PII: S0377-0427(11)00224-X
DOI: 10.1016/j.cam.2011.04.030
Reference: CAM 8334

To appear in: *Journal of Computational and Applied Mathematics*

Received date: 16 April 2010
Revised date: 28 March 2011

Triangular Bézier sub-surfaces on a triangular Bézier surface

Wenyu Chen[1], Rongdong Yu[2], Jianmin Zheng[1,3], Yiyu Cai[3,4], Chikit Au[5]

[1] *School of Computer Engineering, Nanyang Technological University, Singapore*
[2] *College of Computer Science, Zhejiang University, China*
[3] *Institute for Media Innovation, Nanyang Technological University, Singapore*
[4] *School of Mechanical and Aerospace, Nanyang Technological University, Singapore*
[5] *Department of Engineering, The University of Waikato, New Zealand*

Abstract

This paper considers the problem of computing the Bézier representation for a triangular sub-patch on a triangular Bézier surface. The triangular sub-patch is defined as a composition of the triangular surface and a domain surface that is also a triangular Bézier patch. Based on de Casteljau recursions and shifting operators, previous methods express the control points of the triangular sub-patch as linear combinations of the construction points that are constructed from the control points of the triangular Bézier surface. The construction points contain too many redundancies. This paper derives a simple explicit formula that computes the composite triangular sub-patch in terms of the blossoming points that correspond to distinct construction points and then an efficient algorithm is presented to calculate the control points of the sub-patch.

Keywords: Composition; Sub-patches; Bézier representation; Triangular surfaces; de Casteljau algorithm; blossoming.

1. Introduction

Many applications in CAD/CAM or computer graphics industry require creating geometric entities such as curves or patches on surfaces. Isoparametric curves on a surface are easy to derive. In many cases, however, the curves need to be in a general position such as the intersection curve of two surfaces, the boundary for surface trimming. DeRose [1] examined the curves on triangular Bézier surfaces via functional composition. Jüttler and Wang [2] analyzed the curves on a sphere. Both approaches generate curves on surfaces by parameter space representation.

Beside curves, sub-patches on surfaces are also important. Two types of surfaces are widely used: triangular Bernstein-Bézier surface (TB or TBB surface) and tensor product Bézier surface (TP or TPB surface). For instance, the subdivision of a Bézier surface [3, 4, 5, 6, 7] falls into this category and so do the conversions between TB surfaces and TP surfaces. Brueckner [8] represented a TB surface as a trimmed TP surface. Waggenspack and Anderson [9] transformed a TP surface to a TB representation. Jie [10] extended the equations to rational cases. In most cases, the sub-patches do not have isoparametric boundary curves. For example, the explicit formula of Goldman and Filip [11] converted a TP surface of degree (m, n) into two TB surfaces of degree $m + n$. Hu [12] developed a method to divide a TB surface into three TP surfaces. In another way, Sheng and Hirsch [13] divided a trimmed surface into many TB surfaces.

Subdivision, reparametrization and surface extensions are possible applications of composition [1]. Both blossoming and product methods can be used to obtain the composition of a TB/PB and a TP/PB [14]. However, as pointed out by DeRose [14], the product algorithm was more efficient for machine implementation whereas the blossom algorithm was geometrically more intuitive. There are four different compositions: TP over TP,TP over TB, TB over TP and TB over TB. DeRose [14] used blossoming algorithm to study the four compositions. However, the algorithm for the control points of the compositions is not sufficiently efficient in practice. Lasser [15, 16] studied the composition of TP over TB, and the composition of TB over TP. Explicit formulae are provided for the control points of the compositions. Feng and Peng [17] considered a simpler case using shifting operator to derive the composition of a triangle with a TB surface.

Lasser [15, 16] formulated the control points of the composition as the linear combinations of some intermediate points called the construction points. However, the number of the construction points is huge and many of them actually have the same positions. In this paper, we provide a more compact formula to compute the control points of the composition and detailed algorithms are presented for practical uses.

2. Preliminaries and notations

A TB surface [18] $\mathbf{T}(u,v,w)$ of degree n can be defined by

$$\mathbf{T}(u,v,w)=\sum_{i+j+k=n} B_{ijk}^{n}(u,v,w)\mathbf{T}_{ijk}, \quad (u,v,w)\in D_T. \tag{1}$$

where $\mathbf{T}_{ijk}\in R^3$ are the control points, $B_{ijk}^{n}(u,v,w)$ are Bernstein polynomials

$$B_{ijk}^{n}(u,v,w)=\frac{n!}{i!j!k!}u^i v^j w^k, \quad i+j+k=n, \quad u,v,w\geq 0, \quad u+v+w=1,$$

and $D_T=\{(u,v,w)|\, u+v+w=1, u,v,w\in[0,1]\}$ is a triangle domain.

A domain surface $\mathbf{P}(u,v,w)$ of degree m is a TB surface

$$\mathbf{P}(u,v,w)=\sum_{i+j+k=m} B_{ijk}^{m}(u,v,w)\mathbf{P}_{ijk}, \quad u,v,w\geq 0, \quad u+v+w=1 \tag{2}$$

where the control points $\mathbf{P}_{ijk}\in D_T$ are parameter points.

A hyper-index $\mathbf{\Gamma}_n^m$ is defined as

$$\begin{array}{l}\mathbf{\Gamma}_n^m=(\mathbf{I}_n^m,\mathbf{J}_n^m,\mathbf{K}_n^m)\,,\\ \mathbf{I}_n^m=(I_1,\cdots,I_n),\mathbf{J}_n^m=(J_1,\cdots,J_n),\mathbf{K}_n^m=(K_1,\cdots,K_n),\\ I_l+J_l+K_l=m, \qquad I_l,J_l,K_l\in\{0,\cdots,m\}.\end{array} \tag{3}$$

The norm for the hyper-index is

$$\begin{array}{l}|\mathbf{\Gamma}_n^m|=|\mathbf{I}_n^m|+|\mathbf{J}_n^m|+|\mathbf{K}_n^m|=mn,\\ |\mathbf{I}_n^m|=\sum\limits_{l=1}^{n} I_l, |\mathbf{J}_n^m|=\sum\limits_{l=1}^{n} J_l, |\mathbf{K}_n^m|=\sum\limits_{l=1}^{n} K_l.\end{array}$$

An index (I_l,J_l,K_l) corresponds to a parameter point $\mathbf{P}_{I_lJ_lK_l}$ in Eq.2. Thus, $\mathbf{\Gamma}_n^m$, with n indices, corresponds to a parameter vector $\mathbf{P}_{\mathbf{\Gamma}_n^m}^n$ with n parameter points:

$$\mathbf{P}_{\mathbf{\Gamma}_n^m}^n=\mathbf{P}_{\mathbf{I}_n^m\mathbf{J}_n^m\mathbf{K}_n^m}^n=\left((u_{I_1J_1K_1},v_{I_1J_1K_1},w_{I_1J_1K_1}),\cdots,(u_{I_nJ_nK_n},v_{I_nJ_nK_n},w_{I_nJ_nK_n})\right)$$

where $(u_{I_iJ_iK_i},v_{I_iJ_iK_i},w_{I_iJ_iK_i})\in D_T, \quad i=1,\cdots,n$. A sub-vector $\mathbf{P}_{\mathbf{\Gamma}_s^m}^s, s=0,1,\cdots,n$ can be obtained satisfying

$$\mathbf{P}_{\mathbf{\Gamma}_s^m}^s=\mathbf{P}_{\mathbf{I}_s^m\mathbf{J}_s^m\mathbf{K}_s^m}^s=((u_{I_1J_1K_1},v_{I_1J_1K_1},w_{I_1J_1K_1}),\cdots,(u_{I_sJ_sK_s},v_{I_sJ_sK_s},w_{I_sJ_sK_s})).$$

The hyper-index can be used for the product of n Bernstein polynomials of degree m:

$$\prod_{l=1}^{n} B^{m}_{I_l J_l K_l}(u,v,w) = C^{mn}_{\mathbf{I}^m_n \mathbf{J}^m_n \mathbf{K}^m_n} B^{mn}_{|\mathbf{I}^m_n|,|\mathbf{J}^m_n|,|\mathbf{K}^m_n|}(u,v,w). \tag{4}$$

where

$$C^{mn}_{\mathbf{\Gamma}^m_n} = C^{mn}_{\mathbf{I}^m_n \mathbf{J}^m_n \mathbf{K}^m_n} = \left(\prod_{l=1}^{n} \frac{m!}{I_l! J_l! K_l!}\right) \Big/ \left(\frac{(mn)!}{|\mathbf{I}^m_n|!\,|\mathbf{J}^m_n|!\,|\mathbf{K}^m_n|!}\right).$$

Lemma 1. *Suppose $R+S+T=mn$. Then*

$$\sum_{|\mathbf{I}^m_n|=R,|\mathbf{J}^m_n|=S,|\mathbf{K}^m_n|=T} C^{mn}_{\mathbf{I}^m_n \mathbf{J}^m_n \mathbf{K}^m_n} = 1. \tag{5}$$

Proof: Suppose we have mn different balls and put them into 3 different boxes B_1, B_2 andB_3. There are $\frac{(mn)!}{R!S!T!}$ different cases for B_1 containing R balls, B_2 containing S balls, and B_3 containing T balls with $R+S+T=mn$.

On the other hand, we divide the mn balls into n groups $G_1,...,G_n$ such that each group G_l contains m balls. If bos B_1 contains I_l balls from G_l, box B_2 contains J_l balls from G_l and box B_3 contains K_l balls from G_l, then the numbers of balls in B_1, B_2, B_3 are $|\mathbf{I}^m_n| = R, |\mathbf{J}^m_n| = S, |\mathbf{K}^m_n| = T$ where $\mathbf{I}^m_n = (I_1,\cdots,I_n), \mathbf{J}^m_n = (J_1,\cdots,J_n), \mathbf{K}^m_n = (K_1,\cdots,K_n), I_l+J_l+K_l = m$. Note that there are $\frac{m!}{I_l!J_l!K_l!}$ different cases to distribute m balls in G_l into the three boxes. Therefore, for a given $(\mathbf{I}^m_n, \mathbf{J}^m_n, \mathbf{K}^m_n)$, there are $\prod_{l=1}^{n} \frac{m!}{I_l!J_l!K_l!}$ different cases to put mn different balls into B_1, B_2, B_3. Hence, we get

$$\sum_{|\mathbf{I}^m_n|=R,|\mathbf{J}^m_n|=S,|\mathbf{K}^m_n|=T} \left(\prod_{l=1}^{n} \frac{m!}{I_l!J_l!K_l!}\right) = \frac{(mn)!}{R!S!T!}.$$

This is equivalent to Eq.5.

