**INFORMATION TO USERS** 

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of

computer printer.

The quality of this reproduction is dependent upon the quality of the

copy submitted. Broken or indistinct print, colored or poor quality illustrations

and photographs, print bleedthrough, substandard margins, and improper

alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript

and there are missing pages, these will be noted. Also, if unauthorized

copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by

sectioning the original, beginning at the upper left-hand corner and continuing

from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced

xerographically in this copy. Higher quality 6" x 9" black and white

photographic prints are available for any photographs or illustrations appearing

in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning 300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA 800-521-0600

**UMI**®

	·			

# DEVELOPMENT OF A MULTIRESOLUTION FRAMEWORK FOR NURBS BY MOHAMMED ALI SIDDIQUI A Thesis Presented to the DEANSHIP OF GRADUATE STUDIES KING FAHD UNIVERSITY OF PETROLEUM & MINERALS DHAHRAN, SAUDI ARABIA In Partial Fulfillment of the Requirements for the Degree of MASTER OF SCIENCE In COMPUTER SCIENCE DECEMBER 2001

**UMI Number: 1409818** 

# IMI

### UMI Microform 1409818

Copyright 2002 by ProQuest Information and Learning Company.
All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company 300 North Zeeb Road P.O. Box 1346 Ann Arbor, MI 48106-1346

# KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS DHAHRAN 31261, SAUDI ARABIA

### **DEANSHIP OF GRADUATE STUDIES**

This thesis, written by

### MOHAMMED ALI SIDDIQUI

under the direction of his thesis advisor and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE.

THESIS COMMITTEE

Dr. Muhammad Sarfraz (Chairman)

Dr. Kanaan A. Faisal (Member)

Dr. Muhammad A.U. Khan (Member)

Dr. Kanaan A Faisal (Department Chairman)

Prof. Osama A. Jannadi (Dean of Graduate Studies)

Date

# Dedicated to

my beloved parents, brother,

and

 $all\ my\ teachers.$ 

### **ACKNOWLEDGEMENTS**

All praises and glory to Almighty Allah (Subhaanahu wa Ta'ala) for bestowing me with health, knowledge and patience to complete this work. Peace and blessing of Allah be upon His Prophet, Hazrat Muhammad (Sallallaho Alaihi Wasallam).

Acknowledgement is due to King Fahd University of Petroleum and Minerals for providing support for carrying out this work and for giving me the opportunity to pursue graduate studies.

I acknowledge, with deep gratitude and appreciation for the inspiration, encouragement, technical and moral support, valuable time, and guidance given to me by my Advisor and Thesis Committee Chairman, Dr. Muhammad Sarfraz throughout the course of this work.

Thanks are due to my thesis committee members Dr. Faisal A. Kanaan and Dr. Muhammad Asmatullah Khan for their interest, suggestions, invaluable cooperation and support.

Thanks are also due to the Administrative Assistants of ICS Department Mr. Mir Azmath Ansari and Mr. Madakkara Purushothaman for their help and assistance.

Thanks are due to my colleagues at the University, Murtaza Ali Khan, Syed Arshad Raza, Faisal Abdul-Razzaq, Owais Ahmed Malik, Muhammad Wasiq, Faisal

Alvi, Abdul-Baqi Sharafaleslam and others for their timely help and motivating the friendly learning environment.

Special thanks are due to my house mates, and friends for their help and encouragement. A few of them are Mr. Faiz Mohammed Khan, Salman Abdul-Gaffar (BR), Syed Imran Ali (AW), Aurifullah Vantala (1-2, Khamooshi), Mohammed Abdul Majid (PS, GBS, DW), Alisher Abdullah (Baba, Abbu sher), Mohammed Abdul-Baseer (Sannata), Mohammed Hameeduddin Sameer (RW), Syed Anwar Basha (Peddarayudu), Shaikh Khasim-ul-Khasim, Mohammed Jalaluddin Shah (Manager), Mohammed Mujataba Hussain, Syed Omar Farooq, Mohammed Faheem (Khabri), and Atiul Ahmed. They have made my work and stay at KFUPM (specially at NC) very pleasant and provided wonderful company and a lot of memories that will last a lifetime. Thanks are also due to my seniors Mr. Gulam Farooq, Mr. Syed Abdul-Vakeel Imran, Mr. Syed Mazharuddin Ahmed, and others for their help and encouragement.

Thanks are due to Mr. Syed Ahmed Hussain and Anees, who were always there for help. And finally, my heartfelt thanks to my mother, late father, and brother for their emotional and moral support throughout my academic career and also for their love, patience, encouragement, and prayers.

# **Contents**

	Acl	knowledgements	ii
	Lis	t of Tables	vii
	List	of Figures v	iii
	Abs	stract (English) x	iii
	Abs	stract (Arabic)	iv
1	Intr	roduction	1
	1.1	Theory of Splines	4
	1.2	Spline Definitions	4
	1.3	A Bit of History	5
	1.4	Types of Spline Curves	6
		1.4.1 Approximatory Splines	8
		1.4.2 Interpolatory Splines	۱7

	1.5	Some Other Splines	18
	1.6	Objective	18
	1.7	Thesis Organization	19
2	NU	RBS	21
	2.1	Introduction	21
	2.2	Mathematical Definition	22
	2.3	Basis Functions	24
		2.3.1 Knots and Knot Vector	24
		2.3.2 Important Properties of NURBS Basis Functions	26
	2.4	Important Properties of NURBS Curves	27
	2.5	Effect of Modifying Weights on NURBS	30
3	Mu	tiresolution Representation of Splines	35
	3.1	Background Material on Wavelets	37
		3.1.1 Introduction to Wavelets	38
		3.1.2 Mathematical Description of Wavelet Transform	46
		3.1.3 Important Properties of Wavelets	47
		3.1.4 Wavelets and Multiresolution Analysis	50
	3.2	Multiresolution Decomposition of End-point Interpolating B-splines	
		Using Wavelets	55
		3.2.1 Construction of B-spline Wavelets	55

		3.2.2 Application to Curves and Surfaces	<b>56</b>
	3.3	Multiresolution of NUBS Using Knot Decimation	63
		3.3.1 Least-Squares Multiresolution Decomposition	64
		3.3.2 Manipulation of Multiresolution Decompositions	67
4	Mu	tiresolution Representation of NURBS 7	70
	4.1	Development of the Model	71
	4.2	Results	30
		4.2.1 Multiresolution Decomposition	30
		4.2.2 Editing Curves using Multiresolution Representation 8	38
		4.2.3 Multiresolution Surfaces	)3
5	Mu	tiresolution of Splines: A Comparative Study 12	<b>:</b> 0
	5.1	Limitations of the Multiresolution Models	21
	5.2	Multiresolution Model Comparison: Wavelets v/s Control Point Dec-	
		imation	!2
	5.3	Multiresolution Model Comparison: Knot Decimation v/s Control	
		Point Decimation	6
3	Con	clusion and Future Work 130	0
	6.1	Conclusion	0
	62	Futura Wark	

# List of Tables

4.1	Details of Curves of figure 4.9	80
4.2	Details of Curves of figure 4.13	85
5.1	Multiresolution Method's Ability to Represent the Types of B-splines.	l <b>2</b> 1
5.2	Figure 5.1 Attributes	23
5.3	Figure 5.4 Attributes	26

# List of Figures

1.1	A Draftsman Spline	6
1.2	Curve Approximates the Control Points : An Approximatory Spline.	7
1.3	Curve Interpolates Through All Control Points: An Interpolatory	
	Spline	7
1.4	Global Control in Bezier Curves as a Result of Change in Position of	
	Single Control Point	10
1.5	A B-spline Curve with Open Uniform Knot Vector	13
1.6	A B-Spline Curve with Periodic Knot Vector	13
1.7	B-Spline Curve Basis Functions	15
1.8	Local Control in B-spline Curves	20
2.1	B-Spline Curve Basis Functions	25
2.2	Variation Diminishing Property in NURBS.	29
2.3	NURBS Curve with Default Shape Parameters	31
2.4	A NURBS curve with application of different weights at $P_4$	32

2.5	Application of Different Weight Values on $P_4$ and its Effect on the	
	Shape of the Curve	33
3.1	Signal $x(t)$ with 10, 25, 50, and 100Hz Frequencies	41
3.2	Fourier Transform of the Signal $x(t)$ of figure 3.1	41
3.3	A Non-Stationary signal with 10, 25, 50, and 100Hz Frequencies	42
3.4	Fourier Transform of the Signal of figure 3.3	43
3.5	Mother Wavelet.	49
3.6	Shift-Invariant Wavelets	49
3.7	Shift-Variant Wavelets	52
3.8	The Filter Bank.	52
3.9	Cubic B-spline Scaling functions and wavelets for $j=3.$	59
3.10	Smoothing a curve continuously. From left to right: the original curve	
	at level 8, and smoother versions at level 5.4 and 3.1	59
3.11	Changing the overall sweep of a curve without affecting its character.	
	From left to right: the original curve, its extracted overall sweep, the	
	sweep modified by the user, the details reapplied to the modified sweep.	59
3.12	Changing the character of the curve without affecting its sweep	60
3.13	Decomposition of a Surface.	62
3.14	Surface manipulation at different levels of detail	64

3.15	Multiresolution Decomposition of a B-spline Star Curve $C_5(t)$ (of or-	
	der 3, defined with 100 control points.). The Original Curve is shown	
	in thin lines throughout while $C_i(t)$ s, $0 \le i < 5$ are shown in thick	
	lines	68
3.16	Editing the same curve location, the center of the s in the signature,	
	at different resolution levels, from the lowest (top left) to the highest	
	resolution (right bottom). The original curve is shown in thin lines	
	and the low resolution curves are displayed in thick lines	68
3.17	(a) Low resolution quadratic B-spline curve (with 22 control points),	
	representing a cross-section of an airplane. (b) The curve of (a) is	
	locally refined to create the degrees of freedom that are necessary to	
	interactively model the missiles at the wing tips, the elevators and	
	the steering wings, as well as the cockpit	69
4.1	B-spline curve shape control by changing the type of knot vector	79
4.2	B-spline curve shape control by changing the Order (k) of the Basis	12
1.0		
	Functions	73
4.3	B-spline curve shape control by changing the <i>Position of Control Points</i> .	73
4.4	B-spline curve shape control by using Multiple Control Points	74
4.5	B-spline curve shape control by using Multiple Knots	74
4.6	NURBS curve shape control by changing Weight values	75

4.7	Flow Chart of the Multiresolution Decomposition Process	<b>78</b>
4.8	A NURBS Curve with default weight values	81
4.9	Multiresolution levels of the curve in figure 4.8	82
4.10	A NURBS Curve with non-default weight values	83
4.11	Multiresolution levels of the curve in figure 4.10	84
4.12	A Closed NURBS Curve	86
4.13	Multiresolution levels of the curve in figure 4.12	87
4.14	Original NURBS Curve.	89
4.15	Diagrammatic representation of editing at level 3	90
4.16	NURBS Curve with editing performed at level 3	91
4.17	Diagrammatic representation of editing at level 4	93
4.18	NURBS Curve with editing performed at level 4	94
4.19	Diagrammatic representation of editing at level 5	95
4.20	NURBS Curve with editing performed at level 5	96
4.21	Diagrammatic representation of editing at all levels	97
4.22	NURBS Curve with editing performed at all levels	98
4.23	A NURBS Curve	99
1.24	Decomposition levels of the NURBS Curve	00
1.25	All levels of Locally Edited NURBS Curve	01
1.26	Locally Edited Final NURBS Curve	02
1.27	A Bi-Cubic NURBS Surface with a 30 X 30 mesh of Control Points 10	05

4.28	Decomposed NURBS Surface with 15 X 15 mesh of Control Points 106
4.29	Decomposed NURBS Surface with 8 X 8 mesh of Control Points 107
4.30	Decomposed NURBS Surface with 4 X 4 mesh of Control Points 108
4.31	A NURBS Surface with 15 X 15 mesh of Control Points 109
4.32	Decomposed NURBS Surface with 10 X 10 mesh of Control Points 110
4.33	Decomposed NURBS Surface with 8 X 8 mesh of Control Points 111
4.34	Decomposed NURBS Surface with 7 X 7 mesh of Control Points 112
4.35	A NURBS Surface with 33 X 33 mesh of Control Points
4.36	Decomposed NURBS Surface with 17 X 17 mesh of Control Points 114
4.37	Decomposed NURBS Surface with 9 X 9 mesh of Control Points 115
4.38	Decomposed NURBS Surface with 5 X 5 mesh of Control Points 116
4.39	A NURBS Surface with 19 X 19 mesh of Control Points
4.40	Decomposed NURBS Surface with 10 X 10 mesh of Control Points 118
4.41	Decomposed NURBS Surface with 5 X 5 mesh of Control Points 119
5.1	A Uniform B-spline Curve
5.2	Multiresolution levels using Wavelets
5.3	Multiresolution levels by Control Point Decimation Method 125
5.4	A Star Shaped NUBS Curve
5.5	Multiresolution levels by Knot Decimation Method
5.6	Multiresolution levels by Control Point Decimation Method 129

### THESIS ABSTRACT

Name: Mohammed Ali Siddiqui

Title: Development of a Multiresolution Framework for NURBS

**Degree:** MASTER OF SCIENCE

Major Field: Computer Science

Date of Degree: December 2001

The piecewise polynomial B-spline representation is a flexible tool in CAGD for representing and designing the geometric objects. In the field of Computer Graphics or Computer Aided Design, a very useful property for a given spline model is to have locally supported basis functions, in order to allow localized modifications of the shape. Unfortunately this property can also become a serious disadvantage when the user wishes to edit the global shape of a complex object. Multiresolution representation is proposed as a solution to alleviate this problem. Various multiresolution methods are proposed for different B-spline models.

In our work, we propose a multiresolution representation for NURBS (Non-Uniform Rational B-Spline). Among the types of B-splines, NURBS have been receiving considerable attention in the areas of computer graphics and geometric modeling. The proposed multiresolution model uses control point decimation strategy for decomposition of NURBS curves. It is extended to represent using multiresolution the NURBS surfaces. Our proposed method is efficient in both time and space utilization. This work makes a comparative study of our proposed method and two of the already existing multiresolution representation methods for B-splines, one that uses the B-spline wavelets and the other is based on knot decimation and least squares approximation strategy.

King Fahd University of Petroleum and Minerals, Dhahran.

December 2001.

### خلاصة الرسالة

الاسم: محمد على صديقي العنوان: تطوير اطار التحليل المتعدد لـ NURBS

الدرجة: المأجستير في العلوم

التخصيص الرنيسي: علوم الحاسب الآلي

تاريخ التخرج: كانون الأول "ديسمبر" 2001

يعتبر التمثيل عن طريق متعدد الحدود المكون من قطع الشرائح- ب (B-spline) أداة مرنه في التصميم الهندسي بمساعدة الحاسب الآلي (CAGD) لوصف وتصميم الأشكال الهندسية. ففي مجال الرسم الحاسوبي أو التصميم بمساعدة الحاسب، يعتبر وجود دوال أساسية خاصية مفيده جدا لنموذج مكون من شرائح حيث تمكننا من أجراء تعديلات محليه على الشكل دون المساس بمظهره العام. غير أن هذه الخاصيه يمكن أن تصبح نقطة ضعف بليغة وذلك عندما يرغب المستخدم رسم الشكل الكلي لجسم معقد. في هذه الحالة يُقترح التحليل المتعدد (multiresolution) كوسيله لتخفيف هذه المعضلة. هناك عدة طرق للتحليل المتعدد اقترحت لنماذج مختلفة من شرائح- ب.

لقد تم في هذه الرساله اقتراح تمثيل التحليل المتعدد له NURBS حيث لقي اهتماما كبيرا من بين عدة أنواع لشرائح - ب في مجالات الرسم الحاسوبي والنمذجة الهندسية يستخدم النموذج المقترح للتحليل المتعدد استراتيجية تبديد نقاط التحكم وذلك لتفصيل منحنيات NURBS. تعتبر الطريقة المقترحة عملية وفعالة في استغلال الزمن والحيز التخزيني. من خلال هذا العمل ، أجريت دراسة مقارنة بين طريقتنا المقترحة واثنان من الطرق المستعملة سلفا في التحليل المتعدد الشرائح - ب. تستعمل إحدى هذه الطرق مويجات شرائح - ب بينما تتركز الثانية على استراتيجية تيديد العقد والتقريب التربيعي الأدنى.

جامعة الملك فهد للبترول و المعادن كانون اول "ديسمبر" 2001

# Chapter 1

## Introduction

In the field of Geometric Modeling, the construction of efficient, intuitive, and interactive editors for geometric objects is a fundamental objective, but it is yet a difficult challenge. In many freeform geometric modeling systems the users are allowed to work in the framework of a specific data model, e.g. Bezier or non-uniform rational B-splines [9]. This imposes constraints on the set of geometric manipulation operations that can be performed, the man-machine interface and the type of objects which can be modeled.

There are various curve manipulation techniques have been proposed. The Euclidean distances between the point of modification and the control points of a B-spline curve were used as weights to affect the control points in [5]. The difficulty with this approach appears when the two separate portions of the curve are close. To alleviate the difficulty in editing freeform shapes while matching engineer-

ing specifications, constraint based approaches were proposed in [3, 78]. Direct and interactive manipulation tools of freeform curves and surfaces are investigated in [6].

In the field of Computer Graphics or Computer Aided Design, a very useful property for a given spline model is to have locally supported basis functions, in order to allow localized modifications of the shape. Unfortunately this property can also become a serious disadvantage when the user wishes to edit the global shape of a complex object. Piecewise polynomial B-spline representation is common in many contemporary geometric modeling systems. While this is a powerful tool with many desirable properties, the same properties impose some undesirable constraints on the user. For example, the most attractive property, *locality*, restricts the user to perform global operations on the object being modeled. To perform a global operation, it has to be transformed into a series of local operations affecting only a small portion of the curve, which makes the process time wasting and precision hazardous [23]. The ability to simultaneously perform both local and global operations at will would add significant functionality to any modeling system.

Multiresolution representation is a possible solution which addresses this problem, because it allows the user to edit objects at different resolution levels. Both local as well as global operations can be performed on curves by representing them using multiresolution decomposition. Several approaches have been proposed for multiresolution representation of splines, mostly based on wavelets. All these approaches involve expensive precalculations and in the case of open curves and surfaces, often require specific treatment of boundary control points. Moreover, these approaches depend on the given spline model they manipulate; the whole scheme has to be redefined when it comes to manipulating other spline models, only the philosophy of the calculus can potentially be reused [23].

All the approaches presented are either for the uniform B-splines or non-uniform B-splines. None of the previous work addresses the case of rational splines, which are the generalization of B-splines. Among the type of B-splines, NURBS (Non Uniform Rational B-Splines) have been receiving considerable attention in the areas of computer graphics and geometric modeling. In a very short period of time, NURBS are industry standard tools for the representation and design of geometry. The term NURBS given to it because they are defined on a knot vector where the interior knots spans are not equal.

