Refinement Geometric Algorithms for Type-2 Fuzzy Set Operations

Long Thanh Ngo, Long The Pham, Phuong Hoang Nguyen and Kaoru Hirota, *Member, IEEE*

***Abstract*— The paper deals with an approach to applications of ϵ-approximate representation of type-2 fuzzy sets using triangulated irregular network (TIN). Geometric algorithms are designed for operations of type-2 fuzzy sets without using manner of upper or lower surfaces. Operations involving meet under minimum, join under minimum, negation, inference process of type-2 fuzzy sets are presented as applications of geometric representation to operations of type-2 fuzzy sets.**

I. Introduction

Type-2 fuzzy logic is widely applied in real world problems. However, almost deployed applications are interval type-2 fuzzy logic because computational complexity of type-2 fuzzy logic is large. There are many researches on problems arising from reducing the complexity of these systems. Mendel et al [6], [7], [8], [9], [12], [17] have developed theories and computations of type-2 fuzzy sets and systems. Starczewski [17] proposed a method for complexity reduction of operations on triangular type-2 fuzzy sets. For the this purpose, Coupland et al [3], [4] proposed geometric method for representation type-1 and interval type-2 fuzzy sets, new algorithms for various operations on type-1 and type-2 fuzzy sets and for defuzzification. An approach to the representation [13] and geometric operations using upper and lower surfaces of type-2 fuzzy sets are introduced by using triangulated irregular network (TIN). Coupland et al [5] presented new techniques also using upper and lower surfaces for performing logical operations on type-2 fuzzy sets, given a full exposition of the geometric inference operations with considering computational speed and accuracy.

The paper deals with applications of ϵ-approximate representation to operations of type-2 fuzzy sets without using upper or lower surfaces. These approach allows to represent and compute for generalized type-2 fuzzy sets, for example, their vertical slices is non-convex function. Geometric algorithms and theorems on ϵ-approximation of resultant sets of Meet and Join under minimum operations are introduced detail. Computations of inference process of type-2 fuzzy logic systems are proposed as geometric algorithms. The paper also discusses the feasibility of proposed algorithms by implementing on various machines with reported runtime tables.

The paper is organized as follows: II presents type-2 fuzzy sets and inference, ϵ-approximation representation of type-2 fuzzy sets; III introduces applications to operations involving meet and join under minimum, negation and reference process; IV presents implementation and discusses the feasibility of the proposed algorithms with run-times; V is conclusion and future works.

Long Thanh Ngo, Long The Pham, is with the Center of Simulation Technology, Le Quy Don University of Technology, Hanoi, Vietnam (email: ngotlong@gmail.com, thelong@math.ac.vn). Phuong Hoang Nguyen is with the Center of Health Information Technology, Ministry of Health, Hanoi, Vietnam (email: phuongnh@fmail.vnn.vn) Kaoru Hirota is with the Tokyo Institute of Technology, Mail-box: G3-49, 4259 Nagatsuta, Midori-ku, Yokohama 226-8502, Japan (email:hirota@hrt.dis.titech.ac.jp)

II. Type-2 Fuzzy Sets and Their Geometric Representation

A. Type-2 Fuzzy Sets

A type-2 fuzzy set in X is $\tilde{A}$, and the membership grade of $x \in X$ in A 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 the domain of $\mu_{\tilde{A}}(x,u)$ are called primary memberships of x in $\tilde{A}$ and the memberships of the primary memberships in $\mu_{\tilde{A}}(x,u)$ are called secondary memberships of x in $\tilde{A}$.

Definition 2.1: 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.,

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

or

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x,u))/(x,u), J_x \subseteq [0,1] \quad (2)$$

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

At each value of x, say $x = x'$, the 2-D 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 $\forall u \in J_{x'} \subseteq [0,1]$, ie.,

$$\mu_{\tilde{A}}(x = x', u) \equiv \mu_{\tilde{A}}(x') = \int_{u \in J_{x'}} f_{x'}(u)/u, J_{x'} \subseteq [0,1] \quad (3)$$

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

Theoretic operations of type-2 fuzzy sets such as union, intersection and complement are described [7] as follows:

$$\mu_{\tilde{A}\cup\tilde{B}}(x) = \mu_{\tilde{A}}(x) \sqcup \mu_{\tilde{B}}(x) = \int_u \int_v (f_x(u) \star g_x(w))/(u \vee w) \quad (4)$$

$$\mu_{\tilde{A}\cap\tilde{B}}(x) = \mu_{\tilde{A}}(x) \sqcap \mu_{\tilde{B}}(x) = \int_u \int_v (f_x(u) \star g_x(w))/(u \star w) \quad (5)$$

$$\mu_{\tilde{A}}(x) = \mu_{\neg\tilde{A}}(x) = \int_u (f_x(u))/(1-u) \quad (6)$$

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 $f_{x'}(u) = 1\ \forall u \in J_x$.

