

University of Kentucky UKnowledge

Theses and Dissertations--Computer Science

**Computer Science** 

2015

# Smooth Subdivision Surfaces: Mesh Blending and Local Interpolation

Jianzhong Wang University of Kentucky, wangjzde@gmail.com

Right click to open a feedback form in a new tab to let us know how this document benefits you.

#### **Recommended Citation**

Wang, Jianzhong, "Smooth Subdivision Surfaces: Mesh Blending and Local Interpolation" (2015). *Theses and Dissertations--Computer Science*. 37. https://uknowledge.uky.edu/cs\_etds/37

This Doctoral Dissertation is brought to you for free and open access by the Computer Science at UKnowledge. It has been accepted for inclusion in Theses and Dissertations--Computer Science by an authorized administrator of UKnowledge. For more information, please contact UKnowledge@lsv.uky.edu.

#### STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

#### **REVIEW, APPROVAL AND ACCEPTANCE**

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Jianzhong Wang, Student Dr. Fuhua Cheng, Major Professor Dr. Miroslaw Truszczynski, Director of Graduate Studies

## SMOOTH SUBDIVISION SURFACES: MESH BLENDING AND LOCAL INTERPOLATION

#### DISSERTATION

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the College of Engineering at the University of Kentucky

> By Jianzhong Wang Lexington, Kentucky

Director: Dr. Fuhua Cheng, Professor of Computer Science Lexington, Kentucky 2015

Copyright<sup>©</sup> Jianzhong Wang 2015

#### ABSTRACT OF DISSERTATION

## SMOOTH SUBDIVISION SURFACES: MESH BLENDING AND LOCAL INTERPOLATION

Subdivision surfaces are widely used in computer graphics and animation. Catmull-Clark subdivision (CCS) is one of the most popular subdivision schemes. It is capable of modeling and representing complex shape of arbitrary topology. Polar surface, working on a triangle-quad mixed mesh structure, is proposed to solve the inherent ripple problem of Catmull-Clark subdivision surface (CCSS).

CCSS is known to be  $C^1$  continuous at extraordinary points. In this work, we present a  $G^2$  scheme at CCS extraordinary points. The work is done by revising CCS subdivision step with Extraordinary-Points-Avoidance model together with mesh blending technique which selects *guiding* control points from a set of regular sub-meshes (named *dominative control meshes*) iteratively at each subdivision level. A similar mesh blending technique is applied to Polar extraordinary faces of Polar surface as well.

Both CCS and Polar subdivision schemes are approximating. Traditionally, one can obtain a CCS limit surface to interpolate given data mesh by iteratively solving a global linear system. In this work, we present a universal interpolating scheme for all quad subdivision surfaces, called Bezier Crust. Bezier Crust is a specially selected bi-quintic Bezier surface patch. With Bezier Crust, one can obtain a high quality interpolating surface on CCSS by parametrically adding CCSS and Bezier Crust. We also show that with a triangle/quad conversion process one can apply Bezier Crust on Polar surfaces as well. We further show that Bezier Crust can be used to generate hollowed 3D objects for applications in rapid prototyping. An alternative interpolating approach specifically designed for CCSS is developed. This new scheme, called One-Step Bi-cubic Interpolation, uses bicubic patches only. With lower degree polynomial, this scheme is appropriate for interpolating large-scale data sets.

In sum, this work presents our research on improving surface smoothness at extraordinary points of both CCS and Polar surfaces and present two local interpolating approaches on approximating subdivision schemes. All examples included in this work show that the results of our research works on subdivision surfaces are of high quality and appropriate for high precision engineering and graphics usage. KEYWORDS: Catmull-Clark, subdivision surface, Polar surface, Bezier Crust, Onestep Bi-cubic Interpolating

Author's signature: Jianzhong Wang

Date: June 19, 2015

## SMOOTH SUBDIVISION SURFACES: MESH BLENDING AND LOCAL INTERPOLATION

By Jianzhong Wang

Director of Dissertation: Fuhua Cheng

Director of Graduate Studies: Miroslaw Truszczynski

Date: June 19, 2015

To my parents, my wife Qian and daughters Yunru and Ann

#### ACKNOWLEDGMENTS

First of all, I would like to thank my thesis advisor, Dr. Fuhua Cheng, for providing an exciting research environment on the cutting edge. Through him, many research ideas have been implemented and presented in this dissertation work. He reviews my research topics and research questions, goes through content and algorithms of my research work and gives valuable comments. Without him, many opportunities have been presented to me would have been impossible.

I would like also to thank other members of my advisory committee, Dr. Jun Zhang, Dr. Jinze Liu, Dr. Ruigang Yang, and Dr. James McDonough, and outside examiner Dr. Tongguang Zhai. Thanks for taking time to review my dissertation work and give deep, insightful comments. I am indeed grateful for their participation.

The arduous progress through course works, quals, and finally the dissertation, has actually been very pleasurable. I would like to thank my wife Qian for her understanding and patience. Without her support and encouragement, it would be impossible for me to finish this work. Finally, I would like to thank my Mom and Dad, they have been always supporting me from the very beginning.

### CONTENTS

Acknowledgments	iii
Contents	iv
List of Figures	vi
List of Tables	viii
Chapter 1 Introduction	1
1.1 Motivation	1
1.2 Contributions	3
1.3 Notations of Surface Smoothness	7
1.4 Summary	8
Chapter 2 Subdivision Surfaces	10
2.1 Overview of Subdivision Surfaces	10
2.2 Catmull-Clark Subdivision Surface	12
2.3 Summary	19
Chapter 3 $G^2$ Bi-cubic Subdivision with Mesh Blending	20
3.1 Introduction	21
3.2 EPA Model	25
3.3 Guided Catmull-Clark Subdivision with Mesh Blending	37
3.4 Eigenstructure of GCCS	41
3.5 Parametrization of GCCSS's	44
3.6 Conditions for Curvature Continuity	47
3.7 Smoothness Evaluation of GCCSS's	49
3.8 Summary	56
Chapter 4 Polar embedded Catmull-Clark Subdivision Surface	57
4.1 Polar Catmull-Clark Mesh	59
4.2 Subdivision Rules on Quad Faces	61
4.3 Guided U-Subdivision	62
4.4 Applying GUS to Polar Parts	68
4.5 Evaluating the PCC surface	76
4.6 Summary	81
Chapter 5 Bezier Crust	83
5.1 Introduction $\ldots$	84
5.2 Previous Works	86
5.3 Bezier Crust on Quad Subdivision Surface	89

$5.4 \\ 5.5$	Implementation and Discussion	94 96		
Chapter	6 G2 Interpolation on Polar Surfaces	99		
6.1	Introduction	99		
6.2	Polar surface and PCCSS	101		
6.3	A Heuristic interpolation scheme on PCCSS	104		
6.4	Implementation and analysis	109		
6.5	Summary	111		
Chapter	7 Thin Shell Modeling on Catmull Clark Subdivision Surface	112		
7 1	Introduction	110		
1.1 7.9	Farlier works	114		
7.2	effecting surface on CCSS with Bozier crust	110		
7.0	behavior of the new effecting surface and discussion	120		
7.4 7.5	Summary	122 197		
1.0	Summary	141		
Chapter	8 One-step Bicubic Interpolation	128		
8.1	Introduction	129		
8.2	Related Works	131		
8.3	One Step Bi-cubic Interpolation	136		
8.4	Mathematical Setup	141		
8.5	Behavior of New Interpolation Scheme	148		
8.6	Summary	151		
Chapter	9 Conclusion and Future Work	153		
9.1	Conclusion	153		
9.2	Future Research	156		
Bibliogr	Bibliography			
0	• •			
Vita .		165		

#### LIST OF FIGURES

$2.1 \\ 2.2 \\ 2.3$	Mesh structure of CCS $\ldots$ $\ldots$ $\ldots$ Subdivision masks of CCS $\ldots$ $\ldots$ $\ldots$ $\Omega$ -Partition of CCS $\ldots$ $\ldots$ $\ldots$	13 14 15
2.4	Subdivision process of a CCS face	16
$3.1 \\ 3.2$	a GCCSS example "Thinker"	$\begin{array}{c} 20\\ 25 \end{array}$
3.3 3.4	CCS $\Omega$ -partition and vertex indexing $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ EPA CCS mesh structure	26 28
3.5	ordering of EPA CCS control points	30
$3.6 \\ 3.7$	Selection of last three layers of $G_1$ Selecting $G_{n,2}$ in the $n^{th}$ EPA CCS	32 33
$3.8 \\ 3.9$	picking 30 control points of the $n^{th}$ EPA CCS on $f_i$	34 39
$3.10 \\ 3.11$	GCCS mesh blending	$40 \\ 42$
3.12 3.13	A GCCSS example of valence 8	$51\\54$
4.1	PCCS example of an airplane	59
4.2 4.3	mesh structure of Polar, CCS and PCC	60 61
4.4	Control mesh conversion for quad faces adjacent to triangular faces	62
4.5 4.6	U-Subdivision	65 65
4.7	Obtaining $\mathbf{P}$ for GUS	69
4.8 4.0	GUS 5 layers	71 71
4.10	GUS mesh blending	74
4.11 4.12	A PCCSS example	75 82
5.1	Two examples of Bezier crust applied on Catmull-Clark subdivision surfaces	83
$5.2 \\ 5.3$	quintic Bezier curve after displacement	90 92
5.4	Displacement of limit surface at corner data points and cross-boundary data points by applying Bezier crust	94
5.5	Top shows limit surface of CCS and its control mesh, bottom shows limit	0 <b>-</b>
5.6	surface and control mesh by applying our new interpolating scheme Two scenaria of Bezier crust, when shown standalone	95 96
5.7	Two Bezier Crust examples	98

6.1	A Polar example with our new interpolating scheme	101
6.2	Polar mesh, CCS mesh and PCC mesh	102
6.3	Convert odd Polar valence to even by one subdivision	103
6.4	Guided U-Subdivision and mesh blending	104
6.5	Difference vectors and selection of Bezier Crust	106
6.6	Polar face conversion by vertex splitting	107
6.7	Difference vectors of Bezier crust on Polar faces	108
6.8	PCC interpolating surface of an airplane	109
6.9	One Polar interpolating example with Bezier Crust shown standalone	110
6.10	Three examples of PCC interpolating surface	112
7.1	an example of hollowed solid with our new offsetting scheme	115
7.2	mesh structure of CCS	117
7.3	Solve creasing issue of offsetting surface	118
7.4	Difference vectors of Bezier Crust	119
7.5	two examples of constructed interpolating surfaces with Bezier Crust on	
	CCSS	120
7.6	CCS control mesh, data mesh and normals	121
7.7	Our new offsetting scheme	122
7.8	Three thin shell examples	124
7.9	Comparison of offsetting surface after 3 recursive subdivisions on CCS mesh	126
8.1	Frog example by our new interpolation scheme	129
8.2	CCS mesh and subdivision masks	132
8.3	An example of a hollowed cube	136
8.4	new scheme of generating CCS interpolating surface	137
8.5	Illustration of various mesh decomposition	138
8.6	1/3 scheme on regular and extraordinary faces	140
8.7	$\Omega$ -Partition of CCS	141
8.8	Base surface and offsetting surfaces	143
8.9	base mesh and offsetting mesh	145
8.10	linear independence of interpolation offsetting data points	146
8.11	A face mask	150
8.12	A statue	151

#### LIST OF TABLES

8.1	Comparing running time of CCS and new one-step interpolation scheme	
	for large data sets.	151

## Chapter 1

## Introduction

### 1.1 Motivation

Computer Graphics is an important, fascinating, and very active field in Computer Science that has revolutionized many endeavors in entertainment, design, education, computer human interaction, and medicine. It is an interdisciplinary field drawing on algorithm design and system building from fields of computer science, applied mathematics, applied physics and electrical engineering. My research work in the field of Computer Graphics focuses on subdivision surface.

Subdivision surfaces have been widely used in surface representation. Compared to traditional spline methods (e.g. Bezier Surface), advantages include simpler to use and can work on arbitrary topology.

Subdivision schemes generate smooth surfaces of arbitrary shape by iteratively applying simple subdivision/refinement rules to the given control meshes. The sequence of meshes generated through this process can quickly converge to a smooth limit surface. In practice, the limit surface will be sufficiently smooth after a few iterations of refinement steps. In addition to representing surface with a sequence of refined meshes, subdivision surfaces like Loop [43] and Catmull-Clark [6] can be parameterized. With parametrization techniques [61, 5, 62, 36], numerically stable algorithms are developed to fast and efficiently generate limit surfaces at arbitrary resolution.

Subdivision schemes mainly use three types of mesh structure: quadrilateral, triangular and hexagonal. quadrilateral faces and triangular faces are most commonly used for practical applications. Subdivision surfaces can be classified into two types: face-split and vertex-split. Vertex-split schemes (midedge [17], biquartic [57]) are not as popular as face-split schemes because they do not generate well behaved surfaces as face-split schemes. In a face-split scheme, vertices of the control mesh are refined recursively. Each vertex of the current control mesh is redefined in the next subdivision level. If the original vertex and its corresponding vertex in the next subdivision step are the same, we call this scheme interpolating (e.g. Modified Butterfly [18], Kobbelt [34]), otherwise the scheme is approximating (e.g. Loop [43], Catmull-Clark [6]). Interpolating is attractive, since vertices in the original control mesh remain in the control meshes in subsequent subdivisions, making subdivision more intuitive. However, surface quality of interpolating schemes is not as good as that of approximating schemes. As a comparison, interpolating schemes such as Modified Butterfly and Kobbelt scheme are  $C^1$  continuous on regular meshes, while approximating schemes such as Catmull-Clark and Loop are  $C^2$  continuous on regular meshes. Among various subdivision schemes, Loop and Catmull-Clark are most widely used on triangular meshes and quadrilateral meshes, respectively.

After reviewing current subdivision schemes, we identified two research questions, and this dissertation work focuses on solving these two questions.

- (1) Can we improve the smoothness at extraordinary points of subdivison surfaces more effectively?
- (2) Is it possible to develop more efficient interpolating scheme on approximating

subdivision surfaces like Catmull-Clark subdivision surface?

With above research questions, we set two objectives in this dissertation work. One objective is to develop necessary geometric models and algorithms to improve smoothness of limit surfaces at extraordinary points of subdivision surfaces. With local control mesh blending model developed, the limit surfaces at extraordinary points of subdivision surfaces like Catmull-Clark and Polar are improved. Another objective is to develop new direct interpolating schemes. One interpolating scheme developed is Bezier Crust. By applying Bezier Crust on arbitrary subdivision surfaces with quadrilateral mesh structure, one can generate an interpolating surface which maintains the surface continuity of underlying subdivision limit surface. The scheme can be used also in surface offsetting and be extended to Polar surface. As an alternative to Bezier Crust, a bi-cubic interpolating scheme was developed, which specifically works on Catmull-Clark subdivision surface. This new scheme avoids to solve a global linear system iteratively. With local control added, the new scheme can change the shape of limit surface locally. In the next section, we list our main contributions in this work.

### **1.2** Contributions

### $G^2$ Bi-cubic Subdivision with Mesh Blending

The Catmull-Clark subdivision (CCS) scheme is re-interpreted and a CCS equivalent, *Extraordinary-Points-Avoidance (EPA) subdivision model* is presented. In this CCS equivalent, EPA subdivision scheme, extraordinary vertices are not explicitly involved in each recursive subdivision step. Hence, it is possible for one to adjust the subdivision process so that eigenstructures for all the extraordinary valences are the same and eigenspaces of the subdivision matrices include only those eigenvalues of a regular face. Based on the EPA model, a generalized CCS scheme, called *Guided Catmull-Clark Subdivision* (GCCS) [65], is developed. The subdivision process of the GCCS scheme is guided by a special layer of control vertices chosen with *mesh blending* technique. Issues related to the resulting limit surface such as parametrization, evaluation, behavior and conditions for curvature continuity at an extraordinary point are studied. By properly choosing control vertices of the special layer recursively, the resulting limit surface is curvature continuous at extraordinary points and  $C^2$  continuous elsewhere. Hence, the classic problem of "how to make a Catmull-Clark subdivision surface  $G^2$  continuous everywhere" is solved.

### Polar embedded Catmull-Clark Subdivision Surface

In this work, a new subdivision scheme with Polar Catmull-Clark (PCC) mesh structure [67] is presented. In this new subdivision scheme, the control mesh consists of two parts, quadrilateral part and triangular part, and one can generate a limit surface that is a CCSS on the quadrilateral part and  $G^2$  on the triangular part. The ripple effect commonly found at high-valence extraordinary points of a CCS surface is improved by replacing high-valence CCS extraordinary faces with triangular Polar faces. The new scheme is valence independent and is stationary. By using the same subdivision mask on both the quadrilateral part and the triangular part, artifacts that occurred in earlier approaches (mismatching subdivision masks, exceedingly huge amount of subfaces produced by the recursive subdivision process) are resolved. Test results show that with the new scheme, one can generate very high quality, curvature continuous subdivision surfaces on the triangular part. Combined with current CCS  $G^2$  schemes, one can generate high quality PCC subdivision surfaces appropriate for most engineering applications.

### Bezier Crust on Quad Subdivision Surfaces

Subdivision surfaces have been widely used in computer graphics and can be classified into two groups, approximating and interpolating. Representatives of approximating schemes are Catmull-Clark (quad) and Loop (triangular). However, one issue remains with approximating schemes, the interpolation of data points are global such that it will be difficult to interpolate when a data point set is large . In this work, we present a local interpolation scheme on quad subdivision surfaces by appending a  $G^2$ Bezier crust [66]. With special construction of bi-quintic Bezier crust, we can avoid to solve a global linear system common in earlier interpolation schemes, such that the computation is local and simple. And we show that this local interpolation scheme will not change the curvature on the boundaries of underlying subdivision patches, such that one can obtain high quality interpolating limit surface for engineering and graphics usage efficiently.

### $G^2$ Interpolation on Polar Surfaces

In this work, we present a  $G^2$  interpolating scheme for Polar surfaces, such that polar surfaces can be used in high precision CAD/CAM applications as well. The new scheme is Bezier crust based, i.e., the interpolating surface is generated by parametrically attaching an especially selected bi-degree 5 Bezier surface to a Polar surface. While Bezier crust based scheme handles quad faces only, we show that through a conversion process, we can handle triangular faces in the Polar part as well. Surface continuity of our new interpolating scheme is consistent with that of the corresponding Polar surface. In case of a Polar embedded Catmull-Clark subdivision surface, the limit surface of our new scheme is  $G^2$  on the Polar part.

## A Heuristic Offsetting Scheme for Catmull-Clark Subdivision Surfaces

In rapid prototyping, a hollowed prototype is preferred and significantly reduces the building time and material consumption in contrast to a solid model. Most rapid prototyping obtains solid thin shell by gradually adding or solidifying materials layer by layer. This is a non-trivial problem to offset a solid which involves finding all selfintersections and filling gaps after raw offsetting. While Catmull-Clark subdivision (CCS) surfaces are widely used in solid modeling, the hollow solid/thin shell problems are not well addressed yet. In this work, we explore earlier methods of obtaining thin shell solid and present a new CCS thin shell solid approach [68]. With this new scheme, one can efficiently avoid creases and handle gaps. The new scheme is heuristic, but inner surface is parametric, so computation of the inner surface is simplified. And with offsetting Bezier crust applied, the inner surface maintains the continuity of the outer surface. The obtained thin shell solid is  $C^2$  continuous everywhere, except at extraordinary points, where it is  $C^1$  continuous.

### **One-step Bicubic Interpolation**

In this work, a new interpolation scheme for Catmull-Clark subdivision (CCS) surface is introduced. The construction process is based on two techniques: surface offsetting and mesh decomposition. The surface offsetting technique ensures the shape of the data set is faithfully resembled, so the method has the power of a global method; the mesh decomposition technique enables us to solve the problem using a one-step, local approach, instead of solving a global linear system using an iterative approach. The decomposition process of an offsetting mesh preserves the number of extraordinary points in the CCS mesh. Therefore, the interpolating surface preserves the continuity of a CCS surface. Furthermore, with heuristic selection of offsetting mesh, the computed interpolating surface can also maintain the same normal and curvature at interpolating points as CCS surface. Test results show that interpolation of large-scale data sets can be efficiently handled with our new method and the generated interpolating surfaces have very high surface quality. Hence, the new scheme is especially suitable for applications in reverse engineering and 3D printing.

### **1.3** Notations of Surface Smoothness

Subdivision surfaces are typically constructed with piecewise surface patches. Two surface patches are said to meet with  $C^n$  or  $n^{th}$  order *parametric continuity*, if derivatives up to order n are continuous at the boundary between two patches.

Parametric continuity of surface requires the piecing together of surface patches so that a given number of parametric derivatives match at the boundaries between surface patches. Parametric continuity is an important measure of subdivision surface smoothness. However, there are some cases where parametric continuity does not apply, e.g. at extraordinary points of Catmull-Clark subdivision surface. To remedy this situation, a parametrization independent measure of *geometric continuity* is introduced.

Geometric continuity is first introduced to curve [2] [22] [47] [59], and then extended to surface [12] [13] [15] [14] [24] [25] [27] [28] [31] [32] [42] [69]. Two parametric surface patches F and G are said to meet with  $G^n$  or  $n^{th}$  order geometric continuity, if there exist reparametrizations of these two surface patches,  $\tilde{F}$  and  $\tilde{G}$ , and they meet with  $C^n$  continuity. The most commonly used surface measures are  $G^0$ ,  $G^1$  and  $G^2$ .

 $G^0$  continuity: positional continuity, i.e. two surfaces share a common edge/point.

- $G^1$  continuous: tangent plane continuity, i.e. two surfaces share the same tangent plane at their common edge/point.
- $G^2$  continuous: curvature continuity, i.e. two surfaces are curvature continuous at their common edge/point.

### 1.4 Summary

In this chapter, we present our motivation on this dissertation work and our main contributions.

The remaining parts of this dissertation work are organized as follows:

**Chapter 2**: we provide an overview of subdivision surfaces and introduce details of Catmull-Clark subdivision surface, incl. parametrization, eigen analysis and surface evaluation.

**Chapter 3**: we show a  $G^2$  scheme with mesh blending at extraordinary points of Catmull-Clark subdivision surface.

**Chapter 4**: we show a  $G^2$  scheme with mesh blending at Polar extraordinary points of Polar subdivision surface.

**Chapter 5**: we present a  $G^2$  interpolation scheme for subdivision surfaces on mesh of quadrilateral faces. The new scheme is called Bezier Crust, a specially selected biquintic Bezier surface.

**Chapter 6**: we present a  $G^2$  interpolation scheme with Bezier Crust on Polar surface after face conversion.

**Chapter 7**: we present a scheme to generate hollowed 3D object with Catmull-Clark subdivision surface and offsetting Bezier Crust.

**Chapter 8**: we present a  $G^2$  bi-cubic interpolation scheme on Catmull-Clark subdivision surface. This scheme is alternative to Bezier Crust, but with lower degree

of polynomial. Such that this scheme is more appropriate for processing large-scale data sets.

Chapter 9: We summarize our work and present the directions for our future research.

Copyright<sup>©</sup> Jianzhong Wang, 2015.

## Chapter 2

## **Subdivision Surfaces**

Subdivision surfaces have been widely used in graphical modeling and animation. In this chapter, we review most known stationary subdivision schemes, and introduce the concept, eigen analysis and surface evaluation of Catmull-Clark subdivision surface.

### 2.1 Overview of Subdivision Surfaces

The concept of subdivision surfaces are first described in 1978 by Catmull and Clark [6] and Doo and Sabin [17]. Subdivision schemes generate a sequence of meshes by iteratively applying simple refinement rules to the given control mesh. This sequence of meshes quickly converge to a smooth limit surface. The subdivision surfaces become popular in modeling and representing complex shape of 3D objects because of their high visual quality, easy implementation, and stability in numerical computation.

In this research work, we focus on *stationary subdivision schemes* [7]. A subdivision scheme is said to be stationary if its refinement rules do not depend on the subdivision level, i.e., the control points of mesh in current subdivision level will be computed solely by the control points of mesh in last subdivision level. Using of stationary schemes make the implementation highly efficient, and make it easier to

analyze the behavior of subdivision surfaces.

Most stationary subdivision schemes can be classified with four criteria [72]:

- 1. the type of refinement rules
- 2. the type of generated mesh
- 3. whether the scheme is approximating or interpolating
- 4. smoothness of the limit surfaces on regular mesh

Different subdivision schemes have different refinement rules, but generally, there are only three types of mesh faces, triangular, quadrilateral and hexagonal. Loop scheme [43] is the most popular subdivision schemes on triangular mesh. Catmull-Clark scheme [6] and Doo-Sabin scheme [16] are best known on quad mesh. Few subdivision schemes handle hexagonal mesh, Claes et al. [11] presents a corner cutting scheme which treat hexagonal mesh as a dual to triangular mesh.

The refinement/subdivision rules often can be specified by subdivision masks. Subdivision mask is defined as a graph marked with coefficients on vertices of current subdivision level to compute a new vertex of the next subdivision level. There are two main approaches to perform mesh refinement, named *face split* and *vertex split*. In the first approach, subdivision mask is provided for each new vertex corresponding to each old vertex, with additional subdivision masks defined on newly inserted vertices (e.g. for Loop [43], new edge points, and for Catmull-Clark [6], new edge and face points). So in *face split* schemes, the old vertices are retained while new vertices are inserted to split the old faces. In the second case, as a contrast, for each old vertex, new vertices are inserted, one for each face adjacent to this old vertex. A new face is then created on each old vertex in the new mesh. So in the *vertex split* schemes, old vertices are removed in the new mesh after subdivision, i.e. there are no subdivision masks defined on existing vertices. Popular *vertex split* subdivision schemes include Doo-Sabin [16], Midedge [54], and Biquartic [57]. Given a control mesh, if the iteratively refined mesh passes through all control points in this given mesh, then this subdivision scheme is *interpolating*, otherwise it is *approximating*. Since vertex split schemes don't retain old vertices in mesh refinement, this classification is not applicable for them. In face split schemes, Loop scheme [43] and Catmull-Clark scheme [6] are approximating, while Butterfly scheme [18] and Kobbelt scheme [34] are interpolating.

On regular faces of above introduced subdivision schemes, Loop, Catmull-Clark and Biquartic are  $C^2$  continuous, Modified Butterfly, Kobbelt, Doo-Sabin, and Midedge are  $C^1$  continuous. A visually smooth 3D object modeling generally requires the surface to be  $C^2$  continuous with lower degree polynomial for surface representing, so Loop and Catmull-Clark are most popular subdivision schemes for triangular and quad meshes respectively.

As stated, Catmull-Clark and Loop schemes are both  $C^2$  on regular faces, which make them the most popular schemes for quad and triangular meshes respectively. In this work, we focus on improvement over quad subdivision schemes like Catmull-Clark subdivision surface (CCSS) and its extension Polar subdivision surface. So in the next section, we briefly review the exact behavior of CCSS.

### 2.2 Catmull-Clark Subdivision Surface

A Catmull-Clark subdivision (CCS) surface is the limit surface of a sequence of subdivision steps performed on a given control mesh. At each step, new vertices are added and old vertices are updated. The valence of a vertex is the number of edges meeting at the vertex. A vertex with valence four is called a *regular* vertex, otherwise an *extraordinary* vertex. A mesh face is regular if all vertices are regular, otherwise, it is called extraordinary face. CCS vertices are classified into three categories: vertex points, edge points, and face points. A popular way to index the control vertices is



Figure 2.1: Mesh structure of CCS, (a) regular face, (b) extraordinary face

shown on the left side of Fig. 2.1 for a regular face and the right side for an extraordinary face, where V is a vertex point,  $E_i$ s are edge points,  $F_i$ s are face points, and  $I_{i,j}$ s are inner ring control vertices. New vertices within each subdivision step are generated as follows:

$$V' = \alpha_N V + \beta_N \sum_{i=1}^N E_i / N + \gamma_N \sum_{i=1}^N F_i / N$$
  

$$E'_i = \frac{3}{8} (V + E_i) + \frac{1}{16} (E_{i+1} + E_{i-1} + F_i + F_{i-1})$$
  

$$F'_i = \frac{1}{4} (V + E_i + E_{i+1} + F_i)$$
(2.1)

where N is the valence of vertex V, with  $\alpha_N = 1 - \frac{7}{4N}$ ,  $\beta_N = \frac{3}{2N}$ , and  $\gamma_N = \frac{1}{4N}$ . Equation (2.1) is the math representation of CCS subdivision masks (Fig. 2.2)

### Parametrization of CCSS

The CCS limit surface can be obtained by performing equation (2.1) sequentially. However, to show the limit surface, it needs also the unit normal defined on each



Figure 2.2: Subdivision masks of CCS, (a) face point, (b) edge point, (c) vertex point

vertex of refined mesh. This is calculated through parametrization on CCSS.

On a regular face, CCSS can be represented by the bi-cubic B-Spline patch. We define S(u, v) as the CCS limit surface on regular face with parametric values (u, v),  $u, v \in [0, 1]$ , such that a regular bi-cubic B-spline patch with parameters u and v can be expressed as

$$S(u,v) = \begin{bmatrix} 1 \ u \ u^2 \ u^3 \end{bmatrix} M \mathbf{P} M^T \begin{bmatrix} 1 \ v \ v^2 \ v^3 \end{bmatrix}^T$$
(2.2)

where **P** is a  $4 \times 4$  matrix of control points  $P_{ij}$ ,  $1 \le i, j \le 4$  ( $P_{ij}$  takes the value in 16 control points of a regular face, as shown on the left of Fig. 2.1), M is the B-Spline coefficient matrix and  $M^T$  is its transpose.

With parametrization of CCSS on regular face (equation (2.2)), one can explicitly compute the limit points of the CCS limit surface on regular faces with arbitrary resolution without iteration.

The parametrization of CCSS on extraordinary face is not developed until Stam's work [61]. Boier et al. [5] and Lai and Cheng [38] further improve the parametrization to be more efficient.

As shown in [36], the parametrization of an extraordinary face  $f_i$  of valence N is as belows. First we define the limit surface of  $f_i$  as S(u, v), and initial control mesh on  $f_i$  as G with size 2N+8, the three regular bi-cubic B-Spline patches after the *n*-th CCS as  $S_{n,b}$ ,  $n \ge 1$ , b = 1, 2, 3. The  $\Omega$ -partition is defined by:  $\Omega_{n,b}$ ,  $n \ge 1$ , b = 1, 2, 3,



Figure 2.3:  $\Omega$ -Partition of CCS

with

$$\Omega_{n,1} = \left(\frac{1}{2^n}, \frac{1}{2^{n-1}}\right] \times \left[0, \frac{1}{2^n}\right] 
\Omega_{n,2} = \left(\frac{1}{2^n}, \frac{1}{2^{n-1}}\right] \times \left(\frac{1}{2^n}, \frac{1}{2^{n-1}}\right] 
\Omega_{n,3} = \left(0, \frac{1}{2^n}\right] \times \left(\frac{1}{2^n}, \frac{1}{2^{n-1}}\right]$$
(2.3)

For any  $(u, v) \in [0, 1] \times [0, 1]$ ,  $(u, v) \neq (0, 0)$ , there is an  $\Omega_{n,b}$  containing (u, v). We can find the value of S(u, v) by mapping  $\Omega_{n,b}$  to the unit square  $[0, 1] \times [0, 1]$  and finding the corresponding  $(\overline{u}, \overline{v})$ . After the mapping, we compute  $S_{n,b}$  at  $(\overline{u}, \overline{v})$ . The value of S(0, 0) is the limit of extraordinary vertices.

In the above process, n and b can be computed by:

$$n(u, v) = \min\{\lceil \log_{\frac{1}{2}} u \rceil, \lceil \log_{\frac{1}{2}} v \rceil\}$$
  

$$b(u, v) = \begin{cases} 1, & \text{if } 2^{n}u \ge 1 \text{ and } 2^{n}v < 1 \\ 2, & \text{if } 2^{n}u \ge 1 \text{ and } 2^{n}v \ge 1 \\ 3, & \text{if } 2^{n}u < 1 \text{ and } 2^{n}v \ge 1 \end{cases}$$
(2.4)

The mapping from  $\Omega_{n,b}$  to the unit square is defined by

$$(u, v) \to (\overline{u}, \overline{v}) = (\phi(u), \phi(v))$$
, with

$$\phi(t) = \begin{cases} 2^n t, & \text{if } 2^n t \le 1\\ 2^n t - 1, & \text{if } 2^n t > 1 \end{cases}$$
(2.5)



Figure 2.4: Subdivision process of a CCS face

The CCSS S(u, v) can be expressed as follows

$$S(u,v) = W^T(\overline{u},\overline{v})MG_n^b$$
(2.6)

where  $\boldsymbol{G}_{n}^{b}$  is the control point vector of  $S_{n,b}$ , W(u,v) is the 16-power-basis vector with  $W^{T}(u,v) = [1, u, v, u^{2}, uv, v^{2}, u^{3}, u^{2}v, uv^{2}, v^{3}, u^{3}v, u^{2}v^{2}, uv^{3}, u^{3}v^{2}, u^{2}v^{3}, u^{3}v^{3}]$ . M is the B-spline coefficient matrix. We can express  $W^{T}(\overline{u}, \overline{v})$  as follows

$$W^{T}(\overline{u},\overline{v}) = W^{T}(u,v)K^{n}D_{b}$$
(2.7)

where K is a diagonal matrix, with

K = Diag(1, 2, 2, 4, 4, 4, 8, 8, 8, 8, 16, 16, 16, 32, 32, 64),

and  $D_b$ , an upper triangular matrix depending on b only, maps  $(\overline{u}, \overline{v})$  to (u, v). So we can rewrite the CCSS as

$$S(u,v) = W^T(u,v)K^n D_b M \boldsymbol{G}_n^b$$
(2.8)

By defining  $\boldsymbol{G}_{n}^{b} = \overline{P}_{b}\boldsymbol{G}_{n}$ , where  $\overline{P}_{b}$  is a picking matrix choosing 16 control points from the 2N + 17 control points  $\boldsymbol{G}_{n}$ , we get

$$S(u,v) = W^{T}(u,v)K^{n}D_{b}M\overline{P}_{b}\boldsymbol{G}_{n},$$
(2.9)

with

$$\boldsymbol{G}_n = \overline{A} A^{n-1} \boldsymbol{G} \tag{2.10}$$

where A and  $\overline{A}$  are the CCS subdivision matrix and extended subdivision matrix with size  $(2N + 8) \times (2N + 8)$  and  $(2N + 17) \times (2N + 8)$  respectively (illustrated in Fig. 2.4).