### NURBS are useful because

- By manipulating the control points, knot vector and weights, NURBS provides
  the facility to design a large variety of shapes.
- They offer a common mathematical form for representing and designing both standard analytic shapes (conics, quadrics) and free form curves and surfaces.
- Evaluation is reasonably fast and computationally stable.
- NURBS have clear geometric tool kit (knot insertion/deletion, degree elevation etc.), which can be used to design, analyze, process and interrogate objects.

The main objective of this work is to propose a multiresolution representation for NURBS. This work is also oriented towards closely studying two of the proposed multiresolution representations, one based on wavelets for the uniform B-spline case, and the other for non-uniform B-splines.

### 1.1 Theory of Splines

Prior to the development of mathematical and computer models to support engineering design and manufacturing, descriptive geometry was used. Many of these geometric design techniques have been carried over into CAD design. The techniques for obtaining a mathematical curve model from digitized data are generally referred as curve fitting techniques. Cubic spline is an example of such technique. They are characterized by the fact that the mathematically derived curve passes through each and every data point. Another technique by which the mathematical descriptions of a space curve is generated without any prior knowledge of the curve shape or form. Bezier and B-spline are the examples of such technique. These techniques are frequently referred as curve fairing technique.

### 1.2 Spline Definitions

The Mathematical or Natural Spline is a piecewise polynomial of degree k with continuity of derivatives of order k-1 at common joints between segment.

### 1.3 A Bit of History

Back in the days before computers, architects, engineers, and artists would draw their designs for buildings, roads, machine parts, and the like by using pencil, paper, and various drafting tools. These tools included rulers and T-squares for drawing straight lines, compasses for drawing circles and circular arcs, and triangles and protractors for making precise angles.

Of course, a lot of interesting-shaped objects couldn't be drawn with just these simple tools, because they had curved parts that weren't just circles or ellipses. Often, a curve was needed that went smoothly through a number of predetermined points. This problem was particularly acute in shipbuilding: although a skilled artist or draftsman could reliably hand-draw such curves on a drafting table, shipbuilders often needed to make life-size (or nearly life-size) drawings, where the sheer size of the required curves made hand drawing impossible. Because of their great size, such drawings were often done in the loft area of a large building, by a specialist known as a loftsman. To aid in the task, the loftsman would employ long, thin, flexible strips of wood, plastic, or metal, called splines. The splines were held in place with lead weights, called ducks because of their resemblance to the feathered creature of the same name (see Figure 1.1).



Figure 1.1: A Draftsman Spline.

### 1.4 Types of Spline Curves

A spline curve is specified by a giving set of coordinates positions, called *Control Points*, indicating the shape of a curve. Spline curve is defined, modified and manipulated with operations on the control points. Control points are then fitted with piecewise continuous parametric polynomial functions in one of two ways.

The first type of splines is called *Approximatory Splines* in which the polynomials are fitted to the general control point path without necessarily passing through any control point and the resulting curve is said to approximate the set of control points (as shown in figure 1.2). These type of splines are used as design tools to structure object surfaces. Cubic spline Interpolation methods are often used to set up paths for object motions or to provide a representation for an existing object or drawing, and also to design object shapes.

The second type of splines is called *Interpolatory Splines* in which the curve passes through each control point and the resulting curve is said to interpolate the set of control points (as Shown in figure 1.3). They are used to digitize drawings, specify animation paths etc.

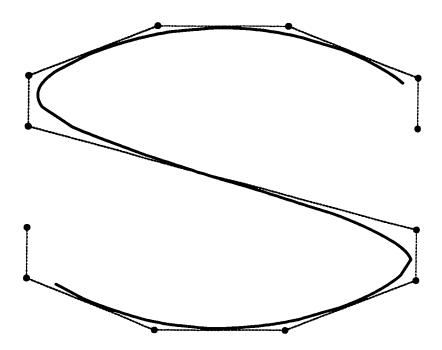


Figure 1.2: Curve Approximates the Control Points : An Approximatory Spline.

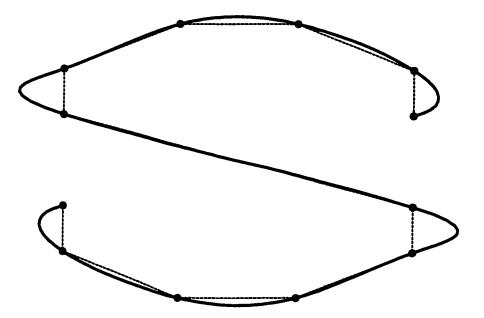


Figure 1.3: Curve Interpolates Through All Control Points: An Interpolatory Spline.

### 1.4.1 Approximatory Splines

### **Bezier Curves**

P. Bezier pioneered the Bezier Curve for computer modeling of surfaces for the design of automobiles. It interpolates the two end control points and approximates the other two. Mathematically a Bezier curves is defined as:

$$P(t) = \sum_{i=0}^{n} P_i B_{n, i}(t) \qquad 0 \le t \le 1$$
 (1.1)

The key of the Bezier method is the use of blending or basis functions. The Bezier (or Bernstein) basis function is

$$B_{n,i}(t) = \frac{n!}{i! (n-i)!} t^i (1-t)^{n-i}$$
 (1.2)

These affect the behavior of the curve from four control points. The four blending functions represent the 'influence' that each control point has on the curve. The major properties of Bezier curve are:

- Interpolates through first  $(P_0)$  and last  $(P_n)$  point.
- Non Negativity: For all i, p and t,  $B_{i,p}(t)$  is non-negative.
- Convex Hull Property: The curve always lies within the control polygon.
- Variation diminishing Property: No straight line intersects the curve more

times than it intersects the curves control polygon.

 Affine Invariance: The curves remains unchanged under various transormations like rotation, scaling etc.

There are several drawbacks in using this method. The blending functions affect all points along the curve. In other words, it does not have localized control over the curve so change in the postion of one control point will affect the whole shape of the curve as shown in figure 1.4. As we increase the weight value at  $P_5$ , the curve gets a pull and the result is the change in the overall shape of the curve as we can see for curves 1, 2, and 3. Also, the Bezier curve is a polynomial of degree one less than the number of control points. so, the number of control points affected the degree of the curve. The higher the degree, the higher the lack of control over the curve there was. This makes calculation of higher degree polynomial functions expensive in terms of computation.

### **B-Splines**

B-Spline curves were used to overcome the problems (such as global control and the relation between number of control points and degree of the curve) encountered by the Bezier curves, by providing a set of blending functions that only had effect on a few control points. This gave the local control that was lacking. Also, the problem of piecing curves together was avoided by allowing only those curves that possessed

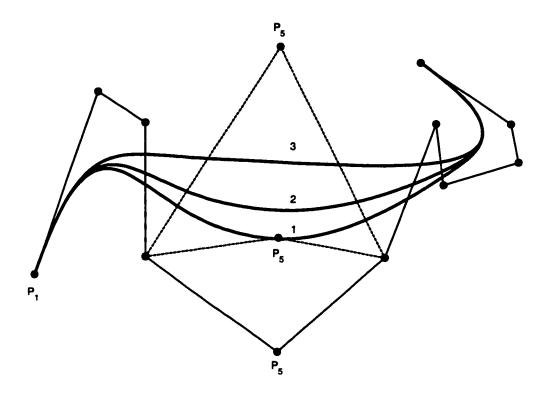


Figure 1.4: Global Control in Bezier Curves as a Result of Change in Position of Single Control Point.

the required continuity at the joints. Most other spline techniques provided this at the loss of local control.

There are three types of B-splines: Uniform Nonrational B-Splines, Non-uniform Nonrational B-Splines and Non-uniform Rational B-Splines [14].

The term uniform means that the joints (knots) are spaced at equal intervals of the parameter t. The term rational is used where x(t), y(t) and z(t) are each defined as the ratio of two polynomials.

Uniform Nonrational B-Splines: The B-splines curves were evolved in order to overcome the problems encountered by Bezier curves by providing a set of blending functions that only had effect on a few control points.

The general expression for the calculation of coordinate positions along a B-Spline curve in a blending function formulation is of the form:

$$P(t) = \sum_{i=0}^{n} P_i B_{i,p}(t) \qquad t_{min} \le t < t_{max}, \qquad 2 \le p \le n+1$$
 (1.3)

where  $P_i$  is an input set of n+1 control points and the B-Spline Blending functions  $B_{i,p}$  are polynomials of degree p.

The major properties of B-Spline curves are:

- B-spline curve P(t) is a piecewise curve with each component a degree p curve.
- Strong Convex Hull Property.

- P(t) is  $C^{p-k}$  continuous at a knot of multiplicity k.
- Variation Diminishing Property:
- Bézier Curves Are Special Cases of B-spline Curves .
- Affine Invariance
- If the standard knot vector is used, the B-Spline curve will interpolate the first
  and the last control points. Its initial and final directions are along the first
  and last edges of the control polygon respectively.

B-splines consists of curve segments whose polynomial coefficients depend on just a few control points, exhibiting local control, thereby moving a single control point affects only a small part of the curve. The time to compute the coefficients is greatly reduced and they have same continuity as natural splines. B-spline curves with different knot vectors are shown in figures 1.5 and 1.6.

The Cox Deboor [8] recursive formula for the B-Spline basis can be defined as:

$$B_{i,1}(t) = \left\{ egin{array}{ll} 1 & ext{if} & t_i \leq t < t_{i+1}; \ 0 & otherwise. \end{array} 
ight.$$

and

$$B_{i,p}(t) = \frac{(t-t_i)B_{i,p-1}(t)}{t_{i+p-1}-t_i} + \frac{(t_{i+p}-t)B_{i+1,p-1}(t)}{t_{i+p}-t_{i+1}}$$
(1.4)

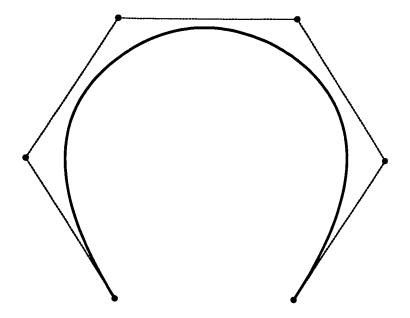


Figure 1.5: A B-spline Curve with Open Uniform Knot Vector.

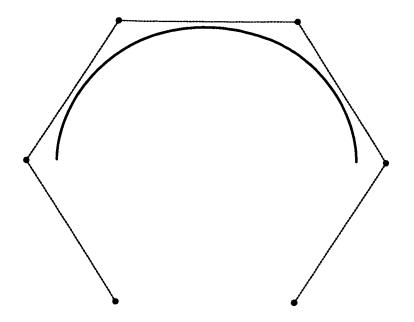


Figure 1.6: A B-Spline Curve with Periodic Knot Vector.

The major properties of the B-Spline Basis function  $B_{i,d}$  are:

- $B_{i,p}(t)$  is a degree p polynomial in t.
- Nonnegativity For all i, p and t,  $B_{i,p}(t)$  is non-negative.
- Local Support  $B_{i,p}(t)$  is a non-zero polynomial on  $[t_i, t_{i+p+1}]$ .
- On any span  $[t_i, t_{i+1}]$ , at most p+1 degree p basis functions are non-zero, namely:  $B_{i-p,p}(t), B_{i-p+1,p}(t), B_{i-p+2,p}(t), \ldots$ , and  $B_{i,p}(t)$ .
- Partition of Unity The sum of all non-zero degree p basis functions on span  $[t_i, t_{i+1}]$  is 1.
- If the number of knots is m+1, the degree of the basis functions is p, and the number of degree d basis functions is n+1, then m=n+p+1.
- At a knot of multiplicity k, basis function  $B_{i,p}(t)$  is  $C^{p-k}$  continuous.

Figure 1.7 shows the basis funcions of B-spline curve with five control points.

The important feature of the B-Spline blending functions is that they are non-zero in only a small portion of the range of the particular parameter. B-Spline shares many of the advantages of Bezier Curves, but the main advantage is the local control of the curve shape as shown in figure 1.8. In other words, moving a control point affects only in the region near the control point of a curve. In addition, the time needed to compute the coefficients is greatly reduced. B-splines have the same

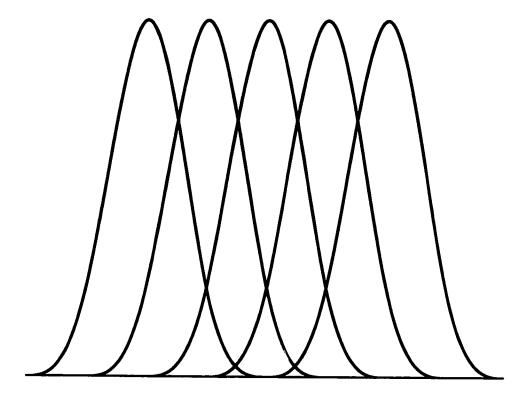


Figure 1.7: B-Spline Curve Basis Functions.

can be added at will without increasing the degree of the curve, thereby retaining the control over the curve that would be lost with a Bezier curve.

Nonuniform Nonrational B-Splines: Nonuniform nonrational B-splines permit unequal spacing between the knots. These curves have several advantages over uniform B-splines.

First, continuity at selected join points can be reduced from second derivative  $(C^2)$  to first derivative  $(C^1)$  to  $C^0$  to none. If the continuity is reduced to  $C^0$ , then the curve interpolates a control point, but without the undesirable effect of uniform B-splines, where the curve segments on either side of the interpolated control point are straight lines. Also, starting and ending points can be easily interpolated exactly without introducing line at segments at the same time.

It is possible to add an additional knot and control point to nonuniform B-splines, so the resulting curve can be easily reshaped, whereas this cannot be done with uniform B-splines.

Nonuniform Rational B-Splines: NURBS are Non-Uniform Rational B-Splines and is the term given to curves that are defined on a knot vector where the interior knot spans are not equal. As an example, we may have interior knots with spans of zero. Some common curves require this type of non-uniform knot spacing. The use of this option allows better shape control and the ability to model a larger class of

shapes.

Given a set of n+1 control points  $P_1, P_2, ..., P_n$ , each of which is associated with a non-negative weight  $w_i$  (i.e.,  $P_i$  has weight  $w_i \ge 0$ ), and a knot vector  $t = \{t_0, t_1, ..., t_m\}$  of m+1 knots, the degree p NURBS curve is defined as follows:

$$P(t) = \sum_{i=0}^{n} P_i R_{i,p}(t)$$
 (1.5)

Where

$$R_{i,p}(t) = \frac{B_{i,p}(t) w_i}{\sum_{j=0}^{n} B_{j,p}(t) w_j}$$
(1.6)

The knot vector uniquely determines the B-spline as it is obvious from figure 1.7. The relation between the number of knots (m + 1), the degree (p) of  $B_{i,p}$  and the number of control points (n + 1) is given by m = n + p + 1.

Shape of NURBS not only depends on the control points but also on the weight parameter associated with each control point. By changing the weight  $w_i$  of a control point  $P_i$  affects only the range  $t_i$ ,  $t_{i+p+1}$ .

### 1.4.2 Interpolatory Splines

Natural Cubic Splines, Hermite Splines, Cardinal Splines, and Kochanek-Bartels Splines come under the category of Interpolatory splines. For further discussion on Interpolatory splines, please refer [4].

# 1.5 Some Other Splines

A large number of spline methods exist in literature. Some of them are useful to achieve one objective and others are useful to achieve other objective. For brevity, reader is referred to [74, 70, 61, 57, 58, 17, 62, 66, 56, 30, 44, 34, 59, 60, 54, 55, 51, 53, 52, 49, 50, 48, 47, 16, 20, 73, 46, 72, 71, 75, 31, 40, 35, 37, 36, 45, 43, 21, 69, 39, 64, 38, 67, 68, 65, 41, 42, 63].

# 1.6 Objective

The work presented here aims at the following aspects.

- Study of the already proposed multiresolution representation methods for splines.
- 2. Coming up with a multiresolution representation for NURBS.
- 3. Application of the proposed method on NURBS curves and surfaces.
- 4. NURBS curve editing by using the proposed methodology.
- 5. Comparison of the proposed work with some of the previously proposed methods.

# 1.7 Thesis Organization

Chapter 2 is about Non-uniform Rational B-Splines (NURBS). It covers the theoretical details and the properties of NURBS which make them superior to the other parametric curves. Chapter 3 addresses two of the previously proposed multiresolution representation methods for B-splines. Chapter 4 deals with the proposed method for the multiresolution representation of NURBS, it also covers the application of proposed method by using it for editing the NURBS curves. Chapter 5 gives a brief comparative study of proposed method and previous works. Chapter 6 concludes our work and gives suggestions for possible future directions.

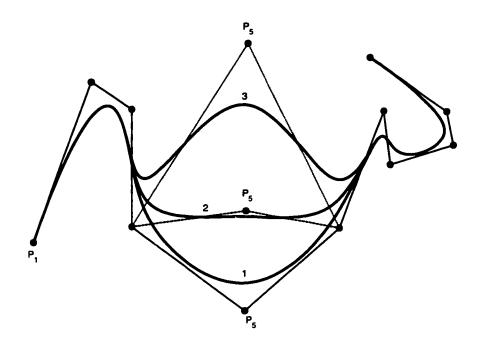


Figure 1.8: Local Control in B-spline Curves.

# Chapter 2

# **NURBS**

### 2.1 Introduction

B-spline curves are polynomial curves. While B-spline curves are flexible and have many nice properties for curve design, they are not able to represent the simplest of curves like the Circle, Ellipse, and many other curves that cannot be represented by polynomials. To cope with this inability, an extension to B-splines is needed. This motivates for the generalization of B-splines to rational curves using homogeneous coordinates. Therefor the name Non-Uniform Rational B-Splines (NURBS).

The major ingredients of NURBS are Non Uniform knots, Rational and B Splines. Rational curve and surfaces was first introduced into computer graphics literature by Coons around 1967. He also suggested their usage to represent conic sections. Forrest gives a rigorous treatment of rational conics and cubics. Following

the work done by De Boor, Cox and Riesenfeld, Versprille, in 1975 proposed the Non uniform Rational B splines or NURBS. After that, Piegl and Tiller form the basis of the current implementation. The NURBS shape representation for geometric design generalized Riesenfeld's B-Splines. NURBS quickly gained popularity and were incorporated into several commercial modeling systems like IEGS, PHIGS+, Product Data Exchange Specification, International Standard Office Standard for the Exchange of Product Model Data [33]. It offers a unified mathematical formulation for representing not only free-form curves and surfaces, but also standard analytic shapes such as conics, quadrics, and surfaces of revolution. By adjusting the positions of control points and manipulating associated weights, one can design a large variety of shapes using NURBS.

NURBS are defined on a knot vector where the interior knot spans are not equal.

As an example, we may have interior knots with spans of zero. Some common curves require this type of non-uniform knot spacing. The use of this option allows better shape control and the ability to model a larger class of shapes.

