Patch Interpolation Using Edge-based Blending Function

Kenjiro T. Miura
Department of Software, University of Aizu,
Aizu-Wakamatsu, Fukushima 965-80, Japan
E-mail: ktmiura@u-aizu.ac.jp

Masashi Adachi
Department of Software, University of Aizu,
Aizu-Wakamatsu, Fukushima 965-80, Japan
E-mail: s1011100@u-aizu.ac.jp

Hiroaki Chiyokura
Faculty of Environmental Information, Keio University,
5322 Endo, Fujisawa, Kanagawa 252, Japan
E-mail: chiyo@sfc.keio.ac.jp

ABSTRACT

By introducing the edge-based blending function derived from the surface-based one, we can capture the effects of the edges on the shape of the n-sided patch more directly. Because of the good natures of the edge-based blending function, the designers can manipulate it as well as the geometry of the boundary curves to design and modify the patch shape. In this paper, we have discussed the derivation of the edge-based blending function, its properties, especially the differences between ones for odd-sided and for even-sided patches, and easiness of designing it by the users. We have demonstrated its validty by showing several triangular, rectangular, and pentagonal patchs using the modifiable blending functions.

1. Introduction

The need for the surface representation scheme to cover n-sided faces was pointed out during the first half of 1980's by several pioneers, for example, Gregory[3], Sabin[6], and Hosaka and Kimura[5]. Recent researches done by Loop and DeRose, Grim and Hughes have extended Bézier[7] and B-spline[8,4] surface schemes to include non-rectangular topology. Anodther type of research to interpolate arbitrary topological type surfaces was carried out by smoothly approximating an irregular control mesh[9].

Most of the approches to model surfaces are based on the idea of control points. Manipulating positions of the control points, one can modify the surface shape according to his/her intensions. The policy behind the control points methodology is the mixture the geometrical elements and the blending functions. In Bézier surface scheme, the blending functions for the control points depend only on the number of the control points, i.e, order in one parameter direction and that of the other. In B-spline surface scheme, it depends on the orders and knot vectors of the surface. This makes the designers concentrate on a relatively simple work of positioning points to design complicated surfaces without considering how to define blending functions.

That is the main reason of using the control point.

However, Blanc and Schlick[1] showed the validity and versatility of manipulating the blending functions as well as the positions of the control points by using a simple parameter given to each control point. Their work suggests that it be very useful even for the end-user to design the blending function itself if we can supply a simple tool to do it.

Charrot and Gregory[2] developed a pentagonal surface patch which is a convex combination or blend of five component interpolants: surface patches. Each of the interpolants match given boundary curves and cross boundary tangent vectors on two sides of a regular pentagon. They used a newly defined blending function for each interpolant. Their interpolation scheme was based on the boundary curves and the cross boundary tangent vectors, but their blending function indicates the rate of the effects of each interpolant patch to the final surface shape and it is very difficult for the designers to predict the effect of changing one of the boundary curves. Furthermore, because the blending function is defined in 2 dimensional space of the pentagonal domain, it is practically impossible to change, or design in stronger sense, the blending function with a reasonably simple tool.

Hence in this paper, by introducing blending functions for the edges of the n-sided patch($n \geq 3$) instead of those for the interpolant patches, we interpolate arbitorary topological type surfaces for given boundary curves. The blending function is practically modifiable by simple parameters because it is virtually of one variable. In the following section, we review Charrot and Gregory's pentagonal patch in more detail and clarify problems of their blending function. Section 3 introduce a new edge-based blending function called β blending function and discuss our interpolation method for odd-sided and even-sided patches. In section 4, we explain how to modify β blending function. Several examples with variously modified blending function are shown to demonstrate the validity of our method.

2. The Pentagonal Patch Interpolation Using α Blending Function

Charrot and Gregory[2] developed a method to interpolate a pentagonal domain using a new blending function. In this paper, we call it α blending function to clearify difference between it and our new blending function: β blending function. In this section, we will review their method and address its problems on α blending function.

2.1. The component interpolant

The component interpolants of the pentagonal patch can be described in terms of a parametric coordinate system (s,t) as follows. Let

$$\boldsymbol{F}(s,t) = [x(s,t), y(s,t), z(s,t)] \tag{1}$$

be a given vector valued function. Then

$$\boldsymbol{P}(s,t) = \boldsymbol{F}(0,t) + \boldsymbol{F}(s,0) - \boldsymbol{F}(0,0) \tag{2}$$

defines an interpolant which is such that

$$\boldsymbol{P}(0,t) = \boldsymbol{F}(0,t) \quad \textit{and} \quad \boldsymbol{P}(s,0) = \boldsymbol{F}(s,0) \tag{3}$$

2.2. The pentagonal domain

We define a local coordinate system for the pentagonal domain. Let Ω be a regular pentagon of height unity and with vertices $V_i = (u_i, v_i)$, $i = 0, \ldots, 4$ in the (u, v) plane. As shown in Fig. 1(a), the patch will be represented in terms of the variables λ_i, $i = 0, \ldots, 4$ which denote a perpendicular distances of the general point $V = (u, v)$ from the sides opposite the vertices V_i, $i = 0, \ldots, 4$. Each λ_i becomes 0 on one side of the pentagonal domain and is greater than 0 otherwise.

