이학석사 학위논문

Freeform surface modeling with Bicubic Coons patch and its approximation to Bézier surface

(쿤스 패치를 이용한 자유곡면 모델링과 베지에 곡면으로의 근사)

2018년 2월

서울대학교 대학원

수 리 과 학 부

신 연

Freeform surface modeling with Bicubic Coons patch and its approximation to Bézier surface

by

Synn, Yeon

A DISSERTATION

Submitted to the faculty of the Graduate School
in partial fulfillment of the requirements
for the degree Master of Science
in the Department of Mathematics
Seoul National University
February 2018

Abstract

We approximate a freeform NURBS surface using the bicubically blended Coons patches. From the Coons surface construction, each subpatch of a NURBS object can be bounded very efficiently using the bicubic surface determined by the four boundary curves. To use the bounding volume hierarchy for a freeform surface modeling in many geometric algorithms, the explicit value of control points are essential and required. Also, there is a interrelation between the size of the domain of a surface and the approximation rate (or the approximation speed) from a subdivision in the domain. We present the control points of the bicubically blended Coons patches for a Bézier surfaces. Also we proposed a hypothesis about the decay rate in a freeform surface approximation, how to find approximation rate of errors for the bicubically blended Coons patch between the previous step of subdivision state and the subsequent step of subdivision state from the error analysis.

Keywords : Freeform Surface, NURBS, Coons patch, Bézier surface, Control points, Surface approximations

Student number : 2015-20265

Contents

Chapter 1

Introduction

1.1 Introduction

The Coons patch is a pioneer of the freeform surface representation scheme in computer aided geometric design(CAGD) and it was developed in 1964 by S. A. Coons [4]. With other freeform surface modeling methods, such as B-splines, Non-uniform rational basis splines(NURBS) surfaces, or Bézier surfaces, the Coons patches were used fairly infrequently in of the same period freeform surface modeling applications. In spite of the few quantity of that consumed, there are a lot of useful characteristics which we investigate in this work. The important properties of the Coons patches are the following:

(i) It can interpolates arbitrary boundaries exactly.

(ii) The smoothness of surface is equivalent to the minimum smoothness of boundary curves.

(iii) It don't provide a higher continuity across boundaries.

(iv) It can be uniquely defined by their boundary curves.

The freeform geometric modelings are more compact and smoother than the polygon meshes. The bounding volume structure of the freeform geometry can be generated by recursively subdividing a freeform surfaces. In spite of it is not clear, generally, where to stop the recursive subdivisions and how to progress using the geometric computations when that computations were reached leaf level in the computation such as accelerating algorithms. Most of freeform geometric modelings have high computation cost, so the approximation with Coons patches can reduce these computation cost. We first note that if the subdivision level goes to be higher, then for each Coons subpatches, size of domain must be reduced in general. So to reduce the computation cost, we must have a subdivision level of domain as few as possible. Note that almost all of freeform surfaces can be reconstructed by the NURBS. Moreover, the well known fact is NURBS can be subdivided by the finite number of Bézier surfaces. So in this paper, we mainly deal with the error comparison depending on the subdivision levels between constructed Coons patch approximation and NURBS object structure especially, subdivided by Bézier surfaces (i.e. the distance concerning the size of domain) also control points of the Coons patches to Bézier surfaces.

A chain of recursive subdivision of Bézier surfaces gives Coons subpatches that is the same number of pieces of subdivided Bézier surfaces. This procedure contains these Coons patch which is composed by a number of Coons subpatches depend on a subdivision level approximating the original NURBS or a freeform surfaces to within for an arbitrary error bound. Interior nodes of bounding volume hierarchy correspond to freeform surface patches these are recursively subdivided [7]. Nevertheless it might seem there are no dramatic differences from conventional approaches, but the remarkable difference is in the decreasing rate of between the previous step of subdivision state and the subsequent step of subdivision state(or the leaf nodes). Because the bicubic Coons patch interpolates pretty close the freeform surface more tightly than bilinear Coons patches, polygonal meshes or the others, the recursive subdivision process terminates earlier and the size of meshes in bicubic Coons patches are much larger than bilinear one or polygonal mesh approximation for a same value of tolerances, in general.

The next contents of introduction is composed by the related works. In the chapter 2, we explain the bicubic Coons patch and its bounding volume. Also we proposed the control points of Coons patch of an corresponding boundary curves from arbitrary Bézier surfaces. In the chapter 3, we have a experiment about some specific geometric models and find error between original surfaces and corresponding bicubic Coons patches. Also we compared the errors from the subdivision level in the same geometric model.

The main contributions of in this paper can be condensed as follows:

- We propose the control point(in the Bézier form) of bicubically blended Coons patch with the bicubic Hermite basis for Bounding Volume Hierarchy(BVH)

- We present a hypothesis about approximation rate which comes from the subdivision level of a Bézier surface and the error between Bézier surface and bicubically blended Coons patch.

1.2 Related works

The error analysis and recognizing approximating aspect of a surface approximation have been very difficult task to demonstrate. In the case of curve approximation, Cohen et al[3] found the approximation order $O(n^{-2})$ between the control polygons and the Bezier curves. This was the start of observing control polygon. Hall [6] demonstrated the error bounds for the cubic splines from the freeform curves. In the book of de Boor [2], almost all of curve approximation algorithm is discussed, and this book also presented the approximation rates in many cases of the freeform curve.

In the case of freeform surfaces, Filip et al. [5] demonstrated the approximation rate of the bilinear Coons patch (i.e. they demonstrated the approximation rate of bilinear Coons, that is $O(n^{-2})$). That is proportional to the length of the side in domain for a triangular or a rectangular domain. Randrianarivony et al. [8] proposed how to improve regularity of Coons surfaces, and they also gives a efficient subdivision method for a Bézier surface. Also they [8] presented the explicit control point form of bilinear Coons patch to a Bézier surfaces. Kim et al. [7] proposed a compact boundary volume hierarchy(BVH) representation scheme for the freeform NURBS surfaces, which needs quite less memory space than conventional BVHs, and also they demonstrated the effectiveness of the proposed BVH scheme by developing real-time algorithms for a collision detection and a minimum distance computation, for freeform geometric models of non-trivial complexity. Aumann [1] presented the algorithm based upon the algorithm of de Casteljau to degree elevation of the developable Bézier surface and it gives accuracy of the approximation.