### 2.2 Mathematical Definition

Given a set of n+1 control points  $P_1, P_2, ..., P_n$ , each of which is associated with a non-negative weight  $w_i$  (i.e.,  $P_i$  has weight  $w_i \ge 0$ ), and a knot vector  $t = \{t_0, t_1, ..., t_m\}$  of m+1 knots, the degree p NURBS curve is defined as follows:

$$P(t) = \sum_{i=0}^{n} P_i R_{i, p}(t)$$
 (2.1)

where  $R_{i, p}(t)$  is defined as:

$$R_{i, p}(t) = \frac{B_{i, p}(t) w_i}{\sum_{i=0}^{n} B_{i, p}(t) w_i}$$
(2.2)

 $P_i$ 's are the control points (forming a control polygon), the  $w_i$  are the weights, and  $B_{i,p}(t)$  are the pth-degree B-spline basis functions.

The  $R_{i,p}(t)$  are the rational basis functions; they are piecewise rational functions on  $t \in [0,1]$ .

The  $i^{th}$  normalized B-spline basis function or order p+1 (degree p) is defined by the Cox-deBoor recursion formulas.

$$B_{i,1}(t) = \left\{ egin{array}{ll} 1 & ext{if} & t_i \leq t < t_{i+1}; \ 0 & ext{otherwise}. \end{array} 
ight.$$

and

$$B_{i,p}(t) = \frac{(t-t_i)B_{i,p-1}(t)}{t_{i+p}-t_i} + \frac{(t_{i+p+1}-t)B_{i+1,p}(t)}{t_{i+p+1}-t_{i+1}}$$
(2.3)

### 2.3 Basis Functions

The function  $B_{i,p}(t)$ , which determines how strongly control point  $P_i$  influences the curve at time t, is called the basis function for that control point. There are two interesting properties of Basis Functions, namely:

- The domain is subdivided by the so called knots.
- Basis functions are not non-zero on the entire interval.

In fact, each B-spline basis function is non-zero on a few adjacent subintervals and, as a result, B-spline basis functions are quite "local".

Figure 2.1 shows the B-spline basis functions.

#### 2.3.1 Knots and Knot Vector

In figure 2.1, all the basis functions have same shape and cover equal interval of time. In order to change the effect of different control points on smaller or larger portions, we vary the width of the intervals, so that the non-uniformity comes into picture (hence the name Non Uniform). We define a series of points that partition the time into intervals, which can be used in basis functions to achieve the desired result. By varying the relative lengths of the intervals, we can vary the amount of time each control point affects the curve. The points demarcating the intervals are known as *knots*, and the ordered list of them is a *knot vector*.

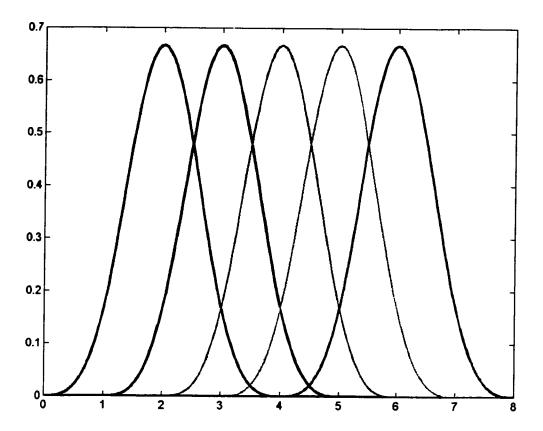


Figure 2.1: B-Spline Curve Basis Functions.

Let t be a set of m+1 non-decreasing real numbers,  $t_0 \le t_1 \le t_2 \le \ldots \le t_m$ . The  $t_i$ 's are called knots, the set t the knot vector, and the half open interval;  $[t_i, t_{i+1}]$  the  $i^{th}$  knot span. Note that since some  $t_i$ 's may be equal, some spans may not exist. If a knot  $t_i$  appears k times, where k > 1,  $t_i$  is a multiple knot of multiplicity k, written as  $t_i(k)$ . Otherwise if  $t_i$  appears only once, it is a simple knot. If the knots are equally spaced (i.e.,  $t_{i+1} - t_i$  is a constant for  $0 \le i \le m-1$ ), the knot vector or the knot sequence is called uniform otherwise it is non-uniform.

# 2.3.2 Important Properties of NURBS Basis Functions

Since NURBS is the generalization of B-spline, it enjoys all the properties of B-spline. Following are some of the main properties of NURBS Basis Functions.

- $R_{i,p}(t)$  is a degree p rational function in t.
- Non-negativity- for all i, P and  $t, R_{i,p}(t)$  is non-negative.
- local Support-  $N_{i,p}(t)$  is a non-zero polynomial on  $[t_i, t_{i+p+1}]$ .
- Since  $N_{i,p}(t)$  is non-zero on  $[t_i, t_{i+p+1}]$ , so does  $R_{i,p}(t)$ . Note that we assume  $w_i s$  are non-negative.
- On any span  $[t_i, t_{i+1}]$ , at most p+1 degree p basis functions are non-zero, namely:  $N_{i-1,p}(t), N_{i-p+1,p}(t), N_{i-p+2,p}(t), \dots$  and  $N_{i,p}(t)$ .

- Partition of Unity- The sum of all non-zero degree p basis functions on span  $[t_i, t_{i+1}]$  is 1.
- If the number of knots is m+1, the degree of the basis functions is p, and the number of degree p basis functions is n+1, then m=n+p+1.
- At a knot multiplicity of k, basis function  $N_{i,p}(t)$  is  $C^{p-k}$  continuous. Therefore, increasing multiplicity decreases the level of continuity, and increasing degree increases continuity.
- Basis function  $R_{i,p}(t)$  is a composite curve of degree p rational functions with joining points at knots in  $[t_i, t_{i+p+1}]$ .
- If  $w_i = c$  for all i, where c is a non-zero constant,  $R_{i,p}(t) = N_{i,p}(t)$ .
- Therefore, B-spline basis functions are special cases of NURBS basis functions when all weights become a non-zero constant. We have mentioned the special case of c=1.

# 2.4 Important Properties of NURBS Curves

The following are the important properties of NURBS curves.

• NURBS curve C(t) is a piecewise curve with each component a degree p rational curve.

- Equality m = n + p + 1 must be satisfied.
- A clamped NURBS curve C(t) passes through two end control points  $p_0$  and  $p_n$ .
- Strong Convex Hull Property: The NURBS curve is contained in the convex hull of its control polyline. More over, if t is in knot span  $[t_i, t_{i+1}]$ , then p(t) is in the convex hull of control points  $p_{i-p}, p_{i-p+1}, ..., p_i$ .
- Local Modification Scheme: Changing the position of control point  $p_i$  only affects the curve p(t) on interval  $[t_i, t_{i+1}]$ . This local modification scheme is very important to curve design, because we can modify a curve locally without changing the shape in a global way.
- p(t) is  $C^{p-k}$  continuous at a knot of multiplicity k.
- Variation Diminishing Property: The variation diminishing property also holds
  for NURBS curves. If the curve is contained in a plane (resp., space), this mean
  no straight line (resp., plane) intersects a NURBS curve more than it intersects
  the curve's control polyline (see Figure 2.2).
- B-spline Curves and Bezier are special cases of NURBS Curves.
- If all weights are equal, a NURBS curve becomes a B-spline curve. If further more n = p (i.e., the degree of a B-spline curve is equal to n, the number of

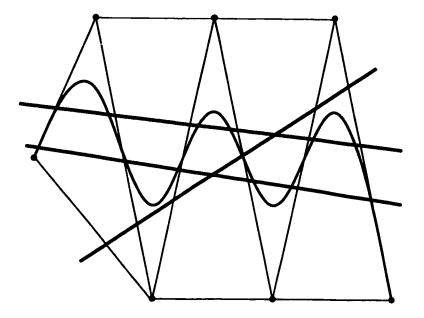


Figure 2.2: Variation Diminishing Property in NURBS. control points minus 1) and there are 2(p+1)=2(n+1) knots with p+1 of them clamped at each end, this NURBS curve reduces to a Bezier curve.

• Projective Invariance: If a projective transformation is applied to a NURBS curve, the result can be constructed from the projective images of its control points. This is a nice property. When we want to apply a geometric or even projective transformation to a NURBS curve, this property guarantees that we can apply transformation to control points, which is quite easy, and once the transformed control points are available the transformed NURBS curve is the one defined by these transformed points. Therefore, we do not have to transform the curve. On the other hand, Bezier curves and B-spline curves

only satisfy the affine invariance property. This is because only NURBS curves involve projective transformations.

# 2.5 Effect of Modifying Weights on NURBS

Since NURBS curves contain B-spline curves as special cases, methods for modifying the shape of a B-spline curve such as moving control points and modifying knot vector also work for NURBS. NURBS curves are defined with a set of control points, a knot vector, a degree and a set of weights, we have on e more parameter for modifying the shape of a NURBS curve, the weights.

The basis functions of a NURBS curve has been shown in Fig.2.1. Therefore, increasing and decreasing the value of  $w_i$  will increase and decrease the value of  $R_{i,p}(t)$ , respectively. By changing the weight  $w_i$  of a control point  $P_i$  affects only the range  $[t_i, t_{i+p+1}]$ . More precisely, increasing the value of  $w_i$  will pull the curve toward the control point  $P_i$ . In fact all affected points on the curve will also be pulled in the direction of  $P_i$ . When  $w_i$  approaches infinity, the curve will pass through control point  $P_i$ . On the other hand, decreasing the value of  $w_i$  will push the curve away form control point  $P_i$ .

A Simple NURBS curve is shown in Figure 2.3 and Figures 2.4 and 2.5 show NURBS curve with varying weights at a control point.

In summary, we can say that "Increasing (resp., decreasing) the value of weight

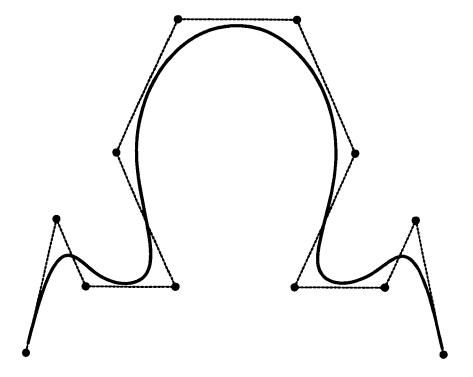


Figure 2.3: NURBS Curve with Default Shape Parameters.

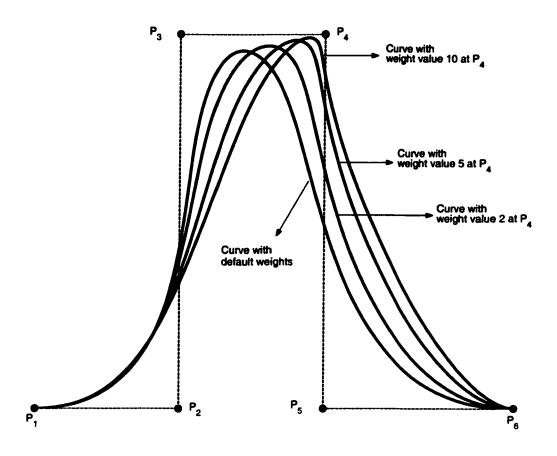


Figure 2.4: A NURBS curve with application of different weights at  $P_4$ .

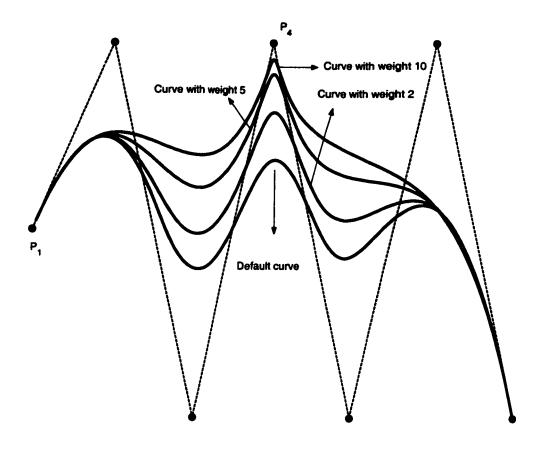


Figure 2.5: Application of Different Weight Values on  $P_4$  and its Effect on the Shape of the Curve.

 $w_i$  pulls (resp., pushes) the curve toward (resp., away from) control point  $P_i$ . When the value of of  $w_i$  becomes infinity, the curve passes through control point  $P_i$  and when  $w_i$  is zero, control point  $P_i$  does not have impact on the curve".

Non-uniform Rational B-splines(NURBS) are useful for two reasons. The first and the most important reason is that they are invariant under perspective transformations of the control points. The second advantage of rational splines is that they define precisely any of the conic sections. A conic section can only be approximated with rational, by using many control points close to the conic.

# Chapter 3

# Multiresolution Representation of

# **Splines**

In the field of Computer Graphics or Computer Aided Design, a very useful property for a given spline model is to have locally supported basis functions, in order to allow localized modifications of the shape. Unfortunately this property can also become a serious disadvantage when the user wishes to edit the global shape of a complex object. Piecewise polynomial B-spline representation is common in many contemporary geometric modeling systems. While this is a powerful tool with many desirable properties, the same properties impose some undesirable constraints on the user. For example, the most attractive property, *locality*, restricts the user to perform global operations on the object being modeled. To perform a global operation, it has to be transformed into a series of local operations affecting only a small portion

of the curve, which makes the process time wasting and precision hazardous [23]. The ability to simultaneously perform both local and global operations at will would add significant functionality to any modeling system.

Multiresolution representation is a possible solution which addresses this problem, because it allows the user to edit objects at different resolution levels. Both
local as well as global operations can be performed on curves by representing them
using multiresolution decomposition. Multiresolution analysis can be defined as an
ability to simultaneously perform both local and global operations on the analyzed
object[9]. Several approaches have been proposed for multiresolution representation of splines, mostly based on wavelets. All these approaches involve expensive
precalculations and in the case of open curves and surfaces, often require specific
treatment of boundary control points. Moreover, these approaches depend on the
given spline model they manipulate; the whole scheme has to be redefined when it
comes to manipulating other spline models, only the philosophy of the calculus can
potentially be reused [23].

There are many applications of multiresolution curves, including [13]:

- 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.

Two of the approaches presented are studied in detail. One of them uses B-spline wavelets for endpoint interpolating B-splines. Another approach which deals with the multiresolution control on Non-uniform B-splines for the purpose of curve editing is presented, which uses the knot decimation and least squares approximation. This chapter explores both of these approaches in detail for the sake of completeness.

# 3.1 Background Material on Wavelets

Prior to discussing the wavelet methods for multiresolution analysis, it is desirable to discuss about wavelets, which show the usefulness of wavelets for these purposes.

#### 3.1.1 Introduction to Wavelets

Wavelets are a mathematical tool for hierarchically decomposing functions. Using wavelets one can describe a function in terms of a coarse overall shape, and the details that range from broad to narrow. Wavelets offer an elegant technique for representing the levels of details present in a function, regardless of whether the function of interest is an image, a curve, or a surface [12].

The roots of wavelets are found in approximation theory, signal processing, and physics; they have also been applied to many problems in computer graphics. These applications include image editing and compression, automatic level-of-detail control for editing and rendering curves and surfaces, surface reconstruction from contours, and fast methods for solving simulation problems in 3D modeling, global illumination, and animation [10].

In every scientific and engineering discipline there is a need to analyze, visualize and manipulate large quantities of data. The data in these applications may have many forms, like the functions of a single parameter, such as the time series data used in signal processing, or complex cross sectional contours encountered in medical applications. The data may also be functions of two or more parameters, such as two-dimensional images, surface models, or higher dimensional "global illumination" solutions for photorealistic lighting.

In all of these cases, the simplest way to represent the information is with a se-

quence of points. For example, a time series is most easily represented as a sequence (ti, yi). Each such point provides complete information about the behavior of the series at ti but absolutely no information about the behavior of the series elsewhere. In contrast, many applications require an analysis of the series at broader scales; for example, a region of rapid change in the series can only be detected by examining many points at once.

The classic tool for addressing these issues is Fourier analysis, which can be used to convert point data into a form that is useful for analyzing frequencies. A difficulty with Fourier technique is that each Fourier coefficient contains complete information about the behavior of the series at one frequency but no information about its behavior at other frequencies. Fourier techniques are also difficult to adapt to many situations of practical importance; for instance, most of the time series encountered in practice are finite and aperiodic, but the Fourier representation is clearly not appropriate for aperiodic or non-stationary functions (whose spectral content change in time).

For example consider the following signal;

$$x(t) = \cos(2\pi.10.t) + \cos(2\pi.25.t) + \cos(2\pi.50.t) + \cos(2\pi.100.t)$$
 (3.1)

it has the frequency components of 10, 25, 50, and 100 Hz at any given time instant. Its plot is given in figure 3.1.

If we look at the Fourier analysis of this signal (figure 3.2), it contains four peaks, which are the Fourier coefficients corresponding to the four frequencies contained in the signal.

Now looking at another example (figure 3.3), this signal also contains the same four frequency components but at four different time intervals (It is called a non-stationary or aperiodic). The interval 0 to 300ms (millisecond) has a 100Hz sinusoid, the interval 300 to 600ms has a 50Hz sinusoid, the interval 600 to 800ms has a 25Hz sinusoid, and finally the interval 800 to 1000ms has a 10Hz sinusoid. Figure 3.4 shows its Fourier transform (FT).

The little ripples in figure 3.4 are due to sudden changes from one frequency component to another, which have no significance at present. Note that the amplitudes of higher frequency components (100 and 50Hz) are higher than those of the lower frequency (25 and 10Hz) ones. This is due to fact that higher frequencies last longer (300 ms each) than the lower frequency components (200 ms each). (The exact values of the amplitudes are also not important here.)

Other than those ripples, everything in figures 3.2 and 3.4 seems to be same, i.e., the discrete FT has four peaks, corresponding to four frequencies with reasonable amplitudes. But there is a significant difference in the signals of figure 3.1 and 3.3. To observe this difference let us answer the following question for both the signals [28].

At what time intervals, do these frequency (i.e., 10, 25, 50, and 100Hz) compo-

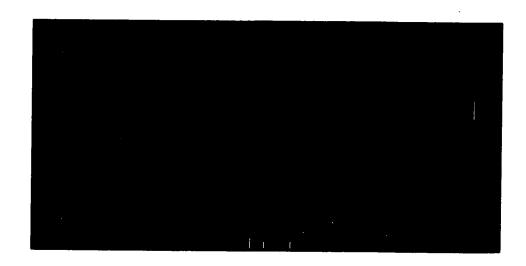


Figure 3.1: Signal x(t) with 10, 25, 50, and 100Hz Frequencies.

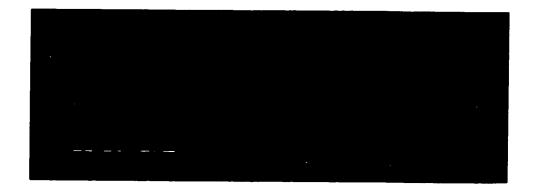


Figure 3.2: Fourier Transform of the Signal x(t) of figure 3.1.