We define a local coordinate system (s_i, t_i) for the pentagon as follows:

$$s_i = \frac{\lambda_{i+2}}{\lambda_{i-1} + \lambda_{i+2}}, \quad t_i = \frac{\lambda_{i-2}}{\lambda_{i+1} + \lambda_{i-2}}, \tag{4}$$

with the property that for $V \in \Omega$ we have that

$$0 \leq s_i \leq 1 \quad \text{and} \quad 0 \leq t_i \leq 1. \tag{5}$$

Here, and else where in this paper, a suffix i is interpreted as i mod 5 to make it inside the range $0 \leq i \leq 4$ when we deal with the pentagonal domain and i mod n to make it inside the range $0 \leq i \leq n-1$ when with the n-sided domain.

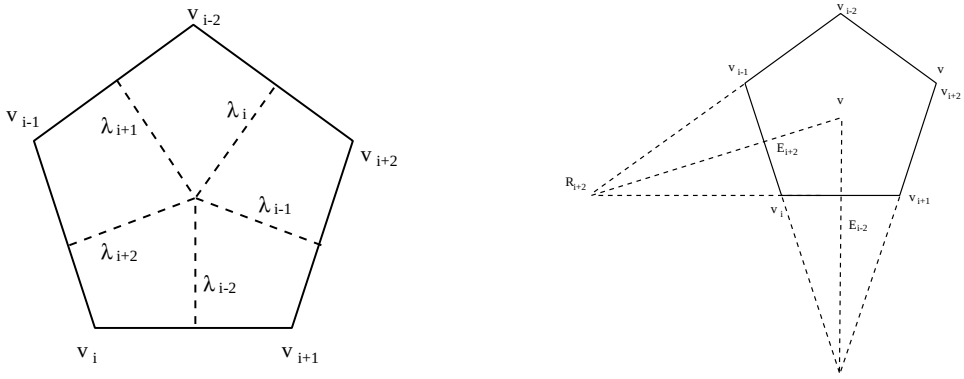
Using this coordinate system, the points E_{i-2} and E_{i+2} on the boundary of the pentagon Ω shown in Fig. 1(b) are expressed as

$$E_{i-2} = (1 - s_i)V_i + s_i V_i, \tag{6}$$

$$E_{i+2} = (1 - t_i)V_i + t_i V_i. \tag{7}$$

Let the intersection point between the sides $\lambda_{i+1} = 0$ and $\lambda_{i+1} = 0$ be R_i. Then the radial line joining V and R_i goes through E_i.

The detailed discussion about the local coordinate system mentioned above can be found in Charrot and Gregory[2]. Other type of local coordinate systems for general n-sided domain was proposed by Loop and DeRose[7]. Note that the shape of the interpolated patch depends on the used coordinate system. However, our interpolation method using β blending function as well as α one can be applied for any local



(a) The pentagon Ω (b) The local coordinate system

Figure 1: The Pentagonal Domain

coordinate systems.

2.3. The pentagonal patch

We review Charrot and Gregory's interpolation method of a pentagonal domain, especially for the C^0 patch. Let

$$\boldsymbol{F}(V) = [x(V), y(V), z(V)], \quad V \in \Omega, \tag{8}$$

denote a vector valued function defined on the pentagon Ω. We assume that $\boldsymbol{F}$ is defined everywhere on the sides λ_i, $i = 0, \ldots, 4$ of the pentagon Ω. From Eq. (2), with the local coordinate system (s_i, t_i), the vector valued function can de defined as

$$\boldsymbol{P}_i(V) = \boldsymbol{F}(E_{i+2}) + \boldsymbol{F}(E_{i-2}) - \boldsymbol{F}(V_i). \tag{9}$$

This function interpolates $\boldsymbol{F}$ on the sides $\lambda_{i+2} = 0$ and $\lambda_{i-2} = 0$ of the pentagon, that is

$$\boldsymbol{P}_i(E_{i+2}) = \boldsymbol{F}(E_{i+2}), \quad \boldsymbol{P}_i(E_{i-2}) = \boldsymbol{F}(E_{i-2}). \tag{10}$$

The pentagonal patch is defined by the convex combination using α blending function $\alpha_i(V)$

$$\boldsymbol{P}(V) = \sum_{i=0}^{4} \alpha_i(V) \boldsymbol{P}_i(V) \tag{11}$$

where

$$\alpha_i(V) = \frac{\lambda_{i-1}\lambda_i\lambda_{i+1}}{\sum_{k=0}^{4} \lambda_{k-1}\lambda_k\lambda_{k+1}}. \tag{12}$$