The above equations provide a formal parametrization for an extraordinary face  $f_i$  of a CCSS. However, since the computation of  $G_n$  involves multiplication of subdivision matrix A, this parametrization is a costly process.

### **Eigen Analysis**

The subdivision matrix A of a CCSS face with valence N is obtained by subdivision rules shown in (2.1) and illustrated in Fig. 2.4.

With the work of Ball and Storry [1] and Stam [61], an exact solution to obtain arbitrary point on the CCS limit surface can be developed without iteratively computing control points at all consequent subdivision levels. The evaluation of  $A^{n-1}$ can be simplified by eigen decomposition on A,  $A = X^{-1}\Lambda X$ , where  $\Lambda$  is a diagonal matrix of eigenvalues of A, X is an invertible matrix whose columns are the corresponding eigenvectors. Such that  $A^{n-1} = X^{-1}\Lambda^{n-1}X$ . With eigen decomposition, we can compute  $A^{n-1}$  directly without (n-1) multiplications.

With discrete Fourier transform (DFT), eigen decomposition of A can be expressed as

$$A_{\omega} = X_{\omega}^{-1} \Lambda X_{\omega} \tag{2.11}$$

The eigenvalues in  $\lambda$  can be computed via discrete Fourier transform. As shown in [61] and [36], there are totally N + 6 different eigen values,

$$\lambda_{0} = (4\alpha_{N} - 1 + \sqrt{16\alpha_{N}^{2} - 8\alpha_{N} + 8\beta_{N} - 3})/8$$

$$\lambda_{1} = (4\alpha_{N} - 1 - \sqrt{16\alpha_{N}^{2} - 8\alpha_{N} + 8\beta_{N} - 3})/8$$

$$\lambda_{2\omega} = (c_{\omega} + 5 + \sqrt{c_{\omega}^{2} + 10c_{\omega} + 9})/16$$

$$\lambda_{2\omega+1} = (c_{\omega} + 5 - \sqrt{c_{\omega}^{2} + 10c_{\omega} + 9})/16$$

$$\lambda_{N+1} = 1$$

$$\lambda_{N+2} = 1/8$$

$$\lambda_{N+3} = 1/16$$

$$\lambda_{N+4} = 1/32$$

$$\lambda_{N+5} = 1/64$$
(2.12)

where  $1 \le \omega \le N/2$ ,  $c_{\omega} = \cos(2\pi\omega/N)$ ,  $\alpha_N$  and  $\beta_N$  are defined in (2.1).

### **Evaluation of Catmull-Clark Subdivision Surface**

With parametrization of CCS limit surface on regular and extraordinary faces, we can easily derive that a CCSS is  $C^2$  everywhere except at extraordinary points [61] [1].

As illustrated in (2.12), since  $\lambda_{N+1} > \lambda_0 > \lambda_1$  and  $\lambda_2 > \lambda_i$  for  $i \in [3, N]$ , with the proof in [1], we can show that CCSS is  $C^1$  at extraordinary points.

So, CCS limit surface is  $C^2$  everywhere except at extraordinary points, where it is  $C^1$  only.

## 2.3 Summary

In this chapter, we review some best known subdivision schemes and four classification criteria. We also show Catmull-Clark subdivision surfaces in more details, refinement rules, parametrization, eigen analysis and limit surface evaluation.

In the next chapters, we will present our research results on improvement of surface quality at extraordinary points of Catmull-Clark subdivision scheme and Polar subdivision scheme. We will further show our work on smooth interpolation schemes for both CCS surface and Polar surface.

## Chapter 3

# $G^2$ Bi-cubic Subdivision with Mesh Blending



Figure 3.1: A GCCSS example "Thinker". Left side: control mesh and limit surface; right side: control mesh, limit surface and limit surface colored with Gaussian curvature of the enlarged nose.

In this work, the Catmull-Clark subdivision (CCS) scheme is re-interpreted and a CCS equivalent, *Extraordinary-Points-Avoidance (EPA) subdivision model* is presented. In this EPA model, extraordinary vertices are not involved in the control meshes of newly generated regular sub-patches in each recursive subdivision step. Based on the EPA model, a generalized CCS scheme, called *guided Catmull-Clark* subdivision (GCCS), is developed. The subdivision process of the GCCS scheme is guided by a special layer of control vertices that are chosen from the CCS refinement on a set of regular meshes called *dominative control meshes*. Issues related to the resulting limit surface such as parametrization, evaluation, behavior and conditions for curvature continuity at an extraordinary point are studied. By properly choosing choosing dominative control meshes, the resulting limit surface is curvature continuous at extraordinary points and  $C^2$  continuous elsewhere. Hence, the classic problem of "how to make a Catmull-Clark subdivision surface  $G^2$  continuous everywhere" is solved. With the new scheme, one can generate high quality  $G^2$  subdivision surfaces for all engineering and graphics applications with only bi-cubic B-spline patches.

### **3.1** Introduction

Subdivision is a powerful technique in modeling/representing free-form shapes. As one of the most popular subdivision schemes, Catmull-Clark subdivision [6] is based on tensor product bi-cubic B-Splines.

### Behavior of Catmull-Clark Subdivision Surface

As noted in Chapter 2, a Catmull-Clark subdivision surface (CCSS) is the limit surface of a sequence of subdivision steps performed on a given control mesh. At each step new vertices are introduced and old vertices are updated. The *valence* of a vertex is the number of edges meeting at the vertex. A vertex with valence four is called a *regular vertex*, all other vertices are called *extraordinary vertices*. A mesh face is called an *extraordinary face* if one of its vertices is an extraordinary vertex. Otherwise, it is called a *regular face*. A point on the limit surface is called an *extraordinary point* if it is the limit of a sequence of extraordinary vertices.

Properties of the limit surface have been analyzed by Doo and Sabin [17] in terms of eigenvalues of the subdivision matrices. Conditions for tangent plane continuity at extraordinary points have been given by Ball and Storry [1]. It has been shown by Jos Stam [61] that the surface and all its derivatives can be evaluated in terms of a set of eigenbasis functions, and theoretical foundation for the development of parametric representation has been given by him as well. It is proved by Prautzsch [56] and Reif [58] that it is not possible to construct a  $G^2$  CCSS with non-zero curvature at extraordinary points, and it is also pointed out by them that a bi-degree-6 subdivision scheme is required to obtain  $G^2$  surfaces.

From the aforementioned works, one can conclude that a CCSS is  $C^2$  continuous everywhere except at extraordinary points, where it is  $C^1$  continuous and also known to be curvature unbounded.

#### **Previous Works on Extraordinary Points**

Many researches have been performed to improve the smoothness of a CCSS at extraordinary points.

Prautzsch [56] modifies the subdivision scheme near extraordinary points to generate a  $C^2$  everywhere surface with zero curvature at extraordinary points. Zorin [73] and Levin [40] present schemes to yield a  $C^2$  continuous surface by blending the limit surface with a low degree polynomial defined over the characteristic map in the vicinity of each extraordinary point. Loop and Schaefer [44] present a second order smooth filling of an n-valence Catmull-Clark spline ring with n bi-septic patches, with shape optimization for free parameters. Peters and Karčiauskas [53] introduce a guided subdivision scheme that uses a Bezier surface as a guide for each subdivision step, and a  $C^2$  accelerated Bi-3 guided subdivision that uses  $2^m$  sub-faces in the m-th level for surface patches surrounding extraordinary points. In the second case, they show that although this scheme is not practical for Catmull-Clark surfaces, it can be applied to a polar configuration.

However, these solutions are not completely satisfactory yet. Blending the limit surface with a precomputed curvature continuous surface patch is not flexible in surface representation. Filling the holes with bi-degree-6 patches will result in higher Gaussian curvature near the extraordinary points and make the limit surface unattractive The bi-cubic subdivision scheme that generates  $2^m$  sub-patches in the *m*-th subdivision is also undesired.

By going through these previous approaches on extraordinary points, the following question arises naturally: Does a simple, low degree subdivision scheme that can generate  $G^2$  surfaces at extraordinary points of CCSS exist?

### **Our New Scheme**

Prautzsch and Reif's work [56] [58] already showed that a bi-degree-6 subdivision scheme is required to obtain  $G^2$  continuity at extraordinary points, and the condition for their conclusion is that all eigenvalues of the subdivision matrices have equal algebraic and geometric multiplicities. Hence, to obtain a  $G^2$ -continuous surface with less than bi-degree-6 subdivision, one needs to use a non-regular subdivision mask, such as the approach of Peters and Karciauskas [53]. Peters and Karciauskas' approach is not attractive due to the fact that it generates exponentially many sub-patches after the recursive subdivision steps, but it shows us that getting a bi-cubic  $G^2$ -continuous limit surface is possible, without contradicting the conclusion of Prautzsch and Reif.

In this work, we introduce a new subdivision model, called the *Extraordinary*-*Points-Avoidance* (EPA) model. The EPA subdivision scheme is equivalent to CCS in the sense that its limit surfaces are the same as those of CCS, but extraordinary vertices are not involved in the control meshes of newly generated regular sub-patches
in the recursive subdivision steps. This property is important in that it allows one to adjust the subdivision process so that eigenstructures for all the extraordinary valences could be the same and eigenspaces of the subdivision matrices could include only those eigenvalues of a regular face. Based on the EPA model, a generalized CCS, called the *guided Catmull- Clark subdivision* (GCCS) is developed. The subdivision process of the GCCS scheme is guided by a special layer of control vertices that can be set by the user. By properly choosing control vertices of the special layer, the resulting limit surface is curvature continuous at extraordinary points and  $C^2$ continuous elsewhere. The new scheme also provides flexibility in surface rendering.

In this  $G^2$  GCCS scheme, we do not use Catmull-Clark subdivision mask for extraordinary point, but inserting successive control vertices from regular bi-cubic B-spline patches. Key features of our approach are as follows:

- it is stationary, weights of subdivision are all regular, independent of valence of the extraordinary point;
- 12 sub-faces are generated at each subdivision level, therefore the implementation is practical and equally efficient as CCS;
- it is parameterizable, and its eigenvalues are the same as those of the regular patches;
- curvature at an extraordinary point of valence N is determined by  $2N 9 \times 9$  regular control meshes;
- the limit surface is curvature continuous at extraordinary points and C<sup>2</sup>-continuous elsewhere.

Fig. 3.1 shows an example of the new scheme. On the right side, the enlarged limit surface of the *nose* colored with Gaussian curvature is smooth and with non-zero curvature at an extraordinary point (green indicates positive, blue negative).

## 3.2 EPA Model

In this section, we introduce the concept of a new subdivision model, called the Extraordinary-Points-Avoidance (EPA) Model, due to the fact that the this subdivision model avoids extraordinary points in the subdivision process, so that the formulation of a new  $G^2$  continuous subdivision scheme would be possible.

#### **Revisit CCS**



Figure 3.2: Control meshes of CCS faces. Left side: a regular face; right side: an extraordinary face

Recall that the CCS scheme divides the control vertices into three categories: vertex points, edge points, and face points. A popular way to index the control vertices of a subdivision face is shown on the left side of Fig. 3.2 for a regular face and the right side for an extraordinary face, where V is a vertex point,  $E_i$ 's are edge points,  $F_i$ 's are face points and  $I_{i,j}$ 's are inner ring control vertices. New vertices within each subdivision step are generated as follows:

$$V' = \alpha_N V + \beta_N \sum_{i=1}^N E_i / N + \gamma_N \sum_{i=1}^N F_i / N$$
  

$$E'_i = \frac{3}{8} (V + E_i) + \frac{1}{16} (E_{i+1} + E_{i-1} + F_i + F_{i-1})$$
  

$$F'_i = \frac{1}{4} (V + E_i + E_{i+1} + F_i)$$
(3.1)

where N is the valence of vertex V, with  $\alpha_N = 1 - \frac{7}{4N}$ ,  $\beta_N = \frac{3}{2N}$ , and  $\gamma_N = \frac{1}{4N}$ . These subdivision rules work for the inner ring control vertices as well since these control vertices and the subsequently generated new control vertices are also vertex, edge or face points.



Figure 3.3: Left side: CCS  $\Omega$ -partition; right side: indexing and ordering of vertices after subdivision

A CCS generates 2N+17 new control vertices (red control mesh shown on the right of Fig. 3.3) from the 2N + 8 control vertices that define the current extraordinary patch. The new control vertices define three uniform B-spline patches (Fig. 3.3 left) and an extraordinary patch. Therefore, only the 2N + 8 new control vertices that define the extraordinary sub-patch of the current extraordinary patch has to be further subdivided. The subdivision process hence can be formulated as follows:

$$C_n = AC_{n-1} = A^n C_0$$
 and  
 $\bar{C}_n = \bar{A}C_{n-1} = \bar{A}A^{n-1}C_0, n \ge 1$  (3.2)

where  $C_0$  is the set of 2N + 8 control vertices defining the original surface patch shown in Fig. 3.2,  $C_n$  is the set of 2N + 8 control vertices defining the extraordinary sub-patch generated after the  $n^{th}$  subdivision step, and  $\boldsymbol{A}$  is the  $(2N+8) \times (2N+8)$ matrix that performs the subdivision to get these 2N+8 new vertices.  $\bar{\boldsymbol{C}}_n$  is the set of all 2N + 17 control vertices generated after the  $n^{th}$  subdivision step, and  $\bar{\boldsymbol{A}}$  is the  $(2N+17) \times (2N+8)$  extended subdivision matrix.

 $C_n$  and  $\overline{C}_n$  can be explicitly listed as

$$\boldsymbol{C}_{n}^{T} = (V, E_{i}, ..., E_{i-1}, F_{i}, ..., F_{i-1}, I_{i-1,4}, ..., I_{i+1,2}),$$
  
and  
$$\bar{\boldsymbol{C}}_{n}^{T} = (\boldsymbol{C}_{n}^{T}, O_{i-1.6}, ..., O_{i+1,2}),$$
(3.3)

respectively. The 9 extra control vertices  $O_{i,j}$ 's in  $\overline{C}_n$  are called the outer ring control vertices.

Properties of the subdivision matrices A and  $\overline{A}$  have been discussed in [1] and [61].

#### EPA Model of CCS

In this section, we present the Extraordinary-Points-Avoidance (EPA) Model. With this model, the CCS is revised to obtain an equivalent subdivision scheme that excludes the extraordinary vertices in the subdivision step.

Note that a sufficient condition for a subdivision surface to be  $G^2$  continuous at the extraordinary points [56] is that eigenvalues of each subdivision matrix take either the form of  $\lambda_2^{\alpha} \lambda_3^{\beta}$  or are smaller than  $\lambda_3^2$ , where  $\lambda_2$  and  $\lambda_3$  are the second and third largest eigenvalues of the subdivision matrix of the extraordinary face, and  $\alpha, \beta \in N$ . By eigendecomposition of the CCS subdivision matrix [1, 61, 37], one obtains N + 6 eigenvalues:  $\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{N+6}$ , listed in decreasing order with  $\lambda_1 = 1$ , and  $\lambda_2$  and  $\lambda_3$  being the second and third largest eigenvalues. These eigenvalues in general do not satisfy the condition given by Prautzsch [56]. A possible solution to get a  $G^2$ - continuous limit surface for the CCS scheme is to modify the subdivision process so that eigenvalues of the subdivision matrix would satisfy Prautzsch's condition.

On the other hand, note that, as a special case, a regular face is  $G^2$  continuous, with different eigenvalues 1,  $\frac{1}{2}$ ,  $\frac{1}{4}$ ,  $\frac{1}{8}$ ,  $\frac{1}{16}$ ,  $\frac{1}{32}$ ,  $\frac{1}{64}$ , and these eigenvalues satisfy Prautzsch's condition for a  $G^2$  surface. Hence, if one can develop a subdivision process with a unique eigenstructure for all the extraordinary valences and the eigenspace includes only those eigenvalues of a regular face, then the Prautzsch's condition would be satisfied.

Furthermore, note that the reason the eigenstructure of the current CCS scheme is valence-dependent is because of the first rule of equation (3.1). If one can find a way to override the first rule of (3.1) without involving extraordinary points in the subdivision steps, we will have a chance to achieve our goal of obtaining a unique eigenstructure for arbitrary extraordinary faces.



Figure 3.4: Left side: 12 sub-faces generated by a CCS equivalent EPA subdivision; right side: obtain  $n^{th}$  EPA CCS control points by applying one more CCS on an  $n^{th}$  CCS control mesh.

The revision of CCS starts with building an equivalent scheme of CCS first. Note that if one performs one more CCS on the 3 regular sub-faces obtained after a CCS (regions shadowed in red on the right of Fig. 3.3), one gets 12 regular sub-faces (regions shadowed in blue on the right of Fig. 3.4). Control vertices defining these 12 sub-faces are important, because these vertices are obtained by regular CCS subdivision and, therefore, are not valence-dependent and yet they preserve the impact of the extraordinary points on the shape of the limit surface; there are totally 45 of them (the blue control mesh underneath the black line on the right side of Fig. 3.4). Indeed, while V' is involved in all the control meshes defining the three regular subfaces shown in Fig. 3.3, the new V' obtained after one more CCS is not involved in the control mesh of any of the 12 regular sub-faces shown in Fig. 3.4 at all. Nevertheless, the limit surface obtained this way (performing two CCS's during each subdivision step) is exactly the same as the one obtained with only one CCS performed during each subdivision step, since the 12 sub-faces obtained after performing one more CCS are just the tessellations of the 3 regular sub-faces obtained after each CCS. For the convenience of subsequent references, we will call this equivalent scheme of CCS the EPA CCS, a short-hand for Extraordinary-Points-Avoidance CCS.

We can put EPA CCS in a mathematical setting as follows: given an extraordinary face  $f_i$  of valence N, the 45 control vertices defining the 12 sub-faces obtained after the  $n^{th}$  EPA CCS are expressed as

$$G_n | f_i = (g_{n,1}, ..., g_{n,45})^T.$$

These control vertices are ordered in the way shown on the left side of Fig. 3.5.

With this particular ordering, we can regroup  $G_n$  into five layers:

$$\begin{aligned} \boldsymbol{G}_{n,1} | f_i &= (g_{n,1}, ..., g_{n,5})^T, & \boldsymbol{G}_{n,2} | f_i &= (g_{n,6}, ..., g_{n,12})^T, \\ \boldsymbol{G}_{n,3} | f_i &= (g_{n,13}, ..., g_{n,21})^T, & \boldsymbol{G}_{n,4} | f_i &= (g_{n,22}, ..., g_{n,32})^T, \\ \boldsymbol{G}_{n,5} | f_i &= (g_{n,33}, ..., g_{n,45})^T. \end{aligned}$$

 $G_n$  is obtained by performing one more CCS on 24 control vertices picked from  $\bar{C}_n$ .



Figure 3.5: Left side: ordering of the 45 control vertices generated by an EPA CCS. Right side: insertion of new control vertices during the  $n^{th}$  EPA CCS, colored circles are control vertices generated by previous EPA CCS and solid dots are those by current EPA CCS

We denote these 24 vertices as  $\widetilde{C}_n$ ,

$$\widetilde{\boldsymbol{C}}_n = (\widetilde{c}_{n,1}, \ \widetilde{c}_{n,2}, \dots \ \widetilde{c}_{n,24})^T$$

with the vertices ordered in the way shown on the right side of Fig. 3.4. These 24 control vertices can be regrouped into 4 layers:

$$\begin{split} \widetilde{\boldsymbol{C}}_{n,1} &= (\widetilde{c}_{n,1}, \dots \ \widetilde{c}_{n,3})^T, \qquad \widetilde{\boldsymbol{C}}_{n,2} &= (\widetilde{c}_{n,4}, \dots \ \widetilde{c}_{n,8})^T, \\ \widetilde{\boldsymbol{C}}_{n,3} &= (\widetilde{c}_{n,9}, \dots \ \widetilde{c}_{n,15})^T, \qquad \widetilde{\boldsymbol{C}}_{n,4} &= (\widetilde{c}_{n,16}, \dots \ \widetilde{c}_{n,24})^T. \end{split}$$

We have

$$\widetilde{C}_n = P_1 \overline{C}_n,$$
  
 $G_n | f_i = A_1 \widetilde{C}_n, \ n \ge 1,$ 
(3.4)

where  $A_1$  is a subdivision matrix of dimension  $45 \times 24$  and  $P_1$  is a picking matrix of dimension  $24 \times (2N + 17)$ . The entries of  $P_1$  are defined as follows: for each  $\tilde{c}_{n,j}$  (on the right side of Fig. 3.4), compare it with control vertices in  $\bar{C}_n$  (equation (3.3)), if it is in the  $k^{th}$  position of  $\bar{C}_n$ , then entry (j, k) of  $P_1$  is set to 1. All other entries of  $P_1$  are set to zero. Since control vertices in  $\widetilde{C}_n$  are selected from  $\overline{C}_n$  and control vertices in  $G_n$  are obtained by performing one CCS on  $\widetilde{C}_n$ , all these control vertices are CCS control vertices. We can further conclude that  $\widetilde{C}_n$  is a subset of the  $n^{th}$  CCS control mesh, and  $G_n$  is a subset of the  $(n+1)^{st}$  CCS control mesh. From equations (3.2) and (3.4) (comparing the ordering of  $\widetilde{C}_n$  shown on the right of Fig. 3.4 with that of  $G_n$  shown on the left of Fig. 3.5), we can further derive that

$$G_{n,1} = \widetilde{C}_{n+1,2}, \quad G_{n,2} = \widetilde{C}_{n+1,3}, \quad G_{n,3} = \widetilde{C}_{n+1,4}.$$
 (3.5)

Equation (3.4) transforms the CCS scheme to its equivalent form by performing one more midpoint insertion in each CCS step. However, since one of the control vertices in  $\tilde{C}_n$  is an extraordinary point, the involvement of extraordinary points in the subdivision process is not completely avoided yet. Further work is needed.

#### Recursive definition of $G_{n,3}$ , $G_{n,4}$ and $G_{n,5}$

 $A_1$  of equation (3.4) can be decomposed into

$$\boldsymbol{A}_{1} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ 0 & \boldsymbol{A}_{2} \end{bmatrix}$$
(3.6)

where  $A_{11}$ ,  $A_{12}$  and  $A_2$  are matrices of dimension  $12 \times 3$ ,  $12 \times 21$  and  $33 \times 21$ , respectively.  $A_{11}$  and  $A_{12}$  together form the subdivision matrix for  $G_{n,1}$  and  $G_{n,2}$ ,  $A_2$  is the subdivision matrix for  $G_{n,3}$ ,  $G_{n,4}$  and  $G_{n,5}$ , with

$$(\boldsymbol{G}_{n,3}^{T}, \boldsymbol{G}_{n,4}^{T}, \boldsymbol{G}_{n,5}^{T})^{T} = \boldsymbol{A}_{2} (\widetilde{\boldsymbol{C}}_{n,2}^{T}, \widetilde{\boldsymbol{C}}_{n,3}^{T}, \widetilde{\boldsymbol{C}}_{n,4}^{T})^{T}.$$
(3.7)

Through equations (3.5) and (3.7), we obtain a recursive definition on the last three layers of EPA CCS control vertices, that is, one obtains control vertices in  $G_{n,3}$ ,  $G_{n,4}$  and  $G_{n,5}$  by performing one midpoint knot insertion on  $G_{n-1,1}$ ,  $G_{n-1,2}$  and  $G_{n-1,3}$ . Using this relationship,  $G_1$  can be written as

$$\boldsymbol{G}_{1}|f_{i} = (\boldsymbol{G}_{1,1}^{T}, \boldsymbol{G}_{1,2}^{T}, (\boldsymbol{A}_{2}P_{2}\widetilde{\boldsymbol{C}}_{1})^{T})^{T}, \qquad (3.8)$$

and when  $n \geq 2$ ,

$$\boldsymbol{G}_{n}|f_{i} = (\boldsymbol{G}_{n,1}^{T}, \boldsymbol{G}_{n,2}^{T}, (\boldsymbol{A}_{2}P_{3}\boldsymbol{G}_{n-1})^{T})^{T}, \qquad (3.9)$$

where  $P_2$  and  $P_3$  are picking matrices of dimension  $21 \times 24$  and  $21 \times 45$ , respectively, with the right 21 columns of  $P_2$  and left 21 columns of  $P_3$  forming an identity matrix  $I_{21}$  and all other columns being zero (to pick the last three layers of  $\tilde{C}_n$  or the first three layers of  $G_{n-1}$ ). Fig. 3.6 and Fig. 3.5 show picking process of equations (3.8) and (3.9), respectively.



Figure 3.6: Selection of last three layers of  $G_1$ . Left side: initial control mesh of  $f_i$  and inserting new vertices by applying a CCS; right side: obtaining 33 control vertices by applying one more CCS.

Equation (3.9) together with the initial conditions set in equation (3.8) show that one can obtain the last three layers of the  $n^{th}$  EPA CCS control mesh:  $\boldsymbol{G}_{n,3}$ ,  $\boldsymbol{G}_{n,4}$  and  $\boldsymbol{G}_{n,5}$ , from the previous EPA CCS control mesh  $\boldsymbol{G}_{n-1}$ , recursively, and the process is independent of the extraordinary vertices.

#### Representing $G_{n,1}$ and $G_{n,2}$

Equation (3.5) shows the control vertices in  $G_{n,1}$  are the control vertices in  $\tilde{C}_{n+1,2}$ (edge/face points generated by the  $(n + 1)^{st}$  CCS). In this step we leave them unchanged, with

$$\boldsymbol{G}_{n,1}|f_i = \widetilde{\boldsymbol{C}}_{n+1,2} \tag{3.10}$$

The control vertices in  $G_{n,2}$  are obtained by performing one CCS on  $\tilde{C}_{n,1}$ ,  $\tilde{C}_{n,2}$ , and  $\tilde{C}_{n,3}$  (black circles shown on the left side of Fig. 3.6). By equation (3.5),  $\tilde{C}_{n,2}$ , and  $\tilde{C}_{n,3}$  are  $G_{n-1,1}$ ,  $G_{n-1,2}$ , respectively. Also note that the control vertices in  $\tilde{C}_{n,1}$ are actually  $E_{i-1}^{(n)}$ ,  $V^{(n)}$  and  $E_{i+2}^{(n)}$  (edge points and vertex point in the  $(n)^{th}$  CCS). In the following, , we show that the control vertices in  $G_{n,2}$  can be reversely computed from  $G_{n,1}$  together with  $G_{n-1,1}$  and  $G_{n-1,2}$ .



Figure 3.7: Selecting  $G_{n,2}$  in the  $n^{th}$  EPA CCS: left side shows generation of  $G_{n,2}[2]$  (green dot), right side shows generation of  $G_{n,2}[3]$ ,  $G_{n,2}[4]$  and  $G_{n,2}[5]$  (green dots).

Since the control vertices in  $G_{n,2}$  are either vertex points or edge points, we will show here how to rewrite  $G_{n,2}[2]$  and  $G_{n,2}[3]$  only, other control vertices can be adjusted similarly.

By equation (3.4), as shown in Fig. 3.7, we put  $\tilde{C}_{n,1}[1]$  and  $\tilde{C}_{n,1}[2]$  (purple circles) into the control mesh of  $G_{n-1,1}$  (black circles) and  $G_{n-1,2}$  (blue circles). If we treat  $G_{n,1}[2]$  as a new edge point and  $G_{n,1}[3]$  as a new face point, then by equation (3.1) of the CCS, control vertex  $\tilde{C}_{n,1}[2]$  can be derived from  $G_{n,1}[3]$  and its surrounding control vertices in  $G_{n-1}$ , and control vertex  $\tilde{C}_{n,1}[1]$  can be derived from  $G_{n,1}[2]$ ,  $\widetilde{C}_{n,1}[2]$  and its surrounding control vertices in  $G_{n-1}$ , as follows:

$$\widetilde{C}_{n,1}[2] = 4G_{n,1}[3] - G_{n-1,1}[2] - G_{n-1,1}[3] - G_{n-1,1}[4]$$

$$\widetilde{C}_{n,1}[1] = 16G_{n,1}[2] - 6\widetilde{C}_{n,1}[2] - 6G_{n-1,1}[2] - G_{n-1,1}[1]$$

$$- G_{n-1,1}[3] - G_{n-1,1}[4]$$
(3.11)

we can then calculate  $G_{n,2}[2]$ ,  $G_{n,2}[3]$ ,  $G_{n,2}[4]$ ,  $G_{n,2}[5]$  by performing a CCS on the regular control mesh consisting of

$$\begin{bmatrix} \widetilde{\boldsymbol{C}}_{n,1}[1] & \widetilde{\boldsymbol{C}}_{n,1}[2] & \boldsymbol{G}_{n-1,1}[4] & \boldsymbol{G}_{n-1,2}[6] \\ \boldsymbol{G}_{n-1,1}[1] & \boldsymbol{G}_{n-1,1}[2] & \boldsymbol{G}_{n-1,1}[3] & \boldsymbol{G}_{n-1,2}[5] \\ \boldsymbol{G}_{n-1,2}[1] & \boldsymbol{G}_{n-1,2}[2] & \boldsymbol{G}_{n-1,2}[3] & \boldsymbol{G}_{n-1,2}[4] \end{bmatrix}$$



Figure 3.8: Graph showing the picking of 30 control points (black dots) of the  $n^{th}$  EPA CCS on  $f_i$ , i = 1, ..., N. Putting all 30N control points together, they are the control points for surface ring of  $n^{th}$  EPA CCS at extraordinary points.

By canceling out  $\widetilde{C}_{n,1}[2]$  and  $\widetilde{C}_{n,1}[1]$  using equation (3.11), we get an expression for  $G_{n,2}[2]$ ,  $G_{n,2}[3]$ ,  $G_{n,2}[4]$  and  $G_{n,2}[5]$  in terms of the first layer of the current control mesh and the first two layers of the previous control mesh. One can get  $G_{n,2}[1]$ ,

 $G_{n,2}[6]$  and  $G_{n,2}[7]$  using a similar process on  $f_{i-1}$  and  $f_{i+1}$ . After this construction process, we get all seven control points of  $G_{n,2}$  as follows:

$$\boldsymbol{G}_{n,2}|f_i = A_7 \begin{bmatrix} \boldsymbol{G}_{n,1} \\ \boldsymbol{G}_{n-1,1} \\ \boldsymbol{G}_{n-1,2} \end{bmatrix}, \quad where$$
(3.12)

$$A_{7} = \begin{bmatrix} \frac{1}{4} & 0 & 0 & 0 & 0 & \frac{5}{16} & \frac{5}{16} & 0 & 0 & 0 & \frac{1}{16} & \frac{1}{16} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 & 0 & \frac{5}{64} & \frac{15}{32} & \frac{5}{64} & 0 & 0 & \frac{1}{64} & \frac{3}{32} & \frac{1}{64} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & \frac{5}{16} & \frac{5}{16} & 0 & 0 & 0 & \frac{1}{16} & \frac{1}{16} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{16} & 0 & 0 & 0 & \frac{5}{64} & \frac{35}{64} & \frac{5}{64} & 0 & 0 & \frac{1}{64} & \frac{3}{32} & \frac{1}{64} & \frac{3}{32} & \frac{1}{64} & 0 \\ 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & \frac{5}{16} & \frac{5}{16} & 0 & 0 & 0 & 0 & \frac{1}{16} & \frac{1}{16} & 0 \\ 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & \frac{5}{64} & \frac{15}{32} & \frac{5}{64} & 0 & 0 & 0 & 0 & \frac{1}{64} & \frac{3}{32} & \frac{1}{64} \\ 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & \frac{5}{16} & \frac{5}{16} & 0 & 0 & 0 & 0 & 0 & \frac{1}{64} & \frac{3}{32} & \frac{1}{64} \end{bmatrix}$$

#### Formulation of EPA CCS

In the above, we have defined all the five layers of  $G_n$ . We can put equations (3.8), (3.9), (3.10) and (3.12) into one piece, with

$$\boldsymbol{G}_{1}|f_{i} = (\widetilde{\boldsymbol{C}}_{2,2}^{T}, (A_{7} \begin{bmatrix} \widetilde{\boldsymbol{C}}_{2,2} \\ \widetilde{\boldsymbol{C}}_{1,2} \\ \widetilde{\boldsymbol{C}}_{1,3} \end{bmatrix})^{T}, (\boldsymbol{A}_{2}P_{2}\widetilde{\boldsymbol{C}}_{1})^{T})^{T}, \qquad (3.13)$$

and when  $n \ge 2$ ,

$$\boldsymbol{G}_{n}|f_{i} = (\widetilde{\boldsymbol{C}}_{n+1,2}^{T}, (A_{7} \begin{bmatrix} \widetilde{\boldsymbol{C}}_{n+1,2} \\ \boldsymbol{G}_{n-1,1} \\ \boldsymbol{G}_{n-1,2} \end{bmatrix})^{T}, (\boldsymbol{A}_{2}P_{3}\boldsymbol{G}_{n-1})^{T})^{T}$$
(3.14)

For simplicity, we put the above equations into a matrix form as follows.

$$\boldsymbol{G}_{n} = S_{1} \widetilde{\boldsymbol{C}}_{n+1,2} + S_{2} \boldsymbol{G}_{n-1}, \qquad n \ge 1$$
with 
$$\boldsymbol{G}_{0} = \begin{bmatrix} P_{2} \widetilde{\boldsymbol{C}}_{1} \\ 0 \end{bmatrix}$$
(3.15)

where  $S_1$  is a matrix of dimension  $45 \times 5$  and  $S_2$  is a matrix of dimension  $45 \times 45$ .