978-1-4244-3597-5/09/$25.00 ©2009 IEEE

B. Inference of Type-2 Fuzzy Logic System

Consider a type-2 FLS having p inputs, $x_1 \in X_1, x_2 \in X_2, ..., x_p \in X_p$, and one output $y \in Y$. Suppose that it has M rules where the l^{th} rule has the form

$$R^l : \text{IF } x_1 \text{ is } \tilde{F}_1^l \text{ AND ... AND } x_p \text{ is } \tilde{F}_p^l \text{ THEN } y \text{ is } \tilde{G}^l. \tag{7}$$

This rule represents a type-2 fuzzy relation between the input space $X_1 \times X_2 \times ... \times X_p$ and the output space Y of the FLS. The membership function of this type-2 relation is denoted as $\mu_{\tilde{F}_1^l \times ... \times \tilde{F}_p^l \rightarrow \tilde{G}^l}(x, y)$, where $\tilde{F}_1^l \times ... \times \tilde{F}_p^l$ denotes the Cartesian product of $\tilde{F}_1^l$, ..., $\tilde{F}_p^l$, and $x = \{x_1, x_2, ..., x_p\}$.

When an input $\mathbf{x}'$ is applied, the composition of the fuzzy set $\tilde{X}'$ to which $\mathbf{x}''$ belongs and the rule R^l is found by using the extended sup-star composition

$$\mu_{\tilde{X}' \circ \tilde{F}_1^l \times ... \times \tilde{F}_p^l \rightarrow \tilde{G}^l}(y) = \sqcup_{x \in \tilde{X}'} [\mu_{\tilde{X}'}(x) \sqcup \mu_{\tilde{F}_1^l \times ... \times \tilde{F}_p^l \rightarrow \tilde{G}^l}(x, y)] \tag{8}$$

Denote $\tilde{X}' \circ \tilde{F}_1^l \times ... \times \tilde{F}_p^l \rightarrow \tilde{G}^l$ as $\tilde{B}^l$, the output set corresponding to the l^{th} rule. Singleton fuzzification and the product or minimum implication are used as two process of FLS, inference process of rule l^{th} is described (detail in [6]) as follows:

$$\begin{aligned} \mu_{\tilde{B}^l}(y) &= \mu_{\tilde{F}_1^l}(x_1) \sqcap \mu_{\tilde{F}_2^l}(x_2) \sqcap ... \sqcap \mu_{\tilde{F}_p^l}(x_p) \sqcap \mu_{\tilde{G}^l}(y) \\ &= \mu_{\tilde{G}^l}(y) \sqcap [\sqcap_{i=1}^p \mu_{\tilde{F}_i^l}(x_i)] \end{aligned} \tag{9}$$

C. Geometric Representation of Type-2 Fuzzy Sets

In [13], geometric representation of type-2 fuzzy sets is presented as an approach using triangulated irregular network. Concept of ϵ-approximation set is used to approximate membership grades of type-2 fuzzy sets. ϵ-approximation is defined as follows:

Definition 2.2: A type-2 fuzzy set is called ϵ-approximation set, denoted $\tilde{A}^*$, of $\tilde{A}$ in continuous domain D if

$$\|\mu_{\tilde{A}}(x, u) - \mu_{\tilde{A}^*}(x, u)\| \le \epsilon, (x, u) \in D \tag{10}$$

The theorem on representation of a type-2 fuzzy set referenced as *Approximation Theorem* is presented as follows:

Theorem 2.1 (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,

$$\|\mu_{\tilde{A}}(x, u) - \mu_{\tilde{A_T}}(x, u)\| < \epsilon, \ \forall (x, u) \in D. \tag{11}$$

Fig. 1 is the TIN consisting of 36 vertices and 48 faces that represents approximately of Gaussian type-2 fuzzy sets with $\epsilon = 0.1$. The primary membership function is a Gaussian with fixed deviation and mean $m_k \in [m_1, m_2]$ and the secondary membership function is a triangular membership function that its top is on the Gaussian function with mean $(m_1 + m_2)/2$. At the point (x, u) that its color is dark, the value of $\mu_{\tilde{A}}(x, u)$ is close 1.0, otherwise $\mu_{\tilde{A}}(x, u)$ is close 0.0.

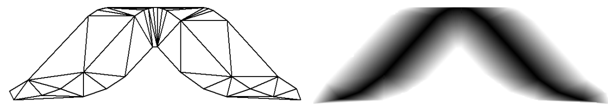

Fig. 1. Example of representation of a type-2 Gaussian fuzzy sets

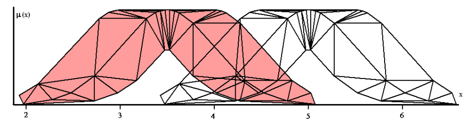

Fig. 2. Example of two approximate Gaussian type-2 fuzzy sets

III. Applications to Operations

This section introduces some applications of the representation to operations of type-2 fuzzy sets. Fig. 2 shows ϵ-approximation representations of two type-2 Gaussian fuzzy sets $\tilde{A}$ and $\tilde{B}$ are depicted with parameters as $m_{\tilde{A}} \in [3, 4], \sigma_{\tilde{A}} = 0.5$ and $m_{\tilde{B}} \in [4.5, 5.5], \sigma_{\tilde{B}} = 0.5$.