It is easy to prove the following equality:

$$\left(\sum_{i+j+k=m} x_{ijk}\right)^n = \sum_{R+S+T=mn} \left(\sum_{|\mathbf{I}^m_n|=R,|\mathbf{J}^m_n|=S,|\mathbf{K}^m_n|=T} \prod_{l=1}^{n} x_{I_l J_l K_l}\right) \tag{6}$$

The TB surface can be rewritten using shifting operators [19]

$$\mathbf{T}(u,v,w) = (uE_1 + vE_2 + wE_3)^n \mathbf{T}_{000} \tag{7}$$

where the shifting operators are

$$E_1\mathbf{T}_{ijk} = \mathbf{T}_{i+1,j,k}, E_2\mathbf{T}_{ijk} = \mathbf{T}_{i,j+1,k}, E_3\mathbf{T}_{ijk} = \mathbf{T}_{i,j,k+1}.$$

With $\mathbf{T}^0_{ijk} = \mathbf{T}_{ijk}$, given a parameter vector $\mathbf{P}^n_{\mathbf{I}^m_n\mathbf{J}^m_n\mathbf{K}^m_n}$ for a TB surface, the de Casteljau algorithm [20] yields

$$\begin{aligned} &\mathbf{T}^n_{ijk}(\mathbf{P}^n_{\mathbf{I}^m_n\mathbf{J}^m_n\mathbf{K}^m_n}) \\ &= u_{I_nJ_nK_n}\mathbf{T}^{n-1}_{i+1,j,k}(\mathbf{P}^{n-1}_{\mathbf{I}^m_{n-1}\mathbf{J}^m_{n-1}\mathbf{K}^m_{n-1}}) + v_{I_nJ_nK_n}\mathbf{T}^{n-1}_{i,j+1,k}(\mathbf{P}^{n-1}_{\mathbf{I}^m_{n-1}\mathbf{J}^m_{n-1}\mathbf{K}^m_{n-1}}) \\ &\quad + w_{I_nJ_nK_n}\mathbf{T}^{n-1}_{i,j,k+1}(\mathbf{P}^{n-1}_{\mathbf{I}^m_{n-1}\mathbf{J}^m_{n-1}\mathbf{K}^m_{n-1}}). \end{aligned} \tag{8}$$

Alternatively, Eq.8 can be rewritten using shifting operators

$$\begin{aligned} &\mathbf{T}^n_{ijk}(\mathbf{P}^n_{\mathbf{I}^m_n\mathbf{J}^m_n\mathbf{K}^m_n}) \\ &= (u_{I_nJ_nK_n}E_1 + v_{I_nJ_nK_n}E_2 + w_{I_nJ_nK_n}E_3)\,\mathbf{T}^{n-1}_{ijk}(\mathbf{P}^{n-1}_{\mathbf{I}^m_{n-1}\mathbf{J}^m_{n-1}\mathbf{K}^m_{n-1}}) \\ &= \prod_{l=1}^{n}(u_{I_lJ_lK_l}E_1 + v_{I_lJ_lK_l}E_2 + w_{I_lJ_lK_l}E_3)\mathrm{T}_{ijk}. \end{aligned} \tag{9}$$

Lemma 2. *The number of different choices of the d variables $(N_1, ..., N_d)$ satisfying $\sum_{i=1}^{d} N_i = s$ is*

$$\left(\begin{array}{c} s+d-1 \\ d-1 \end{array}\right) = \left(\begin{array}{c} s+d-1 \\ s \end{array}\right).$$

Proof: Consider the problem of putting s balls into d boxes. Place the s balls in a line and use $d-1$ bars to separate these balls. There are $s+d-1$ positions for balls and bars. Select $d-1$ positions for the bars. Then put all balls to the left positions. The s balls are separated into d sets. Hence, we have $\left(\begin{array}{c} s+d-1 \\ d-1 \end{array}\right)$ choices.

Let $\mathbf{M}^s_d \subset \mathbb{Z}^d$ be the following set

$$\mathbf{M}^s_d = \left\{ \mathbf{M} |\, \mathbf{M} = (M_1, M_2, \cdots, M_d)\,, \sum_{i=1}^{d} M_i = s, M_i = 0, 1, \cdots, s \right\}.$$

If $\mathbf{M} \in \mathbf{M}^s_d \subset \mathbb{Z}^d$, $\mathbf{M}$ is a $1 \times d$ vector and the summation of its elements is s. According to Lemma 2, there are $\left(\begin{array}{c} d-1+s \\ s \end{array}\right)$ elements in $\mathbf{M}^s_d$.

3. Triangle sub-patch from a triangle surface

Consider a TB surface $\mathbf{T}(x, y, z)$ whose parameter domain is a triangle D_T. A sub-patch is derived from the TB surface by limiting the parameter domain to an area within D_T. This section presents a simple way to express a class of sub-patches as a new TB surface.

3.1. Domain surface

On domain D_T, three Bézier curves $\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3$ form a closed sub-domain $D_{\mathbf{C}_1,\mathbf{C}_2,\mathbf{C}_3}$ (Figure 1(a)). Assume that the three boundary curves $\mathbf{C}_1(t), \mathbf{C}_2(t)$ and $\mathbf{C}_3(t)$ are of degrees n_1, n_2 and n_3, respectively. Let $m = \max(n_1, n_2, n_3)$. We can make the degrees of $\mathbf{C}_1(t), \mathbf{C}_2(t), \mathbf{C}_3(t)$ be m using degree elevation. Denote the control points of $\mathbf{C}_i(t)\,(i = 1, 2, 3)$ by $\mathbf{P}_{i,j}, j = 0, ..., m$. For each $\mathbf{P}_{i,j}$, it has a corresponding point $\mathbf{Q}_{i,j}$ on the curve $\mathbf{C}_i(t)$:

$$\mathbf{Q}_{i,j} = \mathbf{C}_i(\frac{j}{m}) \tag{10}$$

which we call an influence point (see Figure 1(b)).

Re-label the control points $\mathbf{P}_{i,j}$ such that they represent the control points of the boundary curves of a triangular Bézier patch:

$$\mathbf{P}_{ijk}, \min(i, j, k) = 0. \tag{11}$$

The corresponding influence points are also re-labeled accordingly: $\mathbf{Q}_{ijk} \in D_T$.

To define a domain surface for the area $D_{\mathbf{C}_1,\mathbf{C}_2,\mathbf{C}_3}$, we need to define some interior control points. The interior control points $\mathbf{P}_{ijk}, \min(i, j, k) \neq 0$ can be specified using $\mathbf{Q}_{ijk}$ (see Figure 1(c)):

$$\begin{aligned}\mathbf{P}_{ijk} = {} & \tfrac{1}{3}\left(\tfrac{k}{m-i}\mathbf{Q}_{i,0,m-i} + \tfrac{j}{m-i}\mathbf{Q}_{i,m-i,0}\right) + \tfrac{1}{3}\left(\tfrac{k}{m-j}\mathbf{Q}_{0,j,m-j} + \tfrac{i}{m-j}\mathbf{Q}_{m-j,j,0}\right) \\ & + \tfrac{1}{3}\left(\tfrac{j}{m-k}\mathbf{Q}_{0,m-k,k} + \tfrac{i}{m-k}\mathbf{Q}_{m-k,0,k}\right),\end{aligned} \tag{12}$$

Thus Eq.11 and Eq.12 define the control points or parameter points for a domain surface that is a a TB surface in the form of Eq.2 (see Figure 1(d)).

It can be seen that the interior control points are linear combinations of the influence points. The combination involves six influence points along the u, v, w directions (Figure 2). The number of interior control points is $(m-1)(m-2)/2$. If $m = 3$, for example, there is only one interior control point:

$$\mathbf{P}_{111} = (\mathbf{Q}_{102} + \mathbf{Q}_{120} + \mathbf{Q}_{012} + \mathbf{Q}_{210} + \mathbf{Q}_{201} + \mathbf{Q}_{021})/6.$$

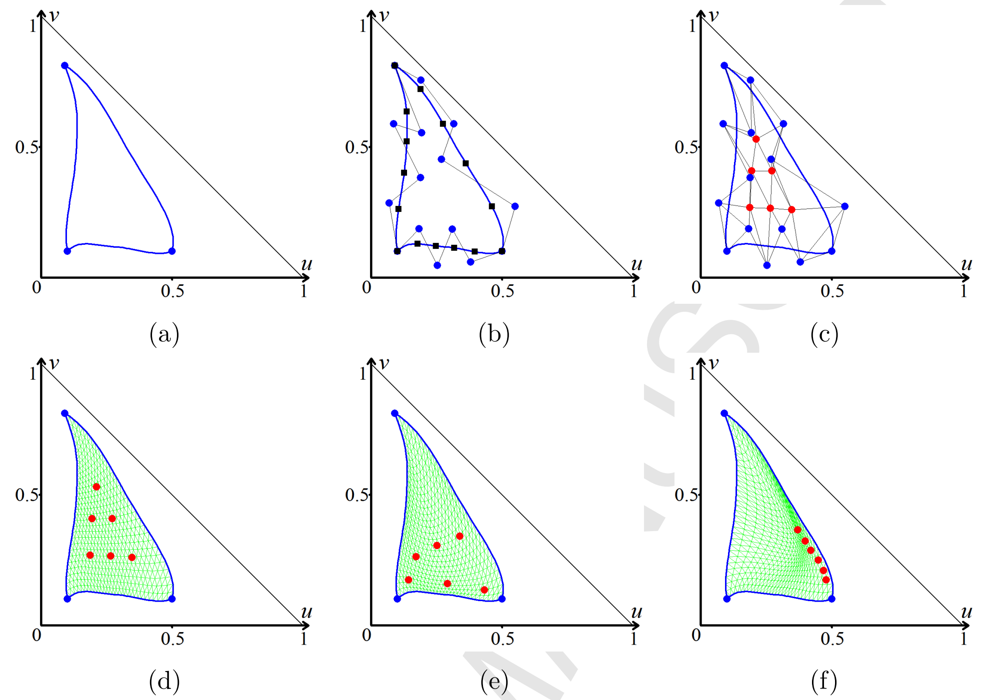


Figure 1: Domain surface: (a) three boundary curves form a closed sub-domain; (b) the control points (blue dots) and the influence points (black squares); (c) the interior control points (red dots); and (d-f) different parameter curves by different interior control points.