But these efforts do not mentioned any information about error analysis about bicubic approximation case or over in a freeform surface. So we present a detailed conjecture about that. If this conjecture will be proved, then the freeform geometric modeling methods can be improved with our experiments have useful aspect that gives influence to generate another more efficient new geometric algorithm to decreasing the error bounds(bounding volume) or definite errors.

Chapter 2

Coons patch with Hermite Basis

2.1 Preliminaries

Before talking up the main subjects, we will state some preliminaries. First of all, our works have fully relevance to the freeform surface modeling. The freeform surface modeling means that is a technique for engineering freeform Surfaces with a CAD or CAID system in the computer graphics. Here are some definitions to basic terminologies. I quote the following definitions from Wolfram Mathworld straightforwardly.

Definition 2.1.1. ***Control point*** is a member of a set of points used to determine the shape of a spline curve or, more generally, a surface or higher-dimensional object in Computer-Aided Geometric Design.

In the case of a freeform surface modeling, we must not explain everything without the control point. In the following order, we explain the

Bézier surfaces and the NURBS. These are the essential tools to generate freeform surface in CAGD.

Definition 2.1.2. ***Bézier Surfaces*** are a species of mathematical spline used in computer graphics, computer-aided design, and finite element modeling. As with the Bézier curve, a Bézier surface is defined by a set of control points. Let $B(u, v)$ be a Bézier surface It can be defined explicitly:

$$B(u,v) = \sum_{i=1}^{m}\sum_{j=1}^{m} \mathbf{b}_{ij} B_i^m(u) B_j^n(v)$$

where $\mathbf{b}_{ij}$s are the control points and B_i^m and B_j^n are the Bernstein Basis element. i.e. $B_i^m = (1-t)^{m-i}t^i$.

Definition 2.1.3. ***Non-uniform rational basis spline (NURBS)*** is a mathematical model commonly used in computer graphics for generating and representing curves and surfaces.

A ***NURBS curve*** defined by

$$C(t) = \frac{\sum_{i=0}^{n} N_i^p(t) w_i \mathbf{p}_i}{\sum_{i=0}^{n} N_i^p(t) w_i},$$

where p is the order, N_i^ps are the B-spline basis functions, $\mathbf{p}_i$ are control points, and the weight w_i of $\mathbf{p}_i$ is the last ordinate of the homogeneous point $\mathbf{p}_i^w$. These curves are closed under perspective transformations and can represent conic sections exactly.

A ***NURBS surface*** of degree (p, q) defined by

$$S(u,v) = \frac{\sum_{i=0}^{m}\sum_{j=0}^{n} N_i^p(u)N_j^q(v)w_{ij}\mathbf{p}_{ij}}{\sum_{i=0}^{m}\sum_{j=0}^{n} N_i^p(u)N_j^q(v)w_{ij}},$$

where N_i^p and N_j^q are the B-spline basis functions, $\mathbf{p}_{ij}$ are control points, and the weight w_{ij} of $\mathbf{p}_{ij}$ is the last ordinate of the homogeneous point $\mathbf{p}_{ij}^w$.

Now, we explain the main object of this paper. That is the Coons patch. This object is one of the simplest tool in the freeform surface modeling. Coons patch is generated in an intuitive method by the information of boundary of a freeform surface. The definition of the Coons patch is the following:

Definition 2.1.4. ***Coons patch*** is a type of manifold parametrization used in computer graphics to smoothly join other surfaces together, and in computational mechanics applications, particularly in finite element method and boundary element method, to mesh problem domains into elements. In short, this implies the **boundary interpolating surface**. Here is an example.

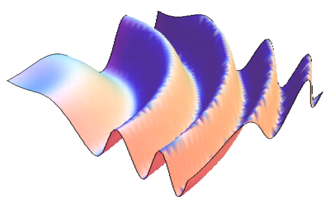

Figure 2.1: The Original Surcface

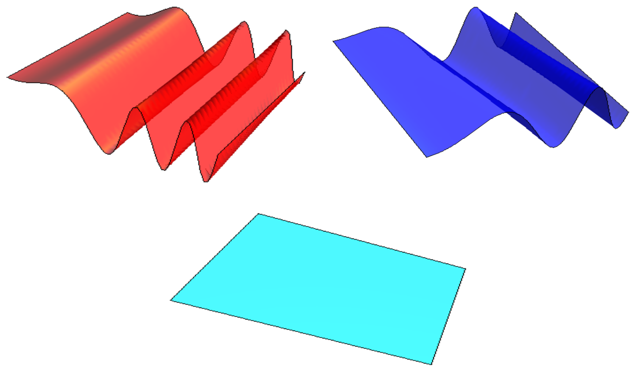

Figure 2.2: Constructing coons patch-Ruled Surface and Error Surface

For any given freeform surface in Figure 2.1, let Figure 2.2 be a first ruled surface parametrized by $\mathcal{P}$, a second ruled surface parametrized by $\mathcal{Q}$, a bilinear patch(or an error surface) parametrized by $\mathcal{PQ}$ in the order from 2.2.3 and 2.2.4. Then from a summation of the ruled surfaces and a subtraction of error surface, we get the Coons patch as in Figure 2.3

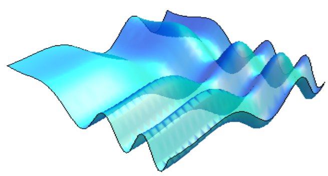

Figure 2.3: Generated Coons patch from $\mathcal{P} + \mathcal{Q} - \mathcal{PQ}$

