

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Approach to Representation of Type-2 Fuzzy Sets Using Computational Methods of Computer Graphics

Long Thanh Ngo and Long The Pham

Department of Information Systems, Faculty of Information Technology, Le Quy Don, Technical University, Vietnam

1. Introduction

The type-2 fuzzy sets was introduced by L. Zadeh as an extension of ordinary fuzzy sets. So the concept of type-2 fuzzy sets is also extended from type-1 fuzzy sets. If A is a type-1 fuzzy set and membership grade of $x \in X$ in A is $\mu_A(x)$, which is a crisp number in $[0, 1]$. A type-2 fuzzy set in X is \tilde{A} , and the membership grade of $x \in X$ in \tilde{A} is $\mu_{\tilde{A}}(x)$, which is a type-1 fuzzy set in $[0, 1]$. The elements of the domain of $\mu_{\tilde{A}}(x)$ are called *primary memberships* of x in \tilde{A} and the memberships of the primary memberships in $\mu_{\tilde{A}}(x)$ are called *secondary memberships* of x in \tilde{A} .

Recently, there are many researches and applications related to type-2 fuzzy sets because of the advancing in uncertainty management. Karnik et al (2001A) proposed practical algorithms of operations on type-2 fuzzy sets as union, intersection, complement. Karnik et al (2001B) proposed the method of type-reduction of type-2 fuzzy sets based on centroid defuzzification. Mendel et al (2002) have developed new representation of type-2 fuzzy sets based on embedded type-2 fuzzy sets. This representation easily have designing of type-2 fuzzy logic system is easy to use and understand. Mendel (2004), Liu (2008) proposed some practical algorithms in implementing and storing data to speed-up the computing rate of type-2 fuzzy logic systems. Coupland et al (2007), Coupland et al (2008A), Coupland et al (2008B) proposed representation type-1 and interval type-2 fuzzy sets and fuzzy logic system by using computational geometry, the fast approach to geometric defuzzification of type-2 fuzzy sets, the approach is better in computing than analytic approaches. TIN is a method of representation of curved surface in 3D space for many applications in computer graphics and simulation. Many approaches Shewchuck (2002), Ruppert (1997), Chew (1989) are use to generate TIN from set of points based Delaunay algorithms.

The chapter deals with the new representation of type-2 fuzzy sets using TIN. The membership grades of type-2 fuzzy sets in 3D surfaces that are discretized into triangular faces with planar equations. Size of triangle is difference depending on slope of the surface. Authors proposed practical algorithms to implement operations on type-2 fuzzy sets by designing computational geometry algorithms on TIN. The result is shown and corroborated for robustness of the approach, rendering type-2 fuzzy sets in 3-D environment using OpenSceneGraph SDK.

The chapter is organized as follows: II presents TIN and geometric computation; III introduces type-2 fuzzy sets; IV presents approximate representation of type-2 fuzzy sets; V is operations of TIN and geometric operations of type-2 fuzzy sets; VI is conclusion and future works.

2. TIN and geometric computation

2.1 Delaunay triangulation

A *topographic surface* v is the image of a real bivariate function f defined over a domain D in the Euclidean plane, as

$$v = \{(x, u, f(x, u)) | (x, u) \in D\} \quad (1)$$

A polyhedral model is the image of a piecewise-linear function f that is described on a partition of D into polygonal regions $\{D_1, \dots, D_k\}$ and the image of f over each region $D_i (i = 1, \dots, k)$ is a linear patch. If all D_i s ($i = 1, \dots, k$) are triangles then the polyhedral model is called a *Triangulated Irregular Network* (TIN). Hence, v may be represented approximately by a TIN, as

$$v \cong \sum_{i=1}^k \{(x, u, f_i(x, u)) | (x, u) \in T_i\}, \bigcup_{i=1}^k T_i \equiv D \quad (2)$$

where f_i s ($i = 1, \dots, k$) are planar equations.

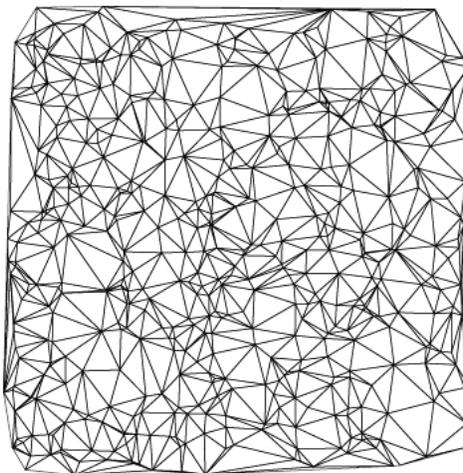


Fig. 1. A Delaunay Triangulation

The *Delaunay triangulation* of a set V of points in \mathbb{R}^2 is a subdivision of the convex hull of V into triangles that their vertices are at points of V , and such that triangles are as much equiangular as possible. More formally, a triangulation τ of V is a Delaunay triangulation if and only if, for any triangle t of τ , the circumcircle of t does not contain any point of V in its interior. This property is called the *empty circle property* of the Delaunay triangulation. Let u and v be two vertices of V . The edge uv is in D if and only if there exists an empty circle that passes through u and v . An edge satisfying this property is said to be *Delaunay*. Figure 1 Chew (1989) illustrates a Delaunay Triangulation.

An alternative characterization of the Delaunay triangulation is given based on the *max – min angle property*. Let τ be a triangulation of V . An edge e of τ is said to be *locally optimal* if and only if, given the quadrilateral Q formed by the two triangles of τ adjacent to e , either Q is not convex, or replacing e with the opposite diagonal of Q (*edge flip*) does not increase

the minimum of the six internal angles of the resulting triangulation of Q . τ is a Delaunay triangulation if and only if every edge of τ is locally optimal. The repeated application of edge flips to non-optimal edges of an arbitrary triangulation finally leads to a Delaunay triangulation.

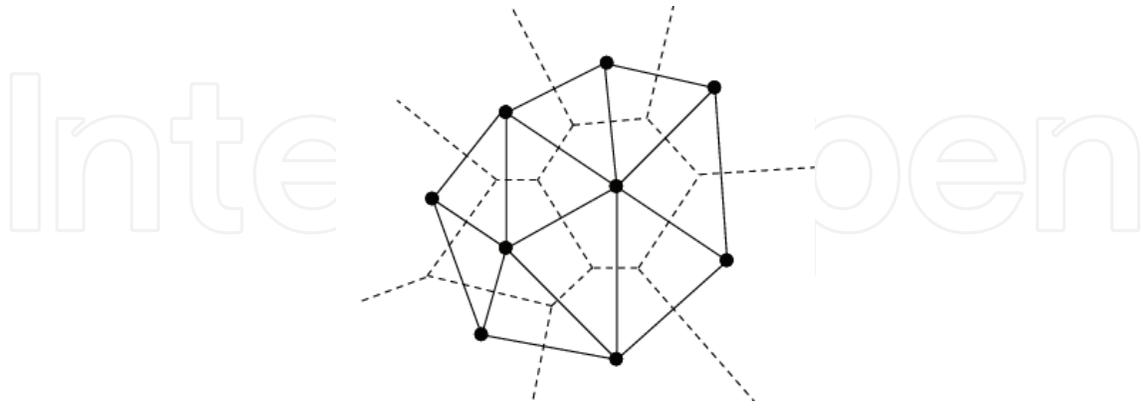


Fig. 2. The Delaunay triangulation (solid lines) and the Voronoi diagram (dash lines) from a point set.