For $m = 1, 2$, there is no interior control point.

It is worth pointing out that above we have just provided a way to construct interior control points. Users may also choose to modify them interactively. Figure 1(e) and Figure 1(f) show two different choices of the interior control points. Different choices of the interior control points may lead to different parameterization of the domain surface and hence affect the parameterization of the composite surface.

3.2. Sub-patch via composition

If $\mathbf{P}(u, v, w)$ is a surface on the domain of a TB surface $\mathbf{T}(x, y, z)$, the composition $\mathbf{S}(u, v, w) = \mathbf{T}(\mathbf{P}(u, v, w))$ is a sub-surface of the TB surface $\mathbf{T}(x, y, z)$ (Figure 3). Moreover, we have

Theorem 1. *(Composition of two TB surfaces). Suppose* $\mathbf{T}(x, y, z)$ *is a TB surface of degree* n *with control points* $\mathbf{T}_{ijk} \in R^3, i+j+k = n$, *and* $\mathbf{P}(u, v, w)$

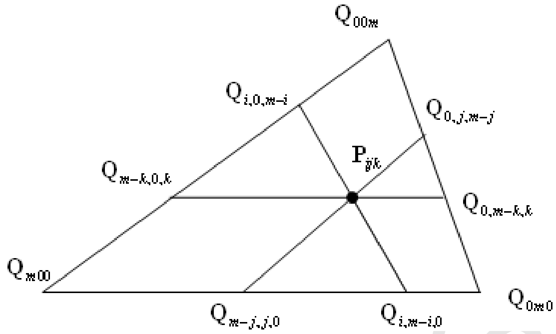


Figure 2: Construction of an interior control point.

is a domain surface of degree m with parameter points

$$\mathbf{P}_{ijk} = (x_{ijk}, y_{ijk}, z_{ijk}), i + j + k = m, x_{ijk} + y_{ijk} + z_{ijk} = 1.$$

Then the composition $\mathbf{S}(u, v, w) = \mathbf{T}(\mathbf{P}(u, v, w))$ is a TB surface of degree mn:

$$\mathbf{S}(u, v, w) = \sum_{R+S+T=mn} B^{mn}_{RST}(u, v, w)\mathbf{S}_{RST} \tag{13}$$

with control points

$$\mathbf{S}_{RST} = \sum_{|\mathbf{I}^m_n|=R, |\mathbf{J}^m_n|=S, |\mathbf{K}^m_n|=T} C^{mn}_{\mathbf{I}^m_n \mathbf{J}^m_n \mathbf{K}^m_n} \mathbf{S}^n_{\mathbf{I}^m_n \mathbf{J}^m_n \mathbf{K}^m_n} \tag{14}$$

where $\mathbf{S}^n_{\mathbf{I}^m_n \mathbf{J}^m_n \mathbf{K}^m_n} = \mathbf{T}^n_{000}(\mathbf{P}^n_{\mathbf{I}^m_n \mathbf{J}^m_n \mathbf{K}^m_n})$ are construction points corresponding to parameter vectors

$$\mathbf{P}^n_{\mathbf{I}^m_n \mathbf{J}^m_n \mathbf{K}^m_n} = [\mathbf{P}_{I_l J_l K_l} \mid \ l = 1, \cdots, n]. \tag{15}$$

Proof: Let $\mathbf{P}(u, v, w) = (x(u, v, w), y(u, v, w), z(u, v, w))$. Then

$$\begin{aligned} x(u, v, w) &= \sum_{i+j+k=m} B^m_{ijk}(u, v, w)x_{ijk}, \\ y(u, v, w) &= \sum_{i+j+k=m} B^m_{ijk}(u, v, w)y_{ijk}, \\ z(u, v, w) &= \sum_{i+j+k=m} B^m_{ijk}(u, v, w)z_{ijk}. \end{aligned} \tag{16}$$

Following Eq.7, the TB surface $\mathbf{T}(x, y, z)$ can be represented as

$$\mathbf{T}(x, y, z) = (xE_1 + yE_2 + zE_3)^n \, \mathbf{T}_{000}. \tag{17}$$

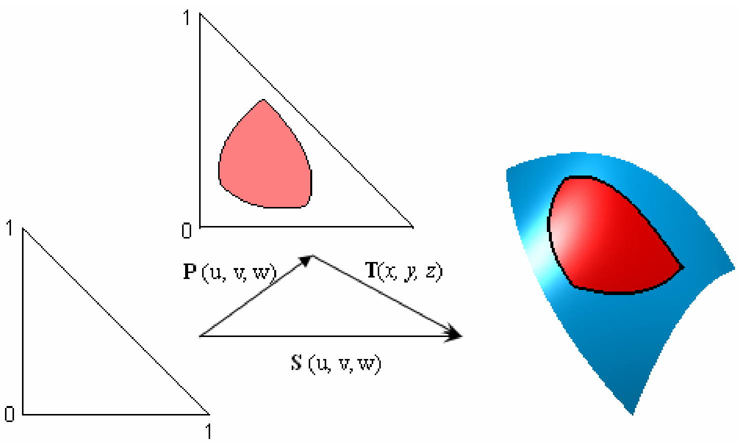


Figure 3: Composition of a TB surface and a domain surface.

Substituting Eq.16 into Eq.17 yields

$$
\begin{aligned}
\mathbf{S}(u,v,w) &= \mathbf{T}(\mathbf{P}(u,v,w)) \\
&= \left((x(u,v,w), y(u,v,w), z(u,v,w)) \begin{pmatrix} E_1 \\ E_2 \\ E_3 \end{pmatrix} \right)^n \mathbf{T}_{000} \\
&= \left(\sum_{i+j+k=m} (x_{ijk}, y_{ijk}, z_{ijk})\, B^m_{ijk}(u,v,w) \begin{pmatrix} E_1 \\ E_2 \\ E_3 \end{pmatrix} \right)^n \mathbf{T}_{000} \\
&= \left(\sum_{i+j+k=m} B^m_{ijk}(u,v,w)\,(x_{ijk}E_1 + y_{ijk}E_2 + z_{ijk}E_3) \right)^n \mathbf{T}_{000}.
\end{aligned}
$$

Applying Eq.6 gives

$$
\begin{aligned}
&\mathbf{S}(u,v,w) \\
&= \sum_{R+S+T=mn} \left(\sum_{|\mathbf{I}^m_n|=R, |\mathbf{J}^m_n|=S, |\mathbf{K}^m_n|=T} \prod_{l=1}^{n} \left(B^m_{I_l J_l K_l}(u,v,w) \left(x_{I_l J_l K_l} E_1 + y_{I_l J_l K_l} E_2 + z_{I_l J_l K_l} E_3 \right) \right) \right) \mathbf{T}_{000}.
\end{aligned}
$$

With Eq.4 and Eq.9, the composition becomes

$$\begin{aligned}
&\mathbf{S}(u,v,w)\\
&=\sum_{R+S+T=mn}\left(\sum_{|\mathbf{I}_n^m|=R,|\mathbf{J}_n^m|=S,|\mathbf{K}_n^m|=T}\left(C^{mn}_{\mathbf{I}_n^m\mathbf{J}_n^m\mathbf{K}_n^m}B^{mn}_{RST}(u,v,w)\mathbf{T}^n_{000}(\mathbf{P}^n_{\mathbf{I}_n^m\mathbf{J}_n^m\mathbf{K}_n^m})\right)\right)\\
&=\sum_{R+S+T=mn}\left(\sum_{|\mathbf{I}_n^m|=R,|\mathbf{J}_n^m|=S,|\mathbf{K}_n^m|=T}\left(C^{mn}_{\mathbf{I}_n^m\mathbf{J}_n^m\mathbf{K}_n^m}\mathbf{T}^n_{000}(\mathbf{P}^n_{\mathbf{I}_n^m\mathbf{J}_n^m\mathbf{K}_n^m})\right)\right)B^{mn}_{RST}(u,v,w)\\
&=\sum_{R+S+T=mn}\mathbf{S}_{RST}B^{mn}_{RST}(u,v,w).
\end{aligned}$$

This completes the proof.

As shown in Eq.8, the construction points $\mathbf{S}^n_{\mathbf{I}_n^m\mathbf{J}_n^m\mathbf{K}_n^m}$ are linear combinations of the control points $\mathbf{T}_{ijk}$. From Eq.5, the control points $\mathbf{S}_{RST}$ in Eq.14 are linear combinations of $\mathbf{S}^n_{\mathbf{I}_n^m\mathbf{J}_n^m\mathbf{K}_n^m}$. Therefore, $\mathbf{S}_{RST}$ are linear combinations of the control points $\mathbf{T}_{ijk}$.

The above formula is similar to the blossoming algorithm [14]. Later we will further simplify it to a more compact one.

