Blending Implicit Shapes Using Fuzzy Set Operations

Qingde Li
University of Hull
Department of Computer Science
HULL, HU6 7RX
UK
q.li@hull.ac.uk

Jie Tian
The Chinese Academy of Sciences
Institute of Automation
Beijing 100080
China
tian@doctor.com

Abstract: Implicit modelling is a powerful technique to design geometric shapes, where a geometric object is described by a real function. In general, the real functions used in implicit modelling are unbounded and can take any values in space R. In general, the shapes described by different level sets of an unbounded implicit function can be varied significantly and are very unpredictable. In addition, allowing the underlying implicit function to take negative values also makes the construction of shape blending operations a difficult task. In this paper, we propose an implicit shape modelling technique, where each implicit shape is represented as the membership function of a fuzzy set which is bounded and nonnegative with value taken in [0, 1]. The most obvious benefit of representing an implicit shape as fuzzy set is that the blending of a set of implicit shapes is simply a problem of aggregating a set of fuzzy sets, which can be done in various ways by choosing a proper fuzzy set aggregation operator from a wide variety of fuzzy set operations.

Key–Words: Implicit curves and surfaces, isosurfaces, blending operations, Generalized algebraic operations, Piecewise algebraic operations, Fuzzy sets, Soft computing

1 Introduction

In computer aided geometric modelling, parametric surfaces and polygonal meshes have long been used as the major representation forms of geometric shapes, due to their advantages in tessellation and rendering. However, parametric surfaces and polygonal meshes are not convenient forms of shape representation in terms of shape composition, shape matching[3] and recognition [10], collision detection, and ray tracing. In many practical applications, implicitly representing geometric shapes as real functions are preferred. With the dramatic increases in computational power of modern computers in both their CPUs and GPUs, realtime implicit shape rendering is becoming a reality.

Describing a geometric shape implicitly as a real function is not something new. In fact, it is a well investigated research subject within the realm of mathematics. However, it is not until the publication of Ricci's work [9] in 1973 that it becomes a popular shape modelling technique in computer graphics related areas. Compared with parametric shape modelling, many difficult and time consuming computational tasks, such as collision detection, soft shape blending and shape matching, can be solved in a much more efficient and more effective way.

Depending on the view angles, implicit shape modelling is frequently referred to as functional based method, isosurface or level set method. Mathematically, an implicit shape is described by a real function $F : \mathcal{R}^n \to \mathcal{R}, (n = 2, 3)$, which in general partitions the space $\mathcal{R}^n$ into three parts corresponding to $F(\mathbf{P}) < 0$, $F(\mathbf{P}) = 0$, and $F(\mathbf{P}) > 0$ respectively. Therefore, a real function F naturally associates with a geometric shape either by representing its boundary as the real roots of the equation $F(\mathbf{P}) = 0$ or as a solid defined by $\{\mathbf{P} : F(\mathbf{P}) \geq 0\}$ or by $\{\mathbf{P} : F(\mathbf{P}) \leq 0\}$. By viewing a solid shape as a collection of points, a complex solid geometric object can be considered to be the result of performing a sequence of set-theoretic operations over a set of simple solid geometric primitives.

When each geometric primitive is characterized by a real function, the key to this technique is to define blending operations that combine a set of functions representing simple geometric objects into a single function representing the required geometric shape[11]. The simplest and the most natural blending operation is the one corresponding exactly to the Boolean operations $\max(x, y)$ or $\min(x, y)$. Assume that the two solid objects A and B are represented implicitly by functions $F_A(\mathbf{P})$ and $F_B(\mathbf{P})$ respectively as $A = \{\mathbf{P} : F_A(\mathbf{P}) \geq 0\}$ and $B = \{\mathbf{P} : F_B(\mathbf{P}) \geq 0\}$. Then, the union $A \cup B$, the intersection $A \cap B$, and the difference $A - B$

of set A and B can be represented by functions $\max\{F_A(\mathbf{P}), F_B(\mathbf{P})\}$, $-\max\{-F_A(\mathbf{P}), -F_B(\mathbf{P})\}$, and $-\max\{-F_A(\mathbf{P}), F_B(\mathbf{P})\}$ respectively. Depending on the problem of application, different kinds of real functions may require different function composition operations. For instance, when the underlying real functions of a set of implicit shapes represent certain density distribution functions, it is reasonable to blend these implicit shapes by finding the sum of these functions directly. When the real functions used to specify implicit shapes represent the drops of certain fluid, the shape combination operations developed based on the fluid's physical properties are more appropriate. In this paper, we aim to develop a fuzzy set based implicit modelling technique, where each implicit function involved in the modelling process is considered to be a fuzzy set.

There are several reasons why it is desirable to model a geometric shape as a fuzzy set. Firstly, representing implicit shapes as fuzzy sets allow us to treat the underlying implicit functions naturally as a kind of point set. Depending on the problem of applications, an ordinary fuzzy set operation can be chosen to blend implicit shapes from a wide variety of well established fuzzy set operations. Secondly, as each fuzzy set associates with a real function which is bounded and nonnegative with value taking in [0, 1], fuzzy sets represented implicit shapes can be considered as normalized implicit functions. As a result, implicit shapes can be blended more fairly and naturally. Thirdly, the level values used to generate different level sets are all in [0, 1], which is much more convenient in computing various level sets of a given implicit function. Moreover, compared with unbounded real functions, fuzzy set represented implicit functions are in general much less sensitive to numerical error when extracting their underlying geometry.

In the rest part of the paper, we first briefly discuss how to model a geometric shape as a fuzzy set. We then focus on the development of piecewise polynomial fuzzy set operations and demonstrate the strengths of these operations with implicit shape design examples.

In the following discussion, we will identify a fuzzy set with its membership function and all the fuzzy sets discussed in the paper are thought to be in the space $R^n (n = 1, 2, 3)$.

2 Implicit shapes as fuzzy sets

In this section, we consider how to represent a general implicit shape as a fuzzy set.

2.1 Converting a general real function into a fuzzy set

For a given real function $f(P) : R^n \to R$ that represents a geometric shape, there are several ways to convert it directly into a fuzzy set. This is a process to convert a mapping from R^n to R into a mapping from R^n to $[0, 1]$. This can be done easily by compounding function f with a smooth unit step function $\mu(r)$ that satisfies the following conditions:

1. $\mu(r)$ is smooth and nondecreasing.
2. $\lim_{r\to-\infty} \mu(r) = 0, \lim_{r\to\infty} \mu(r) = 1.$
3. $\mu(0) = 0.5.$

Such a kind of function can be constructed in one of several ways. One simple function that satisfies conditions listed above is the function given below:

$$H_\infty(x) = \frac{1}{1 + e^{-\alpha x}}, \tag{1}$$

where $\alpha > 0$ is a parameter specifying the rising gradient of the function around 0.