A. Geometric Algorithms

Geometric algorithms such as break-line creation or computing the intersection of two TINs are designed for computations of operations of type-2 FS and inference process. Firstly, we define the depth of a TIN in direction α.

Definition 3.1: Depth of a TIN at point P in direction α is number of triangles that are intersected by the ray that is from P in direction α.

The following is the algorithm for finding the triangle that contains a point P. The algorithm is used in some operations on TIN such as break-line creation or polyline intersection. Fig. 3 illustrates an example of finding the triangle containing P_i with starting vertex V_k.

Algorithm 3.1 (Finding triangle containing P): .

Input: TIN T, the point P.

Output: triangle t containing P.

1) Find vertex V that is closest to P.
2) Identify the edge h_e of vertex V that is on the left and closest to P.
   a) Verify the condition: triangle t formed from two edges that are closest to P and contains P.
   b) If t does not contain P: Set V = the end vertex of h_e and go to 1).
   c) Otherwise terminate the algorithm.

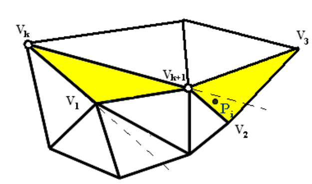

Fig. 3. Finding the triangle containing P

Definition 3.2: Let T be a TIN and $L_g = \{P_1, P_2, \cdots, P_n\}$ be a polyline in which $P_i (i = 1, 2, \ldots, n)$ are vertices of T. L_g is called break-line of T if $n - 1$ line segments $P_1P_2, \ldots, P_{n-1}P_n$ are edges of T.

The following algorithm is to compute the intersection of two TINs from a point P which is an intersection point on a boundary edge. The output of the algorithm is a polyline that is break-line of both TINs. The following is the description of the algorithm:

Algorithm 3.2 (Computing intersection): .

Input: TIN T_1 and T_2.

Output: Intersection polyline and two TINs T'_1, T'_2.

1) Find intersection points $v^*_k (k = 1, .., M)$ of L_1 and L_2 that are at boundary polylines of T_1 and T_2.
2) If $M = 0$ or set of points is empty then return.
3) For each $v^*_k (k = 1, ..., M)$
   $v^* \leftarrow v^*_k$. Initialize queue Q_k.
   While not find v^*
   a) $v \leftarrow v^*$. Insert v into Q_k.
   b) Insert v into each of T_1, T_2, becoming v_{T_1}, v_{T_2}.
   c) Find adjacent triangles 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 there is a new v^* point so that vv^* is a intersecting segment of t^*_1 and t^*_2 then
      $v \leftarrow v^*$
      Go to step a).
      Else
      Go to step 2).

B. Join Operation

Theoretic union operation is described as (4). Suppose more than one calculation of u and w gives the same point $u \vee w$, for example, $u_1 \vee w_1 = \theta^*$ and $u_2 \vee w_2 = \theta^*$. Then within the computation of (4), we would have

$$f_x(u_1) \star g_x(w_1)/\theta^* + f_x(u_2) \star g_x(w_2)/\theta^* \qquad (12)$$

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.

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 computed as follows:

$$f_{F \sqcup G}(\theta) = \phi_1(\theta) \vee \phi_2(\theta) \qquad (13)$$

where

$$\phi_1(\theta) = \sup_{u \in (-\infty, \theta]} \{f_x(u) \wedge g_x(\theta)\} = g_x(\theta) \wedge \sup_{u \in (-\infty, \theta]} \{f_x(u)\} \qquad (14)$$

and

$$\phi_2(\theta) = f_x(\theta) \wedge \sup_{w \in (-\infty, \theta]} \{g_x(w)\} \qquad (15)$$

The following is the theorem on ϵ-approximation of resultant set between two ϵ-approximation sets.

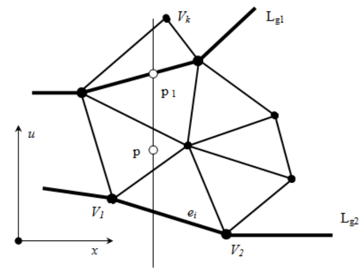

Fig. 4. $\mathbf{p}$ in the overlapped region.

Theorem 3.1: Let $T_{\tilde{A}}$, $T_{\tilde{B}}$ be TINs and ϵ-approximation of $\tilde{A}, \tilde{B}$, respectively, and $L_g s$ are all polylines being break-lines and intersections or boundaries of $T_{\tilde{A}}$ and $T_{\tilde{B}}$ so that there exist no edge of both TINs that is intersected by their intersections in 2-D plane **O**xu. If $T_{\tilde{C}}$ is a TIN formed from set of triangle $t_i (i = 1, \ldots, n)$, in which t_i is a triangle of $T_{\tilde{A}}$ or $T_{\tilde{B}}$ and three vertices of t_i meet equation (4), and type-2 fuzzy set $\tilde{C}$ is the resultant set of the join operation under minimum of $\tilde{A}$ and $\tilde{B}$ then $T_{\tilde{C}}$ is ϵ-approximation of $\tilde{C}$.