Eq. (11) is a convex combination since the α_i, $i = 0, \ldots, 4$ are positive on Ω and, by construction

$$\sum_{i=0}^{4} \alpha_i(V) = 1. \tag{13}$$

Since

$$\alpha_i(E_i) = 0, \quad i = j-1, j, j+1, \tag{14}$$

it follows that

$$\boldsymbol{P}(E_j) = \boldsymbol{F}(E_j), \quad j = 0, \ldots, 4, \tag{15}$$

that is, $\boldsymbol{P}$ interpolates $\boldsymbol{F}$ on the entire boundary of the pentagon.

2.4. Problems of α blending function

Eq. (11) can be easily extended to a general case where the domain is not only pentagonal, but also n-sided if a certain local coordinate system is defined there :

$$\boldsymbol{P}(V) = \sum_{i=0}^{n-1} \alpha_i(V) \boldsymbol{P}_i(V) \tag{16}$$

where

$$\alpha_i(V) = \frac{\lambda_{i-1} \ldots \lambda_{i+n-4}}{\sum_{k=0}^{n-1} \lambda_{k-1} \ldots \lambda_{k+n-4}} \tag{17}$$

with some variables λ_i, $i = 0, \ldots, n-1$ that possess the following properties :

$$\lambda_i \geq 0 \quad \text{and} \quad \lambda_i(E_i) = 0. \tag{18}$$

However, there are several problems of using α blending function in Eq. (16). The worst one is that as shown in the definition of the vector vlaued function $\boldsymbol{P}_i(V)$ expressed by Eq. (9), $\boldsymbol{P}_i(V)$ uses $\boldsymbol{F}(E_{i+2})$ and $\boldsymbol{F}(E_{i-2})$. This means that $\boldsymbol{F}(E_i)$ is used by two different functions $\boldsymbol{P}_{i+2}(V)$ and $\boldsymbol{P}_{i-2}(V)$. It makes unclear the effects of $\alpha_i(V)$ on the shape of the interpolated patch. Especially to design or modify the shape by changing the values of $\alpha_i(V)$ is a very difficult task because of its unclarity.

The second problem is that α blending function is defined in 2 dimensional space of the n-sided domain and it is essentially not an easy task to change and design the function of two variables.

The third problem is that the degree and complexity might be unnecessarily high although the requirements for the function $\alpha_i(V)$ are only three conditions as follows:

$$0 \leq \alpha_i(V) \leq 1, \quad i = 0, \ldots, n-1, \tag{19}$$

$$\sum_{i=0}^{n-1} \alpha_i(V) = 1, \tag{20}$$

$$\alpha_i(E_j) = 0, \quad i = j-1, \ldots, j+n-4. \tag{21}$$

To overcome the problems mentioned above, we will introduce β blending function derived from α one. The formulation of β blending function is simple, but it has much better properties than those of α that clarify the effects of the blending function on the shape and make it possible for the designers to design the shape by changing the function values. Also it makes clear one of the essential differences between even-sided and odd-sided patches.

3. β Blending Function

In this section, first we will define β blending function and address the requirements for it inherited from those for α blending function expressed by Eqs. (19), (20), and (21). Then two interpolation cases: even-sided and odd-sided patches are treated separately.

3.1. Definition of β blending function

As mentined in the previous section, even though the patch-formulation by Eq. (16) are based on the shapes of the boundary curves $\boldsymbol{F}_i$, $i = 0, \ldots, n-1$, the effect on the patch shape of each boudary curve is not clear enough. Therefore we introduce β blending function to clarify the effect. Before we define it, we would like to point out a fact that the requirements for α blending function expressed by Eq. (19), (20), and (21) lead to

$$\alpha_i(E_{i-2}) + \alpha_{i+1}(E_{i-2}) = 1, \quad i = 0, \ldots, n-1. \tag{22}$$

Considering the fact mentioned above, we define β blending function as follows:

$$\beta_i(V) \equiv \alpha_i(V) + \alpha_{i+1}(V), \quad i = 0, \ldots, n-1. \tag{23}$$

This formulation looks rather simple and it does not seem to have much better properties than those of α blending function at the first glance. However, for example, the requirements become simpler and more intuitive.

