful identities:

$$A^j Q^j = B^j P^j = \mathbf{0} \tag{10}$$

$$A^j P^j = B^j Q^j = P^j A^j + Q^j B^j = \mathbf{1} \tag{11}$$

where $\mathbf{0}$ and $\mathbf{1}$ are the matrix of zeros and the identity matrix, respectively.

4 Spline Wavelets

There are a variety of ways to construct wavelets. One such class of wavelets can be constructed from piecewise-polynomial splines. The Haar basis is in fact the simplest instance of spline wavelets, resulting when the polynomial degree is set to zero. Here we use endpoint-interpolating cubic B-spline wavelets.

4.1 B-spline scaling functions

Our first step is to define the scaling functions for a nested set of function spaces. We shall start with the general definition of B-splines, then specify how to make uniformly spaced, endpoint-interpolating B-splines from these.

Given positive integers d and k, with $k > d$, and a collection of non-decreasing values $x_0, ..., x_{k+d+1}$ called knots, the nonuniform Bspline basis functions of degree d are defined recursively by the Cox-de Boor recursion formula as follows. For $i = 0, ..., k$ and for $r = 1, ..., d$, let

$$N_i^0(x) = \begin{cases} 1 & \text{if } x_i \leq x \leq x_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_i^0(x) = \frac{x - x_i}{x_{i+r} - x_i} N_i^{r-1}(x) + \frac{x_{i++r+1} - x}{x_{i+r+1} - x_{i+1}} N_{i+1}^{r-1}(x) \tag{12}$$

The fractions in these equations are taken to be 0 when their denominators are 0.

The endpoint-interpolating B-splines of degree d on $[0, 1]$ result when the first and last $d + 1$ knots are set to 0 and 1, respectively. In this case, the functions $N_0^d(x), ..., N_k^d(x)$ form a basis for the space of piecewise-polynomials of degree d.

To make uniformly spaced B-splines, we choose $k = 2^j + d - 1$ and $x_{d+1}, ..., x_k$ to produce 2^j equally spaced interior intervals. This construction gives $2^j + d$ B-spline basis functions for a particular degree d and level j. We will use these functions as the endpoint-interpolating B-spline scaling functions. Figure 2 shows examples of these functions at level $j = 1$ (two interior intervals) for various degrees d.

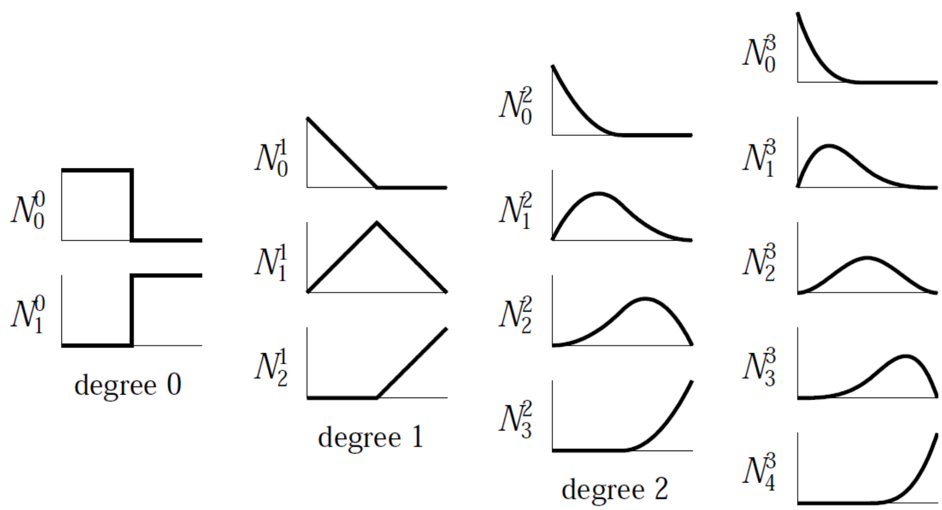

Figure 2: B-spline scaling functions for $V^1(d)$ with degree $d = 0, 1, 2,$ and 3.

4.2 Calculation of P^j

The rich theory of B-splines can be used to develop expressions for the entries of the refinement matrix P^j.

$$\Phi^{j-1} = \Phi^j P^j \tag{13}$$

As suggested by the notation, the refinement matrix in equation 12 turns out to be the same as the synthesis filter P^j.