Proof: Let $v_i (i = 1, \ldots, n)$ are vertices of $T_{\tilde{C}}$. Because of the formation of $T_{\tilde{C}}$ using join operation under minimum, $T_{\tilde{C}}$ is an ϵ-approximation of $\tilde{C}$ at their all vertices and points on the intersection polylines or boundary polylines. We prove the theorem at points in the domain that not be vertices. If $p(x, u)$ is a point in the domain of $\tilde{C}$ that $\mu_{\tilde{A}}(x, u) = 0$ or $\mu_{\tilde{B}}(x, u) = 0$ then $T_{\tilde{C}}$ is ϵ-approximation of $\tilde{C}$. Now we prove the theorem at points that $\mu_{\tilde{A}}(x, u) > 0$ and $\mu_{\tilde{B}}(x, u) > 0$.

Suppose that these exist $\mathbf{p}(x_p, u_p)$ is a point in triangle t in the region depicted in Fig. (4) that $T_{\tilde{C}}$ is non-ϵ-approximation of $\tilde{C}$. $\mu_{\tilde{C}}(\mathbf{p})$ is computed as follows:

$$\begin{aligned} \mu_{\tilde{C}}(x_p, u_p) = \quad & (\mu_{\tilde{A}}(x_p, u_p) \wedge \sup_k(\mu_{\tilde{B}}(x_p, u_k))) \\ & \vee(\mu_{\tilde{B}}(x_p, u_p) \wedge \sup_k(\mu_{\tilde{A}}(x_p, u_k))) \end{aligned} \qquad (16)$$

In the figure, L_{g1} may be a intersection break-line or boundary break-line. So we have

$$\begin{aligned} \sup_k(\mu_{\tilde{A}}(x_p, u_k)) = \sup_k(\mu_{\tilde{B}}(x_{p_1}, u_k)) \\ \sup_k(\mu_{\tilde{B}}(x_p, u_k)) = \sup_k(\mu_{\tilde{B}}(x_{p_1}, u_k)) \end{aligned} \qquad (17)$$

The cause of non-ϵ-approximation at $\mathbf{p}$ may be $\mu_{\tilde{A}}(x_p, u_p)$ or $\mu_{\tilde{B}}(x_p, u_p)$ because $T_{\tilde{C}}$ meets ϵ-approximation criterion at $\mathbf{p}_1$ and equation (17). These is reasonless because of the supposition of ϵ-approximation of $T_{\tilde{A}}$ and $T_{\tilde{C}}$. Hence, $T_{\tilde{C}}$ is ϵ-approximation of $\tilde{C}$ at $\mathbf{p}$. ∎

On the basis of the theoretic operation between two type-2 fuzzy sets, we proposed an algorithm for join operation between $\tilde{A}$ and $\tilde{B}$ using geometric algorithms. Let $\tilde{C}$ that its ϵ-approximation TIN $T_{\tilde{C}}$ be the resultant type-2 fuzzy set.

To describe computation of geometric operation, the first is the algorithm used to compute the supremum of $f_x(u)$, where $u \in [0, \theta]$ or $u \in [\theta, 1]$, of a TIN. Let Pt is a ray

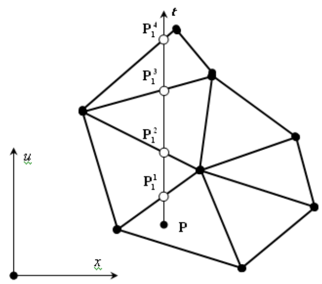

Fig. 5. Compute the supremum of a TIN at P.

from the point (x_0, θ) to the end point ($(x_0, 0)$ or $(x_0, 1)$). The algorithm is described as follows (Fig. 5).

Algorithm 3.3: getSupremum(P, y_{sup}, opt)

Input: TIN T, point P, operation opt (JOIN or MEET).

Output: y_{sup} is the supremum value.

1) Find the triangle t_k containing $\mathbf{p}$. This operation uses the algorithm described in the above section.
   If no existing t_k then return 0;
2) If opt is join operation then ray $\mathbf{p_t}$ is segment from $\mathbf{p}$ to $(x_p, 0)$. Otherwise, if opt is meet operation then ray $\mathbf{p_t}$ is segment from $\mathbf{p}$ to $(x_p, 1)$.
3) Find $\mathbf{p_1}$ is the intersection point of ray $\mathbf{p_t}$ and edges of t_k. Note that $\mathbf{p_1}$ is different to $\mathbf{p}$, so existing only one point $\mathbf{p_1}$.
   If no existing the point $\mathbf{p_1}$ then return 0;
4) If y_{sup} is not higher the elevation y^* of T at P_1 then set $y_{sup} = y^*$.
5) Compute recursion for the point $\mathbf{p_1}$.
   return getSupremum($\mathbf{p_1}$, y_{sup}, opt);

Note that the ray $\mathbf{p_t}$ in the above algorithm for join operation is the segment from $\mathbf{p}$ to the point ($x_{\mathbf{p}}$, 0) in 2-D plane. The complexity of this algorithm depends on the depth d of the TIN at P in direction of u axis, i.e. $O(d)$.