The requirements for β blending function are derived from Eqs. (19) and (20),

$$0 \le \beta_i(V) \le 1, \quad i = 0, \ldots, n-1, \tag{24}$$

from Eq. (20), since each $\alpha_i(V)$ appears twice in $\sum_{i=0}^{n-1} \beta_i(V)$,

$$\sum_{i=0}^{n-1} \beta_i(V) = 2 \sum_{i=0}^{n-1} \alpha_i(V) = 2. \tag{25}$$

The last one expressed by Eq. (21) is rewritten as three requirements:

$$\begin{aligned} \beta_i(E_j) &= 0, \quad i = j, \ldots, j+n-4, \\ \beta_i(E_{i-2}) &= 1, \\ \beta_{i-1}(E_{i-2}) + \beta_{i+1}(E_{i-2}) &= 1. \end{aligned} \tag{26}$$

These three requirements imply that each $\beta_i(V)$ is equal to 1 on one of the boundary curve E_{i-2} and on its adjacent boundary curves E_{i-3} and E_{i-1}, the sum of β_i with β_{i-1} and with β_{i+1} are equal to 1, respectively.

The patch formulation expressed by Eq. (16) is rewritten by using β and α blending functions as follows:

$$\begin{aligned} \boldsymbol{P}(V) &= \sum_{i=0}^{n-1} \alpha_i(V)\boldsymbol{P}_i(V) \\ &= \sum_{i=0}^{n-1} \beta_i(V)\boldsymbol{F}(E_{i-2}) - \sum_{i=0}^{n-1} \alpha_i(V)\boldsymbol{F}(V_i). \end{aligned} \tag{27}$$

The above expression tells us two things. One is that each boundary curve affects the patch shape at the weight of its corresponding β blending function. The other is that we still need the value of α blending function to calculate $\boldsymbol{P}(V)$. In order to get its value, we have to solve the following equations:

$$\alpha_i(V) + \alpha_{i+1}(V) = \beta_i(V), \quad i = 0, \ldots, n-1. \tag{28}$$

In matrix form, Eq. (28) becomes

$$\boldsymbol{M}\boldsymbol{\alpha} = \boldsymbol{\beta} \tag{29}$$

where

$$\boldsymbol{M} = \begin{bmatrix} 1 & 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 1 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & 1 & 1 \\ 1 & 0 & 0 & 0 & \ldots & 0 & 1 \end{bmatrix}, \quad \boldsymbol{\alpha} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{n-2} \\ \alpha_{n-1} \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{n-2} \\ \beta_{n-1} \end{bmatrix}. \tag{30}$$

The matrix $\boldsymbol{M}$ is not singular if n is an odd number. On the other hand, it is singular if n is even. In case where n is odd, since $\boldsymbol{M}$ is not singular, it is straightforward to calculate the value of α blending function from that of β blending function by solving Eq. (29). However, in the other case where n is even, we have to apply other method to determine the value of α blending function. Because of this fact, in the following subsections, each case is discussed separately.

3.2. Odd-sided patch

As discussed above, when n is an odd number, it is possible to solve Eq. (29) to calculate the value of α blending function. This means that we can define or design

β blending function freely with the requirements expressed by Eqs. (24), (25), and (26).

The inverse matrix $\boldsymbol{M}^{-1}$ of $\boldsymbol{M}$ in Eq. (29) is given as

$$\boldsymbol{M}^{-1} = \frac{1}{2}\begin{bmatrix} 1 & -1 & 1 & -1 & \dots & -1 & 1 \\ 1 & 1 & -1 & 1 & \dots & 1 & -1 \\ -1 & 1 & 1 & -1 & \dots & -1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & -1 & 1 & -1 & \dots & 1 & -1 \\ -1 & 1 & -1 & 1 & \dots & 1 & 1 \end{bmatrix}. \tag{31}$$

Hence α_i, $i = 0, \dots, n-1$ are obtained as

$$\alpha_i = \frac{1}{2}\{\sum_{j=0}^{i-1}(-1)^{i+j+1}\beta_j + \sum_{j=i}^{n-1}(-1)^{i+j}\beta_j\}, \quad i = 0, \dots, n-1. \tag{32}$$

Note that from Eqs. (19), (25), and (32),

$$1 \le \frac{1}{2}\{\sum_{j=0}^{i-1}(-1)^{i+j+1}\beta_j + \sum_{j=i}^{n-1}(-1)^{i+j}\beta_j\} + \frac{1}{2}\sum_{i=0}^{n}\beta_i \le 2, \quad i = 0, \dots, n-1, \tag{33}$$

i.e., for example, in case of $i = 0$,

$$1 \le \beta_0 + \beta_2 + \dots + \beta_{n-1} \le 2. \tag{34}$$

3.3. Even-sided patch

Before we deal with the general even-sided patch, we discuss the rectangular patch, i.e. the case where n is equal to 4. As a local coordinate system for the rectangular domain(unit square), we use the ordinary coordinate system (u, v) shown in Fig.2. Then, λ_i, $i = 0, \dots, 3$ can be given as

$$\lambda_0 = 1 - u, \quad \lambda_1 = 1 - v, \quad \lambda_2 = u, \quad \lambda_3 = v. \tag{35}$$