The P^j matrix is straightforward to derive from the Cox-de Boor recursion formula; it encodes how each endpoint-interpolating B-spline can be expressed as a linear combination of B-splines that are half as wide. The scaling basis functions Φ^j are calculated for 1000 points for $j = 1$ to 9 and the pseudo inverse is taken based on equation 12 to find P^j.

$$P^1 = \frac{1}{16}\begin{bmatrix} 16 & 0 & 0 & 0 \\ 8 & 8 & 0 & 0 \\ 0 & 8 & 8 & 0 \\ 0 & 0 & 8 & 8 \\ 0 & 0 & 0 & 16 \end{bmatrix} \tag{14}$$

$$
P^2 = \frac{1}{16} \begin{bmatrix} 16 & 0 & 0 & 0 & 0 \\ 8 & 8 & 0 & 0 & 0 \\ 0 & 12 & 4 & 0 & 0 \\ 0 & 3 & 10 & 3 & 0 \\ 0 & 0 & 4 & 12 & 0 \\ 0 & 0 & 0 & 8 & 8 \\ 0 & 0 & 0 & 0 & 16 \end{bmatrix} \tag{15}
$$

For $j > 2$, P^j can be written in general as shown in Figure 3.

$$
P^j = \frac{1}{8} \begin{bmatrix}
8 & & & & & & & & & \\
4 & 4 & & & & & & & & \\
 & 6 & 2 & & & & & & & \\
 & \frac{3}{2} & \frac{11}{2} & 1 & & & & & & \\
 & & 4 & 4 & & & & & & \\
 & & 1 & 6 & 1 & & & & & \\
 & & & 4 & 4 & & & & & \\
 & & & 1 & 6 & & & & & \\
 & & & & 4 & \ddots & & & & \\
 & & & & 1 & \ddots & 1 & & & \\
 & & & & & \ddots & 4 & & & \\
 & & & & & & 6 & 1 & & \\
 & & & & & & 4 & 4 & & \\
 & & & & & & 1 & \frac{11}{2} & \frac{3}{2} & \\
 & & & & & & & 2 & 6 & \\
 & & & & & & & & 4 & 4 \\
 & & & & & & & & & 8
\end{bmatrix}
$$

Figure 3: $P^{j \geq 3}$

The columns of P^j are sparse, reflecting the fact that the B-spline basis functions are locally supported. Note that P^j is a matrix with dimensions $(2^j + 3) \times (2^{j-1} + 3)$ whose middle columns, for $j \geq 3$, are given by vertical translates of the fourth column, shifted down by 2 places for each column.

4.3 B-spline wavelets

To derive the Q^j matrix, the following procedure is adopted. Given two row vectors of functions X and Y, let $[\langle X|Y \rangle]$ be the matrix of inner products $\langle X_k, Y_l \rangle$. Since by definition, scaling functions and wavelets at the same level j are orthogonal, we have

$$
\left[\langle \Phi^j | \Psi^j \rangle\right] = \left[\langle \Phi^j | \Phi^{j+1} \rangle\right] Q^{j+1} = \mathbf{0} \tag{16}
$$

So the columns of Q^{j+1} span the null space of $[\langle \Phi^j | \Phi^{j+1} \rangle]$. The null space was found using Singular Value Decomposition. A basis for this null space was chosen by finding the matrix Q^{j+1} that has columns with the shortest runs of non-zero coefficients; this matrix corresponds to the wavelets with minimal support. Matrix Q^j has the same structure for $j \geq 4$, except with dimensions $(2^j+3) \times (2^{j-1})$.

4.4 B-spline filter bank

At this point, we have completed the steps in designing a multiresolution analysis. However, to use spline wavelets, we must implement a filter bank procedure incorporating the analysis filters A^j and B^j. These matrices allow us to determine C^{j-1} and D^{j-1} from C^j using matrix multiplication.The analysis filters are uniquely determined by the inverse relation:

$$\left[\frac{A^j}{B^j}\right] = \left[P^j|Q^j\right]^{-1} \tag{17}$$

5 Smoothing

Given a curve with m control points C, we can construct a best least-squares-error approximating curve with m' control points C', where $m' < m$. Here, we will assume that both curves are endpoint-interpolating uniform B-spline curves. The multiresolution analysis framework allows this problem to be solved trivially, for certain values of m and m'. Assume for the moment that $m = 2^j + 3$ and $m' = 2^{j'} + 3$ for some non-negative integers $j' < j$. Then the control points C' of the approximating curve are given by

$$C' = A^{j'+1}A^{j'+2}....A^jC \tag{18}$$

In other words, we simply run the decomposition algorithm, as described by equation(1), until a curve with just m' control points is reached.