If we define  $A_7$  and  $A_2$  in block structure with each block corresponding to one layer of control vertices, we have

$$A_{7} = \begin{bmatrix} A_{7,1} & A_{7,2} & A_{7,3} \end{bmatrix}, \quad A_{2} = \begin{bmatrix} A_{2,11} & A_{2,12} & 0 \\ A_{2,21} & A_{2,22} & A_{2,23} \\ 0 & A_{2,32} & A_{2,33} \end{bmatrix}$$

where  $A_{7,1}$ ,  $A_{7,2}$ ,  $A_{7,3}$ ,  $A_{2,11}$ ,  $A_{2,12}$ ,  $A_{2,21}$ ,  $A_{2,22}$ ,  $A_{2,23}$ ,  $A_{2,32}$ ,  $A_{2,33}$  are matrices of dimensions  $7 \times 5$ ,  $7 \times 5$ ,  $7 \times 7$ ,  $9 \times 5$ ,  $9 \times 7$ ,  $11 \times 5$ ,  $11 \times 7$ ,  $11 \times 9$ ,  $13 \times 7$  and  $13 \times 9$ , respectively, then

$$S_{1} = \begin{bmatrix} I_{5} \\ A_{7,1} \\ 0 \end{bmatrix}, \quad S_{2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ A_{7,2} & A_{7,3} & 0 & 0 & 0 \\ A_{2,11} & A_{2,12} & 0 & 0 & 0 \\ A_{2,21} & A_{2,22} & A_{2,23} & 0 & 0 \\ 0 & A_{2,32} & A_{2,33} & 0 & 0 \end{bmatrix}$$
(3.16)

where  $I_5$  is an identity matrix of dimension 5.

The EPA CCS limit surfaces generated by equation (3.15) are exactly the same as those of CCS. However the new scheme has the advantage that it limits the impact of extraordinary points to the 1<sup>st</sup> layer control vertices  $G_{n,1}$  (edge/face points in  $(n+1)^{st}$  CCS control mesh defined in equation (3.8)) only. This is the key feature of EPA CCS.

By adjusting the conditions set for  $G_{n,1}$ , it will then be possible to construct a subdivision scheme that is curvature continuous at the extraordinary points.

## 3.3 Guided Catmull-Clark Subdivision with Mesh Blending

In EPA CCS, instead of applying equation (3.10), if we leave  $G_{n,1}$  undefined, we obtain a generalized EPA CCS as follows:

$$\boldsymbol{G}_{n} = S_{1}\boldsymbol{G}_{n,1} + S_{2}\boldsymbol{G}_{n-1}, \qquad n \ge 1$$
with 
$$\boldsymbol{G}_{0} = \begin{bmatrix} P_{2}\tilde{\boldsymbol{C}}_{1} \\ 0 \end{bmatrix}.$$
(3.17)

We can further expand it as follows:

$$\boldsymbol{G}_{n} = (S_{2})^{n} \boldsymbol{G}_{0} + \sum_{k=1}^{n} (S_{2})^{n-k} S_{1} \boldsymbol{G}_{k,1}, \ n \ge 1.$$
(3.18)

**Theorem 3.1.**  $G_n$  in equation (3.17) can have an eigenstructure with eigenvalues in the powers of  $\frac{1}{2}$  iff  $G_{n,1}$  has an eigenstructure with eigenvalues in the powers of  $\frac{1}{2}$ .

*Proof.* This can be proved by analyzing the eigenstructure of a regular face. For a regular face in the EPA CCS with valence N = 4,  $G_n$  is obtained by performing one CCS on its CCS counterpart  $\tilde{C}_n$  (equation (3.4)), so its eigenvalues are in the powers of  $\frac{1}{2}$ . From equation (3.15), we can conclude that  $S_2$  must have an eigenstructure with all eigenvalues in the powers of  $\frac{1}{2}$ . From equation (3.18), we can then conclude that  $G_n$  has an eigenstructure with eigenvalues in the powers of  $\frac{1}{2}$ .

Equation (3.17) is a recurrence formula. Control vertices in  $G_{n,1}$  determine curvature at the extraordinary point and guide the subdivision process towards the extraordinary point, so we name them **guiding control vertices**, and we name this generalized EPA CCS *Guided Catmull-Clark Subdivision* (GCCS) to reflect the fact that the convergence to the limit surface process is guided by these vertices.

GCCS Control meshes are generated recursively. After each GCCS step, control vertices for 12 new regular sub-faces are generated. The portion of the limit surface corresponding to these sub-faces is obviously  $C^2$  continuous since this portion is formed by 12 regular B-spline sub-patches For borders of the limit surface corresponding to borders between consecutive subdivision steps, since the last three layers of the current control mesh are obtained by performing a CCS on the first three layers of the last control mesh, the limit surface is also  $C^2$  along these borders. So a GCCS limit surface is  $C^2$  everywhere except at extraordinary points.

To achieve curvature continuity at the extraordinary points for the GCCS scheme, as pointed out earlier, one possibility is to have an eigenstructure of the control mesh that is valence independent for each extraordinary patch. Also, eigenvalues of the subdivision matrices need to satisfy the condition set by Prautzsch and Reif. Eigen analysis shows that the subdivision matrix  $S_2$  with  $G_{n,1}$  undefined has eigenvalues of  $\frac{1}{8}$ ,  $\frac{1}{16}$ ,  $\frac{1}{32}$  and  $\frac{1}{64}$  when  $n \geq 3$ . If  $G_{n,1}$  can be defined to have an eigenstructure with eigenvalues in the powers of  $\frac{1}{2}$ , then the eigenstructure of the GCCS control mesh for an extraordinary patch will also have eigenvalues in the powers of  $\frac{1}{2}$ . With these special eigenvalues, it will then be possible to generate a limit surface that is curvature continuous at the extraordinary points.

Note that eigenvalues of the subdivision matrix of a regular face are in the powers of  $\frac{1}{2}$ . This inspired us to map CCS control vertices of a regular face to control vertices in  $G_{n,1}$ , since these control vertices satisfy the above condition on eigenvalues.

For a vertex of valence N, we define 2N sets of *dominative control meshes*  $\hat{C}_k$ , k = 1, 2, ..., 2N. Each  $\hat{C}_k$  consists of 9 vertices, a vertex point, 4 edge points and 4 face points, as follows

$$\hat{C}_k = (\hat{V}_k, \, \hat{E}_{k,1}, \, \hat{E}_{k,2}, \, \hat{E}_{k,3}, \, \hat{E}_{k,4}, \, \hat{F}_{k,1}, \, \hat{F}_{k,2}, \, \hat{F}_{k,3}, \, \hat{F}_{k,4}).$$

The structure of  $\hat{C}_k$  (as shown in Fig. 3.9) is the same as that of the control mesh of a regular face, excluding the seven inner control vertices (left side of Fig. 3.2).

Each control vertex in  $\boldsymbol{G}_{n,1}$  is determined by one  $\hat{C}_k$ .  $\boldsymbol{G}_{n,1}[1]$  is determined by  $\hat{C}_{2i-2}$ ,



Figure 3.9: Control mesh  $\hat{C}_k$ .

 $G_{n,1}[2]$  by  $\hat{C}_{2i-1}$ ,  $G_{n,1}[3]$  by  $\hat{C}_{2i}$ ,  $G_{n,1}[4]$  by  $\hat{C}_{2i+1}$ , and  $G_{n,1}[5]$  by  $\hat{C}_{2i+2}$ . Initialization and picking of the 2N dominative control meshes will be discussed in the section of **Conditions for curvature continuity**.

We define  $\hat{C}_{k}^{(n)}$  as the control mesh obtained after n regular CCS's on  $\hat{C}_{k}$ , with  $\hat{C}_{k}^{(n)} = (\hat{V}_{k}^{(n)}, \hat{E}_{k,1}^{(n)}, \hat{E}_{k,2}^{(n)}, \hat{E}_{k,3}^{(n)}, \hat{E}_{k,1}^{(n)}, \hat{F}_{k,1}^{(n)}, \hat{F}_{k,2}^{(n)}, \hat{F}_{k,3}^{(n)}, \hat{F}_{k,4}^{(n)}), \ k=1,...,2N. \ \hat{C}_{k}^{(n)}$  can be expressed as

$$\hat{C}_{k}^{(n)} = A\hat{C}_{k}^{(n-1)} = \dots = A^{n}\hat{C}_{k}^{(0)}, \ n \ge 1$$
(3.19)

with  $\hat{C}_k^{(0)} = \hat{C}_k$  and A is a 9 × 9 regular midpoint knot insertion subdivision matrix.

Once the 2N dominative control meshes are determined, we can define the five guiding control vertices in  $G_{n,1}$ .

Theoretically we can map any edge or face point of a *dominative control mesh* to a *guiding control vertex*. However, for simplicity of eigen analysis, we map the first edge points of the dominative control meshes  $\hat{C}_{k}^{(n+1)}$  (k=2i-2,...,2i+2) to the *guiding* 



Figure 3.10: Mapping of control vertices in  $\hat{C}_k^{(n+1)}$  (k = 2i - 2, ..., 2i + 2) to the five control vertices in  $\boldsymbol{G}_{n,1}$ .

control vertices in  $G_{n,1}$  (see Fig. 3.10), as follows:

$$\boldsymbol{G}_{n,1}[1] = \hat{E}_{2i-2,1}^{(n+1)} = P_4 \hat{C}_{2i-2}^{(n+1)}, \quad \boldsymbol{G}_{n,1}[2] = \hat{E}_{2i-1,1}^{(n+1)} = P_4 \hat{C}_{2i-1}^{(n+1)},$$
$$\boldsymbol{G}_{n,1}[3] = \hat{E}_{2i,1}^{(n+1)} = P_4 \hat{C}_{2i}^{(n+1)}, \quad \boldsymbol{G}_{n,1}[4] = \hat{E}_{2i+1,1}^{(n+1)} = P_4 \hat{C}_{2i+1}^{(n+1)},$$
$$\boldsymbol{G}_{n,1}[5] = \hat{E}_{2i+2,1}^{(n+1)} = P_4 \hat{C}_{2i+2}^{(n+1)}$$
(3.20)

with  $P_4 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ .

Since all the guiding control vertices defined above have the same subdivision matrix A, we can define a 9 × 5 matrix  $\hat{D}_i$  with its columns corresponding to the 5 dominative control meshes of  $f_i$ , and a 9 × 5 matrix  $\hat{D}_i^{(n+1)}$  with its columns corresponding to the control meshes after (n + 1) CCS's, respectively,

$$\hat{D}_i = [\hat{C}_{2i-2}, \hat{C}_{2i-1}, \hat{C}_{2i}, \hat{C}_{2i+1}, \hat{C}_{2i+2}]$$
(3.21)

$$\hat{D}_{i}^{(n+1)} = [\hat{C}_{2i-2}^{(n+1)}, \hat{C}_{2i-1}^{(n+1)}, \hat{C}_{2i}^{(n+1)}, \hat{C}_{2i+1}^{(n+1)}, \hat{C}_{2i+2}^{(n+1)}]$$
$$= A^{n+1}\hat{D}_{i}$$
(3.22)

From equations (3.20) - (3.22), we get

$$\boldsymbol{G}_{n,1}|f_i = [P_4 \hat{D}_i^{(n+1)}]^T = [P_4 A^{n+1} \hat{D}_i]^T$$
(3.23)

here  $X^T$  denotes the transpose of matrix X.

With all the control vertices in  $G_{n,1}$  selected with mesh blending technique, the picking process of the *n*-th GCCS control mesh  $G_n$  is now complete. By equations (3.18) and (3.23), we get control vertices in all subsequent GCCS control meshes and are able to construct an iteratively generated subdivision surface. Fig. 3.11 shows a *marker cap* represented by both a GCCSS and a CCSS. The enlarged Gaussian curvature data mesh of the GCCSS shows an improved curvature and mesh structure at an extraordinary point over that of the CCSS.

### 3.4 Eigenstructure of GCCS

Equations. (3.18) and (3.23) involves power forms of  $S_2$  and A, respectively, which are not desirable for limit surface representation. In order to simplify the computation process, an eigendecomposition of A and  $S_2$  will be needed.

Decomposition of A is straightforward. It can be expressed as

$$A = X \wedge_A X^{-1}$$

where  $\wedge_A$  is a diagonal matrix filled with eigenvalues of A, X is an invertible matrix whose columns are the corresponding eigenvectors and  $X^{-1}$  is its inverse. A has five eigenvalues. We use  $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$  to denote these eigenvalues. Their values are  $1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}$  and  $\frac{1}{16}$ , respectively. For each  $1 \leq m \leq 5$ , we define  $\theta_m$  as a  $9 \times 9$  matrix with all the entries being zero except the one corresponding to  $\lambda_m$  where the value is 1. We then have

$$A = \sum_{m=1}^{5} \lambda_m X \theta_m X^{-1} = \sum_{m=1}^{5} \lambda_m T_m$$
(3.24)



Figure 3.11: Top three items (left to right): control mesh (red block indicates enlarged area), GCCSS limit surface, CCSS limit surface of a marker cap; bottom three items (top to bottom): enlarged control mesh, GCCSS data mesh and CCSS data mesh with Gaussian curvature shown (green positive, blue negative) of an extraordinary region.

where  $T_m = X \theta_m X^{-1}$ .

A direct eigendecomposition is not practical for a  $45 \times 45$  matrix  $S_2$ . However, we observe that

$$S_{2}^{n} = S_{3}(A_{7,3})^{n-3}S_{4}, \ n \ge 3$$
(3.25)  
with  $S_{3} = \begin{bmatrix} 0 \\ A_{7,3}^{2} \\ A_{2,12}A_{7,3} \\ A_{2,22}A_{7,3} + A_{2,23}A_{2,12} \\ A_{2,32}A_{7,3} + A_{2,33}A_{2,12} \end{bmatrix}$ , and  $S_{4} = \begin{bmatrix} A_{7,2} & A_{7,3} & 0 & 0 & 0 \end{bmatrix}$ . where  $S_{3}$ 

is a  $45 \times 7$  matrix with the first 5 rows set to zero and  $S_4$  is a  $7 \times 45$  matrix with the right 33 columns set to zero.

So eigendecomposition of  $S_2$  can be reduced to that of  $A_{7,3}$ . Eigen decomposition of  $A_{7,3}$  can be obtained easily as

$$A_{7,3} = Y \wedge_{A_{7,3}} Y^{-1}$$

where  $\wedge_{A_{7,3}}$  is a diagonal matrix filled with eigenvalues of  $A_{7,3}$ , Y is an invertible matrix whose columns are the corresponding eigenvectors and  $Y^{-1}$  is its inverse.  $A_{7,3}$ has four eigenvalues, denoted  $\lambda_4, \lambda_5, \lambda_6, \lambda_7$ . The values are  $\frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}$ , respectively. For each  $4 \leq m \leq 7$ , we define  $\theta'_m$  as a 7 × 7 matrix with all the entries being zero except the one corresponding to  $\lambda_m$  where the value is 1. We then obtain

$$A_{7,3} = \sum_{m=4}^{7} \lambda_m Y \theta'_m Y^{-1} = \sum_{m=4}^{7} \lambda_m T'_m$$
(3.26)

where  $T'_m = Y \theta'_m Y^{-1}$ .

With the above eigendecomposition of A and  $A_{7,3}$ ,  $G_n$ ,  $n \ge 4$  can be redefined as follows:

$$G_{n} = \sum_{m=4}^{7} (\lambda_{m}^{n-3}S_{3}T_{m}'S_{4}C_{i}^{(0)}) + \sum_{t=1}^{5} (\lambda_{t}^{n+1}S_{1}[P_{4}T_{t}\hat{D}_{i}]^{T}) + \sum_{t=1}^{5} (\lambda_{t}^{n}S_{2}S_{1}[P_{4}T_{t}\hat{D}_{i}]^{T}) + \sum_{t=1}^{5} (\lambda_{t}^{n-1}S_{2}^{2}S_{1}[P_{4}T_{t}\hat{D}_{i}]^{T}) + \sum_{k=0}^{n-4} \sum_{m=4}^{7} \sum_{t=1}^{5} (\lambda_{m}^{n-k-4}\lambda_{t}^{k+2}S_{3}T_{m}'S_{4}S_{1}[P_{4}T_{t}\hat{D}_{i}]^{T})$$

$$(3.27)$$

We define  $B_{m,t,n-4} = \sum_{k=0}^{n-4} (\lambda_m^{n-k-4} \lambda_t^{k+2})$ , such that

$$B_{m,t,n-4} = \begin{cases} \lambda_m^{n-4} \lambda_t^2 \frac{1 - (\frac{\lambda_t}{\lambda_m})^{n-3}}{1 - \frac{\lambda_t}{\lambda_m}}, & \text{if } m < t \\ (n-3)\lambda_m^{n-2}, & \text{if } m = t \\ \lambda_t^{n-2} \frac{1 - (\frac{\lambda_m}{\lambda_t})^{n-3}}{1 - \frac{\lambda_m}{\lambda_t}}, & \text{if } m > t \end{cases}$$
(3.28)

With equation (3.28), equation (3.27) can be further simplified as

$$G_{n} = \sum_{m=4}^{7} (\lambda_{m}^{n-3} S_{3} T_{m}' S_{4} C_{i}^{(0)}) + \sum_{t=1}^{5} ((\lambda_{t}^{n+1} S_{1} + \lambda_{t}^{n} S_{2} S_{1} + \lambda_{t}^{n-1} S_{2}^{2} S_{1}) [P_{4} T_{t} \hat{D}_{i}]^{T}) + \sum_{m=4}^{7} \sum_{t=1}^{5} (B_{m,t,n-4} S_{3} T_{m}' S_{4} S_{1} [P_{4} T_{t} \hat{D}_{i}]^{T})$$
(3.29)

By using equation (3.15) for n = 1, 2, 3 and equation (3.21) for  $n \ge 4$ , we can compute all 45 control points of each  $G_n$  in constant time.

## 3.5 Parametrization of GCCSS's

With the availability of an explicit control mesh computation process after each GCCS, the parametrization of an extraordinary face  $f_i$  is actually quite simple. First we define the limit surface of  $f_i$  as S(u, v), the twelve regular bi-cubic B-Spline patches after the *n*-th GCCS as  $S_{n,b}$ ,  $n \ge 1$ , b = 1, 2, ..., 12. The  $\Omega$ -partition is defined by:

 $\Omega_{n,b}, n \ge 1, b = 1, 2, .., 12$ , with

$$\Omega_{n,1} = \left(\frac{3}{2^{n+1}}, \frac{1}{2^{n-1}}\right] \times \left[0, \frac{1}{2^{n+1}}\right] \\
\Omega_{n,2} = \left(\frac{3}{2^{n+1}}, \frac{1}{2^{n-1}}\right] \times \left(\frac{1}{2^{n+1}}, \frac{1}{2^{n}}\right] \\
\Omega_{n,3} = \left(\frac{3}{2^{n+1}}, \frac{1}{2^{n-1}}\right] \times \left(\frac{1}{2^{n}}, \frac{3}{2^{n+1}}\right] \\
\Omega_{n,4} = \left(\frac{3}{2^{n+1}}, \frac{1}{2^{n-1}}\right] \times \left(\frac{3}{2^{n+1}}, \frac{1}{2^{n-1}}\right] \\
\Omega_{n,5} = \left(\frac{1}{2^{n}}, \frac{3}{2^{n+1}}\right] \times \left(\frac{3}{2^{n+1}}, \frac{1}{2^{n-1}}\right] \\
\Omega_{n,6} = \left(\frac{1}{2^{n+1}}, \frac{1}{2^{n}}\right] \times \left(\frac{3}{2^{n+1}}, \frac{1}{2^{n-1}}\right] \\
\Omega_{n,7} = \left[0, \frac{1}{2^{n+1}}\right] \times \left(\frac{3}{2^{n+1}}, \frac{1}{2^{n-1}}\right] \\
\Omega_{n,8} = \left(\frac{1}{2^{n}}, \frac{3}{2^{n+1}}\right] \times \left[0, \frac{1}{2^{n+1}}\right] \\
\Omega_{n,9} = \left(\frac{1}{2^{n}}, \frac{3}{2^{n+1}}\right] \times \left(\frac{1}{2^{n+1}}, \frac{3}{2^{n+1}}\right] \\
\Omega_{n,10} = \left(\frac{1}{2^{n}}, \frac{3}{2^{n+1}}\right] \times \left(\frac{1}{2^{n}}, \frac{3}{2^{n+1}}\right] \\
\Omega_{n,11} = \left(\frac{1}{2^{n+1}}, \frac{1}{2^{n}}\right] \times \left(\frac{1}{2^{n}}, \frac{3}{2^{n+1}}\right] \\
\Omega_{n,12} = \left[0, \frac{1}{2^{n+1}}\right] \times \left(\frac{1}{2^{n}}, \frac{3}{2^{n+1}}\right] \qquad (3.30)$$

For any  $(u, v) \in [0, 1] \times [0, 1]$ ,  $(u, v) \neq (0, 0)$ , there is an  $\Omega_{n,b}$  containing (u, v). We can find the value of S(u, v) by mapping  $\Omega_{n,b}$  to the unit square  $[0, 1] \times [0, 1]$  and finding the corresponding  $(\overline{u}, \overline{v})$ . After the mapping, we compute  $S_{n,b}$  at  $(\overline{u}, \overline{v})$ . The value of S(0, 0) is the limit of extraordinary vertices.

In the above process, n and b can be computed by:

$$n(u, v) = \min\{\lceil \log_{\frac{1}{2}} u \rceil, \lceil \log_{\frac{1}{2}} v \rceil\}$$
  
$$b(u, v) = k, if (u, v) \in \Omega_{n,k}, k = 1, ..., 12$$

The mapping from  $\Omega_{n,b}$  to the unit square is defined by

$$(\overline{u},\overline{v}) = (\phi(u),\phi(v))$$
, with

$$\phi(t) = \begin{cases} 2^{n+1}t, & \text{if } 2^n t \le 0.5\\ 2^{n+1}t - 1, & \text{if } 1 \ge 2^n t > 0.5\\ 2^{n+1}t - 2, & \text{if } 1.5 \ge 2^n t > 1\\ 2^{n+1}t - 3, & \text{if } 2^n t > 1.5 \end{cases}$$

The GCCSS S(u, v) can be expressed as follows

$$S(u,v) = W^T(\overline{u},\overline{v})MG_n^b$$
(3.31)

where  $\boldsymbol{G}_{n}^{b}$  is the control point vector of  $S_{n,b}$ , W(u,v) is the 16-power-basis vector with  $W^{T}(u,v) = [1, u, v, u^{2}, uv, v^{2}, u^{3}, u^{2}v, uv^{2}, v^{3}, u^{3}v, u^{2}v^{2}, uv^{3}, u^{3}v^{2}, u^{2}v^{3}, u^{3}v^{3}]$ . M is the B-spline coefficient matrix. We can express  $W^{T}(\overline{u}, \overline{v})$  as follows

$$W^{T}(\overline{u},\overline{v}) = W^{T}(u,v)K^{n}D_{b}$$
(3.32)

where K is a diagonal matrix, with

$$K = Diag(1, 2, 2, 4, 4, 4, 8, 8, 8, 8, 8, 16, 16, 16, 32, 32, 64),$$

and  $D_b$ , an upper triangular matrix depending on b only, maps  $(\overline{u}, \overline{v})$  to (u, v). So we can rewrite the GCCSS as

$$S(u,v) = W^T(u,v)K^n D_b M \boldsymbol{G}_n^b$$
(3.33)

By defining  $G_n^b = \overline{P}_b G_n$ , where  $\overline{P}_b$  is a picking matrix choosing 16 control points from the 45-point control mesh  $G_n$ , we get

$$S(u,v) = W^{T}(u,v)K^{n}D_{b}M\overline{P}_{b}\boldsymbol{G}_{n}$$
(3.34)

The above equation provides a formal parametrization for an extraordinary face  $f_i$  of a GCCSS. Together with the definition of  $G_n$ , this parametrization provides a linear time computation of a GCCSS.

### **3.6** Conditions for Curvature Continuity

Through parameterization and eigen-decomposition of control point meshes of GCCSS, we obtain a unique eigenstructure for extraordinary faces with arbitrary valence N. Based on the above works, we can now analyze the conditions for curvature continuity at extraordinary points.

By equations (3.26) and (3.21), we have all the control points in GCCSs except those guiding control points in the first layer of an  $n^{th}$  GCCS. These guiding control points are mapped from *dominative* control meshes  $\hat{C}_k$ . These *dominative* control meshes are key to obtain curvature continuity at extraordinary points, and we will present the conditions for curvature continuity for these control meshes in this section.

Before we pick the control points for  $\hat{C}_k$ , we need to do a preprocessing on the extraordinary vertex V, i.e. we need to find its limit point and unit normal. We define  $d_V$  and  $n_V$  as the limit point and unit normal of V, obtained from the rules of CCS.

Each  $\hat{C}_k$  needs to satisfy the following conditions:

$$d_{\hat{V}_k} = d_V \quad and \quad n_{\hat{V}_k} = n_V, \ k = 1, ..., 2N$$
 (3.35)

where  $d_{\hat{V}_k}$  and  $n_{\hat{V}_k}$  are the limit point and unit normal of  $\hat{V}_k$  (after applying infinitely many subdivision steps on the dominative control mesh  $\hat{C}_k$ ).

The initial choice of  $\hat{C}_k$  is as follows. We regroup all the edge points and face points surrounding the extraordinary vertex V of the original mesh into  $(H_1, ..., H_{2N})$ , where  $H_{2j-1} = E_j, H_{2j} = F_j, j = 1, ..., N.$  We initialize  $\hat{C}_k$  with

$$\hat{E}_{k,1} = H_k, \qquad \hat{E}_{k,3} = H_{k+N}, 
\hat{F}_{k,1} = H_{k+1}, \qquad \hat{F}_{k,2} = H_{k-1+N}, 
\hat{F}_{k,3} = H_{k+1+N}, \qquad \hat{F}_{k,4} = H_{k-1}, 
\hat{V}_k = \hat{E}_{k,2} = \hat{E}_{k,4} = \frac{3}{2} (d_V - \frac{1}{9} (\hat{E}_{k,1} + \hat{E}_{k,3}) - \frac{1}{36} \sum_{i=1}^4 \hat{F}_{k,i})$$
(3.36)

This initialization fulfills the first condition of equation (27), i.e., the limit point  $d_{\hat{V}_k}$  of the dominative control mesh equals  $d_V$ .

We still need to include additional constraints to satisfy the 2-nd condition of (3.27), i.e., the dominative control meshes have the same unit normal  $n_V$  at the limit point  $d_{\hat{V}_k}$  ( $k \in \{1, ..., N\}$ ). We process the dominative control mesh  $\hat{C}_k$  as follows:

- get the first order derivatives D<sub>u</sub>, D<sub>v</sub> at d<sub>V<sub>k</sub></sub>. Since C<sub>k</sub> is a part of a regular patch,
   D<sub>u</sub> and D<sub>v</sub> can be easily calculated.
- (2) get  $t = D_u \cdot n_V$ , the projection of  $D_u$  on  $n_V$
- (3) let  $\hat{F}_{k,1} = 3t$ ,  $\hat{F}_{k,4} = 3t$ ,  $\hat{F}_{k,2} = 3t$ ,  $\hat{F}_{k,3} = 3t$ , which ensure  $D_u \cdot n_V = 0$
- (4) get  $t = D_v \cdot n_V$ , the projection of  $D_v$  on  $n_V$
- (5) let  $\hat{F}_{k,1} = 3t$ ,  $\hat{F}_{k,2} = 3t$ ,  $\hat{F}_{k,3} = 3t$ ,  $\hat{F}_{k,4} = 3t$ , which ensure  $D_v \cdot n_V = 0$

From the above procedure, we can derive that the control points of  $\hat{C}_k$  are the same as those of  $\hat{C}_{k+N}$ , with the property that

$$\hat{V}_{k} = \hat{V}_{k+N},$$

$$\hat{E}_{k,l} = \hat{E}_{k+N,(l+2)\%4},$$

$$\hat{F}_{k,l} = \hat{F}_{k+N,(l+2)\%4}, \quad l = 1, .., 4.$$
(3.37)

 $\hat{C}_k$  and  $\hat{C}_{k+N}$  share the same set of control points, only differ in the ordering. This is an important property which will be used to prove  $C^2$  continuity at an extraordinary point of even valence.

For an odd valence, we also include the following additional constraint:  $(\frac{K_2}{K'_2})^2 K'_3 = K_3$  (can be explicitly calculated from  $\hat{C}_k$  and  $\hat{C}_{k+N}$ ). Further processing will be needed, which we will show in the next section of *Smoothness Evaluation*.

### 3.7 Smoothness Evaluation of GCCSS's

In this section, we show the smoothness behavior of a GCCSS. Parametrization results will be used to evaluate the value of a patch at a given (u, v) near an extraordinary point.

Each GCCS will generate 12 sub-faces, the control points determining the outer boundary of these sub-faces are generated by regular midpoint knot insertion, which guarantees  $C^2$  continuity with neighboring patches from previous subdivision. So at any (u, v) (excluding (0, 0)), the GCCSS is  $C^2$  continuous.

To evaluate the limit surface of  $f_i$  near the extraordinary point, we need to first analyze  $G_n$ . For  $B_{m,t,n-4}$ , when  $n \to \infty$  in equation (3.20), we have

$$B_{m,t,n-4} = \begin{cases} \lambda_m^{n-4} \lambda_t^2 \frac{1}{1 - \frac{\lambda_t}{\lambda_m}}, & \text{if } m < t \\ (n-3)\lambda_m^{n-2}, & \text{if } m = t \\ \lambda_t^{n-2} \frac{1}{1 - \frac{\lambda_m}{\lambda_t}}, & \text{if } m > t \end{cases}$$
(3.38)

When  $n \to \infty$ , equation (3.23) can be rewritten as

$$\boldsymbol{G}_n = \sum_{m=1}^7 \Psi_{m,i} \lambda_m^n \tag{3.39}$$

where  $\Psi_{m,i}$  is the coefficient vector of size 45 on  $f_i$ . Since 2nd order continuity at the extraordinary vertex will only involve eigen values  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$ , we only need to analyze  $\Psi_{1,i}$ ,  $\Psi_{2,i}$ ,  $\Psi_{3,i}$ . From equation (3.21), we see that these three vectors are solely determined by dominative control meshes of GCCSS, so we define

$$\Psi_{j,i} = \Gamma_j [P_2 T_j \hat{D}_i]^T, \quad j = 1, 2, 3,$$
(3.40)

where  $\Gamma_j$  are matrices of dimension  $45 \times 5$ ,

$$\Gamma_{1} = S_{1} + S_{2}S_{1} + S_{2}^{2}S_{1} + \sum_{m=4}^{7} (B_{m,1,n-4}S_{3}T'_{m}S_{4}S_{1})$$

$$\Gamma_{2} = \frac{S_{1}}{2} + S_{2}S_{1} + 2S_{2}^{2}S_{1} + 2^{n}\sum_{m=4}^{7} (B_{m,2,n-4}S_{3}T'_{m}S_{4}S_{1})$$

$$\Gamma_{3} = \frac{S_{1}}{4} + S_{2}S_{1} + 4S_{2}^{2}S_{1} + 2^{2n}\sum_{m=4}^{7} (B_{m,3,n-4}S_{3}T'_{m}S_{4}S_{1})$$
(3.41)

After computation, we get

$$P_{2}T_{1} = \begin{bmatrix} \frac{4}{9} & \frac{1}{9} & \frac{1}{9} & \frac{1}{9} & \frac{1}{9} & \frac{1}{36} & \frac{1}{36} & \frac{1}{36} & \frac{1}{36} \end{bmatrix}$$

$$P_{2}T_{2} = \begin{bmatrix} 0 & \frac{1}{3} & 0 & -\frac{1}{3} & 0 & \frac{1}{12} & -\frac{1}{12} & -\frac{1}{12} & \frac{1}{12} \end{bmatrix}$$

$$P_{2}T_{3} = \begin{bmatrix} -\frac{2}{9} & \frac{5}{18} & -\frac{2}{9} & \frac{5}{18} & -\frac{2}{9} & \frac{1}{36} & \frac{1}{36} & \frac{1}{36} & \frac{1}{36} \end{bmatrix}$$

$$(3.42)$$

### $C^0$ and tangent plane continuity

We define  $S(u, v)|\Omega_{n,b}(0, 0), f_i$  as the limit point of a GCCSS at (0,0) of  $S_{n,b}$  on  $f_i$ , with  $(0,0) = (\phi(u), \phi(v))$  (mapping defined in section 5). By equation (3.28), the limit point on sub-patch  $S_{n,b}$  is an affine combination of 16 control points selected from 45 control points of  $\mathbf{G}_n$ .

By equation (3.7), all dominative control meshes will converge to the same data point  $d_V$  and share the same unit normal  $n_V$ , we get  $[P_2T_1\hat{D}_i] = [d_V \ d_V \ d_V \ d_V \ d_V]$ .  $\Gamma_1$  can be explicitly calculated and with the property that sum of each row equals one, so vector  $\Psi_{1,i} = [d_V \ \dots \ d_V]^T$ . With equation (3.28), when  $n \to \infty$ , we have

$$S(u,v)|\Omega_{n,b}(0,0), f_i = W^T(0,0)M\overline{P_b}\Psi_{1,i} = d_V$$
(3.43)



Figure 3.12: Valence 8. Left (top to bottom): control mesh, GCCSS limit surface, CCSS limit surface; right (top to bottom): two views of the GCCSS data mesh with Gaussian curvature, two views of the CCSS data mesh.

With the above computation, we can conclude that a GCCSS is  $C^0$  continuous at an extraordinary point.

We define  $D_u | \Omega_{n,b}(0,0), f_i$  and  $D_v | \Omega_{n,b}(0,0), f_i$  as the first order derivatives of the

GCCSS at (0,0) of  $S_{n,b}$  on  $f_i$ , with  $(0,0) = (\phi(u), \phi(v))$  on  $f_i$ . When  $n \to \infty$ , we get

$$D_{u}|\Omega_{n,b}(0,0), f_{i} = 2 \frac{\partial W^{T}(0,0)}{\partial \overline{u}} M \overline{P}_{b} \Psi_{2,i}$$
$$D_{v}|\Omega_{n,b}(0,0), f_{i} = 2 \frac{\partial W^{T}(0,0)}{\partial \overline{u}} M \overline{P}_{b} \Psi_{2,i}$$
(3.44)

If we define the first order derivatives at  $\hat{V}_k$  of  $\hat{C}_k$  as  $D_u | \hat{C}_k$  and  $D_v | \hat{C}_k$ , we observe that  $P_2 T_2 \hat{C}_k = D_u | \hat{C}_k$ , such that, when  $n \to \infty$ ,  $D_u | \Omega_{n,b}(0,0)$  and  $D_v | \Omega_{n,b}(0,0)$  take the values of the linear combination of  $D_u | \hat{C}_{2i-2}$ ,  $D_u | \hat{C}_{2i-1}$ ,  $D_u | \hat{C}_{2i}$ ,  $D_u | \hat{C}_{2i+1}$  and  $D_u | \hat{C}_{2i+2}$ . Also by prerequisite conditions of dominative control meshes, their first order derivatives  $D_u | \hat{C}_k$  and  $D_v | \hat{C}_k$  share the same tangent plane. Hence, we can conclude that the first order derivatives of a GCCSS at an extraordinary point share the same tangent plane, i.e.  $G^1$  continuous.