2.2 Bicubic Coons patch

For any given freeform surface $\mathbf{x}(u,v)$, $0 \leq u,v \leq 1$, a bicubically blended Coons patch $\mathbf{s}(u,v)$ is defined by the information of the four boundary curves and the first-order partial derivatives of the boundary curves of $\mathbf{x}(u,v)$:

$$\begin{aligned}\mathcal{P}\mathbf{x} = \quad & \mathbf{x}(0,v)H_0(u) + \mathbf{x}_u(0,v)H_1(u) + \mathbf{x}_u(1,v)H_2(u) + \mathbf{x}(1,v)H_3(u) \\ \mathcal{Q}\mathbf{x} = \quad & \mathbf{x}(u,0)H_0(v) + \mathbf{x}_v(u,0)H_1(v) + \mathbf{x}_v(u,1)H_2(v) + \mathbf{x}(u,1)H_3(v)\end{aligned} \tag{2.2.1}$$

$$\mathcal{P}\mathcal{Q}\mathbf{x} = \begin{bmatrix} H_0(u) \\ H_1(u) \\ H_2(u) \\ H_3(u) \end{bmatrix}^t \begin{bmatrix} \mathbf{x}(0,0) & \mathbf{x}_v(0,0) & \mathbf{x}_v(0,1) & \mathbf{x}(0,1) \\ \mathbf{x}_u(0,0) & \mathbf{x}_{uv}(0,0) & \mathbf{x}_{uv}(0,1) & \mathbf{x}_u(0,1) \\ \mathbf{x}_u(1,0) & \mathbf{x}_{uv}(0,1) & \mathbf{x}_{uv}(1,1) & \mathbf{x}_u(1,1) \\ \mathbf{x}(1,0) & \mathbf{x}_v(1,0) & \mathbf{x}_v(1,1) & \mathbf{x}(1,1) \end{bmatrix} \begin{bmatrix} H_0(v) \\ H_1(v) \\ H_2(v) \\ H_3(v) \end{bmatrix} \tag{2.2.2}$$

Where H_0, H_1, H_2, H_3 are some arbitrary smooth functions satisfy following conditions:

- $H_0(t) + H_3(t) = 1$ (Partition of unity)
- $H_0(0) = H_3(1) = 1$, $H_0(1) = H_3(0) = 0$
- $H_0'(0) = H_0'(1) = H_3'(0) = H_3'(1) = 0$
- $H_1(0) = H_1(1) = H_2(0) = H_2(1) = 0$
- $H_1'(0) = H_2'(1) = 1$, $H_1'(1) = H_2'(0) = 0$
- $H_1'(t)$ and $H_2'(t)$ need not be the partition of unity.

Now, a Coons patch $\mathbf{s}$ can be defined by the relation of the followings:[8]

$$\mathcal{P} \oplus \mathcal{Q}\mathbf{x}(\text{or } \mathbf{s}) = \mathbf{s}, \quad \text{where} \tag{2.2.3}$$

$$\mathcal{P} \oplus \mathcal{Q} = \mathcal{P} + \mathcal{Q} - \mathcal{P}\mathcal{Q} \tag{2.2.4}$$

Let the cubic blending functions on the most simplest in the unit square domain are following:

$$\begin{aligned} H_0(t) = &\quad 2t^3 - 3t^2 + 1 \\ H_1(t) = &\quad t^3 - t^2 \\ H_2(t) = &\quad t^3 - 2t + t \\ H_3(t) = &\quad -2t^3 + 3t^2 \end{aligned} \tag{2.2.5}$$

We suppose in the sequal that the blending functions are sufficiently smooth. Then according to (2.2.4), we can express the bicubically Coons surface in a matrix form as:

$$\begin{aligned} \mathbf{s}(u,v) &= \mathcal{P}\mathbf{x} + \mathcal{Q}\mathbf{x} - \mathcal{P}\mathcal{Q}\mathbf{x} \\ &= \mathbf{x}(0,v)H_0(u) + \mathbf{x}_u(0,v)H_1(u) + \mathbf{x}_u(1,v)H_2(u) + \mathbf{x}(1,v)H_3(u) \\ &+\mathbf{x}(u,0)H_0(v) + \mathbf{x}_v(u,0)H_1(v) + \mathbf{x}_v(u,1)H_2(v) + \mathbf{x}(u,1)H_3(v) \\ &- \begin{bmatrix} H_0(u) \\ H_1(u) \\ H_2(u) \\ H_3(u) \end{bmatrix}^t \begin{bmatrix} \mathbf{x}(0,0) & \mathbf{x}_v(0,0) & \mathbf{x}_v(0,1) & \mathbf{x}(0,1) \\ \mathbf{x}_u(0,0) & \mathbf{x}_{uv}(0,0) & \mathbf{x}_{uv}(0,1) & \mathbf{x}_u(0,1) \\ \mathbf{x}_u(1,0) & \mathbf{x}_{uv}(0,1) & \mathbf{x}_{uv}(1,1) & \mathbf{x}_u(1,1) \\ \mathbf{x}(1,0) & \mathbf{x}_v(1,0) & \mathbf{x}_v(1,1) & \mathbf{x}(1,1) \end{bmatrix} \begin{bmatrix} H_0(v) \\ H_1(v) \\ H_2(v) \\ H_3(v) \end{bmatrix} \end{aligned} \tag{2.2.6}$$

From (2.2.3) it follows that $\mathbf{x}$ is of the form

$$\mathbf{s}(u,v) = - \begin{bmatrix} -1 \\ H_0(u) \\ H_1(u) \\ H_2(u) \\ H_3(u) \end{bmatrix}^t \begin{bmatrix} \mathbf{0} & \mathbf{x}(u,0) & \mathbf{x}_v(u,0) & \mathbf{x}_v(u,1) & \mathbf{x}(u,1) \\ \mathbf{x}(0,v) & \mathbf{x}(0,0) & \mathbf{x}_v(0,0) & \mathbf{x}_v(0,1) & \mathbf{x}(0,1) \\ \mathbf{x}_u(0,v) & \mathbf{x}_u(0,0) & \mathbf{x}_{uv}(0,0) & \mathbf{x}_{uv}(0,1) & \mathbf{x}_u(0,1) \\ \mathbf{x}_u(1,v) & \mathbf{x}_u(1,0) & \mathbf{x}_{uv}(0,1) & \mathbf{x}_{uv}(1,1) & \mathbf{x}_u(1,1) \\ \mathbf{x}(1,v) & \mathbf{x}(1,0) & \mathbf{x}_v(1,0) & \mathbf{x}_v(1,1) & \mathbf{x}(1,1) \end{bmatrix} \begin{bmatrix} -1 \\ H_0(v) \\ H_1(v) \\ H_2(v) \\ H_3(v) \end{bmatrix} \tag{2.2.7}$$