One notable aspect of the multiresolution curve representation is its discrete nature. Thus, in our application it is easy to construct approximating curves with 4, 5, 7, 11, or any $2^j + 3$ control points efficiently, for any integer level j. To construct curves that have levels of smoothness in between the best solution, a fractional-level curve $f^{j+t}(u)$ is defined for some $0 \le t \le 1$ in terms of a linear interpolation between its two nearest integer-level curves $f^j(u)$ and $f^{j+1}(u)$ as follows :

$$\begin{aligned} f^{j+t}(u) &= (1-t)f^j(u) + tf^{j+1}(u) \\ &= (1-t)\Phi^j(u)C^j + t\Phi^{j+1}(u)C^{j+1} \end{aligned} \tag{19}$$

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

6 Editing

Suppose we have a curve C^n and all of its low-resolution and detail parts $C^0, .., C^{n-1}$ and $D^0, .., D^{n-1}$. Multiresolution analysis allows for two very different kinds of curve editing. If we modify some low-resolution version C^j and then add back in the detail $D^j, D^{j+1}, .., D^{n-1}$, we will have modified the overall sweep of the curve . On the other hand, if we modify the set of detail functions $D^j, D^{j+1}, .., D^{n-1}$ but leave the low-resolution versions $C^0, .., C^j$ intact, we will have modified the character of the curve, without affecting its overall sweep.

Editing the sweep of a curve at an integer level of the wavelet transform is done as follows. 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 $\widehat{C^j}$ be an edited version of C^j given by $\widehat{C^j} = C^j + \triangle C^j$. The edited version of the highest-resolution curve $\widehat{C^n} = C^n + \triangle C^n$ can be computed through reconstruction:

$$\begin{aligned}\widehat{C^n} &= C^n + \triangle C^n \\ &= C^n + P^n P^{n-1} \cdots P^{j+1} \triangle C^j. \qquad (20)\end{aligned}$$

Editing the sweep of the curve at lower levels of smoothing j affects larger portions of the high-resolution curve $f^n(u)$. At the lowest level, when $j = 0$, the entire curve is affected; at the highest level, when $j = n$, only the narrow portion influenced by one original control point is affected.

7 Results

The matrices A^j,B^j,P^j,Q^j,Φ^j and Ψ^j are generated using the procedures that are explained above. The basis function obtained for level 3 are plotted in Figure 4. The wavelets are obtained using $\Psi^{j-1} = \Phi^j Q^j$. The wavelets obtained for level 4 are also plotted in Figure 5.

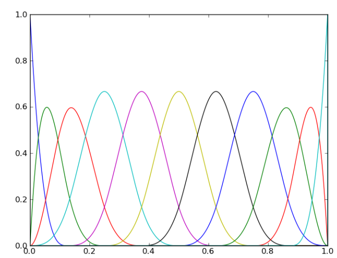

Figure 4: B-spline basis functions for level 3

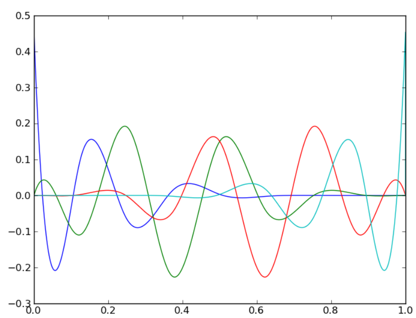

Figure 5: B-spline wavelet functions for level 4

7.1 Smoothing Curve

The experiment is run for curve smoothing at various levels of resolutions. These are done using the analysis matrix A^j. The various smoothened versions of the word 'roll' written free hand, are shown. The idea of fractional level spline representation is also obtained. In Figure 9 the word 'roll' smoothened at level 5.7 is given.

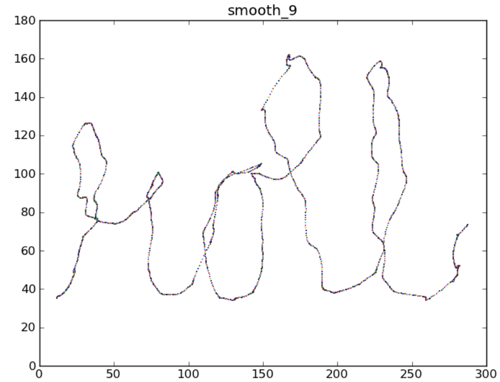

Figure 6: Smoothing a curve continuously. The original curve at level 9

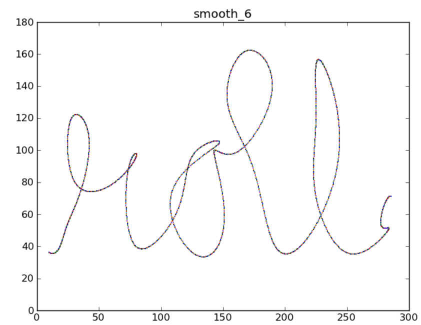

Figure 7: The smoothened curve at level 6

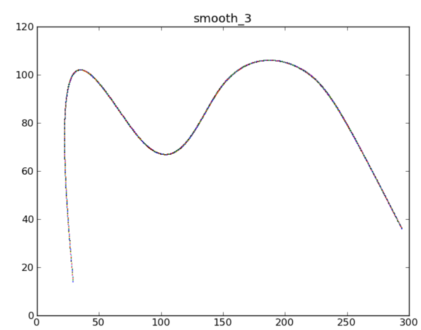

Figure 8: The smoothened curve at level 6

It can be seen that the curves get smoother and smoother as we decrease the resolution. For further decrease say, for level 3 the information in the curve seems almost lost. The smoothening of the curve at a fractional level yields better ways of smoothening.