For curvature continuity at an extraordinary point, we consider the problem in two cases, even valence and odd valence. Fig. 3.13 and Fig. 3.14 show examples of valence 8 and 11 respectively. In both cases, the GCCSS shows smooth and non-zero curvature at the extraordinary point while the CCSS has sharp tip at the extraordinary point (curvature unbounded).

## $C^1$ and $C^2$ continuity of even valence

For an even valence, we can show that it is  $C^1$  and  $C^2$  continuous at the extraordinary point. To prove  $C^1$  continuity, we need to show that  $D_u|\Omega_{n,b}(0,0)$  and  $D_v|\Omega_{n,b}(0,0)$ for the i - th and  $(i + \frac{N}{2}) - th$  patches have the same value but different signs.

For  $f_{i+\frac{N}{2}}$ , when  $n \to \infty$ ,

$$D_{u}|\Omega_{n,b}(0,0), f_{i+\frac{N}{2}} = 2\frac{\partial W^{T}(0,0)}{\partial \overline{u}}M\overline{P}_{b}\Psi_{2,i+\frac{N}{2}}$$
$$D_{v}|\Omega_{n,b}(0,0), f_{i+\frac{N}{2}} = 2\frac{\partial W^{T}(0,0)}{\partial \overline{u}}M\overline{P}_{b}\Psi_{2,i+\frac{N}{2}}$$
(3.45)

By equation (3.9), the prerequisite conditions for dominative control meshes for even valence, we can get

$$\Psi_{2,i+\frac{N}{2}} = -\Psi_{2,i} \tag{3.46}$$

From equation (3.37) and comparison between equations (3.35) and (3.36), we can conclude that a GCCSS is  $C^1$  continuous at an extraordinary vertex of even valence.

We define  $D_{uu}|\Omega_{n,b}(0,0), f_i, D_{uv}|\Omega_{n,b}(0,0), f_i$  and  $D_{vv}|\Omega_{n,b}(0,0), f_i$  as the 2nd order derivatives in uu, uv and vv directions at (0,0) of  $S_{n,b}$  on  $f_i$ , respectively. To prove  $C^2$  continuity, in addition to the  $C^0$  and  $C^1$  continuities shown above, we need to show that  $D_{uu}|\Omega_{n,b}(0,0), D_{uv}|\Omega_{n,b}(0,0)$  and  $D_{vv}|\Omega_{n,b}(0,0)$  for  $f_i$  and  $f_{i+\frac{N}{2}}$  have the same value.

For  $f_k$ , when  $n \to \infty$ ,

$$D_{uu}|\Omega_{n,b}(0,0), f_{k} = 4 \frac{\partial W^{T}(0,0)}{\partial \overline{uu}} M \overline{P}_{b} \Psi_{3,k}$$
  

$$D_{uv}|\Omega_{n,b}(0,0), f_{k} = 4 \frac{\partial W^{T}(0,0)}{\partial \overline{uv}} M \overline{P}_{b} \Psi_{3,k}$$
  

$$D_{vv}|\Omega_{n,b}(0,0), f_{k} = 4 \frac{\partial W^{T}(0,0)}{\partial \overline{vv}} M \overline{P}_{b} \Psi_{3,k}$$
(3.47)

By equation (3.9), the prerequisite conditions for dominative control meshes for even valence, we get

$$\Psi_{3,i+\frac{N}{2}} = \Psi_{3,i} \tag{3.48}$$

From equations (3.38) and (3.39), the second order derivatives of  $f_i$  and  $f_{i+\frac{N}{2}}$  take the same value. Hence, a GCCSS is  $C^2$  continuous at an extraordinary vertex of even valence.

#### Curvature continuity of odd valence

For curvature continuity at an extraordinary point with odd valence, we will not use the same technique used for even valence. Given an extraordinary vertex V of odd valence N, we can not establish direct correlation between u, v directions of  $f_i$  and those of the patch on the opposite site like in the even case. However, alternatively, if we can find a Taylor series such that data points generated by the GCCS on opposite sides satisfy the Taylor series expansion up to the 2nd order, then we prove that it is curvature continuous at the extraordinary vertex.



Figure 3.13: Valence 11. Left (top to bottom): control mesh, GCCSS limit surface, CCSS limit surface with data mesh showing Gaussian curvature; right (top to bottom): upper left detail of GCCSS data mesh, upper right detail of CCSS data mesh, middle and bottom CCSS limit surface and limit surface with data mesh showing gaussian curvature.

According to Taylor Theorem, if  $f(t) = f(0) + \frac{f'(0)}{1!}t + \frac{f''(0)}{2!}t^2 + \delta$  then f(t) is continuous up to the second order, where  $\delta$  is the remainder of the Taylor expansion if there exist higher derivatives. Here we select  $S(0,0)|\Omega_{n,8}, f_i$  and  $S(0,0)|\Omega_{n,10}, f_{i+\frac{N-1}{2}}$ to compare their Taylor expansions (these two points are on opposite sides of the extraordinary vertex V). Note that, for  $f_i$  and  $f_{i+\frac{N-1}{2}}$ , when  $n \to \infty$ , as far as the curvature is concerned, for S(u, v) near V, we only need to consider coefficient vectors for  $\lambda_1^n$ ,  $\lambda_2^n$  and  $\lambda_3^n$ .

$$S(u,v)|\Omega_{n,8}(0,0), f_i \approx W^T(0,0)M\overline{P}_8 \sum_{m=1}^3 \Psi_{m,i}\lambda_m^n$$
  
$$S(u,v)|\Omega_{n,10}(0,0), f_{i+\frac{N-1}{2}} \approx W^T(0,0)M\overline{P}_{10} \sum_{m=1}^3 \Psi_{m,i+\frac{N-1}{2}}\lambda_m^n$$
(3.49)

Equation (3.41) can be redefined as

$$f(t) = \kappa_1 + \kappa_2 t + \kappa_3 t^2$$
  

$$g(t) = \kappa'_1 + \kappa'_2 t + \kappa'_3 t^2$$
(3.50)

with

$$f(t) = S(u, v) |\Omega_{n,8}(0, 0), f_i,$$
  

$$g(t) = S(u, v) |\Omega_{n,10}(0, 0), f_{i+\frac{N-1}{2}},$$
  

$$t = (\frac{1}{2})^n,$$
  

$$\kappa_m = W^T(0, 0) M \overline{P}_8 \Psi_{m,i}, and$$
  

$$\kappa'_m = W^T(0, 0) M \overline{P}_{10} \Psi_{m,i+\frac{N-1}{2}}, for \ m = 1, 2, 3.$$
  
(3.51)

Since  $\kappa_1 = \kappa'_1 = d_V$ , to show that f(t) and g(t) represent the same curve after reparametrization, we only need to show that  $(\frac{\kappa_2}{\kappa'_2})^2 = \frac{\kappa_3}{\kappa'_3}$ .  $\kappa_2, \kappa_3, \kappa'_2, \kappa'_3$  depend on the picking of the dominative control point meshes  $\hat{C}_k$ , which can be computed explicitly. So we can put N additional constraints into the dominative control meshes of 2Nregular bi-cubic B-spline patches. Also note that in an odd case, each dominative control mesh will only determine one control point of each GCCS instead of two for an even case. By putting these N constraints into a linear system, we can guarantee that the the GCCSS is  $G^2$  continuous at the extraordinary vertex. In the above we have proved  $C^2$  continuity of a GCCSS at an extraordinary point of even valence, we have also proved  $G^2$  continuity at an extraordinary point of odd valence. Hence, we can conclude that a GCCSS is curvature continuous at an extraordinary point.

### 3.8 Summary

In this work we have shown that our new GCCS scheme guarantees curvature continuity at CCS extraordinary points. In contrast to previous works on extraordinary points, the new scheme is purely subdivision based and uses only regular bi-cubic subdivision with mesh blending technique. This avoids the hassle to recompute eigenvalues and eigenbases for every valence in the original CCSS, instead the eigenstructures of the new scheme have different eigenvalues of  $1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}$  (the eigenvalues for regular bi-cubic subdivision), so the scheme has a unique eigenbase for any valence.

Furthermore, a GCCSS is flexible, we can adjust the shape of the subdivision surface by fine-tuning the dominative control meshes as far as the choice of control points fulfill the requirement set forth in this work. The linear system for choosing the control points of 2N dominative control meshes is underdetermined, so this leaves room for changing the shape of the subdivision surface without sacrificing the curvature continuity.

Our next step is to evaluate the behavior of a GCCSS near extraordinary vertices by putting additional constraints on the dominative control point meshes.

## Chapter 4

# Polar embedded Catmull-Clark Subdivision Surface

Subdivision surfaces have been widely used in CAD, gaming and computer graphics. Catmull-Clark subdivision (CCS) [6], based on tensor product bi-cubic B-Splines, is one of the most important subdivision schemes. The surfaces generated by the scheme are  $C^2$  continuous everywhere except at extraordinary points, where they are  $C^1$  continuous.

While CCS is popular in applications of computer graphics and animations, a shortcoming inherent in CCS surfaces is the ripple problem, that is, ripples tend to appear around extraordinary points with high valence. In the past, research focused on improving the curvature at extraordinary points. However, with quad mesh structure of CCS surfaces, the ripples could not be avoided in high valence cases. The technique of fairing [29] is used to address the smoothness issue on the limit surface, but the computation is quite expensive and it changed the limit surface to the extent that it does not generate the desired shape.

To handle this artifact, Polar configuration has been studied by a number of researchers. Polar configuration has a quad/triangular mixed mesh structure. It has the following properties: faces adjacent to the extraordinary points are triangular, all other faces are regular. One can find the works of guided and Polar surfacing in [53]. These include a guided subdivision scheme that uses a Bezier surface as a guide for each subdivision step, and a  $C^2$  accelerated bi-cubic guided subdivision that uses  $2^m$  subfaces in the  $m^{th}$  level for surface patches surrounding extraordinary points. In the second case, they show that although this scheme is not practical for CCS surfaces, it can be applied in a Polar configuration. A bi-cubic Polar subdivision scheme is presented in [33] that sets up the control mesh refinement rules for Polar configuration so that the limit surface is  $C^1$  continuous and curvature bounded. As a further step, Myles and Peters [50] presented a bi-cubic  $C^2$  Polar subdivision scheme that gets a  $C^2$  Polar surface by modifying the weights of Polar subdivision scheme for different valences.

Although a Polar surface handles high valence cases well, there are issues preventing its application in subdivision surfaces. Mismatch of subdivision masks between Polar and CCS makes it difficult to connect Polar to CCS meshs. Although in [49], the effort is made to connect Polar to CCS meshes, it suffers the problem of inconsistent limit surfaces with refined control mesh at different subdivision levels.

A free-form quad/triangular scheme was presented in [55], [63] and [60]. However, the scheme was not designed to handle high-valence ripples as Polar surface.

In this work, we redefined a quad/tri mesh structure, named the *Polar Catmull-Clark mesh* (PCC mesh), which embeds Polar configuration into the Catmull-Clark mesh structure to solve the high valence issue. Based on PCC mesh, we show a new subdivision scheme. In contrast to the work in [49], our new scheme has the equivalent subdivision masks on both Polar and CCS parts, such that there are no mismatches of subdivision rules on the boundaries between Polar and CCS parts and avoid the artifact of inconsistent limit surface at different subdivision levels. We also show that the generated limit surface on triangular control meshes is  $G^2$  at extraordinary points

and the artifact of high valence ripples is resolved effectively.



Figure 4.1: The bottom shows two CCS mesh designs for the head of an air plane, right mesh improves ripples by having zero curvature on the tip; the top right shows the control mesh and limit surface of our new scheme with a Polar extraordinary point on the tip of the plane head.

Since our new subdivision does not change the rules on quadrilateral faces, one can apply earlier  $G^2$  solutions on CCS and obtain a  $G^2$  everywhere subdivision surface.

## 4.1 Polar Catmull-Clark Mesh

Before we introduce Polar Catmull-Clark Mesh, we first review the control meshes of CCS and Polar.

CCS works on arbitrary topology. The subdivision requires all quad faces with no extraordinary points neighboring to each other, which is obtained by twice subdivision on original mesh [6]. Polar surfaces have the following properties on mesh structure: faces adjacent to the extraordinary points are triangular, all other faces
are regular [53] [33] [48]. Fig. 4.2 left and middle show typical meshes of Polar and Catmull-Clark respectively.

Efforts are made to combine Polar with Catmull-Clark mesh [49]. However, for the joining of Polar part and CCS part, it has 4 steps. 1) separate subdivision into two parts, 2) performing k times subdivision radially and then k times circularly, 3) performing k times subdivision on remaining CCS mesh, 4) merge boundaries set by 2) and 3). This algorithm suffers the problem that the limit surface of the merged control mesh will be different with different subdivision levels. By analyzing its algorithm, one can find this artifact is caused by mismatch between subdivision masks for Polar parts and CCS parts. This artifact needs to be resolved, since in CAGD and other high precision graphics applications, limit surface is generally required to be unchanged with refined control meshes.

In this work we present a new subdivision scheme on Polar Catmull-Clark mesh (PCC mesh). The *redefined PCC mesh* has a quad/triangular mixed mesh structure, its quad faces have the same structure as a CCS mesh, triangular faces are arranged in Polar configuration and are embedded in N-sided holes of the quad mesh.



Figure 4.2: From left to right, Polar mesh, CCS mesh, and PCC mesh.

The right side of Fig. 4.2 shows a typical PCC mesh structure. The reason we include a ring of quad faces near the triangular faces (shown inside the bold boundary lines) is to ensure there is no edge between two extraordinary points.

Fig. 4.1 shows that a CCS control mesh of an airplane is modified to embed a

Polar configuration at plane head, and our new  $G^2$  scheme on Polar part eliminates the ripples and generates non-zero curvature on the tip of the plane head.

A PCC mesh is flexible to design. Given an arbitrary topology, one just needs do subdivision twice to generate a CCS mesh, and then analyze the mesh and find out where one wants to put Polar meshes, typically for high valence extraordinary faces (which will have ripples by performing only CCS), taking out these extraordinary faces and replacing them with triangular/quad meshes as shown inside the bold ring on the right of Fig. 4.2.



Figure 4.3: Control meshes of Catmull- Clark subdivision. Left side: a regular face; right side: an extraordinary face

# 4.2 Subdivision Rules on Quad Faces

In our new subdivision scheme, for PCC quadrilateral faces not adjacent to triangular faces, the subdivision process will be exactly the same as CCS. Recall that the CCS scheme divides the control vertices into three categories: *vertex points, edge points,* and *face points.* A popular way to index the control vertices is shown in Fig. 4.3, where V is a vertex point,  $E_i$ 's are edge points,  $F_i$ 's are face points and  $I_{i,j}$ 's are

inner ring control vertices. New vertices within each subdivision step are generated as follows:

$$V' = \alpha_N V + \beta_N \sum_{i=1}^N E_i / N + \gamma_N \sum_{i=1}^N F_i / N$$
  

$$E'_i = \frac{3}{8} (V + E_i) + \frac{1}{16} (E_{i+1} + E_{i-1} + F_i + F_{i-1})$$
  

$$F'_i = \frac{1}{4} (V + E_i + E_{i+1} + F_i)$$
(4.1)

where N is the valence of vertex V, with  $\alpha_N = 1 - \frac{7}{4N}$ ,  $\beta_N = \frac{3}{2N}$ , and  $\gamma_N = \frac{1}{4N}$ .

For each extraordinary point V inside a Polar structure, such as the one shown on the left of Fig. 4.4, we perform a simple vertex splitting on V as shown on the right of Fig. 4.4, such that its adjacent triangular faces are converted into quad ones. Then we can apply CCS on the PCC quad faces adjacent to the triangular faces with equation (4.1). This solves the artifact of inconsistent limit surfaces caused by merging control points at different subdivision level in [49].



Figure 4.4: Control mesh conversion for quad faces adjacent to triangular faces

## 4.3 Guided U-Subdivision

In this section, we present our subdivision rules on PCC triangular faces. We first introduce a CCS equivalent subdivision scheme, the U-Subdivision. Then we present a Guided U-Subdivision. This Guided U-Subdivision will recursively generate a deformed limit surface that is  $C^2$  continuous both inside a quad face and on its boundaries with adjacent quad faces. By applying the Guided U-Subdivision, we will be able to generate a  $G^2$  limit surface on Polar part of a PCC mesh.

#### **U-Subdivision**

In CCS, a regular bi-cubic B-spline patch with parameters u and v can be expressed as

$$S(u,v) = \begin{bmatrix} 1 \ u \ u^2 \ u^3 \end{bmatrix} M \mathbf{P} M^T \begin{bmatrix} 1 \ v \ v^2 \ v^3 \end{bmatrix}^T$$
(4.2)

where **P** is a  $4 \times 4$  matrix of control points  $P_{ij}$ ,  $1 \le i, j \le 4$ , M is the coefficient matrix and  $M^T$  is its transpose. The subdivision process of control points are obtained by subdivision rules shown in equation (4.1).

In this work, we use a variant of the regular bi-cubic B-spline subdivision scheme. The first step of the new scheme, called *unilateral subdivision* (U-Subdivision), involves one parameter only. A U-Subdivision involving the parameter u only is defined as follows:

$$V' = \frac{3}{4}V + \frac{1}{8}E_1 + \frac{1}{8}E_3$$
  

$$E'_i = \frac{1}{2}V + \frac{1}{2}E_i$$
(4.3)

A U-Subdivision splits a regular CCS patch into two regular CCS sub-patches.

**PROPERTY** 4.1 : The limit surfaces of the two CCS sub-patches generated by a U-Subdivision are the same as the limit surface of that regular patch.

Proof: The two sub-patches generated by a U-Subdivision can be expressed as follows:

$$S_b(\bar{u}, \bar{v}) = \begin{bmatrix} 1 \ \bar{u} \ \bar{u}^2 \ \bar{u}^3 \end{bmatrix} M A_b \mathbf{P} M^T \begin{bmatrix} 1 \ \bar{v} \ \bar{v}^2 \ \bar{v}^3 \end{bmatrix}^T$$
(4.4)

where b = 1, 2,  $(\bar{u}, \bar{v})$  takes value from  $[0, 1] \times [0, 1]$ ,  $A_1$  and  $A_2$  are U-Subdivision matrices for the 1st and the 2nd sub-patches, respectively. For the 1st sub-patch, because

$$[1 \ \bar{u} \ \bar{u}^2 \ \bar{u}^3] \ MA_1 = [1 \ \frac{1}{2}\bar{u} \ \frac{1}{4}\bar{u}^2 \ \frac{1}{8}\bar{u}^3] \ M$$

we can express the sub-patch as

$$S_1(\bar{u}, \bar{v}) = \begin{bmatrix} 1 & \frac{1}{2}\bar{u} & (\frac{1}{2}\bar{u})^2 & (\frac{1}{2}\bar{u})^3 \end{bmatrix} M \mathbf{P} M^T \begin{bmatrix} 1 & \bar{v} & \bar{v}^2 & \bar{v}^3 \end{bmatrix}^T$$

which is exactly the first half of the original (u, v) regular patch. Similarly, we can see that the 2nd sub-patch represents the 2nd half of the original patch. QED

Consequently, we can prove that after n times U-Subdivision, the limit surfaces of  $2^n$  U-subdivided sub-patches are the same as the original CCS limit surface.

#### Guided U-Subdivision

In this section, we show how to perform a guided U-Subdivision (GUS) and how to obtain a GUS surface.

For a regular patch, if we do a U-Subdivision, we get 2 sub-patches with 20 control points. These points are distributed in 5 layers, with four points each. We denote them  $L_1$ ,  $L_2$ ,  $L_3$ ,  $L_4$  and  $L_5$ , respectively (as shown in Fig. 4.5).

**PROPERTY 4.2**: Only  $L_3$ ,  $L_4$ , and  $L_5$  obtained after a U-Subdivision on a regular patch are needed to ensure  $C^2$  continuity of the limit surface on the common boundary with an adjacent patch underneath it.

Proof: This property is trivial in CCS and can be derived from analysis of equation (4.2). QED

This gives us an opportunity to set up a recursive subdivision scheme that takes  $L_3$ ,  $L_4$ , and  $L_5$  from a U-Subdivision on previous control mesh, but leaves  $L_1$  and



Figure 4.5: Left side shows 5 layers in a U-Subdivision, right shows  $L_1$  and  $L_2$  will not change boundary (red) continuity.

 $L_2$  at the user's choice, so that the shape of the limit surface can be guided by the selected  $L_1$  and  $L_2$ .



Figure 4.6:  $\Omega$ -Partitions, left for Catmull-Clark, right for GUS

Given an arbitrary regular patch with a  $4 \times 4$  control point mesh **P**, we define the limit surface S(u, v) of a GUS surface as the union of recursively generated U-Subdivision surfaces  $S_{n,b}(\bar{u}, \bar{v})$  (limit surface of  $n^{th}$  GUS and  $b^{th}$  sub-patch), with an  $\Omega$ -partition (see Fig. 4.6) defined as follows:

$$\Omega_{n,1} = \left[\frac{1}{2^n}, \frac{3}{2^{n+1}}\right] \times [0,1], \qquad \qquad \Omega_{n,2} = \left[\frac{3}{2^{n+1}}, \frac{1}{2^{n-1}}\right] \times [0,1]$$

Hence, each GUS will generate 2 regular sub-patches which require 5 layers of 20 control points. The GUS process is shown below.

For this given regular patch, we need to define a  $5 \times 4$  basis control mesh  $\mathbf{P}^0$  for the GUS first. The first three layers of  $\mathbf{P}^0$  are obtained by performing a U-Subdivision on the last three layers of  $\mathbf{P}$  and the last two layers of  $\mathbf{P}^0$  are zero, i.e.,

$$\mathbf{P}^{0} = \begin{bmatrix} A_{3}P_{3,4}^{\prime}\mathbf{P} \\ 0 \end{bmatrix}, \qquad \text{with } A_{3} = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{1}{8} & \frac{3}{4} & \frac{1}{8} \\ 0 & \frac{1}{4} & \frac{1}{4} \end{bmatrix} \qquad (4.5)$$

and  $P'_{3,4}$  is a 3 × 4 picking matrix with  $I_3$  (identity matrix of size 3) on the right side of the matrix.

For each  $n \geq 1$ , let  $\mathbf{P}^n$  be the 5 × 4 control point matrix of the  $n^{th}$  GUS with layers  $L_i^n$ ,  $1 \leq i \leq 5$ . The last three layers  $L_3^n$ ,  $L_4^n$  and  $L_5^n$  of  $\mathbf{P}^n$  are obtained by performing a *U*-subdivision on the first three layers  $L_1^{n-1}$ ,  $L_2^{n-1}$  and  $L_3^{n-1}$  of  $\mathbf{P}^{n-1}$ , i.e.,

$$P_{3.5}'\mathbf{P}^n = A_3 P_{3,5} \mathbf{P}^{n-1}, \quad n \ge 1$$
(4.6)

where  $P_{3,5}$  and  $P'_{3,5}$  are  $3 \times 5$  picking matrices with  $I_3$  on the left and right side of the matrix, respectively. The first two layers  $L_1^n$  and  $L_2^n$  of  $\mathbf{P}^n$  are at the choice of the user (the selection criteria of these two layers will be discussed in Section 4 for a Polar configuration). Once these two layers have been selected, the control point computation process for the  $n^{th}$  GUS is complete.

**Theorem 4.1.** Control points in  $L_1^n$  and  $L_2^n$  of the control point matrix  $\mathbf{P}^n$  of an  $n^{th}$ GUS surface can be changed without affecting  $C^2$  continuity of the limit surface inside the parameter space and on the boundary (u = 1) with its adjacent regular patch. *Proof.* For  $\mathbf{P}^n$  of an  $n^{th}$  GUS surface, its  $L_3^n$ ,  $L_4^n$  and  $L_5^n$  are obtained by doing one U-Subdivision on the  $1^{st}$  three layers of  $\mathbf{P}^{n-1}$ , by Property 2, it is  $C^2$  continuous at the boundary with previous GUS patch. Within an  $n^{th}$  GUS surface,  $C^2$  continuity is trivial.

With all control points in  $\mathbf{P}^n$  defined, we can now define the GUS surface. For any  $(u, v) \in [0, 1] \times [0, 1]$ , where  $(u, v) \neq (0, v)$ , there is an  $\Omega_{n,b}$  containing (u, v). We can find the value of S(u, v) by mapping  $\Omega_{n,b}$  to the unit square  $[0, 1] \times [0, 1]$  and finding the corresponding point of (u, v) in the unit square:  $(\overline{u}, \overline{v})$ , then compute  $S_{n,b}$ (the limit surface of  $n^{th}$  GUS and  $b^{th}$  sub-patch) at  $(\overline{u}, \overline{v})$ . The value of S(0, v) is the limit of the GUS.

In the above process, n and b can be computed by:

$$n(u,v) = \left\{ \begin{array}{ll} 1, & \text{ if } 2^n u \leq 1.5 \\ 2, & \text{ else} \end{array} \right.$$

else

The mapping from  $\Omega_{n,b}$  to the unit square is defined as  $(\overline{u},\overline{v}) = (\phi(u),v)$ , with

$$\phi(u) = \begin{cases} 2^{n+1}u - 2, & \text{if } 1.5 \ge 2^n u > 1\\ 2^{n+1}u - 3, & \text{if } 2^n u > 1.5 \end{cases}$$

The limit surface S(u, v) can be defined as follows:

$$S(u,v) = W^T(\bar{u})M\mathbf{P}^{n,b}M^TW(\bar{v})$$
(4.7)

u

where  $\mathbf{P}^{n,b}$ , a 4 × 4 matrix, contains the 16 control points of  $S_{n,b}$ , with  $\mathbf{P}^{n,1} = S_1 \mathbf{P}^n$ and  $\mathbf{P}^{n,2} = S_2 \mathbf{P}^n$ ,  $S_1$  and  $S_2$  are picking matrices of size  $4 \times 5$  with  $I_4$  (identity matrix of size 4) on the left and right side of the matrix respectively. W(x) is the 4-component power basis vector with  $W^T(x) = [1, x, x^2, x^3]$ , M is the B-spline curve

coefficient matrix. We can express  $W^T(\overline{u})$  and  $W^T(\overline{v})$  as follows

$$W^T(\overline{u}) = W^T(u)K^{n+1}D_b, \qquad W^T(\overline{v}) = W^T(v)$$

where K is a diagonal matrix, with K = Diag(1, 2, 4, 8).  $D_b$  is an upper triangular matrix depending on b only, it maps  $(\overline{u}, \overline{v})$  to (u, v). So we can rewrite the subdivision surface as

$$S(u,v) = W^{T}(u)K^{n+1}D_{b}MS_{b}\mathbf{P}^{n}M^{T}W(v)$$

$$(4.8)$$

Thus we can decompose the limit surface into a sequence of recursively generated U-Subdivision surfaces,

$$S(u,v) = S_{1,2} \cup S_{1,1} \cup S_{2,2} \cup S_{2,1} \cup S_{3,2} \cup \dots$$

In the above, we have shown the construction of a GUS surface and proven its  $C^2$  continuity both inside the limit surface and on the boundary of u = 1. In the following section, we show how this subdivision scheme can be applied to the Polar configuration.

# 4.4 Applying GUS to Polar Parts

In this section, we focus on applying GUS on triangular faces. Fig. 4.7 shows typical Polar extraordinary points of even and odd valences. As shown in previous section, GUS starts with the last three control point layers of a regular patch. In order to apply GUS to a triangular face, first we need to identify its control point matrix of **P**.

Since for odd valence, the curvature continuity is more difficult to achieve than even cases, before we apply GUS, we need to convert odd valence to even. Performing one CCS so that the new extraordinary point will have an even valence (as shown on right side of Fig. 4.7). In this subdivision, each triangular face will be treated as a quad face by vertex splitting of Polar extraordinary point V (see Fig. 4.4). The new edge and face points of triangular faces are defined by equation (4.1), but for a new vertex point, we use the original CCS vertex point rule on arbitrary topology [6] by  $V' = \frac{N-2}{N}V + \frac{1}{N^2}\sum_{i=1}^{N}E_i + \frac{1}{N^2}\sum_{i=1}^{N}F'_i.$ 



Figure 4.7: Obtaining P for GUS: left for even valence, right for odd valence

After the above step, for a Polar triangular face (Fig. 4.7), we have

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ V & V & V & V \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P_{41} & P_{42} & P_{43} & P_{44} \end{bmatrix}$$

With equation (4.5), we can derive the  $5 \times 4$  GUS basis control mesh  $\mathbf{P}^0$  from  $\mathbf{P}$ .

For each  $n \ge 1$ , like the situation discussed in the previous section, 2 regular sub-patches defined by a 5 × 4 control point matrix  $\mathbf{P}^n$  will be generated by the GUS process. The last three layers  $L_3^n$ ,  $L_4^n$  and  $L_5^n$  of  $\mathbf{P}^n$  are obtained by performing a U-Subdivision on the first three layers of  $\mathbf{P}^{n-1}$  (see Fig. 4.8). Hence, equation (4.6) works here as well or, equivalently,

$$\begin{bmatrix} L_3^n \\ L_4^n \\ L_5^n \end{bmatrix} = A_3 \begin{bmatrix} L_1^{n-1} \\ L_2^{n-1} \\ L_3^{n-1} \end{bmatrix}$$
(4.9)

where  $A_3$  is defined in equation (4.5).

The computation of  $L_2^n$  involves  $L_1^n$ . We assume  $L_1^n$  is already available to us (this is the case in the real algorithm, i.e.,  $L_1^n$  will be computed before the computation of  $L_2^n$ ).  $L_2^n$  is computed as follows:

$$[L_2^n] = A' \begin{bmatrix} L_1^n \\ L_1^{n-1} \\ L_2^{n-1} \\ L_3^{n-1} \end{bmatrix}$$
(4.10)

where  $A' = \begin{bmatrix} \frac{1}{4} & \frac{5}{8} & \frac{1}{8} & 0 \end{bmatrix}$ . Equation (4.10) is the result of a so-called *virtual U-Subdivision*. Note that, from equation (4.3), if we define a virtual layer of control points  $L_0^{n-1}$  as follows:

$$L_0^{n-1} = 2L_1^n - L_1^{n-1}$$

and use  $L_0^{n-1}$ ,  $L_1^{n-1}$ ,  $L_2^{n-1}$  and  $L_3^{n-1}$  to form a 4 × 4 control mesh of a regular patch, then by performing a U-Subdivision on this 4 × 4 control mesh, we get a 5 × 4 control mesh whose first, third, fourth and fifth layers are exactly  $L_1^n$ ,  $L_3^n$ ,  $L_4^n$  and  $L_5^n$  (see Fig. 4.9). We call such a reverse U-Subdivision a *virtual U-Subdivision* and use the second layer of such a subdivision as the second layer of  $\mathbf{P}^n$ . Since  $L_2^n$  corresponds to a vertex layer, we have (from equation (4.3))

$$L_2^n = \frac{1}{8}L_0^{n-1} + \frac{3}{4}L_1^{n-1} + \frac{1}{8}L_2^{n-1}$$
$$= \frac{1}{4}L_1^n + \frac{5}{8}L_1^{n-1} + \frac{1}{8}L_2^{n-1}$$

which is exactly equation (4.10).

**Theorem 4.2.** By applying virtual U-Subdivision, limit surfaces of the two subpatches obtained in each GUS are the same and can be considered as the limit surface of a regular patch.



Figure 4.8: 5 control point layers generated after  $n^{th}$  GUS

*Proof.* The virtual control point layer  $L_0^{n-1}$  is obtained by reversing a U-Subdivision process for edge point (equation (4.3)), such that this can be derived from PROP-ERTY 4.1.



Figure 4.9: Virtual U-subdivision: grey circles are virtual control points obtained from control points in  $L_1^n$  and  $L_1^{n-1}$ .

We have shown the construction of control point layers  $L_2^n$ ,  $L_3^n$ ,  $L_4^n$  and  $L_5^n$  for  $\mathbf{P}^n$ . We now discuss the choice of control point layer  $L_1^n$ .

Due to properties of GUS, the unknown control points after  $n^{th}$  GUS are those in  $L_1^1, L_1^2, ..., \text{ and } L_1^n$ . These control points determine the shape of the limit surface.

Let the valence of the Polar extraordinary point V that we are considering be N (N is even). We define the data point of V as  $d_V$ , unit normal of limit surface at  $d_V$  as  $n_V$ , patch k surrounding V as  $f_k$ , GUS surface of  $f_k$  as S(k, u, v). The following conditions are what we expect the GUS surface to meet:

**Expectation 1** : all S(k, u, v) shall converge to a fixed data point  $d_V$  of V.

**Expectation 2** : at  $d_V$ , all N GUS patches shall share the same tangent plane, i.e. having the same unit normal  $n_V$ .

**Expectation 3** : At  $d_V$ , the opposite GUS patches  $(S(k, u, v) \text{ and } S(k + \frac{N}{2}, u, v))$  shall be curvature continuous.

From the above expectations, before picking the unknown values  $L_1^1, L_1^2, ..., L_1^n$  of the GUS's, we have to first determine the values of  $d_V$  and  $n_V$ . If we reorganize the control points surrounding V as  $\{V, E_1, E_2, ..., E_N\}$ , where  $E_1, ..., E_N$  are edge points connected to the extraordinary point V in a counterclockwise order, and define the triangular face  $f_k$  by  $\{V, E_k, E_{k\% N+1}\}, k \in [1, N]$ , we can pick the values of these terms as follows:

$$d_{V} = \frac{2}{3}V + \frac{1}{3N}\sum_{k=1}^{N} E_{k}$$
$$n_{V} = Norm(\sum_{k=1}^{N} n_{f_{k}})$$
(4.11)

where Norm(x) is a function which returns unit normal of a normal x.  $n_{f_k}$  is the face normal of  $f_k$ , can be obtained from  $n_{f_k} = (E_k - V) \times (E_{k\%N+1} - V)$ .