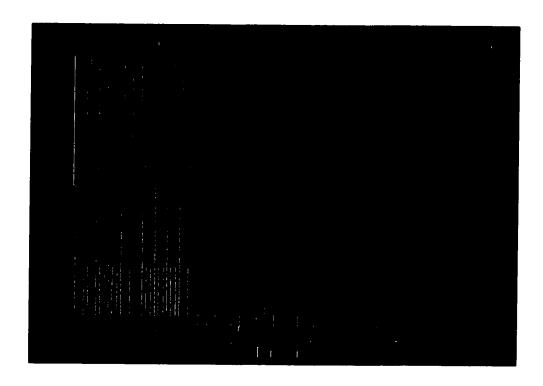


Figure 3.3: A Non-Stationary signal with 10, 25, 50, and 100Hz Frequencies.

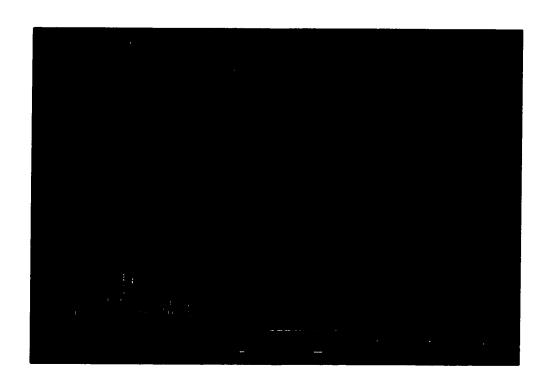


Figure 3.4: Fourier Transform of the Signal of figure 3.3.

#### nents occur?

For the first signal of figure 3.1, the answer to this question is at all times i.e., all the four frequencies exist in the interval 0 to 1000ms. However, for the second signal in figure 3.3 these components exist for different intervals of time. The similarity between figures 3.2 and 3.4 is apparent; both of them show four spectral components at exactly the same frequencies, i.e., the two spectra are almost identical, although the corresponding time-domain signals are not even close to each other. Both signals involve the same frequency components, but the first one has these frequencies at all times, the second one has these frequencies at different intervals. So, how come the spectrums of two entirely different signals look very much alike?

The inability of the Fourier techniques in distinguishing between stationary and non-stationary signals and to provide the time-frequency localization information arises the need for a better representation of signals. This has led researchers in a variety of disciplines (including approximation theory, physics, signal and image processing, as well as computer graphics) to develop various hierarchical representations of the functions [10]. The basic idea behind all hierarchical methods called as multiresolution methods is to represent functions with a collection of coefficients, each of which provides some limited information about both the position and frequency of the function.

Although there are a wide variety of methods for representing a function in a hierarchical fashion [10], the theory of wavelets provide an extremely useful mathe-

matical toolkit for hierarchically decomposing functions in ways that are both efficient and theoretically sound [10]. Broadly speaking, a wavelet representation of a function consists of a coarse overall approximation together with detail coefficients that influence the function at various scales [12, 10].

#### **History of Wavelets**

Wavelets have recently become popular, however the roots of wavelets go back at least a century to the work of Karl Weierstrass [77], who in 1873 described a family of functions that are constructed by superimposing scaled copies of a given basis function [10]. Another important early milestone was Alfred Haar's construction of the first orthonormal system of compactly supported functions in 1909, called the Haar basis [19]. The Haar basis still serves as the foundation of modern wavelet theory. In 1946 another significant advancement came in, when Dennis Gabor [15] described a non-orthogonal basis of wavelets with unbounded support, based on translated Gaussians.

The term wavelet came from the field of seismology, where it was coined by Ricker in 1940 to describe the disturbance that proceeds outwards from a sharp seismic impulse or explosive charge [32]. In 1982 Morlet et al. [22] showed how these seismic wavelets could be effectively modeled with the mathematical functions that Gabor has defined. Later, Grossman and Morlet [18] showed how arbitrary signals could be analyzed in terms of scales and translates of a signal mother wavelet function.

Yves Meyer [26, 27] and Stephane Mallat [24] developed this notion into a theory called multiresolution analysis. In 1989 Mallat [25] showed how multiresolution analysis could be viewed as just another form of the pyramid algorithms used in image processing [2] and the quadrature mirror filters used in signal analysis [1, 76].

### 3.1.2 Mathematical Description of Wavelet Transform

Mathematically, the Fourier Transform is represented as:

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-2j\pi ft} dt$$
 (3.2)

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{2j\pi ft} df$$
 (3.3)

x(t) is the original signal in time domain and X(f) is the transformed signal in frequency domain. The Fourier transform tells whether a certain frequency component exists or not. It does not provide information regarding where in time these spectral components appear.

Short Time Fourier Transform (STFT) is proposed which allows analysis of nonstationary signals by segmenting them into stationary enough short pieces and then computing the FT of each piece. Mathematically it is given as:

$$STFT_x^{(w)}(t,f) = \int_t [x(t) \ w^*(t-t')] \ e^{-2j\pi ft} \ dt \tag{3.4}$$

.

w(t) is the window function, and \* is the complex conjugate. As it is clear from the above equation that STFT of the signal is nothing but its FT multiplied by a window function. The problem with this approach is that it provides constant resolutions for all frequencies, since it uses the same window for the analysis of entire signal.

To alleviate this problem *Wavelet Transform* (WT) provides varying time and frequency resolutions by using windows of different lengths. It is represented mathematically as follows:

$$W(\tau,s) = \frac{1}{\sqrt{s}} \int x(t) \ \psi^*(\frac{t-\tau}{s}) \ dt \tag{3.5}$$

s and  $\tau$  are scale and translation parameters respectively, and  $\psi$  is the mother wavelet.

# 3.1.3 Important Properties of Wavelets

Wavelets representations are beginning to profoundly affect most of the areas of the computer graphics, due to many useful properties associated with them. In addition to hierarchical nature of the wavelets, these properties include:

• Linear Time Complexity: Transforming to and from a wavelet representation can generally be accomplished in linear time, which results in very fast algorithms.

- Sparsity: For functions typically encountered in practice, many of the coefficients in a wavelet representation are either zero or negligibly small. This property offers the opportunity both to compress data and to accelerate the convergence of iterative solution techniques.
- Adaptability: Wavelets are remarkably flexible, unlike Fourier techniques, that they can be applied to represent a wide variety of functions, including functions with discontinuities, functions defined on bounded domains, and functions defined on domains of arbitrary topological type. Consequently, wavelets are equally suited to problems involving images, open or closed curves, and surfaces of just about any variety.

The form of multiresolution analysis, proposed by Meyer and Mallat decomposes signal onto a set of basis functions, called wavelets, in which every wavelet is just a scaled and translated copy of a single unique function, called the mother wavelet, shown in figure 3.5. This approach can be called as shift-invariant theory, since the wavelets lying on different parts of the unbounded real line all look the same. The shift-invariant approach, being remarkably beautiful from theoretical standpoint, is problematic for most computer graphics applications since many functions of interest, such as images or open curves, are defined only on some bounded portion [10].



Figure 3.5: Mother Wavelet.

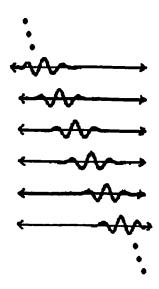


Figure 3.6: Shift-Invariant Wavelets.

A more generalized version, which accommodates more naturally the kind of finite data sets encountered in practical computer graphics applications can be used. Unlike classical theory of multiresolution analysis, the generalized version is shift-variant. It accommodates bounded data sets by introducing different, specially adapted wavelets near the boundaries as shown in figure 3.7.

The shift-variant multiresolution analysis is very closely related to the theory of recursive subdivision. Indeed the functions to which shift-variant multiresolution analysis can be applied turn out to be exactly those functions that can be generated through a subdivision process [10].

# 3.1.4 Wavelets and Multiresolution Analysis

To understand the basic ideas behind wavelets and multiresolution analysis, consider a discrete signal  $C^n$ , expressed as column vector of samples. The samples could be the control points of the curve. If we want to create a low resolution version  $C^{n-1}$  of  $C^n$  with a fewer number of samples m'. The approach is to use some form of filtering and downsampling on m samples of  $C^n$ . This process can be expressed as the matrix equation;

$$C^{n-1} = A^n C^n (3.6)$$

where An is  $m' \times m$  matrix.

Since  $C^{n-1}$  contains fewer samples than  $C^n$ , it is clear that some amount of details is lost in the filtering process. If  $A^n$  is appropriately chosen, it is possible to capture the lost details as another signal  $D^{n-1}$ , given as;

$$D^{n-1} = B^n C^n (3.7)$$

where  $B^n$  is  $(m - m') \times m$  matrix, which is related to matrix  $A^n$ .

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

If  $A^n$  and  $B^n$  are chosen correctly, 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  $Q^n$  as;

$$C^{n} = P^{n} C^{n-1} + Q^{n} D^{n-1}$$
(3.8)

The recovery process of  $C^n$  from  $C^{n-1}$  and  $D^{n-1}$  is called reconstruction, and the pair of matrices  $P^n$  and  $Q^n$  are called synthesis filters.

The procedure of 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 decomposed as a hierarchy of low-resolution signals  $C^0$ ,  $C^1$ , ...,  $C^{n-1}$  and details  $D^0$ ,  $D^1$ , ...,  $D^{n-1}$ , this recursive process is shown in figure 3.8, and is known as a filter bank [13, 10, 11].

Since the original signal  $C^n$  can be recovered from the sequence  $C^0$ ,  $D^0$ ,  $D^1$ , ...,

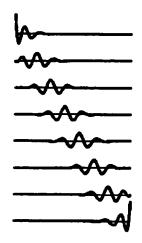


Figure 3.7: Shift-Variant Wavelets.

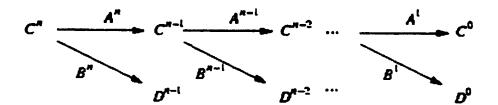


Figure 3.8: The Filter Bank.

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

Many properties of wavelet transforms make them attractive for signal processing.

- Firstly, if the analysis and synthesis filters are constructed to be sparse, the filter bank operation can be performed very fast, often in O(m).
- Second, for many of the signal that are available in practice, a large percentage
  of the entries in the wavelet transform are negligible.

Wavelet compression methods can therefore approximate the original set of samples in  $C^n$  by storing only the significant coefficients of the wavelet transform.

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

$$f^n(u) = \Phi^n(u) C^n \tag{3.9}$$

Where  $\Phi^n(u)$  is a row matrix of basis functions  $[\phi_1^n(u), \phi_2^n(u), \dots, \phi_m^n(u)]$  called scaling functions.

The scaling functions are required to be refinable; i.e., for all j in [1, n] a matrix  $P^{j}$  must exist such that:

$$\Phi^{j-1} = \Phi^j P^j \tag{3.10}$$

each scaling function at level j-1 must be expressible as a linear combination of finer scaling functions at level j.

Now, next, let  $V^j$  be the linear space spanned by the set of scaling functions  $\Phi^j$ . From the equation 3.10, it is implied that these spaces are nested, i.e.,  $V^0 \subset V^1 \subset \ldots \subset V^n$ .

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  $\Psi^j = [\psi^j_1(u), \psi^j_2(u), \dots, \psi^j_{m-m'}(u)]$  are such that  $\Phi^j$  and  $\Psi^j$  together form a basis for  $V^{j+1}$ , and every  $\psi^j_i(u)$  is orthogonal to every  $\phi^j_i(u)$  under the chosen inner product. The basis functions  $\psi^j_i(u)$  are called wavelets.

The synthesis filter  $Q^j$  can be constructed as the matrix that satisfies;

$$\Psi^{j-1} = \Phi^j Q^j \tag{3.11}$$

Equations 3.10 and 3.11 can be expressed as a single equation by concatenating the matrices together:

$$[\Phi^{j-1} \mid \Psi^{j-1}] = \Phi^{j} [P^{j} \mid Q^{j}]$$
 (3.12)

The analysis filters  $A^{j}$  and  $B^{j}$  are formed by the matrices satisfying the inverse relation:

$$[\Phi^{j-1} \mid \Psi^{j-1}] \left[ \frac{A^j}{B^j} \right] = \Phi^j \tag{3.13}$$

The matrices  $[P^j \mid Q^j]$  and  $[A^j \mid B^j]^T$  are both square matrices, thus;

$$\left[\frac{A^{j}}{B^{j}}\right] = \left[P^{j} \mid Q^{j}\right]^{-1} \tag{3.14}$$

## 3.2 Multiresolution Decomposition of End-point Interpolating B-splines Using Wavelets

#### 3.2.1 Construction of B-spline Wavelets

A multiresolution analysis for B-spline curves (specially cubic B-splines) is presented in [13, 10, 11]. The B-splines used were defined on uniform knot sequence everywhere except at ends, where its knots have multiplicity 4.

To construct multiresolution framework from endpoint-interpolating cubic B-splines, the following choices are made [13, 10].

1. Choosing the scaling functions  $\Phi^j(u)$  for all j in [0, n]: This choice determines the synthesis filters  $P^j$ . For each level j, a basis for the end-point interpolating cubic B-spline curve with  $2^j$  interior segments. The basis functions for these

curves are the  $2^{j}+3$  endpoint-interpolating cubic B-splines, which are refinable, as required by equation 3.10.

2. Selection of an inner product for any two functions f and g in  $V^j$ : This choice determines the orthogonal complement spaces  $W^j$ . The standard form of Inner product is:

$$\langle f, g \rangle = \int f(u) g(u) du \qquad (3.15)$$

- 3. Selection of a set of wavelets  $\Phi^j(u)$  that span  $W^j$ : This choice determines the synthesis filters  $Q^j$ .
- 4. Finding the Analysis Filters: Together, the synthesis filters  $P^j$  and  $Q^j$  determine the analysis filters  $A^j$  and  $B^j$ .

Figure 3.9 shows the scaling functions and wavelets for the above construction for j = 3.

#### 3.2.2 Application to Curves and Surfaces

After the development of the multiresolution framework, it can be used in various applications for curves and surfaces. This section shows some of the applications of it.

#### **Smoothing**

The problem here can be defined as: Given a curve  $C^n$  with m control points, construct a best least-squares-error approximating curve  $C^{n'}$  with m' control points (m' < m and n' < n). Using the multiresolution analysis it can be trivially solved as:

$$C^{n'} = A^{n'+1} A^{n'+1} \dots A^n C^n$$
 (3.16)

For using the endpoint-interpolating cubic B-splines, the restriction is m = 2n+3 and m' = 2n' + 3 for some non-negative integers n' < n.

Simply the decomposition algorithm is run on the original curve (with m control points) until a curve with just m' control points is reached.

One notable aspect of the multiresolution curve representation is its discrete nature. 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, using the above method. However, there is no direct way to quickly construct curves that have fractional levels of smoothness.

For this the best solution presented in [13] is to define a fractional level curve  $f^{j+t}(u)$  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:

$$f^{j+t}(u) = (1-t) f^{j}(u) + t f^{j+1}(u)$$
$$= (1-t) \Phi^{j}(u) C^{j} + t \Phi^{j+1}(u) C^{j+1}$$

These fractional level curves allow for continuous levels of smoothing. Figure 3.10 shows some fractional level curves.

#### Editing the Sweep of the Curve

Editing the sweep of a curve at an integer level of the wavelet transform is carried out 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  $\tilde{C}^j$  be an edited version of  $C^j$ , given by  $\tilde{C}^j = C^j + \Delta C^j$ . The edited version of the highest-resolution curve  $\tilde{C}^n = C^n + \Delta C^n$  can be computed through reconstruction:

$$\tilde{C}^n = C^n + \Delta C^n$$

$$= C^n + P^n P^{n-1} \dots P^{j+1} \Delta C^j$$

Here it is worth noting that editing the sweep of the curve at lower levels of smoothing j affects the larger portion 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.

Figure 3.11 shows the editing of the sweep of the curve.

#### Editing the Character of the Curve

Another form of editing that is naturally supported by multiresolution curves is editing the character of a curve, without affecting its overall sweep.

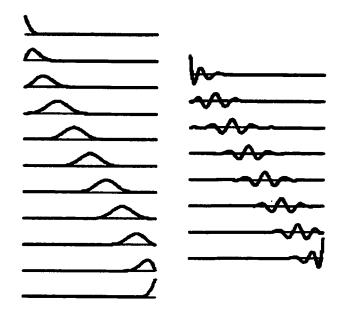


Figure 3.9: Cubic B-spline Scaling functions and wavelets for j = 3.



Figure 3.10: Smoothing a curve continuously. From left to right: the original curve at level 8, and smoother versions at level 5.4 and 3.1.



Figure 3.11: Changing the overall sweep of a curve without affecting its character. From left to right: the original curve, its extracted overall sweep, the sweep modified by the user, the details reapplied to the modified sweep.

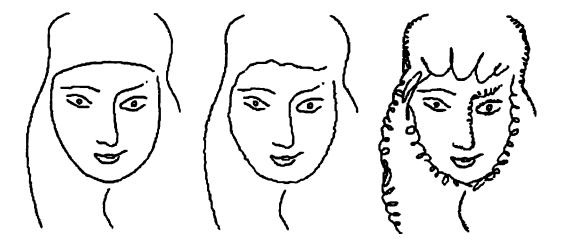


Figure 3.12: Changing the character of the curve without affecting its sweep.

Let  $C^n$  be the control points of a curve, and let  $C^0, C^1, \ldots, C^{n-1}$  and  $D^0, D^1, \ldots, D^{n-1}$  denote the components of its multiresolution decomposition. Editing the character of the curve is simply replacing the existing set of details  $D^j, D^{j+1}, \ldots, D^{n-1}$  with some new set  $\tilde{D}^j, \tilde{D}^{j+1}, \ldots, \tilde{D}^{n-1}$ , and reconstructing. Figure 3.12 shows a curve with change in its character without affecting its sweep.

#### Scan Conversion and Curve Compression

Using curve character libraries and other scan multiresolution editing features very complex curves consisting of hundreds or thousands of control points can be created, but in many cases these curves are printed in small form. So the conventional scan conversion methods using all the complexity of curves are wasteful in terms of the network traffic caused by sending these large files to the printer as well as the processing time required by the printer to render these curves with many control

points within a few square pixels. It is possible to develop a form of curve compression suitable for the purpose of scan conversion using multiresolution method that approximates the original curve within a specified error tolerance [10].

#### **Multiresolution Surfaces**

In many 3-D computer graphics applications surfaces play a central role. For the purpose of compression, multiresolution editing and many other operations that can be applied to images and curves can also be applied to surfaces by means of hierarchical representation [10]. The same idea of multiresolution analysis used for images and curves is applicable for the surfaces as well, i.e., the high-resolution surface is split into a low-resolution part and a detail part. This computation is carried out same as curves, i.e., multiplication by a matrix  $A^j$  for getting the low-resolution version surface and multiplication by another matrix  $B^j$  for obtaining the wavelet coefficients, which represent the details. This process is recursively applied to the low-resolution part until the coarsest representation of the surface is obtained. This process is shown in figure 3.13.

Surface Applications The applications for the surfaces include compression of surface models, continuous level-of-detail control for high-performance rendering, progressive transmission of complex surface models, and multiresolution editing of surfaces.