The geometric dual of the Delaunay triangulations is the *Voronoi diagram*, which describes the proximity relationship among the point of the given set V . The *Voronoi diagram* of a set V of points is a subdivision of the plane into convex polygonal regions, where each region is associated with a point P_i of V . The region associated with P_i is called *Voronoi region* of P_i , and consists of the locus of points of the plane which lie closer to P_i than any other point in V . Two points P_i and P_j are said to be *Voronoi neighbours* when the corresponding Voronoi regions are adjacent. Figure 2 shows the Delaunay triangulation and the Voronoi diagram from a point set.

The usual input for two-dimensional mesh generation is not merely a set of vertices. Most theoretical treatments of meshing take as their input a planar straight line graph (PSLG). A PSLG is a set of vertices and segments that satisfies two constraints. First, for each segment contained in a PSLG, the PSLG must also contain the two vertices that serve as endpoints for that segment. Second, segments are permitted to intersect only at their endpoints. A set of segments that does not satisfy this condition can be converted into a set of segments that does. Run a segment intersection algorithm, then divide each segment into smaller segments at the points where it intersects other segments.

The *constrained Delaunay triangulation* (CDT) of a PSLG X is similar to the Delaunay triangulation, but every input segment appears as an edge of the triangulation. An edge or triangle is said to be *constrained Delaunay* if it satisfies the following two conditions. First, its vertices are *visible* to each other. Here, visibility is deemed to be obstructed if a segment of X lies between two vertices. Second, there exists a circle that passes through the vertices of the edge or triangle in question, and the circle contains no vertices of X that are visible from the interior of the edge or triangle.

The flip algorithm begins with an arbitrary triangulation, and searches for an edge that is not *locally Delaunay*. All edges on the boundary of the triangulation are considered to be locally Delaunay. For any edge e not on the boundary, the condition of being locally Delaunay is similar to the condition of being Delaunay, but only the two triangles that contain e are considered. For instance, Figure 4 demonstrates two different ways to triangulate a subset of

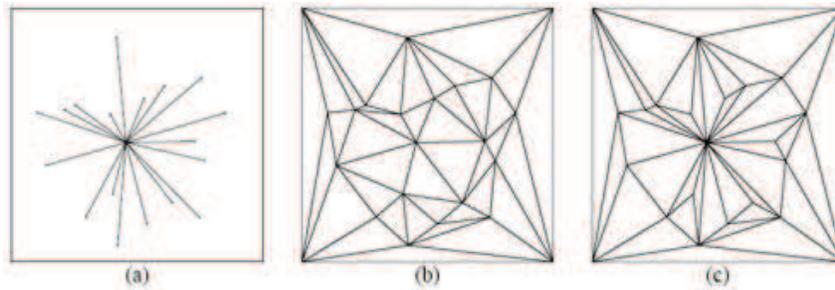


Fig. 3. (a) A planar straight line graph. (b) Delaunay triangulation of the vertices of the PSLG. (c) Constrained Delaunay triangulation of the PSLG.

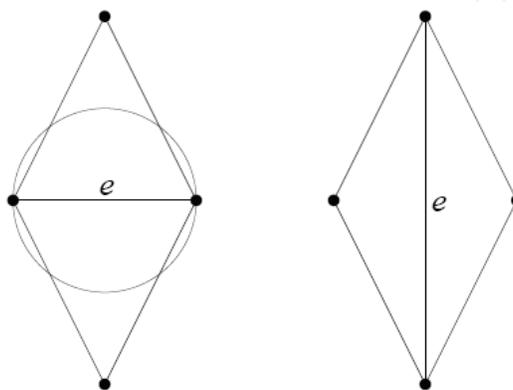


Fig. 4. Two triangulations of a vertex set. At left, e is locally Delaunay; at right, e is not.

four vertices. In the triangulation at left, the edge e is locally Delaunay, because the depicted containing circle of e does not contain either of the vertices opposite e in the two triangles that contain e . In the triangulation at right, e is not locally Delaunay, because the two vertices opposite e preclude the possibility that e has an empty containing circle. Observe that if the triangles at left are part of a larger triangulation, e might not be Delaunay, because vertices may lie in the containing circle, although they lie in neither triangle. However, such vertices have no bearing on whether or not e is locally Delaunay.

Whenever the flip algorithm identifies an edge that is not locally Delaunay, the edge is flipped. To flip an edge is to delete it, thereby combining the two containing triangles into a single containing quadrilateral, and then to insert the crossing edge of the quadrilateral. Hence, an edge flip could convert the triangulation at left in Figure 4 into the triangulation at right, or vice versa.

2.2 Half edge data structure and basic operations

A common way to represent a polygon mesh is a shared list of vertices and a list of faces storing pointers for its vertices. The half-edge data structure is a slightly more sophisticated boundary representations which allows all of the queries listed above to be performed in constant time. In addition, even though we are including adjacency information in the faces, vertices and edges, their size remains fixed as well as reasonably compact.

The half-edge data structure is called that because instead of storing the edges of the mesh, storing half-edges. As the name implies, a half-edge is a half of an edge and is constructed by splitting an edge down its length. Half-edges are directed and the two edges of a pair have

opposite directions. Data structure of each vertex v in TIN contains a clockwise ordered list of half edges gone out from v . Each half edge $h = (eV, lF)$ contains end vertex (eV) and index of right face (lF). Suppose that a TIN has m faces and n vertices, it needs to have n lists of $3m$ half edges and memory is $n * (3 * m) * (2 * 4)$ bytes. Figure 5 shows data structure of vertex v with 6 half-edges indexed from 0 to 5, the i^{th} half-edge contains the vertex v_i and the right face f_i of the edge.

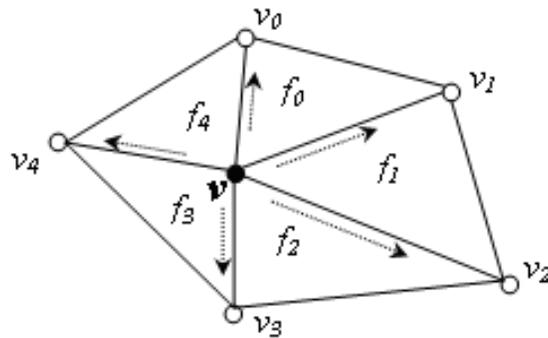


Fig. 5. List of half-edges of a vertex.

Some operations are built based on half-edges such as edge collapse operation, flip operation, insertion or deletion operation... The following is description of half-edge based algorithms.

Algorithm 2.1 (Insertion Operation). *Insert a new half edge h into the list of vertex v .*

Input: *The list of half edge of vertex v and new vertex eP .*

Output: *The new list of half edge of vertex v .*

1. Identity i in the list of half edges of v so that the ray (v, eP) is between two rays (v, v_i) and (v, v_{i+1}) .
2. Move $k - i$ half edges from position i to $i + 1$ in the list.
3. Insert the half edge h into position i .