From Expectations 1 & 2, we come up with the concept of *dominative control* meshes. A dominative control mesh  $C_m$  of size 9 is defined as

$$C_m = [V_m, E_{m,1}, \dots, E_{m,4}, F_{m,1}, \dots, F_{m,4}]^T,$$

which is exactly the control point mesh of a regular bi-cubic patch without  $[I_1, I_2, I_3, I_4, I_5, I_6, I_7]^T$ .

By applying midpoint knot insertion to  $C_m$ , we get

$$C_m^{(n)} = A_9 C_m^{(n-1)} = \dots = (A_9)^n C_m, n \ge 1$$
(4.12)

where  $A_9$  is the midpoint insertion coefficient matrix, its values can be derived from equation (4.1).  $C_m^{(n)}$  is the control point mesh after  $n^{th}$  midpoint knot insertion on  $C_m$ , and can be expressed as

$$C_m^{(n)} = [V_m^{(n)}, E_{m,1}^{(n)}, \ \dots \ , E_{m,4}^{(n)}, F_{m,1}^{(n)}, \ \dots \ , F_{m,4}^{(n)}]^T$$

The reason  $I_i$  (i = 1, ..., 7) are ignored is: as shown in equation (4.1), the new vertex point, edge points and face points obtained from the midpoint knot insertion are independent of these inner ring control vertices. Since we plan to map recursively generated edge points of dominative control meshes into unknown values of  $L_1^n$  in GUS's, it will not be necessary to include these vertices into the control mesh.

There are totally N faces surrounding V, so we need N dominative control meshes to map these values, see Fig. 4.10 for the mapping from the dominative control meshes to the control points of the  $n^{th}$  GUS on face  $f_k$ . The mapping is defined as follows:

$$L_1^n[1] = E_{k-1,1}^{(n+1)}; \qquad L_1^n[2] = E_{k,1}^{(n+1)};$$
  

$$L_1^n[3] = E_{k+1,1}^{(n+1)}; \qquad L_1^n[4] = E_{k+2,1}^{(n+1)}$$
(4.13)

Due to the ring structure of control points in GUS, for the  $n^{th}$  GUS, the last three points in  $L_1^n$  of  $f_{k-1}$  are exactly the first three points in  $L_1^n$  of  $f_k$ . Hence, for each  $f_k$ , we only need to consider the mapping from  $E_{k,1}^{(n+1)}$  to  $L_1^n[2]$  and, yet, we get all the control points for each  $L_1^n$  once this mapping is considered for all k.

To get the values of  $L_1^n[2]$   $(n \ge 1)$  for  $f_k$ , we initialize the dominative control mesh  $C_k$  as follows:

$$E_{k,1} = E_k; \qquad E_{k,3} = E_{k+\frac{N}{2}};$$

$$F_{k,1} = E_{k+1}; \qquad F_{k,2} = E_{k+\frac{N}{2}-1};$$

$$F_{k,3} = E_{k+\frac{N}{2}+1}; \qquad F_{k,4} = E_{k-1};$$



Figure 4.10: Mapping the recursively generated control points in dominative control meshes to  $L_1^n$  of  $n^{th}$  GUS on  $k^{th}$  face  $f_k$ .

As mentioned before, we treat a triangular face as a special case of a quad face by merging two control points into a single point. We let  $E_{k,2}=E_{k,4}=V_k$ . Then we have:

$$V_k = E_{k,2} = E_{k,4} = \frac{3}{2}(d_V - \frac{1}{9}(E_{k,1} + E_{k,3}) - \frac{1}{36}\sum_{i=1}^4 F_{k,i})$$

This initialization guarantees that the limit point of the dominative control mesh equals  $d_V$  ( to meet Expection 1). In order to satisfy the  $2^{nd}$  expectation that the GUS surface is tangent plane continuous at the extraordinary point, we will further process the dominative control meshes such that they have the same unit normal  $n_V$ at the limit data point. The algorithm is as follows:

- (1) get the first order derivatives D<sub>u</sub>, D<sub>v</sub> at d<sub>Vk</sub>. Since C<sub>k</sub> is a part of a regular patch, it can be easily calculated.
- (2) get  $t = D_u \cdot n_V$ , the projection of  $D_u$  on  $n_V$
- (3) let  $F_{k,1} = 3t$ ,  $F_{k,4} = 3t$ ,  $F_{k,2} = 3t$ ,  $F_{k,3} = 3t$ , which ensure  $D_u \cdot n_V = 0$
- (4) get  $t = D_v \cdot n_V$ , the projection of  $D_v$  on  $n_V$



(5) let  $F_{k,1} = 3t$ ,  $F_{k,2} = 3t$ ,  $F_{k,3} = 3t$ ,  $F_{k,4} = 3t$ , which ensure  $D_v \cdot n_V = 0$ 

Figure 4.11: left: original CCS mesh and its limit surface, right: revised PCC mesh and its limit surface. The bottom left photo shows irregularity at boundaries of high-valence CCS extraordinary faces, and the bottom right is smooth.

From above algorithm and initialization, since N is even, the opposite dominative control meshes  $C_k$  and  $C_{k+\frac{N}{2}}$  will share the same set of control points, differing only in the ordering.

With all control points in  $L_1^n$  defined in equation (4.13), we are now complete with the selection process for control points in  $\mathbf{P}^n$ . Let us reinstate equation (4.8) of parameterization surface at  $f_k$  as follows:

$$S(k, u, v) = W^{T}(u)K^{n}D_{b}MS_{b}\mathbf{P}^{n}M^{T}W(v)$$

$$(4.14)$$

 $T_k = [C_{k-1} \ C_k \ C_{k+1} \ C_{k+2}]$ , is a matrix of size  $9 \times 4$ , with each column representing one of the four dominative control meshes related to  $f_k$ .  $A_9$  is defined in equation (4.12).

In this section, we have shown how to construct a GUS surface on Polar triangular faces in a PCC mesh. In next section, we will show the behavior of the PCC surfaces.

# 4.5 Evaluating the PCC surface

A PCC surface composes of two parts, CCS part and Polar part. For the CCS part, the behavior of the limit surface was already covered in [17]. In this section, we focus on the behavior of the limit surface on Polar part.

As shown in the previous sections, a GUS surface of a triangular face is  $C^2$  on the limit surface and also  $C^2$  continuous with its adjacent quad faces. We will now evaluate the surface at Polar extraordinary points. Equation (4.15) is a recursive formula , the evaluation of the GUS surface at Polar extraordinary point needs an explicit expression for  $\mathbf{P}^{n}$ . We can expand (4.15) as follows:

$$\mathbf{P}^{n} = A_{5}^{n} \mathbf{P}^{0} + A_{5}^{n-1} S_{5} A_{9}^{2} T_{k} + A_{5}^{n-2} S_{5} A_{9}^{3} T_{k} + \dots + A_{5} S_{5} A_{9}^{n} T_{k} + S_{5} A_{9}^{n+1} T_{k} = A_{5}^{n} \mathbf{P}^{0} + \sum_{i=1}^{n} A_{5}^{n-i} S_{5} A_{9}^{i+1} T_{k} \quad n \ge 1$$
(4.16)

 $A_5$  has a single eigenvalue of  $\frac{1}{8}$ , and has the following properties:

$$A_{5} = \frac{1}{8} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 5 & 1 & 0 & 0 & 0 \\ 4 & 4 & 0 & 0 & 0 \\ 1 & 6 & 1 & 0 & 0 \\ 0 & 4 & 4 & 0 & 0 \end{bmatrix}, \quad A_{5}^{2} = \frac{1}{8}^{2} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 5 & 1 & 0 & 0 & 0 \\ 20 & 4 & 0 & 0 & 0 \\ 34 & 10 & 0 & 0 & 0 \\ 36 & 20 & 0 & 0 & 0 \end{bmatrix},$$
$$A_{5}^{n} = \frac{1}{8}^{n} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 5 & 1 & 0 & 0 & 0 \\ 5 & 1 & 0 & 0 & 0 \\ 20 & 4 & 0 & 0 & 0 \\ 50 & 10 & 0 & 0 & 0 \\ 100 & 20 & 0 & 0 & 0 \end{bmatrix} = \frac{1}{8}^{n} \Theta, \quad n \ge 3$$

 $A_9$  is a 9×9 regular midpoint insertion coefficient matrix, its eigenstructure is studied in an earlier work on CCS surfaces [65]. The eigenvalues of  $A_9$  are 1,  $\frac{1}{2}$ ,  $\frac{1}{4}$ ,  $\frac{1}{8}$  and  $\frac{1}{16}$ , and we define their corresponding eigenbases as  $\Theta_1$ ,  $\Theta_2$ ,  $\Theta_3$ ,  $\Theta_4$  and  $\Theta_5$ , with

$$A_9^n = \Theta_1 + \frac{1}{2}^n \Theta_2 + \frac{1}{4}^n \Theta_3 + \frac{1}{8}^n \Theta_4 + \frac{1}{16}^n \Theta_5$$

Thus, equation (4.16) can be rewritten as:

$$\mathbf{P}^{n} = \frac{1}{8}^{n} \Theta \mathbf{P}^{0} + S_{5} A_{9}^{n+1} T_{k} + A_{5} S_{5} A_{9}^{n} T_{k} + A_{5}^{2} S_{5} A_{9}^{n-1} T_{k}$$
$$+ \sum_{i=1}^{n-3} \frac{1}{8}^{n-i} \Theta S_{5} (\Theta_{1} + \frac{1}{2}^{i+1} \Theta_{2} + \frac{1}{4}^{i+1} \Theta_{3}$$
$$+ \frac{1}{8}^{i+1} \Theta_{4} + \frac{1}{16}^{i+1} \Theta_{5}) T_{k}, \ n \ge 3$$

Since at Polar extraordinary points, when  $n \to \infty$ , the sum of n - 3 coefficients of  $\Theta S_5 \Theta_j T_k$  (j = 1, ..., 5) can be calculated explicitly.

After further simplification, we get

$$\mathbf{P}^{n}|_{n \to \infty} = (\frac{1}{8})^{n} \Theta \mathbf{P}^{0} + \kappa_{1} \Theta_{1} T_{k} + (\frac{1}{2})^{n+1} \kappa_{2} \Theta_{2} T_{k} + (\frac{1}{2})^{2(n+1)} \kappa_{3} \Theta_{3} T_{k} + (\frac{1}{2})^{3(n+1)} \kappa_{4} \Theta_{4} T_{k} + ((\frac{1}{2})^{3(n+1)} \kappa_{5} + (\frac{1}{2})^{4(n+1)} \kappa_{6}) \Theta_{5} T_{k}$$

where  $\kappa_j$ 's (j = 1, ..., 6) are the 5 × 9 coefficient matrices calculated when  $n \to \infty$ . All  $\kappa_j$ 's are constant, except for  $\kappa_4$ . Because  $\Theta$  and  $\Theta_4$  relate to the same eigenvalue of  $\frac{1}{8}$ ,  $\kappa_4$  takes value of  $(S_5 + 8A_5S_5 + 8^2A_5^2S_5 + (n-3)\Theta S_5)$ .

Since curvature continuity only involves up to second order derivatives of the parametric surface, we only need  $\kappa_1$ ,  $\kappa_2$  and  $\kappa_3$  in  $\mathbf{P}^n$ .  $\kappa_1$ ,  $\kappa_2$  and  $\kappa_3$  are  $5 \times 9$  matrices with the  $2^{nd}$  columns equal to  $[1, 1, 1, 1, 1]^T$ ,  $[1, 2, 3, 4, 5]^T$  and  $[1, 5.5, 13, 23.5, 37]^T$ , respectively, and all the other entries 0. Let  $\phi_j$  be the row vector of  $2^{nd}$  row of  $\Theta_j$ , when  $n \to \infty$ ,

$$\mathbf{P}^{n} = \begin{pmatrix} \phi_{1} \\ \phi_{2} \\ (2\phi_{2} \\ 3\phi_{3} \\ 4\phi_{4} \\ 5\phi_{5} \\ \phi_{3} \\ (2\phi_{2}) \\ (2\phi_{$$

where

$$\begin{split} \phi_1 &= \left[\frac{4}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{36}, \frac{1}{36}, \frac{1}{36}, \frac{1}{36}, \frac{1}{36}\right]\\ \phi_2 &= \left[0, \frac{1}{3}, 0, -\frac{1}{3}, 0, \frac{1}{12}, -\frac{1}{12}, -\frac{1}{12}, \frac{1}{12}\right] \cdot\\ \phi_3 &= \left[-\frac{2}{9}, \frac{5}{18}, -\frac{2}{9}, \frac{5}{18}, -\frac{2}{9}, \frac{1}{36}, \frac{1}{36}, \frac{1}{36}, \frac{1}{36}\right] \end{split}$$

Let  $\psi_1(k, v)$  be a vector of size 9 obtained by doing cubic B-spline blending on parametric value v over the 4 dominative control mesh vector  $C_{k-1}$ ,  $C_k$ ,  $C_{k+1}$  and  $C_{k+2}$ ,  $\psi_2(k, v)$  is its first order derivative and  $\psi_3(k, v)$  its second order derivative.

$$\psi_1(k,v) = T_k M^T [1, v, v^2, v^3]^T$$
  
$$\psi_2(k,v) = T_k M^T [0, 1, 2v, 3v^2]^T$$
  
$$\psi_3(k,v) = T_k M^T [0, 0, 2, 6v]^T$$

We can compute the boundary limit points (at  $u = 0, n \to \infty$ ) of the  $1^{st}$  sub-patch of the  $n^{th}$  GUS , when  $n \to \infty$ ,

$$S(k,0,v)|\Omega_{n,1} = [1,0,0,0]MS_1 \mathbf{P}^n M^T [1,v,v^2,v^3]^T$$
  
=  $\phi_1 \psi_1(k,v) = d_V$  (4.18)

By applying equation (4.18) to all faces surrounding V we can conclude that the recursively generated GUS surfaces is  $C^0$  continuous at the Polar extraordinary point.

Similarly, we can calculate all first and second order derivatives of  $f_k$  at (0,0) as follows, when  $n \to \infty$ ,

$$D_{u}(0,0)|\Omega_{n,1} = [0,1,0,0]2^{n+1}MS_{1}\mathbf{P}^{n}M^{T}[1,0,0,0]^{T}$$
  
=  $\phi_{2}\psi_{1}(k,0)$   
$$D_{v}(0,0)|\Omega_{n,1} = [1,0,0,0]MS_{1}\mathbf{P}^{n}M^{T}[0,1,0,0]^{T}$$
  
=  $\frac{1}{2}^{n+1}2\phi_{2}\psi_{2}(k,0)$  (4.19)

$$D_{uu}(0,0)|\Omega_{n,1} = [0,0,2,0]2^{2(n+1)}MS_{1}\mathbf{P}^{n}M^{T}[1,0,0,0]^{T}$$

$$= 3\phi_{3}\psi_{1}(k,0)$$

$$D_{uv}(0,0)|\Omega_{n,1} = [0,1,0,0]2^{n+1}MS_{1}\mathbf{P}^{n}M^{T}[0,1,0,0]^{T}$$

$$= \phi_{2}\psi_{2}(k,0) + \frac{1}{2}^{n+1}6\phi_{3}\psi_{2}(k,0)$$

$$D_{vv}(0,0)|\Omega_{n,1} = [1,0,0,0]MS_{1}\mathbf{P}^{n}M^{T}[0,0,2,0]^{T}$$

$$= \frac{1}{2}^{n+1}2\phi_{2}\psi_{3}(k,0) + \frac{1}{2}^{2n+2}6\phi_{3}\psi_{3}(k,0) \qquad (4.20)$$

Due to properties of the dominative control meshes we chose,  $C_k$  and  $C_{k+\frac{N}{2}}$  use the same dominative control meshes with different order of control points, i.e.,  $C_k = [V, E_{k,1}, E_{k,2}, E_{k,3}, E_{k,4}, F_{k,1}, F_{k,2}, F_{k,3}, F_{k,4}]^T$ ,  $C_{k+\frac{N}{2}} = [V, E_{k,3}, E_{k,4}, E_{k,1}, E_{k,2}, F_{k,3}, F_{k,4}, F_{k,1}, F_{k,2}]^T$ . We can get  $\phi_2 \psi_i(k, 0) = -\phi_2 \psi_i(k + \frac{N}{2}, 0), \quad i = 1, 2, 3$  $\phi_3 \psi_i(k, 0) = \phi_3 \psi_i(k + \frac{N}{2}, 0), \quad i = 1, 2, 3$  (4.21)

Let us define the first order and second order derivatives of  $f_{k+\frac{N}{2}}$  as  $D'_u$ ,  $D'_v$ ,  $D'_{uu}$ ,  $D'_{uv}$ and  $D'_{vv}$ . With (21), we can get  $D_u = -D'_u$  and  $D_v = -D'_v$ ,  $D_{uu} = D'_{uu}$ . However,  $D_{uv} \neq D'_{uv}$  and  $D_{vv} \neq D'_{vv}$ .

 $C^1$  continuity is trivial from above results.

For curvature continuity, it is not intuitive, because  $D_{uv} \neq D'_{uv}$  and  $D_{vv} \neq D'_{vv}$ . However there is a special property for values of  $D_{uv}$  and  $D_{vv}$ . Note that the first term of each of them takes value of a linear combination of  $D_u$ 's of dominative control meshes  $(D_u|C_k = \phi_2 C_k)$ , which we designate to have the same unit normal  $n_V$ . So

$$D_{vv} \cdot n_V = \left(\frac{1}{2}^{2n+2} 6\phi_3 \psi_3(k,0)\right) \cdot n_V$$
$$D_{uv} \cdot n_V = \left(\frac{1}{2}^{n+1} 6\phi_3 \psi_2(k,0)\right) \cdot n_V$$

From equation (4.21), we can derive that  $D_{uv} \cdot n_V = D'_{uv} \cdot n_V$  and  $D_{vv} \cdot n_V = D'_{vv} \cdot n_V$ . We can further derive that the second fundamental forms are equal at (0,0) for  $f_k$ and  $f_{k+\frac{N}{2}}$  (the opposite faces). Therefore, they have the same Gaussian curvature.

We can now conclude that the PCC surface is curvature continuous everywhere on the Polar parts.

## 4.6 Summary

In this work, a new subdivision scheme with Polar embedded Catmull-Clark mesh structure is introduced. By introducing Polar configuration on high valence vertex, the ripple problem inherent in a CCS surface is solved.

The subdivision scheme developed has the properties that the limit surface on the CCS part is exactly the same as a CCS limit surface and the limit surface on the Polar part is  $G^2$  continuous everywhere.

Since it is inevitable to have high valence extraordinary points in some cases, e.g. airplanes, rockets and engineering parts, the currently available CCS meshes can be easily converted to PCC meshes, such that one can avoid redesigning the complete mesh.

In contrast to commonly used Polar subdivision rules, the subdivision masks of proposed GUS subdivision scheme on Polar part is equivalent to those of CCS. The properties of GUS surfaces are studied and proven. The GUS scheme is a stationary scheme.

The curvature at a Polar extraordinary point is independent of nearby control points, but relies on some selected dominative control meshes. Implementation results (Fig. 4.12) show that very high quality, curvature continuous subdivision surfaces can be generated with this new scheme on the Polar part. Furthermore, the scheme is WYSIWYG (what you see is what you get): as far as the ring of control points connected around the Polar extraordinary point is smooth, there will be no ripples.

Our next step is to develop a general geometric framework to incorporate some  $G^2$ schemes for CCS meshes into the PCC subdivision scheme, so that a  $G^2$  everywhere PCC surface can be generated.



Figure 4.12: Various primitives of GUS surfaces on Polar parts

# Chapter 5

# **Bezier** Crust



Figure 5.1: Two examples of Bezier crust applied on Catmull-Clark subdivision surfaces

As we discussed before, Subdivision surfaces can be classified into two groups, approximating and interpolating. Representatives of approximating schemes are Catmull-Clark [6] (quad) and Loop [43] (triangular). Approximating schemes are widely used in computer graphics and animation, because their limit surfaces are generally of higher quality than interpolating ones. Although their wide application, for some applications, especially in CAD/CAM, interpolating schemes are preferred. Efforts have been made to convert these approximating scheme into interpolating ones. However, one issue remains in current interpolating schemes for Catmull-Clark and Loop, the interpolation of data points are global such that it will be difficult to interpolate when data point set is large . In this work, we present a local interpolation

scheme on quad subdivision surfaces by appending a  $G^2$  Bezier crust, and we show that this local interpolation scheme will not change the curvature on the boundaries of underlying subdivision patches, such that one can obtain high quality interpolating limit surface for engineering and graphics usage efficiently.

#### 5.1 Introduction

Subdivision surfaces have been widely used in surface representation. Comparing to traditional spline methods (e.g. Bezier Surface), they are simpler and are able to work on arbitrary topology.

Subdivision schemes have mainly three types of mesh structure, quadrilateral, triangular and hexagonal. Quad faces and Triangular faces are mostly commonly used for practical purpose. Subdivision surfaces can be classified into two types, face-split and vertex-split. Vertex-split schemes (midedge [17], biquartic [57]) are not as popular as those of face-split because on arbitrary topology they do not generate well behaved surfaces as face-split ones. In face-split subdivision schemes, vertices of control meshed are refined recursively. Each vertex of current control mesh is redefined in next subdivision level. If the original vertex and its corresponding vertex in next subdivision step are the same, we call this scheme interpolating (e.g. Modified Butterfly [18], Kobbelt [34]), otherwise the scheme is approximating (e.g. Loop [43], Catmull-Clark [6]). Interpolating is attractive, since the vertices in the original control mesh will be contained in the control meshes in subsequent subdivisions, which makes subdivision more intuitive. However, the surface quality of interpolating schemes is not as good as those of approximating ones. As a comparison, interpolating schemes of Modified Butterfly and Kobbelt scheme are  $C^1$  continuous on regular mesh, while approximating schemes of Catmull-Clark and Loop are  $C^2$  continuous on regular mesh. Overall, among variuous subdivision schemes, Loop and Catmull-Clark are the most widely used on triangular mesh and quadrilateral mesh respectively.

Both Loop and Catmull-Clark are approximating schemes, the limit surface generated will not interpolate the control mesh. However, the construction of smooth interpolating surfaces is important in many applications including CAD design, statistical data modelling and face recognition. In this work, we focus on interpolation of quad data mesh, especially on Catmull-Clark scheme.

Typical input to an interpolating method is a control mesh with a collection of data points to be interpolated, we call this control mesh as "data mesh", sometimes normals are specified on these data points, which is usually required in traditional spline method (Bezier surface). It is generally difficult to construct a  $G^2$  piecewise Tensor-Product Bezier surface with well bahaved limit surface without carefully picked normals. In contrast, for subdivision surfaces, the subdivision mask is predetermined, so surface normals are determined by subdivision.

Given a data mesh, in Catmull-Clark scheme first step is to calculate its control mesh which generates the limit surface interpolating given data mesh. There are two methods available, direct and iterative. Earlier work of Harlstead [29] solves interpolation problem by solving a square linear system. However, the direct method (calculate inverse of matrix) is not feasible, when control points set are typically of hundreds.

In this work, we present a  $G^2$  scheme on quad subdivision surfaces, called *Bezier Crust.* We show that, with parametrically appending piecewise bi-quintic Bezier crust on quad subdivision surfaces like CCSS, the generated interpolating limit surface will maintain the surface continuity of original approximating schemes.

## 5.2 Previous Works

In this section, we discuss earlier methods for interpolating schemes of Catmull-Clark and  $G^2$  Bezier surface. In CAGD and computer graphics, a smooth limit surface is required. In this respect, tangent plane continuous ( $G^1$ ) or curvature continuous  $G^2$ surface is required to obtain visual smoothness. In this work, we focus on the  $G^2$ surface, which is appropriate for most engineering and graphics usage.

#### Interpolating Schemes of Catmull-Clark

Catmull-Clark subdivision (CCS) is the most widely used subdivision scheme. Control points in CCS control mesh can be categorized into vertex, edge and face [6]. In each CCS, a new face point is inserted for each face, a new edge point is inserted in each edge, original vertex points are updated by applying subdivision rules. By recursively subdividing, one can obtain a limit surface which is  $C^2$  everywhere except at extraordinary, where it is  $C^1$  (tangent plane) continuous [1] [17].

Interpolation of CCS can be performed by solving a linear system,

$$\mathbf{A}x = b \tag{5.1}$$

where  $\mathbf{A}$  is a square matrix determined by interpolation conditions and mesh topology, x is the column vector of control points in unknown control mesh, b is the corresponding column vector of data points in the given data mesh [29]. If  $\mathbf{A}$  is of small size and nonsingular, we can directly obtain the control mesh by calculating  $\mathbf{A}^{-1}$ . However, by interpolation conditions and size of data mesh,  $\mathbf{A}$  can be singular or of larger size, then direct method will not work or not work well. In this case, an iterative method needs to be applied. Traditionally, stationary iterative methods like Jacobi, Gauss-Seidal or Successive Over-relaxation can be used to solve this large linear system. The issues with these methods are the convergence rate, they are slow when data set is large. When **A** is singular, then least-squares method can be applied. There are some faster iterative methods developed to solve larger scale data set [3] [64], however since (1) is global, when we are handling thousands of data points, convergence rate will still be not satisfatory.

Despite the convergence speed, the interpolating surface obtained by solving equation (5.1) is unsatisfatory because of its excessive undulations [29]. Halstead [29] notes that the undulations appear because they are not indicated by the shape of original mesh, e.g. the surface has a number of concavities while the original mesh is convex. In this case, it is necessary to optimizing fairness norm (combination of the energy of a membrance and a thin plate) by introducing additional degrees of freedom into the surface. The Fairing techniques [45] [70] smooth interpolating surface by including more constraints and increasing size of control mesh.

Since the global method of solving equation (5.1) is unsatisfactory due to its convergence speed or surface artifact, we have the following research question arising: "Is it possible to have a local scheme instead of global one, such that the computation time can be significantly reduced, while preserving the curvature properties of CCS?"

#### $G^2$ Bezier Surface

In CAGD, piecewise tensor-product Bezier Patch is one of the most widely used models in free-form surface modeling. Each Bezier Patch interpolates the control points at four corners, such that an interpolating scheme is natural in the surface construction.

Two-dimensional Bezier surface can be parameterized,

$$\mathbf{p}(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} \mathbf{b}_{i,n}(u) \mathbf{b}_{j,m}(v) \mathbf{P}_{i,j},$$
(5.2)

where  $\mathbf{b}_{i,n}(u)$  and  $\mathbf{b}_{j,m}(v)$  are the Bernstein coefficients on u direction and v direction respectively,  $\mathbf{P}_{i,j}$  is the control point at (i, j). Since the commonly used Bezier surface has m = n, so here we focus on piecewise tensor-product Bezier surface only.

From the definition of Bezier surface, four corner control points are interpolated by its limit surface. Conditions of  $G^1$  continuity have been discussed in [4] [14] [42]. It was pointed out, to obtain  $G^1$  continuity, one must ensure that partial derivatives cross the boundary of Bezier patches  $(n \ge 2)$  must be coplanar.

In CAGD,  $G^2$  continuity is necessary to ensure the existence of visually well behaved surfaces. Conditions for  $G^2$  continuity are discussed in [12] [32] [69]. From these previous work, a piecewise bi-quintic Bezier patch scheme is necessary to get  $G^2$  continuous.

Although one can theoretically construct a  $G^2$  Bezier surface, it is known that construction of such  $G^2$  surface with Bezier surface is more difficult than with subdivision surface. One has to solve linear systems of partial derivatives up to second order across the boundaries, and the linear system has too much freedom. Gregory reduces the freedom by introducing additional constraints on internal control points of Bezier patch [26], while its construction is still not an easy task.

Above we introduced two main interpolating schemes based on subdivision surface and Bezier patch. The first scheme suffers problem of convergence speed and unnecessary undulations, while the latter one is more difficult to construct.

# 5.3 Bezier Crust on Quad Subdivision Surface

In this section, we introduce a new interpolating scheme for approximating subdivision surfaces like Catmull-Clark. First, we start from interpolating the space curves.

#### Bezier Crust on Space Curve

Bezier surface can be treated as the tensor-product of Bezier curves. A Bezier curve takes the following form,

$$\mathbf{B}(t) = \sum_{i=0}^{n} \mathbf{b}_{i,n}(t) \mathbf{P}_{i},$$
(5.3)

where  $\mathbf{b}_{i,n}(t) = {n \choose i} t^i (1-t)^{n-i}$ , i = 0, ..., n (Bernstein basis polynomials of degree n),  $\mathbf{P}_i$  is its control point.

Bezier spline is defined as the spline formed by patching together piecewise Bezier curves. which can represent complex shape. Bezier spline interpolates all starting and ending control points of its Bezier curves. While quadratic and cubic Bezier splines are mostly used in font and 3D animation, they are not  $G^2$  continuous between adjacent Bezier curves. To obtain a  $G^2$  continuous spline, quintic Bezier curves are needed to construct a Bezier spline which is  $G^2$  everywhere.

Given a constructed  $G^2$  continuous quintic Bezier spline, if we want to change the interpolating points but maintain  $G^2$  continuity along the spline, a simple solution can be achieved by,

$$\mathbf{B}(t)' = \mathbf{B}(t) + \sum_{i=0}^{5} \mathbf{b}_{i,5}(t) \Delta \mathbf{P}_i, \qquad (5.4)$$

where  $\Delta \mathbf{P}_0 = \Delta \mathbf{P}_1 = \Delta \mathbf{P}_2$  and  $\Delta \mathbf{P}_3 = \Delta \mathbf{P}_4 = \Delta \mathbf{P}_5$ . The new quintic curve  $\mathbf{B}(t)'$  is obtained by moving the first three control point of  $\mathbf{B}(t)$  with the same vector of  $\Delta \mathbf{P}_0$ and the last three control point with  $\Delta \mathbf{P}_5$ , such that it maintains  $C^2$  on the curve.

After calculating the first and second derivative at starting and ending points of  $\mathbf{B}(t)'$  and  $\mathbf{B}(t)$ , we can prove that the first and second derivates at the starting and ending points of  $\mathbf{B}(t)'$  are the same as those of  $\mathbf{B}(t)$ . We can further prove that the new Bezier spline obtained is also  $G^2$  continuous.



Figure 5.2: Control points of  $\mathbf{B}(t)'$  after movement of  $\Delta \mathbf{P}(t)$  from original  $\mathbf{B}(t)$ .

If we define

$$\Delta \mathbf{B}(t) = \sum_{i=0}^{5} \mathbf{b}_{i,5}(t) \Delta \mathbf{P}_{i}, \qquad (5.5)$$

we find the following properties of  $\Delta \mathbf{B}(t)$  of degree 5:

- (1) when  $\Delta \mathbf{B}(t)$  is displayed alone, it is a line segment independent of its degree. So it is  $C^2$  on  $\Delta \mathbf{B}(t)$  except at starting and ending points.
- (2)  $1^{st}$  and  $2^{nd}$  derivatives on starting and ending points of  $\Delta \mathbf{B}(t)$  are vanished. So it will not change  $1^{st}$  and  $2^{nd}$  derivatives on starting and ending points of  $\mathbf{B}(t)$ .
- (3) The new quintic Bezier spline obtained by adding  $\Delta \mathbf{B}(t)$  to its each Bezier curve will maintain  $G^2$  continuous if the original Bezier spline is  $G^2$ .

With above properties, we name  $\Delta \mathbf{B}(t)$  of degree 5 as Quintic Bezier crust. We notice that this Quintic Bezier crust can be added to an arbitrary  $C^2$  curve.

**Theorem 5.1.** New curve obtained by parametrically adding a Quintic Bezier crust to a  $C^2$  parametric curve is  $C^2$  continuous and has the same curvature on the starting and ending point as the original curve.

*Proof.* A  $C^2$  parametric curve can be writen in polynomial form at a parametric value  $t_0$  as

$$f(t) = f(t_0) + f'(t_0)(t - t_0) + \frac{f''(t_0)}{2}(t - t_0)^2 + \delta.$$
 (5.6)

New curve  $f(t)' = f(t) + \Delta \mathbf{B}(t)$ , by calculating its first and second derivatives, we can prove that f(t)' is  $C^2$  on the curve and has the same curvature on the starting and ending points as f(t).

#### Bezier Crust on Quad Subdivision Surface

Subdivision surfaces have been widely used in surface representation. With their simplicity and well behaved limit surfaces, approximating schemes are popular in computer graphics and animation. Most approximating schemes work on quadrilateral or triangular meshes. Within various quad schemes, Doo Sabin [17], Mid-Edge [54] are  $C^1$  continuous, Catmull-Clark [6] is  $C^2$  everywhere except at extraordinary points. In this work, we present a new unified interpolating scheme for quad approximating subdivision surfaces, with main focus on Catmull-Clark.

Similar to quintic Bezier crust, we define a bi-quintic Bezier crust  $\Delta \mathbf{p}(u, v)$  on subdivision surfaces as follows, given a quad subdivision scheme,  $\Delta \mathbf{P}_0$ ,  $\Delta \mathbf{P}_1$ ,  $\Delta \mathbf{P}_2$ and  $\Delta \mathbf{P}_3$  are difference vectors between its corner control points and its corresponding data points respectively,

$$\Delta \mathbf{p}(u,v) = \sum_{i=0}^{5} \sum_{j=0}^{5} \mathbf{b}_{i,5}(u) \mathbf{b}_{j,5}(v) \Delta \mathbf{P}_{i,j},$$
(5.7)

where  $\Delta \mathbf{P}_{i,j}$  is control point of bi-quintic Bezier surface, and  $\Delta \mathbf{P}_{i,j} = \Delta \mathbf{P}_0$  if  $i \in [0,2]$  &  $j \in [0,2]$ ,  $\Delta \mathbf{P}_{i,j} = \Delta \mathbf{P}_1$  if  $i \in [0,2]$  &  $j \in [3,5]$ ,  $\Delta \mathbf{P}_{i,j} = \Delta \mathbf{P}_2$  if  $i \in [3,5]$  &  $j \in [0,2]$ ,  $\Delta \mathbf{P}_{i,j} = \Delta \mathbf{P}_3$  if  $i \in [3,5]$  &  $j \in [3,5]$ .



Figure 5.3: Control points of  $\mathbf{B}(t)'$  after movement of  $\Delta \mathbf{P}(t)$  from original  $\mathbf{B}(t)$ .