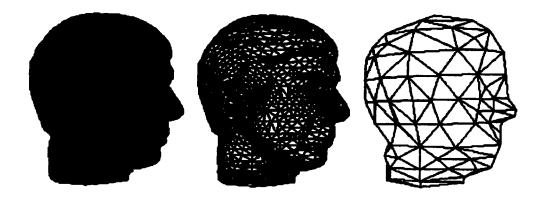


Figure 3.13: Decomposition of a Surface.

For the compression of a surface, first the surface is decomposed then all those wavelet coefficients are set to zero that are less than a predefined threshold value.

When viewing a complex object, it is unnecessary and inefficient to draw a highly detailed representation if the viewer is far away from the object. So it is better to use some form of level-of-detail control. Another use of continuous level-of-detail control is in the animation where switching suddenly between models with different levels of detail in an animation can produce objectionable popping, this problem is easily solved by using continuous level of smoothing. As the viewer approaches an object, each wavelet coefficient is smoothly varied from zero to its correct value. Likewise, as the viewer goes away from the object, each wavelet coefficient is smoothly reduced to zero. As a result, each wavelet coefficient can be made a continuous function of viewing distance.

On the World Wide Web, text, images, and video are very widely transmitted.

Complex geometric objects are also becoming very common. The ever-growing pro-

duction and distribution of these geometric objects motivates the need for efficient transmission of models across relatively low bandwidth networks.

A more attractive way is to use the wavelet representation for progressive transmission. Wavelet representation is used as; first the decomposed object is transmitted, as it contains the fewer control points it is received and displayed very quickly. Next, the normalized wavelet coefficients are transmitted in order of decreasing magnitude. As these coefficients are received, the renderer can update and redisplay the object with more resolution.

Multiresolution representation for surfaces can also be used for editing the surfaces. The editing of surfaces is used in much the same way as that for curves. Figure 3.14 shows an example of surface editing.

## 3.3 Multiresolution of NUBS Using Knot Decimation

Another method studied is for the multiresolution representation of Non-Uniform B-Splines (NUBS) as presented in [9]. The multiresolution decomposition of the freeform NUBS curve is computed using least-squares approximation, based on existing data reduction techniques. The least-squares decomposition allows the support of NUBS curves, but it also imposes some processing penalties in both time and space compared to techniques for multiresolution uniform B-spline curves [9].



Figure 3.14: Surface manipulation at different levels of detail.

#### 3.3.1 Least-Squares Multiresolution Decomposition

Let  $C_k(t)$  be a B-spline curve of order n and  $l_k$  control points, defined over the knot vector  $\tau_k$ , where  $k \in \mathbb{Z}^+$ . Let  $V_k$  be the space induced by  $\tau_k$ , and let  $\tau_{k-1} \subset \tau_k$ . The new space induced by  $\tau_{k-1}$ , denoted by  $V_{k-1}$  is clearly a strict subspace of  $V_k$ . Now, suppose  $C_{k-1}(t) (\in V_{k-1})$  be the least-squares approximation of  $C_k(t)$  in the space  $V_{k-1}$ , and their difference be the detail  $D_{k-1}(t) \in V_k$ , given by

$$D_{k-1}(t) = C_k(t) - C_{k-1}(t)$$
(3.17)

This process of decomposing a curve into two parts, one low resolution approximation and one high resolution detail can be applied recursively.  $C_k(t)$  could then be expressed as:

$$C_k(t) = C_0(t) + \sum_{i=0}^{k-1} D_i(t)$$
(3.18)

where  $C_0(t) \in V_0$  and  $D_i(t) \in V_{i+1}$ .

In order to construct a multiresolution decomposition of a NUBS curve as in

equation 3.18, the knot sequence  $\tau_i$ , inducing the subspaces  $V_i$  must first be defined.  $\tau_k$  is the knot vector of the original curve, the subsequent knot vectors  $\tau_i$ ,  $0 \le i < k$ , can be constructed such that  $\tau_i \subset \tau_{i+1}$  and  $2|\tau_i| \approx |\tau_i|$ , where  $|\cdot|$  denotes the size of the knot vector. The end conditions of the original curve must be preserved, hence the knots  $t_j \in \tau_i$ ,  $0 \le j < n$  and  $l_i \le j < l_i + n$ ,  $\forall 0 \le i < k$  are unmodified, where  $l_i$  denotes the number of control points defining  $C_i(t)$  over  $\tau_i$ . In general,  $l_i = |\tau_i| + n$ . This knot decimation process defines the function space hierarchy and is independent of the specific curve being decomposed.

For a B-spline curve with knot vector  $\tau_k$  of size  $2^k$ , k subspaces will be constructed, each induced by approximately half the knots of the previous level. The lowest resolution approximation  $C_0(t)$  will a single polynomial curve, i.e., the knot vector  $\tau_0$  has no interior knots ( $|\tau_0| = 2n$ ). Least-squares techniques [29] are employed to find the curve  $C_i(t) \in V_i$ , defined over  $\tau_i$ , best approximating  $C_k(t)$ . Following algorithm summarizes the multiresolution decomposition process.

 $C_i(t)$ s are regular Euclidean curves, where as the  $D_i(t)$ s are vector fields curves that can be used to reconstruct  $C_j(t)$  as:

$$C_j(t) = C_0(t) + \sum_{i=0}^{j-1} D_i(t)$$
 (3.19)

#### **INPUT:**

 $C_k(t)$ , a NUBS Curve.

#### **OUTPUT:**

 $C_0(t)$ ,  $D_i(t)$ ,  $0 \le i < k$ , the multiresolution decomposition of  $C_k(t)$ .

#### **ALGORITHM:**

- $\tau_k \leftarrow \text{Knot Sequence of } C_k(t)$ ;
- ullet for i=k-1 to 0 step -1 do

 $\tau_i$  = half the knots of  $\tau_{i+1}$ , preserving end conditions;

- ullet  $C_0(t) \Leftarrow$  Least Squares Approximation of  $C_k(t)$  in  $V_0$ , defined over  $au_0$ ;
- for i=1 to k do

begin

$$C_k(t) \Leftarrow C_k(t) - C_{i-1}(t);$$

 $C_i(t) \Leftarrow$  Least Squares Approximation of  $C_k(t)$  in  $V_i$ , defined over  $au_i$ ;

$$D_{i-1}(t) \Leftarrow C_i(t) - C_{i-1}(t)$$
;

end;

The problem of knot sequence decimation for the purpose of data reduction purposes is addressed in [7]. In which knots are selected for removal by weighing their possible effect on the curve. Where as in this case, knots are selected so as to minimize the local effect on the curve due to removals from level i to level i+1. Hence, consecutive knots should not be removed in one step. Removing every  $n^{th}$ knot, where n is the order of the curve will cause the least change from one level to the next, yet affect the entire curve. As the degree of a Bezier or B-spline curve is increased, the curve is becomes smoother and smoother due to the low pass property of the basis functions of the representation. Therefore, as n increases, by selecting every  $n^{th}$  knot for removal, the knots are removed at larger intervals yet the curve becomes smoother. In practice, it is found that removing every alternate knot still retains a sufficient number of resolution levels to enable an effective multiresolution control. Moreover, the computational overhead required for the algebraic summation is kept at interactive speeds. Figure 3.15 shows the multiresolution decomposition for a star shaped curve.

#### 3.3.2 Manipulation of Multiresolution Decompositions

A main purpose for applying multiresolution decomposition to B-spline curves is to obtain the ability to manipulate and edit the curve at different resolution levels. Figure 3.16 shows a signature curve modified at different levels. Editing is performed at the center of s character.

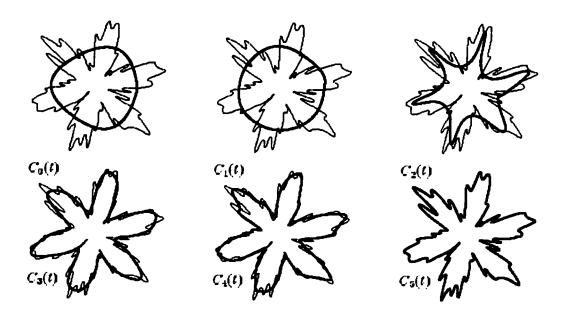


Figure 3.15: Multiresolution Decomposition of a B-spline Star Curve  $C_5(t)$  (of order 3, defined with 100 control points.). The Original Curve is shown in thin lines throughout while  $C_i(t)$ s,  $0 \le i < 5$  are shown in thick lines.

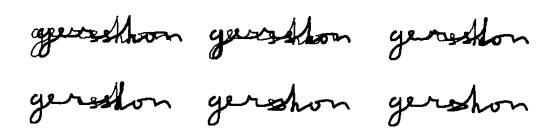


Figure 3.16: Editing the same curve location, the center of the s in the signature, at different resolution levels, from the lowest (top left) to the highest resolution (right bottom). The original curve is shown in thin lines and the low resolution curves are displayed in thick lines.

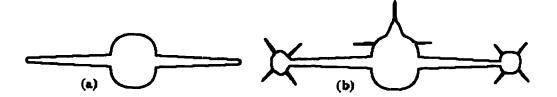


Figure 3.17: (a) Low resolution quadratic B-spline curve (with 22 control points), representing a cross-section of an airplane. (b) The curve of (a) is locally refined to create the degrees of freedom that are necessary to interactively model the missiles at the wing tips, the elevators and the steering wings, as well as the cockpit.

The ability to handle multiresolution NUBS curves allows one to not only edit and manipulate them at different resolution levels, but also to provide local refinement control, inserting new knots into the curve in the neighborhood of the domain to be manipulated. A fixed number of knots are heuristically inserted in the neighborhood of the refined location while preserving continuity as much as possible by prohibiting knot multiplicities. Figure 3.17(a) shows a low resolution cross-section of an airplane, edited and refined to form missiles at its wing tips as well as its elevators, steering wings, and cockpit in figure 3.17(b).

### Chapter 4

# Multiresolution Representation of NURBS

All the multiresolution decomposition approaches presented for splines are restricted to non-rational splines. There is no model suggested which can address the case of rational splines. Non-Uniform Rational B-Splines (NURBS) are the generalization of B-splines. Among the types of B-splines, NURBS have been receiving considerable attention in the areas of computer graphics and geometric modeling. NURBS or the splines with similar geometric properties as those of NURBS [4, ?, ?, ?, ?, ?, ?], have got various applications in computer graphics. In a very short period of time, NURBS are industry standard tools for the representation and design of geometry. Hence there is a need to represent NURBS using multiresolution decomposition for the purpose of obtaining the ability to manipulate and edit the curve at different

resolution levels. In this work, we propose a model based on control point decimation for multiresolution representation of NURBS. As NURBS are the generalization of B-splines, the same model can be used for other types of splines like uniform non-rational and non-uniform non-rational B-splines as well.

#### 4.1 Development of the Model

Because of the flexibility of B-spline basis functions and hence of the resulting B-spline curves, different types of control handles are used to influence the shape of the curve [8]. Control is achieved by:

- Changing the type of the knot vector and hence basis functions (periodic uniform, open uniform or non-uniform), figure 4.1.
- Changing the Order of the basis functions, figure 4.2.
- Changing the position and order of the defining polygon vertices (Control Points), figure 4.3.
- Using multiple polygon vertices, figure 4.4.
- Using multiple knot values in the knot vector, figure 4.5.

And for the NURBS an additional control handle is provided by modifying the weight values of the control points, figure 4.6.

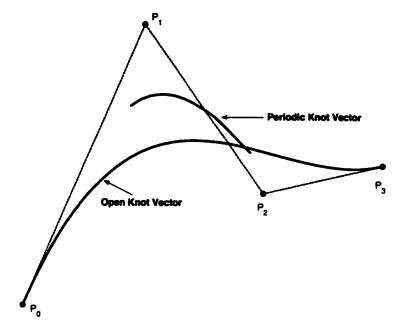


Figure 4.1: B-spline curve shape control by changing the type of knot vector.

By using the ability to control a B-spline curve by changing the position and order of the control points, we can come up with a multiresolution representation for NURBS. In this work we used the control point decimation for the purpose of multiresolution representation of NURBS.

Let  $C_k(t)$  be a NURBS curve, defined over the set of polygon vertices or control points  $P_k$  (consisting of corresponding weight values for each point in addition to X and Y co-ordinate values) containing  $l_k$  points, using the knot vector  $T_k$ , where k is a positive integer, greater than zero. There are various methods proposed for the calculation of non-uniform knots, a popular method is to calculate the knot vector proportional to the chord lengths between the defining polygon vertices. We use

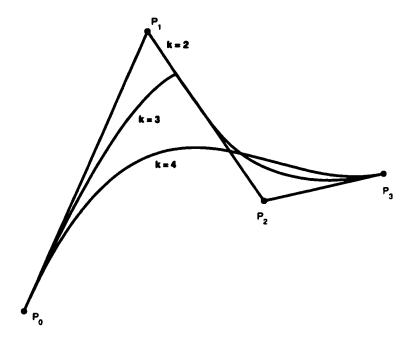


Figure 4.2: B-spline curve shape control by changing the Order(k) of the Basis Functions.

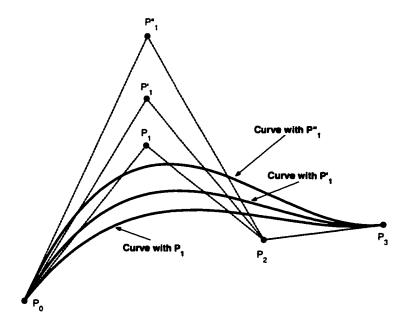


Figure 4.3: B-spline curve shape control by changing the Position of Control Points.

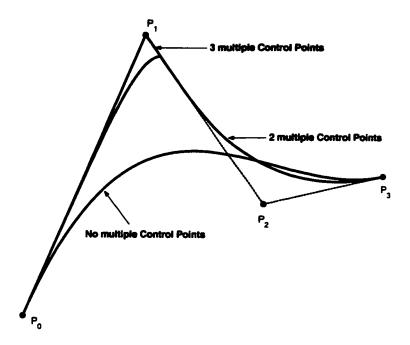


Figure 4.4: B-spline curve shape control by using Multiple Control Points.

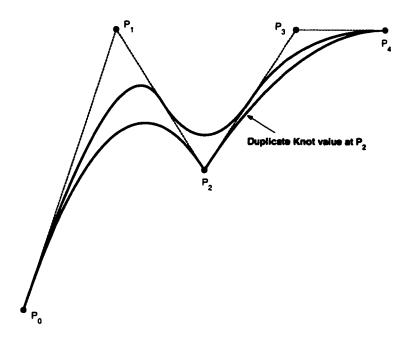


Figure 4.5: B-spline curve shape control by using Multiple Knots.

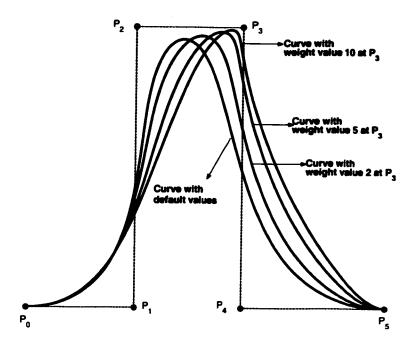


Figure 4.6: NURBS curve shape control by changing Weight values.

the same knot calculation method. The NURBS curve  $C_k(t)$  is calculated from the control points  $P_k$  as given in the following equation and described in detail in section 2.2.

$$C_k(t) = \sum_{i=0}^{l_{k-1}} P_{k,i} R_{i,p}(t)$$
 (4.1)

Let  $V_k$  be the space of all the curves that can be defined using control points  $P_k$ . Now, we find a subset  $P_{k-1}$  of  $P_k$  ( $P_{k-1} \subset P_k$ ), clearly the space  $V_{k-1}$  induced by  $P_{k-1}$  is a subset of  $V_k$ . Let  $C_{k-1}(t) \in V_{k-1}$  is a curve defined over the control points  $P_{k-1}$ , and we found out that it is the approximation to the higher resolution curve  $C_k(t)$ . To find  $P_{k-1}$  from  $P_k$ , we use the process of decimation. Let a unary operator  $\mathbf{d}_j$  is defined for decimation, where j denotes the interval that is used to decimate the control points. If j is 2 then every  $2^{nd}$  (alternate) control point is decimated, if j is 3 then select every  $3^{rd}$  control point (i.e., control points numbered 3, 6, 9, ...) for removal. Similarly, if j is i then decimate every i<sup>th</sup> control point. Mathematically control point decimation is given by:

$$P_{k-1} = \mathbf{d}_j(P_k) \tag{4.2}$$

To minimize the local effect on the resulting curve  $C_{k-1}(t)$ , consecutive control points from  $P_k$  should not be removed to obtain  $P_{k-1}$ . It is observed that removing every alternate point causes the acceptable amount of local effect and still retains a sufficient number of resolution levels to enable an effective multiresolution control. The lost control points can be captured as  $Q_{k-1}$ .

Let another unary operator  $c_j$  is defined to capture the decimated control points. Here also j denotes the interval used to decimate the points. Mathematically  $Q_{k-1}$  can be computed as:

$$Q_{k-1} = \mathbf{c}_j(P_k) \tag{4.3}$$

The process of decomposition can be applied recursively until  $P_0$ , which contains only n control points, where n is the order of the B-spline curve.

Following algorithm summarizes the multiresolution decomposition process and the flow chart in figure 4.7 shows it pictorially.

**INPUT:** 

 $C_k(t)$ , a NUBS Curve.

**OUTPUT:** 

 $P_0, Q_i, 0 \leq i < k$ , the multiresolution decomposition of  $C_k(t)$ .

#### **ALGORITHM:**

- ullet  $P_k \Leftarrow$  Control Points of  $C_k(t)$ ;
- ullet for i=k-1 to 0 step -1 do

begin

$$P_i = d_j(P_{i+1});$$

$$Q_i = c_j(P_{i+1});$$

end;

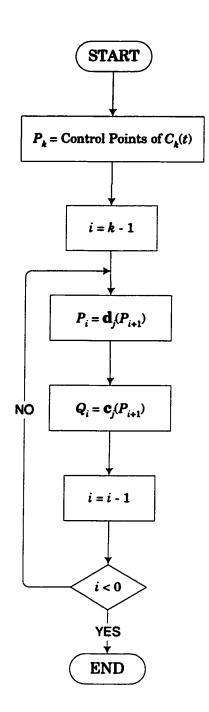


Figure 4.7: Flow Chart of the Multiresolution Decomposition Process.

The reconstruction of  $P_i$  from  $P_{i-1}$  and  $Q_{i-1}$  is carried out by merging the sets  $P_{i-1}$  and  $Q_{i-1}$ . Let a binary operator  $\mathbf{r}_j$  is defined for the process of reconstruction of  $P_i$  from  $P_{i-1}$  and  $Q_{i-1}$ . The reconstruction is mathematically represented as:

$$P_i = \mathbf{r}_j(P_{i-1}, Q_{i-1}) \tag{4.4}$$

While reconstructing, the criteria used for the decomposition should be followed. For example, if every  $j^{th}$  point is decimated during decomposition, then the reconstruction of  $P_i$  is obtained by rearranging  $P_{i-1}$  and  $Q_{i-1}$  as; place (j-1) points from  $P_{i-1}$  and one point from  $Q_{i-1}$  in the same order and so on.