2.3 Bounding Volume Hierarchy

Given a freeform surface $\mathbf{x}(u,v)$, we recursively subdivide the surface until the maximum error (from [7]): $\max_{(u,v)} \|\mathbf{x}(u,v) - \mathbf{s}(u,v)\|$ becomes less than for any given error bound. The maximum error can be bounded using

$$\begin{aligned} & \|\mathbf{x}(u,v) - \mathbf{s}(u,v)\| \\ = & \left\| \sum\sum (\mathbf{x}_{ij} - \mathbf{s}_{ij}) B_i(u) B_j(v) \right\| \\ \leq & \sum\sum \|\mathbf{x}_{ij} - \mathbf{s}_{ij}\| B_i(u) B_j(v) \\ \leq & \max \|\mathbf{x}_{ij} - \mathbf{s}_{ij}\| \end{aligned}$$

where $\mathbf{x}_{ij}$ and $\mathbf{s}_{ij}$ are the control points for $\mathbf{x}(u,v)$ and $\mathbf{s}(u,v)$, and $B_i(u)$ and $B_j(v)$ are the basis functions with a partition of unity. ([7] also improved the upperbound estimation from this.)

Note that every one-variable smooth planar curves can be rearranged by the Bézier curves with some sufficient weights (without the control points). It is a kind of degree elevation. Thus

- $H_3(t) = 1 - H_0(t) = \sum_{i=1}^{n} \phi_i B_i^n(t)$ [8]
- $H_1(t) = \sum_{i=0}^{n} \psi_i^1 B_i^n(t)$
- $H_2(t) = \sum_{i=0}^{n} \psi_i^2 B_i^n(t)$

where ϕ_i, ψ_i^1 and ψ_i^2 are the sufficient constant weights, $B_i^n(t)$ is a ith basis element of Bernstein basis.

To estimate distance between control points, we need following theorem:

Theorem 2.3.1. *Suppose the boundary curves of the Bezier surface* $\mathbf{x}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{n}\mathbf{x}_{ij}B_i^n(u)B_j^n(v)$ *and the blending functions are given by* $B_i^n = (1-t)^{n-i}t^i$*. Then the Bicubic Coons patch with the Hermite basis is a Bezier surface*

$$\mathbf{s}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{n}\mathbf{s}_{ij}B_i^n(u)B_j^n(v) \tag{2.3.8}$$

where the control points are

$$\begin{aligned}\mathbf{s}_{ij} = \; & \mathbf{x}_{0j} + \mathbf{x}_{i0} - \mathbf{x}_{00} + \psi_j^1(n(\mathbf{x}_{i1} - \mathbf{x}_{i0}) - n(\mathbf{x}_{01} - \mathbf{x}_{00})) \\ & + \psi_j^2(n(\mathbf{x}_{in} - \mathbf{x}_{i,n-1}) - n(\mathbf{x}_{0n} - \mathbf{x}_{0,n-1})) + \phi_j(\mathbf{x}_{in} - \mathbf{x}_{i0} + \mathbf{x}_{00} - \mathbf{x}_{0n}) \\ & + \psi_i^1(n(\mathbf{x}_{1j} - \mathbf{x}_{0j} - \mathbf{x}_{10} + \mathbf{x}_{00}) - n^2(\mathbf{x}_{11} - \mathbf{x}_{10} - \mathbf{x}_{01} + \mathbf{x}_{00})\psi_j^1 \\ & - n^2(\mathbf{x}_{1n} - \mathbf{x}_{1,n-1} - \mathbf{x}_{0n} + \mathbf{x}_{0,n-1})\psi_j^2 + n(\mathbf{x}_{1n} - \mathbf{x}_{0n} - \mathbf{x}_{10} + \mathbf{x}_{00})\phi_j) \\ & + \psi_i^2(n(\mathbf{x}_{nj} - \mathbf{x}_{n-1,j} - \mathbf{x}_{n0} + \mathbf{x}_{n-1,0}) - n^2(\mathbf{x}_{n1} - \mathbf{x}_{n0} - \mathbf{x}_{n-1,1} + \mathbf{x}_{n-1,0})\psi_j^1 \\ & - n^2(\mathbf{x}_{nn} - \mathbf{x}_{n,n-1} - \mathbf{x}_{n-1,n} + \mathbf{x}_{n-1,n-1})\psi_j^2 + n(\mathbf{x}_{n0} - \mathbf{x}_{n-1,0} - \mathbf{x}_{1n} + \mathbf{x}_{0n})\phi_j) \\ & + \phi_i((\mathbf{x}_{nj} - \mathbf{x}_{0j} - \mathbf{x}_{n0} + \mathbf{x}_{00}) + n(\mathbf{x}_{01} - \mathbf{x}_{00} - \mathbf{x}_{n1} + \mathbf{x}_{n0})\psi_j^1 \\ & + n(\mathbf{x}_{0n} - \mathbf{x}_{0,n-1} - \mathbf{x}_{nn} + \mathbf{x}_{n,n-1})\psi_j^2 + (\mathbf{x}_{0n} - \mathbf{x}_{00} + \mathbf{x}_{n0} - \mathbf{x}_{nn})\phi_j)\end{aligned} \tag{2.3.9}$$

The control point is more complicated than the bilinear Coons patch case, but the BVH from this control points is gives much more accurate result than the bilinear and other boundary blended surface patches. Also, bounding volume can be more narrowed down than the bilinear case. From the setting of the Theorem, we can rearrange the blending function using NURBS. So we can find the control points of Coons patch in a NURBS form or other B-spline surface form. Now, we will propose a subdivision method (we bring the subdivision method in [8] straightforwardly) to achieve low computational cost at the same time and efficient results, we will find the error based on this method.