When displayed alone, the Bezier crust defined in equation (5.7) has exactly the same limit surface as a bilinear Coons patch, so Bezier crust can be treated as bilinear Coons patch with degree elevation. By analysis on the 1<sup>st</sup> and 2<sup>nd</sup> order derivatives, we find that 1<sup>st</sup> and 2<sup>nd</sup> order derivatives on the four corners and cross boundary of Bezier crust vanishes. On the Bezier crust, we can also prove that it is  $C^2$  continuous with zero Gaussian curvature everywhere (property of Coons patch) except when the difference vectors on the four corners coincide or a twist effect exist, then it is possible that 1<sup>st</sup> order derivative vanish at some (u, v).

We can derive the properties of bi-quintic Bezier crust as follows:

(1) On the four corners, the  $1^{st}$  and  $2^{nd}$  order derivatives vanish.

- (2) On the four boundaries, the  $1^{st}$  and  $2^{nd}$  order derivatives cross the boundaries vanish.
- (3) On (u, v) of the Bezier crust, the 1<sup>st</sup> order derivative is continuous or vanishes, and 2<sup>nd</sup> order derivative is continuous and lies on its tangent plane or vanishes.
- (4) When displayed alone, Bezier crust has zero Gaussian curvature everywhere.

Given an arbitrary parametric subdivision surface S(u, v), we define its interpolating surface S(u, v)' after adding Bezier crust as follows,

$$S(u,v)' = S(u,v) + \Delta \mathbf{p}(u,v)$$
(5.8)

**Theorem 5.2.** Interpolating surface patch S(u, v)' obtained by adding a bi-quintic Bezier crust  $\Delta \mathbf{p}(u, v)$  to a  $C^2$  parametric subdivision surface patch S(u, v) interpolate the four corner control points of S(u, v). Its limit surface is  $C^2$  continuous and has the same curvature on the four corners and the same cross-boundary curvature along the boundaries as the original surface patch.

*Proof.* The proof is trivial by properties of Bezier crust mentioned above. By properties (1) and (2), the first order derivatives and the second order derivatives on four corners and cross the boundaries of Bezier crust vanishes, so the curvature maintain the same on the corners and cross the boundaries. By property (3), we can derive that on the limit surface it maintains  $C^2$  continuity.

With Theorem 5.2, we can further conclude that if underlying piecewise parametric surface patches form a  $C^1/C^2$  surface, then the interpolating surface formed by parametrically adding their corresponding piecewise Bezier crust is also  $C^1/C^2$ continuous.

This can be proven by Properties (1) and (2) of bi-quintic Bezier crust on the corner points and cross-boundaries. Fig. 5.4 shows linear displacement of original limit surface at corner points and cross-boundaries by adding a Bezier crust.



Figure 5.4: Displacement of limit surface at corner data points and cross-boundary data points by applying Bezier crust

Above we showed the construction of bi-quintic Bezier crust on parametric quad subdivision surfaces. Next, we will show the implementation of Bezier crust on Catmull-Clark subdivision surfaces.

# 5.4 Implementation and Discussion

Catmull-Clark subdivision (CCS) surfaces are widely used in computer graphics and animation. The CCS is an approximating schemes, the limit surface will approximate the control mesh but not interpolating. The interpolating schemes developed so far for CCS mainly deal with a global linear system, and they suffer from the convergence speed and are difficult to get well-behaved interpolating surfaces when data set is large.

CCS surfaces can be parameterized both on regular and extraordinary faces, such that it is possible to add a piecewise Bezier crust on each individual surface patch without impacting it curvature continuity. Note that parametrically adding a piecewise Bezier crust shown in equation (5.7) is local, which means that the calculation of limit surface can be done locally. Its running time is much less than the earlier global interpolation schemes. The algorithm will be stable and compact. Fig. 5.1



Figure 5.5: Top shows limit surface of CCS and its control mesh, bottom shows limit surface and control mesh by applying our new interpolating scheme

shows two examples of adding piecewise Bezier crusts on engineering parts. Fig. 5.5 shows limit surface of original CCS and limit surface of our new interpolating scheme. From photo, we can see that our new scheme interpolates original control mesh well and the limit surface has no unwanted undulations. The implementations show that adding Bezier crusts to CCS surface will generate visually smooth interpolating surfaces which will satisfy most CAD and CAGD usage.
We notice some limitations from Fig. 5.6. When bi-quintic Bezier patch is displayed standalone,  $\Delta \mathbf{p}(u, v)$  is enclosed by volume of  $\Delta \mathbf{P}_0$ ,  $\Delta \mathbf{P}_1$ ,  $\Delta \mathbf{P}_2$  and  $\Delta \mathbf{P}_3$ . Such that by adding Bezier crust to the underlying convex surface, the generated interpolating surface will show diminishing effect on curvature towards the center. From Fig. 5.5, we can see slightly flattened surface areas, however we have not seen the adverse effects yet on the interpolating surface on CCS yet. More experiments need to be done on other quad subdivision surfaces to see if this will cause any unwanted surface artifacts. Also, we notice that it is possible to have twisting effect when Bezier Crust shown standalone (see the right of Fig. 5.6). However, because of the compensation from underlying subdivision surface, we have not noticed the adverse impact on surface quality of our new interpolating surface (see Fig. 5.1, 5.5 and 5.7) caused by the twisting effect of Bezier Crust .



Figure 5.6: Two scenaria of Bezier crust, when shown standalone

#### 5.5 Summary

In this work, we introduce a simple interpolating scheme for quad parametric subdivision surface. We show that by adding a special bi-quintic Bezier crust on original subdivision surface one can generate an interpolating surface which maintains curvature conditions of original subdivision surface. With special construction of bi-quintic Bezier crust, we can avoid to calculate a global linear system common in earlier interpolation schemes, such that the computation is local and simple.

Implementation results on Catmull-Clark subdivision surfaces show that the Bezier crust interpolating scheme can generated visually well behaved limit surfaces, such that barely no fairing needed to correct the undulations caused by computation of global linear system used in earlier interpolation schemes.

The Bezier crust on quad subdivision surface shows also advantages over direct Bezier surface methods. To obtain  $C^2$  surface, it is required to have bi-quintic piecewise Bezier surface, however, the Bezier surface method requires also interpolation of normals on vertices, which is complex and difficult to obtain a well behaved surface when data set is large.

Our Bezier crust interpolating scheme is limited to quad subdivision surface. For interpolating of triangular subdivision surfaces (e.g. Loop) which are also popular in computer graphics, we will put it into our next research.

Overall, in this work we provide a local  $G^2$  interpolating scheme for quad subdivision surface. With simplicity of this new scheme, it can be easily applied to quad approximating subdivision surfaces and convert them to interpolating schemes, making them more appropriate for CAD, CAGD, face recognition and other interpolation required applications.



Figure 5.7: Two examples. Top to bottom: interpolating surface, interpolating surface shown with control mesh, CCS limit surface, CCS limit surface shown with control mesh.

# Chapter 6

# G2 Interpolation on Polar Surfaces

In this work, we present a  $G^2$  interpolating scheme for Polar surfaces, so that polar surfaces can be used in high precision CAD/CAM applications as well. The new scheme is Bezier crust based, i.e., the interpolating surface is generated by parametrically attaching an especially selected bi-degree 5 Bezier surface to a Polar surface. While Bezier crust based scheme handles quad faces only [66], we show that through a conversion process, we can handle triangular faces in the Polar part as well. Curvature continuity of the generated limit surface of our new scheme is consistent with the corresponding Polar surface. In case of a PCCSS [67], the generated limit surface of our new scheme is  $G^2$  on the Polar part.

#### 6.1 Introduction

Subdivision surfaces have been widely used in CAD, gaming and computer graphics. Catmull-Clark subdivision (CCS) [6], based on tensor product bi-cubic B-Splines, is one of the most widely used subdivision schemes. The surfaces generated by the scheme are  $C^2$  continuous everywhere except at extraordinary points, where they are  $C^1$  continuous. A shortcoming inherent in CCS surfaces is the ripple problem, that is, ripples tend to appear around an extraordinary point with high valence. In the past, research focused on improving curvature distribution at extraordinary points. However, with quad mesh structure of CCS surfaces, ripples could not be avoided in high valence cases. To handle this artifact, Polar surface are studied by a number of researchers. A Polar surface has a quad/triangular mixed mesh structure. A bicubic Polar subdivision scheme is presented in [33] which sets up the control mesh refinement rules for Polar configuration so that the limit surface is C1 continuous and curvature bounded. A Polar surface handles high valence cases well, but there are some issues to solve for connecting them to Catmull-Clark meshes. For instance, because of the mismatch on the mesh between radial subdivision and Catmull-Clark subdivision, in [49], given a polar vertex of valence n, at the kth level, its generalized bi-cubic subdivision scheme generates  $2^k$  subfaces and expands the valence to  $2^k n$ . Recently a new subdivision scheme was developed in [67]. This new scheme, named PCCSS, subdivides triangular faces in Polar embedded Catmull-Clark (PCC) mesh without generating exponential number of subfaces and without doubling valences in each subdivision step, and its limit surface is  $G^2$  at Polar extraordinary points. The polar surface can handle high valence very well. However, all current polar subdivision schemes are approximating, i.e. the generated limit surface will not interpolate the given control mesh. Given the complexity of quad/triangular mesh structure, no known interpolation scheme was developed yet. But, since many applications require an interpolation scheme, Polar surface is not well adopted in CAD/CAM. In this work, we present a  $G^2$  interpolating scheme on Polar surface, such that it can be used in high precision CAD/CAM application. Our new scheme is based on Bezier crust [66], where an interpolating surface was generated by parametrically adding Catmull-Clark subdivision surface and a special selected bi-degree 5 Bezier surface. Although scheme of Bezier crust handles quad faces only, we show by conversion, we can handle triangular faces in Polar mesh as well. The curvature continuity of generated limit surface of our new scheme is consistent with the corresponding Polar surface, in case of the PCCS [67], it is  $G^2$  on Polar extraordinary points,  $C^1$  at CCS extraordinary points, and everywhere else  $C^2$ .



Figure 6.1: A Polar example. Left: Polar mesh; middle: two views of PCCS surface; right: two views of our new Interpolation surface.

#### 6.2 Polar surface and PCCSS

In this section, we review earlier works on Polar surfaces. A Polar surface has the following properties on its mesh structure: faces adjacent to extraordinary points are triangular, all other faces are regular. A typical Polar mesh is shown in Fig. 6.2(a). A Polar surface has a quad/triangular mixed mesh structure. Peters and Karciauskas [53] introduce concept of Polar surface. A bi-cubic Polar scheme is presented in [33] that sets up the control mesh refinement rules for Polar configuration such that the limit surface is C1 continuous and curvature bounded. A  $C^2$  Polar surface is shown in [50] by modifying weights of the Polar subdivision for different valences. Traditional Polar surface can handle high valence ripple problems inherent in Catmull-Clark Subdivision (CCS) surfaces very well, but it is difficult to design a traditional Polar surface for a complex object with thousand of control points. Efforts are made to combine Polar meshes and CCS meshes. In [49], mesh refinement on Polar part is

done in two steps at the  $k^{th}$  level, (1) k times radially, then (2) k times circularly. This scheme doubles valence in each subdivision steps, i.e. given a Polar vertex of valence n, after k subdivisions, the Polar valence is expanded to  $2^k n$ . Recently a new scheme is presented in [67], this new scheme works on Polar/CCS hybrid mesh structure (as shown in Fig. 2(c)), named as Polar Embedded Catmull-Clark Subdivision (PCC) mesh. PCC mesh allows extraordinary points to exist also in the quad mesh part. The new subdivision scheme PCCS [67] comes with the following properties and improvements: (1). In each subdivision step, the Polar valence does not double, instead, it remains the same. (2). Only O(k) sub-surfaces are generated after k subdivisions. (3). A natural C2 join between Polar part and CCS part.



Figure 6.2: From left to right: Polar mesh, CCS mesh, PCC mesh

The subdivision scheme of PCCSS works as follows. Given an arbitrary mesh, first step is to convert it to CCS mesh with all quad faces and no two extraordinary points neighboring to each other, this is done by up to twice CCS subdivisions. Then, we replace faces surrounding high valence extraordinary points with Polar structure shown in Fig. 6.2(a). The obtained PCC mesh structure is shown in Fig. 6.2(c). PCC mesh is more generalized than both CCS mesh and Polar mesh. If PCC mesh has Polar extraordinary points, then it is Polar mesh, and if PCC mesh has only CCS extraordinary points, then it is CCS mesh. So in this paper, we focus our work on the subdivision surface PCCSS on PCC mesh.

The Polar extraordinary point can have a valence of even or odd. Since the odd



Figure 6.3: Convert odd Polar valence to even by one subdivision

valence is more difficult to achieve in terms of curvature continuity, a preprocessing step is performed by a special subdivision (Fig. 6.3). The new edge point and face points of triangular faces are defined by CCS rules, but for a new vertex point, it uses the original CCS rule on arbitrary topology, an affine combination of old vertex point, new face points and new vertex points. After valence conversion, the PCCSS uses Guided U-subdivision (GUS) for consequent subdivisions. The GUS is shown in Fig. 6.4. Each GUS will generate 5 layers of control points, control points in the last three layers (red dots in Fig. 6.4(a)) are generated by CCS equivalent U-subdivision on the first three layers of last subdivision step (black circles in Fig. 6.4(a)). The control points in the first layer (blue dots in Fig. 6.4(a)) are selected from the dominative control meshes (as shown in Fig. 6.4(b)). The control points in the second layers of are selected by a process called virtual U-Subdivision, i.e. these control points are reverse calculated from the new 1st layer control points and the last three layer control points in previous GUS.

In [67], it shows that the limit surface of PCCSS as above is  $C^2$  everywhere except at extraordinary points, where it is  $G^2$  at Polar extraordinary points and  $C^1$  at CCS extraordinary points.



Figure 6.4: Guided U-Subdivision and mesh blending (a) Guided U-Subdivision (b) select 1st layer control points in GUS

# 6.3 A Heuristic interpolation scheme on PCCSS

Subdivision schemes can be classified into two types. If the original vertices in the control mesh is the same as its corresponding limit points after subdivisions, we call such scheme interpolating, otherwise, the scheme is approximating. Current Polar schemes are all approximating. Although Polar surface can handle high valence ripples common in Catmull-Clark Subdivision surface, for high precision CAD/CAM usage, an interpolating scheme is highly desirable. Due to the triangle/quad mixed mesh structure, no known interpolating scheme was developed so far. In this section, we present a heuristic G2 interpolation scheme on PCCSS.

In PCCSS, A PCC mesh can be separated into two parts, Polar part and CCS part. The limit surface on CCS part is exactly the same as that of the CCSS. So it makes it possible to construct an interpolating scheme at quad faces the same as the interpolating scheme for CCSS. Interpolation of CCSS is traditionally performed by solving a global linear system of

$$Ax = b \tag{6.1}$$

Where A is the coefficient matrix determined by CCS subdivision rules, x is the column vector of control points to be determined, and b is the column vector of data

points in the given data mesh [29]. Stationary iterative methods like Jacobi, Gauss-Seidel or Successive Over-relaxation can be used to solve equation (3.1). However when data set is large, the convergence rate of above methods is slow. Some faster iterative methods [3] [8] [9] [64] were developed to improve the convergence rate. However, the iterative methods above suffer excessive undulation [29]. To improve the shape of interpolation surface, fairing techniques are required and the final shape of interpolation surface are non-predictable. Recently, a direct interpolation scheme called Bezier Crust is introduced in [66]. The idea of our new scheme is to apply a bi-degree 5 piecewise specially selected Bezier Surface on CCSS, such that the interpolating surface can be generated in one step instead of iterations. Piecewise bi-degree 5 Bezier surface is the necessary condition to obtain a  $G^2$  limit surface, but its computation is generally costly and not a simple task. Bezier crust is a simplified Bi-degree 5 Bezier surface, in that its 1st and 2nd order derivatives vanishes at boundaries of each patch. Given a CCS mesh M, The limit surface of each face f of M (regular or extraordinary) can be represented in parametric form S(u,v). For each f,  $\Delta P_0$ ,  $\Delta P_1$ ,  $\Delta P_2$ , and  $\Delta P_3$  are defined as the difference vectors between the corner control points and their corresponding CCS limit points, respectively. In order to interpolate the given control points, a bi-quintic Bezier crust  $\Delta S(u, v)$  is defined as follows,

$$\Delta S(u,v) = \sum_{i=0}^{5} \sum_{j=0}^{5} b_{i,5}(u) b_{j,5}(v) \Delta P_{i,j}$$
(6.2)

With  $\Delta P_{i,j}$  takes value of  $\Delta P_0$ ,  $\Delta P_1$ ,  $\Delta P_2$ , and  $\Delta P_3$ .  $\Delta P_{i,j} = \Delta P_0$  if  $i \in [0, 2]$  and  $j \in [0, 2]$ ;  $\Delta P_{i,j} = \Delta P_1$  if  $i \in [0, 2]$  and  $j \in [3, 5]$ ;  $\Delta P_{i,j} = \Delta P_2$  if  $i \in [3, 5]$  and  $j \in [0, 2]$ ;  $\Delta P_{i,j} = \Delta P_3$  if  $i \in [3, 5]$  and  $j \in [3, 5]$ ;  $\Delta P_0$ ,  $\Delta P_1$ ,  $\Delta P_2$ , and  $\Delta P_3$  are the difference vectors at four corners of a CCS face (Fig. 6.5). With offsetting Bezier crust  $\Delta S(u, v)$  defined, the interpolating parametric surface  $\overline{S}(u, v)$  can be expressed as follows:

$$\overline{S}(u,v) = S(u,v) + \Delta S(u,v) \tag{6.3}$$



Figure 6.5: Difference Vectors between control points and limit points, (a) regular face (b) extraordinary face (c) offsetting Bezier Crust

As discussed in last chapter, the offsetting Bezier Crust has the following properties:

- 1st order and 2nd order derivatives vanish across the face boundaries and at 4 corners.
- Underlying subdivision rules independent, can handle arbitrary quad subdivision surfaces.
- $C^2$  on each Bezier Crust.

The new CCS interpolation surface obtained by equation (3.3) has the following properties:

- $C^2$  everywhere except at extraordinary points, where it is  $C^1$  continuous
- Interpolates not only data points, but also the surface normal and curvature at these data points.

Since PCCSS has the same limit surface as CCS at quad face part, so on quad faces, equation (6.3) can be applied to obtain its interpolation limit surface. For Polar

faces, these faces are triangular, equation (6.3) cannot be applied directly. However, we note that the PCCSS treats Polar faces as quad faces by technique of vertex splitting (Fig. 6.6).



Figure 6.6: Polar face conversion by vertex splitting

By vertex splitting in PCCSS, a Polar extraordinary point V is duplicated at each Polar face, such that the Polar face can apply Bezier Crust as well. The limit surface of each Polar face f in PCCSS can be represented in parametric form  $S(u, v)|_p olar$ . For each f,  $\Delta P_0$ ,  $\Delta P_1$ , and  $\Delta P_2$  (Fig. 6.7(a)) are defined as the difference vectors between the corner control points and their corresponding PCCSS limit points respectively. By vertex splitting of  $\Delta P_0$  (Fig. 6.7(b)), we obtain 4 difference vectors on each converted quad face (Fig. 6.7(c)). In order to interpolate the difference vectors at corners of Polar face f, a bi-quintic Bezier crust  $\Delta S(u, v)|_{polar}$  is defined as follows,

$$\Delta S(u,v)|_{polar} = \sum_{i=0}^{5} \sum_{j=0}^{5} b_{i,5}(u) b_{j,5}(v) \Delta P_{i,j}$$
(6.4)

With  $\Delta P_{i,j}$  takes value of  $\Delta P_0$ ,  $\Delta P_1$ , and  $\Delta P_2$ .  $\Delta P_{i,j} = \Delta P_0$  if  $i \in [0,2]$ ;  $\Delta P_{i,j} = \Delta P_1$  if  $i \in [3,5]$  and  $j \in [0,2]$ ;  $\Delta P_{i,j} = \Delta P_2$  if  $i \in [3,5]$  and  $j \in [3,5]$ ;  $\Delta P_0$ ,  $\Delta P_1$ , and  $\Delta P_2$  are the difference vectors at three corners of a Polar face (Fig. 6.5).

With offsetting Bezier crust on a Polar face  $\Delta S(u, v)|_{polar}$  defined, the interpolating parametric surface on a Polar face  $\overline{S}(u, v)|_{polar}$  can be expressed as follows:



Figure 6.7: (a) difference vectors on Polar face, (b) vertex splitting of difference vectors in (a), (c) bi-quintic offsetting Bezier Crust on Polar face

$$\overline{S}(u,v)|_{polar} = S(u,v)|_{polar} + \Delta S(u,v)|_{polar}$$
(6.5)

Given a PCC mesh, with equation (6.4) defining Bezier Crust on Polar face and equation (6.2) defining Bezier Crust on Quad face, one can construct a piecewise offsetting bi-quintic Bezier Crust on PCCSS to interpolate difference vectors between PCC mesh control points and their PCCSS limit points. By parametrically adding Bezier Crust to PCCS limit surface (equation (6.3) and (6.5)), one can obtain an interpolating limit surface with properties as follows,

- $C^2$  continuous everywhere, except at extraordinary points, where it is  $G^2$  on Polar extraordinary points,  $C^1$  at CCS extraordinary points.
- It interpolates control points in PCC mesh, and interpolates the normals and curvature at their corresponding data points at PCCS limit surface.

Above we introduce our new interpolation scheme for PCCSS. Fig. 6.8 shows an airplane with Polar part on plane head, the new interpolating surface of plane head (Fig. 6.8(c)) is smooth and without ripples. Since most Polar subdivision scheme uses vertex splitting to match quad and triangular faces and Bezier Crust is subdivision rules independent, the above methods can be applied to these scheme.



Figure 6.8: (a) A PCC mesh of an airplane (b) enlarged plane head with Polar configuration, (c) interpolating limit surface with our new scheme on (b)

#### 6.4 Implementation and analysis

In previous section, we introduced the concept of our new interpolation surface for PCCSS. This new interpolation surface is generated by parametrically adding a piecewise offsetting Bezier Crust to PCCS limit surface. Given a PCC mesh M, for each (u,v) of quad face or converted Polar face (vertex splitting), the interpolating algorithm is implemented as follows:

- (1) Compute S(u, v) limit point for each (u, v)
- (2) Compute difference vectors on all control points in M
- (3) Derive Bezier Crust on converted difference vectors
- (4) Computer  $\Delta S(u, v)$  on derived Bezier Crust
- (5) Obtain  $\overline{S}(u, v)$  by adding S(u, v) and  $\Delta S(u, v)$

With the special selection of Bezier Crust, the computation of each limit point on interpolation surface only increase constant time to the computation of each PCCSS limit point. So the running cost of adding a Bezier Crust in our new scheme is not expensive. Fig. 6.9(a) shows a typical Polar mesh. Fig. 6.9(b) shows the interpolating surface generated by our new interpolation scheme. Fig. 6.9(c) shows the limit surface of Bezier Crust when it is shown standalone. We see that when Bezier Crust is drawn standalone, the limit surface is actually only G0 continuous. This is consistent with the duplications of control points at 4 corners. However, when we show the Bezier crust (Fig. 6.9(c)) parametrically added to the PCCSS limit surface of a flat mesh converted by projection of this Polar mesh onto (x,y) plane. We see that although the underlying PCCS limit surface of projected flat mesh has zero Gaussian curvature everywhere, the parametrically added interpolating surface is smooth(Fig. 6.9(d)).



Figure 6.9: (a) Polar mesh (b) new interpolating surface (c) Bezier Crust shown alone (enlarged) (d) Bezier Crust is shown on projected flat PCCSS limit surface (enlarged)

As stated in the last section, the new interpolating surface on PCC mesh has two properties. Here we provide a proof.

**Property 1**:  $C^2$  continuous everywhere, except at extraordinary points, where it is  $G^2$  on Polar extraordinary points,  $C^1$  at CCS extraordinary points.

**Proof**: Interpolating surface continuity on quad face part is already discussed in [66]. Here we show the continuity on Polar face part. On a Polar face, with vertex splitting, the Bezier Crust obtained has vanished 1st order and 2nd order derivatives at 4 corners, across the boundaries and along the boundary connecting split vertices, and the Bezier Crust is also  $C^2$  along the boundries and inside Bezier Crust limit surface. Since PCCS limit surface is  $G^2$  on Polar extraordinary points and  $C^2$  on Polar faces, with analysis of 1st order and 2nd order derivative on equation (6.5), we can conclude that the interpolating surface maintains the continuity of its underlying PCCS limit surface. QED

**Property 2**: It interpolates control points in PCC mesh, and interpolates the normal and curvature of their corresponding limit points at PCCS limit surface.

**Proof**: Since the 1st order and 2nd order derivatives of Bezier Crust vanishes at 4 corners, by analyzing equation (6.5), we can conclude that 1st order and 2nd order derivatives of PCCS limit surface on the corner control points of each face are the same as those of our new interpolating surface. QED

Implementation results (Fig. 6.1, 6.8, 6.9, 6.10) show that our new interpolating scheme can generate high quality Polar limit surface. Our new interpolating scheme on PCC mesh is heuristic, it can efficiently compute each limit point of interpolating surface without drastically adding computation time.

#### 6.5 Summary

In this work, we introduce a new heuristic interpolation scheme on Polar surfaces, especially on PCCSSs. We show that, by vertex splitting, we can treat a Polar face as a quad face, such that the bi-quintic offsetting Bezier Crust can be applied to the Polar faces as well. The generated interpolating surface maintains the continuity of underlying PCCS limit surface, i.e.  $G^2$  at Polar extraordinary points,  $C^1$  at CCS extraordinary points, and  $C^2$  everywhere else.

Implementation results show that our new scheme can generate high quality images appropriate for engineering and computer graphics usage. While Polar surface is studied to solve high valence artifact inherent in CCS, less work is developed on interpolation schemes on Polar surfaces. With our subdivision independent interpolation scheme of Bezier Crust, we can efficiently generate a smooth interpolating surface on a Polar mesh.



Figure 6.10: Three examples of interpolating on Polar mesh, (a) Polar mesh, (b) Polar limit surface with PCCSS, (c) mesh with limit surface together, approximating, (d) new interpolating limit surface with Bezier Crust on PCCSS (e) mesh shown with interpolating surface.

# Chapter 7

# Thin Shell Modeling on Catmull-Clark Subdivision Surface

In rapid prototyping, a hollowed prototype is preferred and significantly reduces the building time and material consumption in contrast to a solid model. Most rapid prototyping obtains solid thin shell by gradually adding or solidifying materials layer by layer. This is a non-trivial problem to offset a solid which involves finding all selfintersections and filling gaps after raw offsetting. While Catmull-Clark subdivision (CCS) surfaces are widely used in solid modeling, the hollow solid/thin shell problems are not well addressed yet. In this work, we explore earlier methods of obtaining thin shell CCS solid and present a new thin solid approach. With this new scheme, one can efficiently avoid creases and handle gaps. The new scheme is heuristic, but inner surface is parametric, so computation of the inner surface is simplified. And with offsetting Bezier crust applied, the inner surface maintains the mesh structure and continuity of the outer surface. The obtained thin shell solid is  $C^2$  continuous everywhere, except at extraordinary points, where it is  $C^1$  continuous.

#### 7.1 Introduction

In 3D modeling, building a hollowed prototype instead of a solid model is required to reduce the building time and material consumption. When we use CCS to generate a hollowed object, the intuitive way is to construct CCS meshes for both the outer and the inner surfaces. However, it is not effective and many issues arise during construction of the inner surface, e.g., surface collision, self-intersections. It is not an easy task to design a CCS control mesh to generate a thin-shell hollowed 3D object.

In CAD/CAM, rapid prototyping (RP) builds a part layer by layer faster than traditional prototyping methods. The RP process involves slicing the CAD model perpendicular to the building direction sequentially and gradually adding or solidifying materials layer by layer. RP applications are used in the making of molds, manufacturing parts, and most recently 3D-printing. In RP, when each layer is solid, it not only consumes more materials, but also is time consuming. To reduce the building time and material consumption, the method of hollowing out the 3D solids is applied to reduce the cross-sectional area to be traced. Some spatial enumerations have been used to obtain hollow solids, such as a sub-boundary octree [39] located inside the original solid, voxel model [10] featuring one-dimensional Boolean operations between the ray representation and voxel elements. The main problem with enumeration techniques is the staircase effect, which make offsetting surface not attractive. Another method developed is constructive solid geometry (CSG) [41]. CSG works by subtracting the original solid from its offset counterpart. This method is known to perform well on simple primitives, such as cylinder, spheres and boxes. However it is difficult to offset a free-form surface like CCSS. 2D curve offsetting method [23] slices the original solid sequentially and obtains internal cross-sectional curves by offsetting external cross-sectional curves of each slice. This method is simple and easy to implement, but it is hard to achieve uniform wall thickness. A further work [51]

achieves more uniform wall thickness and proposed a new algorithm that computes internal contour without computing the offset model. There are also some surface offsetting methods. Non-uniform offsetting method [35] employs a vertex offsetting approach which is based on an averaged surface normal method. Main issue with this method is the existing of many self-intersections and invalid triangles. Computing the correct offset model of a STL model is a non-trivial task [46].

Several isocurve-based methods are developed to offset free-form surfaces. These methods are based on 3D curve offsetting [52]. In methods of tool-path generation [19] and adaptive isocurve-based rendering [20], a set of parallel curves called isodistance curves are obtained by trimming iso-parametrics situated at fixed distances from the original curves. An iterative method of interference-free 3D offset contours [30] is proposed to offset parametric surfaces.

Given a free-form parametric surface like CCSS, if we apply above methods, although 3D offset surface generated will maintain uniform wall thickness, but the surface quality will not be satisfactory. None of above can generate an  $C^2$  offset surface. It will be acceptable if there is no surface quality requirement for the offset surface. However, when the model is used to make mold, it is generally required the 3D offset surface is also smooth.



Figure 7.1: an example of hollowed solid with our new offsetting scheme: a) CCS mesh, b) CCS limit surface, c) our offsetting surface, d) cross-section view, e) enlarged detail from cross-section.

In this work, we present an  $C^2$  offsetting scheme on CCS surfaces. With this new scheme, one can generate hollow 3D solids efficiently with one layer of CCS control mesh and maintain the curvature continuity of CCS scheme. Due to the parametric properties of CCS, in our new scheme, we use a new surface offsetting approach, which offsets the limit surface directly by adding a thin layer of bi-quintic Bezier surface. Fig. 7.1 shows a hollowed solid after applying our new scheme, from Fig. 7.1(c) and (e) we see that the offsetting surface is smooth and the wall thickness is visually uniform.

#### 7.2 Earlier works

As stated in earlier chapters, a Catmull-Clark subdivision (CCS) surface is the limit surface of a sequence of subdivision steps performed on a given control mesh. At each step, new vertices are added and old vertices are updated. The valence of a vertex is the number of edges meeting at the vertex. A vertex with valence four is called a regular vertex, otherwise an extraordinary vertex. A mesh face is regular if all vertices are regular, otherwise, it is called extraordinary face. CCS vertices are classified into three categories: vertex points, edge points, and face points. A popular way to index the control vertices is shown on the left side of Fig. 7.2 for a regular face and the right side for an extraordinary face, where V is a vertex point,  $E_i$ s are edge points,  $F_i$ s are face points, and  $I_{i,j}$ s are inner ring control vertices. New vertices within each subdivision step are generated as follows:

$$V' = \alpha_N V + \beta_N \sum_{i=1}^N E_i / N + \gamma_N \sum_{i=1}^N F_i / N$$
  

$$E'_i = \frac{3}{8} (V + E_i) + \frac{1}{16} (E_{i+1} + E_{i-1} + F_i + F_{i-1})$$
  

$$F'_i = \frac{1}{4} (V + E_i + E_{i+1} + F_i)$$
(7.1)

where N is the valence of vertex V, with  $\alpha_N = 1 - \frac{7}{4N}$ ,  $\beta_N = \frac{3}{2N}$ , and  $\gamma_N = \frac{1}{4N}$ .



Figure 7.2: mesh structure of CCS, (a) regular face, (b) extraordinary face

The CCS limit surface obtained by performing (7.1) sequentially can be parameterized [36]. We define S(u,v) as the CCS limit surface with parametric values (u,v),  $u, v \in [0, 1]$ , such that the CCS limit/data point S(0,0) of vertex point V is

$$S(0,0) = (5V + (12\beta_N + 8\gamma_N)E + (2\beta_N + 8\gamma_N)F)/(5 + 14\beta_N + 16\gamma_N)$$
(7.2)  
where  $E = \sum_{i=1}^{N} E_i/N, F = \sum_{i=1}^{N} F_i/N.$ 

The unit normal  $\mathbf{n}_{S(u,v)}$  on each data point S(u,v) of CCS limit surface can be explicitly calculated with its first order partial derivatives  $\frac{\partial S(u,v)}{\partial u}$  and  $\frac{\partial S(u,v)}{\partial v}$ ,

$$\boldsymbol{n}_{S(u,v)} = \frac{\partial S(u,v)}{\partial u} \times \frac{\partial S(u,v)}{\partial v}$$
(7.3)

Given an offset thickness d, the simplest solution of constructing an offset surface  $\overline{S}(u, v)$  is to subtract from each data point S(u, v) a vector of size d along the direction of unit normal,

$$\overline{S}_{u,v} = S(u,v) - d \cdot \boldsymbol{n}_{S(u,v)}$$
(7.4)



Figure 7.3: (a) a CCS control mesh and limit face; (b) the two neighboring offsetting data points crease; (c) solve creasing issue by decreasing the number of data points; (d) solve creasing issue by a shorter |d|.

This scheme works fine when the limit surface is concave or flat, but it will possibly generate a creased surface when the limit surface is convex. The creases (self intersections) (Fig. 7.3 (b)) are caused by intersection of  $n_{S(u,v)}$  s of neighboring data points along the surface, which can only be reduced by decreasing the number of data points on each face (Fig. 7.3 (c)) or shortening the offset thickness d (Fig. 7.3 (d)). However, since less number of data points means more roughness of limit surface and the reduction of thickness d is usually unwanted, the creases cannot be effectively removed.

In [66], a Bezier crust scheme is applied to CCS limit surface to obtain a parametric interpolating surface, the bi-quintic Bezier crusts added will maintain the curvature continuity of underlying CCS parametric surfaces. The scheme of Bezier crust works on difference vectors between control points and their corresponding data points.