By means of recursively applying the reconstruction operator the original set of control points can be represented in terms of its multiresolution components as:

$$P_k = \mathbf{r}_j(P_0, Q_0, Q_1, Q_2, ..., Q_{k-1})$$
(4.5)

The above recursion can be expanded as follows:

$$P_k = \mathbf{r}_j(\mathbf{r}_j(P_0, Q_0), Q_1, Q_2, ..., Q_{k-1})$$

$$= \mathbf{r}_j(P_1, Q_1, Q_2, ..., Q_{k-1})$$

$$= \mathbf{r}_j(\mathbf{r}_j(P_1, Q_1), Q_2, ..., Q_{k-1})$$

$$\cdot \cdot \cdot \cdot$$

$$= \mathbf{r}_j(P_{k-1}, Q_{k-1})$$

#### 4.2 Results

This section presents the results obtained using the proposed model on curves and surfaces.

#### 4.2.1 Multiresolution Decomposition

Figure 4.8 shows a degree 3 NURBS curve drawn with 259 control points with the default weight values i.e., all wight values are set to 1. This curve can be decomposed into six lower level curves as shown in figure 4.9. There are six lower resolution levels (from level 5 down to level 0) are possible for this curve, hence the original curve is assumed to be at level 6.

Table 4.1: Details of Curves of figure 4.9.

Figure Number	No. of Control Points	Level
4.9(a)	130	5
4.9(b)	66	4
4.9(c)	34	3
4.9(d)	18	2
4.9(e)	10	1
4.9(f)	6	0

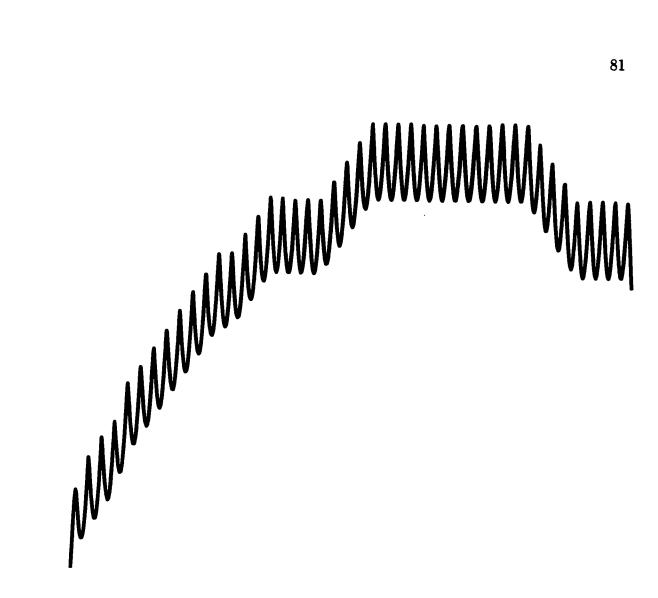


Figure 4.8: A NURBS Curve with default weight values.

The original curve is decomposed by decimating alternative control points to get the curve at level 5 with 130 control points (figure 4.9(a)). The curve at level 4 (figure 4.9(b)) is obtained by decimating every alternative control point from the level 5 curve. Similarly curve at a lower level is obtained by decimating alternative control points from the curve at one level higher to it. Table 4.1 gives the details of all the lower resolution level curves of figure 4.9.

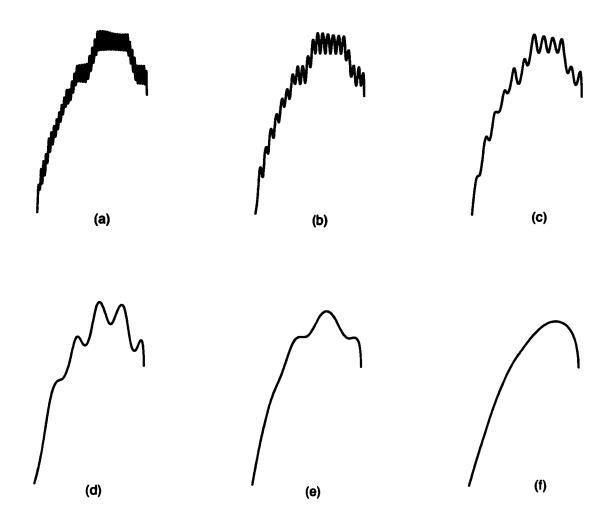


Figure 4.9: Multiresolution levels of the curve in figure 4.8.

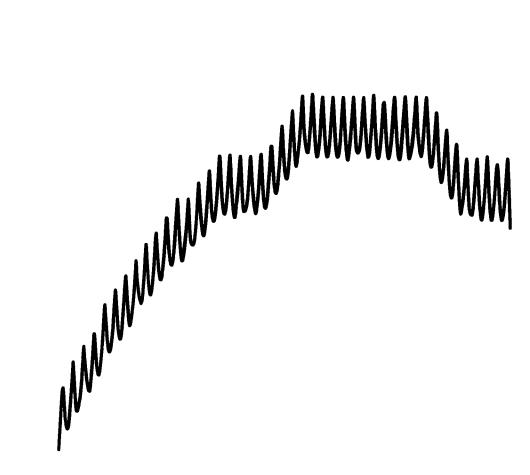


Figure 4.10: A NURBS Curve with non-default weight values.

Figure 4.10 shows the same NURBS curve as in figure 4.8, but with non-default weight values assigned to the control points. This curve can also be decomposed into lower resolution versions by means of control point decimation method in the same manner as performed for the curve in figure 4.8. Figure 4.11 shows all its multiresolution levels. The attributes of these curves are same as those of the curves in figure 4.9, as given in table 4.1.

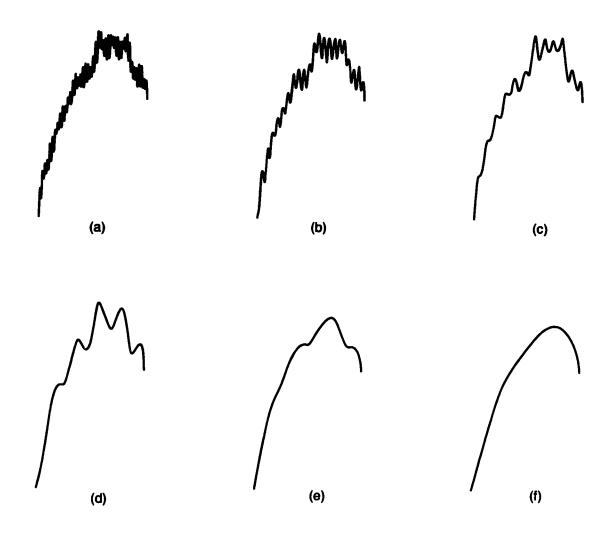


Figure 4.11: Multiresolution levels of the curve in figure 4.10.

Table 4.2: Details of Curves of figure 4.13.

Figure Number	No. of Control Points	Level
4.13(a)	159	5
4.13(b)	80	4
4.13(c)	40	3
4.13(d)	20	2
4.13(e)	10	1
4.13(f)	5	0

Figure 4.12 shows another NURBS curve of degree 3 consisting of 319 control points with default weight values. In total, six multiresolution levels are obtained for this curve, as shown in figure 4.13. This is an example of applying the multiresolution decomposition on closed curves. Table 4.2 lists the details of the curves in figure 4.13.

While performing control point decimation, a possible variation is to decimate every odd numbered point in one iteration and in the next iteration all the even numbered points can be decimated. This gives a little better control on the shape of decomposed curves.

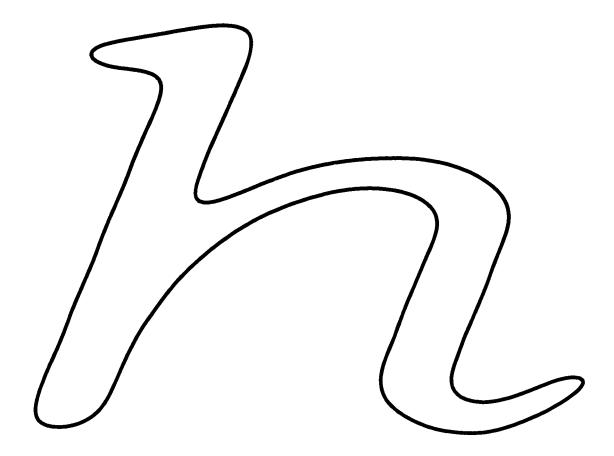


Figure 4.12: A Closed NURBS Curve.

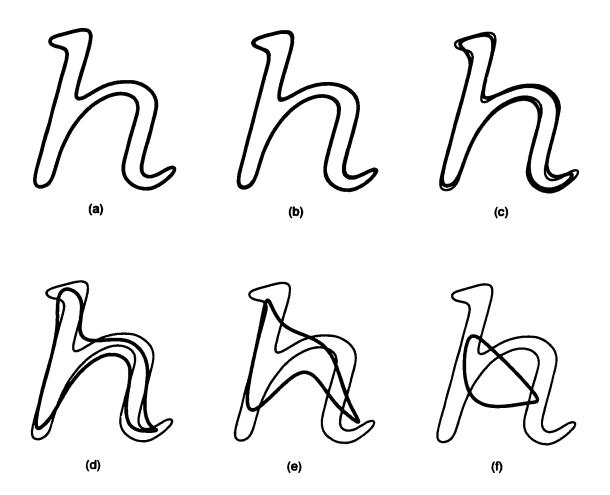


Figure 4.13: Multiresolution levels of the curve in figure 4.12.

#### 4.2.2 Editing Curves using Multiresolution Representation

Multiresolution analysis is defined as an ability to simultaneously perform both local and global operations on the analyzed object. Using multiresolution representation there are two types of editings of the curves (in general shapes) are possible, one that affects the entire shape and the other which affects only a part of the shape.

#### Global Editing

Figures 4.14 to 4.22 demonstrate the ability to perform global editing (editing on the entire curve) on a NURBS curve using multiresolution representation. Figure 4.14 is same as figure 4.12 on which the editing is performed. Figure 4.16 is a result of editing performed on the Q component of level 3 curve and then reconstructing back to the highest level. This process is explained as follows:

The original curve is  $C_6(t)$  with a set of control points  $P_6$  as shown in figure 4.14.

- Decompose the curve  $C_6(t)$  up to level 3 to obtain  $C_3(t)$ , thus obtaining the multiresolution components  $P_3, Q_3, Q_4$ , and  $Q_5$ .
- Perform editing on  $Q_3$  to get  $Q_3'$ . For the process of editing, let e be an operator which is applied on  $Q_3$  to get  $Q_3'$  based on some criteria(this criteria can be anything depending upon the interest of the user, for example user can add or subtract a particular value from each point in  $Q_3$  to get  $Q_3'$ , or he/she can add a value to a point and subtract the same value from the next point

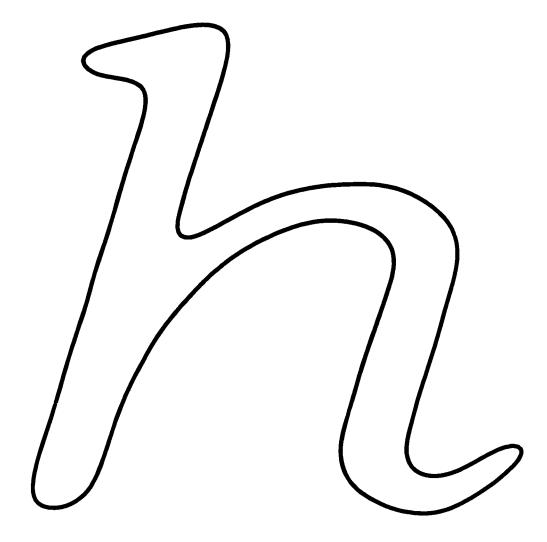


Figure 4.14: Original NURBS Curve.

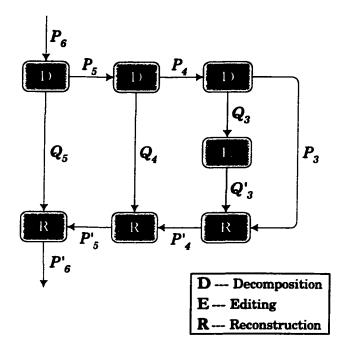


Figure 4.15: Diagrammatic representation of editing at level 3.

and so on). Hence the editing operation can be defined as:

$$Q_3' = \mathbf{e}(Q_3) \tag{4.6}$$

• Reconstruct  $P'_4$  from  $P_3$  and  $Q'_3$ . Similarly obtain  $P'_5$  from  $P'_4$  and  $Q_4$  and finally  $P'_6$  (figure 4.16) is obtained from  $P'_5$  and  $Q_5$ .

Figure 4.15 shows the above process in a diagrammatic representation.



Figure 4.16: NURBS Curve with editing performed at level~3.

Figure 4.18 shows the curve after performing editing at level 4 of the decomposition. To obtain this edited curve first the original curve  $C_6(t)$  of figure 4.14 is decomposed to two levels to get the lower resolution curve  $C_4(t)$ , and then its Q component (i.e.,  $Q_4$ ) is edited to get  $Q'_4$ . It is then combined with its corresponding P component to reconstruct the curve at level 5. And again the level 5 curve is subjected to reconstruction to get the highest level curve. This whole process is explained diagrammatically in figure 4.17.

Figure 4.20 is the result of performing editing at *level* 5, similar to that of explained above. Its diagrammatic representation is given in figure 4.19.

Whereas the figure 4.22 is an example of editing at all levels of the multiresolution representation. For performing this kind of editing first the original curve is decomposed to the lowest level (level 0), now the Q component i.e.,  $Q_0$  is edited to get  $Q'_0$ . It is now combined with  $P_0$  by means of reconstruction process to get  $P'_1$ . Similarly  $Q_1$  is edited to get  $Q'_1$ , and  $P'_2$  is reconstructed by combining  $Q'_1$  and  $P'_1$ . This process of editing the Q component and combining with the corresponding Pcomponent is repeated until the last level curve is reconstructed, as shown in figure 4.21.

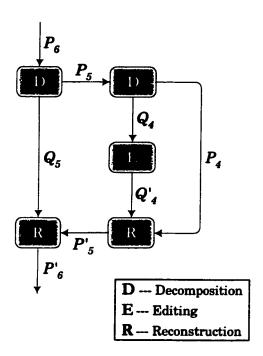


Figure 4.17: Diagrammatic representation of editing at level 4.

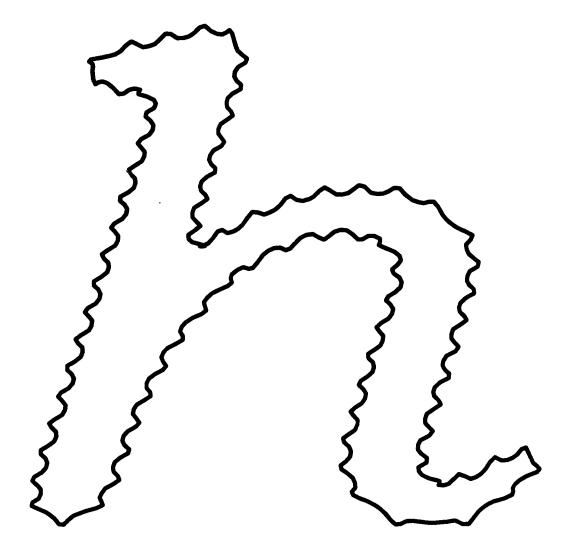


Figure 4.18: NURBS Curve with editing performed at level 4.

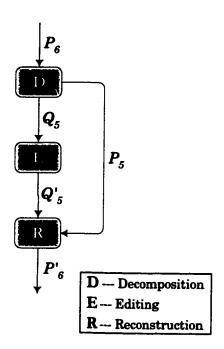


Figure 4.19: Diagrammatic representation of editing at  $level\ 5$ .

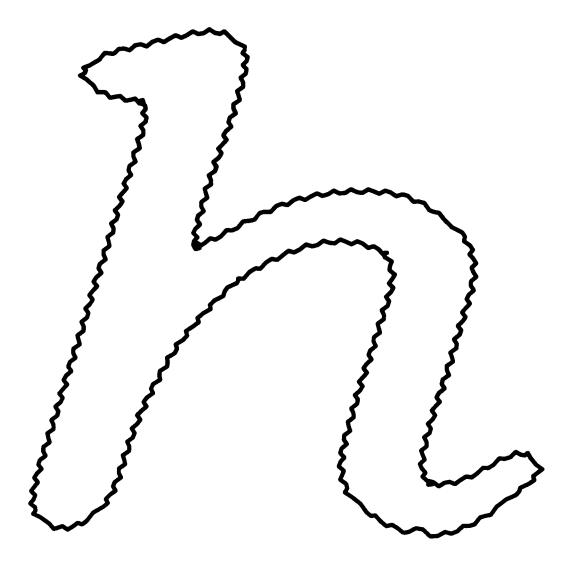


Figure 4.20: NURBS Curve with editing performed at level 5.

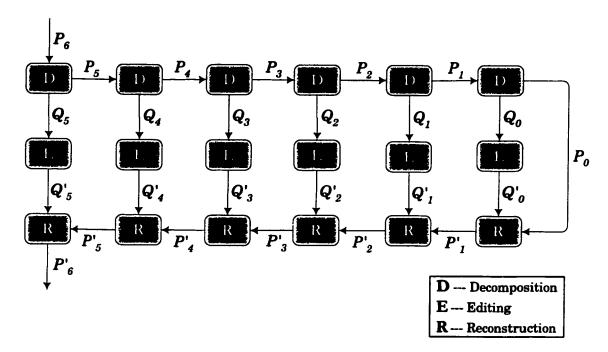


Figure 4.21: Diagrammatic representation of editing at all levels.

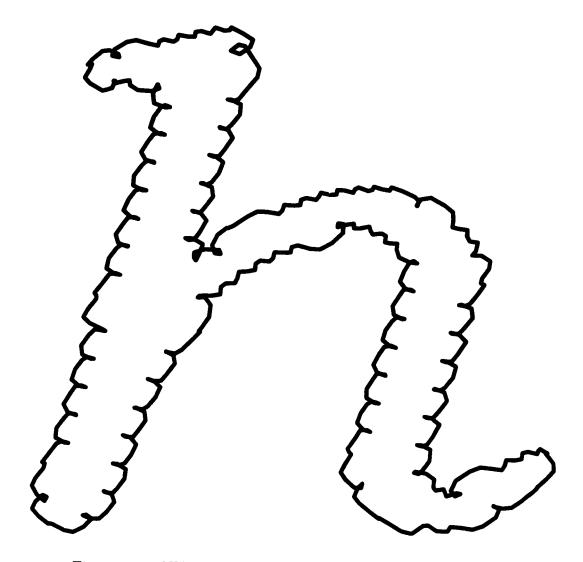


Figure 4.22: NURBS Curve with editing performed at all levels.