Remark. Subdivision *[8] A Bezier surface* $\mathbf{x}$ *defined on* $[a, b] \times [c, d]$ *can be subdivided into four Bézier surfaces* A*,* B*,* C*,* D *which are respectively defined on the following subdivided domains:*

$$\begin{aligned} I^A = &\quad [a, 0.5(a+b)] \times [c, 0.5(c+d)] \\ I^B = &\quad [a, 0.5(a+b)] \times [0.5(c+d), d] \\ I^C = &\quad [0.5(a+b), b] \times [c, 0.5(c+d)] \\ I^D = &\quad [0.5(a+b), b] \times [0.5(c+d), d] \end{aligned} \tag{2.3.10}$$

by using the following recursions. Suppose the control points of $\mathbf{x}$ *are* $\mathbf{x}_{ij}$ $i, j = 0, ..., n$*. We define*

$$\begin{cases} \mathbf{x}_{ij}^{[0]} := & \mathbf{x}_{ij} \quad \text{and} \\ \mathbf{x}_{ij}^{[k]} := & 0.5(\mathbf{x}_{i-1,j}^{[k-1]} + \mathbf{x}_{ij}^{[k-1]}) \end{cases} \tag{2.3.11}$$

$$\begin{cases} \mathbf{p}_{ij}^{[0]} := & \mathbf{x}_{ij}^{i} \quad \text{and} \\ \mathbf{p}_{ij}^{[k]} := & 0.5(\mathbf{p}_{i-1,j}^{[k-1]} + \mathbf{p}_{ij}^{[k-1]}) \end{cases} \begin{cases} \mathbf{q}_{ij}^{[0]} := & \mathbf{x}_{nj}^{[n-i]} \quad \text{and} \\ \mathbf{q}_{ij}^{[k]} := & 0.5(\mathbf{q}_{i,j-1}^{[k-1]} + \mathbf{q}_{ij}^{[k-1]}) \end{cases} \tag{2.3.12}$$

The control points of A*,* B*,* C*,* D *are* $A_{ij} = \mathbf{p}_{ij}^{[j]}$*,* $B_{ij} = \mathbf{p}_{in}^{[n-j]}$*,* $C_{ij} = \mathbf{q}_{ij}^{[j]}$*,* $D_{ij} = \mathbf{q}_{in}^{[n-j]}$*, respectively.*

Chapter 3

Experimental Results

We have implemented the proposed surface construction in C++ on an Intel Core i7-6700K 4.2GHz CPU with a 32GB main memory and an NVIDIA GeForce GTX1070. To demonstrate approximation rate of structure, we have also implemented maximum distance computation between the Coons patches and the original Bézier surfaces and the proportion of the previous subdivision level and the subsequent subdivision level.

Table1, 2, 3 report the error analysis of surface approximation using Coons patches for the different levels of a subdivision with 2^{2k} pieces of a surface, for $k = 0, 1, 2, 3, 4$. Also shown are the decay rates(or the approximation rates). This guarantees that the Coons patches are approximated by bicubic surfaces with some suitable approximation error, and consequently the bicubic surfaces approximate the original freeform surface with appropriate errors.

We also include the bilinear case in each table. The decay rate and approximation aspects under a subdivision level is well-known by a square of a decay rate of sides of domain. We can compare errors between a bilinear case and bicubic case easily. Also each table contains its distance computing time for each bilinear approximation case or bicubic approximation case.

3.1 For an arbitrary Bézier surface

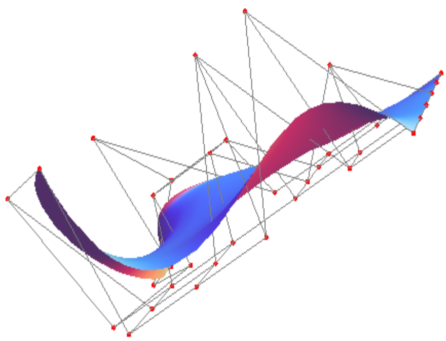

Figure 3.1: Arbitrary Bézier surface on the unit square domain

Table 3.1 shows the one surface construction result for the 1000 arbitrary Bézier surface simulations. Each model was generated randomly with arbitrary control points, but it is much less complicated than other freeform surfaces or NURBS models. Figure 3.1 is one of the sample of a lot of simulations. The second row of Table 3.1 shows the number of subdivision and the number of pieces of those Bézier surfaces in this model. The third row shows the error for the bilinear Coons patch approximation in the corresponding subdivision levels. The fourth row shows the decay rate between the previous subdivision level and the present subdivision level.

The fifth row shows the distance computing time between given geometric model and constructed bilinear Coons patches. The sixth row to eighth row shows the same result for the bicubic Coons patch approximation in the corresponding subdivision levels.

Bézier					
#Subdiv.(Pieces)	0(1)	1(4)	2(16)	3(64)	4(256)
Bilinear Error	2.58053	0.65123	0.17123	0.04101	0.01002
decay rate	-	3.97^{-1}	3.80^{-1}	4.17^{-1}	4.09^{-1}
computing time(sec)	0.0004	0.0015	0.0056	0.0182	0.0585
Bicubic Error	13.8117	1.68323	0.22342	0.02804	0.00360
decay rate	-	8.21^{-1}	7.53^{-1}	7.97^{-1}	7.79^{-1}
computing time(sec)	0.0012	0.0061	0.0331	0.1676	0.7095

Table 3.1: The result of Coons approximation and error estimation for Bézier surface