3.3. Number of different construction points

A parameter point is a control point of the domain surface. There are $(m+1)(m+2)/2$ parameter points for the domain surface. Every parameter vector $\mathbf{P}^n_{\mathbf{I}_n^m\mathbf{J}_n^m\mathbf{K}_n^m}$ consists of n parameter points. Therefore, the number of parameter vectors is $[(m+1)(m+2)/2]^n$, which implies there are $[(m+1)(m+2)/2]^n$ cases of construction points $\mathbf{S}^n_{\mathbf{I}_n^m\mathbf{J}_n^m\mathbf{K}_n^m}=\mathbf{T}^n_{000}(\mathbf{P}^n_{\mathbf{I}_n^m\mathbf{J}_n^m\mathbf{K}_n^m})$.

Define a power set

$$\mathbf{B}_n^m=\mathbf{M}^n_{(m+1)(m+2)/2}\subset\mathbb{Z}^{(m+1)(m+2)/2}$$

as

$$\mathbf{B}_n^m=\left\{(\beta_{0,0,m},\beta_{0,1,m-1},\beta_{1,0,m-1},\cdots,\beta_{0,m,0},\beta_{1,m-1,0},\cdots,\beta_{m,0,0})\mid\sum_{i+j+k=m}\beta_{ijk}=n,\beta_{ijk}\geq0\right\}. \quad (18)$$

Based on the $(m+1)(m+2)/2$ parameter points $\mathbf{P}_{ijk}=(x_{ijk},y_{ijk},z_{ijk})$ of the domain surface $\mathbf{P}(u,v,w)$, the blossoming point set $\mathbf{Q}_n^m(\mathbf{P})=\mathbf{Q}_n^m(\mathbf{P}(u,v,w))$ is defined by

$$\mathbf{Q}_n^m(\mathbf{P})=\left\{\mathbf{Q}_\mathbf{B}\,\middle|\,\mathbf{Q}_\mathbf{B}=\prod_{i+j+k=m}(x_{ijk}E_1+y_{ijk}E_2+z_{ijk}E_3)^{\beta_{ijk}}\mathbf{T}_{000},\ \ \mathbf{B}\in\mathbf{B}_n^m\right\}. \quad (19)$$

Lemma 3. *A construction point is a blossoming point:* $\mathbf{S}^n_{\mathbf{I}^m_n\mathbf{J}^m_n\mathbf{K}^m_n} \in \mathbf{Q}^m_n(\mathbf{P})$.

Proof: For construction point $\mathbf{S}^n_{\mathbf{I}^m_n\mathbf{J}^m_n\mathbf{K}^m_n} = \mathbf{T}^n_{000}(\mathbf{P}^n_{\mathbf{I}^m_n\mathbf{J}^m_n\mathbf{K}^m_n})$, suppose the parameter point $\mathbf{P}_{ijk} = (x_{ijk}, y_{ijk}, z_{ijk})$ repeats $\beta^{i,j,k}_{\mathbf{I}^m_n\mathbf{J}^m_n\mathbf{K}^m_n}$ times in the parameter vector $\mathbf{P}^n_{\mathbf{I}^m_n\mathbf{J}^m_n\mathbf{K}^m_n}$, which means that the index (i,j,k) repeats $\beta^{i,j,k}_{\mathbf{I}^m_n\mathbf{J}^m_n\mathbf{K}^m_n}$ times in the hyper-index $(\mathbf{I}^m_n, \mathbf{J}^m_n, \mathbf{K}^m_n)$. Then, $\mathbf{S}^n_{\mathbf{I}^m_n\mathbf{J}^m_n\mathbf{K}^m_n}$ can be formulated as

$$\begin{aligned} \mathbf{S}^n_{\mathbf{I}^m_n\mathbf{J}^m_n\mathbf{K}^m_n} &= \mathbf{T}^n_{000}(\mathbf{P}^n_{\mathbf{I}^m_n\mathbf{J}^m_n\mathbf{K}^m_n}) \\ &= \prod_{l=1}^{n} (x_{I_lJ_lK_l}E_1 + y_{I_lJ_lK_l}E_2 + z_{I_lJ_lK_l}E_3)\mathbf{T}_{000} \\ &= \prod_{i+j+k=m} (x_{ijk}E_1 + y_{ijk}E_2 + z_{ijk}E_3)^{\beta^{i,j,k}_{\mathbf{I}^m_n\mathbf{J}^m_n\mathbf{K}^m_n}}\mathbf{T}_{000} \\ &= \mathbf{Q}_{\mathbf{B}_{\mathbf{I}^m_n\mathbf{J}^m_n\mathbf{K}^m_n}} \end{aligned} \tag{20}$$

where

$$\mathbf{B}_{\mathbf{I}^m_n\mathbf{J}^m_n\mathbf{K}^m_n} = \left(\beta^{0,0,m}_{\mathbf{I}^m_n\mathbf{J}^m_n\mathbf{K}^m_n}, \beta^{0,1,m-1}_{\mathbf{I}^m_n\mathbf{J}^m_n\mathbf{K}^m_n}, \beta^{1,0,m-1}_{\mathbf{I}^m_n\mathbf{J}^m_n\mathbf{K}^m_n}, \cdots, \beta^{0,m,0}_{\mathbf{I}^m_n\mathbf{J}^m_n\mathbf{K}^m_n}, \beta^{1,m-1,0}_{\mathbf{I}^m_n\mathbf{J}^m_n\mathbf{K}^m_n}, \cdots, \beta^{m,0,0}_{\mathbf{I}^m_n\mathbf{J}^m_n\mathbf{K}^m_n}\right) \in \mathbf{B}^m_n. \tag{21}$$

This proves the lemma.

According to Lemma 2, the number of blossoming points is

$$|\mathbf{Q}^m_n(\mathbf{P})| = |\mathbf{B}^m_n| = \left|\mathbf{M}^n_{(m+1)(m+2)/2}\right| = \begin{pmatrix} (m+1)(m+2)/2 - 1 + n \\ (m+1)(m+2)/2 - 1 \end{pmatrix}. \tag{22}$$

This number is much smaller than the number of construction points, which is $[(m+1)(m+2)/2]^n$. For example, if $m = 1$, the number of blossoming points is $(n+1)(n+2)/2$ which is the same as the number of control points $\mathbf{T}_{ijk}$, but the number of the construction points is 3^n. Therefore several construction points may correspond to the same blossoming point.

3.4. Geometric algorithm for blossoming points

A blossoming point can be obtained using the blossoming algorithm [14]. In this section, given $\mathbf{B} \in \mathbf{B}^m_n$, we discuss the geometric algorithm for the blossoming point $\mathbf{Q}_\mathbf{B} \in \mathbf{Q}^m_n(\mathbf{P})$. Suppose

$$\mathbf{B} = (\beta_{0,0,m}, \beta_{0,1,m-1}, \beta_{1,0,m-1}, \cdots, \beta_{0,m,0}, \beta_{1,m-1,0}, \cdots, \beta_{m,0,0}). \tag{23}$$

Hyper-index $\mathbf{I}^n_\mathbf{B}$ is defined by repeating (i,j,k) for β_{ijk} times

$$\mathbf{I}^n_\mathbf{B} = \left[\underbrace{(0,0,m), \cdots, (0,0,m)}_{\beta_{0,0,m}}, \underbrace{(0,1,m-1), \cdots, (0,1,m-1)}_{\beta_{0,1,m-1}}, \cdots, \underbrace{(m,0,0), \cdots, (m,0,0)}_{\beta_{m,0,0}}\right]. \tag{24}$$

The parameter vector $\mathbf{P}^n_{\mathbf{B}}$ is defined by repeating $\mathbf{P}_{ijk}$ for β_{ijk} times

$$\mathbf{P}^n_{\mathbf{B}} = \left[\underbrace{\mathbf{P}_{0,0,m},\cdots,\mathbf{P}_{0,0,m}}_{\beta_{0,0,m}},\underbrace{\mathbf{P}_{0,1,m-1},\cdots,\mathbf{P}_{0,1,m-1}}_{\beta_{0,1,m-1}},\cdots,\underbrace{\mathbf{P}_{m,0,0},\cdots,\mathbf{P}_{m,0,0}}_{\beta_{m,0,0}}\right]. \quad (25)$$

Denote the intermediate points at level n by $\mathbf{R}^n_{ijk} = \mathbf{T}_{ijk}$. Suppose the α-th parameter point of $\mathbf{P}^n_{\mathbf{B}}$ is $\mathbf{P}_\alpha = (x_\alpha, y_\alpha, z_\alpha)$. Then there are $(\alpha+2)(\alpha+1)/2$ intermediate points $\mathbf{R}^\alpha_{ijk}$ at level α and $\mathbf{R}^\alpha_{ijk}$ are linear combinations of $\mathbf{R}^{\alpha+1}_{ijk}$ as

$$\mathbf{R}^\alpha_{ijk} = x_{\alpha+1}\mathbf{R}^{\alpha+1}_{i+1,j,k}+y_{\alpha+1}\mathbf{R}^{\alpha+1}_{i,j+1,k}+z_{\alpha+1}\mathbf{R}^{\alpha+1}_{i,j,k+1}, i+j+k=\alpha, \alpha=0,\cdots,n-1.$$

Then the intermediate point at level 0 is the blossoming point $\mathbf{R}^0_{000} = \mathbf{Q}_{\mathbf{B}} \in \mathbf{Q}^m_n(\mathbf{P})$. This is similar to the de Casteljau algorithm [20].

Figure 4 shows an example with $n = 3$. The parameter vector $\mathbf{P}^3_{\mathbf{B}}$ contains 3 parameter points

$$\mathbf{P}_1 = (0.7, 0.2, 0.1), \mathbf{P}_2 = (0.1, 0.1, 0.8), \mathbf{P}_3 = (0.1, 0.5, 0.4).$$

There are 10 intermediate points $\mathbf{R}^3_{ijk}$ in level 3, 6 intermediate points $\mathbf{R}^2_{ijk}$ in level 2, 3 intermediate points $\mathbf{R}^1_{ijk}$ in level 1, and the construction point $\mathbf{R}^0_{000}$ in level 0. All the blossoming points $\mathbf{Q}^m_n(\mathbf{P})$ can be obtained easily.