From Eq. (12), α_0 is given as

$$\begin{aligned} \alpha_0 &= \frac{\lambda_3\lambda_0}{\sum_{k=0}^{3}\lambda_{k-1}\lambda_k} \\ &= \frac{v(1-u)}{v(1-u) + (1-u)(1-v) + (1-v)u + uv} \\ &= v(1-u). \end{aligned} \tag{36}$$

Similarly,

$$\alpha_1 = (1-u)(1-v), \quad \alpha_2 = (1-v)u, \quad \alpha_3 = uv. \tag{37}$$

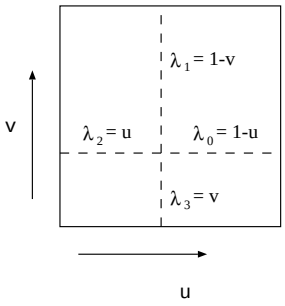


Fig. 2. The Local Coordinate system (u, v).

Hence β_i, $i = 0, \ldots, 3$ are given as

$$\begin{aligned}
\beta_0 &= \alpha_0 + \alpha_1 = v(1-u) + (1-u)(1-v) = 1-u, && (38)\\
\beta_1 &= \alpha_1 + \alpha_2 = (1-u)(1-v) + (1-v)u = 1-v, && (39)\\
\beta_2 &= \alpha_2 + \alpha_3 = (1-v)u + uv = u, && (40)\\
\beta_3 &= \alpha_3 + \alpha_0 = uv + v(1-u) = v. && (41)\\
& && (42)
\end{aligned}$$

The above expressions suggest that α blending function can be calculated by using β blending function as follows:

$$\alpha_0 = \beta_3\beta_0, \quad \alpha_1 = \beta_0\beta_1, \quad \alpha_2 = \beta_1\beta_2, \quad \alpha_3 = \beta_2\beta_3. \tag{43}$$

This is possible because the sum of two β blending functions β_{i-1} and β_{i+1} is equal to 1, i.e.

$$\begin{aligned}
\alpha_i + \alpha_{i+1} &= \beta_{i-1}\beta_i + \beta_i\beta_{i+1} && (44)\\
&= (\beta_{i-1} + \beta_{i+1})\beta_i && (45)\\
&= \beta_i. && (46)
\end{aligned}$$

We extend the procedure discussed above to calculate α blending function for the general even-sided domain. We add the following requirements on β blending function such that

$$\begin{aligned}
\sum_{i=0}^{\frac{n}{2}-1} \beta_{2i} &= \beta_0 + \beta_2 + \ldots + \beta_{n-2} = 1, && (47)\\
\sum_{i=0}^{\frac{n}{2}-1} \beta_{2i+1} &= \beta_1 + \beta_3 + \ldots + \beta_{n-1} = 1. && (48)
\end{aligned}$$

Then β blending function can be expressed as follows:

$$\beta_{2i} = \alpha_{2i} + \alpha_{2i+1} = \beta_{2i}(\sum_{j=0}^{\frac{n}{2}-1} \beta_{2i+1}), \quad i = 0, \ldots, \frac{n}{2} - 1, \tag{49}$$

$$\beta_{2i+1} = \alpha_{2i+1} + \alpha_{2(i+1)} = \beta_{2i+1}(\sum_{j=0}^{\frac{n}{2}-1} \beta_{2i}), \quad i = 0, \ldots, \frac{n}{2} - 1. \tag{50}$$

Several methods are applicable to determine α blending function from Eqs. (49) and (50). One rational method is that since, for example, the edge corresponding to β_0 is located relatively far away from the edges corresponding to β_3, β_5,..., and β_{n-3}, we assume that there is no interference among these β blending functions, i.e.

$$\beta_{2i}\beta_{2j+1} = 0, \quad j = 0, \ldots, \frac{n}{2} - 1, j \neq i-1, i, \tag{51}$$

$$\beta_{2i+1}\beta_{2j} = 0, \quad j = 0, \ldots, \frac{n}{2} - 1, j \neq i, i+1. \tag{52}$$

Then α blending functions are given

$$\alpha_i = \beta_{i-1}\beta_i, \quad i = 0, \ldots, n. \tag{53}$$

Another possible method is that we define α blending function as follows:

$$\alpha_{2i} = \beta_{2i-1}\beta_{2i} + \frac{1}{2}\beta_{2i} \sum_{j=0, j\neq i-1, i}^{\frac{n}{2}} \beta_{2j+1}, \tag{54}$$

$$\alpha_{2i+1} = \beta_{2i}\beta_{2i+1} + \frac{1}{2}\beta_{2i+1} \sum_{j=0, j\neq i-1, i}^{\frac{n}{2}} \beta_{2j} \tag{55}$$

with an extra requirement such that

$$\begin{aligned} \beta_0(\sum_{j=0, j\neq i-1, i}^{\frac{n}{2}} \beta_{2j+1}) &= \beta_1(\sum_{j=0, j\neq i-1, i}^{\frac{n}{2}} \beta_{2j}) \\ &= \ldots \\ &= \beta_{\frac{n}{2}-1}(\sum_{j=0, j\neq i-1, i}^{\frac{n}{2}} \beta_{2j+1}). \end{aligned} \tag{56}$$

The above definition of α blending function is consistent to the original definition of β blending function expressed by Eq. (23).