Figure 6 depicts an example of *edge collapse* operation after deleting the edge (v_0, v_1) from V . The first step of edge collapse is to identity indices i_0, i_1 of half edges h_0, h_1 in the lists of half edges of v_0, v_1 , respectively. Then moving half edges $(v_1, v_4), (v_1, v_5)$ of vertex v_1 into the list of v_0 at i_0 , rejecting half edges $h_0, (v_3, v_1), (v_6, v_1)$, setting the endpoint of half edges $(v_4, v_1), (v_5, v_1)$ to be v_0 . The following is the algorithm for edge collapse:

Algorithm 2.2 (Edge Collapse). *Remove the edge (v_0, v_1) and vertex v_1 from TIN.*

Input: *TIN T , edge (v_0, v_1) , vertex v_1 .*

Output: *TIN T' is the collapsed TIN.*

1. Identity i_0, i_1 of half edges h_0, h_1 in lists of v_0, v_1 , respectively.
2. Copy half edges of v_1 from position $i + 2$ to $i - 2$ (if exist) in the list to the list of half edges of v_0 at i_0 . Then set endpoint of respective inverse half edges is v_0 .
3. Delete half edges from position $i - 1$ to $i + 1$ of v_1 and their inverse half edges.
4. Delete vertex v_1 and its related data.

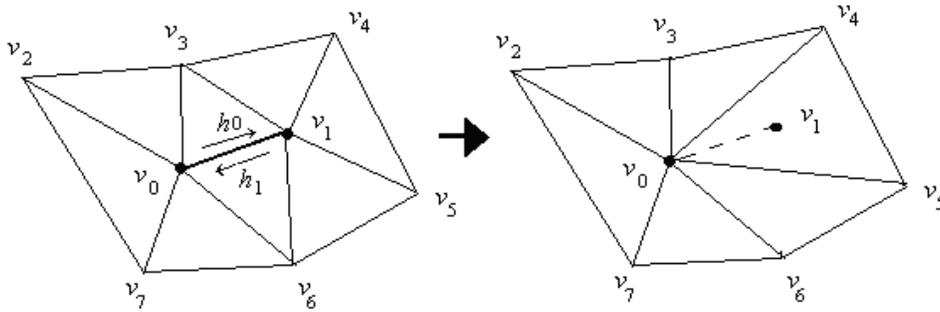


Fig. 6. Edge collapse.

Flip operation mentioned above is shown in Figure 7. The algorithm is applied to the edge which does not satisfy the empty circle property of Delaunay triangulation. The following is algorithm for flip operation:

Algorithm 2.3 (Flip Operation). *Flipping edge (v_0, v_1) become edge (v_2, v_3) .*

Input: TIN T , edge (v_0, v_1) .

Output: TIN T' is the flipped TIN.

1. Replace edge (v_0, v_1) become edge (v_2, v_3) in TIN.
2. Move half edges h_0, h_1 of vertices v_0, v_1 to vertices v_2, v_3 and their endpoints are v_3, v_2 , respectively.
3. Change right face of half edges $(v_0, v_3), (v_1, v_2)$.

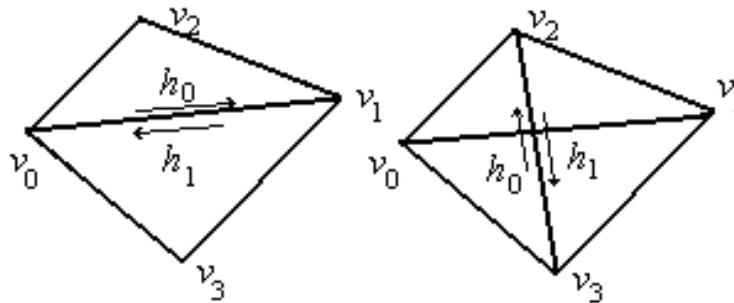


Fig. 7. Flip Operation.

3. Type-2 fuzzy sets

3.1 Fuzzy sets

Fuzzy set concept was proposed by L. Zadeh Zadeh (1975) in 1965. A fuzzy set A of a universe of discourse X is characterized by a *membership function* $\mu_A : U \rightarrow [0, 1]$ which associates with each element y of X a real number in the interval $[0, 1]$, with value of $\mu_A(x)$ at x representing the "grade of membership" of x in A .

A fuzzy set F in U may be represented as a set of ordered pairs of a generic element x and its grade of membership function: $F = \{(x, \mu_F(x)) | x \in U\}$. When U is continuous, F is re-written as $F = \int_U \mu_F(x)/x$, in which the integral sign denotes the collection of all points

$x \in U$ with associated membership function $\mu_F(x)$. When U is discrete, F is re-written as $F = \sum_U \mu_F(x)/x$, in which the summation sign denotes the collection of all points $x \in U$ with associated membership function $\mu_F(x)$.

In the same crisp theoretic set, basic operations of fuzzy set are union, intersection and complement. These operations are defined in term of their membership functions. Let fuzzy sets A and B be described by their membership functions $\mu_A(x)$ and $\mu_B(x)$. One definition of *fuzzy union* leads to the membership function

$$\mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x) \quad (3)$$

where \vee is a t -conorm, for example, maximum.

and one definition of *fuzzy intersection* leads to the membership function

$$\mu_{A \cap B}(x) = \mu_A(x) \star \mu_B(x) \quad (4)$$

where \star is a t -norm, for example minimum or product.

The membership function for *fuzzy complement* is

$$\mu_{\neg B}(x) = 1.0 - \mu_B(x) \quad (5)$$

Fuzzy Relations represent a *degree* of presence or absence of association, interaction, or interconnectedness between the element of two or more fuzzy sets. Let U and V be two universes of discourse. A fuzzy relation, $R(U, V)$ is a fuzzy set in the product space $U \times V$, i.e, it is a fuzzy subset of $U \times V$ and is characterized by membership function $\mu_R(x, y)$ where $x \in U$ and $y \in V$, i.e., $R(U, V) = \{((x, y), \mu_R(x, y)) | (x, y) \in U \times V\}$.

Let R and S be two fuzzy relations in the same product space $U \times V$. The intersection and union of R and S , which are *compositions* of the two relations, are then defined as

$$\mu_{R \cap S}(x, y) = \mu_R(x, y) \star \mu_S(x, y) \quad (6)$$

$$\mu_{R \cup S}(x, y) = \mu_R(x, y) \bullet \mu_S(x, y) \quad (7)$$

where \star is a any t -norm and \bullet is a any t -conorm.

Sup-star composition of R and S :

$$\mu_{R \circ S}(x, z) = \sup_{y \in V} [\mu_R(x, y) \star \mu_S(y, z)] \quad (8)$$

3.2 Type-2 fuzzy sets

A type-2 fuzzy set in X is denoted \tilde{A} , and its membership grade of $x \in X$ is $\mu_{\tilde{A}}(x, u), u \in J_x \subseteq [0, 1]$, which is a type-1 fuzzy set in $[0, 1]$. The elements of domain of $\mu_{\tilde{A}}(x, u)$ are called primary memberships of x in \tilde{A} and memberships of primary memberships in $\mu_{\tilde{A}}(x, u)$ are called secondary memberships of x in \tilde{A} .

