# Visualisation of Fuzzy Systems: Requirements, Techniques and Framework

Binh Pham & Ross Brown

Faculty of IT, Queensland University of Technology,
GPO Box 2434 Brisbane AUSTRALIA
{b.pham, r.brown}@qut.edu.au

This is the authors' version of this work. Later published as:
Pham, Binh and Brown, Ross (2005) Visualisation of Fuzzy Systems: Requirements, Techniques and Framework. *Future Generation Computer Systems* 21(7):pp. 1199-1212.



**Abstract.** Complex fuzzy systems exist in many applications and effective visualisation is required to gain insights into the nature and working of these systems, especially in the implication of imprecision, its propagation and impacts on the quality and reliability of the outcomes. This paper presents a holistic approach towards the design of a visualisation system for fuzzy systems. We firstly analyse the requirements for such a visualization system by articulating fundamental ontologies that underpin the structure and operations of fuzzy systems. A software framework using a multi-agent approach is then presented with the aim to facilitate the organisation and flow of complex tasks, their inter-relationships and their interactions with users. Finally, we discuss visualization techniques for fuzzy data and fuzzy rules, and introduce methods to extend and improve some existing techniques.

## 1. Introduction

Many real world problems can be represented as complex fuzzy systems which may involve a large amount of fuzzy data, fuzzy variables and fuzzy relationships. Fuzzy logic has been used extensively to model these systems in many application areas, ranging from engineering, science and medicine to environmental planning and social sciences [25]. While mathematical models are based on algebraic operations (equations, integrals), logic models rely on logic-type connectives (*and*, *or*, *if-then*), often with linguistic parameters, which give rise to rule-based and knowledge-based systems. Fuzzy logic models can combine both of these types of modelling via the fuzzification of algebraic and logical operations [1]. There are three common classes of fuzzy logic models: *information processing models* which describe probabilistic relationships between sets of inputs and outputs; *control models* which control the operations of systems governed by many fuzzy parameters; and *decision models* which model human behaviour by incorporating subjective knowledge and needs, using decision variables [6].

For some applications, fuzzy systems often perform better than traditional systems because of their capability to deal with non-linearity and uncertainty. The main reason is that while traditional systems make precise decisions at every stage, fuzzy systems retain the information about uncertainty as long as possible and only draw a crisp decision at the last stage. Another advantage is that linguistic rules, when used in fuzzy systems, would not only make tools more intuitive, but also provide better understanding and appreciation of the outcomes. However, the complexity arisen from information uncertainty makes it more difficult for a human to understand the way these systems work, especially how to interpret the implication of the imprecision of each variable on its interaction with other variables, and how the propagation of such imprecision affects the level of confidence in the outcomes at every stage. This understanding is required not only by the users of these systems, but also by their designers who seek for ways to optimise the systems. It would be beneficial, therefore, to be able to visualise these effects.

Visualisation has been used extensively during the last decade, but the bulk of research work has been focused on those systems which involve crisp data and crisp relationships. A few current approaches have some limitations due to either their ad hoc nature, or their ability to deal with only a specific aspect of the problem of visualisation of fuzzy systems [2, 4, 5, 11, 12, 16, 17]. In addition, visualisation methods are often focused on data sets and only loosely coupled with the analytical process. It is left to users to decide how they deploy those visualisation tools provided. For an inexperienced user, this might mean many trialand-error attempts to determine how best to obtain insight into specific tasks. The usefulness of a visualisation system would therefore be enhanced if it is driven primarily by those tasks that need to be performed, and not by data sets, because such a system would link more tightly with the analytical process which underpins human understanding and decision making. Another aspect that needs to be considered is how to cater for different types of users. The needs of users of a fuzzy system are very different from those who design such a system, or from those who design the visualisation system.

This paper unifies the work discussed in three previous conference papers [3,18,19] which focused on different aspects of visualization of fuzzy systems. In addition, we present our own methods to provide significant improvement for certain existing techniques.

Section 2 discusses briefly the fundamental ontologies that underpin the characteristics of fuzzy systems and their visualisation requirements. Section 3 provides an overview of the system and focuses on the design of a multi-agent based visualisation framework with the aim to facilitate the organisation and flow of complex tasks, their inter-relationships and their interactions with users. Section 3 presents the structure and activities of each of these agent classes, and a plan for their implementation. Section 4 discusses the relevancy and adaptability of certain visual features to the representation of fuzzy data, and Section 5 deals with the problem of higher spatial dimensions. Section 6 gives a critical review of existing visualization techniques for fuzzy rules, while Section 7 describes our methods for extending and improving some of these techniques.