The following part is the sequence of computations needed to obtain for join operation under minimum and product.

Algorithm 3.4: Join operation under minimum.

Input: Two TINs $T_{\tilde{A}}$ and $T_{\tilde{B}}$ of two type-2 fuzzy sets $\tilde{A}$ and $\tilde{B}$.

Output: $T_{\tilde{C}}$ is ϵ-approximation of the resultant type-2 fuzzy set.

1) Initialize a TIN $T_{\tilde{C}}$.
2) Find polylines that are intersection of $T_{\tilde{A}}$ and $T_{\tilde{B}}$. Create break-lines for $T_{\tilde{A}}$ and $T_{\tilde{B}}$ from these polylines.
3) For each triangle t_j of $T_{\tilde{A}}$ and $T_{\tilde{B}}$, suppose that (v_{1j}, v_{2j}, v_{3j}) are three vertices of t_j and v_{4j} is center of gravity of t_j.
   Set $d_k = 0$
   For each vertex $v_{kj} = (x_{kj}, u_{kj}, y_{kj})(k = 1, 2, 3, 4.)$ of t_j do.
   a) Compute
      $y_{sup} = getSupremum(v_{kj}, 0, JOIN)$ for $T_{\tilde{B}}$
      $y_1 = min\{\mu_{\tilde{A}}(x_{kj}, u_{kj}), y_{sup}\}$.
   b) Compute
      $y_{sup} = getSupremum(v_{kj}, 0, JOIN)$ for $T_{\tilde{A}}$
      $y_2 = min\{\mu_{\tilde{B}}(x_{kj}, u_{kj}), y_{sup}\}$.
   c) Set $d_k += fabs(y_{kj} - max\{y_1, y_2\})$
   If $d_k < \epsilon$ then
   Insert t_j into $T_{\tilde{C}}$.
4) Reject triangles t of $T_{\tilde{C}}$ that abscissa y of its three vertices equals 0.

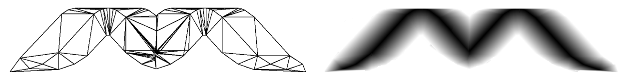

Fig. 6. The resultant set of join operation under minimum

Computational complexity. The algorithm involves two steps. The first step is to find the intersection polylines of two TINs and the complexity is discussed in above algorithm. The second step computes the join operation at vertices and gravity center of triangles, so the complexity of this step is $O(N)$, where N are sum of number of vertices of TINs.

Fig. 6 shows the resultant set of join operation under minimum between two type-2 Gaussian fuzzy sets in Fig. 2.

C. Meet Operation

Theoretic meet operation is described as (5). If $\theta \in F \sqcap G$, the possible $\{u, w\}$ pairs that can give θ as the result of 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 $\tilde{A} \sqcap \tilde{B}$ can be computed as follows:

$$f_{F_1 \sqcup F_2}(\theta) = \phi_1(\theta) \wedge \phi_2(\theta) \tag{18}$$

where

$$\phi_1(\theta) = g_x(\theta) \wedge \sup_{u \in [\theta, \infty)} \{f_x(u)\} \tag{19}$$

and

$$\phi_2(\theta) = f_x(\theta) \wedge \sup_{w \in [\theta, \infty)} \{g_x(w)\} \tag{20}$$

Theorem 3.2: Let $T_{\tilde{A}}, T_{\tilde{B}}$ be TINs and ϵ-approximation of $\tilde{A}$, $\tilde{B}$, respectively, and $L_g s$ are all polylines that are break-lines and intersections or boundaries of $T_{\tilde{A}}$ and $T_{\tilde{B}}$ so that there exist no edge of both TINs that is intersected by their intersections in 2-D plane **O**xu. If $T_{\tilde{C}}$ is a TIN formed from set of triangle $t_i (i = 1, \ldots, n)$, in which t_i is a triangle of $T_{\tilde{A}}$ or $T_{\tilde{B}}$ and three vertices of t_i meet equation (5), and type-2 fuzzy set $\tilde{C}$ is the resultant set of the meet operation under minimum of $\tilde{A}$ and $\tilde{B}$ then $T_{\tilde{C}}$ is ϵ-approximation of $\tilde{C}$.

Proof: In the same way as the proof of theorem (3.1), we prove the theorem at points that $\mu_{\tilde{A}}(x, u) > 0$ and $\mu_{\tilde{B}}(x, u) > 0$.