Definition 3.1. A type – 2 fuzzy set, denoted \tilde{A} , is characterized by a type-2 membership function $\mu_{\tilde{A}}(x, u)$ where $x \in X$ and $u \in J_x \subseteq [0, 1]$, i.e.,

$$\tilde{A} = \{((x, u), \mu_{\tilde{A}}(x, u)) | \forall x \in X, \forall u \in J_x \subseteq [0, 1]\} \quad (9)$$

or

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x, u) / (x, u), J_x \subseteq [0, 1] \quad (10)$$

in which $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$.

At each value of x , say $x = x'$, the 2D plane whose axes are u and $\mu_{\tilde{A}}(x', u)$ is called a *vertical slice* of $\mu_{\tilde{A}}(x, u)$. A *secondary membership function* is a vertical slice of $\mu_{\tilde{A}}(x, u)$. It is $\mu_{\tilde{A}}(x = x', u)$ for $x \in X$ and $\forall u \in J_{x'} \subseteq [0, 1]$, i.e.

$$\mu_{\tilde{A}}(x = x', u) \equiv \mu_{\tilde{A}}(x') = \int_{u \in J_{x'}} f_{x'}(u) / u, J_{x'} \subseteq [0, 1] \quad (11)$$

in which $0 \leq f_{x'}(u) \leq 1$.

In manner of embedded fuzzy sets, a type-2 fuzzy sets Mendel et al (2002) is union of its type-2 embedded sets, i.e

$$\tilde{A} = \sum_{j=1}^n \tilde{A}_e^j \quad (12)$$

where $n \equiv \prod_{i=1}^N M_i$ and \tilde{A}_e^j denoted the j^{th} type-2 embedded set of \tilde{A} , i.e.,

$$\tilde{A}_e^j \equiv \{ (u_i^j, f_{x_i}(u_i^j)), i = 1, 2, \dots, N \} \quad (13)$$

where $u_i^j \in \{u_{ik}, k = 1, \dots, M_i\}$.

Let \tilde{A}, \tilde{B} be type-2 fuzzy sets whose secondary membership grades are $f_x(u), g_x(w)$, respectively. Theoretic operations of type-2 fuzzy sets such as union, intersection and complement are described Karnik et al (2001A) as follows:

$$\mu_{\tilde{A} \cup \tilde{B}}(x) = \mu_{\tilde{A}}(x) \sqcup \mu_{\tilde{B}}(x) = \int_u \int_v (f_x(u) \star g_x(w)) / (u \vee w) \quad (14)$$

$$\mu_{\tilde{A} \cap \tilde{B}}(x) = \mu_{\tilde{A}}(x) \sqcap \mu_{\tilde{B}}(x) = \int_u \int_v (f_x(u) \star g_x(w)) / (u \star w) \quad (15)$$

$$\mu_{\tilde{A}}(x) = \mu_{\neg \tilde{A}}(x) = \int_u (f_x(u)) / (1 - u) \quad (16)$$

where \vee, \star are t-cornorm, t-norm, respectively. Type-2 fuzzy sets are called an interval type-2 fuzzy sets if the secondary membership function $f_{x'}(u) = 1 \forall u \in J_x$ i.e. a type-2 fuzzy set are defined as follows:

Definition 3.2. An interval type-2 fuzzy set \tilde{A} is characterized by an interval type-2 membership function $\mu_{\tilde{A}}(x, u) = 1$ where $x \in X$ and $u \in J_x \subseteq [0, 1]$, i.e.,

$$\tilde{A} = \{ ((x, u), 1) | \forall x \in X, \forall u \in J_x \subseteq [0, 1] \} \quad (17)$$

Uncertainty of \tilde{A} , denoted FOU, is union of primary functions i.e. $FOU(\tilde{A}) = \bigcup_{x \in X} J_x$. Upper/lower bounds of membership function (UMF/LMF), denoted $\bar{\mu}_{\tilde{A}}(x)$ and $\underline{\mu}_{\tilde{A}}(x)$, of \tilde{A} are two type-1 membership function and bounds of FOU.

4. Approximate representation of type-2 fuzzy sets

Extending the concept of interval type-2 sets of upper MF and lower MF, we define a membership grade of type-2 fuzzy sets by dividing them into subsets: upper (lower) surface and normal surface as follows:

Definition 4.1 (Upper surface). \tilde{A}_{US} is called a upper surface of type-2 fuzzy set \tilde{A} and defined as follows:

$$\tilde{A}_{US} = \int_{x \in X} \left[\int_{u \in J_x^+} f_x(u)/u \right] / x \quad (18)$$

in which $J_x^+ \subseteq [u_x^+, 1]$ and $u_x^+ = \sup\{u | \mu_{\tilde{A}}(x, u) = 1\}$.

Definition 4.2 (Lower surface). \tilde{A}_{LS} is called lower surface of type-2 fuzzy set \tilde{A} and defined as follows:

$$\tilde{A}_{LS} = \int_{x \in X} \left[\int_{u \in J_x^-} f_x(u)/u \right] / x \quad (19)$$

in which $J_x^- \subseteq [0, u_x^-]$ and $u_x^- = \inf\{u | \mu_{\tilde{A}}(x, u) = 1\}$.

Definition 4.3 (Normal surface). \tilde{A}_{NS} is called a normal surface of type-2 fuzzy set \tilde{A} and defined as follows:

$$\tilde{A}_{NS} = \int_{x \in X} \left[\int_{u \in J_x^*} f_x(u)/u \right] / x \quad (20)$$

in which $J_x^* = [u_x^-, u_x^+]$.

For this reason, a type-2 fuzzy set \tilde{A} is union of above defined sub-sets, i.e. $\tilde{A} = \tilde{A}_{US} \cup \tilde{A}_{NS} \cup \tilde{A}_{LS}$. Figure 8 is an example of type-2 fuzzy set that is union of subsets: upper surface, normal surface and lower surface.

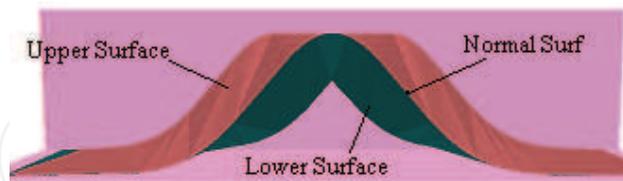


Fig. 8. Example of surfaces of type-2 fuzzy sets

A proximate representation of type-2 fuzzy sets is proposed by using a TIN that be able to approximately represent the 3-D membership function, is expressed as the following theorem.