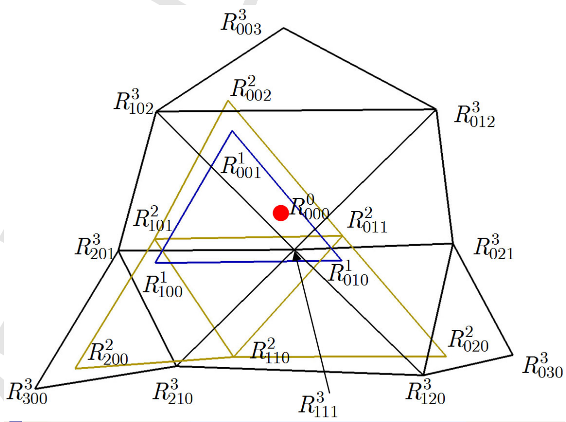


Figure 4: Algorithm for a blossoming point with $n = 3$.

3.5. Control points by blossoming points

In Eq.14, different parameter vectors $\mathbf{P}^n_{\mathbf{I}^m_n\mathbf{J}^m_n\mathbf{K}^m_n} = [\mathbf{P}_{I_l J_l K_l} \mid l = 1, \cdots, n]$ may lead to the same construction point $\mathbf{S}^n_{\mathbf{I}^m_n\mathbf{J}^m_n\mathbf{K}^m_n} = \mathbf{T}^n_{000}(\mathbf{P}^n_{\mathbf{I}^m_n\mathbf{J}^m_n\mathbf{K}^m_n})$. For example,

$$m = n = 2, R = \left|\mathbf{I}^2_2\right| = 0, S = \left|\mathbf{J}^2_2\right| = 1, T = \left|\mathbf{K}^2_2\right| = 3.$$

The construction points are

$$\bar{\mathbf{P}}^n_{\mathbf{I}^2_2\mathbf{J}^2_2\mathbf{K}^2_2} = [\mathbf{P}_{002}, \mathbf{P}_{011}], \tilde{\mathbf{P}}^n_{\mathbf{I}^2_2\mathbf{J}^2_2\mathbf{K}^2_2} = [\mathbf{P}_{011}, \mathbf{P}_{002}],$$

which are equal to the blossoming point $\mathbf{P}^n_{\mathbf{B}} = \{\mathbf{P}_{002}, \mathbf{P}_{011}\}$ with $\mathbf{B} = (1, 1, 0, 0, 0, 0)$. Hence $\mathbf{S}_{013}$ is defined by only one blossoming point. Reformulating Eq.14 using blossoming points from $\mathbf{Q}_{\mathbf{B}} \in \mathbf{Q}^m_n(\mathbf{P})$ in Eq.19 yields a control point $\mathbf{S}_{RST}$

$$\mathbf{S}_{RST} = \sum_{\mathbf{B}\in\mathbf{B}^m_n} F^{RST}_{\mathbf{B}} G^{RST}_{\mathbf{B}} Q_{\mathbf{B}}. \tag{26}$$

Eq.14 describes a control point as a linear combination of the construction points. In a compact way, Eq.26 formulates a control point as a linear combination of the blossoming points. By Eq.26, we avoid the huge number of construction points.

For $\mathbf{B}$ in Eq.23, we define $f_{\mathbf{B}}$ as

$$f_{\mathbf{B}} = \beta_{0,0,m}\begin{pmatrix}0\\0\\m\end{pmatrix} + \beta_{0,1,m-1}\begin{pmatrix}0\\1\\m-1\end{pmatrix} + \cdots + \beta_{0,m,0}\begin{pmatrix}0\\m\\0\end{pmatrix} + \cdots + \beta_{m,0,0}\begin{pmatrix}m\\0\\0\end{pmatrix}.$$

If a blossoming point $\mathbf{Q}_{\mathbf{B}}$ is used to construct $\mathbf{S}_{RST}$, then

$$f_{\mathbf{B}} - (R, S, T)^T = 0.$$

Hence, each blossoming point is used for only one control point. By defining $F^{RST}_{\mathbf{B}}$ as

$$F^{RST}_{\mathbf{B}} = \begin{cases} 0, & f_{\mathbf{B}} - (R, S, T)^T \neq 0, \\ 1, & f_{\mathbf{B}} - (R, S, T)^T = 0. \end{cases} \tag{27}$$

The blossoming points used for a control point can be labeled. Given a $\mathbf{B}$, we get n indices: the number of the index (i, j, k) is β_{ijk} with

$$i + j + k = m, \sum \beta_{ijk} = n.$$

From $\mathbf{B}$, n indices (i, j, k) and $\frac{n!}{\prod(\beta_{ijk}!)}$ different hyper-indices can be obtained. From Eq.14 and Eq.24, for all these hyper-indices $(\mathbf{I}_n^m, \mathbf{J}_n^m, \mathbf{K}_n^m)$, the coefficients of the construction point $\mathbf{S}^n_{\mathbf{I}_n^m \mathbf{J}_n^m \mathbf{K}_n^m}$ are $C^{mn}_{\mathbf{I}_n^m \mathbf{J}_n^m \mathbf{K}_n^m}$:

$$C^{mn}_{\mathbf{I}_n^m \mathbf{J}_n^m \mathbf{K}_n^m} = \frac{\prod\limits_{i+j+k=m} \left(\prod\limits_{l=1}^{\beta_{ijk}} \left(\frac{m!}{i!j!k!} \right) \right)}{\frac{(mn)!}{R!S!T!}}.$$

As a result, for control point $\mathbf{S}_{RST}$, the coefficient of the blossoming point $\mathbf{Q_B}$ is

$$G^{RST}_{\mathbf{B}} = \frac{\prod\limits_{i+j+k=m} \left(\prod\limits_{l=1}^{\beta_{ijk}} \left(\frac{m!}{i!j!k!} \right) \right)}{\frac{(mn)!}{R!S!T!}} \frac{n!}{\prod(\beta_{ijk}!)}. \tag{28}$$

4. Algorithms

In the previous sections the formulae for the control points of the composite surface are derived. Each control point of the composite surface is a linear combination of the blossoming points (Eq.26). In this section, some practical functions or algorithms for computing the control points are presented. The parameter point list, the control point list, and the resulting control point list are denoted by $\mathbf{P}$, $\mathbf{T}$ and $\mathbf{S}$, respectively. They are

$$\begin{aligned}
&\mathbf{P} = \{\mathbf{P}_{00m}, \mathbf{P}_{0,1,m-1}, \mathbf{P}_{1,0,m-1}, \cdots, \mathbf{P}_{0,m,0}, \mathbf{P}_{1,m-1,0}, \cdots, \mathbf{P}_{m00}\}\,, \\
&\mathbf{T} = \{\mathbf{T}_{00n}, \mathbf{T}_{0,1,n-1}, \mathbf{T}_{1,0,n-1}, \cdots, \mathbf{T}_{0,n,0}, \mathbf{T}_{1,n-1,0}, \cdots, \mathbf{T}_{n00}\}\,, \\
&\mathbf{S} = \{\mathbf{S}_{0,0,mn}, \mathbf{S}_{0,1,mn-1}, \mathbf{S}_{1,0,mn-1}, \cdots, \mathbf{S}_{0,mn,0}, \mathbf{S}_{1,mn-1,0}, \cdots, \mathbf{S}_{mn,0,0}\}\,.
\end{aligned} \tag{29}$$

4.1. Power set

Blossoming points are based on the power set. Here a function is provided to derive all the elements of $\mathbf{M}_d^s$. Algorithm 1 returns a matrix $\mathbf{M}$ of size $\begin{pmatrix} d-1+s \\ s \end{pmatrix} \times d$, which is called the power set matrix. Each row of the power set matrix is an element of $\mathbf{M}_d^s$, which is a vector of dimension d.

To illustrate the algorithm, we examine the situation where $d = 3, s =$

```
FUNCTION: M = GetPowerMatrix(d, s)

    c = ( d - 1 + s  choose  s );
    FOR i = d to 1 DO
        r = 1;
        WHILE r < c DO
            v = 0;
            FOR j = i + 1 to d DO
                v+ = M(r, j) ;
            END
            v = s - v;
            IF i == 1 DO
                M(r, i) = v ;
                r + +;
                CONTINUE;
            END
            FOR l = v to 0 DO
                e = ( i - 2 + v - l  choose  v - l );
                FOR k = 1 to e DO
                    M(r, i) = l ;
                    r + +;
                END
            END
        END
    END
```

Algorithm 1: Algorithm for power set matrix $\mathbf{M}$.

$2, c = 6$. Algorithm 1 works as follows.

$$
\left.\begin{array}{lllll}
i=3 & v=2 & l=2 & e=1 & \Rightarrow \mathbf{M}_{1,3}=2 \\
 & & l=1 & e=2 & \Rightarrow \mathbf{M}_{2,3}=\mathbf{M}_{3,3}=1 \\
 & & l=0 & e=3 & \Rightarrow \mathbf{M}_{4,3}=\mathbf{M}_{5,3}=\mathbf{M}_{6,3}=0 \\
i=2 & v=0 & l=0 & e=1 & \Rightarrow \mathbf{M}_{1,2}=0 \\
 & v=1 & l=1 & e=1 & \Rightarrow \mathbf{M}_{2,2}=1 \\
 & & l=0 & e=1 & \Rightarrow \mathbf{M}_{3,2}=0 \\
 & v=2 & l=2 & e=1 & \Rightarrow \mathbf{M}_{4,2}=2 \\
 & & l=1 & e=1 & \Rightarrow \mathbf{M}_{5,2}=1 \\
 & & l=0 & e=1 & \Rightarrow \mathbf{M}_{6,2}=0 \\
i=1 & v=0 & & & \Rightarrow \mathbf{M}_{1,1}=0 \\
 & v=0 & & & \Rightarrow \mathbf{M}_{2,1}=0 \\
 & v=1 & & & \Rightarrow \mathbf{M}_{3,1}=1 \\
 & v=0 & & & \Rightarrow \mathbf{M}_{4,1}=0 \\
 & v=1 & & & \Rightarrow \mathbf{M}_{5,1}=1 \\
 & v=2 & & & \Rightarrow \mathbf{M}_{6,1}=1
\end{array}\right\} \Rightarrow \mathbf{M} = \begin{pmatrix} 0 & 0 & 2 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}
$$

4.2. Point index

Note that $\mathbf{P}_{ijk}$ is the I^m_{ijk}-th point in $\mathbf{P}$ where

$$I^m_{ijk} = 1 + i + \frac{1}{2}(m-k)(m-k+1) = 1 + i + \frac{1}{2}(i+j)(i+j+1). \quad (30)$$

Thus we have

FUNCTION: index = PointIndex(i, j, k)

index $= 1 + i + \frac{1}{2}(i+j)(i+j+1)$;

Algorithm 2: Algorithm for point index.