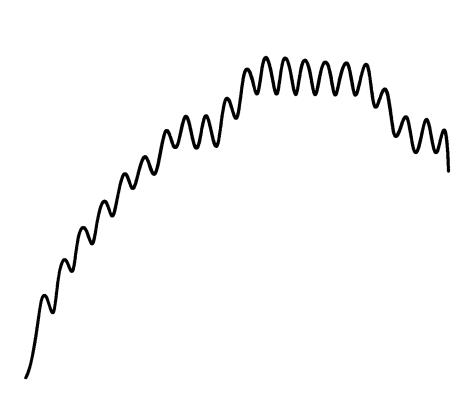


Figure 4.23: A NURBS Curve.

#### **Local Editing**

In the context of the local editing, we edit only a part of the shape without affecting the whole shape. Figure 4.23 shows a curve on which the editing is performed. Figure 4.24 shows the curve of 4.23 after decomposing to each level. After decomposing to the last level of multiresolution, it is edited and the curve is reconstructed at all levels. Figure 4.25 shows the reconstructed curves corresponding to curves of figure 4.24 after performing editing locally at a point in the middle of the curve. Figure 4.26 shows the final curve after reconstruction.

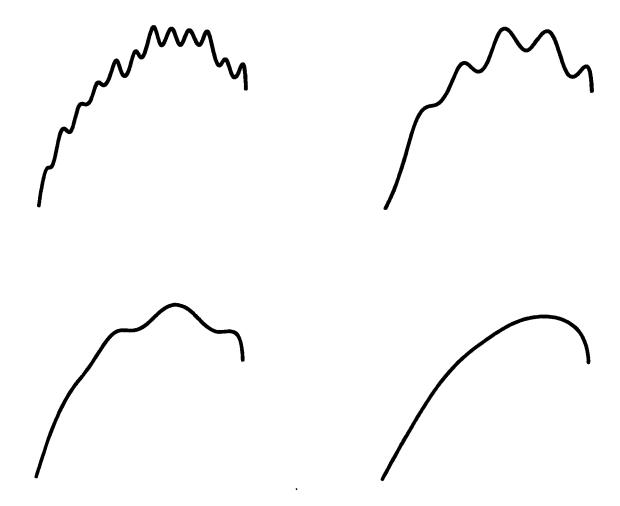


Figure 4.24: Decomposition levels of the NURBS Curve.

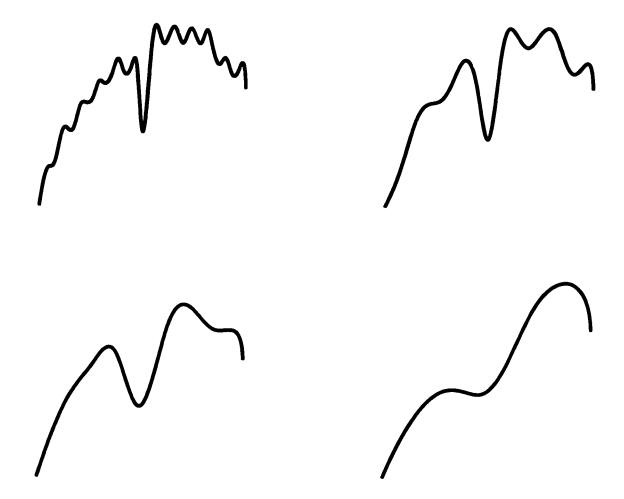


Figure 4.25: All levels of Locally Edited NURBS Curve.

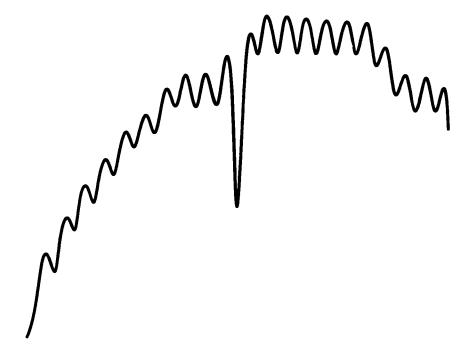


Figure 4.26: Locally Edited Final NURBS Curve.

#### 4.2.3 Multiresolution Surfaces

The same technique of multiresolution decomposition can be extended for surfaces, to achieve the control that is possible for the curves. A NURBS surface S(u, v) with  $l_u$  control points and degree m in the direction of the parameter u and  $l_v$  control points and degree n in the direction of parameter v with a set of control points P (here P contains 3-Dimensional data) is represented as:

$$S(u,v) = \sum_{i=0}^{l_u-1} \sum_{j=0}^{l_v-1} P_{i,j} R_{i,m}(u) R_{j,n}(v)$$
 (4.7)

where R is the NURBS basis function as described in section 2.2.

If  $P_k$  is the set of control points for a NURBS surface  $S_k(u, v)$  (where k is a positive integer used to denote a surface at a particular level of multiresolution), the operators for decimation, capturing of decimated points, and reconstruction can be defined by following the same reasoning as described in section 4.1. The decimation process is then written as:

$$P_{i-1} = \mathbf{d}_j(P_i) \tag{4.8}$$

As the surfaces are represented by means of a mesh (Two-Dimensional Array) of control points, the control point decimation strategy is to be modified a little for applying it to surfaces. For surfaces, the control point decimation is carried out as follows: first all rows are decimated based on a criteria (like every alternative

row, or every  $k^{th}$  row, etc.) from the mesh of control points used to represent a surface. The columns are now decimated from thus obtained mesh of control points after decimating rows. The columns are decimated by using the same criteria used to decimate rows in the first step. Here the role of rows and columns can be interchanged, i.e., instead of rows, columns can be decimated first.

Similarly, the process of capturing the decimated points and the reconstruction process are represented in the form of following two mathematical equations.

$$Q_{i-1} = \mathbf{c}_j(P_i) \tag{4.9}$$

$$P_i = \mathbf{r}_j(P_{i-1}, Q_{i-1}) \tag{4.10}$$

Figure 4.27 shows a bi-cubic NURBS surface in the shape of an Ice Cream Cup. Figures 4.28 to 4.30 show the surface at its decomposition levels.

By means of hierarchical representation of surfaces using multiresolution, editing and other operations that can be applied to curves can also be applied to surfaces. For multiresolution editing of the surfaces the same technique is applicable which is applied to curves i.e., first by decomposing, after that performing editing on the decomposed version and then reconstructing back to the original resolution level.

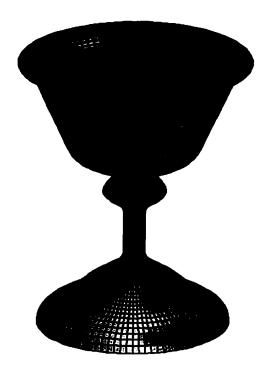


Figure 4.27: A Bi-Cubic NURBS Surface with a 30 X 30 mesh of Control Points.

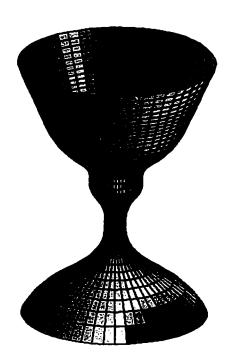


Figure 4.28: Decomposed NURBS Surface with 15 X 15 mesh of Control Points.

.

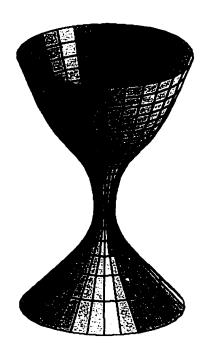


Figure 4.29: Decomposed NURBS Surface with 8 X 8 mesh of Control Points.



Figure 4.30: Decomposed NURBS Surface with 4 X 4 mesh of Control Points.

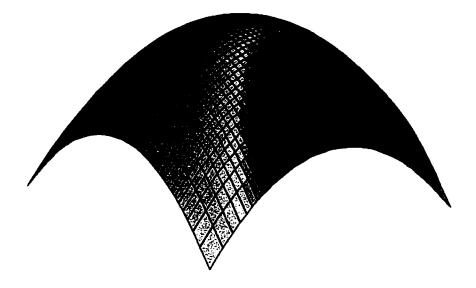


Figure 4.31: A NURBS Surface with 15 X 15 mesh of Control Points.

Figure 4.31 shows another NURBS surface. This is also a bi-cubic surface, this is an example of open NURBS surface. The ability of multiresolution decomposition on this surface is shown in figures 4.32 to 4.34. As it is clear from the figures that in the decomposition process, first three and the last three rows as well as columns are preserved so as to retain the shape of the surface.

Figure 4.35 shows a NURBS surface and its multiresolution decompositions are shown in figures 4.36 to 4.38. As another example of multiresolution decomposition of surfaces, figure 4.39 shows the original NURBS surfaces and in figures 4.40 and 4.41 its multiresolution decompositions are shown.

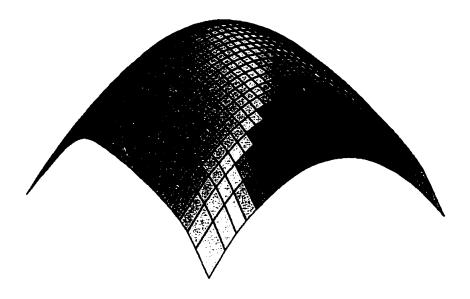


Figure 4.32: Decomposed NURBS Surface with 10 X 10 mesh of Control Points.

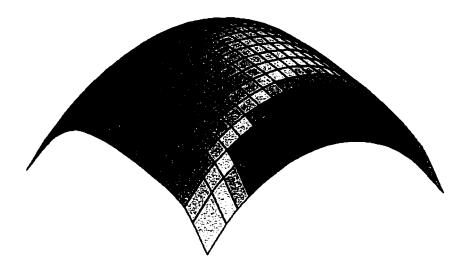


Figure 4.33: Decomposed NURBS Surface with 8 X 8 mesh of Control Points.

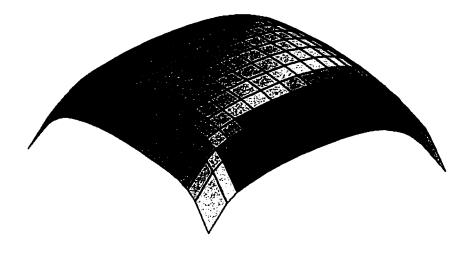


Figure 4.34: Decomposed NURBS Surface with 7 X 7 mesh of Control Points.

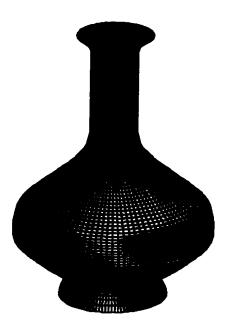


Figure 4.35: A NURBS Surface with 33 X 33 mesh of Control Points.



Figure 4.36: Decomposed NURBS Surface with 17 X 17 mesh of Control Points.

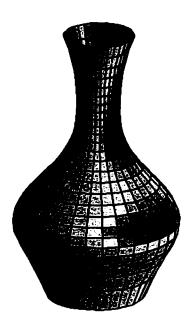


Figure 4.37: Decomposed NURBS Surface with 9 X 9 mesh of Control Points.



Figure 4.38: Decomposed NURBS Surface with 5 X 5 mesh of Control Points.



Figure 4.39: A NURBS Surface with 19 X 19 mesh of Control Points.

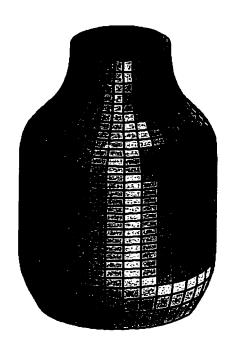


Figure 4.40: Decomposed NURBS Surface with 10 X 10 mesh of Control Points.

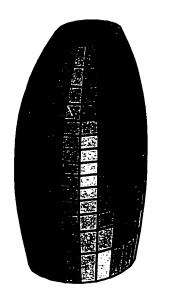


Figure 4.41: Decomposed NURBS Surface with 5 X 5 mesh of Control Points.

### Chapter 5

## Multiresolution of Splines: A

## **Comparative Study**

This chapter discusses the comparison of our proposed method with each of the two methods of multiresolution described in chapter 3. In this chapter, the ability of representing various B-spline models by these multiresolution models are described. Next the constraints on the number of control points that are imposed on B-splines so as to use these methods are described. Finally, the results from the proposed method is compared with those of the methods studied in chapter 3.

#### 5.1 Limitations of the Multiresolution Models

The multiresolution representation using wavelets method [13, 10, 11] described in section 3.2 can represent only the uniform non-rational B-splines, it fails to represent the non-uniform B-splines as well as the rational B-splines. Whereas the method that uses knot decimation and least-squares approximation [9] (section 3.3) can represent non-uniform non-rational B-splines. As uniform B-spline is the special case of non-uniform B-spline, it can be used for the uniform B-splines as well. In this work, a multiresolution representation for NURBS is proposed. Non-rational spline is the special case of NURBS, hence the proposed method can be used to represent non-uniform non-rational B-spline as well as the uniform B-splines. Table 5.1 summarizes the types of B-splines and the ability of these three models to represent them using multiresolution.

Table 5.1: Multiresolution Method's Ability to Represent the Types of B-splines.

Multiresolution Model	Uniform B-splines	NUBS	NURBS
Using Wavelets (section 3.2)	✓		
By Knot Decimation (section 3.3)	✓	✓	
By Control Point Decimation	✓	<b>√</b>	✓

The multiresolution representation of end-point interpolating B-splines using wavelets (section 3.2) imposes a constraint on the number of control points of the

curve that is to be represented by this model; the curve should have  $2^{j} + 3$  control points, where j is a positive integer greater than 0. While the other two models, including the proposed one do not impose this constraint on the curves being modeled.

# 5.2 Multiresolution Model Comparison: Waveletsv/s Control Point Decimation

Figure 5.1 shows a B-spline curve, whose details are given in table 5.2. Figure 5.2 shows all its decomposition levels using the wavelets multiresolution method, six lower resolution levels are obtained. The average execution time is recorded as 45.1 seconds <sup>1</sup>. Figure 5.3 shows the multiresolution decomposition levels of the same curve by control point decimation method, with this method there are four lower resolution levels are possible. The average execution time is recorded as 0.068 seconds.

<sup>&</sup>lt;sup>1</sup>All programs are executed on a Pentium III, 700MHz machine, running Windows 2000 operating system with 128MB of RAM.

Table 5.2: Figure 5.1 Attributes.

Property	Value	
Type of Curve	Uniform B-spline	
Degree	3	
No. of Control Points	67	

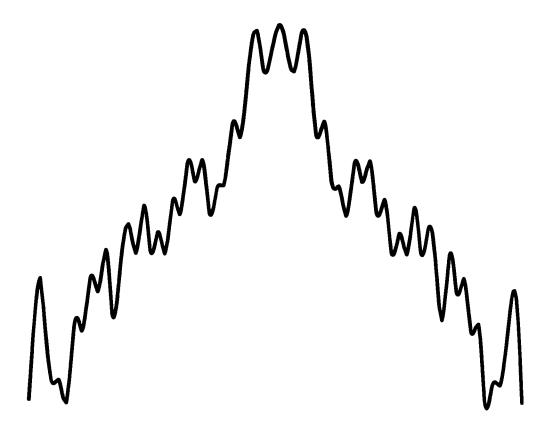


Figure 5.1: A Uniform B-spline Curve.

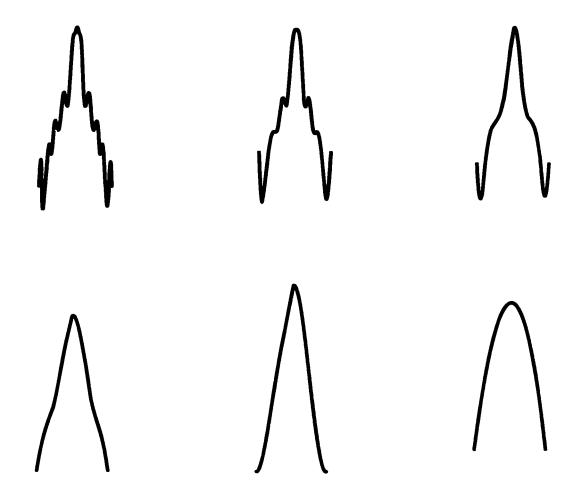


Figure 5.2: Multiresolution levels using Wavelets.

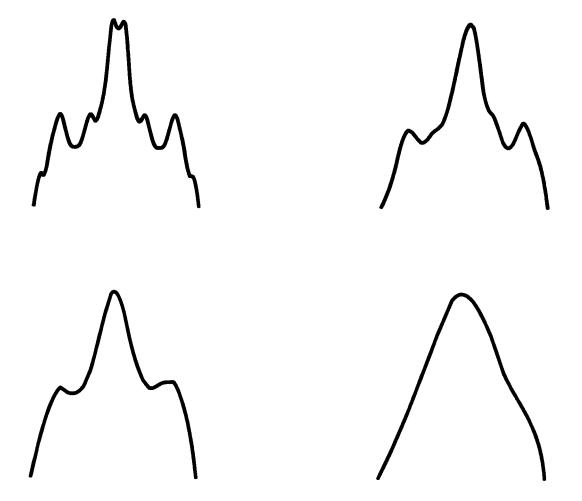


Figure 5.3: Multiresolution levels by Control Point Decimation Method.

# 5.3 Multiresolution Model Comparison: Knot Decimation v/s Control Point Decimation

Figure 5.4 shows a Star Shaped NUBS curve <sup>2</sup>, whose details are given in table 5.3. Figure 5.5 shows all its decomposition levels using the knot decimation multiresolution method, five lower resolution levels are obtained. Figure 5.5(a) is the original curve. Figures 5.5(b) to 5.5(f) show its lower level curves. In each figure the original curve is shown in thin line and the decomposed curve is in thick line. The average execution time is recorded as 1.73 seconds. Figure 5.6 shows the multiresolution decomposition levels of the same curve by control point decimation method, with this method also there are five lower resolution levels are possible. The average execution time is recorded as 0.07 seconds.

Table 5.3: Figure 5.4 Attributes.

Property	Value
Type of Curve	NUBS
Degree	2
No. of Control Points	100

<sup>&</sup>lt;sup>2</sup>The data was kindly provided by Dr. Gershon Elber [9].

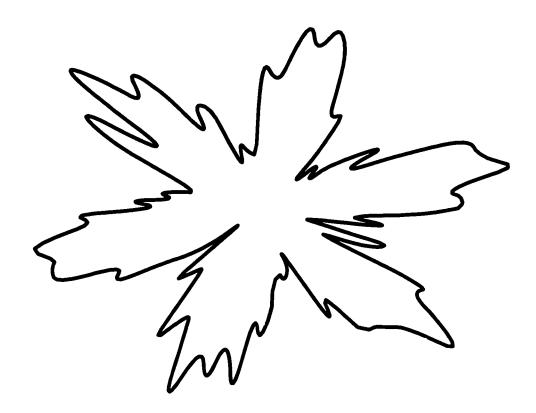


Figure 5.4: A Star Shaped NUBS Curve.