Suppose that these exist $\mathbf{p}(x_p, u_p)$ is a point in triangle t in the region depicted in Fig. (4) that $T_{\tilde{C}}$ is non-ϵ-approximation of $\tilde{C}$. $\mu_{\tilde{C}}(\mathbf{p})$ is computed as follows:

$$\begin{aligned} \mu_{\tilde{C}}(x_p, u_p) = \quad & (\mu_{\tilde{A}}(x_p, u_p) \wedge \sup_k(\mu_{\tilde{B}}(x_p, u_k))) \\ & \wedge(\mu_{\tilde{B}}(x_p, u_p) \wedge \sup_k(\mu_{\tilde{A}}(x_p, u_k))) \end{aligned} \tag{21}$$

In the figure, L_{g1} may be a intersection break-line or boundary break-line. So we have

$$\sup_k(\mu_{\tilde{A}}(x_p, u_k)) = \sup_k(\mu_{\tilde{B}}(x_{p_1}, u_k))$$
$$\sup_k(\mu_{\tilde{B}}(x_p, u_k)) = \sup_k(\mu_{\tilde{B}}(x_{p_1}, u_k)) \quad (22)$$

The cause of non-ϵ-approximation at $\mathbf{p}$ may be $\mu_{\tilde{A}}(x_p, u_p)$ or $\mu_{\tilde{B}}(x_p, u_p)$ because $T_{\tilde{C}}$ meets ϵ-approximation criterion at $\mathbf{p}_1$ and equation (22). These is reasonless because of the supposition of ϵ-approximation of $T_{\tilde{A}}$ and $T_{\tilde{C}}$. Hence, $T_{\tilde{C}}$ is ϵ-approximation of $\tilde{C}$ at $\mathbf{p}$. ∎

We also proposed the algorithm for the computation of the meet operation between two T2FSs $\tilde{A}$ and $\tilde{B}$ using geometric algorithms on TIN. Call $\tilde{C}$ with the approximate set $\tilde{C}_T$ (its TIN is $T_{\tilde{C}}$) is the resultant type-2 fuzzy set of meet operation.

Algorithm 3.5: Meet operation
Input: Two TINs $T_{\tilde{A}}$, $T_{\tilde{B}}$ represent $\tilde{A}$, $\tilde{B}$.
Output: $T_{\tilde{C}}$ is the resultant T2FS.

1) Initialize a TIN $T_{\tilde{C}}$.
2) Find polylines that are intersection of $T_{\tilde{A}}$ and $T_{\tilde{B}}$. Create break-lines from these polylines.
3) For each triangle t_j of TIN $T_{\tilde{C}}$, suppose that (v_{1j}, v_{2j}, v_{3j}) is three vertices of t_j and v_{4j} is center of gravity of t_j.
   Set $d_k = 0$
   For each vertex $v_{kj} = (x_{kj}, u_{kj}, y_{kj})(k = 1, 2, 3, 4.)$ of t_j do.
   a) Compute
      $y_{sup} = getSupremum(v_{kj}, 0, MEET)$ for $T_{\tilde{B}}$
      $y_1 = min\{\mu_{\tilde{A}}(x_{kj}, u_{kj}),\ y_{sup}\}$.
   b) Compute
      $y_{sup} = getSupremum(v_{kj}, 0, MEET)$ for $T_{\tilde{A}}$
      $y_2 = min\{\mu_{\tilde{B}}(x_{kj}, u_{kj}),\ y_{sup}\}$.
   c) Set $d_k += fabs(y_{kj} - max\{y_1, y_2\})$
   If $d_k < \epsilon$ then
   Insert t_j into $T_{\tilde{C}}$.
4) Reject triangles t of $T_{\tilde{C}}$ that y-dimension of its three vertices equals 0.

Computational complexity. As the same way of the join operation under minimum, the algorithm involves two steps. The first step is to find intersection polylines of two TINs. The second step computes meet operation at vertices and gravity center of triangles, so the complexity of this step is $O(N)$, where N is sum of number of vertices of TINs T_1, T_2.

Note that the above algorithm contains a procedure of $getSupremum$ using algorithm 3.3 in which the ray $\mathbf{P_t}$ is the segment connecting from $\mathbf{P_t}$ to the point ($\mathbf{P}.x$, 1) in 2-D plane.

Fig. 7 shows the resultant set of meet operation under minimum between two type-2 Gaussian fuzzy sets depicted in Fig. 2.

D. Negation operation

Theoretic complement operation is described as (6). Let $\tilde{A}$ be a type-2 fuzzy set and $\tilde{C}$ be the resultant one from the negation operation. The algorithm is described as follows:

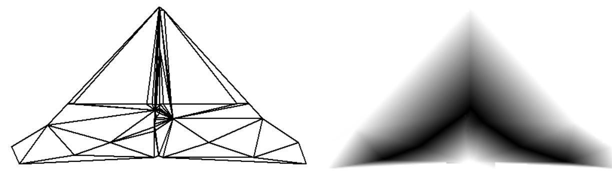

Fig. 7. Meet operation under minimum

Algorithm 3.6: Negation operation
Input: TIN $T_{\tilde{A}}$ represents T2FS $\tilde{A}$.
Output: $T_{\tilde{C}}$ is the resulting T2FS.