In the Bezier crust scheme, given a quad control mesh M, the CCS scheme generates a limit surface that approximates the control mesh. The limit surface of each face f of M (regular or extraordinary) can be represented in parametric form S(u,v). For each f,  $\Delta P_0$ ,  $\Delta P_1$ ,  $\Delta P_2$  and  $\Delta P_3$  are defined as the difference vectors between the corner control points and their corresponding CCS data points, respectively. In



Figure 7.4: difference vectors of  $\Delta P_0$ ,  $\Delta P_1$ ,  $\Delta P_2$  and  $\Delta P_3$  on (a) a regular face and (b) an extraordinary face.

order to interpolate the control points, a bi-quintic Bezier crust  $\Delta p(u, v)$  is defined as follows,

$$\Delta p(u,v) = \sum_{i=0}^{5} \sum_{j=0}^{5} b_{i,5}(u) b_{j,5}(v) \Delta P_{i,j}$$
(7.5)

With  $\Delta P_{i,j}$  takes value of  $\Delta P_0$ ,  $\Delta P_1$ ,  $\Delta P_2$  and  $\Delta P_3$ .  $\Delta P_{i,j} = \Delta P_0$  if  $i \in [0, 2]$  and  $j \in [0, 2]$ ;  $\Delta P_{i,j} = \Delta P_1$  if  $i \in [0, 2]$  and  $j \in [3, 5]$ ;  $\Delta P_{i,j} = \Delta P_2$  if  $i \in [3, 5]$  and  $j \in [0, 2]$ ;  $\Delta P_{i,j} = \Delta P_3$  if  $i \in [3, 5]$  and  $j \in [3, 5]$ .  $\Delta P_0$ ,  $\Delta P_1$ ,  $\Delta P_2$  and  $\Delta P_3$  are the difference vectors at four corners of a CCS face (Fig. 7.4).

An interpolating surface constructed by appending a bi-quintic Bezier crust on CCSS ( shown in equation (7.5) ) has the following properties:

- It interpolates exactly the corner control points em It maintains the CCSS 1st and 2nd order derivatives at the corner control points
- It is  $C^2$  continuous everywhere, except at extraordinary points, where it is  $C^1$  continuous



Figure 7.5: two examples of constructed interpolating surfaces with Bezier Crust on CCSS

Fig. 7.5 shows that the interpolating surface is smooth and appropriate for most engineering/CAD usage. This inspired our interest to apply Bezier crust on CCSS to obtain a hollowed solid, such that a smooth offsetting surface can be constructed similar to CCSS, while maintains the curvature continuity of original CCS surfaces.

# 7.3 offsetting surface on CCSS with Bezier crust

In previous section, we review the interpolating scheme with Bezier crust on CCSS. In this section, we show how this scheme can be applied to construct a smooth offsetting surface on CCS surfaces. Given a CCS control mesh M, on an arbitrary face f, we define f with a set of 2N+8 control points  $V, E_1, E_N, F_1, F_N, I_1, I_7, i = 1, ..., N$ (shown in Fig. 7.2). With parametric form S(u,v) of CCS, we define the data points at four corners of CCS limit surface as  $p_0 = S(0,0), p_1 = S(1,0), p_2 = S(1,1)$ , and  $p_3 = S(0,1)$ , and their unit normal as  $\mathbf{n}_i$  (Fig. 7.6).

If we set the desired thin-shell thickness as d, then we can define a set of difference vectors of  $\Delta p_i$  on their corresponding data point  $p_i$ , then  $(p_i - \Delta p_i)$  will be the desired corner data points on the offsetting surface. When we apply Bezier crust on these difference vectors at four corners of a CCS face, we can obtain a parametric offsetting surface having uniform distance of d at all corners of each CCS face with its corresponding CCS corner points, while CCS continuity will be kept after offsetting. Our scheme select  $\Delta p_i$  with



$$\Delta p_i = d \cdot \boldsymbol{n}_i \tag{7.6}$$

Figure 7.6: CCS control mesh and its corner data points and normals: (a) regular face, (b) extraordinary face

The computation of the four corner points on the new offsetting surface is consistent with the method used in 3D surface offsetting presented by [35]. With equation (7.6), we now define the offsetting Bezier crust  $\Delta S(u, v)$  on difference vectors of  $\Delta p_i$ (i = 1, 4), with expression of

$$\Delta S(u,v) = \sum_{i=0}^{5} \sum_{j=0}^{5} b_{i,5}(u) b_{j,5}(v) \Delta P_{i,j}$$
(7.7)

With  $\Delta P_{i,j}$  takes value of  $\Delta p_0$ ,  $\Delta p_1, \Delta p_2$  and  $\Delta p_3$ .  $\Delta P_{i,j} = \Delta p_0$  if  $i \in [0, 2]$  and  $j \in [0, 2]$ ;  $\Delta P_{i,j} = \Delta p_3$  if  $i \in [0, 2]$  and  $j \in [3, 5]$ ;  $\Delta P_{i,j} = \Delta p_1$  if  $i \in [3, 5]$  and  $j \in [0, 2]$ ;  $\Delta P_{i,j} = \Delta p_2$  if  $i \in [3, 5]$  and  $j \in [3, 5]$ .  $\Delta p_0, \Delta p_1, \Delta p_2$  and  $\Delta p_3$  are the offsetting difference vectors at four corners of a CCS face, as shown in Fig. 7.7.

With offsetting Bezier crust  $\Delta S(u, v)$  defined, the offsetting parametric surface  $\overline{S}(u, v)$  can be expressed as follows:

$$\overline{S}(u,v) = S(u,v) - \Delta S(u,v) \tag{7.8}$$

In the above, the construction of an offsetting parametric surface for a CCSS is shown. Next, we will analyze the behavior of this new scheme and show some properties of this offsetting surface.



Figure 7.7: offsetting surface (blue) obtained after subtracting offsetting Bezier crust from the CCS limit surface: (a) regular face (b) extraordinary face.

# 7.4 behavior of the new offsetting surface and discussion

In this section, we discuss the behavior of our new offsetting surface.

**Theorem 7.1.** the new offsetting parametric surface  $\overline{S}(u, v)$  is  $C^2$  continuous everywhere, except at extraordinary points, where it is  $C^1$  continuous. *Proof.* Our new offsetting parametric surface is constructed by subtracting an offsetting Bezier crust from the CCS limit surface. CCS limit surface is  $C^2$  continuous everywhere, except at extraordinary points, where it is  $C^1$  continuous. And by Wang and Cheng [66], the bi-quintic Bezier crust is  $C^2$  continuous everywhere except at corner control points and across the face boundaries, where its derivatives vanish up to the 2nd order. Computing 1st and 2nd order derivatives on equation (7.8), it will show that  $\overline{S}(u, v)$  will maintain the curvature continuity of CCS limit surface S(u,v).

One of the benefits to subtract an offsetting Bezier crust from CCS limit surface is that the offsetting Bezier crust has the same behavior to handle both regular face and extraordinary face. This derives from the fact that, given the required thickness of d, the offsetting Bezier crust works on difference vectors of size d and with the direction of the unit normal of corner data points. Given a regular or extraordinary face of degree N, computation of the offsetting Bezier crust is independent of N.

With Eigen-decomposition, each individual data point on the CCS limit surface can be computed in O(1). Calculating an arbitrary point on offsetting Bezier crust by equation (7.7) is also O(1). Such that computation of each individual limit point on offsetting surface is O(1). It is apparently more efficient in comparison with constructing offsetting surface layer by layer by slicing.

Given a CCS face, when it is flat, the difference vectors on all four corners are equal, with equation (7.8), each limit point on obtained offsetting face has exactly the same geodesic distance d to original limit surface. When the face is concave or convex, then a limit point of  $\overline{S}(u, v)$  on parametric surface is the sum of the limit point S(u,v) and affine combination of four difference vectors  $\Delta p_i = d \cdot (\mathbf{n}_i), i = 1, 2, 3, 4$ . We derive that



Figure 7.8: Three examples: (a) CCS control mesh; (b) new offsetting surface (without the CCS surface); (c) cross-sectional view - yellow is outer CCS surface, gray is the offsetting surface.

$$|\Delta S(u,v)| \le |d \cdot \boldsymbol{n}_{S(u,v)}| \tag{7.9}$$

where  $|\cdot|$  is the size of the enclosed vector. With (7.9), we can further derive that

$$||\overline{S}(u,v) - S(u',v')|| = d - \epsilon,$$
(7.10)

where  $|| \cdot ||$  represents the shortest distance between the offsetting surface  $\overline{S}$  and the original CCS surface S.  $\epsilon$  is the maximum error. Further, we can also show that our new offsetting surface is enclosed in the original CCS surface and the conventional offsetting surface defined in (7.6). Since  $\Delta S(u, v)$  is affine combination of difference vectors at 4 corners. To reduce , we can perform more CCS subdivisions on original CCS control mesh. If we define  $M^n$  as the CCS control mesh after nth subdivision, we can derive that

$$|\Delta S(u,v)| \approx |d \cdot \boldsymbol{n}_{S(u,v)}|, \text{ when } n \to \infty$$
(7.11)

which is exactly the representation of the conventional way of surface offsetting shown in equation (7.4) with  $\epsilon \approx 0$ . Note the surface generated with equation (7.4) is generally not a smooth surface. Fig. 7.9 shows how the subdivisions impact the surface quality of the offsetting surface. In the center where curvature is high, the offsetting surface shows increasing creases after three times subdivision, whereas original one is smooth. This is consistent with our analysis shown in Fig. 7.3.

Since in general cases  $\epsilon$  is small and we do want to avoid the offsetting surface obtained from equation (7.4) (many creases and self-intersection when outer surface is convex), so it will not be necessary to perform further subdivision if  $\epsilon$  is within the tolerance.

Our scheme is based on the assumption of regular CCS mesh that all corner CCS data points have non-zero unit normal, we will also include discussion of scenario when unit normal does not exist (control mesh collapses). Prerequisite of equation (7.8) is that on each corner data point of the CCS limit surface its unit normal exists. In most cases, it is true, however there are some special cases where unit normal does not exist (1st order derivative along one parametric direction is 0, due to control vertices coinciding). In such rare cases, we propose to add the unit normal to such



Figure 7.9: Comparison of offsetting surface after 3 recursive subdivisions on CCS mesh: (a) CCS control mesh, (b) generated offsetting surface, (c) enlarged central part of the offsetting surface.

corner data point with the average of the unit normals on its neighboring data point. The algorithm is as follows,

- (a) If all CCS corner data points have unit normal then go to (c), otherwise pick up a data point where unit normal does not exist, go to (b).
- (b) For this data point, we put average of its neighboring unit normal as its unit normal. Go back to (a).
- (c) End of the algorithm, start to construct the offsetting surfaces.

Above algorithm is heuristic, since it defines the unit normal on some collapsed control vertex as the average of its neighboring unit normals when its unit normal does not exist. Further research needs to be made to handle such special cases. Implementation results in Fig. 7.1 and Fig. 7.8 show that a smooth thin offsetting surface can be generated by applying our new scheme. The offsetting surface keeps a quasi-uniform thickness with CCS limit surfaces, which formed a nice hollowed 3D solid appropriate for common CAD usage.

#### 7.5 Summary

In this work, we introduce a new thin shell hollowing model on 3D objects represented by Catmull-Clark subdivision surfaces. Our new method for inward offsetting works by subtracting a thin layer of bi-quintic Bezier crust from the original CCS surface.

The new offsetting surface generated is visually smooth and has the same continuity as the original CCS limit surface, i.e.  $C^2$  continuous everywhere, except at extraordinary points, where it is  $C^1$  continuous. The properties of new offsetting surface are also discussed in this work.

Implementation results show that the offsetting surface generated is free from creases, and filling the gaps is trivial due to the fact that the offsetting surface is the parametric sum of the original CCS surface and a Bezier crust on difference vectors of size d on each face. Since a bi-quintic Bezier crust does not change the curvature at a corner data point of the CCS limit surface, one would not get gaps at connections of offsetting faces commonly found in earlier methods. Our next step is to explore current solutions of removing unwanted loops, and apply them to our new scheme to generate a smooth 3D offsetting surface without creases, loops and self-intersections.

### Chapter 8

### **One-step Bicubic Interpolation**

In this work, a new interpolation scheme for Catmull-Clark subdivision (CCS) surfaces is introduced. The construction process is based on two techniques: surface offsetting and mesh decomposition. The surface offsetting technique ensures the shape of the data set is faithfully resembled, so the method has the power of a global method; the mesh decomposition technique enables us to solve the problem using a one-step, local approach, instead of solving a global linear system using an iterative approach. The decomposition process of an offsetting mesh preserves the number of extraordinary points in the CCS mesh. Therefore, the interpolating surface preserves the continuity of a CCS surface. Furthermore, with heuristic selection of offsetting mesh, the computed interpolating surface can also maintain the same normal and curvature at interpolating points as CCS surface. Test results show that interpolating surfaces can be efficiently generated by the new method for large data sets and the generated interpolating surfaces have very high surface quality. Hence, the new scheme is especially suitable for applications in reverse engineering and 3D printing.

#### 8.1 Introduction

Freeform surfaces are widely used in computer graphics. Traditionally NURBS surfaces handle freeform surfaces in CAD/CAM [21]. A NURBS surface has a rigid rectangular control grid. Therefore surfaces are represented by collections of trimmed patches, continuity across patch boundaries have to be manually enforced.



Figure 8.1: Frog example by our new interpolation scheme. Left is the interpolating surface. Right two rows are enlarged views of frog eye and back pattern, from left to right: a) original surface (before interpolation); b) control mesh inside the blue box in a) enlarged; c) our interpolating surface; d) data mesh inside the blue box in c) enlarged.

As we stated earlier, subdivision surfaces became popular in surface representation due to the facts that they are simpler than traditional spline methods and are able to handle arbitrary topology. Catmull-Clark subdivision scheme [6] and Loop scheme [43] are the most widely used schemes in quad and triangular mesh structures, respectively. In particular, Catmull-Clark subdivision surfaces (CCSS) have become a standard modeling/representation scheme in computer animation and gaming.

The CCS scheme is an approximating scheme, i.e., a CCSS smoothly approximates, but does not interpolate the given control mesh. However, construction of smooth interpolating surfaces is important in many applications, including computer aided design, statistical data modeling and face recognition. This means, given a "data mesh", one needs to construct a control mesh so that the CCS limit surface of this control mesh would interpolate the given data mesh.

This interpolation problem can be solved directly or iteratively. A direct method such as the earlier work of Halstead [29] can be used if the data mesh is relatively small or the corresponding linear system is non-singular. For data mesh with hundreds of data points, or the corresponding linear system is singular, a progressive subdivision scheme [8] [9] can be used. This method iteratively generates a new control mesh by adding to the current control mesh the difference between the current control mesh and its corresponding data points on the CCS limit surface. The resulting linear system is positive definite and improves the convergence speed of the CCS control mesh generation process.

Besides the convergence speed issue, the interpolating surface obtained sometimes could possess excessive undulations [29]. The Fairing techniques proposed in [45] [70] smooth an interpolating surface by including more constraints but that also increases the size of the control mesh. Some alternative methods [37] [71] improve shapes by choosing good initial control mesh or adding more control points to control the shape locally.

A recent iterative approach has the advantages of both a local method and a global method [38], i.e., it can handle meshes with thousands of data points and complex topology while capable of faithfully reproducing the shape of any given mesh. Besides, this approach provides a way to expand a mesh into an infinite series of meshes (surfaces) which allows classical applications such as texture mapping and morphing to be solved differently.

But the above iterative interpolating methods have an efficiency problem and computation errors when the number of data points is millions. We will present a solution to this problem in this work, i.e., we will present a precise interpolating scheme for CCS that can efficiently handle data sets with millions of data points. The new method can generate a bicubic surface to interpolate a set of millions of data points in just one step. Furthermore, the computed interpolating surface has the same local property as a CCS surface, i.e., changing a set of data points will only change the shape of the interpolating surface locally around these data points.

The construction process is based on two techniques: mesh decomposition and surface offsetting. The mesh decomposition technique enables us to solve the problem using a one-step, local approach; the surface offsetting technique ensures the shape of the data set is faithfully resembled. Hence the method has the advantages of both a local method and a global method, and yet it does not require an iterative approach. Test results show that the new method produce very good results for large data sets. Fig. 8.1 shows a frog with 1,200,002 interpolation points, running time of our new scheme to compute all data points on the limit surface is 49.912 seconds, only slightly higher than that of the CCS scheme (38.530 seconds). From the enlarged views of frog eye and back pattern, we see that even though the frog has million of control points, without subdivision it is only  $C^0$  continuous after zoom-in, while after our interpolation, the limit surface is  $C^2$  everywhere except at extraordinary points where it is  $C^1$  continuous.

In contrast to our bi-quintic Bezier Crust interpolating scheme [66], this new interpolating scheme is bi-cubic only and especially designed for CCSS.

#### 8.2 Related Works

#### CCS and mesh structure

CCS can convert a mesh of arbitrary topology into a CCS mesh with only quadrilateral faces and each face has at most one extraordinary vertex in at most two recursive subdivision steps [6]. The CCS scheme divides the vertices of a given/converted CCS
mesh into three categories: vertex points, edge points, and face points.



Figure 8.2: (a): CCS mesh;(b),(c),(d): CCS subdivision masks for new face, edge and vertex points.

A popular way to index the vertices of a CCS mesh face is shown in Fig. 8.2(a), where V is a vertex point,  $E_i$ 's are edge points,  $F_i$ 's are face points and  $I_{i,j}$ 's are inner ring control vertices. New vertices within each subdivision step are generated as follows:

$$V' = \alpha_N V + \beta_N \sum_{i=1}^N E_i / N + \gamma_N \sum_{i=1}^N F_i / N$$
  

$$E'_i = \frac{3}{8} (V + E_i) + \frac{1}{16} (E_{i+1} + E_{i-1} + F_i + F_{i-1})$$
  

$$F'_i = \frac{1}{4} (V + E_i + E_{i+1} + F_i)$$
(8.1)

where N is the valence of vertex V,  $\alpha_N = 1 - \frac{7}{4N}$ ,  $beta_N = \frac{3}{2N}$ , and  $gamma_N = \frac{1}{4N}$ . These subdivision rules (Fig. 8.2 (b),(c),(d)) work for the inner ring control vertices as well since these control vertices and the subsequently generated new control vertices are also vertex, edge or face points. For control vertices generated after the  $n^{th}$ subdivision, we have

$$\boldsymbol{C}_n = \boldsymbol{A}^n \boldsymbol{C}_0, \qquad \bar{\boldsymbol{C}}_n = \bar{\boldsymbol{A}} \boldsymbol{A}^{n-1} \boldsymbol{C}_0, \quad n \ge 1,$$
(8.2)

where  $C_n$  is a vector of 2N + 8 control vertices of  $f_i$  after  $n^{th}$  subdivision,  $\bar{C}_n$  is a vector of 2N + 17 control vertices after one subdivision on  $C_{n-1}$ , N is the valence of V, A and  $\bar{A}$  are the corresponding extended subdivision matrices of size  $(2N + 8) \times$ (2N + 8) and  $(2N + 17) \times (2N + 8)$ , respectively, and  $C_0$  is the vector of the original (2N + 8) control vertices of  $f_i$ .

#### CCS interpolation schemes

Note that a CCS limit surface does not interpolate the vertices of its control mesh, but approximate them. To interpolate the vertices of a given data mesh with a CCSS, it is traditionally achieved by solving a global linear system [29] [3] [64] [8] [9],

$$\widetilde{A}x = b \tag{8.3}$$

where  $\widetilde{A}$  is a square matrix determined by subdivision rules and mesh topology, x is a column vector of control points to be determined, b is a column vector of data points in the given data mesh. If  $\widetilde{A}$  is small and nonsingular, we can obtain the control mesh by calculating its inverse  $\widetilde{A}^{-1}$  directly. However, a direct method will not work or not work well if  $\widetilde{A}$  is singular or large. In such a case, an iterative method needs to be applied. Traditionally, stationary iterative methods like Jacobi, Gauss-Seidel, Successive Over-relaxation, Krylov subspace or parallel direct sparse solver can be used to solve a large linear system. The issue with these methods is the convergence rate - they are slow when the data set is large. There are faster iterative methods to solve large scale data sets [3] [64], however since equation (8.3) is a global system, convergence rate will still not be satisfactory when we are dealing with thousands of data points.

To improve iteration speed, a progressive subdivision scheme [8] [9] was developed. This method iteratively generates a new control mesh by adding to the current control mesh the difference between this control mesh and its corresponding data points on the CCS limit surface. The linear system developed is positive definite and can improve the convergence speed of CCS control mesh generation process which satisfies equation (8.3). Recently in [38] a fast iterative scheme is presented and it is shown that their iterative process converges to a unique solution.

Besides convergence speed, the interpolating surface obtained by solving equation (8.3) sometimes is unsatisfactory because of excessive undulations [29]. Halstead [29] notices that the undulations appear because they are not indicated by the shape of the original mesh. The fairing techniques proposed in [45] [70] smooth an interpolating surface by including more constraints but increasing the size of the control mesh.

An alternative local method is developed to accommodate local controls by dividing each face into  $2 \times 2$  subfaces through one subdivision [37] [71]. The resulting linear system is under-determined. If the control points in the divided control mesh are classified into interpolating control points (corresponding to interpolating data points) and non-interpolating control points (added control points), then the basic idea of this approach is simply to change all interpolating control points to interpolate the given data points. By equation (8.1), one can show that the divided control mesh can be computed directly. Unfortunately, such an approach would create big curvature variation at interpolating control points and, consequently, the resulting interpolating surface tends to have undesired undulations and ripples. To avoid such surface artifacts, these local schemes still use an iterative process to compute the control points of the interpolating surface and generally are required to choose a good initial control mesh. But due to a large degree of freedom in the linear system, it is not an easy task to generate a well-shaped interpolating surface with such local methods.

The above iterative methods focus on improving convergence speed of solving equation (8.3) or introducing additional constraints to handle surface artifact, they are all approximating, not exactly interpolating methods. Two problems remain unresolved,

- by solving a global linear system, the obtained interpolation control mesh depends on all vertices in the original data mesh, hence the scheme lacks local support.
- 2. convergence speed is not satisfactory when handle large data-sets.

It is natural to ask the following question:

"Is it possible to have a precise interpolating scheme other than approximating ones, without solving a global linear system, and not iterative either, while preserving the surface quality and local support features of CCS?"

Recently, a direct scheme is developed in [66]. This scheme generates an interpolating surface by attaching a bi-quintic Bezier crust to a CCS limit surface. While the quality of the resulting interpolating surface is similar to that of the CCS limit surface, we would like to explore the possibility of attaching lower degree polynomials instead of bi-quintic spline surfaces to a CCS limit surface.

Subsequently, we present a bicubic one-step CCS interpolation scheme by adding a local bicubic offsetting surface to the CCS limit surface of a given data mesh. Fig. 8.3 shows a hollowed cube example implemented both by our new interpolation scheme and the traditional scheme. We see that with the traditional scheme (top row), the converted control mesh will be away from the shape of the original mesh, so that its interpolation surface has undesired undulation, while the interpolating surface generated by our new scheme (bottom row) is similar to the original mesh and does not show undulation at all.



Figure 8.3: Top row: (a) converted control mesh from (c) by solving equation (3); (b) CCS limit surface of converted mesh; bottom row: (c): given data mesh; (d): limit surface of the new interpolation scheme;

## 8.3 One Step Bi-cubic Interpolation

As stated in previous sections, current CCS interpolation schemes using iterative approaches suffer from slow convergence when data set size is large. In particular, when the data size is millions, these iterative approaches could not even handle the problem. In this work, we introduce a global/local hybrid method. The idea is to divide the interpolating surface into two parts: *base surface* and *offsetting surface*.

Given a data mesh to be interpolated, the base surface is its CCS limit surface, and the offsetting surface is the surface interpolating the difference vectors between interpolating points and their CCS limit points. The new interpolating surface is obtained by combining these two surfaces parametrically (as shown in Fig. 8.4). With a proper selection of the offsetting surface, we can generate an interpolating surface directly without iteration. Hence the new scheme can handle extremely large data sets efficiently.



Figure 8.4: Left: given data mesh and its CCS limit surface (shaded in gray), middle: offsetting surface (shaded in gray) interpolating difference vectors; right: interpolating surface (shaded in gray)

The base surface of our new scheme is obtained through CCS, so we restrict the subdivision scheme for the offsetting surface to be CCS as well. Also, to make the generated interpolating surface  $C^1$  at extraordinary data points and  $C^2$  everywhere else, control mesh of the offsetting surface should have at least the same number of extraordinary points as that of the base surface. We assume the mesh structure of the offsetting surface corresponding to one base face to be  $k \times k$ ,  $k \ge 1$  (Fig. 8.5). In such a case we say *decomposition* of the offsetting mesh is  $k \times k$ .

We would like to have an offsetting surface that is computed directly and shape of the final interpolating surface to be as close to that of the base surface as possible. One way to achieve the second goal is to include additional constraints so that normals



Figure 8.5: (a):a base face; (b),(c),(d): corresponding offsetting mesh face with decompositon: 1x1, 2x2, 3x3, ...

of the interpolating surface at the interpolated data points are the same as those of the base surface.

If decomposition of the offsetting mesh is 1x1, then the offsetting mesh has the same structure as the base mesh. By equation (8.3), the offsetting mesh has to be computed globally and iteratively. So we can not achieve the first goal.

If decomposition of the offsetting mesh is 2x2, then computing the offsetting mesh process is just the partial interpolation scheme of adding one layer of control vertices to the original mesh [37] [71]. The partial interpolation scene can change the shape locally. But the way of directly changing interpolating control points will increase the curvature variations at the given data points and, consequently, will create undesired undulations and ripples at these points. To avoid such surface artifacts, these local schemes use an iterative approach to compute the control points of the interpolating surface. Besides, these schemes require the selection of a good initial mesh [37]. So we can not achieve the first goal either.

If decomposition of the offsetting mesh is 3x3 or larger, then by equation (8.1), one can set normal and curvature locally at the given data points. Here we let decomposition of the offsetting mesh be 3X3 so that the number of free variables can be minimized while interpolation of the difference vectors can still be computed locally. With this selection, the coefficient matrix  $\tilde{A}$  for the offsetting mesh has the following form,

$$\widetilde{A} = \begin{bmatrix} a_1 & & & \\ & a_2 & & \\ & & \dots & \\ & & & \dots & \\ & & & & a_m \end{bmatrix}$$
(8.4)

where for each k,  $a_k$  is a row vector for the  $k^{th}$  interpolation data point  $b_k$ , and all other entries in  $\widetilde{A}$  are zero. With equation (8.4), equation (8.3) can be reformed as

$$a_1 x_1 = b_1, \ a_2 x_2 = b_2, \ \dots, \ a_m x_m = b_m$$
(8.5)

With the mesh structure shown in Fig. 8.5(d) and equation (8.5), the global linear system of equation (8.3) is split into a group of local linear systems with each local linear system corresponding to an interpolation data point. With a setting like this, it is possible now to construct a direct interpolation scheme on a CCS data mesh. Since the interpolating control mesh of the offsetting surface divides each face of the base mesh into three equal parametric segments in u and v directions, respectively (Fig. 8.6(c)), we name it a 1/3 scheme.

With the selection of the 1/3 scheme on the offsetting mesh, our algorithm works as follows. Given a CCS data mesh M, we compute the difference vectors (between data points and their CCS limit points if we perform CCS on M) for all the data points. Then we construct offsetting mesh using the 1/3 scheme shown in Fig. 8.5(d). By solving equation (8.4), we obtain the offsetting mesh. If we parametrically add the limit surfaces of M and  $\Delta M$  (Fig. 8.6), we obtain a limit surface which interpolates M. This interpolation surface has the following properties.

- 1. has the same surface continuity as CCS limit surface
- 2. on interpolating data points, one can set normal and curvature locally without global iterations



Figure 8.6: 1/3 scheme on regular and extraordinary faces: (a) base mesh; (b) offsetting mesh together with the base mesh; (c) offsetting mesh.

The local linear systems of equation (8.4) is under-determined. We observe that if we set the same value on interpolating control points of offsetting mesh and their surrounding edge/face points, then from (8.4) we can derive that this value is exactly the difference vectors between the given data point and its CCS limit point. This is heuristic, the resulting offsetting mesh has vanished 1st order and 2nd order derivatives on interpolating control points. The generated interpolating surface will have the same curvature at given data points. By maintaining the curvature at given data points, with the given data mesh, the interpolating surface of our new scheme is similar to its CCS limit surface.

With our new scheme on offsetting mesh, we separate the global linear system into a set of local linear sub-systems, the computed offsetting mesh is less fluctuated, also the normals and curvature on given data points can be set locally. And with heuristic offsetting, the resulting interpolation surface can be computed directly without iterations and has the same local support on each data point, such that the quality of CCS limit surface on M can be preserved.

### 8.4 Mathematical Setup

In this section, we put our new scheme into rigorous mathematical setting, and show the properties of the resulting interpolating surface.

Our new interpolating surface is the sum of two parametric surfaces, one is the base surface, just CCS limit surface of given data mesh, another is the CCS offsetting surface, which interpolates the difference vectors between given data points and their corresponding points on its CCS limit surface.

The base surface, the CCS limit surface of given data mesh, can be parameterized, its parameterization is as follows,



Figure 8.7:  $\Omega$ -Partition of CCS

First we define the limit surface of a CCS face  $f_i$  as S(u, v), the three regular

bicubic B-Spline patches after the *n*-th CCS as  $S_{n,b}$ ,  $n \ge 1$ , b = 1, 2, 3. The  $\Omega$ partition (Fig. 8.7) is defined by:  $\Omega_{n,b}$ ,  $n \ge 1$ , b = 1, 2, 3, with

$$\Omega_{n,1} = \left(\frac{1}{2^n}, \frac{1}{2^{n-1}}\right] \times \left[0, \frac{1}{2^n}\right]$$
  

$$\Omega_{n,2} = \left(\frac{1}{2^n}, \frac{1}{2^{n-1}}\right] \times \left(\frac{1}{2^n}, \frac{1}{2^{n-1}}\right)$$
  

$$\Omega_{n,3} = \left[0, \frac{1}{2^n}\right] \times \left(\frac{1}{2^n}, \frac{1}{2^{n-1}}\right]$$

For any  $(u, v) \in [0, 1] \times [0, 1]$ ,  $(u, v) \neq (0, 0)$ , there is an  $\Omega_{n,b}$  containing (u, v). We can find the value of S(u, v) by mapping  $\Omega_{n,b}$  to the unit square  $[0, 1] \times [0, 1]$  and finding the corresponding  $(\overline{u}, \overline{v})$ .

After the mapping, we compute  $S_{n,b}$  at  $(\overline{u}, \overline{v})$ . The value of S(0,0) is the limit at (0,0).

In the above process, n and b can be computed by:

$$n(u, v) = \min\{\lceil \log_{\frac{1}{2}} u \rceil, \lceil \log_{\frac{1}{2}} v \rceil\} + 1$$
  
$$b(u, v) = k, if (u, v) \in \Omega_{n,k}, k = 1, 2, 3$$

The S(u, v) can be expressed as follows

$$S(u,v) = W^T(u,v)K^n D_b M \boldsymbol{C}_n^b$$
(8.6)

where W(u, v) is the 16-power-basis vector with  $[1, u, v, u^2, uv, v^2, u^3, u^2v, uv^2, v^3, u^3v, u^2v^2, uv^3, u^3v^2, u^2v^3, u^3v^3]$ , M is the B-spline coefficient matrix, K is a diagonal matrix with Diag(1, 2, 2, 4, 4, 4, 8, 8, 8, 8, 16, 16, 16, 32, 32, 64), and  $D_b$  is an upper triangular matrix depending on b only.

 $\boldsymbol{C}_{n}^{b}$  is the control points vector of  $S_{n,b}$ , with

$$\boldsymbol{C}_{n}^{b} = P_{b}\bar{A}A^{n-1}\boldsymbol{C}_{0} \tag{8.7}$$

where  $P_b$  is the selection matrix of b = 1, 2, 3.  $\overline{A}$ , A and  $C_0$  are extended subdivision matrices and original control vertices as shown in equation (8.2).

Equations (8.6) and (8.7) illustrate the parametric form of base surface, here we introduce the offsetting surfaces. The offsetting surfaces are defined as the CCS surface working on difference vectors between interpolation points and their corresponding data points with 9 times faces of original mesh (Fig. 8.6). The offsetting surfaces on  $f_i$  have 9 CCS sub-faces (Fig. 8.8 right), we define them as  $f_{i,1}$ ,  $f_{i,2}$ , ...,  $f_{i,9}$ , and they can be parameterized by using equations (8.6) and (8.7) with parametric values  $(u_1, v_1)$ ,  $(u_2, v_2)$ , ...,  $(u_9, v_9)$ .



Figure 8.8: left is base surface, right is the offsetting surfaces

We define  $\Delta S_m(u_m, v_m)$  as the parametric offsetting surface for  $f_{i,m}$ , m = 1, ..., 9.

$$\Delta S_m(u_m, v_m) = W^T(u_m, v_m) K^n D_b M C^b_{m,n}$$

$$n(u_m, v_m) = min\{ \lceil \log_{\frac{1}{2}} u_m \rceil, \lceil \log_{\frac{1}{2}} v_m \rceil \} + 1$$

$$b(u_m, v_m) = k, \ if \ (u_m, v_m) \in \Omega_{n,k}, \ k = 1, 2, 3$$
(8.8)

with

$$\boldsymbol{C}_{m,n}^b = P_b \bar{A} A^{n-1} \boldsymbol{C}_{m,0} \tag{8.9}$$

where  $C_{m,0}$  is the initial offsetting control mesh for  $f_{i,m}$ .