4.3. Blossoming points

Let M and N as

$$M = (m+1)(m+2)/2, N = \binom{M-1+n}{n}. \quad (31)$$

Let $\mathbf{B}$ be the power set matrix that has size $N \times M$, and $\mathbf{I}$ be a matrix of size $M \times 3$, each row of which is an index. They can be obtained by

$$\begin{array}{l} \mathbf{B} = \text{GetPowerMatrix}\,(M{,}n)\,, \\ \mathbf{I} = \text{GetPowerMatrix}\,(3{,}m)\,. \end{array} \quad (32)$$

A blossoming point can be computed with the geometric algorithm (Algorithm 3) or the classic blossoming algorithm. The blossoming point list $\mathbf{P}_\mathbf{P}$ contains N points.

4.4. Identifying blossoming points for a control point

Each control point $\mathbf{S}_{RST}$ is a linear combination of the blossoming points. We need to find those blossoming points that are contributing to this point. Let $\mathbf{J} = \mathbf{B} \cdot \mathbf{I}$. The i-th row of $\mathbf{B}$ corresponding to the i-th row of $\mathbf{J}$ which corresponds to an index for $\mathbf{S}$, say (R, S, T). Hence, the i-th row of $\mathbf{B}$ contributes to the only one control point (R, S, T). A matrix C_M with N rows can be obtained (Algorithm 4).

Consider **Example 3** in next section. The fifth and ninth rows of $\mathbf{J}$ both correspond to $\mathbf{S}_{211}$. Thus, $\mathbf{S}_{211}$ is the linear combination of the fifth and ninth blossoming points. The fifth row of C_M has values 11 and 17, which implies that the fifth control point $\mathbf{S}_{112}$ is constructed by the 11th and 17th blossoming points. $\mathbf{S}_{013}$ is the second point which is constructed by the 20-th blossoming point.

FUNCTION: $\mathbf{P_P}$ =GetAllBlossomingPoints($\mathbf{T}, n, \mathbf{P}, m, \mathbf{B}, \mathbf{I}, M, N$)

```
FOR i = 1 to N DO
    L = n;
    I_P = T;
    FOR j = 1 to M DO
        r = B(i, j);
        (u_0, v_0, w_0) =P(PointIndex(I(j,1), I(j,2), I(j,3)));
        FOR k = 1 to r DO
            L = L − 1;
            N_L = GetPowerMatrix(3,L);
            FOR each row (i_0, j_0, k_0) in N_L DO
                I_0 = PointIndex(i_0, j_0, k_0);
                I_1 = PointIndex(i_0 + 1, j0, k_0);
                I_2 = PointIndex(i_0, j_0 + 1, k_0);
                I_3 = PointIndex(i_0, j_0, k_0 + 1);
                T_P(I_0) = u_0∗I_P(I_1)+v_0∗I_P(I_2)+w_0∗I_P(I_3);
            END
            I_P = T_P;
        END
    END
    P_P (i)= I_P;
END
```

Algorithm 3: Algorithm for blossoming points.

4.5. Coefficients of the blossoming points

For each blossoming point, a coefficient defined by Eq.28 is calculated by Algorithm 5.

4.6. All control points

Algorithm 6 computes all the control points $\mathbf{S}$.

FUNCTION: $\mathbf{C_M}$ =GetCorrespondenceMatrix($\mathbf{B}$, $\mathbf{I}$, m, n, M, N)

```
J = B · I;
X[1 : Q] = 0;
FOR i = 1 to N DO
    (i0, j0, k0) = J(i);
    j = PointIndex(i0, j0, k0);
    X(j) + +;
    C_M(j, X(j)) = i;
END
```

Algorithm 4: Algorithm for correspondence.

5. Examples and discussion

Example 1: Refer to Figure 5, where $m = 1$ and the domain surface $\mathbf{P}(u, v, w)$ is defined by three parameter control points

$$\begin{aligned}\mathbf{P}_{100} &= (x_{100}, y_{100}, z_{100}),\\ \mathbf{P}_{010} &= (x_{010}, y_{010}, z_{010}),\\ \mathbf{P}_{001} &= (x_{001}, y_{001}, z_{001}).\end{aligned}$$

Then, the control points for the composition are

$$\mathbf{S}_{RST} = \sum_{|\mathbf{I}_n^m|=R, |\mathbf{J}_n^m|=S, |\mathbf{K}_n^m|=T} C^{1n}_{\mathbf{I}_n^m \mathbf{J}_n^m \mathbf{K}_n^m} \mathbf{T}^n_{000}(\mathbf{P}^n_{\mathbf{I}_n^m \mathbf{J}_n^m \mathbf{K}_n^m}).$$

When $|\mathbf{I}_n^m| = R, |\mathbf{J}_n^m| = S, |\mathbf{K}_n^m| = T$, $\forall I_l, J_l, K_l \in \{0, 1\}$, the index for the parameter vector $\mathbf{P}^n_{\mathbf{I}_n^m \mathbf{J}_n^m \mathbf{K}_n^m}$ is $\mathbf{B}_{\mathbf{I}_n^m \mathbf{J}_n^m \mathbf{K}_n^m} = (R, S, T)$. Hence

$$\mathbf{P}_{RST} = \mathbf{T}^n_{000}(\mathbf{P}^n_{\mathbf{I}_n^m \mathbf{J}_n^m \mathbf{K}_n^m}) = \prod_{l=1}^{n} (x_{I_l J_l K_l} E_1 + y_{I_l J_l K_l} E_2 + z_{I_l J_l K_l} E_3) \mathbf{T}_{000}$$

$$= (x_{100}E_1 + y_{100}E_2 + z_{100}E_3)^R (x_{010}E_1 + y_{010}E_2 + z_{010}E_3)^S (x_{001}E_1 + y_{001}E_2 + z_{001}E_3)^T \mathbf{T}_{000}$$

Therefore,

$$\begin{aligned}\mathbf{S}_{RST} &= \sum_{|\mathbf{I}_n^m|=R, |\mathbf{J}_n^m|=S, |\mathbf{K}_n^m|=T} C^{1n}_{\mathbf{I}_n^m \mathbf{J}_n^m \mathbf{K}_n^m} \mathbf{T}^n_{000}(\mathbf{P}^n_{\mathbf{I}_n^m \mathbf{J}_n^m \mathbf{K}_n^m})\\ &= \mathbf{P}_{RST} \sum_{|\mathbf{I}_n^m|=R, |\mathbf{J}_n^m|=S, |\mathbf{K}_n^m|=T} C^{1n}_{\mathbf{I}_n^m \mathbf{J}_n^m \mathbf{K}_n^m}\\ &= \mathbf{P}_{RST}.\end{aligned}$$

FUNCTION: $\mathbf{C_V}$ =GetCoefficientVector($\mathbf{B}$, $\mathbf{I}$, m, n, M, N)

FOR $s = 1$ to N DO
    $B = \mathbf{B}(s) = (B_1, \cdots, B_M)$;
    $(R, S, T) = B \cdot \mathbf{I}$;
    $a = 1$;
    $b = 1$;
    FOR $i = 0$ to m DO
        FOR $j = 0$ to $m - i$ DO
            $k = m - i - j$;
            l= PointIndex(i, j, k);
            $b = b \cdot (B_l!)$;
            For $t = 1$ to B_l DO
                $a = a \cdot \frac{m!}{i!j!k!}$;
            END
        END
    END
    $\mathbf{C_V}(s) = \frac{a \cdot n! \cdot R! \cdot S! \cdot T!}{b \cdot (mn)!}$;
END

Algorithm 5: Algorithm for coefficients.

This result is the same as that of Chang and Davis [21]. In this case, the number of blossoming points is $(n+2)(n+1)/2$, and they are just the control points for the composition.

Figure 6 shows an example of $m = 1, n = 2$. The algorithms yield $M = 3, N = Q = 6$ and

$$\mathbf{J} = \mathbf{B} \cdot \mathbf{I} = \begin{pmatrix} 0 & 0 & 2 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 1 & 0 \\ 2 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 2 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{pmatrix}, \mathbf{C_M} = \begin{pmatrix} 6 \\ 5 \\ 3 \\ 4 \\ 2 \\ 1 \end{pmatrix}, \mathbf{C_V} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}.$$

The value from $\mathbf{C_M}, \mathbf{C_V}$ indicate that each control point equals one blossoming point.

Example 2: This example shows that the surface subdivision can be achieved by composition. Let $m = 1$. A TB surface is subdivided into a TB sub-surface

```
FUNCTION: S =GetAllControlPoints(T, n, P, m)

    Get M, N, Q in Eq.31;
    B, I in Eq.32;
    A = GetAllBlossomingPoints(T, n, P, m, B, I, M, N);
    C_M = GetCorrespondenceMatrix(B, I, m, n, M, N);
    C_V = GetCoefficientVector(B, I, m, n, M, N);
    FOR i = 1 to Q DO
        S(i) = 0;
        X = C_M(i);
        j = 0;
        WHILE X(j) > 0 DO
            B = A(X(j));
            b = C_V(X(j));
            S(i) = S(i) + B * b;
        END
    END
```

Algorithm 6: Algorithm for all control points.