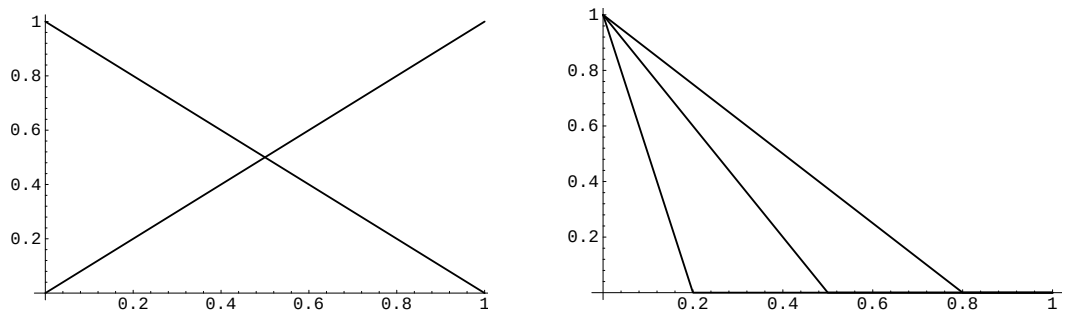
4. Designing β Blending Function

In this section, we will show exmaples of triangular, rectangular, and pentagonal patches. In each case, the design method by changing the value of β blending function is shown.

4.1. The rectangular patch

We deal with the rectangular patch as an example of even-sided patches. As discussed in subsection , $\beta_i(u, v)$, $i = 0, \ldots, 3$ are

$$\beta_0(u, v) = 1 - u, \quad \beta_1(u, v) = 1 - v, \quad , \beta_2(u, v) = u, \quad \beta_3(u, v) = v. \tag{57}$$



(a) The original function (b) Changed β blending functions

Figure 3: Designing β blending function

Each function is of two variables u and v, but it depends on only one variable and it can be described in a 2D graph as shown in Fig.3(a). This property of β blending function gives designers enough easiness and freedom to manipulate the blending functions when the designers want to change the value of the blending functions instead of the shapes of the boundary curves of the patch. For example, Fig.4(a) shows the original rectangular patch using Eq. (57). By changing β_0 from u to

$$\beta_0(u,v) = \begin{cases} 1 - \frac{u}{a} & 0 \le u \le a \\ 0 & a < u \le 1 \end{cases} \tag{58}$$

where a is a given value such that $0 < a < 1$, which is shown in Fig.3(b) for some given value of a, we obtain new shapes shown in Fig.4(b), (c), and (d) for $a = 0.8$, 0.5, and 0.2, respectively. The designers can modify the shape easily by changing one variable such a. Note that from Eq. (47),

$$\beta_2(u,v) = 1 - \beta_0(u,v). \tag{59}$$

Therefore $\beta_2(u,v)$ should be ajusted according to the change of $\beta_0(u,v)$ as

$$\beta_2(u,v) = \begin{cases} \frac{u}{a} & 0 \le u \le a \\ 1 & a < u \le 1 \end{cases} \tag{60}$$

Even though $\beta_0(u,v)$ as well as $\beta_2(u,v)$ have been changed, the same $\beta_1(u,v)$ and $\beta_3(u,v)$ can be used.

4.2. The triangular patch

First, we have to define a local coordinate system for the triangular domain. Let Ω' be a triangle of height unity and with vertices $V_i = (u_i, v_i)$, $i = 0, 1, 2$ in the (u,v)

plane. It is possible to write the general point $V = (u, v)$ as a barycentric combination of V_0, V_1, and V_2:

$$V = rV_0 + sV_1 + tV_2. \tag{61}$$

The variables (r, s, t) are well known as barycentric coordinates of V with respect to V_0, V_1, and V_2. Here we use this system as a local coordinate system of the triangular domain. Note that $r + s + t = 1$.

From Eq. (12), $\alpha_i(r, s, t)$, $i = 0, 1, 2$ are given as

$$\begin{aligned} \alpha_0(r, s, t) &= \frac{r}{r + s + t} = r, & (62)\\ \alpha_1(r, s, t) &= s, & (63)\\ \alpha_2(r, s, t) &= t. & (64) \end{aligned}$$

Then $\beta_i(r, s, t)$, $i = 0, 1, 2$ are given as

$$\begin{aligned} \beta_0(r, s, t) &= r + s = 1 - t, & (65)\\ \beta_1(r, s, t) &= s + t = 1 - r, & (66)\\ \beta_2(r, s, t) &= t + r = 1 - s. & (67) \end{aligned}$$

Fig.5(a) shows the triangular patch using the above blending functions.