Theorem 4.1 (Approximation Theorem). Let \tilde{A} be type-2 fuzzy set with membership grade $\mu_{\tilde{A}}(x, u)$ in continuous domain D . There exists a type-2 fuzzy set with membership grade is a TIN $T_{\tilde{A}}$, denoted \tilde{A}_T , so that \tilde{A}_T is ϵ -approximation set of \tilde{A} , i.e.,

$$\|\mu_{\tilde{A}}(x, u) - \mu_{\tilde{A}_T}(x, u)\| < \epsilon, \quad \forall (x, u) \in D. \quad (21)$$

Proof. If \tilde{A} has membership grade consisting a set of patches of continuous linear surfaces (example of its membership grades are made by only using triangular and trapezoid membership grades), then TIN \tilde{A}_T is created as follows:

1. Set V is the set of vertices of membership grades of \tilde{A} .
2. Set E is the set of edges of membership grades of \tilde{A} .
3. Call $X = (V, E)$ is a planar straight line graph (PSLG). Make a TIN A_T is the constrained Delaunay triangulation from X . \tilde{A}_T is a type-2 fuzzy set with membership function A_T .

Observe that A_T represents faithfully the membership grade of \tilde{A} .

If \tilde{A} has membership grade consisting only one continuous non-linear surfaces. Let A^T is a TIN that represents \tilde{A} in D . Suppose that $\exists(x_k, u_k) \in D$ so that $d_k = \|f_A(x_k, u_k) - f_{A_T}(x_k, u_k)\| \geq \epsilon$.

A^T is modified by inserting new vertex (x_k, u_k) as the following steps:

1. Find the triangle T_j of A^T , in which $(x_k, u_k) \in T_j$.
2. Partition the T_j into sub-triangles depending on the position of (x_k, u_k) on T_j .
 - + If (x_k, u_k) lies on edge e_k of T_j , e_k is the adjacent edge of T_j and T_k . Partitioning T_j, T_k into four sub-triangles as Figure 9a.
 - + If (x_k, u_k) is in T_j . Partitioning T_j into three sub-triangles as Figure 9b.
3. Verify new triangles that meet the constrained Delaunay triangulation. This operation may re-arrange triangles by using *flip* operation for two adjacent triangles.

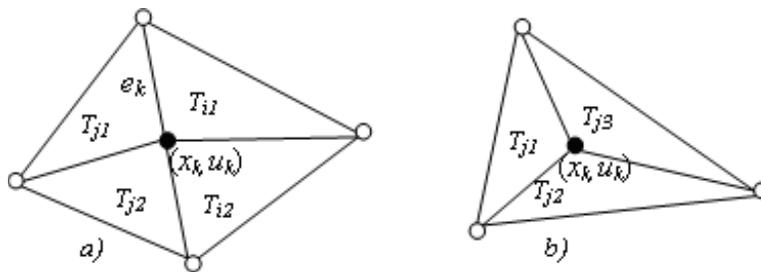


Fig. 9. Partitioning the t_j triangle.

The algorithm results in that T_j is divided into smaller sub-triangles. So we could find triangle T^* of TIN A_T^* , is modified TIN of A_T after N_k steps, so that T^* is small enough and contains (x_k, u_k) . The continuity of the membership grade of \tilde{A} shows that

$$d_k^* = \|f_A(x_k, u_k) - f_{A_T^*}(x_k, u_k)\| < \epsilon \quad (22)$$

We prove the theorem in the case that membership grade of \tilde{A} is set of patches of continuous linear and non-linear surfaces. Suppose that its membership grade involves N patches of discrete continuous linear surfaces and M patches of discrete continuous linear surfaces, S_1, S_2, \dots, S_M . N patches of continuous linear surfaces, that are represented by a TN S_T^* , is proven above section. According to above proof, each continuous non-linear surface A is represented approximately by a TIN A_T . So M continuous non-linear patches, S_1, S_2, \dots, S_M are represented by M TINs $S_{T1}, S_{T2}, \dots, S_{TM}$. Because of the discreteness of M patches, M TINs representing patches are also discrete. For this reason, we could combine M TINs $S_{T1}, S_{T2}, \dots, S_{TM}$ and S_T^* into only one TIN S_T . \square

Definition 4.4. A base-line of a TIN representing a type-2 fuzzy set is a polyline $v_i (i = 1, \dots, N)$ satisfying $v_i \cdot u = 0$ and $v_i v_{i+1}$ is a edge of triangle of TIN.

Figure 10 is the TIN that represent approximately of Gaussian type-2 fuzzy sets with $\epsilon = 0.1$. The primary MF is a Gaussian with fixed deviation and mean $m_k \in [m_1, m_2]$ and the secondary MF is a triangular MF. The dashed-line is a base-line of TIN.

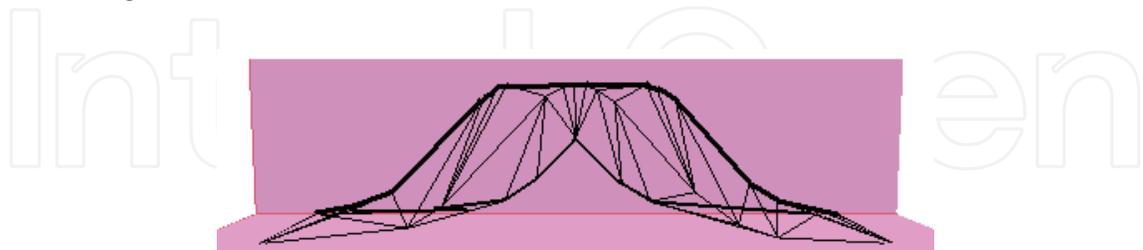


Fig. 10. Example of representation of a type-2 Gaussian fuzzy sets

5. Applications

5.1 Algorithms for operations on TIN

Data of TIN includes vertices, indices of faces and relations of them. Data of vertices is a list of 3D vectors with x , y and z components. Indices of faces are three indices of vertices of triangle. Relations between vertices and faces is used to speed up algorithms on TIN such as searching or computing algorithms.

The section introduces some algorithms operating on TIN such as: intersection of two TINs, minimum or maximum of two TINs. Algorithm on intersection is to create a poly-line that is intersection and break-line of TINs. Algorithm on maximum/minimum is to generate new TIN T_0 from two TINs T_1, T_2 satisfying $\forall(x, u) | \mu^{T_0}(x, u) = \min(\mu^{T_1}(x, u), \mu^{T_2}(x, u))$ or $\mu^{T_0}(x, u) = \max(\mu^{T_1}(x, u), \mu^{T_2}(x, u))$. The following is the detailed descriptions of algorithms.

Algorithm 5.1 (Intersection Algorithm). *Input:* T_1, T_2 are two TINs representing two type-2 fuzzy sets.

Outputs: Modified T_1, T_2 are with some new vertices and edges on intersection poly-lines.

1. Computing L_1, L_2 are base-lines of T_1, T_2 , respectively.
2. Find $v_k^* (k = 1, \dots, M)$ are the intersection points of L_1, L_2 .
3. If $M = 0$ or set of intersection points is empty then return.
4. For each $v_k^* (k = 1, \dots, M)$
 - $v^* \leftarrow v_k^*$. Init queue Q_k .
 - While not find v^*
 - (a) $v \leftarrow v^*$. Insert v into queue Q_k .
 - (b) Insert v into each of T_1, T_2 , become v_{T_1}, v_{T_2} .
 - (c) Find adjacent triangle t_1^*, t_2^* of v_{T_1} and v_{T_2} , respectively, so that t_1^*, t_2^* are intersected by a segment in t_1^* and t_2^* .