$\mathbf{P}_1(u, v, w)$ whose domain is defined by three parameter points

$$(0, 0, 1), (0, 1, 0), (0.5, 0.5, 0)$$

and another TB sub-surface $\mathbf{P}_2(u, v, w)$ whose domain is defined by parameter points

$$(0, 0, 1), (1, 0, 0), (0.5, 0.5, 0).$$

From Example 1,

$$\mathbf{D}^1_{RST} = (E_3)^R (E_2)^S (0.5E_1 + 0.5E_2)^T \mathbf{T}_{000} = \sum_{i=0}^{T} \begin{pmatrix} T \\ i \end{pmatrix} \mathbf{T}_{i,S+(T-i),R} \Big/ 2^T$$

and

$$\mathbf{D}^2_{RST} = (E_3)^R (E_1)^S (0.5E_1 + 0.5E_2)^T \mathbf{T}_{000} = \sum_{i=0}^{T} \begin{pmatrix} T \\ i \end{pmatrix} \mathbf{T}_{S+i,T-i,R} \Big/ 2^T.$$

Then these two composition surfaces

$$\mathbf{S}^i(u, v, w) = \sum_{R+S+T=mn} B^{mn}_{RST}(u, v, w)\mathbf{D}^i_{RST}, i = 1, 2,$$

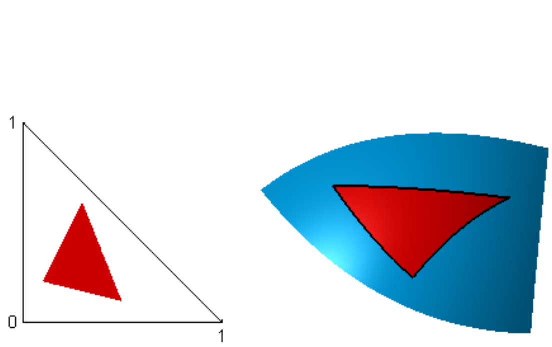
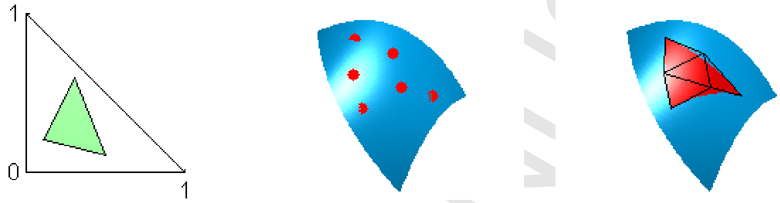


Figure 5: A triangular sub-patch from a TB surface.



(a) (b) (c)

Figure 6: A composite surface with $m = 1, n = 2$: (a) the domain surface; (b) the blossoming points; and (c) the composition surface with the control net.

form a subdivision of the original surface (Figure 7). Figure 8 shows the subdivision of a surface (a leaf) into 6 sub-patches.

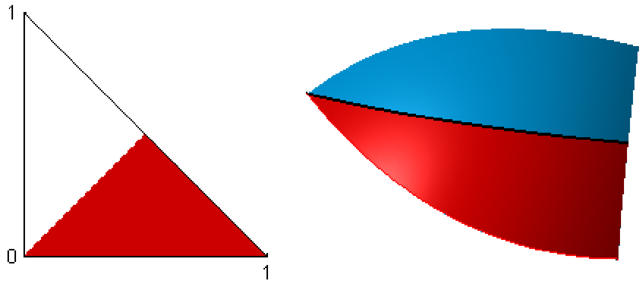
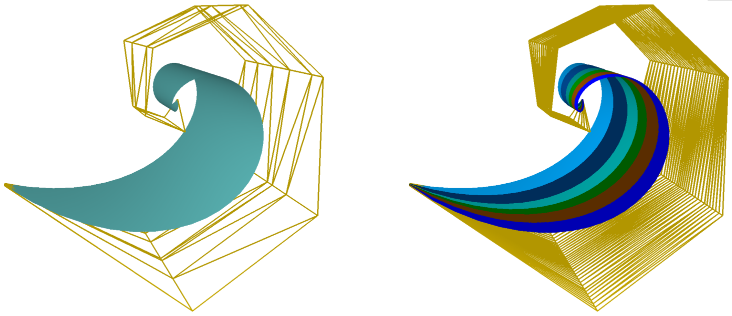


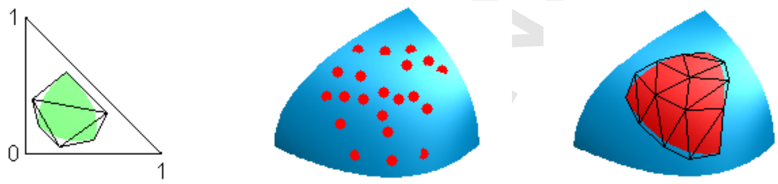
Figure 7: Subdivision of a TB surface.

Example 3: Figure 9 is an example of $m = 2, n = 2$. By the algorithms, we



(a) (b)

Figure 8: Subdivision of a surface (a leaf): (a) the surface with its control nets; and (b) the 6 sub-patches with their control nets.



(a) (b) (c)

Figure 9: A composite surface with $m = 2, n = 2$: (a) the domain surface; (b) the blossoming points; and (c) the composition surface and the control net.

obtain $M = 6, N = 21, Q = 15$, and

$$\mathbf{J}{=}\mathbf{B}\cdot\mathbf{I}{=}\begin{pmatrix} 0&0&0&0&0&2\\ 0&0&0&0&1&1\\ 0&0&0&1&0&1\\ 0&0&1&0&0&1\\ 0&1&0&0&0&1\\ 1&0&0&0&0&1\\ 0&0&0&0&2&0\\ 0&0&0&1&1&0\\ 0&0&1&0&1&0\\ 0&1&0&0&1&0\\ 1&0&0&0&1&0\\ 0&0&0&2&0&0\\ 0&0&1&1&0&0\\ 0&1&0&1&0&0\\ 1&0&0&1&0&0\\ 0&0&2&0&0&0\\ 0&1&1&0&0&0\\ 1&0&1&0&0&0\\ 0&2&0&0&0&0\\ 1&1&0&0&0&0\\ 2&0&0&0&0&0 \end{pmatrix}\begin{pmatrix} 0&0&2\\ 0&1&1\\ 1&0&1\\ 0&2&0\\ 1&1&0\\ 2&0&0 \end{pmatrix}=\begin{pmatrix} 4&0&0\\ 3&1&0\\ 2&2&0\\ 3&0&1\\ 2&1&1\\ 2&0&2\\ 2&2&0\\ 1&3&0\\ 2&1&1\\ 1&2&1\\ 1&1&2\\ 0&4&0\\ 1&2&1\\ 0&3&1\\ 0&2&2\\ 2&0&2\\ 1&1&2\\ 1&0&3\\ 0&2&2\\ 0&1&3\\ 0&0&4 \end{pmatrix},\mathbf{C_M}{=}\begin{pmatrix} 21&0\\ 20&0\\ 18&0\\ 15&19\\ 11&17\\ 6&16\\ 14&0\\ 10&13\\ 5&9\\ 4&0\\ 12&0\\ 8&0\\ 3&7\\ 2&0\\ 1&0 \end{pmatrix},\mathbf{C_V}{=}\begin{pmatrix} 1\\ 1\\ 1/3\\ 1\\ 1/3\\ 1/3\\ 2/3\\ 1\\ 2/3\\ 2/3\\ 1/3\\ 1\\ 1/3\\ 1\\ 1/3\\ 2/3\\ 2/3\\ 1\\ 2/3\\ 1\\ 1 \end{pmatrix}.$$

Hence, 21 different blossoming points and 15 control points are generated.

Figure 10 shows more examples with $m = 3, n = 3$; $m = 3, n = 5$; and $m = 4, n = 3$.

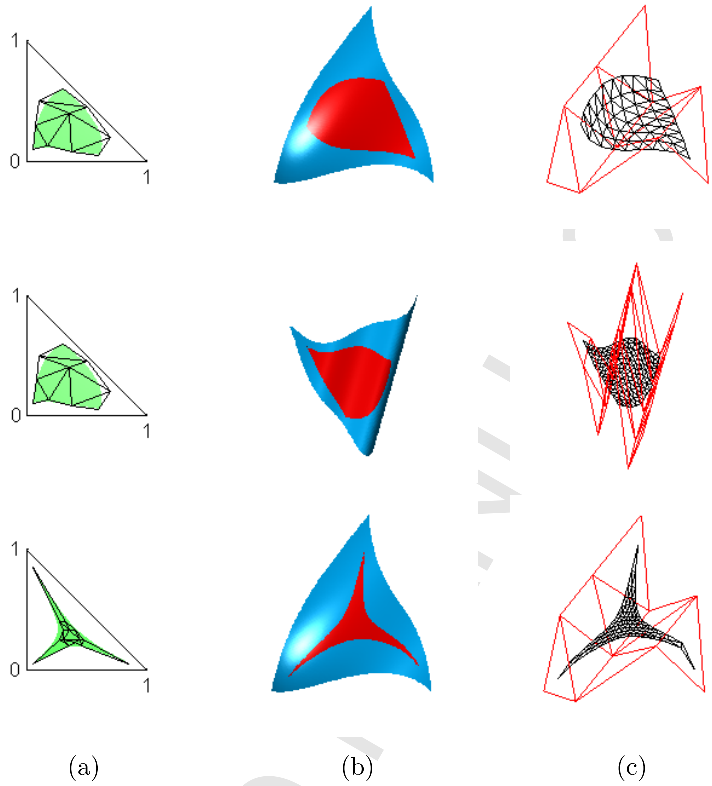


Figure 10: Composite surfaces: first row: $m = 3, n = 3$; second row: $m = 3, n = 5$; third row: $m = 4, n = 3$. (a) the domain surfaces; (b) the blossoming points; and (c) the composition surfaces and the control nets.