By changing $\beta_0(r, s, t)$ from $1 - t$ to

$$\beta_0(u, v) = \begin{cases} 1 - \frac{t}{a} & 0 \le t \le a \\ 0 & a < t \le 1 \end{cases}, \tag{68}$$

$\beta_1(r, s, t)$ and $\beta_2(r, s, t)$ from $1 - r$ and $1 - s$ to

$$\begin{aligned} \beta_1(u, v) &= \begin{cases} 1 - r + \frac{r}{r+s}\frac{t(1-a)}{a} & 0 \le t \le a \\ 1 & a < t \le 1 \end{cases}, & (69)\\ \beta_2(u, v) &= \begin{cases} 1 - s + \frac{s}{r+s}\frac{t(1-a)}{a} & 0 \le t \le a \\ 1 & a < t \le 1 \end{cases}. & (70)\\ & & (71) \end{aligned}$$

respectively. In the above equations, the term $\frac{t(1-a)}{a}$ comes from the difference of the original and the changed $\beta_0(r, s, t)$, i.e.

$$(1 - t) - (1 - \frac{t}{1}) = \frac{t(1 - a)}{a}. \tag{72}$$

The patches using these blending functions are shown in Fig.5(b), (c), and (d) corresponding to different a values. Compare them with the patch using the original

blending function shown in (a).

4.3. The pentagonal patch

Using the local coordinate system defined in the the subsection with Eq. (12), we can define $\beta_0(V)$ as

$$\begin{aligned} \beta_0(V) &= \alpha_0(V) + \alpha_1(V) \\ &= 1 - (\alpha_2(V) + \alpha_3(V) + \alpha_4(V)) \quad (73) \\ &= 1 - \gamma_0(V) \quad (74) \end{aligned}$$

since $\sum_{i=0}^{4} \alpha_i(V) = 1$ where

$$\gamma_0(V) = \alpha_2(V) + \alpha_3(V) + \alpha_4(V) \quad (75)$$

Similarly, $\beta_i(V)$, $i = 1, \ldots, 4$ can be defined as

$$\begin{aligned} \beta_1(V) &= 1 - (\alpha_3(V) + \alpha_4(V) + \alpha_0(V)) = 1 - \gamma_1(V), \quad (76) \\ \beta_2(V) &= 1 - (\alpha_4(V) + \alpha_0(V) + \alpha_1(V)) = 1 - \gamma_2(V), \quad (77) \\ \beta_3(V) &= 1 - (\alpha_0(V) + \alpha_1(V) + \alpha_2(V)) = 1 - \gamma_3(V), \quad (78) \\ \beta_4(V) &= 1 - (\alpha_1(V) + \alpha_2(V) + \alpha_3(V)) = 1 - \gamma_4(V). \quad (79) \end{aligned}$$

Fig.6(a) shows the pentagonal patch using the above blending functions.

By using the same functions $\beta_4(V)$ and $\beta_1(V)$ and changing $\beta_0(V)$, $\beta_2(V)$ and $\beta_3(V)$ to

$$\beta_0(V) = \begin{cases} 1 - \frac{\gamma_0(V)}{a} & 0 \le \gamma_0(V) \le a \\ 0 & a < \gamma_0(V) \le 1 \end{cases}, \quad (80)$$

$$\beta_2(V) = \begin{cases} 1 - \gamma_2(V) + \frac{\delta(V)}{\delta(V)+\epsilon(V)} \frac{\gamma_0(V)(1-a)}{a} & 0 \le \gamma_0(V) \le a \\ 1 - \gamma_2(V) + \frac{\delta(V)}{\delta(V)+\epsilon(V)}(1 - \gamma_0(V)) & a < \gamma_0(V) \le 1 \end{cases}, \quad (81)$$

$$\beta_3(V) = \begin{cases} 1 - \gamma_3(V) + \frac{\epsilon(V)}{\delta(V)+\epsilon(V)} \frac{\gamma_0(V)(1-a)}{a} & 0 \le \gamma_0(V) \le a \\ 1 - \gamma_3(V) + \frac{\epsilon(V)}{\delta(V)+\epsilon(V)}(1 - \gamma_0(V)) & a < \gamma_0(V) \le 1 \end{cases} \quad (82)$$

respectively, where $\delta(V) = \alpha_2(V)$ and $\epsilon(V) = \alpha_4(V)$. The patches using these blending functions are shown in Fig.6(b), (c), and (d) corresponding to different a values. Compare them with the patch using the original blending function shown in (a).