## 2. Fuzzy Systems and their Visualisation Requirements

To design an effective generic framework for visualization of fuzzy systems, we need to understand their essence: what are they composed of, how things are related to each other, what activities are being performed, and who are the main users of these systems. We presented a thorough analysis on visualisation requirements for fuzzy systems by investigating the fundamental ontologies which underpin the structure and requirements of these systems. To facilitate the understanding of the rest of this paper, we briefly describe these requirements here. More detailed analysis on these requirements was presented in [3,18].

A typical fuzzy system consists of 6 main components: entities, data objects, relationships, events, tasks and outcomes. The *entities* include both physical (e.g. machines, workers) and abstract (e.g. returns of investment). *Data objects* may be represented in different forms: numerical, symbolic (e.g. rules), visual (e.g. diagrams, images) or audio. *Relationships* which underpin the working of a fuzzy system can be classified into five categories: data-data, data-task, data-user, task-task, and user-user. Each of these categories needs to be examined carefully in order to find appropriate visualization methods to facilitate the understanding of these relationships. *Events* change the system state and exert influence on the system performance, hence it is crucial to note and record them. Events may be pre-scheduled, or may occur as a result of another event, or of a user’s interaction. To distinguish the level of complexity of *tasks*, they can be grouped into low-level and high-level tasks. The former includes the computation of numerical data, degree of fuzziness and the operation of fuzzy rules. The latter covers the detection of unusual patterns, data mining, learning process, optimization and prediction. The *outcomes* of a fuzzy system include not only the values of state variables, but also the level of acceptance of quality, degree of confidence, and degree of imprecision of the outcomes.

We wish to examine the visualization requirements for fuzzy systems from user- and task-oriented points of view, so that a user is allowed to interact and select what to visualize and how to do it on the fly. Thus, visualization should be interwoven with the tasks being performed to provide more insights to users and to improve their decision making process. It is also necessary to distinguish three main types of users and their different needs. The *users of fuzzy systems* wish to be able to interpret data and its salient characteristics, to understand the implication of each decision by setting up ‘what-if’ scenarios, and to adapt the system to their individual needs and preferences. The *designers of fuzzy systems*, on the other hand, require information on the internal structures of these systems for planning, verification and analysis. They also seek for conditions under which optimal solutions are obtained at each stage. The *designers of visualization systems* wish to understand how users make use of visualization techniques and the effectiveness of these techniques with the intention to identify drawbacks and to find ways to continuously improve the systems. In addition to some specific requirements by these three types of user, there are some common requirements that could be exploited in order to design an effective generic visualization framework. We categorise four main types of visualization tasks:

- *Interactive exploration* to provide insights into: the degree of uncertainty of each variable and its effects on each task; the inter-dependency of two or more fuzzy variables or fuzzy rules; and the effects of different operations performed on fuzzy rules.
- *Automatic computer-supported exploration* to automatically highlight salient characteristics and unusual results; to display and compare alternatives (e.g. using statistical analysis); to optimize tasks under specified constraints; and to support batch processing of tasks via scripting or visual languages.
- *Capturing feedback from users* such as instructions on tasks; input parameters, variables, constraints; users' preferences, judgements and desired degree of fulfillment of outcomes in qualitative forms.
- *Capturing users' profiles and adaptation* in order to re-organise data and re-prioritise tasks to suit; and to automatically provide tasks and data according detected patterns.

## 3. A Multi-agent Visualisation Framework

Our aim is to design a systematic framework based on a high level of abstraction, where visualisation is driven by users' needs which in turn are driven by application tasks and personal view points. Search and navigation methods and tools should be context-sensitive and should operate only on relevant information space. Thus, data should be organised according to task requirements to ensure efficiency. Multi-agent approach has been increasingly adopted for application domains because it provides an effective way to coordinate activities and their interactions in a complex system to satisfy some common goals. An agent in our context is a computer program that can gather data about the environment, interpret the data and modify its behaviour to reflect the requirements of the environment.

We propose a visualisation framework based on 5 classes of agents: control agent, computation agent, symbolic agent, visualisation agent and profile agent. Fig. 1 shows a schematic diagram of the system architecture.

