

# Type Reduction Techniques for Two-dimensional Interval Type-2 Fuzzy Sets

Vaibhav Saxena\*, Nikhil Yadala<sup>†</sup>, Rishav Chourasia<sup>†</sup>, and Frank Chung-Hoon Rhee<sup>‡</sup>, *Member, IEEE*

\*Department of Mathematics

Indian Institute of Technology Guwahati, Guwahati, India 781039

Email: s.vaibhav@iitg.ac.in

<sup>†</sup>Department of Computer Science and Engineering

Indian Institute of Technology Guwahati, Guwahati, India 781039

Email: {yadala, r.chourasia}@iitg.ac.in

<sup>‡</sup>Department of Electronics and Communication Engineering

Hanyang University, Ansan-Si, Korea 15588

Email: frhee@fuzzy.hanyang.ac.kr, Telephone: +82.31.400.5296, Fax: +82.31.436.8152

**Abstract**—In this paper, we address the issue of type reduction of multi-dimensional interval type-2 (IT2) fuzzy sets (FSs). We utilize the Karnik-Mendel (KM) algorithm to estimate the centroid boundary of a multi-dimensional footprint of uncertainty (FOU). We deal with two-dimensional (2-D) fuzzy sets as we can visualize the FOU using 3-D plots, thus making the illustration of the methods simple. However, the basic idea can be extended to multiple dimensions. We give a formal definition of the centroid boundary of a 2-D IT2 fuzzy membership function (FMF) and propose two methods for its estimation. The first method computes embedded type-1 (T1) FSs whose centroids constitute the centroid boundary. We obtain the embedded sets by producing slices of the domain using different sets of parallel planes and then apply the KM algorithm over each slice, to obtain “embedded-curves.” For the second method, we approximate our first method by restricting embedded-curves to be “embedded-lines” thus enhancing computational speed. These type reduction techniques can be applied to applications involving multi-dimensional centroid estimation such as, clustering, support vector estimation for dimensionality reduction, fuzzy logic controllers, to mention a few.

## I. INTRODUCTION

Type reduction refers to the process of mapping a type-2 (T2) fuzzy set (FS) to a type-1 (T1) fuzzy set. [1]. The Karnik-Mendel (KM) algorithm is an iterative algorithm for computing the centroid bound of an interval type-2 (IT2) fuzzy membership function (FMF), which is used for type reduction. The KM algorithm is a superexponentially fast [2] method and is applied to one dimensional (1-D) fuzzy sets, i.e. where the feature domain is a subset of  $\mathbb{R}$ . Recent use of T2 FSs in various applications involve fuzzy logic controllers in aerospace [3] [4], information granule formation [5], and clustering [6], to name a few.

There is an increasing need for multi-dimensional fuzzy sets in several applications that involve modelling uncertainty of datasets [7], dimensionality reduction [8], clustering [9] [10] [11] [12], which require type reduction and defuzzification as a part of output processing. Most of the current methods accomplish this by calculating the centroid for each dimension separately, ignoring the correlation among multiple dimen-

sions. In this paper, we propose two methods for computing the centroid boundary of a two-dimensional (2-D) IT2 FMF, which is essentially a type reduced map from the IT2 FMF to a T1 FMF.

The remainder of the paper is organized as follows. In Section II, we introduce multi-dimensional fuzzy sets and give a formal definition of the centroid boundary of an IT2 FMF. In Section III, we propose a method that extrapolates the KM algorithm for computing the centroid boundary of a 2-D IT2 FMF. We find points that constitute the boundary by computing “embedded-curves” (and corresponding embedded sets) which are obtained through multiple iterations of the KM algorithm on 1-D slices of the domain. This method possesses a significant computation overload due to the requirement of the KM algorithm being performed a significant number of times on multiple 1-D slices. To overcome this burden, we introduce the concept of “embedded-lines” in Section IV which replaces the embedded-curves in our previous method. Approximating embedded-curves as straight lines approximates the previous method and enhances computational speed. Instead of running the KM algorithm independently on each of the 1-D slices considered, we illustrate a method which combines multiple slices into a single entity on which the KM algorithm shall be executed only once. Since both methods use the KM algorithm internally, faster implementations such as enhanced KM (EKM) [13], iterative algorithm with stopping condition (IASC) [14], and enhanced IASC (EIASC) [15] can also be considered.

## II. MULTI-DIMENSIONAL FUZZY SETS AND KARNIK-MENDEL ALGORITHM

A type-2 fuzzy set (T2 FS) with an  $n$  dimensional *primary variable*, denoted as  $\tilde{\mathbf{A}}$ , is a bivariate function on the Cartesian product space represented by  $\mu_{\tilde{\mathbf{A}}} : \mathbf{X} \times [0, 1] \rightarrow [0, 1]$ , where  $\mathbf{X} \subset \mathbb{R}^n$ . The membership function is denoted by  $\mu_{\tilde{\mathbf{A}}}(\mathbf{x}, u)$ , where  $\mathbf{x} \in \mathbf{X}$  and  $u \in U \subset [0, 1]$  [16]. In set builder notation,

a T2 FS is expressed as

$$\tilde{\mathbf{A}} = \{((\mathbf{x}, u), \mu_{\tilde{\mathbf{A}}}(\mathbf{x}, u)) | \forall \mathbf{x} \in \mathbf{X}, \forall u \in U\}, \quad (1)$$

where  $0 \leq \mu_{\tilde{\mathbf{A}}}(\mathbf{x}, u) \leq 1$ . For the case when  $\mu_{\tilde{\mathbf{A}}}(\mathbf{x}, u) = 1 \quad \forall (\mathbf{x}, u) \in \mathbf{X} \times U$ , the fuzzy set is called an interval type-2 fuzzy set (IT2 FS), and is commonly represented as

$$\tilde{\mathbf{A}} = \int_{\mathbf{x} \in \mathbf{X}} \int_{u \in U} 1 / (\mathbf{x}, u). \quad (2)$$

For an IT2 FS,  $\mu_{\tilde{\mathbf{A}}}(\mathbf{x})$  is called the secondary membership at  $\mathbf{x}$ . An embedded set of an IT2 FS with  $n$  dimensional *primary variable* is a type-1 fuzzy set (T1 FS) expressed as

$$\mathbf{A}_e = \int_{\mathbf{x} \in \mathbf{X}} \mu_{\mathbf{A}_e}(\mathbf{x}) / \mathbf{x}, \quad \text{and} \quad \mu_{\mathbf{A}_e}(\mathbf{x}) \in \mu_{\tilde{\mathbf{A}}}(\mathbf{x}), \quad (3)$$

where  $\mu_{\mathbf{A}_e}$  is the membership function of the embedded set  $\mathbf{A}_e$ . In particular, the membership function of the uppermost and lowermost embedded sets (i.e., the upper membership function (UMF) and lower membership function (LMF)) of the IT2 FS  $\tilde{\mathbf{A}}$  are expressed as  $\bar{\mu}_{\tilde{\mathbf{A}}}$  and  $\underline{\mu}_{\tilde{\mathbf{A}}}$ , respectively. The centroid of an  $n$  dimensional T1 FS is an  $n$  dimensional vector given by

$$c(\mathbf{A}_e) = \frac{\int_{\mathbf{x} \in \mathbf{X}} \mu_{\mathbf{A}_e}(\mathbf{x}) \cdot \mathbf{x} \, d\mathbf{x}}{\int_{\mathbf{x} \in \mathbf{X}} \mu_{\mathbf{A}_e}(\mathbf{x}) \, d\mathbf{x}} \quad (4)$$

and the centroid of an IT2 FS is a subset of  $\mathbb{R}^n$  comprising of the centroids of constituting embedded fuzzy sets and can be expressed in terms of centroids of its embedded sets as

$$C(\tilde{\mathbf{A}}) = 1 / \bigcup_{\forall \mathbf{A}_e \in \tilde{\mathbf{A}}} c(\mathbf{A}_e). \quad (5)$$

The Karnik-Mendel (KM) algorithm is an iterative algorithm that is used to find the centroid of a 1-D IT2 FS. The algorithm converges monotonically and superexponentially fast [2] to the centroid bounds of the IT2 FS. Since the KM algorithm operates on only 1-D IT2 FSs, we obtain two centroids namely left and right centroids as the bound of the centroids of the embedded sets. These centroids are computed by the fast KM algorithm [17], using the converging switch points  $L$  and  $R$  respectively, given by

$$c_l(L) = \frac{\int_{x \leq L} x \bar{\mu}_{\tilde{\mathbf{A}}}(x) \, dx + \int_{x > L} x \underline{\mu}_{\tilde{\mathbf{A}}}(x) \, dx}{\int_{x \leq L} \bar{\mu}_{\tilde{\mathbf{A}}}(x) \, dx + \int_{x > L} \underline{\mu}_{\tilde{\mathbf{A}}}(x) \, dx} \quad (6)$$

and

$$c_r(R) = \frac{\int_{x \leq R} x \underline{\mu}_{\tilde{\mathbf{A}}}(x) \, dx + \int_{x > R} x \bar{\mu}_{\tilde{\mathbf{A}}}(x) \, dx}{\int_{x \leq R} \underline{\mu}_{\tilde{\mathbf{A}}}(x) \, dx + \int_{x > R} \bar{\mu}_{\tilde{\mathbf{A}}}(x) \, dx}. \quad (7)$$

The current KM algorithms are used to find the centroid of IT2 FSs with only 1-D *primary variable*. In the next sections, we propose an algorithm to find the centroid of IT2 FSs with 2-D *primary variable*.

### III. METHOD 1: COMPUTING CENTROID BOUNDARIES OF TWO DIMENSIONAL IT2 FSS

In this section, we consider the most intuitive brute force method to obtain the approximate bound for the centroid of a

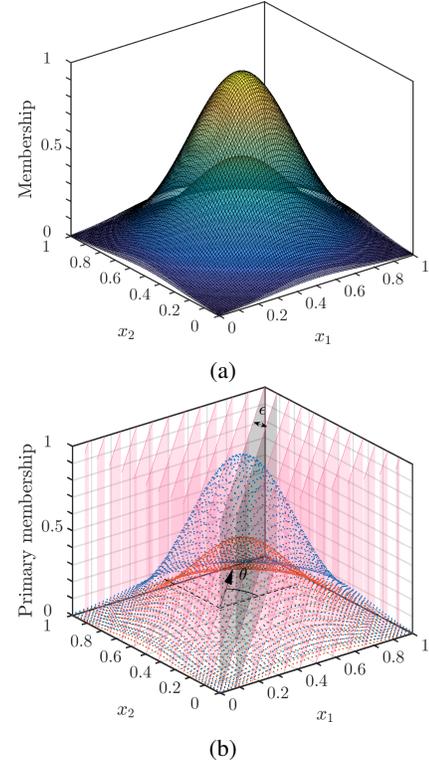


Fig. 1. Illustration of (a) 2-D IT2 FMF  $\tilde{\mathbf{A}}$  and (b) vertical slices on  $\tilde{\mathbf{A}}$  with a highlighted slice  $\mathbf{x}_{\theta,i}$ .

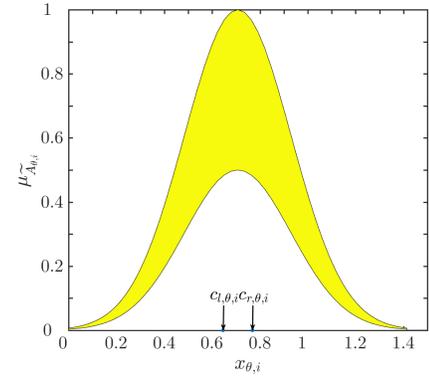


Fig. 2. A 1-D IT2 FS created using the vertical slice and corresponding left and right centroids  $c_{l,\theta,i}$  and  $c_{r,\theta,i}$ .

2-D IT2 FS  $\tilde{\mathbf{A}}$  described above.

For a 1-D IT2 FS, the KM algorithm provides a good approximation to the left and rightmost centroid bounds,  $c_l(L)$  and  $c_r(R)$ , given by (6) and (7), respectively. For a 2-D IT2 FS, analogous to centroid bounds (6) and (7), we consider a closed contour that bounds the centroid which we need to estimate. We express this contour as a subset of  $C(\tilde{\mathbf{A}})$  where each element of this set has the maximum perpendicular distance from the line having a normal vector  $\hat{\theta}$  creating an angle  $\theta$ , for some  $\theta \in [0, \pi]$ , from a reference axis (say  $\hat{x}_1$ ) and passing through the origin. This is represented as  $c_{\theta}(\tilde{\mathbf{A}})$ . Essentially, the contour is the set of  $c_{\theta}(\tilde{\mathbf{A}})$  ( $\theta \in [0, \pi]$ ), where

each  $c_\theta(\tilde{\mathbf{A}})$  is mathematically defined as

$$c_\theta(\tilde{\mathbf{A}}) = c\left(\arg \max_{\mu_{\mathbf{A}_e} \in [\underline{\mu}_{\tilde{\mathbf{A}}}, \overline{\mu}_{\tilde{\mathbf{A}}}] } c(\mathbf{A}_e) \cdot \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}\right). \quad (8)$$

For computing  $c_\theta(\tilde{\mathbf{A}})$ , we consider vertical slices of a 2-D IT2 FS, as shown in Fig. 1, and execute the KM algorithm on each slice to obtain its centroid bounds ( $c_l(L)$  and  $c_r(R)$ ), as illustrated in Fig. 2.

We utilize these centroids in forming an ‘‘embedded-curve’’ (explained later in this section). We define ‘‘slices’’ as subsets of the domain which are obtained by taking points on the domain between pairwise equidistant planes as shown in Fig. 1. For each  $\theta$ , we collect slices that are formed by planes having parallel normal vectors. Slice  $i$  of the domain of the 2-D IT2 FS is defined as

$$\mathbf{x}_{\theta,i} = \{\mathbf{x} \mid i\epsilon - \frac{\epsilon}{2} \leq \mathbf{x} \cdot \begin{bmatrix} -\sin \theta \\ \cos \theta \end{bmatrix} < i\epsilon + \frac{\epsilon}{2}, \mathbf{x} \in X\}, \quad (9)$$

where  $\theta \in [0, \pi]$ , and  $\epsilon$  is a parameter that defines the thickness of the slices. A corresponding 1-D IT2 FS is defined as

$$\tilde{\mathbf{A}}_{\theta,i} = \int_{\mathbf{x} \in \mathbf{x}_{\theta,i}} \int_{u \in \mu_{\tilde{\mathbf{A}}}(\mathbf{x})} 1 / \left( \mathbf{x} \cdot \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}, u \right). \quad (10)$$

Each point on a slice  $\mathbf{x}_{\theta,i}$  is represented as its distance from the line passing through the origin creating an angle of  $\pi/2 + \theta$  from the reference axis as

$$\mathbf{x} \cdot \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} = 0. \quad (11)$$

We now execute the KM algorithm on each  $\tilde{\mathbf{A}}_{\theta,i}$  to obtain its centroid. The two centroids  $c_{l,\theta,i}$  and  $c_{r,\theta,i}$  as shown in Fig. 2, obtained by applying KM on  $\tilde{\mathbf{A}}_{\theta,i}$  are 1-D transformed points that need to be transformed back to their corresponding 2-D coordinates. This can be achieved by rotating them anti-clockwise by an angle  $\theta$ . The transformation is performed by multiplying the coordinates with a rotation matrix expressed as

$$\mathbf{x}_{\theta,i}^* = [c_{r,\theta,i} \ i\epsilon] \cdot \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \quad (12)$$

and

$$\mathbf{x}_{\theta+\pi,i}^* = [c_{l,\theta,i} \ i\epsilon] \cdot \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}. \quad (13)$$

The 1-D embedded sets over the original 2-D domain corresponding to  $\mathbf{x}_{\theta,i}^*$  and  $\mathbf{x}_{\theta+\pi,i}^*$  are represented by  $\mathbf{A}_{\theta,i}$  and  $\mathbf{A}_{\theta+\pi,i}$ , respectively. We define a curve formed by joining the set of points  $\{\mathbf{x}_{\theta,i}^* : i \in \mathbb{Z}\}$  for some  $\theta \in [0, \pi]$  as an ‘‘embedded-curve’’  $S_\theta$ , and the corresponding embedded set is defined as

$$\mathbf{A}_\theta = \int_{i \in \mathbb{Z}} \left( \int_{\mathbf{x} | \mathbf{x} \cdot [\cos \theta \ \sin \theta]^\top \leq c_{r,\theta,i}} \frac{\mu_{\tilde{\mathbf{A}}}(\mathbf{x})}{\mathbf{x}} + \int_{\mathbf{x} | \mathbf{x} \cdot [\cos \theta \ \sin \theta]^\top > c_{r,\theta,i}} \frac{\overline{\mu}_{\tilde{\mathbf{A}}}(\mathbf{x})}{\mathbf{x}} \right), \quad (14)$$

(and similarly  $\mathbf{A}_{\theta+\pi}$  can be defined) as shown in the Fig. 3,

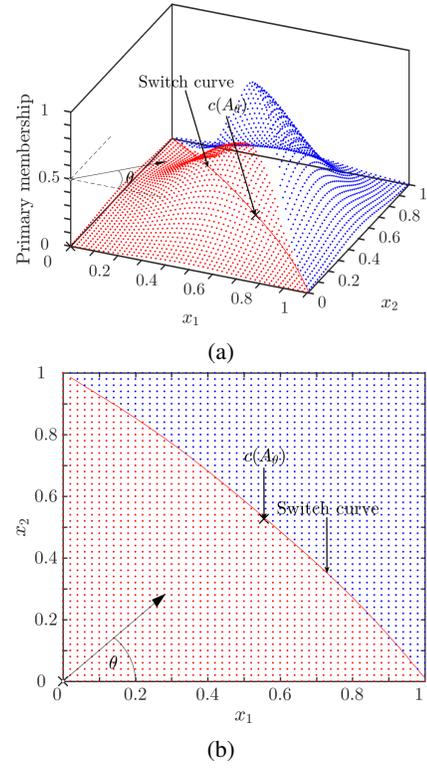


Fig. 3. Embedded set corresponding to an embedded-curve for direction  $\theta$ : (a) side view and (b) top view.

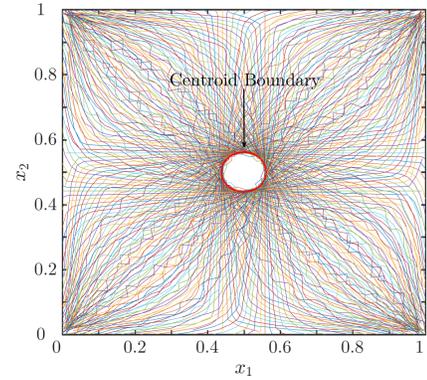


Fig. 4. Centroid bound computed using Method 1 for 2-D IT2 FS  $\tilde{\mathbf{A}}$ .

and the centroid  $c(\mathbf{A}_\theta)$  is calculated using (4).

We now recapitulate the terms defined and propose our method for computing the centroid boundary of a 2-D IT2 FS as follows. For each  $\alpha \in [0, 2\pi]$ , we obtain an embedded-curve  $S_\alpha$  and hence a corresponding embedded set ( $\mathbf{A}_\alpha$ ). For each embedded set  $\mathbf{A}_\alpha$ , we have a centroid given by  $c(\mathbf{A}_\alpha)$  which we take as an approximation for  $c_\alpha(\tilde{\mathbf{A}})$ . A curve joining all the points in  $\{c(\mathbf{A}_\alpha) : \alpha \in [0, 2\pi]\}$  is the required centroid boundary. All the embedded-curves and the centroid boundary are illustrated in Fig. 4.

Now, we explain how  $c(\mathbf{A}_\theta)$  approximates  $c_\theta(\tilde{\mathbf{A}})$  i.e.

$$c(\mathbf{A}_\theta) \approx c_\theta(\tilde{\mathbf{A}}). \quad (15)$$

Analogous to the way we obtain switch-points when executing the KM algorithm on a 1-D IT2 FS, likewise embedded-curves are obtained in the case of 2-D IT2 FS. In the KM algorithm, two switch-points, one for the leftmost centroid and the other for the rightmost centroid, are obtained. Similarly, for a 2-D IT2 FS we obtain embedded-curves for each  $\alpha \in [0, 2\pi]$ . The definition of an embedded set corresponding to an embedded-curve is an extrapolation of a 1-D embedded set giving centroid  $C_r$  from switch-point ‘‘R’’ in the case of a 1-D IT2 FS [2]. Points on the domain that are on one side of an embedded-curve are assigned the upper membership value, and points that are on the other side of the embedded-curve are assigned the lower membership value. The reasoning is that an embedded set with highest membership values for data patterns lying farther away from the line,  $\mathbf{x} \cdot [\cos \alpha \ \sin \alpha]^\top = 0$ , and lowest membership values for data patterns lying closer to the same line, gives a centroid that is farthest from the line, for an initially considered  $\alpha$ .

The centroid of a 2-D embedded FS by definition is the weighted mean of the data points, and the weights are the membership values of the points as given by the embedded set [18]. We can observe that  $c_\theta(\tilde{\mathbf{A}})$  is essentially the weighted average of the centroids  $\mathbf{x}_{\theta,i}^*$  of all 1-D IT2 FSs generated by the slices  $\mathbf{x}_{\theta,i}$ , and the weights being the sum of the membership values of the points for the slice, given by

$$W_{\theta,i} = \sum_x \mu_{\mathbf{A}_{\theta,i}}(x). \quad (16)$$

Thus, the centroid  $c(\mathbf{A}_\theta)$  is equal to the weighted average of  $\mathbf{x}_{\theta,i}^*$  given by

$$c(\mathbf{A}_\theta) = \frac{\sum_{i \in \mathbb{Z}} \mathbf{x}_{\theta,i}^* W_{\theta,i}}{\sum_{i \in \mathbb{Z}} W_{\theta,i}}. \quad (17)$$

#### IV. METHOD 2: SPEED UP ALGORITHM FOR COMPUTING CENTROID BOUNDARIES OF TWO DIMENSIONAL IT2 FSS

In this section, we propose a method to reduce the computation time of Method 1 by restricting all embedded-curves to be linear in 2-D. Enforcing this assumption simplifies the estimation of  $c_\theta(\tilde{\mathbf{A}})$ , as defined in (8). Instead of considering multiple  $\epsilon$ -thick slices and running the KM algorithm on each slice, we project the entire 2-D domain to a single dimension and run the KM algorithm only once. The 1-D embedded set on which the KM algorithm is executed is expressed as

$$\tilde{\mathbf{A}}_\theta = \int_{\mathbf{x} \in \mathbf{X}} \int_{u \in \mu_{\tilde{\mathbf{A}}}(\mathbf{x})} 1 / \left( \mathbf{x} \cdot \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}, u \right). \quad (18)$$

This equation is very similar to (10). By running the KM algorithm on this 1-D IT2 FS, we approximate the estimation of the embedded-line that best estimates  $c_\theta(\tilde{\mathbf{A}})$ , for multiple discrete  $\theta \in [0, \pi]$ .

By executing KM on the 1-D IT2 FS  $\tilde{\mathbf{A}}_\theta$ , we obtain two centroids  $c_{l,\theta}$  and  $c_{r,\theta}$ . The embedded-line corresponding to

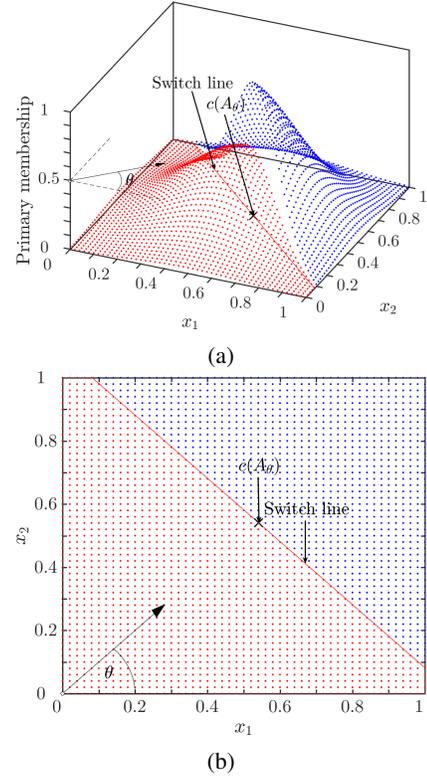


Fig. 5. Embedded set corresponding to an embedded-line for direction  $\theta$ : (a) side view and (b) top view.

$c_{r,\theta}$  is represented as  $L_\theta(\mathbf{x})$  and is the line passing through  $c_{r,\theta}$  that forms an angle  $\pi/2 + \theta$  with the reference axis. The embedded-line is illustrated in Fig. 5(b) and is given by

$$L_\theta(\mathbf{x}) \equiv \mathbf{x} \cdot [\cos(\theta), \sin(\theta)] - c_{r,\theta}. \quad (19)$$

Similarly, the embedded-line corresponding to  $c_{l,\theta}$  is represented as  $L_{\pi+\theta}(\mathbf{x})$  and given by

$$L_{\pi+\theta}(\mathbf{x}) \equiv \mathbf{x} \cdot [\cos(\pi + \theta), \sin(\pi + \theta)] + c_{l,\theta}. \quad (20)$$

The embedded set of the 2-D IT2 FS corresponding to an embedded-line  $L_\alpha(\mathbf{x})$  is shown in Fig. 5(a) and given by

$$\mathbf{A}_\alpha = \int_{\mathbf{x}: L_\alpha(\mathbf{x}) \leq 0} \frac{\mu_{\tilde{\mathbf{A}}}(\mathbf{x})}{\mathbf{x}} + \int_{\mathbf{x}: L_\alpha(\mathbf{x}) > 0} \frac{\bar{\mu}_{\tilde{\mathbf{A}}}(\mathbf{x})}{\mathbf{x}}, \quad (21)$$

where  $\bar{\mu}_{\tilde{\mathbf{A}}}$  ( $\mu_{\tilde{\mathbf{A}}}$ ) is the UMF (LMF) of the IT2 FS  $\tilde{\mathbf{A}}$ . The centroid  $c(\mathbf{A}_\alpha)$  of this embedded set is computed using (4) as shown in the Fig. 5. By connecting all points of the set  $\{c(\mathbf{A}_\alpha) : \alpha \in [0, 2\pi]\}$  where  $\alpha$  takes discrete values in  $[0, 2\pi]$ , we obtain the required centroid boundary. All embedded-lines and the centroid boundary are illustrated in Fig. 6.

It can be observed that this speed up method (with an intrinsic restriction that embedded-curves be lines) is the same as Method 1 when  $\epsilon$  in (9) tends to infinity. When  $\epsilon$  is infinity, only one slice is considered throughout the domain and hence there exists a single point that represents embedded-curve  $S_\alpha$

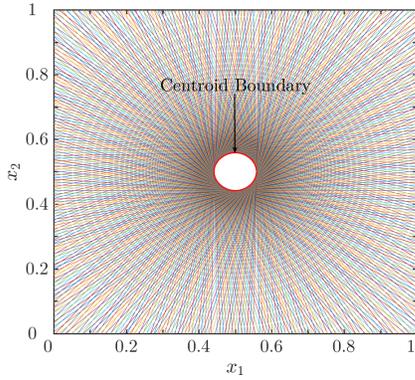


Fig. 6. Illustration of centroid bound computed using the speed up method for 2-D IT2 FS  $\tilde{\mathbf{A}}$ .

for some  $\alpha \in [0, 2\pi]$ . The final candidate for the centroid boundary, as explained earlier, is given by (17). Likewise, for this speed up method, we obtain the same approximation for  $c_\theta(\tilde{\mathbf{A}})$  by forming an embedded-line as given in (19). This is achieved by considering  $\epsilon$  tending to infinity in Method 1 since the primary membership values of the embedded set that approximates  $c_\theta(\tilde{\mathbf{A}})$  results in the same for both methods for every point in the domain. Since this method requires only a single execution of the KM algorithm per  $\alpha \in [0, 2\pi]$ , this may be considered to be significantly faster than Method 1.

It is to be noted that in all the illustrations above, the UMF and LMF of the 2-D IT2 FMF were obtained from gaussian surfaces with no correlation between the two dimensions. In the following example, we illustrate the results obtained by our methods for a 2-D IT2 FMF whose UMF and LMF are obtained from positively correlated 2-D gaussians, as shown in Fig. 7(a). Fig. 7(b) illustrates the embedded-curves and the centroid boundary produced by Method 1 described in Section III. Fig. 7(c) illustrates the embedded-lines and the centroid boundary produced by the speed up method.

## V. CONCLUSION

In this paper, we proposed methods for computing the centroid boundary of a 2-D IT2 FMF. We introduced a generalized concept of a centroid boundary and formally defined it as a collection of points with each point being the farthest centroid, in a particular direction, of some embedded set. In Method 1, we incorporated a greedy approach by breaking down the domain into “slices” and applied the KM algorithm to find the centroid bound for each slice. The centroids of each slice were joined together to form an “embedded-curve” which gave rise to an embedded set, having the upper membership and lower membership as the primary membership values on either side of the embedded-curve. This method served as a good approximation for obtaining the embedded set we were aiming for, however a significant computation overload is inherent due to the requirement of having the KM algorithm applied numerously. In Method 2, we introduced “embedded-lines” which replaced embedded-curves. We projected the 2-D IT2 FS onto a single dimension which significantly reduced the

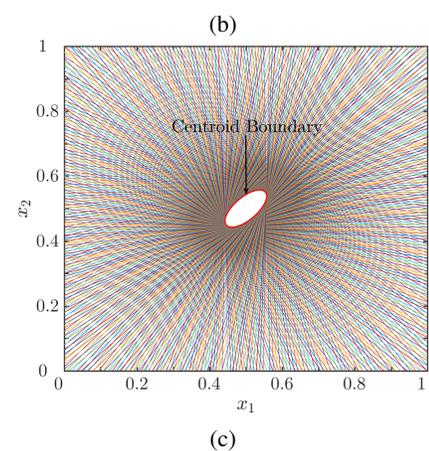
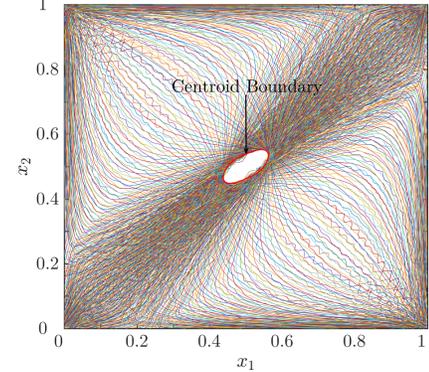
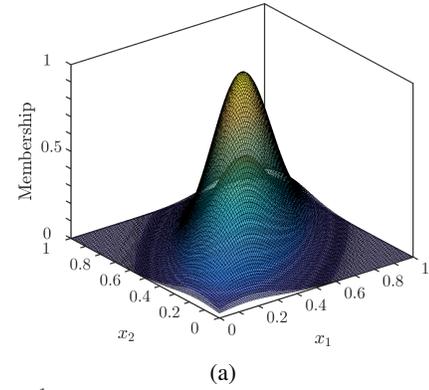


Fig. 7. Proposed methods applied to an IT2 FMF generated from skewed gaussian to obtain centroid bound: (a) 2-D IT2 FS generated from skewed gaussian, (b) result of Method 1, and (c) result of Method 2.

computational complexity. Simulations showed that the decrease in computation load was profound when we used EKM or EIASC. It is to be noted that the two proposed methods involved parameter  $\epsilon$  and the resolution for discretizing  $\alpha$  in the continuous range  $[0, 2\pi]$ . As these parameters affect the accuracy and speed, their values need to be properly managed.

Type reduction of 2-D fuzzy sets is the first step of obtaining 2-D crisp sets. 2-D fuzzy sets can be used to model the uncertainty in 2-D datasets [19] which are involved in applications such as clustering [10] [11] [6], video traffic modeling [20], pattern recognition [21], path estimation in maps [7] [22].

In the methods proposed in this paper, it can be noted that multiple 2-D points belonging to a thick  $\epsilon$  slice may result in the same 1-D transformed value. All such points are logically considered distinct for the KM algorithm when the domain is discrete (which is always the case in practical computations). Future research may include revisiting different methods of generating fuzzy sets from these slices to properly address this issue, and empirical heuristics to tune the hyper-parameters depending on the sparsity of data. It should also be noted that these methods could as well be extended to higher dimensions on similar lines, with analysis on the optimality of proposed variants.

#### ACKNOWLEDGEMENT

This work was supported by the Technology Innovation Program of the Korea Institute for Advancement of Technology (KTA) granted financial resource from the Ministry of Trade, Industry & Energy, Republic of Korea (No. 2015-122).

#### REFERENCES

- [1] N. Karnik and J. Mendel, "Introduction to type-2 fuzzy logic systems," in *1998 IEEE International Conference on Fuzzy Systems Proceedings, IEEE World Congress on Computational Intelligence (Cat. No. 98CH36228)*, Institute of Electrical and Electronics Engineers (IEEE).
- [2] J. Mendel and F. Liu, "Super-exponential convergence of the karnik-mendel algorithms for computing the centroid of an interval type-2 fuzzy set," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 2, pp. 309–320, 2007.
- [3] O. Castillo, L. Cervantes, J. Soria, M. Sanchez, and J. Castro, "A generalized type-2 fuzzy granular approach with applications to aerospace," *Information Sciences*, vol. 354, pp. 165–177, Aug. 2016.
- [4] L. Cervantes and O. Castillo, "Type-2 fuzzy logic aggregation of multiple fuzzy controllers for Airplane Flight Control," *Information Sciences*, vol. 324, pp. 247–256, Dec. 2015.
- [5] M. Sanchez, O. Castillo, and J. Castro, "Information granule formation via the concept of uncertainty-based information with interval type-2 fuzzy sets representation and Takagi-Sugeno-Kang consequents optimized with Cuckoo search," *Applied Soft Computing*, vol. 27, pp. 602–609, Feb. 2015.
- [6] E. Rubio and O. Castillo, "Interval type-2 fuzzy possibilistic C-means optimization using particle swarm optimization," *Nature-Inspired Design of Hybrid Intelligent Systems*, pp. 63–78, 2017.
- [7] H. Hagrass, "Type-2 fuzzy logic controllers: A way forward for fuzzy systems in real world environments," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5050 LNCS, pp. 181–200, 2008.
- [8] N. Pal, V. Eluri, and G. Mandal, "Fuzzy logic approaches to structure preserving dimensionality reduction," *IEEE Transactions on Fuzzy Systems*, vol. 10, pp. 277–286, Jun. 2002.
- [9] F. C.-H. Rhee, "Uncertain fuzzy clustering: Insights and recommendations," *IEEE Computational Intelligence Magazine*, vol. 2, no. 1, pp. 44–56, 2007.
- [10] C. Hwang and F. C.-H. Rhee, "Uncertain fuzzy clustering: Interval type-2 fuzzy approach to C-means," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 1, pp. 107–120, 2007.
- [11] M. Raza and F. C.-H. Rhee, "Interval type-2 approach to kernel possibilistic C-means clustering," in *IEEE International Conference on Fuzzy Systems*, 2012.
- [12] E. Rubio, O. Castillo, and P. Melin, "A new interval type-2 fuzzy possibilistic C-means clustering algorithm," in *Annual Conference of the North American Fuzzy Information Processing Society - NAFIPS*, 2015.
- [13] D. Wu and J. Mendel, "Enhanced karnik-mendel algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 4, pp. 923–934, 2009.
- [14] K. Duran, H. Bernal, and M. Melgarejo, "Improved iterative algorithm for computing the generalized centroid of an interval type-2 fuzzy set," in *Annual Conference of the North American Fuzzy Information Processing Society - NAFIPS*, 2008.
- [15] D. Wu and M. Nie, "Comparison and practical implementation of type-reduction algorithms for type-2 fuzzy sets and systems," in *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, Institute of Electrical and Electronics Engineers (IEEE), Jun. 2011.
- [16] J. Mendel, H. Hagrass, W. Tan, W. Melek, and H. Ying, *Introduction to Type-2 Fuzzy Logic Control: Theory and Applications*. 2014.
- [17] X. Liu, Y. Qin, and L. Wu, "Fast and direct karnik-mendel algorithm computation for the centroid of an interval type-2 fuzzy set," in *2012 IEEE International Conference on Fuzzy Systems*, Institute of Electrical and Electronics Engineers (IEEE), Jun. 2012.
- [18] N. Karnik and J. Mendel, "Centroid of a type-2 fuzzy set," *Information Sciences*, vol. 132, no. 1–4, pp. 195–220, 2001.
- [19] B. Choi and F. C.-H. Rhee, "Interval type-2 fuzzy membership function generation methods for pattern recognition," *Information Sciences*, vol. 179, no. 13, pp. 2102–2122, 2009.
- [20] Q. Liang and J. Mendel, "Mpeg vbr video traffic modeling and classification using fuzzy technique," *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 1, pp. 183–193, 2001.
- [21] W. Pedrycz, "Fuzzy logic in development of fundamentals of pattern recognition," *International Journal of Approximate Reasoning*, vol. 5, no. 3, pp. 251–264, 1991.
- [22] N. Baklouti, R. John, and A. Alimi, "Interval type-2 fuzzy logic control of mobile robots," *Journal of Intelligent Learning Systems and Applications*, vol. 4, no. 4, pp. 291–302, 2012.