3.2 For the Utah teapot

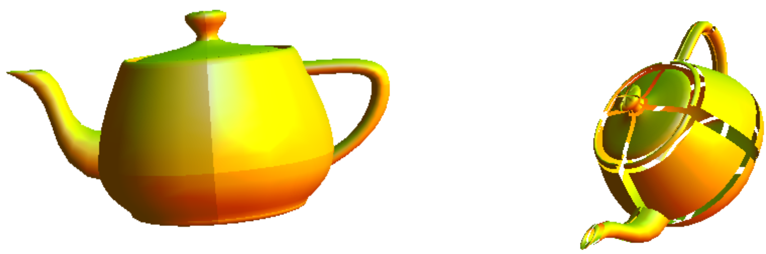

Figure 3.2: The Utah teapot and specific parts of this object

Table 3.2 shows the result for the Utah teapot which is composed by several periodic NURBS surfaces. Each periodic surfaces can be subdivided by two and the total number of NURBS surfaces is shown as eight. Each NURBS surface may contain a different number of piecewise polynomial(especially Bézier) surfaces. Thus also the total number of Bézier surface in this model is 160. We can choose a part of teapot: lid, holder, body, spout, etc., and take a Bézier surface in that specific parts respectively. The second rows shows the number of subdivision and pieces of those Bezier surfaces(also those bicubic Coons surfaces) in the freeform model and the five different levels of subdivision we employ for the Coons approximation. The size of object is measured as a memory space for storing the control points and the knot sequences. The third row reports the approximation error of Coons patch for the corresponding Bézier surfaces in each subdivision level. The fourth rows report the decay rate(or the approximation rate) of error from the previous subdivision level to the present subdivision level. The fifth row shows the distance computing time between given geometric model and constructed bilinear Coons patches. The sixth row to eighth row shows the same result for the bicubic Coons patch approximation in the corresponding subdivision levels.

(a) Lid					
#Subdiv.(Pieces)	0(1)	1(4)	2(16)	3(64)	4(256)
Bilinear Error	0.13247	0.03314	0.00832	0.00210	0.00059
decay rate	-	4.00^{-1}	3.98^{-1}	3.96^{-1}	3.56^{-1}
computing time(sec)	0.0000	0.0001	0.0004	0.0015	0.0058
Bicubic Error	0.15997	0.02000	0.00257	0.00033	0.00004
decay rate	-	8.00^{-1}	7.78^{-1}	7.79^{-1}	8.25^{-1}
computing time(sec)	0.0001	0.0006	0.0032	0.0193	0.0989

(b) Holder					
#Subdiv.(Pieces)	0(1)	1(4)	2(16)	3(64)	4(256)
Bilinear Error	0.19212	0.04901	0.01211	0.00303	0.00080
decay rate	-	3.92^{-1}	4.05^{-1}	4.00^{-1}	3.79^{-1}
computing time(sec)	0.0000	0.0001	0.0005	0.0017	0.0068
Bicubic Error	0.20514	0.02565	0.00321	0.00041	0.00005
decay rate	-	8.00^{-1}	8.00^{-1}	7.83^{-1}	8.20^{-1}
computing time(sec)	0.0001	0.0008	0.0046	0.0231	0.1308

(c) Body					
#Subdiv.(Pieces)	0(1)	1(4)	2(16)	3(64)	4(256)
Bilinear Error	0.40010	0.10030	0.02570	0.00630	0.00171
decay rate	-	3.99^{-1}	3.90^{-1}	4.07^{-1}	43.68^{-1}
computing time(sec)	0.0001	0.0003	0.0010	0.0042	0.0137
Bicubic Error	0.40123	0.05012	0.00630	0.00079	0.00010
decay rate	-	8.01^{-1}	7.96^{-1}	7.97^{-1}	7.90^{-1}
computing time(sec)	0.0002	0.0011	0.0061	0.0350	0.2061

(d) Spout					
#Subdiv.(Pieces)	0(1)	1(4)	2(16)	3(64)	4(256)
Bilinear Error	0.64825	0.0.16306	0.04112	0.01102	0.00281
decay rate	-	3.98^{-1}	3.97^{-1}	3.73^{-1}	3.92^{-1}
computing time(sec)	0.0002	0.0008	0.0030	0.0121	0.0418
Bicubic Error	0.80279	0.10035	0.01260	0.00156	0.00020
decay rate	-	8.00^{-1}	7.94^{-1}	8.08^{-1}	7.80^{-1}
computing time(sec)	0.0003	0.0018	0.00105	0.0575	0.3372

Table 3.2: The results of Coons approximation and error estimation for parts of Utah teapots

3.3 For the Stanford Bunny

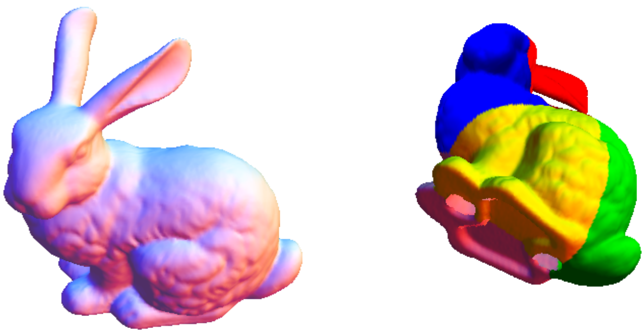

Figure 3.3: The Stanford bunny and the five specific parts of this object

Table 3.3 reports the result for the Stanford bunny model shown in Figure 3.3. This model has many prominences and depressions in the body as the NURBS representation has multiple knots. The initial NURBS surfaces are also subdivided along the ears, paws, tail, face and the periodic NURBS surfaces are also subdivided in the middle. The total number of NURBS surfaces after the subdivision is shown in the caption of Table 3.3, together with the total number of Bézier surfaces. Table 3.3(a) is the result for the ers of Standard bunny. Table 3.3(b) is the result for the paws for the Stanford bunny. And the rests (Table 3.3(c),(d),(e)) are the results for the tail, body and face, respectively. The error in ears(between Coons and Bézier) has smallest value because it is similar to the part of cylinder. Also, the error in body(between Coons and Bézier) is largest because of prominences and depressions in the body.