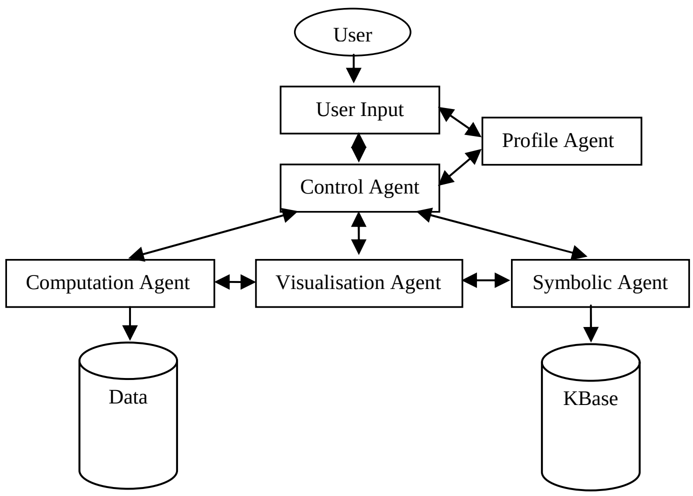



**Fig. 1** System overview diagram.

The *control agent* receives users' input which includes specifications, queries and parameters. Based on such input, this agent distributes tasks to appropriate agents. It also receives results and demands from other agents when a task is completed or when further information is needed. Another duty for this agent is to generate new tasks if required based on the results sent by other agents. The control agent may be viewed as a representative of the user in an automatic mode. In our model, the user can be included in the loop and allowed to intercept the control agent in order to give different instructions if desired. The user can also intercept other agent to select different methods for performing an operation instead of the default ones built in the system. The *computation agent* performs all numerical computation required by the system (e.g. statistics, probabilistic calculus, rough set operations, fuzzy set operations). It receives instructions from both the control agent and the visualisation agent. The *symbolic agent* makes use of the knowledge base to perform rule inferencing. It receives instructions from both the control agent and the visualisation agent. The *visualisation agent* receives instructions from the control agent and requests information from the computation agent and rule agent, in order to select appropriate visualisation techniques to provide displays. The results of the display then trigger the control agent or the user to issue another task. Another cycle then continues.

The *profile agent* records the pattern of the user's behaviour in terms of the selection of tasks, visualisation techniques, numerical methods or inference rules. Based on this information, the profile agent then modifies the instructions issued by the control agent (e.g. re-prioritise tasks, change preferences, modes of display, etc.).

Using an object-based paradigm, the following subsections show, the general methods and variables within the agents as a specification of their functionality. It should be noted that these different agents run as different threads within the memory space of the visualization system, and so can interact with each other, while still allowing the user to interrupt the process to enforce new queries or parameters on the visualization.

### 3.1 Control Agent

The control agent maintains a list of the other agent instantiations within the system and thus is able to control the flow of data around the agent-based visualisation system. It is an essentially autonomous agent, its main tasks are to process user events, distribute tasks to other agents and to process agent results and demands. Therefore it communicates with the profile, computation, visualisation and symbolic agents. It is a form of automatic user within the system. However, the user can override its choices by an interruption process.

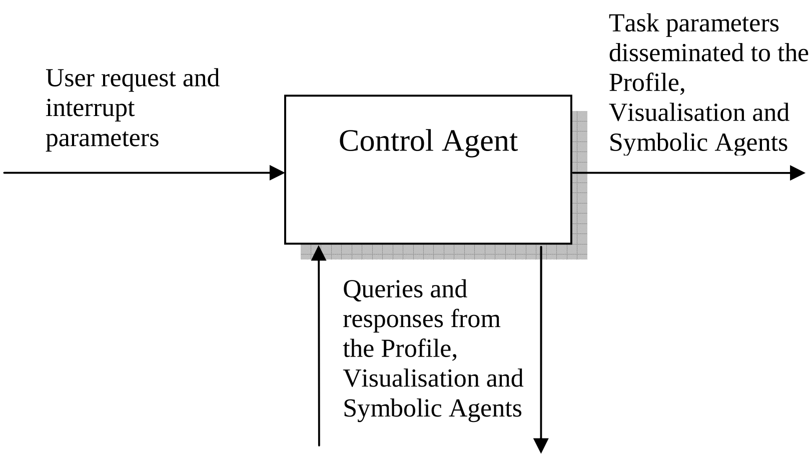



**Fig. 2** Diagram illustrating major data flows for the Control Agent.

### 3.2 Computation Agent

The main data stored by this agent includes the data to be visualised within the system, and a complete list of operations that can be performed by the computation agent. It is not an autonomous agent, as it is entirely controlled by the visualisation and control agents. The computation agent processes the data upon requests from the control and or visualisation agent – for example, statistics, probabilistic calculus, rough set operations, fuzzy set operations, etc.