Example 4: Different parameterizations of the composite surface. Different choices of the interior control points for a domain surface could lead to different parameterizations of the composite surface. Figure 11 shows an example of a composition with $m = n = 5$. Figures 11(b-f) are different choices of interior control point. In each case, the domain surface is uniformly sampled (the green curves are parameter curves). Uniform parameter curves (Figure 11(b)) in the domain surface lead to uniform parameter curves (Figure 11(h)) in the composite surface. Moving interior control points causes the change in the density of parameter curves for both the domain surface (Figure 11(c) and Figure 11(d)) and the composite surfaces (Figure 11(i) and Figure 11(j)). It may also cause the parameter curves intersecting with each other (Figure 11(e), Figure 11(f), Figure 11(k) and Figure 11(l)).

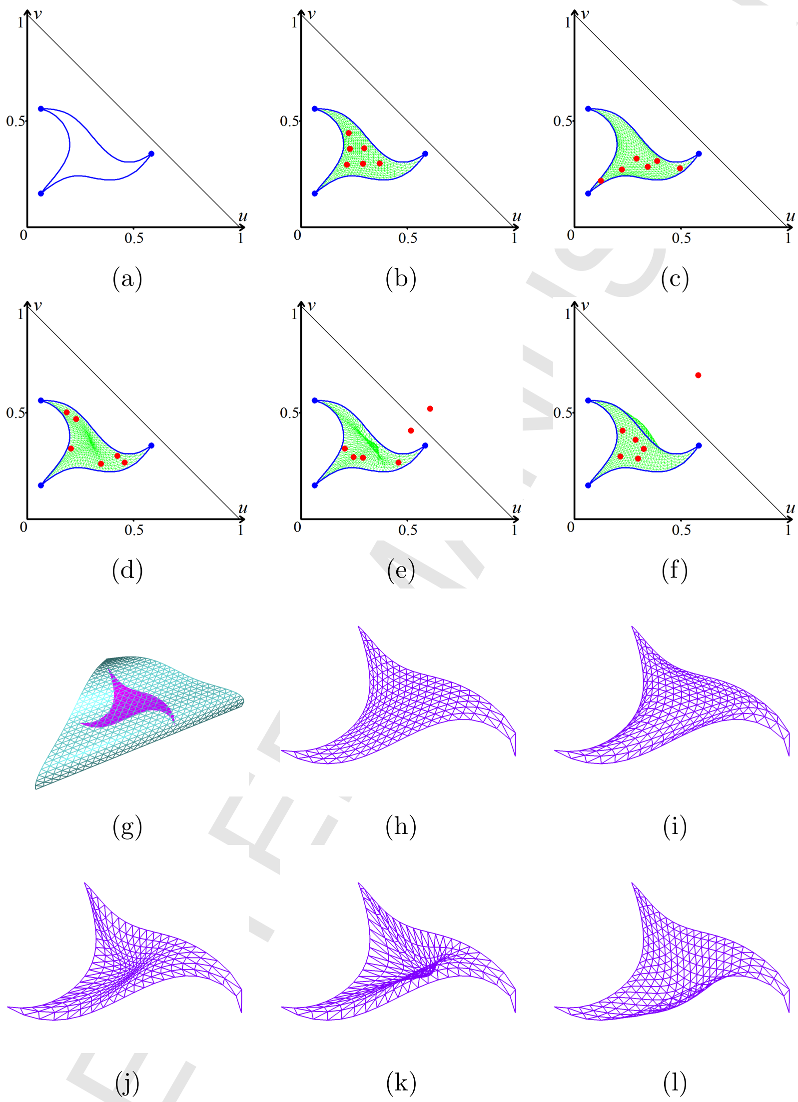


Figure 11: Different parameterizations of a composite surface: (a) an area bounded by three curves; (b-f) different choices of interior control points; (g) the sub-patch defined by the three curves in (a); and (h-l) different parameterizations of the sub-patch.

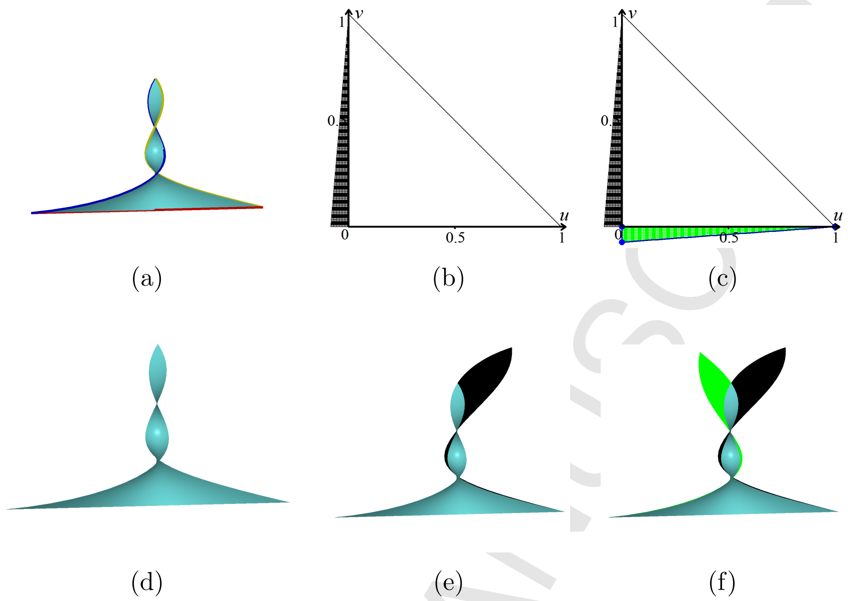


Figure 12: Surface extensions: (a) a surface with three boundary curves in red, blue and yellow; (b) one domain surface in black; (c) another domain surface in green; (d) the surface; (e) the composite surface from (b); and (f) the composite surface from (c).

Example 5: Surface extensions. Figure 12(a) shows a TB surface with three boundary curves in red, blue and yellow. The surface domain for the TB surface (Eq.1) is D_T. If we extend the surface domain to the black area in Figure 12(b), the surface is extended. The composite surface (Figure 12(e)), which is also a TB surface, becomes a nature extension of the original TB surface (Figure 12(d)) on the yellow boundary. Similarly, Figure 12(c) and Figure 12(f) show the extension on the blue boundary.

6. Conclusions

In this paper, an approach to generating a TB sub-patch from a TB surface is presented. The TB sub-patch is formed by composition of the TB surface and the domain surface, which is also a TB surface. An explicit formula for computing the control points of the composition is derived. These new control points are linear combinations of the construction points. However, the number of construction points is huge especially when the surface degrees are high. Thus we simplify the formula to express the control points of the TB sub-patch as linear combinations of the blossoming points. The total number of the blossoming points is much smaller than the number of construction points. The geometric algorithm for the blossoming points is analyzed. Finally, detailed algorithms are provided to efficiently derive the composition.

Acknowledgement This work is partially supported by the ARC 9/09 Grant (MOE2008-T2-1-075) of Singapore.

References

[1] T. DeRose, Composing Bézier simplexes, ACM Transactions on Graphics 7 (3) (1988) 221.

[2] B. Jüttler, W. Wang, The shape of spherical quartics, Computer Aided Geometric Design 20 (8-9) (2003) 621–636.

[3] S. Hu, G. Wang, T. Jin, Generalized subdivision of Bézier surfaces, Graphical Models and Image Processing 58 (3) (1996) 218–222.

[4] W. Boehm, G. Farin, Concerning subdivision of bézier triangles, letter to the editor, Computer-Aided Design 15 (5) (1983) 260 – 261.

[5] W. Böhm, G. Farin, J. Kahmann, A survey of curve and surface methods in CAGD, Computer Aided Geometric Design 1 (1) (1984) 1–60.

[6] R. Goldman, Subdivision algorithms for Bézier triangles, Computer-Aided Design 15 (3) (1983) 159–166.

[7] H. Seidel, A general subdivision theorem for Bézier triangles, Mathematical Methods in Computer Aided Geometric Design 573–581.

[8] I. Brueckner, Construction of Bézier points of quadrilaterals from those of triangles, Computer-Aided Design 12 (1) (1980) 21–24.

[9] W. Waggenspack Jr, D. Anderson, Converting standard bivariate polynomials to Bernstein form over arbitrary triangular regions, Computer-Aided Design 18 (10) (1986) 529–532.

[10] T. Jie, A geometric condition for smoothness between adjacent rational Bézier surfaces, Computers in Industry 13 (4) (1990) 355–360.

[11] R. Goldman, D. Filip, Conversion from Bézier rectangles to Bézier triangles, Computer-Aided Design 19 (1) (1987) 25–27.

[12] S. Hu, Conversion of a triangular Bézier patch into three rectangular Bézier patches, Computer Aided Geometric Design 13 (3) (1996) 219–226.

[13] X. Sheng, B. Hirsch, Triangulation of trimmed surfaces in parametric space, Computer-Aided Design 24 (8) (1992) 437–444.

[14] T. DeRose, R. Goldman, H. Hagen, S. Mann, Functional composition algorithms via blossoming, ACM Transactions on Graphics (TOG) 12 (2) (1993) 113–135.

[15] D. Lasser, Tensor product Bézier surfaces on triangle Bézier surfaces, Computer Aided Geometric Design 19 (8) (2002) 625 – 643.

[16] D. Lasser, Triangular subpatches of rectangular bézier surfaces, Computers & Mathematics with Applications 55 (2008) 1706–1719.

[17] J. Feng, Q. Peng, Functional compositions via shifting operators for Bézier patches and their applications, Chinese Journal of Software 10 (12) (1999) 1316–1322.

[18] G. Farm, Triangular Bernstein-Bézier patches, Computer Aided Geometric Design 3 (1986) 83–127.

[19] G. Chang, Bernstein polynomials via the shifting operator, American Mathematical Monthly 91 (10) (1984) 634–638.

[20] G. Farin, G. Farin, Curves and Surfaces for CAGD: a Practical Guide, Morgan Kaufmann Pub, 2002.

[21] G. Chang, P. Davis, The convexity of Bernstein polynomials over triangles, Journal of Approximation Theory 40 (1) (1984) 11–28.