4.4. Modeling example

Fig.7 shows a modeling exmaple using a cubic β blending function with a controlable variable a. The model consists of 6 rectangular and 12 triangular patches.

5. Conclusions

By introducing the edge-based blending function derived from the surface-based

one, we can capture the effects of the edges on the shape of the n-sided patch more directly. Because of the good natures of the edge based blending function, the designers can manipulate it as well as the geometry of the boundary curves to design and modify the patch shape.

In this paper, we have discussed the derivation of the edge-based blending function, its properties, especially the differences between ones for odd-sided and for even-sided patches, and easiness of designing it by the users. We have demonstrated its validty by showing several triangular, rectangular, and pentagonal patchs using the modifiable blending functions.

In the future work, methods should be investigated how to connect several patches with G^1 or G^2 continuity as well as G^0 treated in this paper. To develped a CAD/CAGD system based on the edge-based blending function, data structures should be constructed to implement its scheme efficiently. It may be possible to convert the n-sided patch proposed in the paper to a general surface representation like generalized Bézier or B-spline surfaces.

6. References

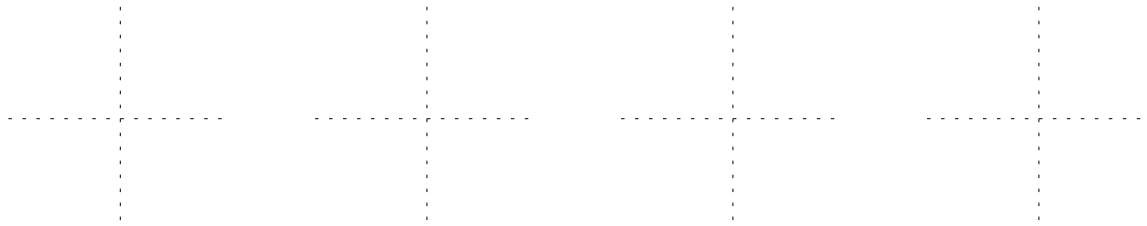
1. C. Blanc and C. Schlick : X-Splines: A Spline Model Designed for the Rnd-User, Computer Graphics'95, 1995, pp.377-386.
2. P. Charrot and J.A. Gregory : A Pentagonal Surface Patch for Computer Aided Geometric Design, CAGD, Vol.1, pp.87-94 (1984)
3. J.A. Gregory : C^1 Rectangular and Non-rectangular Surface Patches, In Surfaces and CAGD, R.E. Barnhill and W. Boehm, Eds. North-Holland, Amsterdam, 1983, pp.25-33.
4. C.M. Grimm and J.F. Hughes : Modeling Surfaces of Arbitrary Topologiy using Manifolds, Computer Graphics'94, 1995, pp.359-368.
5. M. Hosaka and F. Kimura : Non-four-sided Patches Expressions with Control Points. CAGD, Vol.1, No.1, 1984, pp.75-86.
6. M.A. Sabin : Non-rectangular Surface Patches Suitable For Inclusion in a B-spline Surface, Proceedings of Eurographics'83, P. Ten Hagen, Ed. North-Holland, Amsterdam, 1983, pp.57-69.
7. C. Loop and T. DeRose : A Multisided Generalization of Bézier Surfaces, ACM Transactions on Graphics, 8(3):204-234, (1989).
8. C. Loop and T. DeRose : Generalized B-spline Surfaces of Arbitrary Topology, Computer Graphics, Vol.24, No.4, 1990, pp.347-356.
9. C. Loop : Smooth Spline Surfaces over Irregular Meshes, Computer Graphics'94, 1994, pp.303-310.

(a) Original (b) $a = 0.8$ (c) $a = 0.5$ (d) $a = 0.2$

Fig. 4. Rectabgular Patch Examples

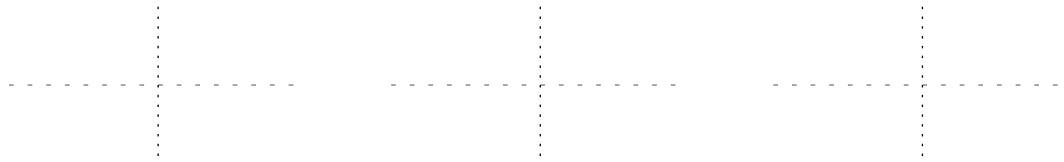
(a) Original (b) $a = 0.8$ (c) $a = 0.5$ (d) $a = 0.2$

Fig. 5. Triangular Patch Examples



(a) Original (b) $a = 0.8$ (c) $a = 0.5$ (d) $a = 0.2$

Fig. 6. Pentagonal Patch Examples



(a) $a = 0.9$ (b) $a = 0.5$ (c) $a = 0.1$

Fig. 7. A Plane