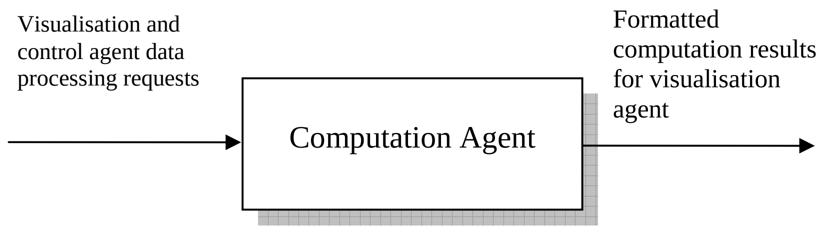



**Fig. 3** Diagram illustrating major data flows for the Computation Agent.

### 3.3 Symbolic Agent

This agent is the interface to the knowledge database for the visualization system. It is used by the control agent and the visualization agent. It is not autonomous, as it simply provides a front end query interface to the knowledge database. This database contains knowledge of appropriate visualisation techniques for the fuzzy data. Thus, the agent returns information about techniques, parameters to use etc., as responses to queries from the control and visualisation agents.

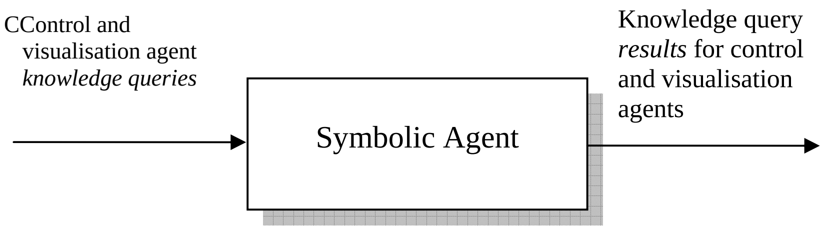



**Fig. 4** Diagram illustrating major data flows for the Symbolic Agent.

### 3.4 Visualization Agent

The visualization agent handles the graphical rendering of data to the output device. This agent draws directions from the symbolic agent, and is able to thus recommend a visualisation technique automatically, based upon the qualities inherent in the data. The data is received from querying the computational agent, which is directed by the visualisation agent to provide the data in a valid format for the visualisation technique. The visualisation agent is fairly autonomous in its ability to organise a visualisation of data. Any information required for the visualisation is queried from the computational and symbolic agents, by using their querying methods. The control agent can, at the behest of the user, override the visualisation agent via an interrupt process.

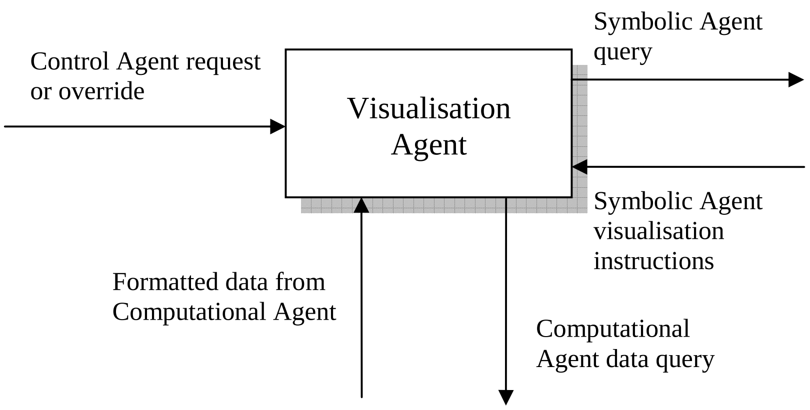


**Fig. 5** Diagram illustrating major data flows for the Visualisation Agent.

As an example, we trace the execution of the visualization of the Iris data shown in [11], which is a commonly used test data set for classification algorithms. The classification is done based on a training set of 75 plants, 11 fuzzy rules, 4 features (sepal length, sepal width, petal length and petal width) and 3 classes. In this case the user commences the system and inputs the Iris data file as the beginning of the visualization. The control agent then commences the dialog by noting the multidimensional nature of the data to be visualized. Information about the nature of the visualization is elicited from the user, who specifies a visualisation for rule culling purposes. The control agent then passes this information onto the visualization agent.