The drawback of using H_∞ to convert a general real function into a fuzzy set is that it will lead to a non-piecewise polynomial representation of the shape.

A better way is to use a low order piecewise polynomial smooth unit step function introduced in [6]. Piecewise polynomial smooth unit step function can be construct iteratively from the Heaviside unit step function $H_0(x)$ in the following way:

$$\begin{aligned} H_0(x) &= \begin{cases} 0, & x < 0; \\ \frac{1}{2}, & x = 0; \\ 1, & x > 0. \end{cases} \\ H_n(x) &= \frac{1}{2}(\ (1 + \frac{x}{n})H_{n-1}(x + 1) \\ &\qquad +(1 - \frac{x}{n})H_{n-1}(x - 1)\ ), \\ &\qquad n = 1, 2, 3, \cdots . \end{aligned} \tag{2}$$

It can be shown that $H_n(x)$ has the following properties:

(1) $H_n(x)$ is C^{n-1}-continuous for $n > 1$.

(2) $H_n(x)$ is a piecewise-polynomial function.

(3) $H_n(x)$ is monotonically increasing and takes value 1 when $x \geq n$, and 0 when $x \leq -n$.

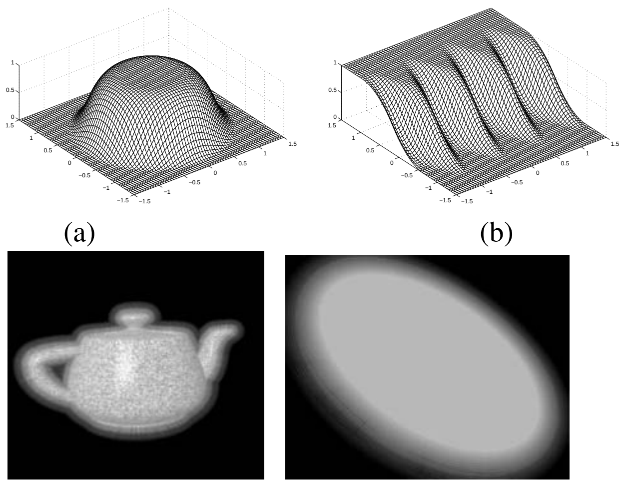
(4) $H_n(x)$+$H_n(-x) = 1$, $H_n(0) = \frac{1}{2}$.

For an arbitrary number $R > 0$, a smooth unit step function with supporting range $[-R, R]$ can be defined as follows:

$$H_{n,R}(x) = H_n(nx/R). \quad (3)$$

Let $H = F(P)$ be a real function and $\mu(r)$ a smooth unit step function. Then the shape corresponding to $F(P) \geq 0$ is equivalent to the level-set of fuzzy set membership function $\mu(F(P)) \geq 0.5$.

Figure 1 shows some 2D and 3D fuzzy sets obtained by converting a real function into a fuzzy set membership function using a smooth unit step function.



(a) (b)

(c) (d)

Figure 1: Fuzzy shapes obtained by converting a known geometry using a unit step function. (a) and (b): 2D fuzzy sets; (c) and (d): 3D fuzzy sets.

2.2 Fuzzy spline functions

Implicit shapes can also be designed directly as fuzzy sets using implicit splines. With implicit spline technique, geometric shapes are constructed out from implicit spline basis functions $\{B_k(P)\}$, where each $B_k(P)$ is a real function on shape space $R^n (n = 1, 2, 3)$ satisfying the following properties:

1. $0 \leq B_k(P) \leq 1$.
2. For each variable parameter P, $\sum_k B_k(P) = 1$.
3. Each $B_k(P)$ is nonnegative and piecewise polynomial.

There are various techniques to construct a set of implicit spline basis functions. Figure 2 shows some examples of 2D spline basis functions constructed from an arbitrarily specified 2D polygonal net that partitions the 2D plane using the algorithm described in [7].

Now consider the implicit shapes defined in the following way:

$$F(P) = \sum L_k(P)B_k(P),$$

where each $L_k(P)$ is a local implicit shape associated to $B_k(P)$. If each $L_k(P)$ is represented as a fuzzy set membership function, it can be seen directly that $F(P)$ is also a mapping from R^n to $[0, 1]$ and thus can again be considered as the membership function of a fuzzy set. The implicit shapes displayed in Figure 3 and 4 is generated in this way. Readers who are interested in the implicit spline modelling technique can refer to the work presented in [7].

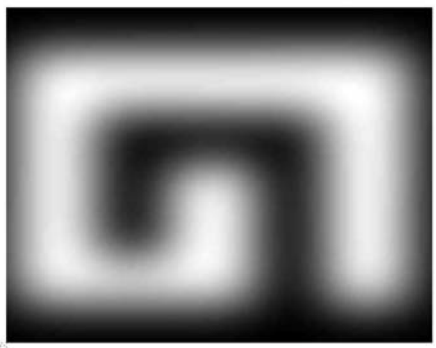


Figure 3: A 2D fuzzy shape designed using implicit spline basis functions.

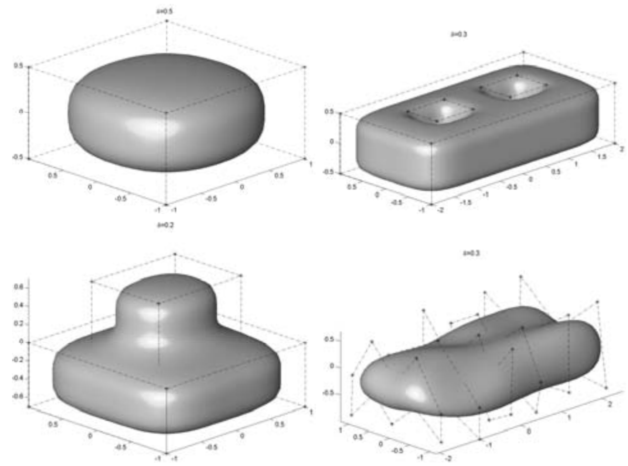


Figure 4: Fuzzy set represented freeform implicit surfaces designed using implicit spline basis functions.