(a) Ears					
#Subdiv.(Pieces)	0(1)	1(4)	2(16)	3(64)	4(256)
Bilinear Error	1.17315	0.30318	0.07402	0.01890	0.00510
decay rate	-	3.87^{-1}	4.10^{-1}	3.92^{-1}	3.71^{-1}
computing time(sec)	0.0002	0.0010	0.0035	0.0132	0.0484
Biucbic Error	1.25794	0.15984	0.01999	0.00250	0.00033
decay rate	-	7.87^{-1}	8.00^{-1}	8.00^{-1}	7.58^{-1}
computing time(sec)	0.0003	0.0015	0.0093	0.0557	0.3899

(b) Paws					
#Subdiv.(Pieces)	0(1)	1(4)	2(16)	3(64)	4(256)
Bilinear Error	3.71229	0.92910	0.0.24100	0.05912	0.01620
decay rate	-	4.00^{-1}	3.86^{-1}	4.08^{-1}	3.65^{-1}
computing time(sec)	0.0002	0.0008	0.0032	0.0125	0.0464
Bicubic Error	3.69887	0.47115	0.05879	0.00744	0.00096
decay rate	-	7.85^{-1}	8.01^{-1}	7.90^{-1}	7.75^{-1}
computing time(sec)	0.0004	0.0021	0.0119	0.0714	0.4286

(c) Tail					
#Subdiv.(Pieces)	0(1)	1(4)	2(16)	3(64)	4(256)
Bilinear Error	4.00101	1.00121	0.26017	0.06394	0.01698
decay rate	-	4.00^{-1}	3.84^{-1}	4.07^{-1}	3.77^{-1}
computing time(sec)	0.0002	0.0009	0.0036	0.0136	0.0533
Bicubic Error	3.96772	0.49606	0.06201	0.00781	0.00098
decay rate	-	8.00^{-1}	8.00^{-1}	7.94^{-1}	7.97^{-1}
computing time(sec)	0.0005	0.0027	0.0149	0.0906	0.4882

(d) Body					
#Subdiv.(Pieces)	0(1)	1(4)	2(16)	3(64)	4(256)
Bilinear Error	4.11087	1.02672	0.25790	0.06487	0.01712
decay rate	-	4.00^{-1}	3.98^{-1}	3.98^{-1}	3.79^{-1}
computing time(sec)	0.0004	0.0013	0.0044	0.0166	0.0626
Bicubic Error	4.13298	0.51592	0.06458	0.00818	0.00103
decay rate	-	8.01^{-1}	7.99^{-1}	7.89^{-1}	7.94^{-1}
computing time(sec)	0.0007	0.0045	0.0235	0.1253	0.6401

(e) Face					
#Subdiv.(Pieces)	0(1)	1(4)	2(16)	3(64)	4(256)
Bilinear Error	3.72112	0.93130	0.24315	0.06021	0.01478
decay rate	-	4.00^{-1}	3.83^{-1}	4.04^{-1}	4.07^{-1}
computing time(sec)	0.0003	0.0013	0.0047	0.0178	0.0639
Bicubic Error	3.82711	0.47849	0.06001	0.00751	0.00094
decay rate	-	8.00^{-1}	7.97^{-1}	7.99^{-1}	7.99^{-1}
computing time(sec)	0.0005	0.0029	0.0170	0.0876	0.5183

Table 3.3: The results of Coons approximation and error estimation for parts of Stanford Bunny(with 251 NURBS & 1038 Bezier Surfaces)

3.4 Error analysis and Property for the decay rate

For the bilinear case, Wang [9] and Filip et al.[5] stated the theorem about the decay rate for bilinear Coons patch approximation in the subdivision level:

Theorem 3.4.1 ([5]). *Let $T \subset \mathbb{R}^2$ be a right triangle with vertices $\{A, B, C\}$ of the form $B = A + (r_1, 0)$ and $C = A + (0, r_2)$. Let $\mathbf{x} : T \to \mathbb{R}^3$ be any C^2 surface, and let $\mathbf{s}(u, v)$ be the linearly parametrised triangle with $\mathbf{s}(A) = \mathbf{x}(A)$, $\mathbf{s}(B) = \mathbf{x}(B)$, and $\mathbf{s}(C) = \mathbf{x}(C)$. Then*

$$\sup_{(u,v)\in T} \|\mathbf{x}(u, v) - \mathbf{s}(u, v)\| \leq \frac{1}{8}(r_1^2 M_1 + 2r_1 r_2 M_2 + r_2^2 M_3) \tag{3.4.1}$$

where

$$\begin{aligned} M_1 = & \sup_{(u,v)\in T} \left\| \frac{\partial^2 \mathbf{x}(u, v)}{\partial u^2} \right\| \\ M_2 = & \sup_{(u,v)\in T} \left\| \frac{\partial^2 \mathbf{x}(u, v)}{\partial u \partial v} \right\| \\ M_3 = & \sup_{(u,v)\in T} \left\| \frac{\partial^2 \mathbf{x}(u, v)}{\partial v^2} \right\| \end{aligned} \tag{3.4.2}$$

This theorem applies for a a triangular mesh case. This can be extended in the rectangular domain case following as known as Wang's Theorem from [5]:

Theorem 3.4.2 (Wang's Theorem [9]). *Let $R \subset \mathbb{R}^2$ be the rectangular region with vertices $\{A, B, C, D\}$ of the form $B = A + (r_1, 0)$, $C = A + (0, r_2)$ and $D = A + (r_1, r_2)$. Let $\mathbf{x} : R \to \mathbb{R}^3$ be any C^2 function, and let $\mathbf{s}(u, v)$ be the linearly parametrised quadrilateral with $\mathbf{s}(A) = \mathbf{x}(A) - \Delta$, $\mathbf{s}(B) = \mathbf{x}(B) + \Delta$, $\mathbf{s}(C) = \mathbf{x}(C) + \Delta$, and $\mathbf{s}(D) = \mathbf{x}(D) - \Delta$, where $\Delta =$*