- (d) If existing new v^* point so that vv^* is a intersecting segment of t_1^* and t_2^* then
 $v \leftarrow v^*$
 Come back step a).
 Else
 Come back step 2).

Algorithm 5.2 (maximum/minimum Algorithm). Input: T_1, T_2 are two TINs that represent two type-2 fuzzy sets.

Output: T_0 is result TIN of minimum/maximum operation.

1. Computing intersection of T_1, T_2 (using the algorithm of computing intersection).
2. Init queue Q .
3. for each triangle t of T_1 or T_2 .
 - (a) With maximum algorithm:
if t is triangle of $T_1(T_2)$ and be upper than $T_2(T_1)$ then push t into Q .
 - (b) With minimum algorithm:
if t is triangle of $T_1(T_2)$ and be lower than $T_2(T_1)$ then push t into Q .
 - (c) Generating TIN from triangles in Q .

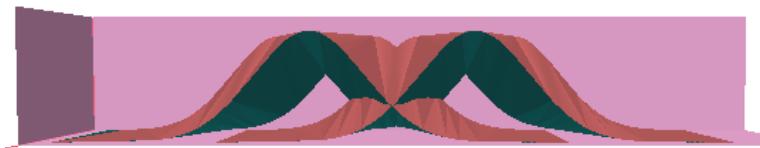


Fig. 11. Example of two fuzzy sets for operations

5.2 Join operation

Theoretic union operation is described the following using Zadeh's Extension Principle.

$$\mu_{\tilde{A} \cup \tilde{B}}(x) = \mu_{\tilde{A}}(x) \sqcup \mu_{\tilde{B}}(x) = \int_u \int_v (f_x(u) \star g_x(w)) / (u \vee w) \quad (23)$$

where \vee represents the max t-conorm and \star represents a t-norm. If $\mu_{\tilde{A}}(x)$ and $\mu_{\tilde{B}}(x)$ have discrete domains, (23) is rewritten as follows:

$$\mu_{\tilde{A} \cup \tilde{B}}(x) = \mu_{\tilde{A}}(x) \sqcup \mu_{\tilde{B}}(x) = \sum_u \sum_v (f_x(u) \star g_x(w)) / (u \vee w) \quad (24)$$

In (23) and (24), if more than one calculation of u and w gives the same point $u \vee w$, then in the union the one with the largest membership grade is kept. Suppose, for example, $u_1 \vee w_1 = \theta^*$ and $u_2 \vee w_2 = \theta^*$. Then within the computation of (23) and (24) we would have

$$f_x(u_1) \star g_x(w_1) / \theta^* + f_x(u_2) \star g_x(w_2) / \theta^* \quad (25)$$

where $+$ denotes union. Combining these two terms for the common θ^* is a type-1 computation in which t-conorm can be used, e.g. the maximum.

Theoretic join operation is described as follows. For every pair of points $\{u, w\}$, such that $u \in F \subseteq [0, 1]$ of \tilde{A} and $w \in G \subseteq [0, 1]$ of \tilde{B} , we find the maximum of v and w and the minimum of their memberships, so that $v \vee w$ is an element of $F \sqcup G$ and $f_x(v) \wedge g_x(w)$ is the corresponding membership grade. If more than one $\{u, w\}$ pair gives the same maximum (i.e., the same element in $F \sqcup G$), maximum of all the corresponding membership grades is used as the membership of this element.

If $\theta \in F \sqcup G$, the possible $\{u, w\}$ pairs that can give θ as the result of the maximum operation are $\{u, \theta\}$ where $u \in (-\infty, \theta]$ and $\{\theta, w\}$ where $w \in (-\infty, \theta]$. The process of finding the membership of θ in $\tilde{A} \sqcup \tilde{B}$ can be divided into three steps: (1) find the minimum between the memberships of all the pairs $\{u, \theta\}$ such that $u \in (-\infty, \theta]$ and then find their supremum; (2) do the same with all the pairs $\{\theta, w\}$ such that $w \in (-\infty, \theta]$; and, (3) find the maximum of the two supremum, i.e.,

$$h_{F \sqcup G}(\theta) = \phi_1(\theta) \vee \phi_2(\theta) \tag{26}$$

where

$$\phi_1(\theta) = \sup_{u \in (-\infty, \theta]} \{f_x(u) \wedge g_x(\theta)\} = g_x(\theta) \wedge \sup_{u \in (-\infty, \theta]} \{f_x(u)\} \tag{27}$$

and

$$\phi_2(\theta) = \sup_{w \in (-\infty, \theta]} \{f_x(\theta) \wedge g_x(w)\} = f_x(\theta) \wedge \sup_{w \in (-\infty, \theta]} \{g_x(w)\} \tag{28}$$

Based-on theoretic join and meet operation, we proposed TIN-based geometric algorithm for join operation. This algorithm uses two above mentioned algorithms involving intersection and min/max.

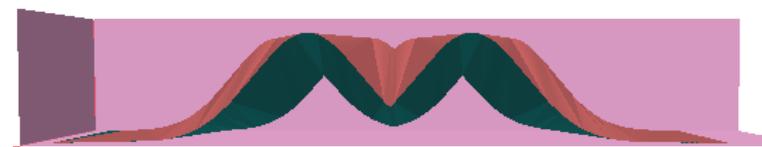


Fig. 12. Join operation

Algorithm 5.3 (Join Operation). Input: \tilde{A}, \tilde{B} are two type-2 fuzzy sets with TINs $T_{\tilde{A}}, T_{\tilde{B}}$.

Output: \tilde{C} is result of join operation.

1. Find the upper surface by using the max-algorithm:

$$T_{\tilde{C}_{US}} = \max(T_{\tilde{A}_{US}}, T_{\tilde{B}_{US}})$$

2. Find the lower surface by using the max-algorithm:

$$T_{\tilde{C}_{LS}} = \max(T_{\tilde{A}_{LS}}, T_{\tilde{B}_{LS}})$$

3. Generate normal surface from $T_{\tilde{C}_{US}}$ and $T_{\tilde{C}_{LS}}$ using Delaunay Triangulation.

Figure 11 is two type-2 fuzzy sets that its primary MF is Gaussian MF and its secondary MF is triangular MF. Figure 12 is the result T2FS that are rendered in 3D environment.

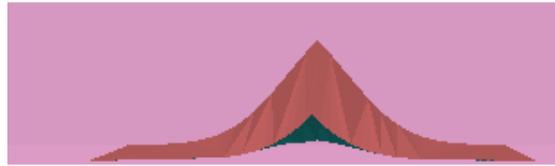


Fig. 13. Meet operation

5.3 Meet operation

Recall theoretic meet operation is described as follows:

$$\mu_{\tilde{A} \cap \tilde{B}}(x) = \mu_{\tilde{A}}(x) \cap \mu_{\tilde{B}}(x) = \int_u \int_v (f_x(u) \star g_x(w)) / (u \star w) \quad (29)$$

where \star represents a t-norm. If $\mu_{\tilde{A}}(x)$ and $\mu_{\tilde{B}}(x)$ have discrete domains, (29) is rewritten as follows:

$$\mu_{\tilde{A} \cap \tilde{B}}(x) = \mu_{\tilde{A}}(x) \cap \mu_{\tilde{B}}(x) = \sum_u \sum_v (f_x(u) \star g_x(w)) / (u \wedge w) \quad (30)$$

In a similar way, the point $u \star w$ with the largest membership grade is kept if more than one calculation of u and w gives the same one.