1) Call $T_{\tilde{C}}$ is a clone of $T_{\tilde{A}}$.
2) For each vertex $v_k = (x_k, u_k, y_k)$ of $T_{\tilde{C}}$
   Set $u_k = 1.0 - u_k$.

E. Inference

Consider a type-2 FLS having n inputs, $x_1 \in X_1, x_2 \in X_2, ..., x_n \in X_n$ and one output $y \in Y$. Suppose that it has M rules as follows

R^l: IF x_1 is $\tilde{A}_1^l$ AND ... AND x_p is $\tilde{A}_p^l$ THEN y is $\tilde{B}^l$

The used fuzzification is singleton. Inference process is described as equation (9). For each point (y, u), (9) is rewritten as follows:

$$\mu_{\tilde{B}^l}(y, u) = [\ldots[\mu_{\tilde{G}^l}(y, u) \sqcap \mu_{\tilde{F}_1^l}(x_1, u)] \ldots \sqcap \mu_{\tilde{F}_p^l}(x_p, u)] \quad (23)$$

Observe that $\mu_{\tilde{B}^l}(y, u)$ is computed by the meet operation for $\mu_{\tilde{G}^l}(y, u)$ and $\mu_{\tilde{F}_1^l}(x_1, u)$ at x_1. Then resultant type-2 fuzzy set is applied by the meet operation with the next fuzzified set $\mu_{\tilde{F}_2^l}(x_2, u)$ at x_2. The procedure is repeated p times until the last fuzzified set $\mu_{\tilde{F}_p^l}(x_p, u)$ is applied at x_p. The following is the modified algorithm for inference process based on TIN of type-2 fuzzy sets $\tilde{G}^l, \tilde{F}_1^l, \ldots, \tilde{F}_p^l$.

Algorithm 3.7: Inference process for the l^{th} rule.
Input: $T_{\tilde{F}_1}, \ldots, T_{\tilde{F}_p}, T_{\tilde{G}}$ are linguistic T2FSs of the l^{th} rule and $x_1, \ldots, x_p$ are crisp inputs of the system.
Output: $T_{\tilde{C}}$ is the resultant T2FS.

1) Initialize array of 3-D vectors aIP.
2) For each pair $(x_k, T_{\tilde{F}_k})$ do
   Make a segment L_k from $(x_k, 0)$ to $(x_k, 1)$
   Find intersection points $P_j = (x_{P_j}, u_{P_j}, y_{P_j})$ between L_k and edges of the TIN $T_{\tilde{F}_k}$ and insert P_j into aIP.
3) For each point P_j in aIP do
   Make a break-line of $T_{\tilde{G}^l}$ from the segment connected from (x_{min}, u_{P_j}) to (x_{max}, u_{P_j}).
4) For each pair $(x_k, T_{\tilde{F}_k})$ do
   For each vertex $v_j = (x_{v_j}, u_{v_j}, y_{v_j})$ of TIN $T_{\tilde{G}^l}$
   a) Set v = v_j
   b) Compute $y_{sup} = getSupremum(v, 0, MEET)$ for $T_{\tilde{F}_k}$.
      Compute $y_1 = min\{\mu_{\tilde{G}}(x_v, u_v),\ y_{sup}\}$.
   c) Set $x_{\mathbf{v}}$ = x_k.
   d) Compute $y_{sup} = getSupremum(v, 0, MEET)$ for $T_{\tilde{G}}$.

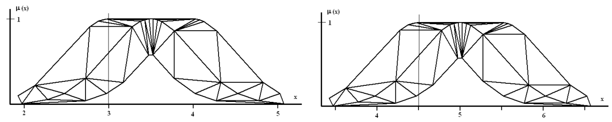

Fig. 8. Example of singleton fuzzification of rule

Fig. 9. Example of inference process

Compute $y_2 = min\{\mu_{\tilde{F}_k}(x_v, u_v),\, y_{sup}\}$.

e) Set $y_{v_j} = max\{y_1, y_2\}$

Computational complexity. The algorithm can be divided into three steps. The first step is to find all of the intersection points between L_k and edges of antecedent TINs. The complexity of this step depends on the depth of the TIN at x_k in direction L_k, i.e. $O(d)$ where $d \ll n_k$ is the depth at x_k in direction L_k and n_k is number of triangles of $T_{\tilde{F}_k}$. The second step is to create break-lines in x axis for the resultant TIN of the rule from set of intersection points. The complexity is $O(n_k)$. The last step is to compute fuzzy value of inference process at the vertices of TIN, so the complexity of this step is linear, i.e. $O(n)$, where n is number of vertices of consequent TIN.

Example: Suppose that rules of type-2 fuzzy logic system has the form as follows:

IF X_1 is $\tilde{F}_1$ AND X_2 is $\tilde{F}_2$ THEN Y is $\tilde{G}$

The type fuzzy sub-sets $\tilde{F}_1$, $\tilde{F}_2$ are described in Fig. 8(a) and Fig. 8(b), respectively, $\tilde{G}$ is described in Fig. 8(b). Suppose that $X_1 = 3.0$ and $X_2 = 4.5$ are the inputs of the inference process. The result of this process is depicted in Fig. 9; the original elevation image of $\tilde{G}$ is in Fig. 9(a) and the output type-2 fuzzy set is in Fig. 9(b).