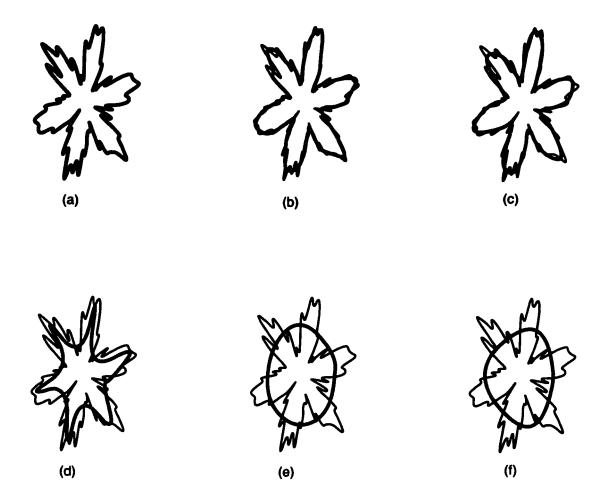


Figure 5.5: Multiresolution levels by Knot Decimation Method.

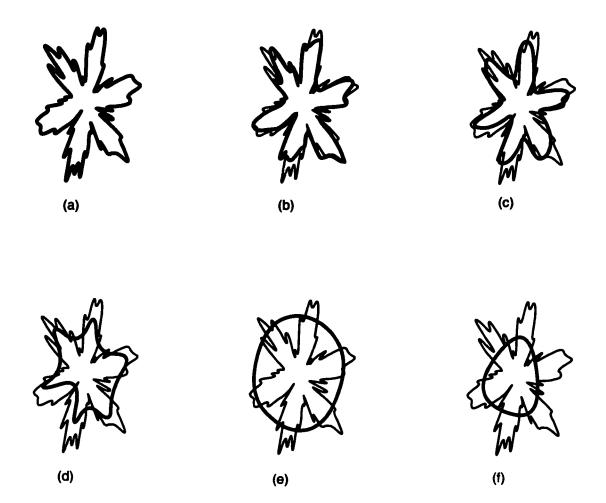


Figure 5.6: Multiresolution levels by Control Point Decimation Method.

## Chapter 6

### Conclusion and Future Work

#### 6.1 Conclusion

As NURBS have got various applications in computer graphics, there was a need to represent NURBS using multiresolution decomposition for the purpose of obtaining the ability to manipulate and edit the curve at different resolution levels. A framework for multiresolution representation of NURBS is developed for use in various computer graphics applications which require both local as well as global operations to be performed on B-splines. A method for representing NURBS using multiresolution is successfully devised, the objective of the work is achieved. The idea of multiresolution representation of NURBS curves is extended to achieve multiresolution control for surfaces as well.

It is observed that the wavelet method provides an efficient representation but

it suffers from the fact that it is relatively very slow. Although our proposed model does not approximate the original shape as efficiently as the other models do, but it is very efficient with respect to execution time when compared with two of the existing methods for decomposing B-splines using multiresolution representation. In view of this observation the proposed model for the multiresolution representation of the NURBS is a very good choice in applications where the tolerance in the approximation is not of much importance.

The method presented is not capable of providing a continuous resolution control, i.e., the decomposition at fraction levels.

Multiresolution analysis is defined as an ability to simultaneously perform both local and global operations on the analyzed object. The work presented here can be used for this purposes as it provides the ability to perform both local as well as the global operations.

#### 6.2 Future Work

- The ability to have a continuous multiresolution control would add a significant functionality to this method. Investigation of any such method would be a major addition to the proposed model.
- It is observed that the proposed approach for multiresolution representation of splines is not that efficient in approximating the lower levels of a curve. It

can be improved by means of optimizing the weights of the control points. It will add a significant improvement if some method is developed that optimizes the weights. This enhancement would make the proposed multiresolution representation method very efficient.

- Sometimes we encounter curves and surfaces in which some control points are significant. Decimation of those control points may drastically change the shape of the object. In view of this it is sought that there should be an intelligent technique which can check these significant points at the time of decimation. This can be achieved by assigning a weight value for each points based on their significance.
- Extension of the method to the splines that have the similar properties as those of NURBS.
- As the use of spline wavelets provides a very efficient way of multiresolution representation of splines, it would be a significant contribution if it is extended to NURBS. A finite initial subdivision relationship is needed for constructing the wavelets that can be used to solve the multiresolution problem for NURBS.

# **Bibliography**

- [1] D. Esteban A. Croisier and C. Galand. Perfect channel splitting by use of interpolation/decimation/tree decomposition techniques. In International Conference on Information Sciences and Systems, pages 443-446, 1977. Hemisphere, Washington DC.
- [2] Peter J. Burt and Edward H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532-540, April 1983.
- [3] G. Celniker and D. Gossard. Deformable Curve and Surface Finite Elements for Freeform Shape Design. Computer Graphics, 25(4), 1991.
- [4] M. S. Adeeb Chugtai. ANURBS: An Alternative to NURBS of Degree Three. Master's thesis, King Fahd University of Petroleum and Minerals, December 1999.
- [5] E. Cobb. Design of Sculptured Surfaces using the B-spline Representation. PhD thesis, University of Utah, June 1984.

- [6] Snibble S. Herndon K.-Robbins D. Zeleznik R. Conner, D. and A. Van-Dam. Three Dimensional Widgets. In Proceedings of the Symposium on Interactive 3D Graphics, 1992.
- [7] M. Daehlen and T. Lyche. Decompositin of splines. In T. Lyche, L. L. Schumaker, editors, Mathematical Methods in Computer Aided Geometric Design II, pages 135-160, 1992. Academic Press, New York.
- [8] Alan J. Adams David F. Rogers. Mathematical Elements for Computer Graphics. McGraw-Hill International, second edition, 1990.
- [9] Gershon Elber and Craig Gotsman. Multiresolution Control for Nonuniform
   B-spline Curve Editing. In Pacific graphics '95, 1995.
- [10] Tony D. Rose Eric, J. Stollnitz and H. Salesin David. Wavelets for Computer Graphics: Theory and Applicatons. Morgan Kaufman Publishers, San Francisco, 1996.
- [11] Tony D. Rose Eric, J. Stollnitz and H. Salesin David. Wavelts for Computer Graphics: A Primer, Part-2. IEEE Computer Graphics and Application, 15(4):75-85, July 1995.
- [12] Tony D. Rose Eric, J. Stollnitz and H. Salesin David. Wavelts for Computer Graphics: A Primer, Part-1. IEEE Computer Graphics and Application, 15(3):76-84, May 1995.

- [13] Adam Finkelstein and H. Salesin David. Multiresolution Curves. In Proceedings of SIGGRAPH 1994, ACM, New York, pages 261-268, 1994.
- [14] Feiner Hughes Foley, Van Dam and Phillips. Computer Graphics. Prentice Hall International, 1994.
- [15] Dennis Gabor. Theory of Communication. Journal of the Institute of Electrical Engineers, 1946.
- [16] J.A. Gregory and M. Sarfraz. A rational spline with tension. Computer Aided Geometric Design, 7:1-13, 1990.
- [17] Sarfraz M. Gregory, J. A. and P. K Yuen. Interactive curve design using C<sup>2</sup> rational splines. *Computers and Graphics*, 2(153-159), 18.
- [18] A. Grossman and J. Morlet. Decomposition of functions into wavelets of constant shape, and related transforms. In L. Streit, editor, Mathematics and Physics: Lectures on Recent Results. World Scientific, Singapore, 1985.
- [19] Alfred Haar. Zur Theorie der Orthogonalen Funktionen-Systeme. Mathematische Annalen, 1910.
- [20] Z. Habib and M. Sarfraz. A rational cubic spline for the visualization of convex data. The Proceedings of IEEE International Conference on Information Visualization-IV'2001-UK, IEEE Computer Society Press, USA, pages 744-748, 2001.

- [21] F. Hussain and M. Sarfraz. On visualization of statistical data. The Proceedings of International Conference on Information Visualization-IV'97, UK, IEEE Computer Society Press, USA, pages 343-346, 1997.
- [22] E. Fourgeau J. Morlet, G. Arens and D. Giard. Wave Propagation and Sampling
  Theory. *Geophysics*, February 1982.
- [23] Christophe Schlick Laurent Grisoni and Carole Blanc. An Hermitian Approach for Multiresolution Splines. Technical report no. 1192-97, LaBRI, 1997.
- [24] Stephane Mallat. Multiresolution Representation and Wavelets. PhD thesis, University of Pennsylvania, 1988.
- [25] Stephane Mallat. A theory for multiresolution signal decomposition: the wavelet representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(7):674-693, July 1989.
- [26] Yves Meyer. Ondelettes et fonctions splines. Technical Report, Sminaire EDP, cole Polytechnique, Paris, 1986.
- [27] Yves Meyer. Wavelets: Algorithms and applications. SIAM, Philadelphia, 1993.
- [28] Robi Polikar. The Wavelet Tutorial. http://www.public.iastate.edu/~rpolikar/WAVELETS/WTtutorial.html, 1996.

- [29] Flannery B. Teukolsky S. Press, W. and W. Vetterling. Numerical Recipes in C. Cambridge University Press, 1988.
- [30] R. Qu and M. Sarfraz. Efficient method for curve interpolation with monotonicity preservation and shape control. Neural, Parallel, and Scientific Computations (Special Issue on Computer Aided Geometric Design), Dynamic Publishers, USA, 5(1-2):275-288, 1997.
- [31] A. Raheem and M. Sarfraz. Designing using a rational cubic spline with point shape control. Proc. International Conference on Imaging Science, Systems, and Technology (CISST'99), Las Vegas, Nevada, USA, CSREA Press, USA, pages 565-571, 1999.
- [32] Norman Ricker. The Form and Nature of Seismic Waves and the Structure of Seismograms. Geophysics, October 1940.
- [33] Javier Sanches-Rayes. A simple technique for NURBS shape Modification.

  IEEE Computer Graphics and Applications, Jan-Feb 1997.
- [34] Al-Mulhem M. Sarfraz, M. and F. Ashraf. Preserving monotonic shape of the data using piecewise rational cubic functions. Computers and Graphics, 21(1):5-14, 1997.
- [35] Al-Mulhem M. Al-Ghamdi J. Sarfraz, M. and A. Hussain. Quadratic representation to a C<sup>1</sup> rational cubic spline with interval shape control. *Proc. Interna-*

- tional Conference on Imaging Science, Systems, and Technology (CISST'98),

  Las Vegas, Nevada, USA, CSREA Press, USA, pages 322-329, 1998.
- [36] Al-Mulhem M. Al-Ghamdi J. Sarfraz, M. and M.A. Raheem. Modeling by a rational spline with interval shape control. Proc Computer Graphics International '98, Hanover (Germany), IEEE Computer Society Press, pages 730-737, 1998.
- [37] Al-Shishtawi M. S. Sarfraz, M. and A. E. Ibrahim. On generating circles, disks, and spheres. Proc. International Conference on Imaging Science, Systems, and Technology (CISST'98), Las Vegas, Nevada, USA, CSREA Press, USA, pages 330-337, 1998.
- [38] Ashraf-F. Sarfraz, M. and M. Al-Mulhem. Preserving shape of scientific data with monotonic behaviour: A curve method and algorithm. The Proceedings of the CADEX'96: The International Conference and Exhibition on Computer Aided Design, Austria, 1996.
- [39] Butt-S. Sarfraz, M. and M. Z. Hussain. Simulating a model for the scientific data. The Proceedings of International Conference on Operations and Quantitative Management, India, 1997.
- [40] Habib-Z. Sarfraz, M. and M. Hussain. Piecewise interpolation for designing of parameteric curves. The Proceedings of IEEE International Conference on In-

- formation Visualization-IV'98-UK, IEEE Computer Society Press, USA, pages 307-313, 1998.
- [41] Hussain-F. Sarfraz, M. Preserving the monotonic shape of the scientific data using rational cubics. The Proceedings of IASTED International Conference on Modelling and Simulation, USA, pages 169-173, 1996.
- [42] Hussain-M. Sarfraz, M. A rational quadratic b-spline method with shape control. The Proceedings of ISCIS X: The Tenth International Symposium on Computer and information Sciences, Turkey, 2:573-580, 1995.
- [43] Hussain-M. Sarfraz, M. and Z. Habib. Local convexity preserving rational cubic spline curves. , The Proceedings of International Conference on Information Visualization-IV'97, UK, IEEE Computer Society Press, USA, pages 211-218, 1997.
- [44] Hussain-M. Siddiqi S. S. Sarfraz, M. and J. M. Ali. Designing with rational cubic splines. *Journal PERTANIKA*, pages 520–526, 1997.
- [45] Hussain-M. Z. Sarfraz, M. and S. Butt. Nonnegative rational spline interpolation. Proc Computer Graphics International '98, Hanover (Germany), IEEE Computer Society Press, pages 200-204, 1997.
- [46] Hussain-M.Z. Sarfraz, M. and S. Butt. A rational spline for visualizing positive data. The Proceedings of IEEE International Conference on Information

- Visualization-IV'2000-UK, IEEE Computer Society Press, USA, pages 57-62, 2000.
- [47] M. Sarfraz. Interpolatory rational bicubic spline surface with shape control. J. Sc. Res., 20(1-2):43-64, 1991.
- [48] M. Sarfraz. Convexity preserving piecewise rational interpolation for planar curves. Bull. Korean Math. Soc., 29(2):193-200, 1992.
- [49] M. Sarfraz. Interpolatory rational cubic spline with biased, point and interval tension. *Computers and Graphics*, 16(4):427-430, 1992.
- [50] M. Sarfraz. A C<sup>2</sup> rational cubic alternative to the nurbs. Computers and Graphics, 16(1):69-77, 1992.
- [51] M. Sarfraz. Designing of curves and surfaces using rational cubics. Computers and Graphics, 17(5):529-538, 1993.
- [52] M. Sarfraz. A geometric characterization of parametric rational quadratic curves. The Punjab University Journal of Mathematics, 26:41-48, 1993.
- [53] M. Sarfraz. A geometric rational spline with tension controls: an alternative to the weighted nu-spline. The Punjab University Journal of Mathematics, 26:27-40, 1993.

- [54] M. Sarfraz. Rational interpolation preserving the monotonic shape of the data.
  FACTA UNIVERSITATIS, Ser. Mathematics and Informatics, 8:87-95, 1993.
- [55] M. Sarfraz. Shape preserving rational cubic interpolation. Extracta Mathematicae, 8(2-3):106-111, 1993.
- [56] M. Sarfraz. C<sup>2</sup> rational b-spline surfaces with tension control. New advances in CAD and Computer Graphics, Ed: Z. Tang, International academic publishers, Beijing, 1:314-320, 1993.
- [57] M. Sarfraz. Cubic spline curves with shape control. Computers and Graphics, 18(5):707-713, 1994.
- [58] M. Sarfraz. Generalized geometric interpolation for rational cubic splines. Computers and Graphics, 18(1):61-72, 1994.
- [59] M. Sarfraz. Rational interpolation with shape control. Journal of Transfigural Mathematics, 1(1):19-28, 1994.
- [60] M. Sarfraz. A C<sup>2</sup> rational cubic spline which has linear denominator and shape control. Annales Univ. Sci. Budapest, 37:53-62, 1994.
- [61] M. Sarfraz. Curves and surfaces for CAD using C<sup>2</sup> rational cubic splines.
  Engineering with Computers, 11(2):94-102, 1995.

- [62] M. Sarfraz. Designing of 3D rectangular objects. Lecture Notes in Computer Science 1024: Image Analysis Applications and Computer Graphics, Eds.: R. T. Chin, H. H. S. Ip, A. C. Naiman, and T-C. Pong, Springer-Verlag, pages 411-418, 1995.
- [63] M. Sarfraz. Designing of curves with shape control using rational cubic splines.
  TThe Proceedings of 4th International Conference on CAD and Graphics, SPIE,
  2644:20-27, 1995.
- [64] M. Sarfraz. An alternative to the nurbs of degree three. The Proceedings of IS-CIS XI: The Eleventh International Symposium on Computer and information Sciences, Turkey, 2:735-744, 1996.
- [65] M. Sarfraz. Generalized geometric cubic splines. The Proceedings of Computer Graphics International'96-Korea, IEEE Computer Society Press, USA, pages 220-229, 1996.
- [66] M. Sarfraz. A mathematical model for computer graphics. , J. Sc. Res., 25(1-2), 1996.
- [67] M. Sarfraz. Shape preserving spline interpolation for the visualization of scientific data. The Proceedings of the Fifth International Conference of ISOSS, Indonesia, 2, TH-39(1-2), 1996.

- [68] M. Sarfraz. Visualizing the scientific data preserving the convexity. The Proceedings of Seventh International Conference on Engineering Computer Graphics and Descriptive Geometry, Poland, 1996.
- [69] M. Sarfraz. Rational spline interpolation preserving the shape of the monotonic data. The Proceedings of Computer Graphics International'97-Belgium, IEEE Computer Society Press, USA, pages 238-244, 1997.
- [70] M. Sarfraz. Designing of objects using rational quadratic spline with interval shape control. Proceedings of the International Conference on Imaging Science, Systems, and Technology (CISST'99), Las Vegas, Nevada, USA, CSREA Press, USA, pages 558-564, 1999.
- [71] M. Sarfraz. A smooth rational spline for visualizing monotone data. The Proceedings of IEEE International Conference on Information Visualization-IV'99-UK, IEEE Computer Society Press, USA, pages 372-377, 1999.
- [72] M. Sarfraz. Piecewise rational interpolation preserving positive data. The Proc. International Conference on Imaging Science, Systems, and Technology (CISST'2000), Las Vegas, Nevada, USA, CSREA Press, USA, pages 103-109, 2000.
- [73] M. Sarfraz. Visualization of monotone data with cubic splines. The Proc. International Conference on Imaging Science, Systems, and Technology

- (CISST'2001), Las Vegas, Nevada, USA, CSREA Press, USA, pages 764-770, 2001.
- [74] M. Sarfraz and Raheem A. Curve designing using a rational cubic spline with point and interval shape control. The Proceedings of IEEE International Conference on Information Visualization-IV'2000-UK, IEEE Computer Society Press, USA., pages 63-68, 2000.
- [75] M. Sarfraz and Z. Habib. Conic representation of a rational cubic spline. The Proceedings of IEEE International Conference on Information Visualization-IV'99-UK, IEEE Computer Society Press, USA, pages 232-237, 1999.
- [76] Mark J. T. Smith and Thomas P. Barnwell III. Exact reconstruction techniques for tree structured subband coders. *IEEE Transactions on Acoustics*, Speech, and Signal Processing, 34(3):434-441, June 1986.
- [77] Karl Weierstrass. Matheamtische Werke, Volume II. Mayer and Muller, Berlin, 1985.
- [78] W. Welch and A. Witkin. Variational Surface Modeling. Computer Graphics, 26(2), 1992.

## Vita

- Mohammed Ali Siddiqui.
- Born on January 09, 1977.
- Received Bachelor of Engineering (B.E.) degree in Computer Science & Engineering from Osmania University, Hyderabad, India in 1998.
- Joined King Fahd University of Petroleum and Minerals in January 1999.
- Email: masidiqi@ccse.kfupm.edu.sa / siddiqui\_m\_ali@hotmail.com