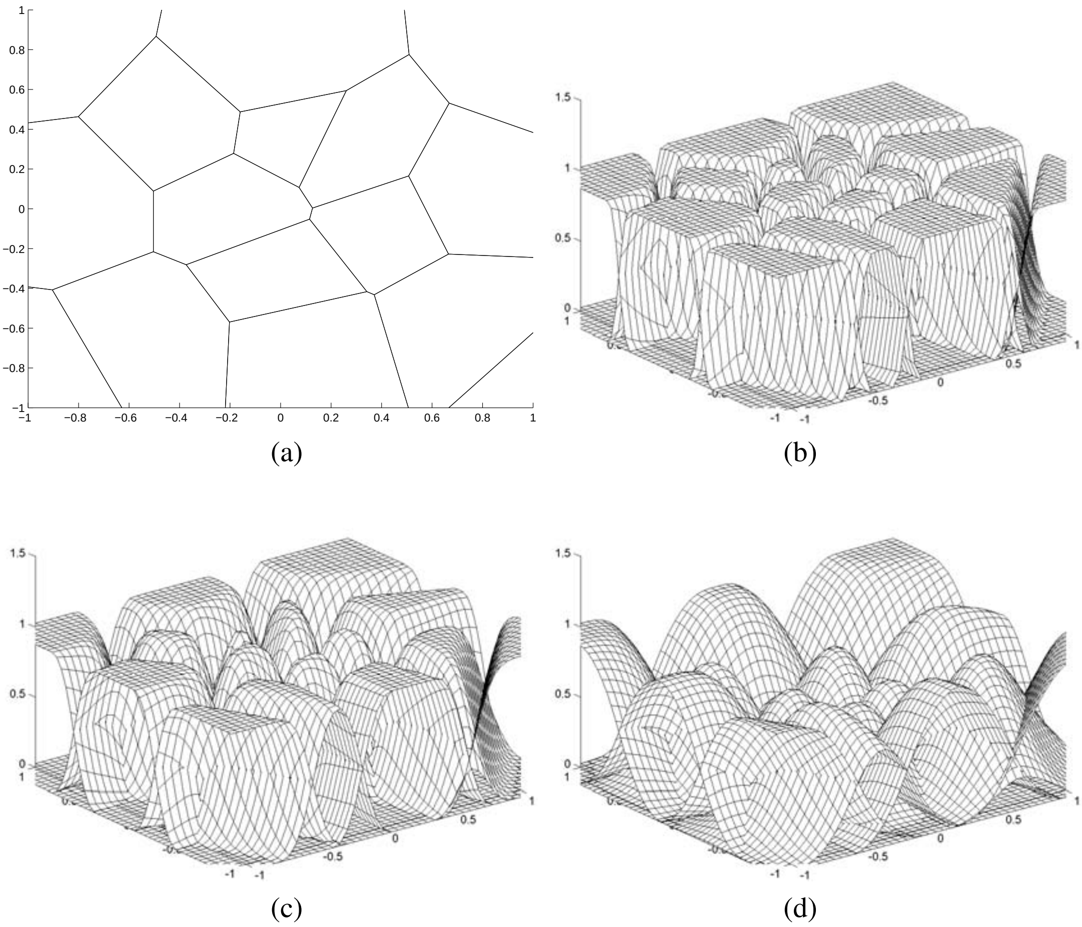


Figure 2: The C^2-smooth implicit spline bases functions created from an arbitrarily specified partition net of 2D plane, with different values of polygon smoothing parameter α.

3 Fuzzy geometric objects blending

When the underline geometric objects are modelled implicitly as fuzzy sets, they can be blended easily using fuzzy set operations. In this section, we discuss the issues relating to the combination of simple fuzzy geometric objects.

When two geometric objects are represented as fuzzy sets $\widetilde{A}$ and $\widetilde{B}$, we can find their union, intersection and difference as usual. One important issue needs to be addressed here is the choice of the fuzzy set operation. One typical feature of the most commonly used fuzzy objects is that their membership functions are smooth to reflect the nature of fuzziness. However, as can be seen from Table 1, most of the conventional fuzzy sets operations are not smooth. The main problem of using non-smooth fuzzy set operations is that they may lead to non-smooth fuzzy sets. Another significant feature of fuzzy sets is that they are frequently described by a piecewise polynomial membership function. Thus algebraic or piecewise algebraic operations are preferred since the fuzzy sets produced by these kinds of operations will also be described by piecewise polynomial membership functions. The requirement that the fuzzy set operations should be smooth and piecewise polynomial becomes even more important when the fuzzy sets involved in the operation are used to represent geometric shapes. This is because the implicit functions used for geometric design are usually smooth and represented in piecewise polynomials.

The simplest and the most natural continuous piecewise polynomial binary operation is the one defined by the minimum function $\min(x, y)$ [14] (see Figure 5 (a)). It is continuous and piecewise linear algebraic over the domain $[0, 1] \times [0, 1]$. Moreover, when it is used to combine piecewise algebraic fuzzy sets, not only the output fuzzy sets remain to be piecewise algebraic, but the degree of polynomials will also be maintained. Although various alternatives to $\min(x, y)$ have been proposed, it is still the most frequently used fuzzy set operator in both theory and practical applications. However, $\min(x, y)$ is not smooth enough as it is not differentiable along

Table 1: Some notable binary operations for defining the intersection operation for fuzzy sets

algebraic	$g(x, y) = xy$
Zadeh	$g(x, y) = min(x, y)$
Lukasiewicz	$g(x, y) = max(0, x + y - 1)$
Dubois-Prade	$g(x, y) = \frac{xy}{\max(x,y,k)}, k \in (0, 1]$
Einstein	$g(x, y) = \frac{xy}{2-(x+y-xy)}$
Dombi[2]	$g(x, y) = \begin{cases} \frac{1}{1+((x^{-1}-1)^n+(y^{-1}-1)^n)^{\frac{1}{n}}}, & x, y > 0, n > 0; \\ 0, & \text{x=0, y=0.} \end{cases}$
Hamacher[4]	$g(x, y) = \frac{xy}{a+(1-a)(x+y-xy)}, a \in [0, 1)$
Weber[12]	$g(x, y) = \max(0, \frac{x+y+kxy-1}{1+k}), k \in (-1, \infty)$
Yager[13]	$g(x, y) = 1 - \min(1, ((1 - x)^k + (1 - y)^k)^{\frac{1}{k}}), k \in (1, \infty)$

the line $y = x$. The simplest differentiable piecewise polynomial binary operation for defining the intersection operation for fuzzy sets is the algebraic operation $f(x, y) = xy$ (see Figure 5 (b)). Though the algebraic operation possesses some nice properties and is differentiable everywhere, it does not approximates the function $\min(x, y)$ well and is not widely used as an effective fuzzy set aggregation operator in practice.

3.1 The Bézier algebraic operators

The gap between the binary operation $\min(x, y)$ and the algebraic operation $f(x, y) = xy$ can be filled up by the Bézier algebraic operators of different degrees.