$\frac{1}{4}(\mathbf{x}(A) - \mathbf{x}(B) - \mathbf{x}(C) + \mathbf{x}(D))$. *Then*

$$\sup_{(u,v)\in T} \|\mathbf{x}(u,v) - \mathbf{s}(u,v)\| \leq \frac{\sqrt{3}}{8}(r_1^2 M_1 + 2r_1 r_2 M_2 + r_2^2 M_3) \tag{3.4.3}$$

where

$$\begin{aligned} M_1 &= \max\left(\left\|\frac{\partial^2 x_1}{\partial u^2}\right\|, \left\|\frac{\partial^2 x_2}{\partial u^2}\right\|, \left\|\frac{\partial^2 x_3}{\partial u^2}\right\|\right) \\ M_2 &= \max\left(\left\|\frac{\partial^2 x_1}{\partial u \partial v}\right\|, \left\|\frac{\partial^2 x_2}{\partial u \partial v}\right\|, \left\|\frac{\partial^2 x_3}{\partial u \partial v}\right\|\right) \\ M_3 &= \max\left(\left\|\frac{\partial^2 x_1}{\partial v^2}\right\|, \left\|\frac{\partial^2 x_2}{\partial v^2}\right\|, \left\|\frac{\partial^2 x_3}{\partial v^2}\right\|\right) \end{aligned} \tag{3.4.4}$$

We also have a experiment about freeform surfaces with another subdivision. That was dividing the domain only vertically or only horizontally and compared the errors between the previous subdivision level and the present subdivision level. In this case, we could not find the significant approximation aspects(the approximation rate or the decay rate). With the Theorem 3.4.1 and 3.4.2, so we can present the following assumption:

Statement (Assumption to Bicubic case)**.** *Let $R \in \mathbb{R}$ be the rectangular region with vertices $A = (a, b)$, $B = A + (r_1, 0)$, $C = A + (0, r_2)$, and $D = A + (r_1, r_2)$ where $(a, b) \in \mathbb{R}^2$, r_1, r_2 are the positive constants. Let $\mathbf{x} : R \to \mathbb{R}^3$ ne any C^3 function and let $\mathbf{s}(u, v)$ be the bicubically parametrised quadrilateral with*

$$\begin{aligned} \mathbf{x}(u,v) &= \mathbf{s}(u,v) \\ \mathbf{x}_u(u,v) &= \mathbf{s}_u(u,v) \\ \mathbf{x}_v(u,v) &= \mathbf{s}_v(u,v) \end{aligned}$$

on the $\overline{R} - \mathrm{int}R$, where $\mathbf{x}_u$ and $\mathbf{x}_v$ are the partial derivatives regarding u and v, respectively. Then

$$\sup_{(u,v)\in R} \|\mathbf{x}(u,v) - \mathbf{s}(u,v)\| \leq C(r_1^3 M_1 + 3r_1^2 r_2 M_2 + 3r_1 r_2^2 M_3 + r_2^3 M_4) \tag{3.4.5}$$

where C is a proper constant and M_1, M_2, M_3 and M_4 are the constants depend on the third partial derivatives of $\mathbf{x}$.

Chapter 4

Conclusion

We have presented a bicubic Coons surface represetation for a freeform rational/polynomial(especialy Bézier) surfaces. These surfaces have information of first derivatives of boundary curves and it interpolate all boundaries exactly. Then the error between a Bézier surface and corresponding bicubic Coons patches reduce in the cubic proportion to a reduced rate of four sides after the subdivision. In this respect, these experiments gives us the hypothesis in the section 3.4.

Also we get another fact. That is the distance between Bézier surface and corresponding bicubically blended Coons patch is approximately equal to the distance between those in the midpoint of domain. If this fact can be applied to all Bezier surfaces, then this fact contribute to reduce computation cost and narrowed down the scope of the inquiry in the surface approximation. In future work, we plan to demonstrate the conjecture and improve the performance of a series of geometric algorithms by developing new geometric techniques with acceleration.

Bibliography

[1] G. Aumann, Degree elevation and developable Bézier surfaces *Computer Aided Geometric Design 21*, 661-670, 2004.

[2] C. de Boor, *A Practical Guid to Splines:Reviced Edition.* 1978, Springer.

[3] E. Cohen, L. L. Schumaker, Rate of Convergence of Control Polygon *Computer Aided Geometric Design 2* 1985.

[4] S. A Coons, Surfaces for Computer-Aided Design *Technical report MIT*, 1964

[5] D. Filip, R. Magedson, R. Markot, Surface algorithms using bounds on derivatives *Computer Aided Geometric Design 3*, 295-311, 1986.

[6] C. Hall, Optimal Error Bounds for Cubic Spline Interpolation *Journal of Approximation Theory 16*, 105-122, 1976.

[7] Y. Kim, Y. Oh, S. Yoon, M. Kim, G. Elber, Coons BVH for Freeform Geometric Models *Computer Graphics(SIGGRAPH2010)*, 2011.

[8] M. Randrianarivony, G. Brunnett, Sufficient and necessary conditions for the regularity of planar Coons maps , 2004.

[9] G. Wang, The subdivision method for finding the intersection between two Bézier curves or surfaces, *Zhejiang University* 1984

국 문 초 록

우리는 자유곡면의 형태를 띈 비균일 유리 B-스플라인 곡면을 쌍삼차로 혼합된 쿤스 패치를 사용하여 근사하였다. 이러한 쿤스 곡면의 설계에 있어서, 모든 부분 패치들은 네 모서리로부터 정의된 쌍삼차 곡면을 사용하여 매우 효율적으로 유계시킬 수 있다. 많은 기하학적 알고리즘 등에서 사용되는 용적 제한 체계를 쓰기 위해서는, 제어점의 명확한 값을 아는것이 필수적이다. 또한, 근사를 하는데 있어서 원래 곡면을 정의역에서 분할할 때에 분할된 정의역의 조각의 크기와 자유곡면 근사에서의 근사율의 크기(또는 근사속도)가 서로 상관관계가 있다. 그래서 우리는 쌍삼차로 혼합된 쿤스 패치의 베지에 변환으로 이것의 제어점을 구하는 방법을 제시하였고, 분할 전과 분할 후 쌍삼차 쿤스 패치로 근사한 것들의 근사율의 크기차이(또는 근사 속도)에 대하여 이전보다 구체화된 가설을 제시하였다.

주요어휘 : 자유곡면, 비균일 기저 스플라인 곡면, 쿤스 패치, 베지에 곡면, 제어점, 곡면 근사

학번 : 2015-20265