For every pair of points $\{u, w\}$, such that $u \in F \subseteq [0, 1]$ of \tilde{A} and $w \in G \subseteq [0, 1]$ of \tilde{B} , we find the minimum or product of v and w and the minimum of their memberships, so that $v \star w$ is an element of $F \cap G$ and $f_x(v) \wedge g_x(w)$ is the corresponding membership grade.

If $\theta \in F \cap G$, the possible $\{u, w\}$ pairs that can give θ as the result of the maximum operation are $\{\theta, u\}$ where $u \in [\theta, \infty)$ and $\{w, \theta\}$ where $w \in [\theta, \infty)$. The process of finding the membership of θ in $\tilde{A} \cap \tilde{B}$ can be broken into three steps: (1) find the minimum between the memberships of all the pairs $\{u, \theta\}$ such that $u \in [\theta, \infty)$ and then find their supremum; (2) do the same with all the pairs $\{\theta, w\}$ such that $w \in [\theta, \infty)$; and, (3) find the maximum of the two supremum, i.e.,

$$h_{F_1 \sqcup F_2}(\theta) = \phi_1(\theta) \wedge \phi_2(\theta) \quad (31)$$

where

$$\phi_1(\theta) = \sup_{u \in [\theta, \infty)} \{f_x(u) \wedge g_x(\theta)\} = g_x(\theta) \wedge \sup_{u \in [\theta, \infty)} \{f_x(u)\} \quad (32)$$

and

$$\phi_2(\theta) = \sup_{w \in [\theta, \infty)} \{f_x(\theta) \wedge g_x(w)\} = f_x(\theta) \wedge \sup_{w \in [\theta, \infty)} \{g_x(w)\} \quad (33)$$

Algorithm 5.4 (Meet Operation). Input: \tilde{A}, \tilde{B} are two type-2 fuzzy sets with TINs $T_{\tilde{A}}, T_{\tilde{B}}$.

Output: \tilde{C} is result of meet operation.

1. Find the upper surface by using the min-algorithm:

$$T_{\tilde{C}_{US}} = \max(T_{\tilde{A}_{US}}, T_{\tilde{B}_{US}})$$

2. Find the lower surface by using the min-algorithm:

$$T_{\tilde{C}_{LS}} = \max(T_{\tilde{A}_{LS}}, T_{\tilde{B}_{LS}})$$

3. Generate normal surface from $T_{\tilde{C}_{US}}$ and $T_{\tilde{C}_{LS}}$ using Delaunay Triangulation.

5.4 Negation operation

Algorithm 5.5 (Negation Operation). *Input: \tilde{A} is a type-2 fuzzy set. Output is result of negation operation.*

1. For each vertex v_k of T_{US} or T_{LS} of \tilde{B} .
 $v_k.y = 1.0 - v_k.y$
2. Set $T'_{US} \leftarrow T_{LS}, T'_{LS} \leftarrow T_{US}$.
3. Set $\tilde{B} = \{T'_{US}, T_{NS}, T'_{LS}\}$.



Fig. 14. Negation operation

5.5 Rendering and performance

The OpenSceneGraph (OSG) [<http://www.openscenegraph.org>] is an open source high performance 3D graphics toolkit, used by application developers in fields such as visual simulation, games, virtual reality, scientific visualization and modelling. Written entirely in Standard C++ and OpenGL it runs on all Windows platforms, OSX, GNU/Linux, IRIX, Solaris, HP-Ux, AIX and FreeBSD operating systems. The OpenSceneGraph is now well established as the world leading scene graph technology, used widely in the vis-sim, space, scientific, oil-gas, games and virtual reality industries.

We use the OSG for rendering of type-2 fuzzy sets. The approach is implemented for representation of general T2FS with various ε -approximation. Let \tilde{A} is a general type-2 fuzzy set. The feature membership functions of \tilde{A} are described as follows:

FOU is Gaussian function with upper MF and lower MF as follows:

Upper MF of FOU:

$$f_u(x) = \begin{cases} e^{-\frac{1}{2}\left(\frac{x-m_1}{\sigma}\right)^2} & \text{if } x < m_1 \\ 1 & \text{if } m_1 \leq x \leq m_2 \\ e^{-\frac{1}{2}\left(\frac{x-m_2}{\sigma}\right)^2} & \text{if } x > m_2 \end{cases} \quad (34)$$

Lower MF of FOU:

$$f_l(x) = \begin{cases} e^{-\frac{1}{2}\left(\frac{x-m_2}{\sigma}\right)^2} & \text{if } x < \frac{m_1+m_2}{2} \\ e^{-\frac{1}{2}\left(\frac{x-m_1}{\sigma}\right)^2} & \text{if otherwise} \end{cases} \quad (35)$$

where $m_1 = 3.0$, $m_2 = 4.0$ and $\sigma = 0.5$.

The next feature of \tilde{A} is set of points where $\mu_{\tilde{A}}(x, u) = 1.0$, involves points belong to the MF described as follows:

$$f_m(x) = e^{-\frac{1}{2}\left(\frac{x-(m_1+m_2)/2}{\sigma}\right)^2} \quad (36)$$

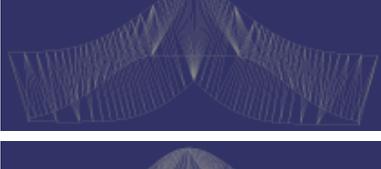
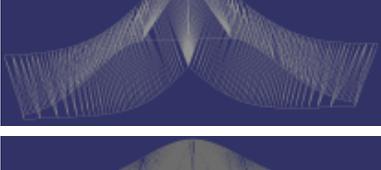
ϵ	Extracted Point Triangles		Time (milliseconds)	TIN
1 E-2	39	49	0.651	
5 E-3	51	65	0.899	
1 E-3	107	145	2.252	
5 E-4	154	211	4.480	
1 E-4	376	549	12.020	
1 E-5	2460	4292	154.96	

Table 1. Results of representation \tilde{A}

The new approach uses memory and computations less than previous approaches. If the TIN has N vertices, M faces then it takes $N \cdot 12$ bytes for vertices, $M \cdot 6$ bytes for faces and $M \cdot 6$ for relations between vertices and faces. For examples, triangular or trapezoid takes about 720 bytes with $N \cong M \cong 30$. Gaussian membership grades take about 200 vertices and 300 faces with accuracy $\epsilon = 0.01$, i.e 6000 bytes. Beside, the memory using for traditional approach takes about 100 000 bytes with step is 0.01 and x takes value in $[0, 10]$.

We also tested the performance of algorithms with different membership grades. We implemented operations in 1000 times for each operation and summarized run-time (in milliseconds) in table 2.