Consider the degree n Bézier surface $\mathbb{B}_n(x, y)$ defined over $[0, 1] \times [0, 1]$ in the following way:

$$\begin{aligned} \mathbb{B}_n(x, y) &= \sum_{k=1}^{n} \frac{k}{n} B_k^{(n)}(x) B_k^{(n)}(y) \\ &+ \sum_{j=1}^{n} \sum_{i=j+1}^{n} \frac{j}{n} \left(B_i^{(n)}(x) B_j^{(n)}(y) \right. \\ &\qquad \left. + B_i^{(n)}(y) B_j^{(n)}(x) \right), \\ &\qquad n = 1, 2, \cdots \end{aligned} \tag{4}$$

where $B_k^{(n)}(t) = C_n^k (1 - t)^{n-k} t^k$ is the k^{th} Bézier function of degree n.

We call the binary operation $\mathbb{B}_n(x, y)$ defined above over [0, 1]x[0, 1] the degree n Bézier operator.

An interesting thing about $\mathbb{B}_n(x, y)$ is that when $n = 1$, $\mathbb{B}_1(x, y) = xy$, which is just the conventional algebraic operator. Therefore, $\mathbb{B}_n(x, y)$ can be referred to as the generalized algebraic operator of degree n. The first three Bézier operators can be written out directly as follows:

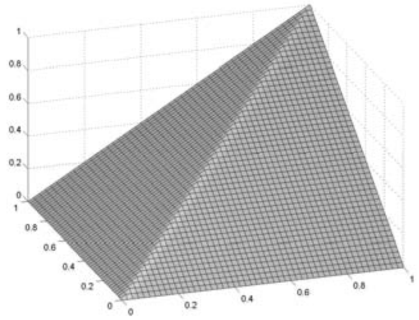
$$\mathbb{B}_1(x, y) = xy, \tag{5}$$
$$\mathbb{B}_2(x, y) = xy(1 + \overline{x}\,\overline{y}), \tag{6}$$
$$\mathbb{B}_3(x, y) = xy(1 + \overline{x}\,\overline{y}(1 + xy + \overline{x}\,\overline{y})), \tag{7}$$

where $\overline{x} = 1 - x, \overline{y} = 1 - y$.

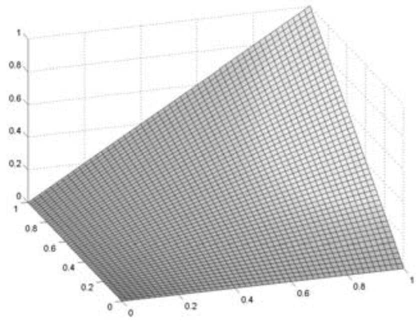
In fact, it can be shown that all the binary operators $\mathbb{B}_n(x, y)$ have similar properties to the conventional algebraic operation. For instance, we can show immediately that

1. $\mathbb{B}_n(0, 0) = \mathbb{B}_n(x, 0) = \mathbb{B}_n(0, y) = 0$.
2. $\mathbb{B}_n(1, 1) = 1, \mathbb{B}_n(x, 1) = x, \mathbb{B}_n(1, y) = y$.
3. $\mathbb{B}_n(x, y) \leq min(x, y)$.
4. $\mathbb{B}_n(x, y) = \mathbb{B}_n(y, x)$.

To approximate $\min(x, y)$ with Bézier operator precisely, a very big n is required and it can be very expensive to evaluate $\mathbb{B}_n(x, y)$. Thus, the use of high degree Bézier operator is not recommended in practice, though they have very good mathematical properties.



(a)



(b)

Figure 5: (a). The minimum operation: $g(x, y) = \min(x, y)$; (b). The algebraic operation: $g(x, y) = xy$;

Figure 6 shows the shapes of the Bézier operators of different orders. In theory, it can be shown that

$$\begin{aligned} \mathbb{B}_1(x, y) &= xy \le \mathbb{B}_2(x, y) \le \mathbb{B}_3(x, y) \le \\ &\cdots \le \sup_n \mathbb{B}_n(x, y) = min(x, y). \end{aligned}$$

As is shown in Figure 7, the degree of Bézier operators can be used to specify the blending range of two fuzzy set defined implicit shapes.

3.2 Piecewise algebraic shape preserving fuzzy geometric blending

Generalized algebraic operator $\mathbb{B}_n(x, y)$ provides us a good fuzzy shape blending operation. It is simple to compute and always produces smooth membership functions as long as the fuzzy shapes involved in the operation are all smooth. The main drawback of generalized algebraic operation is that it is not shape preserving in terms of controlling the blending range when the degree of the algebraic operation is small. When two shapes are combined using the algebraic operation, no matter whether it is a union, intersection, or subtraction, no parts of the newly obtained shape will be the same as those of the original shapes. The operations defined using $\min(x, y)$ are shape preserving but they are not smooth.

The shape preserving feature of a blending operation plays a crucial role in implicit modelling when parts of geometric primitives are reconstructed from real data or a complex shape is constructed procedurally. In these cases, one would hope that those shapes built previously should be kept unchanged as much as possible as one could. In addition, the tessellation of blended implicit surfaces obtained from shape-preserving blending operations can be made much more effectively and efficiently as many parts of the blended shapes can be identical with those implicit shapes involved in the blending. Thus, many polygons in the meshes representing the primitive implicit surfaces can be reused to construct the polygonal mesh corresponding to the blended shapes. In the past few years, several smooth shape preserving blending operations have been proposed [5][8][1][6]. However, these binary operations are not fuzzy set specific operations as they may not always produce a proper fuzzy set membership function when they are applied to fuzzy sets.

In this section, we present two constructive methods to define smooth piecewise algebraic operations for combining two fuzzy shapes with specified blending range, namely $\wedge_{n,\delta}(x, y)$ and $\widehat{\wedge}_{n,\delta}(x, y)$. The main features of these fuzzy set operations are that they can be constructed to have whatever degree of smoothness required and to approximate Zadeh's minimum functions to any required precision. In addition, the binary operator $\widehat{\wedge}_{n,\delta}(x, y)$ can be specified to preserve $\min(x, y)$ to any required extent. More precisely, $\widehat{\wedge}_{n,\delta}(x, y)$ can be defined to be identical to the ordinary binary operation $\min(x, y)$ in a subregion of $[0, 1]^2$, with the area of the subregion to be $(1 - 3\delta)^2, \delta \in (0, 1/2]$. These novel operations are all described by recursively defined functions and can be implemented cheaply and elegantly. The main idea used here is the introduction of smooth absolute functions $|x|_n$ and $\widehat{|x|}_n$, both of which are generalized from the conventional absolute function $|x|$. With smooth absolute functions, the conventional minimum function $\min(x, y)$ is extended directly to smooth ones, namely the degree n smooth minimum functions $min_{n,\delta}(x, y)$ and $\widehat{min}_{n,\delta}(x, y)$, where $\delta > 0$ is a parameter used to control the smooth fusion range along the line $y = x$. These two kinds of minimum functions are then modified further respectively to build the binary operations $\wedge_{n,\delta}(x, y)$ and $\widehat{\wedge}_{n,\delta}(x, y)$ on $[0, 1]$ for fuzzy set aggregation. As will be seen later, $\wedge_{n,\delta}(x, y)$ and $\widehat{\wedge}_{n,\delta}(x, y)$ have the following properties:

1. Both become the conventional Boolean AND op-

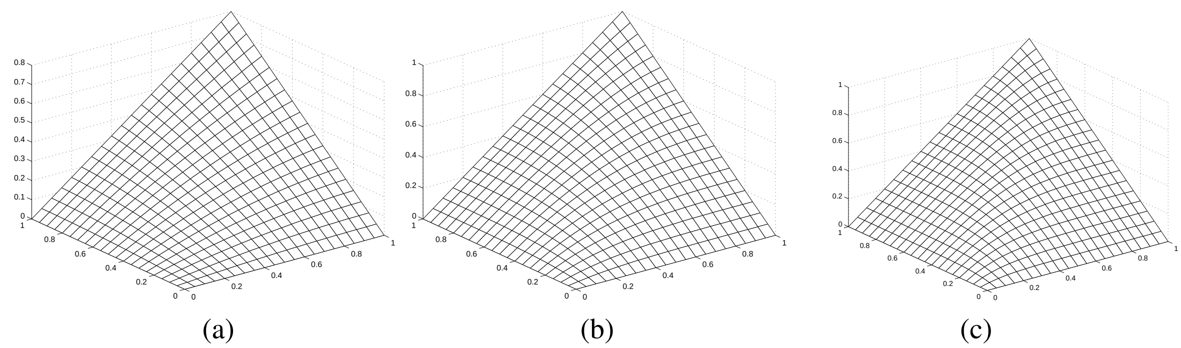


Figure 6: The shape of $\mathbb{B}_n(x,y)$ for $n = 3, 5, 10$. (a). $\mathbb{B}_3(x,y)$; (b). $\mathbb{B}_5(x,y)$; (c). $\mathbb{B}_{10}(x,y)$.

eration when confined on $\{0,1\}$;

2. $\wedge_{n,\delta}(1,x) = \wedge_{n,\delta}(x,1) = x$, $\widehat{\wedge}_{n,\delta}(1,x) = \widehat{\wedge}_{n,\delta}(x,1) = x$;

3. $\wedge_{n,\delta}(0,x) = \wedge_{n,\delta}(x,0) = 0$, $\widehat{\wedge}_{n,\delta}(0,x) = \widehat{\wedge}_{n,\delta}(x,0) = 0$;

4. $\wedge_{n,\delta}(x,y)$ is C^n-smooth, and $\widehat{\wedge}_{n,\delta}(x,y)$ is C^{n-1}-smooth.

5. $\wedge_{n,\delta}(x,y)$ and $\widehat{\wedge}_{n,\delta}(x,y)$ can be set to approximate $\min(x,y)$ with any specified precision by adjusting parameter δ.

The construction of these binary operations will be outlined below. The detailed discussion on these operators can be found in [6].

Definition 1 *Let $|x| : \mathbb{R} \to \mathbb{R}$ be the conventional absolute function. That is, $|x| = x$ when $x \geq 0$ and $|x| = -x$ when $x < 0$. Then we introduce the following generalized absolute functions:*

$$\begin{aligned} |x|_0 &= |x|; \\ |x|_n &= \frac{1}{2(n+1)}\big(\ (n-x)|1-x|_{n-1} \\ &\qquad +(n+x)|1+x|_{n-1}\ \big). \qquad (8) \\ &\qquad n = 1, 2, 3, \cdots \end{aligned}$$

$|x|_n$ is called degree n upper absolute function.

It can be shown that $|x|_n$ has the following properties

Proposition 3.1 *(1) $|x|_n \geq |x|$; and $|x|_n = |x|$ when $|x| \geq n$;*

(2) $|x|_n$ is C^n-continuous;

(3) $|x|_n$ is a piecewise polynomial function.

Due to property 3.1, $|x|_n$ can be considered as the generalization to the conventional absolute function $|x|$. From figure 8, we can see that all functions $|x|_n, n = 1, 2, \cdots$, take a similar form to function $|x|$.

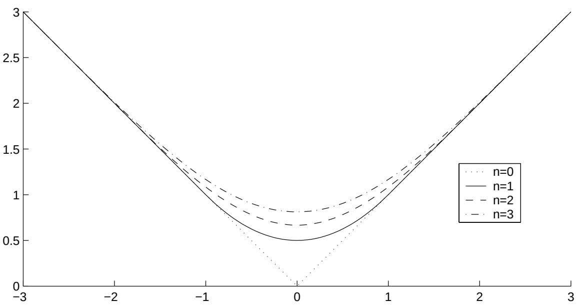


Figure 8: Smooth piecewise polynomial upper absolute functions $|x|_n$ with $n = 0, 1, 2, 3$

For smooth absolute function $|x|_n$, the difference between $|x|_n$ and $|x|$ is in the range $[-n, n]$. In the following discussion, we will refer this range of difference as the span of $|x|_n$. Smooth absolute functions with an arbitrary span $[-\delta, \delta]$ ($\delta > 0$) can be easily introduced using $|x|_n$.

Definition 2 *For $\delta > 0$ and $n > 0$, we define*

$$|x|_{n,\delta} = \frac{\delta}{n}\left|\frac{nx}{\delta}\right|_n. \qquad (9)$$

It can be shown immediately that $|x|_{n,\delta} = |x|$ when $|x| \geq \delta$.

Smooth piecewise polynomial functions that approximate the conventional absolute function can also be introduced using the smooth unit step function $H_n(x)$ introduced in section 2.1.