Such that, the offsetting surface  $\Delta S(u, v)$  on  $f_i$  is the union of all 9 sub-surfaces with the same  $\Omega - Partition$  as in equation (8.6), with

$$\Delta S(u,v) = \Delta S_1(u_1,v_1) \cup \Delta S_2(u_2,v_2) \cup \ldots \cup \Delta S_9(u_9,v_9)$$

Since the resulting limit surface is the sum of base surface and offsetting surfaces (illustrated in Fig. 8.8), we can define the resulting surface  $\bar{S}(u, v)$  as,

$$\overline{S}(u,v) = S(u,v) + \Delta S(u,v) \tag{8.10}$$

In order to define equation (8.10), a reparametrization need to be done on  $\Delta S_m(u_m, v_m)$ . The mapping from parametric values of sub-faces  $f_{i,m}$  to the  $f_i$  is defined by

$$(u_m, v_m) = (\phi(u), \phi(v)) , \text{ with}$$

$$\begin{cases}
1, & \text{if } 3u \in (2, 3] \text{ and } 3v \in [0, 1] \\
2, & \text{if } 3u \in (2, 3] \text{ and } 3v \in (1, 2] \\
3, & \text{if } 3u \in (2, 3] \text{ and } 3v \in (2, 3] \\
4, & \text{if } 3u \in (1, 2] \text{ and } 3v \in (2, 3] \\
5, & \text{if } 3u \in [0, 1] \text{ and } 3v \in (2, 3] \\
5, & \text{if } 3u \in (1, 2] \text{ and } 3v \in (2, 3] \\
6, & \text{if } 3u \in (1, 2] \text{ and } 3v \in (1, 2] \\
8, & \text{if } 3u \in (1, 2] \text{ and } 3v \in (1, 2] \\
8, & \text{if } 3u \in [0, 1] \text{ and } 3v \in (1, 2] \\
9, & \text{if } 3u \in [0, 1] \text{ and } 3v \in [0, 1] \\
and \\
\phi(t) = \begin{cases}
3t, & \text{if } 3t \in [0, 1] \\
3t - 1, & \text{if } 3t \in (1, 2]
\end{cases}$$

Since functions 
$$h$$
  $n$  and  $k$  in equation

Since functions b, n and k in equations (8.6) and (8.8) take different parametric values as input, to combine (8.6) and (8.8) into (8.10), we define  $\tilde{b}$ ,  $\tilde{n}$  as mapping from b and n, we get

$$\Delta S(u,v) = W^T(\phi(u), \phi(v)) K^{\tilde{n}} D_{\tilde{b}} M C^{\tilde{b}}_{m,\tilde{n}}$$
(8.11)

where

$$\widetilde{n}(u,v) = n(\phi(u),\phi(v))$$
$$\widetilde{b}(u,v) = b(\phi(u),\phi(v))$$

Given a data mesh M, in equation (8.10), the new interpolating surface  $\bar{S}(u, v)$  for an arbitrary face  $f_i$  is calculated by adding an offsetting surface  $\Delta S(u, v)$  to the base CCS surface S(u, v).  $\Delta S(u, v)$  is given in equation (8.11), while S(u, v) is represented in equation (8.6). For an arbitrary (u, v), one can calculate the limit point of  $\bar{S}(u, v)$ directly with equations (8.10), (8.6) and (8.11).



Figure 8.9: (a) base mesh (b) offsetting mesh

Since  $C_0$  in equation (8.7) is the original 2N + 8 control vertices of CCS on  $f_i$ , S(u, v) can be explicitly computed. The value of offsetting surface  $\Delta S(u, v)$  depends on the offsetting control meshes  $C_{m,0}$ 's defined in equation (8.9). Given a  $f_i$  with valence N, we define the control points in base mesh M as  $V_0, V_1, ..., V_{2N+7}$  (where  $V_0$ is vertex point,  $V_1, ..., V_N$  edge points, and  $V_{N+1}, ..., V2N$  face points), and control points in offsetting mesh as  $\Delta V_0, ..., \Delta V_{2N+27}$ . The orderings are shown in Fig. 8.9, the additional vertices in offsetting mesh  $\Delta V_{2N+8}, ..., \Delta V_{2N+16}$  and  $\Delta V_{2N+17}, ..., \Delta V_{2N+27}$ (blue and red line in Fig. 8.9(b)) are defined with counter-clock ordering. With  $\Omega$ -Partition (Fig. 8.8), we obtain the control meshes of 9 sub-faces  $C_{m,0}, ..., C_{m,9}$ , where  $C_{m,9}$  has 2N + 8 control vertices, all others have 16 control vertices.

Since each CCS limit point  $d_V$  of a control point V is the affine combination of this vertex point, edge points  $E_i$ 's and face points  $F_i$ 's (Fig. 8.2),

$$d_V = \frac{N}{N+5}V + \frac{4}{N+5}\sum_{i=1}^{N} E_i/N + \frac{1}{N+5}\sum_{i=1}^{N} F_i/N,$$
(8.12)

in order to interpolate all data points in M,  $\Delta S(u, v)$ 's must interpolate the difference vectors  $\Delta M$  between these data points and corresponding CCS limit points, thus these control meshes must satisfy

$$d_{\Delta V_0} = V_0 - d_{V_0}, \qquad d_{\Delta V_{2N+9}} = V_1 - d_{V_1}$$
$$d_{\Delta V_{2N+12}} = V_{N+1} - d_{V_{N+1}}, \quad d_{\Delta V_{2N+15}} = V_2 - d_{V_2}$$
(8.13)



Figure 8.10: linear independence of interpolation offsetting data points

With equation (8.12), for data point  $d_{\Delta V_0}$  at  $\Delta V_0$  (black circle in Fig. 8.9), we have

$$a_0 x_0 = d_{\Delta V_0}, \tag{8.14}$$

where  $a_0$  is a row vector of size 2N + 1 with  $\left[\frac{N}{N+5}, \frac{4}{(N+5)N}, ..., \frac{1}{(N+5)N}, ...\right], x_0$  is a vector of of size 2N + 1 with  $[\Delta V_0, \Delta V_1, ..., \Delta V_{N+1}, ...]$  (black circle/dots in Fig. 8.9). Similarly we can obtain the linear equations for  $d_{\Delta V_{2N+9}}, d_{\Delta V_{2N+12}}$  and  $d_{\Delta V_{2N+12}}$ . In Fig. 8.10, the relevant control points of each interpolation offsetting data points (circles) are marked with different colors, we see that the computation of each interpolation offsetting data point is independent from the computation of other interpolation offsetting data points in the mesh.

With equations (8.13) and (8.14), we can obtain a local linear system for each data point in M as shown in equations (8.4) and (8.5). Since earlier naming of vertices are based on face, here we rename the vertices for entire mesh. For each non-boundary data point  $P_i$  of valence N in M we define 2N + 1 control points in offsetting control mesh  $\Delta M$  as  $P_{i,0}, P_{i,1}, \dots P_{i,2N}$ , we then obtain k (number of non-boundary points in M) linear equations, where each non-boundary data points are computed locally with equation (8.14),

$$a_i \bar{x}_i = d_i, i \in [0, k-1] \tag{8.15}$$

where  $a_i$  is the coefficient row vector of size 2N+1 defined in euation (8.14),  $\bar{x}_i$  is the vector of corresponding control points  $P_{i,0}, ..., P_{i,2N}$  in  $\Delta M$ , and  $\bar{d}_i$  is the difference vector of  $P_i$  and its CCS limit point. To interpolate a data mesh M with k nonboundary data points, we need to construct an offsetting mesh  $\Delta M$  of size 9k, which satisfy the interpolation requirement set in equation (8.15), k local linear equations.

To solve the local linear system set by equation (8.14), we can have too many freedom for  $\Delta M$ . Additional constraints need to be introduced. In this paper, we choose a heuristic solution to solve this local linear system, for each  $P_i$  in M, we choose corresponding control points  $P_{i,m}$  in  $\Delta M$  as

$$P_{i,m} = \bar{d}_i, \ m = 0, ..., 2N.$$
 (8.16)

With choice of control points of  $\Delta M$  shown in equation (8.16), the 1<sup>st</sup> and 2<sup>nd</sup> derivatives on interpolating control points of offsetting limit surface is vanished. The

new interpolating surface has the same curvature on the given data points as its CCS limit surface on M, and their shapes are similar.

Above, we introduced the mathematical setup of our new interpolation scheme. Equations (8.6) and (8.9) define original CCS limit surface, (8.11) and (8.16) define the offsetting surface, then (8.10) defines our new interpolation surface by adding original CCS limit surface with offsetting surface.

### 8.5 Behavior of New Interpolation Scheme

In this section, we discuss behavior of our new one-step interpolation scheme.

Our new scheme will generate 2 CCS meshes, one is the given base mesh M to interpolate, another is the offsetting mesh  $\Delta M$ . If M has k non-boundary data points, then  $\Delta M$  has 9k control points. In order to interpolate data mesh M, our one-step interpolation surface is obtained by adding limit surface of  $\Delta M$  to the limit surface of M.

Since both M and  $\Delta M$  are CCS control meshes, we can derive

**Theorem 8.1.** Our new one-step interpolation surface is  $C^2$  continuous everywhere except at extraordinary point of M, where it is  $C^1$  continuous.

Proof. With mesh structure defined in Fig. 8.9, the offsetting mesh  $\Delta M$  has exactly the same number of extraordinary faces as M. Since M and  $\Delta M$  are both CCS meshes, their CCS limit surfaces are  $C^2$  continuous everywhere except at extraordinary points. At an arbitrary limit point (not extraordinary point) on the new interpolation surface, it is trivial to prove with equation (8.10) (by computing 1st and 2nd order derivatives) that it is  $C^2$  continuous. At arbitrary extraordinary point of M, since both limit surfaces of M and  $\Delta M$  are  $C^1$  at extraordinary points, at  $\bar{S}(0,0)$ , the resulting surface must be also  $C^1$  continuous. We notice from equations (8.10),(8.6) and(8.11) that our new scheme has the local support. This is in contrast with traditional interpolation schemes where local support is lost. In traditional interpolation schemes, if a data point is changed, then by solving a global linear system (8.3), all control points in x might be changed, such that the limit surface will change globally. This artifact prevents us from certain applications requiring matching surfaces between 2 3D objects in CAD/Computer Graphics, such as mold manufacturing, parts assembling. Our new scheme maintains the local support of CCS, such that change one data point will not change the interpolation limit surface globally, instead it will change only 2 rings of surfaces surrounding that data point, the same local support as CCS.

**Theorem 8.2.** The new interpolation surface has the same local support as its CCS base surface.

*Proof.* Our new interpolation surface is obtained by parametrically adding two CCS surfaces, 1st part is original CCS base surface, 2nd part is the offsetting surface. The 1st part has local support. The control points of 2nd part is derived from difference vectors between control points and data points of 1st part shown in equations (8.15) and (8.16), such that the resulting new interpolating surface has the same local support as the original CCS limit surface.

In our new scheme, the CCS base surface part is global, but the offsetting surface part is local. By parametrically adding an offsetting surface which locally interpolating all difference vectors between data points and their CCS limit points to its CCS base surface on the given data mesh, the generated interpolating surface follows the global shape of original CCS base surface while local offsetting surfaces enforce the interpolation directly.

Implementation results in Fig. 8.3 show that the resulting interpolation surfaces generated by our new scheme is smooth and of high quality. No fairing is generally



Figure 8.11: A face mask. Top row shows the views of original data mesh, from left to right: a). original surface; b). enlarged view of blue box in a); c) enlarged mesh view of blue box in b). Bottom row shows the generated interpolation limit surface with one-step scheme, and has the same sequence of enlarged views as in top row

needed to resolve the undulation caused by solving global linear system of traditional schemes. Furthermore the new scheme works well for large data sets. Table 1 shows the comparison of running time for Fig. 8.1, Fig. 8.12 and Fig. 8.13 between CCS and our new interpolation scheme (machine spec: CPU intel i5-2430M, RAM 4GB). From table 8.1, one can conclude that our new scheme can handle millions of interpolating data points efficiently.



Figure 8.12: A statue. Left is our new interpolating limit surface. Right two rows show enlarged views of two blue boxes in left image, where: a). enlarged view of blue box in original surface without interpolation; b) new interpolating surface of a); c) enlarged mesh view of blue box in a); d) enlarged mesh view of interpolating surface of blue box in b).

Table 8.1: Comparing running time of CCS and new one-step interpolation scheme for large data sets.

Data mesh		Running Time (seconds)	
Example	Vertices	CCS	one-step
Fig. 1	1,200,002	38.53	49.212
Fig. 11	$596,\!423$	17.812	23.618
Fig. 12	1,500,354	46.275	61.501

## 8.6 Summary

Traditional CCS interpolation schemes obtain an interpolation surface by solving a global linear system in equation (8.3), this will bring difficulties for the iterative scheme to handle large data set. Besides, the resulting interpolating surface does not have local property.

In this work, by combining two techniques: mesh decomposition and surface offsetting, we present a new interpolation scheme for Carmull-Clark subdivision surfaces that would not only be able to handle very large data sets (with millions of data points), but also allow the generated interpolating surface to have local property. Implementation result shows that a smooth and high quality interpolation surface can be generated by applying this new scheme, this is an important technique for applications with large data sets such as reverse engineering of scanned data sets and 3D printing.

Our next step is to do further research on offsetting surfaces, explore various control points selection on offsetting mesh and verify the impact of different selections.

## Chapter 9

# **Conclusion and Future Work**

### 9.1 Conclusion

This dissertation introduces our research work in subdivision surfaces. Our research work related to this dissertation focuses on two research questions, "How to improve smoothness around extraordinary points?" and "How to efficiently solve interpolation problems of subdivision surfaces?".

First of all, we develop a new subdivision scheme to improve smoothness around extraordinary points of Catmull-Clark subdivision surface. Our new scheme, named *Guided Catmull-Clark Subdivision Surface*. guarantees curvature continuity at CCS extraordinary points. In contrast to Zorin's work on extraordinary points with limit surface blending [73], our new scheme is purely subdivision based and uses mesh blending technique (part of control points are mapped from *dominative control meshes*), hence it is stationary. Furthermore, the new scheme avoids the hassle to recompute eigenvalues and eigenbases for every valence in the original CCSS. Instead, with Extraordinary-Points-Avoidance model and mesh blending technique, the eigenstructures of the new scheme have different eigenvalues of  $1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}$  (the eigenvalues for regular bi-cubic subdivision), so the scheme has a unique eigenbase for

any valence. The GCCSS is also flexible, we can adjust the shape of the subdivision surface by fine-tuning the dominative control meshes as far as the choice of control points fulfill the requirement set forth in this work. The linear system for choosing the control points of 2N dominative control meshes is underdetermined, so this leaves room for changing the shape of the subdivision surface without sacrificing the surface continuity.

We also develop a new subdivision scheme on Polar Catmull-Clark mesh (PCC mesh), named Polar embedded Catmull-Clark Subdivision (PCCS). By introducing Polar configuration on high valence vertex, the ripple problem inherent in a CCS surface is solved. The subdivision scheme developed has the properties that the limit surface on the CCS part is exactly the same as a CCS limit surface and the limit surface on the Polar part is  $G^2$  continuous everywhere. Since it is inevitable to have high valence extraordinary points in some cases, e.g. airplanes, rockets and engineering parts, the currently available CCS meshes can be easily converted to PCC meshes, such that one can avoid redesigning the complete mesh. In contrast to commonly used Polar subdivision rules, the subdivision masks of proposed Guided U-Subdivision(GUS) on Polar part are obtained by mesh blending with *dominative control meshes* (the same mesh blending technique used in GCCS). The properties of GUS surfaces are studied and proven. The GUS scheme is a stationary scheme.

For interpolation problems of subdivision surfaces, we first introduce a simple interpolating scheme for quad parametric subdivision surface, called Bezier Crust. We show that by parametrically adding a special bi-quintic Bezier crust on original subdivision surface one can generate an interpolating surface which maintains curvature conditions of original subdivision surface. With special construction of bi-quintic Bezier crust, we can avoid to calculate a global linear system common in earlier interpolation schemes, such that the computation is local and simple. Implementation results on Catmull-Clark subdivision surfaces show that the Bezier crust interpolating scheme can generated visually well behaved limit surfaces, such that barely no fairing needed to correct the undulations caused by computation of global linear system used in earlier interpolation schemes. The Bezier crust on quad subdivision surface shows also advantages over direct Bezier surface methods. To obatin  $C^2$  surface, it is required to have bi-quintic piecewise Bezier surface, however, the Bezier surface method requires also interpolation of normals on vertices, which is complex and difficult to obtain a well behaved surface when data set is large. Overall, in this work we provide a local  $G^2$  interpolating scheme for quad subdivision surface. With simplicity of this new scheme, it can be easily applied to quad approximating subdivision surfaces and convert them to interpolating schemes, making them more appropriate for CAD, CAGD, face recognition and other interpolation required applications.

We then introduce a new heuristic interpolation scheme on Polar surfaces, especially on PCCSSs. We show that, by vertex splitting, we can treat a Polar face as a quad face, such that the bi-quintic offsetting Bezier Crust [66] can be applied to the Polar faces as well. The generated interpolating surface maintains the continuity of underlying PCCS limit surface, i.e.  $G^2$  at Polar extraordinary points,  $C^1$  at CCS extraordinary points, and  $C^2$  everywhere else. Implementation results show that our new scheme can generate high quality images appropriate for engineering and computer graphics usage.

With the concept similar to solving interpolating problems with Bezier Crust, we further introduce a new thin shell hollowing model on 3D objects represented by Catmull-Clark subdivision surfaces. Our new method for inward offsetting works by subtracting a thin layer of bi-quintic Bezier crust from the original CCS surface. The new offsetting surface generated is visually smooth and has the same continuity as the original CCS limit surface. The properties of new offsetting surface are also discussed in this work. Implementation results show that the offsetting surface generated is free from creases, and filling the gaps is trivial due to the fact that the offsetting surface is the parametric sum of the original CCS surface and a Bezier crust on difference vectors of size d on each face. Since a bi-quintic Bezier crust does not change the curvature at a corner data point of the CCS limit surface, one would not get gaps at connections of offsetting faces commonly found in earlier methods.

While Bezier Crust can solve the CCS interpolation problem well, we still prefer a lower degree solution on interpolating problems. In this dissertation work, we present a new interpolation scheme for Carmull-Clark subdivision surfaces only, called *Onestep Bicubic Interpolation*. The new scheme works by combining two techniques: mesh decomposition and surface offsetting. Since the offsetting surface and base surface have the same topology and share the same subdivision rules, the new scheme would not only be able to handle very large data sets (with millions of data points), but also allow the generated interpolating surface to have local property. Implementation result shows that a smooth and high quality interpolation surface can be generated by applying this new scheme, this is an important technique for applications with large data sets such as reverse engineering of scanned data sets and 3D printing.

Overall, in this dissertation work, we developed two new schemes to solve the smoothness problem at extraordinary points of Catmull-Clark subdivision surface and Polar surface. We also developed two new interpolation schemes, Bezier Crust and One-step Bicubic, to convert an approximating subdivision scheme like CCS into interpolating one. A lot of examples are tested on these new schemes with good results.

#### 9.2 Future Research

We have done some fundamental research, incl. improvement of surface smoothness and surface interpolating issues, related to representation of topologically complex 3D objects, and obtained some good results. The following are some research problems inspired by this dissertation research. They constitute my main research interest and research directions in the near future.

**Exact Evaluation of GCCSS**: in GCCS, linear system of dominative control meshes is under-determined. Evaluation of various solutions to the linear system will be necessary to obtain a unified approach for better shape of GCCSS at extraordinary points.

Local shape control of One-step Bicubic Interpolation: in one-step scheme on CCSS, the offsetting mesh is specially selected. We will explore various selection Scenarios and evaluate these approaches. We will work out some selection criteria on offsetting mesh to further improve the shape of generated interpolating surface.

Subdivision surface modeling with sparse data points: I plan to use hierarchical data mesh structure to present object, to improve the surface quality in different levels while preserving the overall 3D contours.

Medical image processing, with focus on establishing 3D modeling from 2D contours: I will evaluate the current schemes of 3D modeling, and develop new 3D model for medical 2D images, especially the de-noising schemes to obtain a more accurate result.

Heterogeneous composite material modeling: it is common that CAD/CAM will require processing of 3D objects with different material composition, while different materials will generally require different tools. I will evaluate the current schemes on surface representation on heterogeneous composite material, and develop new scheme basing on the schemes of subdivision surface presented in this dissertation work.

Note the copyright notice at the end of each chapter.

Copyright<sup>©</sup> Jianzhong Wang, 2015.

# Bibliography

- BALL, A., AND STORRY, D. Conditions for tangent plane continuity over recursively generated B-spline surfaces. ACM Transactions on Graphics (TOG) 7, 2 (1988), 102.
- [2] BARSKY, B. A., AND DEROSE, T. D. Geometric continuity of parametric curves: three equivalent characterizations. *IEEE Computer Graphics and Applications 9*, 6 (1989), 60–68.
- [3] BECK, A., AND TEBOULLE, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences 2*, 1 (2009), 183–202.
- [4] BEZIER, P. The mathematical basis of the UNISURF CAD system. Butterworth-Heinemann, 1986.
- [5] BOIER-MARTIN, I., AND ZORIN, D. Differentiable parameterization of catmullclark subdivision surfaces. In *Proceedings of the 2004 Eurographics/ACM SIG-GRAPH symposium on Geometry processing* (2004), ACM, pp. 155–164.
- [6] CATMULL, E., AND CLARK, J. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design 10*, 6 (1978), 350–355.
- [7] CAVARETTA, A. S., DAHMEN, W., AND MICCHELLI, C. A. Stationary subdivision, vol. 453. American Mathematical Soc., 1991.
- [8] CHEN, Z., LUO, X., TAN, L., YE, B., AND CHEN, J. Progressive interpolation based on catmull-clark subdivision surfaces. In *Computer Graphics Forum* (2008), vol. 27, Wiley Online Library, pp. 1823–1827.
- [9] CHENG, F.-H. F., FAN, F.-T., LAI, S.-H., HUANG, C.-L., WANG, J.-X., AND YONG, J.-H. Loop subdivision surface based progressive interpolation. *Journal of Computer Science and Technology* 24, 1 (2009), 39–46.
- [10] CHIU, W., AND TAN, S. Using dexels to make hollow models for rapid prototyping. Computer-Aided Design 30, 7 (1998), 539–547.
- [11] CLAES, J., BEETS, K., AND VAN REETH, F. A corner-cutting scheme for hexagonal subdivision surfaces. In *Shape Modeling International*, 2002. Proceedings (2002), IEEE, pp. 13–20.

- [12] DEGEN, W. L. Explicit continuity conditions for adjacent bézier surface patches. Computer Aided Geometric Design 7, 1 (1990), 181–189.
- [13] DEROSE, A. D. Geometric continuity: a parametrization independent measure of continuity for computer aided geometric design. PhD thesis, University of California, Berkeley, 1985.
- [14] DEROSE, T. D. Necessary and sufficient conditions for tangent plane continuity of bezier surfaces. *Computer Aided Geometric Design* 7, 1 (1990), 165–179.
- [15] DEROSE, T. D., AND BARSKY, B. A. An intuitive approach to geometric continuity for parametric curves and surfaces. Springer, 1985.
- [16] DOO, D. A subdivision algorithm for smoothing down irregularly shaped polyhedrons. In *Proceedings on Interactive Techniques in Computer Aided Design* (1978), vol. 157, Bologna, p. 165.
- [17] DOO, D., AND SABIN, M. Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design 10*, 6 (1978), 356–360.
- [18] DYN, N., LEVINE, D., AND GREGORY, J. A. A butterfly subdivision scheme for surface interpolation with tension control. ACM transactions on Graphics (TOG) 9, 2 (1990), 160–169.
- [19] ELBER, G., AND COHEN, E. Tool path generation for freeform surface models. In Proceedings on the second ACM symposium on Solid modeling and applications (1993), ACM, pp. 419–428.
- [20] ELBER, G., AND COHEN, E. Adaptive isocurve-based rendering for freeform surfaces. ACM Transactions on Graphics (TOG) 15, 3 (1996), 249–263.
- [21] FARIN, G. E. Curves and surfaces for CAGD [electronic resource]: a practical guide. Morgan Kaufmann, 2002.
- [22] FOWLER, A., AND WILSON, C. Cubic spline, a curve fitting routine. Tech. rep., Union Carbide Nuclear Co. Y-12 Plant, Oak Ridge, Tenn., 1962.
- [23] GANESAN, M., AND FADEL, G. M. Hollowing rapid prototyping parts using offsetting techniques. In *Proceedings of the Fifth International Conference on Rapid Prototyping* (1994), University of Dayton, pp. 241–251.
- [24] GARRITY, T., AND WARREN, J. Geometric continuity. Computer Aided Geometric Design 8, 1 (1991), 51–65.
- [25] GREGORY, J. A. Geometric continuity. Mathematical methods in computer aided geometric design (1989), 353–371.
- [26] GREGORY, J. A., AND HAHN, J. M. Aci supi 2i/supi polygonal surface patch. Computer Aided Geometric Design 6, 1 (1989), 69–75.

- [27] HAGEN, H. Bézier-curves with curvature and torsion continuity. Journal of Mathematics 16, 3 (1986), 629–638.
- [28] HAHN, J. M. Geometric continuous patch complexes. Computer Aided Geometric Design 6, 1 (1989), 55–67.
- [29] HALSTEAD, M., KASS, M., AND DEROSE, T. Efficient, fair interpolation using Catmull-Clark surfaces. In Proceedings of the 20th annual conference on Computer graphics and interactive techniques (1993), ACM, p. 44.
- [30] HATNA, A., GRIEVE, R., AND BROOMHEAD, P. Offsetting 3d contours on parametric surfaces. The International Journal of Advanced Manufacturing Technology 16, 3 (2000), 189–195.
- [31] HERMANN, T., AND LUKÁCS, G. A new insight into the g n continuity of polynomial surfaces. Computer aided geometric design 13, 8 (1996), 697–707.
- [32] KAHMANN, J. Continuity of curvature between adjacent bézier patches. Surfaces in Computer Aided Geometric Design (1983), 65–75.
- [33] KARČIAUSKAS, K., AND PETERS, J. Bicubic polar subdivision. ACM Transactions on Graphics (TOG) 26, 4 (2007), 14.
- [34] KOBBELT, L. Interpolatory subdivision on open quadrilateral nets with arbitrary topology. In *Computer Graphics Forum* (1996), vol. 15, John Wiley & Sons, pp. 409–420.
- [35] KOC, B., AND LEE, Y.-S. Non-uniform offsetting and hollowing objects by using biarcs fitting for rapid prototyping processes. *Computers in Industry* 47, 1 (2002), 1–23.
- [36] LAI, S., AND CHENG, F. Parametrization of general catmull-clark subdivision surfaces and its applications. *Computer-Aided Design and Applications* 3, 1-4 (2006), 513–522.
- [37] LAI, S., AND CHENG, F. Similarity based interpolation using Catmull-Clark subdivision surfaces. *The Visual Computer 22*, 9 (2006), 865–873.
- [38] LAI, S., AND CHENG, F. Fast mesh interpolation and mesh decomposition with applications. *International Journal of Image and Graphics 12*, 01 (2012).
- [39] LAM, T., YU, K., CHEUNG, K., AND LI, C. Octree reinforced thin-shell rapid prototyping. *Journal of materials processing technology* 63, 1 (1997), 784–787.
- [40] LEVIN, A. Modified subdivision surfaces with continuous curvature. ACM Transactions on Graphics (TOG) 25, 3 (2006), 1040.
- [41] LI, C., YU, K., AND LAM, T. Implementation and evaluation of thin-shell rapid prototype. *Computers in industry 35*, 2 (1998), 185–193.

- [42] LIU, D., AND HOSCHEK, J. Gc; sup¿ 1;/sup¿ continuity conditions between adjacent rectangular and triangular bézier surface patches. *Computer-Aided Design* 21, 4 (1989), 194–200.
- [43] LOOP, C. Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah, Department of Mathematics (1987).
- [44] LOOP, C., AND SCHAEFER, S. G2 tensor product splines over extraordinary vertices. In *Computer Graphics Forum* (2008), vol. 27, John Wiley & Sons, pp. 1373–1382.
- [45] LOTT, N. J., AND PULLIN, D. Method for fairing b-spline surfaces. COMP. AIDED DES. 20, 10 (1988), 597–604.
- [46] MAEKAWA, T. An overview of offset curves and surfaces. Computer-Aided Design 31, 3 (1999), 165–173.
- [47] MANNING, J. Continuity conditions for spline curves. The Computer Journal 17, 2 (1974), 181–186.
- [48] MYLES, A. curvature-continuous bicubic subdivision surfaces for polar configurations, 2008.
- [49] MYLES, A., KARČIAUSKAS, K., AND PETERS, J. Pairs of bi-cubic surface constructions supporting polar connectivity. *Computer Aided Geometric Design* 25, 8 (2008), 621–630.
- [50] MYLES A., P. J. Bi-3 c2 polar subdivision. ACM Trans. Graph 28, 3 (2009), 1–12.
- [51] PARK, S. C. Hollowing objects with uniform wall thickness. Computer-Aided Design 37, 4 (2005), 451–460.
- [52] PATRIKALAKIS, N., AND BARDIS, L. Offsets of curves on rational b-spline surfaces. *Engineering with Computers* 5, 1 (1989), 39–46.
- [53] PETERS, J., AND KARČIAUSKAS, K. An introduction to guided and polar surfacing. Mathematical Methods for Curves and Surfaces (2010), 299–315.
- [54] PETERS, J., AND REIF, U. The simplest subdivision scheme for smoothing polyhedra. ACM Transactions on Graphics (TOG) 16, 4 (1997), 420–431.
- [55] PETERS, J., AND SHIUE, L. Combining 4-and 3-direction subdivision. ACM Transactions on Graphics (TOG) 23, 4 (2004), 980–1003.
- [56] PRAUTZSCH, H. Smoothness of subdivision surfaces at extraordinary points. Advances in Computational Mathematics 9, 3 (1998), 377–389.
- [57] QU, R. Recursive subdivision algorithms for curve and surface design. School of Information Systems, Computing and Mathematics (1990).

- [58] REIF, U. A degree estimate for subdivision surfaces of higher regularity. Proceedings of the American Mathematical Society 124, 7 (1996), 2167–2174.
- [59] SABIN, M. A. Parametric splines in tension. No. VTO/MS/160, British Aircraft Corporation, Weybridge, Surrey, England 23 (1970).
- [60] SCHAEFER, S., AND WARREN, J. On c 2 triangle/quad subdivision. ACM Transactions on Graphics (TOG) 24, 1 (2005), 28–36.
- [61] STAM, J. Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values. In Proceedings of the 25th annual conference on Computer graphics and interactive techniques (1998), ACM, p. 404.
- [62] STAM, J. Fast evaluation of loop triangular subdivision surfaces at arbitrary parameter values. In Computer Graphics (SIGGRAPH98 Proceedings, CD-ROM Supplement) (1998).
- [63] STAM, J., AND LOOP, C. Quad/triangle subdivision. In Computer Graphics Forum (2003), vol. 22, Wiley Online Library, pp. 79–85.
- [64] SZELISKI, R. Fast surface interpolation using hierarchical basis functions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 12*, 6 (1990), 513–528.
- [65] WANG, J., AND CHENG, F. Guided catmull-clark subdivision. Technical Report by Dept of Computer Science at University of Kentucky TR519-11 (2011).
- [66] WANG, J., AND CHENG, F. Bezier Crust on Quad Subdivision Surface. In Pacific Graphics Short Papers (2013), The Eurographics Association, pp. 29–34.
- [67] WANG, J., AND CHENG, F. Polar embedded catmull-clark subdivision surface. In Computer-Aided Design and Computer Graphics (CAD/Graphics), 2013 International Conference on (2013), IEEE, pp. 24–31.
- [68] WANG, J., AND CHENG, F. A heuristic offsetting scheme for catmull-clark subdivision surfaces. *Computer-Aided Design and Applications 12*, 6 (2015).
- [69] YE, X., LIANG, Y., AND NOWACKI, H. Geometric continuity between adjacent bézier patches and their constructions. *Computer aided Geometric design 13*, 6 (1996), 521–548.
- [70] ZHANG, C., ZHANG, P., AND CHENG, F. F. Fairing spline curves and surfaces by minimizing energy. *Computer-Aided Design* 33, 13 (2001), 913–923.
- [71] ZHENG, J., AND CAI, Y. Interpolation over arbitrary topology meshes using a two-phase subdivision scheme. Visualization and Computer Graphics, IEEE Transactions on 12, 3 (2006), 301–310.
- [72] ZORIN, D. Subdivision zoo. Subdivision for modeling and animation (1999), 65–104.

[73] ZORIN, D. Constructing curvature-continuous surfaces by blending. In *Proceedings of the fourth Eurographics symposium on Geometry processing* (2006), Eurographics Association, p. 40.

# Vita

#### Personal Data

Name: Jianzhong Wang Place of Birth: Shanghai, China

### **Educational Background**

- Master of Science in Information Systems, Pforzheim University of Applied Science, Germany 2004
- Bachelor of Science in Mechanical Engineering, China Textile University, China 1994

#### Awards

- Student Travel Award from Graduate School Fellowship of University of Kentucky, 2013
- Graduate School Academic Year Fellowship of University of Kentucky, 2012-2013
- Outstanding Teaching Assistant from ACM UK Student Chapter of University of Kentucky, 2011

#### Journal, Conference Papers and Papers Submitted

- Jianzhong Wang, Fuhua Cheng, Fan Zhang, One-step Bicubic Interpolation, submitted, 2015
- Jianzhong Wang, Fuhua Cheng, A G<sup>2</sup> Interpolation Scheme for Polar Surface, to appear in Journal of Computer-Aided Design and Applications, 2015
- Jianzhong Wang, Fuhua Cheng, A Heuristic Offsetting Scheme for Catmull-Clark Subdivision Surfaces, Journal of Computer-Aided Design and Applications, Vol. 12, No.6, 2015
- Jianzhong Wang, Fuhua Cheng, Polar embedded Catmull-Clark Subdivision Surfaces, 13th International Conference on Computer-Aided Design and Computer Graphics, In Proceeding, 2013
- Jianzhong Wang, Fuhua Cheng, Bezier Crust on Quad Subdivision Surface, Pacific Graphics 2013, In Proceeding, 2013
- Jianzhong Wang, Fuhua Cheng, Guided Catmull-Clark Subdivision, Technical Report TR519-11, Dept of Computer Science, University of Kentucky, 2012, submitted to Journal, 2015
- Fan F., Cheng F., Huang C., Li Y., Wang J., Lai S., Mesh Clustering by Approximating Centroidal Voronoi Tessellation, Proceeding pp. 301-306, 2009
   SIAM/ACM Joint Conference on Geometric and Physical Modeling, 2009