7.2 Editing the sweep

For editing the sweep of a curve by defining a new set of control points $\hat{C}^j$, a curve for the word 'hell' was taken. The new sweep of the curve is defined at level 0 using just 4 control points and the

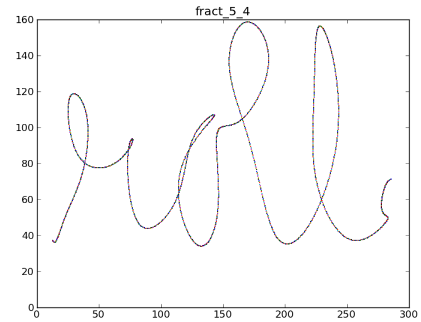

Figure 9: The fractional curve at level 5.7

resulting sweep curve is shown in Figure 10.

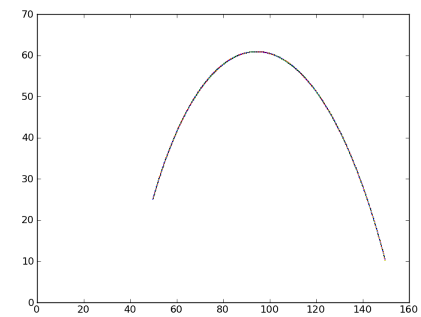

Figure 10: The sweep curve

The original curve and the sweep edited versions are shown below. This shows that in order to edit the sweep, one need not have to work with 1024 points. We can bring the curve to a lower resolution version and then edit it at this lower level and then by using the synthesis filter P^j, we can have its edited version. the results are shown.

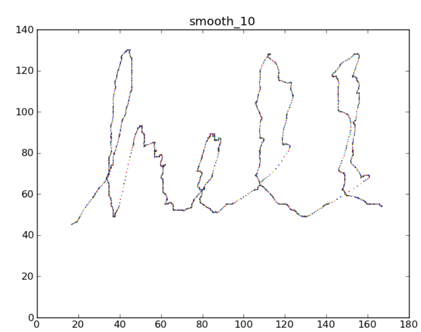

Figure 11: The original curve

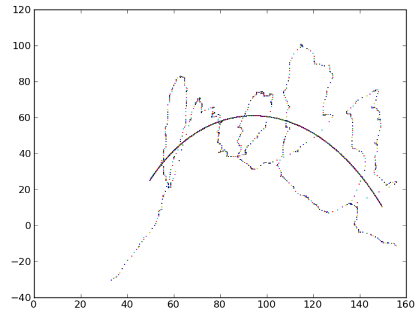

Figure 12: The edited curve at level 9

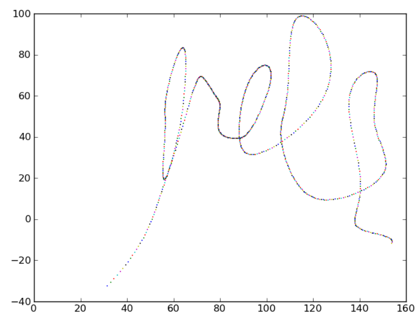

Figure 13: The smoothened version of edited curve

8 Conclusions and Future Work

The B-spline representation yields an effective way of representing a curve by means of control points. By using the idea of multi-resolutional analysis, we can smoothen the curve to any level. The analysis and synthesis filters yields a simpler methodology in implementing this goal. The future work includes editing the character of the curve without affecting its sweep and data compression using the spline wavelets obtained.

References

[1] Adam Finkelstein and David H. Salesin, "Multiresolution Curves", Department of Computer Science and Engineering University of Washington Seattle, Washington 98195, In Proceedings of SIGGRAPH 94, pages 261268. ACM, New York, 1994.

[2] Eric J. Stollnitz Tony D. DeRose David H. Salesin, "Wavelets for Computer Graphics: A Primer Part 1", Wavelets for Computer Graphics: A Primer Part 1, 1994.

[3] Eric J. Stollnitz Tony D. DeRose David H. Salesin, "Wavelets for Computer Graphics: A Primer Part 1", Wavelets for Computer Graphics: A Primer Part 2, 1995.

[4] HTTP Site, "http://www.cs.mtu.edu/ shene/COURSES/cs3621/NOTES/spline"