IV. Experimental Results

We tested the feasibility of algorithms with various membership grades such as triangle-triangle, Gaussian-Triangle(primary and secondary membership functions are Gaussian and triangular functions, respectively), Gaussian-Interval. We implemented operations consisting of join, meet, negation, inference process. Each operation was implemented over static inputs with series of 100 operating cycles. This was repeated 30 times to reduce experimental error. Testing type-2 fuzzy sets are represented by TIN with vertices and faces also shown in the table. Some resultant type-2 fussy sets are described by figures in III. The results were summarized run-times (in milliseconds) in table I.

V. Conclusion

The paper has presented some applications of geometric representation of generalized type-2 fuzzy sets using triangulated irregular network to operations. Geometric algorithms for finding a triangle containing P, intersection of two TINs are designed as basic algorithms for computing operations.

TABLE I

Run-time of computations of type-2 operations

MF	TIN vert., faces	Join	Meet	Neg.	Inf.
Triangle-Triangle	9 v, 8 f	148.0	151.4	< 1	308.6
Gaussian-Triangle	36 v, 48 f	447.9	449.0	< 1	364.3
Gaussian-Interval	17 v, 15 f	114	91	< 1	237

Geometric computations of the operations such as join, meet and negation operations have proposed on the basis of the representation. Computations for inference process of Type-2 FLS have also presented according to the idea of sup-star composition. Algorithms for geometric operations, inference process have implemented for summarizing run-times that speed of algorithms may be fast enough for applications.

The approach based on computational geometry using TIN is one of methods for reduction of computational complexity of type-2 fuzzy logic and could be applied to computations of hybrid systems of type-2 fuzzy sets.

References

[1] M. Berg, M. Kreveld, M. Overmars, and O. Schwarzkopf (1997). Computational Geometry: Algorithms and Applications. Springer-Verlag, Berlin.

[2] L. P. Chew (1989), "Constrained Delaunay Triangulations", *Algorithmica* 4(1),pp. 97-108.

[3] S. Coupland, R. John (2004), "Fuzzy Logic and Computational Geometry", *Proc. RASC* , pp. 3 - 8, Nottingham, UK.

[4] S. Coupland, R. John (2007), "Geometric Type-1 and Type-2 Fuzzy Logic System", *IEEE Trans. on Fuzzy Systems*, 15(1), pp. 3 - 15.

[5] S. Coupland, R. John (2008), "New geometric inference techniques for type-2 fuzzy sets", *International Journal of Approximate Reasoning*, 49(1), pp. 198-211.

[6] N. Karnik, J.M. Mendel and Liang Q. (1999), "Type-2 Fuzzy Logic Systems", *IEEE Trans. on Fuzzy Systems*, 7(6), pp. 643-658.

[7] N. Karnik and J.M. Mendel (2001), "Operations on Type-2 Fuzzy Sets", *Fuzzy Sets and Systems*, 122, pp.327-348.

[8] N. Karnik and J.M. Mendel (2001), "Centroid of a Type-2 Fuzzy Set", *Information Sciences*, 132, pp. 195-220.

[9] Liang Q. and J.M. Mendel (2000), "Interval Type-2 Fuzzy Logic Systems: Theory and Design", *IEEE Trans. on Fuzzy Systems*, 8(5), pp. 535-550.

[10] J.M. Mendel and R. I. John (2002), "Type-2 Fuzzy Sets Made Simple", *IEEE Trans. on Fuzzy Systems*, 10(2), pp. 117-127.

[11] J.M. Mendel , John R. I., Feilong Liu (2006), "Interval Type-2 Fuzzy Logic Systems Made Simple", *IEEE Trans. on Fuzzy Systems*, 14(6), pp. 808-821.

[12] J.M. Mendel (2007), "On a 50% savings in the computation of the centroid of a symmetrical interval type-2 fuzzy set". *Information Sciences*, 172, pp. 417-430.

[13] Long Thanh Ngo, Long The Pham, Phuong Hoang 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, pp. 584-593.

[14] J.R. Shewchuk (1996), *Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator, Proc. First Workshop on Applied Computational Geometry* pp. 124-133, Pennsylvania, USA.

[15] J.R. Shewchuk (1997), *Delaunay refinement mesh generation*, PhD Thesis, Carnegie Mellon University, Pittsburgh, USA.

[16] J.R. Shewchuk (2002), *Delaunay Refinement Algorithms for Triangular Mesh Generation, Computational Geometry: Theory and Applications* 22(1-3):21-74.

[17] Janusz T. Starczewski (2006), "A Triangular Type-2 Fuzzy Logic System", *Proc. FUZZ-IEEE 2006*, pp. 7231-7239, Vancouver, Canada.