Type-2 MF	Join	Meet	Negation
Triangular-Triangular	1	1	1
Gaussian - Triangular	474	290	1
Interval Gaussian	114	86	1

Table 2. The run-time of operations

6. Conclusion

The chapter introduces the new approach to represent a type-2 fuzzy sets using triangular irregular network. TIN has used to represent 3D surfaces by partitioning domain D into sub-triangle satisfying Delaunay criteria or constrained Delaunay criteria. This representation is used for membership functions of type-2 fuzzy sets that are 3D surfaces. We also proposed approach using half-edge to operate TIN in real-time application. Based-on this result, we have developed new computation to implement operations of type-2 fuzzy sets such as join, meet, negation. These operations is the base to develop computing for type-2 fuzzy logic systems.

The next goals is to continue improving geometry algorithms decrease computational time based on GPU. The second is to apply computing for developing type-2 fuzzy logic system using geometry algorithms.

7. Acknowledgment

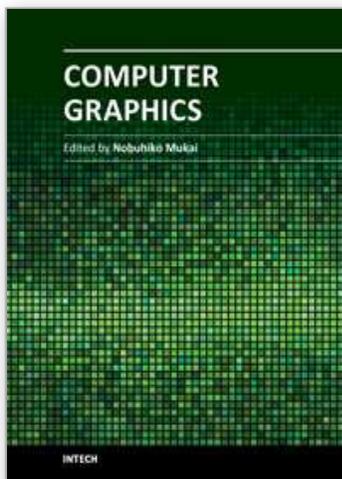
This paper is sponsored by Vietnam National Foundation for Science and Technology Development (NAFOSTED), Grant No 102.012010.12, and RFit@LQDTU, Le Quy Don Technical University, Vietnam.

8. References

- L.A. Zadeh (1975), The concept of linguistic variable and its application to approximate reasoning, *J.Information Science*, Vol. 8, 199 - 249.
- M. Mizumoto, K. Tanaka (1976), Some properties of fuzzy sets of type-2, *J. Information and Control*, Vol. 31, 312 - 340.
- N. Karnik, J.M. Mendel(2001-A), Operations on Type-2 Fuzzy Sets, *Fuzzy Sets and Systems*, Vol. 122, 327-348.
- N. Karnik, J.M. Mendel(2001-B). Centroid of a type-2 fuzzy set, *Information Sciences*, Vol.132, 195-220.
- N.N Karnik, J.M. Mendel, Q. Liang (1999), Type-2 Fuzzy Logic Systems, *IEEE Trans. on Fuzzy Systems*, Vol.7, No.6, 643-658.
- Q. Liang, J.M. Mendel (2000), Interval Type-2 Fuzzy Logic Systems: Theory and Design, *IEEE Trans. on Fuzzy Systems*, Vol.8, No.5, 635-650.
- J.M Mendel, R.B. John (2002), Type-2 Fuzzy Sets Made Simple, *IEEE Trans. Fuzzy Systems*, Vol. 10, no.2, 117-127.
- J. M. Mendel (2004), On Computing the Centroid of a Symmetrical Interval Type-2 Fuzzy Set, *Proc. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*, Perugia, Italy, 4-9.
- S. Coupland, R. John (2007), Geometric Type-1 and Type-2 Fuzzy Logic Systems, *IEEE Trans. on Fuzzy Systems*, Vol. 15, No. 1, 3 - 15.
- S. Coupland, R. John (2008), New geometric inference techniques for type-2 fuzzy sets, *International Journal of Approximate Reasoning*, Vol. 49(1), pp. 198-211.
- S. Coupland, R. John (2008), A Fast Geometric Method for Defuzzification of Type-2 Fuzzy Sets, *IEEE Trans. on Fuzzy Systems*, Vol. 16, No.4, 929-941.
- J. R. Shewchuck (2002), Delaunay Refinement Algorithms for Triangular Mesh Generation, *Computational Geometry: Theory and Applications*, Vol. 22(1-3),21-74.

- J. Ruppert (1995), A Delaunay refinement algorithm for Quality 2-Dimensional Mesh Generation, *Journal of Algorithms*, Vol. 18(3), 548-585.
- L. P. Chew (1989), Constrained Delaunay Triangulations, *Algorithmica* Vol. 4(1), 97-108.
- J. M. Mendel, F. Liu, D. Zhai (2009), α -Plane Representation for Type-2 Fuzzy Sets: Theory and Applications, *IEEE Trans. on Fuzzy Systems*, 17(5), 1189-1207.
- F. Liu (2008), An efficient centroid type-reduction strategy for general type-2 fuzzy logic system, *Information Sciences*, Vol. 178(9), 2224-2236.
- L. T. Ngo, L. T. Pham, P. H. Nguyen, K.Hirota (2007), On approximate representation of type-2 fuzzy sets using triangulated irregular network, *Foundations of Fuzzy Logic and Soft Computing, Lecture Notes in Computer Science*, LNCS 4529, Springer, 584-593.
- L. T. Ngo, L. T. Pham, P. H. Nguyen, K.Hirota (2009), Refinement geometric algorithms for type-2 fuzzy set operations, *Proceedings, IEEE-FUZZ 09*,, 866-871.
- Janusz T. Starczewski (2009), Efficient triangular type-2 fuzzy logic systems, *International Journal of Approximate Reasoning*, Vol. 12, No.5, 799-811.
- H. Hagraš(2004), A Hierarchical Type-2 Fuzzy Logic Control Architecture for Autonomous Mobile Robots, *IEEE Trans. on Fuzzy Systems*, Vol. 12, No.4, 524-539.
- H. Hagraš(2007), Type-2 FLCs: A new Generation of Fuzzy Controllers, *IEEE Computational Intelligence Magazine*, Vol. 2, No. 1, 30-43.
- C. Wagner, H. Hagraš (2010), Toward General Type-2 Fuzzy Logic Systems Based on zSlices, *IEEE Trans. on Fuzzy Systems*, Vol. 18, No.4, pp. 637-660.

IntechOpen



Computer Graphics

Edited by Prof. Nobuhiko Mukai

ISBN 978-953-51-0455-1

Hard cover, 256 pages

Publisher InTech

Published online 30, March, 2012

Published in print edition March, 2012

Computer graphics is now used in various fields; for industrial, educational, medical and entertainment purposes. The aim of computer graphics is to visualize real objects and imaginary or other abstract items. In order to visualize various things, many technologies are necessary and they are mainly divided into two types in computer graphics: modeling and rendering technologies. This book covers the most advanced technologies for both types. It also includes some visualization techniques and applications for motion blur, virtual agents and historical textiles. This book provides useful insights for researchers in computer graphics.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Long Ngo and The Long Pham (2012). An approach to representation of type-2 fuzzy sets using computational methods of computer graphics, *Computer Graphics*, Prof. Nobuhiko Mukai (Ed.), ISBN: 978-953-51-0455-1, InTech, Available from: <http://www.intechopen.com/books/computer-graphics/an-approach-to-representation-of-type-2-fuzzy-sets-using-computational-methods-of-computer-graphics>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen