**Proceedings of the 2007 IEEE Symposium on
Foundations of Computational Intelligence (FOCI 2007)**

Type-2 Fuzzy Sets: Geometric Defuzzification and Type-Reduction

Simon Coupland, *Member, IEEE*

***Abstract*— This paper presents the geometric defuzzifier for generalised type-2 fuzzy sets. This defuzzifier can be executed in real-time and can therefore be applied to control and other real world problems. We believe this to be a significant step forward for generalised type-2 fuzzy logic systems.**

I. Introduction

Type-1 fuzzy logic systems cannot deal appropriately with imprecision and uncertainty [14]. This is because the membership functions of type-1 fuzzy systems are precise. Type-2 fuzzy systems give an improved performance in the face of uncertainty and imprecision due to the membership functions being fuzzy [2], [8]. The type-2 literature is largely focused on interval type-2 fuzzy logic, a cut-down version of generalised type-2 fuzzy logic. This is partly due to a number of successful techniques (e.g. [12], [17]) for reducing the computational complexity of such systems. Some work has been done on reducing the complexity of the logic operators used in a generalised type-2 fuzzy system [13], [3], [6]. However, the main bottleneck in a type-2 fuzzy logic system, defuzzification remains a problem. This paper presents a geometric approach to the defuzzification of a generalised type-2 fuzzy set. It is demonstrated both symbolically and empirically, that the geometric approach yields a massive reduction in computational complexity and consequently a huge increase in execution speed.

The rest of this paper is organised as follows. Section II introduces interval and generalised type-2 fuzzy sets. Section III introduces type-reduction, the only defuzzification technique for generalised type-2 fuzzy set current available. Section IV introduces the notion of geometric defuzzification. Section V describes the geometric defuzzifier for generalised type-2 fuzzy sets. Section VI explores the reduction in computational complexity achieved by the geometric defuzzifer. Section VII concludes this work, stating outcomes and pointing to future avenues of research.

II. Type-2 Fuzzy Sets

Type-2 fuzzy sets have a *fuzzy* membership function, modelling the imprecise nature of a fuzzy membership grade. As the field has developed, two main categories of type-2 fuzzy set have emerged; generalised and interval. Generalised type-2 fuzzy sets model a fuzzy membership grade as a fuzzy number between zero and one. Type-2 interval fuzzy sets model a fuzzy membership grade as a crisp interval in [0,1]. Generalised and interval type-2 fuzzy sets are defined below.

Definition 1: **A Generalised Type-2 Fuzzy Set**
At each value of x, such that $x \in X$, in the generalised type-2 fuzzy set $\widetilde{A}$, i.e., $\mu_{\widetilde{A}}(x)$ maps to a secondary membership function $f(x)$, which maps values in $[0,1]$ to values in $[0,1]$. Let the domain of the secondary membership function denoted by J_x then;

$$\widetilde{A} = \int_{x \in X} \left[\int_{u \in J_x} f_x(u)/u \right] \Big/ x \tag{1}$$

where $J_x \subseteq [0,1]$, $x \in X$, $u \in [0,1]$ and $f_x(u) \in [0,1]$. Adapted from Mendel And John [16].
The membership grade of a type-2 fuzzy set is called a secondary membership function. The secondary membership function maps possible primary membership grades of the set to their respective secondary membership grades. So, the membership grade of a type-2 fuzzy set is itself a fuzzy set.

Definition 2: **A Type-2 Interval Fuzzy Set**
At each value of x, such that $x \in X$, in the type-2 type-2 fuzzy set $\widetilde{A}$, $\mu_{\widetilde{A}}(x)$ maps to a secondary membership function $f(x)$, which map values in $[0,1]$ to values in $\{0,1\}$. Let the domain of the secondary membership function denoted by J_x then;

$$\widetilde{A} = \int_{x \in X} \left[\int_{u \in J_x} 1/u \right] \Big/ x \tag{2}$$

Where $J_x \subseteq [0,1]$, $x \in X$ and $u \in [0,1]$. Adapted from Mendel [15].
Type-2 interval fuzzy sets are a limited version of the generalised type-2 fuzzy set where the secondary membership grade is always 1. This limitation allows type-2 interval fuzzy sets to be processed a great deal more quickly than generalised type-2 fuzzy sets. This can be exploited is by modelling an interval type-2 fuzzy set as two type-1 fuzzy sets, one for the upper and one for the lower bound of the membership function. This model has led to many new and more efficient ways of performing logical operations.

III. Type-Reduction

Type-2 fuzzy sets become useful when deployed in a type-2 fuzzy logic system as depicted in Fig. 1. The job of a type-2 fuzzy logic system is to take decisions by logical

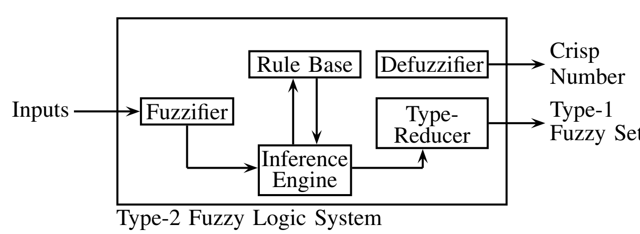

Fig. 1. The Architecture of a Type-2 Fuzzy Logic System.

Simon Coupland is with the Centre for Computational Intelligence at De Montfort University, The Gateway, Leicester, LE1 9BH, United Kingdom. E-mail simonc@dmu.ac.uk

reasoning, reaching crisp conclusions based on crisp inputs. In this deductive reasoning process, type-2 fuzzy sets are used to model the concepts being reasoned with. Therefore the truth of a proposition is measured as an interval when interval type-2 fuzzy sets are used and is measured as a type-1 fuzzy set when generalised type-2 fuzzy sets are used. It is this complex model of a proposition of truth that gives type-2 fuzzy logic its expressive power and the ability to better cope with uncertainties.

This sophisticated reasoning system with additional modelling capabilities also requires additional processing. The major bottleneck when processing a type-2 fuzzy system is type-reduction [17]. So serious is this computational problem that prior to this research, no time-critical applications of generalised type-2 fuzzy logic have been reported. For interval systems the Karnik-Mendel iterative algorithms [12] and later the Wu-Mendel [17] minimax uncertainty bounds have overcome this bottleneck.

Type-reduction [12] takes a type-2 fuzzy set and reduces it to a type-1 fuzzy set. Type-reducing an interval type-2 fuzzy set results in a crisp interval. Type-reducing a generalised type-2 fuzzy set results in a type-1 fuzzy set.

A. Type-Reducing Generalised Type-2 Fuzzy Sets

Broadly speaking type-reduction works as follows. The Mendel-John representation theorem [16] formalises the notion that a type-2 fuzzy set can be represented as a collection of embedded type-2 fuzzy sets. Each of these embedded type-2 fuzzy sets has a centroid that can be calculated a number of ways (centroid, centre of set, height). Each of these centroid values provides a point in the domain of the type-reduced set. The membership grade of a point is found by taking a t-norm of all the secondary grades of the embedded set that produced that point. More formally:

Definition 3: **Type-Reduction: The Generalised Centroid**

The generalized centroid (GC) gives a possibilistic distribution of the centroids of a type-2 fuzzy set. Let $\widetilde{A}$ be a discrete type-2 fuzzy set with L discrete points in its domain. Let n be the number of embedded type-2 sets required to represent $\widetilde{A}$ using the Mendel-John representation theorem [16]. The generalised centroid of $\widetilde{A}$ may be given as

$$GC_{\widetilde{A}} = \sum_{i=1}^{n} [\bigstar_{j=1}^{L} \mu_{\widetilde{A}_e^i}(x_j, u_j)] \Bigg/ \frac{\sum_{j=1}^{L} x_j u_j}{\sum_{j=1}^{L} x_j} \quad (3)$$

where $\mu_{\widetilde{A}_e^i}(x_j, u_j)$, x_j and u_j follow from the definition of a type-2 embedded set given in equation (4) and $\bigstar_{j=1}^{L} \mu_{\widetilde{A}_e^i}(x_j, u_j)$ is the t-norm of all values of $\mu_{\widetilde{A}_e^i}(x_j, u_j)$ from 1 to L.

For discrete universes of discourse X and U, an embedded type-2 set $\widetilde{A}_e$ has N elements, where $\widetilde{A}_e$ contains exactly one element from $J_{x_1}, J_{x_2}, \ldots, J_{x_N}$, namely $u_1, u_2, \ldots, u_N$, each with its associated secondary grade, namely $f_{x_1}(u_1), f_{x_2}(u_2), \ldots, f_{x_N}(u_N)$, i.e.,

$$\widetilde{A}_e = \sum_{i=1}^{N} [f_{x_i}(u_i)/u_i]/x_i \quad u_i \in J_{x_i} \subseteq U = [0,1] \quad (4)$$

where $\widetilde{A}_e^j$ is the j^{th} embedded set in $\widetilde{A}$ and M_i is the number of points in the domain of the i^{th} secondary membership function of $\widetilde{A}$. The number of embedded sets n, within a type-2 fuzzy set is given by equation (5).

$$n = \prod_{i=1}^{N} M_i \quad (5)$$

As the number of points in the the primary and secondary domains increase, the number embedded sets that need to be processed increases dramatically. Figure 2 depicts this increase for primary and secondary domain cardinalities of 0 to 10. Note the logarithmic scale on the y-axis of Figure 2.

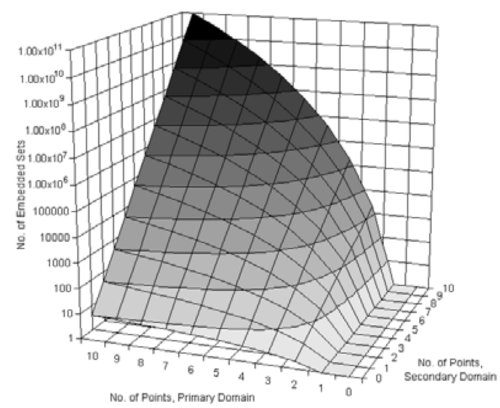

Fig. 2. Number of Embedded Sets Required to Model Type-2 Fuzzy Set of Primary and Secondary Discretisation Levels Between Zero and Ten.

B. Type-Reducing Interval Type-2 Fuzzy Sets

For generalised type-2 fuzzy sets the generalised centroid is currently the only method available for defuzzifying a type-2 fuzzy set, other than the geometric defuzzifier presented in this paper. For interval type-2 fuzzy sets, in addition to the generalised defuzzifier there are the Karnik-Mendel [12] iterative algorithms and the Wu-Mendel [17] minimax uncertainty bounds. The generalised centroid of a type-2 interval fuzzy set $\widetilde{A}$ over the domain X is given below.

$$GC_{\widetilde{A}} = \int_{\theta_1 \in J_{x_1}} \cdots \int_{\theta_1 \in J_{x_N}} 1 \Bigg/ \frac{\sum_{i=1}^{N} x_i \theta_i}{\sum_{i=1}^{N} \theta_i} = [C_l, C_r] \quad (6)$$

where J_{x_N} is the secondary membership grade at N in the secondary membership function J_x and $x \in X$. Since it is a crisp interval, the type-reduced set C only needs two endpoints to define it, C_l and C_r. Each of these points come from the centroid values of a set that is embedded in $\widetilde{A}$. The Karnik-Mendel iterative algorithms exploit the properties of the centroid operation to find these two sets with a relatively low amount of computational effort. The Wu-Mendel minimax uncertain bounds procedure identifies the four embedded sets within $\widetilde{A}$ that can be combined to give the best approximation of the centroid of $\widetilde{A}$. Unlike the iterative method, the Wu-Mendel uncertainty bounds have a finite level of computation that can be calculated prior to execution. This is critical for real-time control systems.

This Section has discussed the process of type-reduction. No matter which technique is used or whether general or interval type-2 fuzzy sets are the subject, type-reduction is trying to achieve the same goal. This goal is to identify the embedded sets that represent the centroid of a given set. With generalised type-2 fuzzy sets, the sheer number of embedded sets make type-reduction by far the most computationally expensive stage of the inference process. To achieve a result in reasonable computational time a different approach must be taken. The next Section describes such an approach, geometric defuzzification.

IV. Geometric Defuzzification

The centre of area defuzzifier is commonly used in type-1 fuzzy logic. This defuzzifier finds the domain value of the centre of the area encompassed by the type-1 fuzzy sets membership function (see Fig 3). It is natural and intuitive

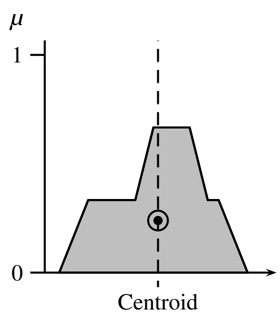

Fig. 3. The Centre of Area of a Type-1 Fuzzy Set

that such an operation gives the centroid of a fuzzy set. If a fuzzy set is characterised by its membership function then the centre of that fuzzy set must surely equate to the centre of that fuzzy sets membership function. This is the approach taken with geometric defuzzification. One way to measure the centroid of a fuzzy set of *any* type is the geometric centre of the area encompassed by that sets membership function.

In [5] we presented a method for finding the geometric centre of the area encompassed by the membership function of a interval type-2 fuzzy set. We now review this method as it forms the basis for the generalised geometric approach described in the next Section of this paper. Let the area encompassed by the membership function of an interval type-2 fuzzy set be a polygon P. P is formed by connecting the upper and lower membership functions of the interval type-2 fuzzy set. Consider the example interval type-2 fuzzy set $\widetilde{A}$ depicted in Figure 4(a). The upper and lower membership functions of $\widetilde{A}$ form the polygon P, depicted in Figure 4(b). One method of finding the centre of the area of the polygon P is to model P as a collection of simple triangles $t_0 \dots t_3$ as depicted in Figures 5 (a) to (c). Each triangle t_i consists of the origin vertex $(0,0)$ and two consecutive vertices from P. This is by no means the only method of modelling P with triangles, this method however, does utilise a computational simple algorithm for identifying these triangles. The centre

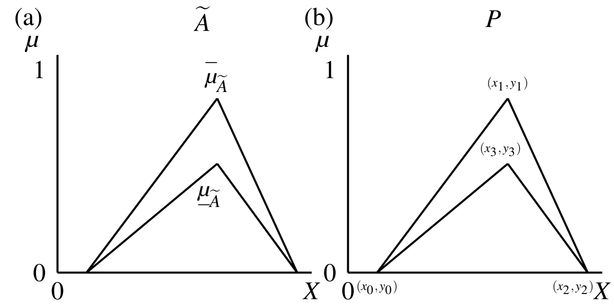

Fig. 4. (a) The Polygon P_{C_R}.

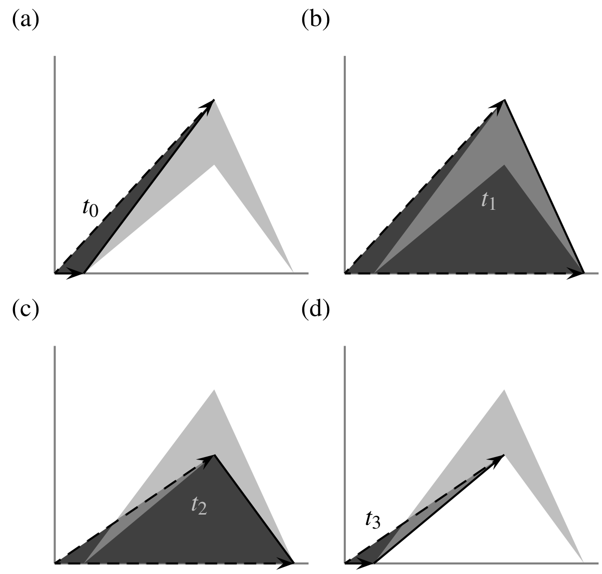

Fig. 5. The Triangles t_0, t_1, t_2 and t_3.

of the area P can then be given as the weighted average of the centroids and areas of each of these triangles [1]. This weighted average calculation is given by equation (7) which reduces to equation (8).

$$C_x = \frac{\sum_{i=0}^{n} \left(\frac{x_i + x_{i+1}}{3} \frac{x_i y_{i+1} - x_{i+1} y_i}{2}\right)}{\sum_{i=0}^{n} \frac{x_i y_{i+1} - x_{i+1} y_i}{2}} \tag{7}$$

$$C_x = \frac{\sum_{i=0}^{n} (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i)}{3\left(\sum_{i=0}^{n} x_i y_{i+1} - x_{i+1} y_i\right)} \tag{8}$$

This Section has given a method for finding the centroid of an interval type-2 fuzzy set using only geometry. This geometric model gives a low computation alternative to type-reduction that complements the Wu-Mendel minimax uncertainty bounds. The approach builds on the geometric model of type-1 fuzzy logic presented in [4]. So, for the type-2 interval approach the geometric model has given an alternative view, with alternative methods that complement existing techniques. The next Section extends these methods, presenting the application of the geometric model to generalised type-2 fuzzy sets. It will be shown that this application yields a massive reduction in computational complexity.

V. The Geometric Defuzzifier for Generalised Type-2 Fuzzy Sets

General type-2 fuzzy sets are truly 3-dimensional entities. The continuous secondary membership grades provide an additional degree of freedom, the third dimension. We believe it is this additional degree of freedom that allows generalised type-2 fuzzy systems to outperform their type-1 and interval type-2 counterparts.

It is clear that 3-dimensional geometric primitives must be used when modelling a generalised type-2 fuzzy set as a geometric object. The membership function of a type-2 fuzzy set is a collection of points in 3-dimensional space. The method for calculating the geometric centroid of this membership function is broken in to two stages:

- Modelling the area encompassed by these points with geometric primitives, and
- Calculating the centre of this area to give a centroid value for the membership function.

A. The Geometric Model of a Generalised Type-2 Fuzzy Set

The way that we propose to model the area encompassed by a type-2 fuzzy sets membership function is both intuitive and simple. Our approach is limited to type-2 fuzzy sets where all the secondary membership functions are convex. This is not a significant limitation. As has been noted in previous work [6], [7], non-convex membership functions are import for type-1 fuzzy sets and the primary membership function of type-2 fuzzy set, but for secondary membership functions non-convexity is not required. A surface that encompasses the area of a type-2 fuzzy sets membership function is found by covering the membership function with a collection of connected 3-dimensional triangles. This covering of the membership function must be done in a methodical manner to ensure the greatest possible accuracy. This is done by breaking down the membership function of a type-2 fuzzy set in to five areas. The triangles need to cover each of the areas comes from a face, a 2-dimensional plane within the 3-dimensional model, or a surface, a specific 3-dimensional area within the 3-dimensional model. The triangles that cover the membership function are therefore constructed as five distinct groups:

1) The triangles that cover the upper surface *a* of the membership function,
2) The triangles that cover the lower surface *b* of the membership function,
3) The triangles that cover the back face *c* of the membership function,
4) The triangles that cover the front face *d* of the membership function, and
5) The triangles that cover the bottom face *e* of the membership function.

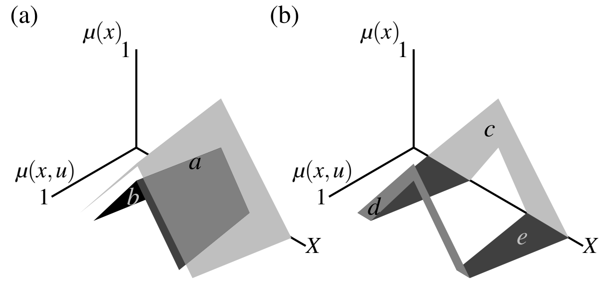

Fig. 6. The Surfaces of the Membership Function of a Type-2 Fuzzy Set.

The areas encompassed by each of these are depicted in Figure 6 (a) and (b) with an example membership function. The combination of all these surfaces forms a solid geometric object, a property that is exploited by the type-2 geometric defuzzifier.

Constructing the triangles that cover faces *c*, *d* and *e* is relatively straight forward since each of these surfaces lays completely in a 2-dimensional plane. Let $\widetilde{A}$ be a generalised type-2 fuzzy set over the discrete domain X consisting of m discrete points. Let each secondary membership function $\mu_{\widetilde{A}}(x_i)$ consist of n discrete points. The set of triangles that cover c is given below.

$$\bigcup \sum_{i=1}^{m-1} \begin{matrix} \{\mu_{\widetilde{A}}(x_i,u_{1_i}),\mu_{\widetilde{A}}(x_{i+1},u_{n_{i+1}}),\mu_{\widetilde{A}}(x_i,u_{n_i})\}, \\ \{\mu_{\widetilde{A}}(x_i,u_{1_i}),\mu_{\widetilde{A}}(x_{i+1},u_{1_{i+1}}),\mu_{\widetilde{A}}(x_{i+1},u_{n_{i+1}})\} \end{matrix} \qquad (9)$$

Equation (9) gives a (classical) set of triangles that cover the FOU of $\widetilde{A}$. These triangles are constructed from the first and last points of adjacent secondary membership grades.

The triangles that cover *d* and *e* can be found in a similar manner. For area *d* any secondary membership functions with more than one point at unity must first be identified. For area *e* any secondary membership functions where $u_1 = 0$ must first be identified.

The triangles that cover areas *c*, *d* and *e* give an accurate model of these areas of the type-2 fuzzy sets membership function. Areas *a* and *b* present a slightly different problem. The surfaces that make up *a* and *b* are non-planar. Modelling a non-planar surface with triangles will always cause some loss of information. We do not believe approximating these surfaces will have a significant impact on the performance of the type-2 system. This assertion is based on the empirical evidence presented in [2].

The method for finding these triangles is simple, however writing this process down is somewhat more complicated. The triangles that cover *a* are formed from the membership grades in $\widetilde{A}$ that are after the apex of the secondary membership function. The triangles that cover *b* are formed from the membership grades in $\widetilde{A}$ that are before the apex of the secondary membership function. Any points in between are covered by *d*. Broadly speaking, the method works as follows; form triangles that model the function from left to right, moving from the edges of each secondary membership function towards the apex.

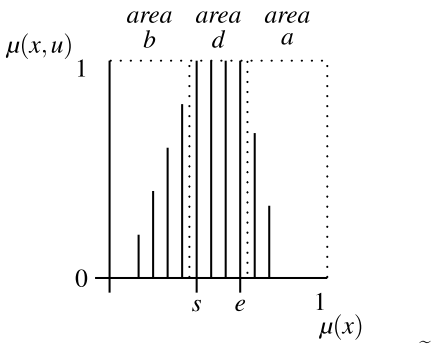

Fig. 7. A Secondary Membership Function in $\widetilde{A}$

Our method will now be given in a more formal manner. Let A be a generalised type-2 fuzzy set over the discrete domain X consisting of m discrete points. Let each secondary membership function $\mu_{\widetilde{A}}(x_i)$ consist of n discrete points. Let the largest secondary membership grade at point x_i be M_i. Let the first point in $\mu_{\widetilde{A}}(x_i)$ with a value of M_i be at s^{th} point in the domain. Let the last point in $\mu_{\widetilde{A}}(x_i)$ with a value of M_i be at e^{th} point in the domain. These variables are further clarified in Figure V-A. The set of triangles that cover a is given below:

$$\bigcup \sum_{i=1}^{m-1} \sum_{j=n}^{e+1} \begin{array}{l} \{\mu_{\widetilde{A}}(x_i,u_j), \mu_{\widetilde{A}}(x_i,u_{j-1}), \mu_{\widetilde{A}}(x_{i+1},u_j)\}, \\ \{\mu_{\widetilde{A}}(x_i,u_{j-1}), \mu_{\widetilde{A}}(x_{i+1},u_{j-1}), \mu_{\widetilde{A}}(x_{i+1},u_j)\} \end{array} \quad (10)$$

The set of triangles that cover b is given below:

$$\bigcup \sum_{i=1}^{m-1} \sum_{j=1}^{s-1} \begin{array}{l} \{\mu_{\widetilde{A}}(x_i,u_j), \mu_{\widetilde{A}}(x_i,u_{j+1}), \mu_{\widetilde{A}}(x_{i+1},u_{j+1})\}, \\ \{\mu_{\widetilde{A}}(x_i,u_j), \mu_{\widetilde{A}}(x_{i+1},u_{j+1}), \mu_{\widetilde{A}}(x_{i+1},u_j)\} \end{array} \quad (11)$$

Essentially, equations (10) and (11) provide a methodical approach to constructing a model of a the membership function of a type-2 fuzzy set with triangles, moving from the edges of each secondary membership function towards the apex. Consider type-2 fuzzy set $\widetilde{A}$ depicted in Figure 8 (a). Two of the triangles that, in our method, would be used to model the upper surface of $\widetilde{A}$ are depicted in Figure 8(b). Two of the triangles that would be used to model the lower surface of $\widetilde{A}$ are depicted in Figure 8(c). With the methods described in this Section we can now model the membership function of a type-2 fuzzy set as a collection of triangles. Such a type-2 fuzzy set is depicted in Figure 9 with equivalent geometric model depicted in Figure 10. The triangles are formed from adjacent points from the membership function of a discrete type-2 fuzzy set. This allows for a simple algorithmic construction method. The triangles in 10 connect in such a way that they form a solid surface. We call such a type-2 fuzzy set a geometric type-2 fuzzy set. The next step is to find the centroid of this geometric type-2 fuzzy set.

B. The Centroid of a Geometric Type-2 Fuzzy Set

The centroid, the centre of this 3-dimensional area can be given by a weighed average of the centroids and areas of the triangles [9]. This is exactly the same approach as applied to interval type-2 fuzzy sets in Section IV. The only difference is that here we are dealing with triangles in 3D rather than

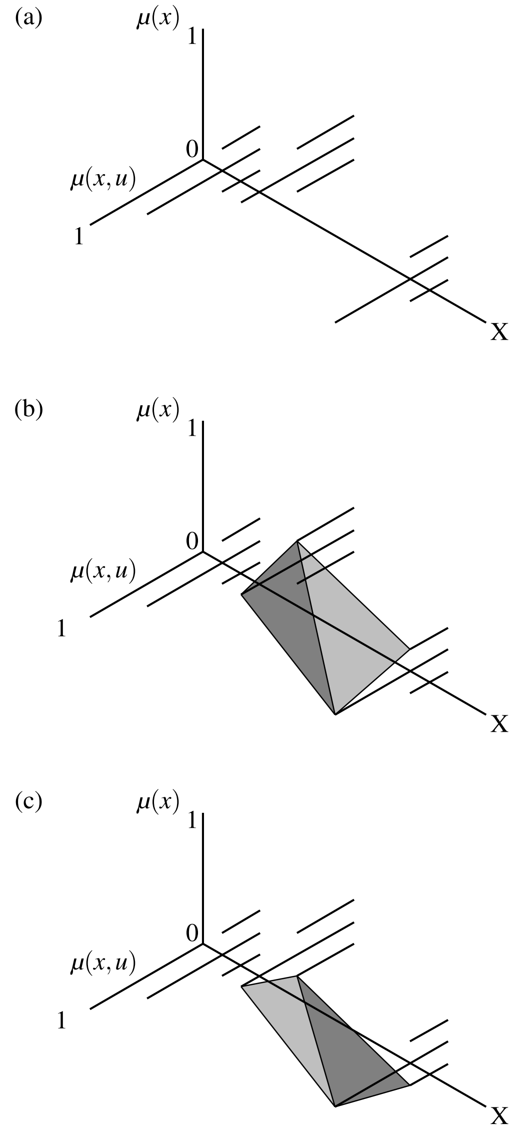

Fig. 8. (a) An Example Type-2 Fuzzy Set. (b) Two Triangles that Model the Upper Surface of the Set. (c) Two Triangles that Model Tthe Lower Surface of the Set.

2D. The area of a triangle is the same in n-dimensions, it's given by half the determinant of the cross product of two edge vectors, as given in equation (12). The domain value of the centroid of a triangle is given by taking the arithmetic mean of the domain values of the three vertices that define the triangle, as given in equation (13). In both equations (12) and (13) the three vertices of a triangle are denoted p, q and r with the component of the vertices being given by a dot notation, $p.x, p.y, p.z$.

$$A_t = \frac{|q \vec{-} p \times r \vec{-} p|}{2} \quad (12)$$

$$C_t = \frac{p.x + q.x + r.x}{3} \quad (13)$$

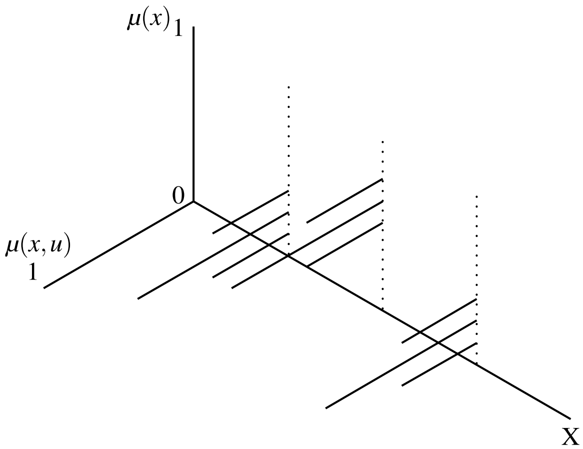

Fig. 9. A Type-2 Fuzzy Set

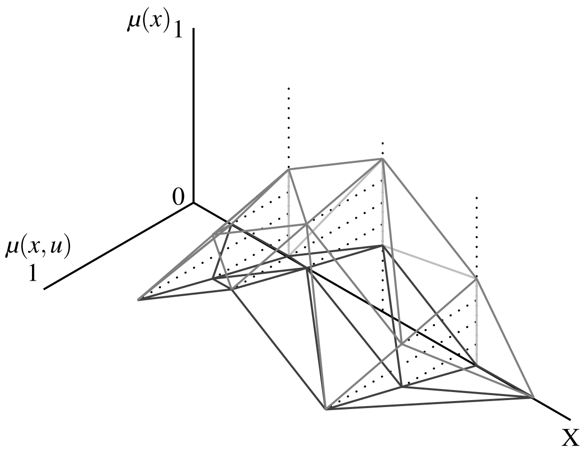

Fig. 10. The Geometric Model of the Type-2 Fuzzy Set Depicted in Figure 9

We can now give a formal definition for the geometric centroid of a type-2 fuzzy set.

Definition 4: Geometric Centroid of a Type-2 Fuzzy Set. Let a generalised type-2 fuzzy set $\widetilde{A}$ be modelled by a set of 3-dimensional triangles t. The centroid C, the centre of area, of $\widetilde{A}$ is given by a weighted average of the centroid and area of every triangle in t, i.e.,

$$C = \frac{\sum_{i=1}^{n} C_{t_i} A_{t_i}}{\sum_{i=0}^{N} A_{t_i}} \tag{14}$$

where t_i is the i^{th} triangle in t and n is the number of triangles in t.

This Section has defined the geometric defuzzifier for a type-2 fuzzy set. Although complicated, this approach gives a massive reduction in computation when compared to the generalised centroid. It is this approach that has, for the first time, allowed a generalised type-2 fuzzy logic controller to be implemented (as reported in [2]). The following Section gives both a symbolic and an empirical comparison of the computational cost of type-reduction and geometric defuzzification for generalised type-2 fuzzy sets.

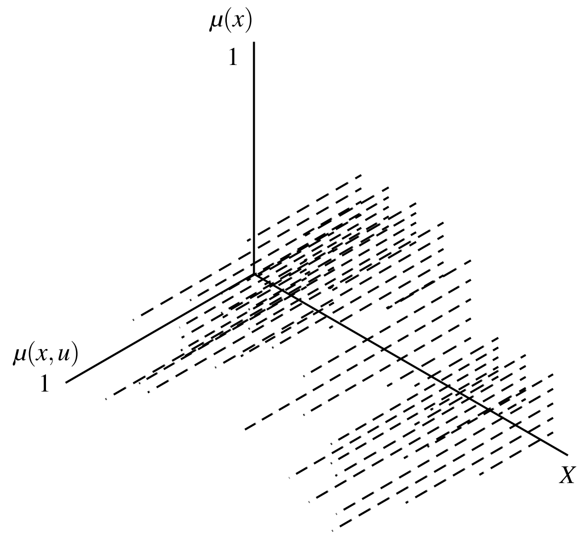

Fig. 11. A Discrete Type-2 Fuzzy Set.

VI. Reduction in Computational Complexity

Having discussed the geometric defuzzifier at length we now set out to demonstrate how this approach achieves the huge reduction in computational complexity that allows for real-time execution.

In this Section we use a worked example to demonstrate the computational differences between type-reduction and geometric defuzzification for a generalised type-2 fuzzy set. The set we will examine is depicted in Figure 11. This set has ten elements in the primary domain and five elements in each of the secondary domains. If we compare this "resolution" of the membership function to that of a type-1 fuzzy set it is clearly not a excessive level of discretisation and some may even think it a little conservative. The number n, of embedded type-2 sets contained within this set is 9,765,625. The geometric equivalent of this set is given in Figure 12. This geometric type-2 fuzzy set was arrived using the algorithm presented in Section V-A of this paper. The number m, of 3-dimensional triangles needed to construct this set is 127. This number follows from the form of the fuzzy set and may be calculated beforehand if require. However, doing so requires

A. Algebraic Comparison

The approach we take in this Section is to, as far as is possible, express the computational cost of the two approaches in terms of the number of floating point calculations. Before we can attempt to compare the level of computation of the two defuzzifier symbolically, a number of terms need to be introduced.

- f_p, the cost of the calculation of the product, division, indices operation of two floating point numbers,
- f_a, the cost of the calculation of the addition or subtraction of two floating point numbers,

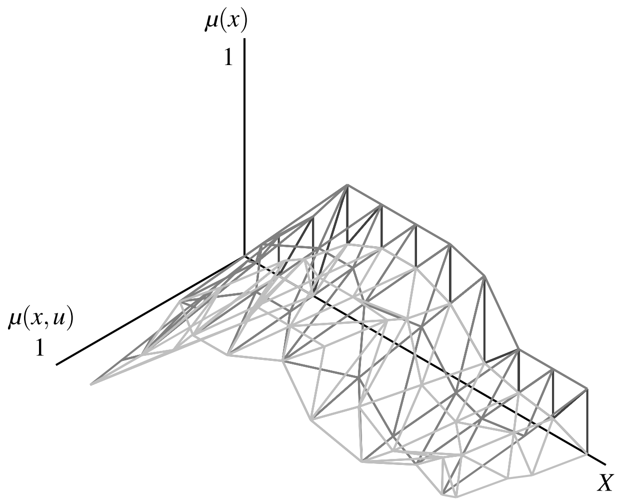

Fig. 12. A Geometric Type-2 Fuzzy Set.

- f_t, the cost of the calculation of the t-norm or t-conorm of two floating point numbers,
- e_i, the cost of enumerating a single type-2 embedded set,
- e_c, the cost of calculating the centroid of a single type-2 embedded set,
- e_t, the cost of finding the t-norm of all secondary grades in a type-2 embedded set,
- t_i, the cost of initialising a single 3-dimensional triangle,
- t_a, the cost of calculating the area of a single 3-dimensional triangle, and
- t_c, the cost of calculating the centroid of a single 3-dimensional triangle.

Throughout this example we assume that the enumerated embedded sets are 95% redundant, we believe this figure to be realistic. By redundant we mean the of the 9,765,625 embedded sets only 488,281 are distinct enough from one another to form a point in the defuzzified set. Embedded sets with very close or identical centroids and t-normed secondary membership functions are not included in the type-reduced set. Assuming this 95% redundancy, the cost of calculating the generalised centroid is therefore:

$$COST_{GC} = n(e_i + e_c + e_t) + 0.05n(2f_a + f_p) + f_p \quad (15)$$

We have no way of knowing the cost of e_i, however since the primary domain has a cardinality of 10, we know the following:

$$COST_{e_c} = 10(2f_a + f_p) + f_p \quad (16)$$

and

$$COST_{e_t} = 10f_t \quad (17)$$

Substituting equations (16) and (17) into (15) gives:

$$COST_{GC} = n(e_i + 10(2f_a + f_p) + f_p + 10f_t) + 0.05n(2f_a + f_p) + f_p \quad (18)$$

which reduces to:

$$COST_{GC} = 20.1nf_a + (11.05n + 1)f_p + 10nf_t + ne_i \quad (19)$$

Moving on to the geometric approach, we know that none of the 127 triangles are redundant, all must be used in the calculation. The cost of the geometric is therefore:

$$COST_{Geo} = m(t_i + t_a + t_c) + m(2f_a + f_p) + f_p \quad (20)$$

We have no way of knowing the cost of t_i, however from equations (13) and (12), we know the following:

$$t_a = 4f_a + 13f_p \quad (21)$$

and

$$t_c = 2f_a + f_p \quad (22)$$

Substituting equations (21) and (22) into (20) gives:

$$COST_{Geo} = m(4f_a + 13f_p + t_i) + m(2f_a + f_p) + f_p \quad (23)$$

which reduces to:

$$COST_{Geo} = 8mf_a + (17m + 1)f_p + mt_i \quad (24)$$

Comparing equations (19) and (24), let us for the moment, ignore the $10e_i$ and t_i terms respectively. If we were to assume n and m were equal then the two operations are, more or less equal in terms of computational complexity. However, we know that this is very much not the case. In our conservative example n is close to 10 million whilst m is only 127. This demonstrates how the massive reduction in computational complexity achieved by the geometric defuzzifier. It is simply achieved through the efficiency of the geometric model, or rather it is achieved because of the massive inefficiency of the generalised centroid.

B. Empirical Comparison

Having shown the huge difference in computational complexity of the two methods, we now explore what this difference means for the execution speed of an actual type-2 fuzzy logic system.

In this example we used an existing type-2 fuzzy logic system designed to perform the task of mobile robot navigation [2]. Each system had an identical rule base consisting of twelve rules, all of which had two fuzzy sets in the antecedent and one fuzzy set in the consequent. The system was given typical inputs, taken from those observed during execution of the system on the robot platform. The time taken for each defuzzifier to calculate the centroid of the final type-2 fuzzy set that was output from the system was recorded over a series of thirty runs. The experiment was conducted using a Dell Dimension PC with a PIII processor running at 450Mhz with 256Mb of RAM under the Fedora 1 distribution of the Linux operating system. The system was executed with the highest possible operating system priority. Table I gives the result of this experiment.

So, in this experiment the geometric defuzzifier was over 200,000 times faster in terms of execution speed than the generalised approach. This is a massive improvement in computational speed which, for the first time, allows generalised fuzzy system to be executed with a reasonable time frame.

	Geometric Defuzzifier	Generalised Centroid
Time	5.81×10^2	1.30×10^8

TABLE I
THE MEAN EXECUTION TIMES (IN MICROSECONDS) OF THE DEFUZZIFIERS OVER 30 RUNS. GIVEN TO 3 SIGNIFICANT FIGURES.

VII. CONCLUSION

This paper has introduced the geometric defuzzifier for generalised type-2 fuzzy sets. The geometric defuzzifier massively reduces the level of computational complexity, and in turn the amount of execution time for defuzzifing a generalised type-2 fuzzy set. This reduction is so large that the real-time execution of a generalised type-2 fuzzy system is now possible. This has significant implications for generalised type-2 fuzzy logic, enabling for the first time application to the control and signal processing domains.

There are important differences between the generalised centroid and geometric defuzzifier. The generalised centroid breaks the defuzzification problem down into many smaller defuzzification problems and aggregates them. Doing this results in a type-reduced set. This set gives a measure of the uncertainty propagated through the fuzzy system. The geometric defuzzifier tackles the problem in one go, translating directly from a generalised type-2 fuzzy set to a crisp number, giving no measure of uncertainty. The two operations are also identifying different properties. The generalised centroid finds a average of the centroids of the embedded fuzzy sets that make up the type-2 set. The geometric defuzzifier finds the centre of the area encompassed by the membership function of the type-2 fuzzy set. These are two quite distinct notions. If it were ever needed, then both operations could be easily extended to take account of a further degree of freedom. However, this would lead to an impossibly slow generalised centroid, whereas the geometric defuzzifier would not suffer from a significant increase in computational complexity.

We believe this work opens up a new wide avenue of research into generalised type-2 fuzzy logic. A great deal of research remains to be undertaken in this field, some important tasks are given below:

1) Implementation of a generalised type-2 fuzzy logic system in hardware,
2) Further comparisons of interval and generalised type-2 fuzzy logic,
3) Deeper investigation of the differences between the generalised centroid and type-reduction, and
4) Formal understanding for the loss of information when a generalised type-2 fuzzy set is modelled with geometry.

This new approach allows, for the first time, generalised type-2 fuzzy logic systems to be applied to real world problems. We hope that this work will allow for greater discussion of the usefulness of generalised type-2 fuzzy logic and of the role this important technology has to play in the field of fuzzy logic.

REFERENCES

[1] P. Bourke. Calculating the area and centroid of a polygon). Available at http:// local.wasp.uwa.edu.au/˜pbourke/geometry/lineline/polyarea/ or on CDROM from http:// local.wasp.uwa.edu.au/˜pbourke/, July 1988.
[2] S. Coupland, M. Gongora, R. John, and K. Wills. A Comparative Study of Fuzzy Logic Controllers for Autonomous Robots. In *Proc. IPMU 2006*, pages 1332 – 1339, Paris, France, July 2006.
[3] S. Coupland and R. John. A New and Efficient Method for the Type-2 Meet Operation. In *Proc. FUZZ-IEEE 2004*, pages 959 – 964, Budapest, Hungary, July 2004.
[4] S. Coupland and R. John. Fuzzy Logic and Computational Geometry. In *Proc. RASC 2004*, pages 3 – 8, Nottingham, England, December 2004.
[5] S. Coupland and R. John. Geometric Interval Type-2 Fuzzy Systems. In *Proc. EUSFLAT 2005*, pages 449 – 454, Barcelona, Spain, September 2005.
[6] S. Coupland and R. John. Towards More Efficient Type-2 Fuzzy Logic Systems. In *Proc. FUZZ-IEEE 2005*, pages 236 – 241, Reno, NV, USA, May 2005.
[7] J. M. Garibaldi, S. Musikasuwan, T. Ozen, and R. I. John. A Case Study to Illustrate the Use of Non-Convex Membership Functions for Linguistic Terms. In *Proc. FUZZ-IEEE 2004*, pages 1403 – 1408, Budapest, Hungary, July 2004.
[8] H. Hagras. A Hierarchical Type-2 Fuzzy Logic Control Architecture for Autonomous Mobile Robots. *IEEE Transactions on Fuzzy Systems*, 12:524–539, 2004.
[9] P. S. Heckbert, editor. *Graphics Gems IV*. AP Professional, 1994.
[10] N. N. Karnik and J. M. Mendel. An Introduction to Type-2 Fuzzy Logic Systems. Technical report, University of Southern California, 1998.
[11] N. N. Karnik and J. M. Mendel. Introduction to Type-2 Fuzzy Logic Systems. In *Proc. IEEE World Congress on Computational Intelligence*, pages 915–920, Anchorage, Alaska, USA, 1998.
[12] N. N. Karnik and J. M. Mendel. Centroid of a type-2 fuzzy Set. *Information Sciences*, 132:195–220, 2001.
[13] N. N. Karnik and J. M. Mendel. Operations on Type-2 Fuzzy Sets. *Fuzzy Sets and Systems*, 122:327–348, 2001.
[14] G. J. Klir and T. A. Folger. *Fuzzy Sets, Uncertainty, and Information*. Prentice-Hall, 1988.
[15] J. M. Mendel. *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Prentice-Hall, Upper Saddle River, NJ, 2001.
[16] J. M. Mendel and R. I. John. Type-2 Fuzzy Sets Made Simple. *IEEE Transaction on Fuzzy Systems*, 10(2):117–127, 2002.
[17] H. Wu and J. M. Mendel. Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems*, pages 622–639, October 2002.