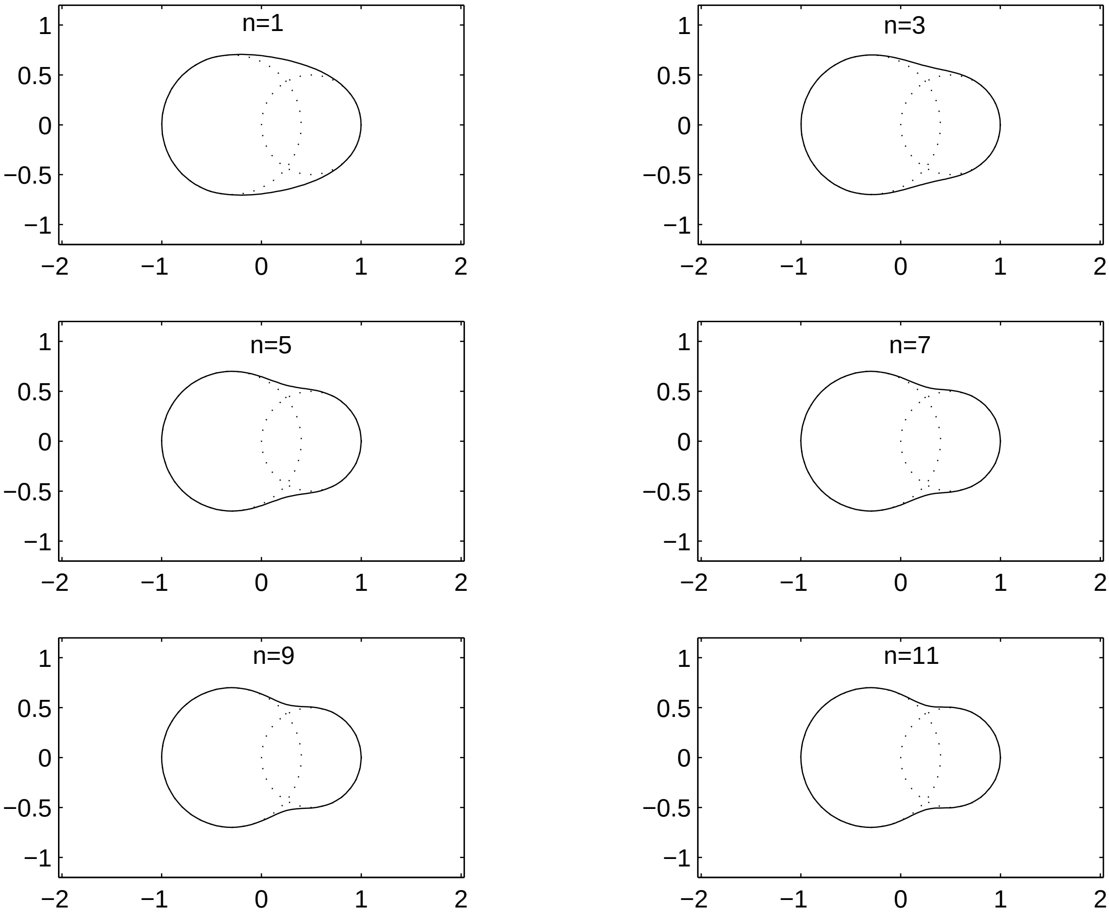


Figure 7: Blending two fuzzy set defined implicit shapes using different degrees Bézier operations $\mathbb{B}_n(x, y)$.

Definition 3 *Consider the sequence of functions defined in the following way using smooth unit step function $H_n(x)$:*

$$\widehat{|x|}_n = 2xH_n(x) - x, \quad n = 0, 1, 2, \cdots . \tag{10}$$

We call $\widehat{|x|}_n$ the degree n lower absolute function.

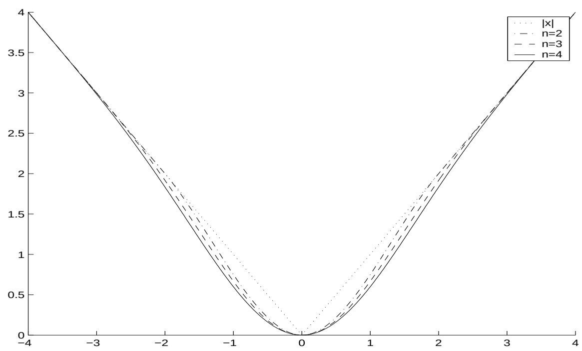


Figure 9: Degree n lower absolute functions $\widehat{|x|}_n$.

The shapes of the first a few lower smooth absolute functions are shown in Figure 9. As can be seen from figure 9, each of this type of function also takes a similar shape to $|x|$. Unlike $|x|_n$, the value of $\widehat{|x|}_n$ can never be larger than $|x|$.

Note that $H_n(x) = 1$ when $x > n$, and $H_n(x) = 0$, when $x < -n$, we can see that

$$\widehat{|x|}_n = |x| \quad \text{when} \quad |x| > n.$$

Secondly, since $H_n(x) \geq \frac{1}{2}$ when $x \geq 0$, and $H_n(x) < \frac{1}{2}$ when $x < 0$, we have $\widehat{|x|}_n \geq 0$. Furthermore, $\widehat{|x|}_n$ is a C^{n-1} continuous piecewise polynomial function.

As with $|x|_{n,\delta}$, lower absolute function with an arbitrary span $\delta > 0$, denoted by $|\widehat{x}|_{n,\delta}$, can be introduced similarly in the following way:

Definition 4 *For $\delta > 0$ and $n > 0$, we define*

$$\widehat{|x|}_{n,\delta} = \frac{\delta}{n} \widehat{\left|\frac{nx}{\delta}\right|}_n. \tag{11}$$

It can be shown directly that $\widehat{|x|}_{n,\delta} = |x|$ when $|x| \geq \delta$.

Note that

$$\inf_n |x|_{n,\delta} = |x|, \qquad \sup_n \widehat{|x|}_{n,\delta} = |x|.$$

Thus, it is reasonable to call $|x|_{n,\delta} = |x|$ the upper smooth absolute function and $\widehat{|x|}_{n,\delta}$ the lower smooth absolute function.

The above two smooth absolute functions can be used to define what we called the smooth minimum functions.

Definition 5 *For $\delta > 0$, let*

$$\begin{aligned} min_{n,\delta}(x,y) &= \frac{1}{2}(x + y - |x-y|_{n,\delta}); \quad (12) \\ \widehat{min}_{n,\delta}(x,y) &= \frac{1}{2}(x + y - \widehat{|x-y|}_{n,\delta}). \quad (13) \end{aligned}$$

We call $min_{n,\delta}(x,y)$ the degree n lower smooth minimum function and $\widehat{min}_{n,\delta}(x,y)$ the degree n upper smooth minimum function, where δ is a parameter referred to as the approximation accuracy to the minimum operation $\min(x,y)$.

These two functions have first been introduced in [6] for implicit shape modelling with controllable blending range. However, they are not in general properly defined fuzzy set operations. For instance, neither of the two function satisfies the Identity Law and the Dominance Law. In addition, function $min_{n,\delta}(x,y)$ may take values outside the interval $[0,1]$. Therefore, they cannot be used to define fuzzy operations directly. In spite of this, the two bivariate functions can be easily modified into fuzzy set operations[6].

Let us first consider how to modify $\min_{n,\delta}(x,y)$. From the definition of $\min_{n,\delta}(x,y)$, we can see that its four boundary curves can be described by using the following two functions:

$$\begin{aligned} C_0(x) &= \frac{1}{2}(x - |x|_{n,\delta}) \quad (14) \\ C_1(x) &= \frac{1}{2}(1 - x - |1-x|_{n,\delta}). \quad (15) \end{aligned}$$

These four curves define a Coon's surface patch, which can be expressed in the following form:

$$S(x,y) = S_1(x,y) + S_2(x,y) - S_3(x,y), \quad (16)$$

where

$$\begin{aligned} S_1(x,y) &= (1-y)C_0(x) + yC_1(x), \quad (17) \\ S_2(x,y) &= (1-x)C_0(y) + xC_1(y), \quad (18) \\ S_3(x,y) &= \alpha(1 - x - y + 2xy), \quad (19) \end{aligned}$$

and $\alpha = -|0|_{n,\delta} < 0$.

The smooth minimum function $\min_{n,\delta}(x,y)$ can then be modified by subtracting the Coons patch defined above:

$$\wedge_{n,\delta}(x,y) = min_{n,\delta}(x,y) - S(x,y); \quad (20)$$

It can be shown immediately that

$$\begin{aligned} \wedge_{n,\delta}(0,x) &= \wedge_{n,\delta}(x,0) = 0, \\ \wedge_{n,\delta}(1,x) &= \wedge_{n,\delta}(x,1) = x. \end{aligned}$$

Figure 10 displays the shapes of $\wedge_{2,\delta}(x,y)$ for different δ values. As can be seen from the figure, $\wedge_{n,\delta}(x,y)$ can well approximate $\min(x,y)$ as long as the value of δ is small enough.

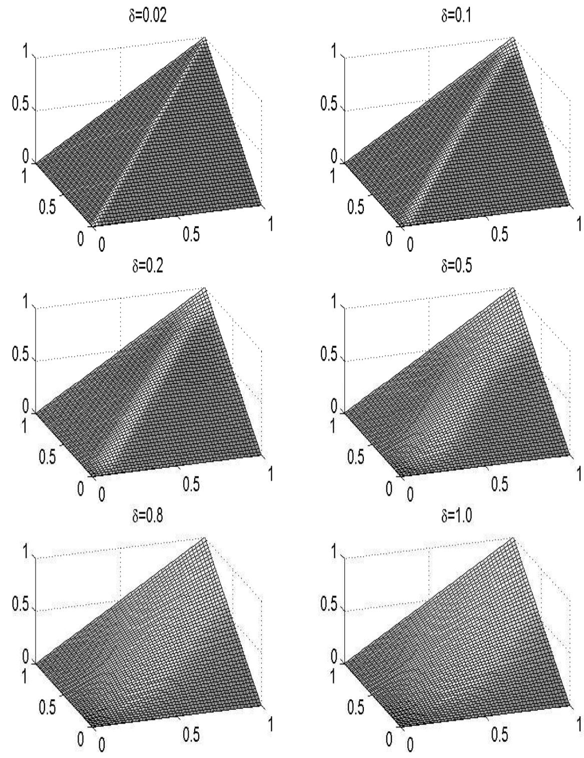


Figure 10: The plot of the C^2-smooth binary operation $\wedge_{2,\delta}(x,y)$ with different values of parameter δ.

As with $min_{n,\delta}(x,y)$, $\widehat{min}_{n,\delta}(x,y)$ can also be modified into a fuzzy set operation. Let

$$\begin{aligned} \widehat{C}_0(x) &= xH_n(-\frac{nx}{\delta}) \quad (21) \\ \widehat{C}_1(x) &= (1-x)H_n(-\frac{n(1-x)}{\delta}) \quad (22) \end{aligned}$$

and let

$$\begin{aligned} \widehat{S}_1(x,y) &= (1-y)\widehat{C}_0(x) + y\widehat{C}_1(x), \quad (23) \\ \widehat{S}_2(x,y) &= (1-x)\widehat{C}_0(y) + x\widehat{C}_1(y). \quad (24) \end{aligned}$$

With $\widehat{S}_1(x,y)$ and $\widehat{S}_2(x,y)$, $\widehat{min}_{n,\delta}(x,y)$ can then be modified in the following way:

$$\widehat{\wedge}_{n,\delta}(x,y) = \widehat{min}_{n,\delta}(x,y) - (\widehat{S}_1(x,y) + \widehat{S}_2(x,y)).$$

It can be shown that $\widehat{\wedge}_{n,\delta}(x,y)$ satisfies the Identity Law and the Dominance Law.

Figure 11 displays the shapes of $\widehat{\wedge}_{n,\delta}(x,y)$ for different δ values. As can be seen from the figure, $\widehat{\wedge}_{n,\delta}(x,y)$ can also well approximate $\min(x,y)$ as long as the value of δ is small enough.

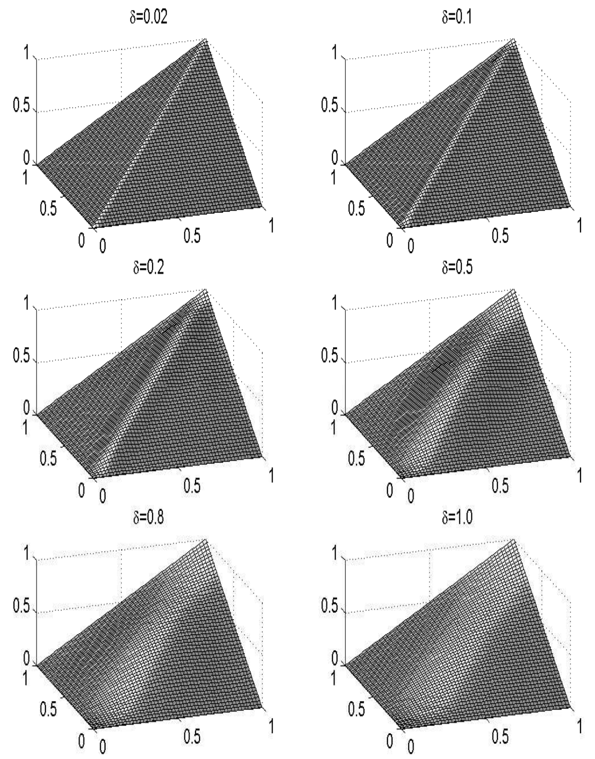
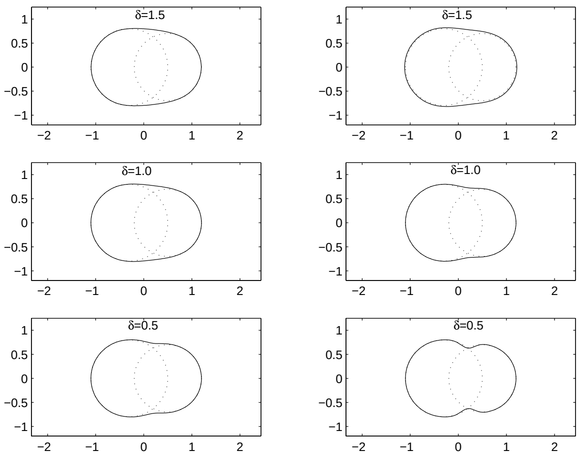


Figure 11: The plot of the C^2-smooth binary operation $\widehat{\wedge}_{3,\delta}(x,y)$ with different values of parameter δ.

Both of these two operations can be used for combining two fuzzy shapes, where their initial shape features can be partially preserved. This can be done easily by choosing a proper value for parameter δ. The differences between the blending operations defined using $\wedge_{n,\delta}(x,y)$ and that defined by $\widehat{\wedge}_{n,\delta}(x,y)$ can be observed by displaying the blended shapes for 2D implicit shapes. As can be seen from Figure 12, for a relatively small value of the blending range parameter, $\widehat{\wedge}_{n,\delta}(x,y)$ based union operation tends to retain more of the original shapes of the geometric objects involved in the blending operation than that defined using $\wedge_{n,\delta}(x,y)$.

Figure 13 shows the shapes corresponding to the unions of two fuzzy set specified spheres $S_1(x,y,z) = 0.5$ and $S_2(x,y,z) = 0.5$, where the



(a) (b)

Figure 12: The unions of two implicit circles based on $\wedge_{n,\delta}(x,y)$(left column) and $\widehat{\wedge}_{n,\delta}(x,y)$ (right column) with $\delta = 1.5, 1.0, 0.5$ respectively .

unions are defined by the fuzzy set $F(x,y,z) = 1 - \wedge_{2,\delta}(1 - S_1(x,y,z), 1 - S_2(x,y,z))$ using $\wedge_{2,\delta}(x,y)$ with δ=1.5, 1.0, and 0.5 respectively. As can be seen from the figure, the blending range of two fuzzy set defined implicit shapes can be well controlled using parameter δ. Similar blending effect can be obtained for $\widehat{\wedge}_{n,\delta}(x,y)$ based fuzzy set operations.

4 Conclusion

In this paper, we proposed a fuzzy set based implicit modelling paradigm. With the proposed method, any implicit shape can be thought of as a level set of certain fuzzy solid in the shape space. Under this shape design paradigm, a complex geometric shape can be constructed by combining a set of simple geometric shapes described by fuzzy set membership functions. Compared with implicit modelling based on general real functions, fuzzy set based implicit modelling is much more intuitive and natural. In addition, three new fuzzy set operations have been developed to meet the the basic requirements of fuzzy solid modelling. All these operations are designed to approximate Zadeh's $\min(x,y)$ operation. They are algebraic or piecewise algebraic and can be constructed up to any required degree of smoothness. A few examples have been given to show that the proposed shape design scheme is not only theoretically solid, they are also effective and flexible to use in practice.

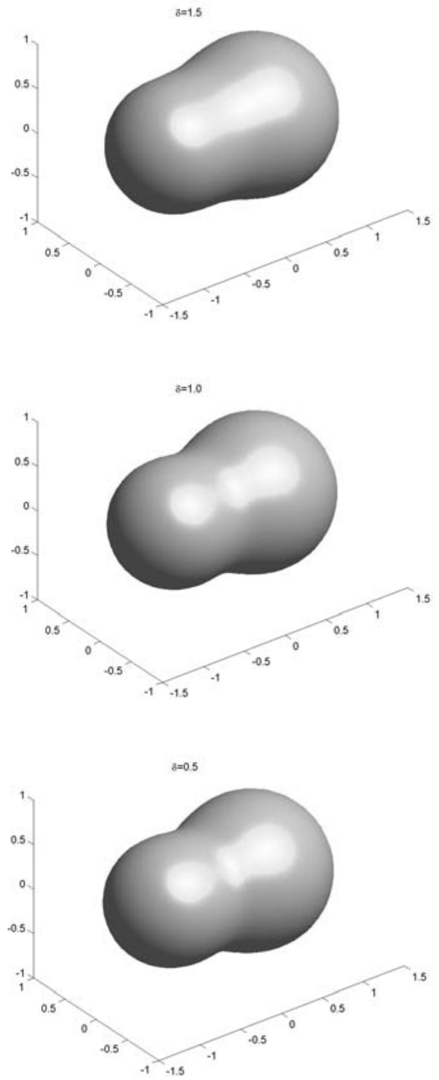


Figure 13: The unions of two implicit sphere based on fuzzy set operation $\wedge_{n,\delta}(x, y)$ with $\delta = 1.5, 1.0, 0.5$ respectively.

References:

[1] L. Barthe, N. A. Dodgson, M. A. Sabin, B. Wyvill, and V. Gaildrat, *Two-dimentional potential fields for advanced implicit modeling operators*, Computer Graphics Forum **22** (2003), no. 1, 23–33.

[2] J. Dombi, *Basic concepts for a theory of evaluation: The aggregative operator*, European Journal of Operational Research (1982), 282–293.

[3] Y. Ebrahim, W. Abdelsalam, M. Ahmed, and S-C. Chau, *Shape representation and matching of 3d objects for computer vision applications*, WSEAS Transactions On Information Science And Applications **2** (2005), no. 9, 1329–1336.

[4] H. Hamacher, *On logical connectives of fuzzy statements and their affiliated true functions*, Proceedings of the third European meeting on Cybernetics and Systems Research (Vienna), 1976.

[5] P. C. Hsu and C. Lee, *The scale method for blending operations in functionally-based constructive geometry*, Computer Graphics Forum **22** (2002), no. 2, 143–158.

[6] Q. Li, *Smooth piecewise polynomial blending operations for implicit shapes*, Computer Graphics Forum **26** (June 2007), 157–171(15).

[7] Q. Li and J. Tian, *2d piecewise algebraic splines for implicit modeling*, Technical Report, Department of Computer Science, University of Hull, 2008.

[8] G. Pasko, A. A. Pasko, M. Ikeda, and T. L. Kunii, *Bounded blending operations*, IEEE Computer Graphics and Applications **25** (2005), 36–45.

[9] A. Ricci, *(1973) A constructive geometry for computer graphics*, Computer Journal **16**, no. 3, 157–160.

[10] M. Rodrigues, A Robinson, L. Alboul, and W. Brink, *3d modelling and recognition*, WSEAS Transactions on Information Science and Applications **3** (2006), 2118.

[11] Zhe Wang and Mohammed Latif, *Reconstruction of 3d solid models using fuzzy logic recognition*, WSEAS Transaction on Circuits and systems **3** (2004), 1018.

[12] S. Weber, *A general concept of fuzzy connectives, negations and implications*, Fuzzy Sets and Systems **11** (1983), 115–134.

[13] R.R. Yager, *On a general class of fuzzy connectives*, Fuzzy Sets and Systems **4** (1980), 235–242.

[14] Lotfi A. Zadeh, *Fuzzy sets*, Information and Control **8** (1965), no. 3, 338–353.