The visualization agent then queries the symbolic agent for suggested visualization techniques. The symbolic agent replies with a suggestion of using the parallel coordinate visualization technique as shown in Fig. 6. In this method, n Cartesian coordinates are mapped into n parallel coordinates, and an n-dimensional point becomes a line connecting the values on n parallel axes. The visualization agent then requests the fuzzy data in an appropriate format for the parallel coordinate technique. The visualization agent then commences the rendering of the 2D parallel coordinate visualisation. However, the user requires a 3D version of the visualization, and chooses this using the appropriate menu options.

The control agent at this stage interrupts the visualization agent thread, and then enforces a 3D parallel coordinate visualization, as shown in Fig. 9 (details on our new 3D parallel coordinate approach will be discussed later in section 7.1). This is then rendered by the visualisation agent, whereupon control is given back to the user to interact with the visualization. This process is then repeated until termination by the user.

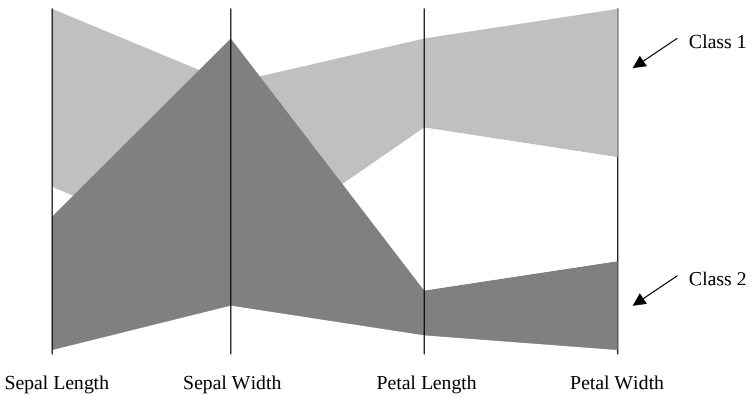



**Fig. 6** 2D parallel coordinates representing 2 fuzzy rules for classification of Iris [11].

### 3.5 Profile Agent

This agent records a user’s profile in various ways: patterns of tasks performed; patterns of usage of data and operations applied on fuzzy rules; specific types of constraints; desired degree of fulfillment of outcomes; and choice of visualization techniques. By communicating with both the user and the Control Agent, the Profile Agent uses these detected patterns to issue instructions to re-organise data and re-prioritise tasks. It also automatically offers choice of operations on fuzzy rules and degree of fulfillment of outcomes. Alternative visualization techniques can also be suggested. Such adaptation would allow a user to gradually customize the visualization system to his or her own application and subjective preferences. This would also remove the tedium of having to go through a fixed pattern of steps which might not be relevant to one’s specific task, as often offered by a hard-wired visualization system.

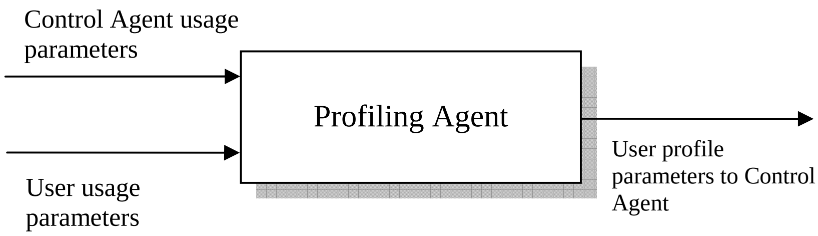



**Fig. 7** Diagram illustrating major data flows for the Profiling Agent.

## 4. Visual Features for Fuzzy Data

In previous work we have analysed in detail the mapping of various visual features to visualisation of fuzzy logic information [3]. In this section, we summarise some of the most importance features and show examples of their mappings to visualisation tasks.

*Hue* is heavily used to highlight data that is different, or to represent gradients in the data [13][22]. It can be used in a number of ways to represent fuzzy scalar data via variation of the *saturation*. The more saturated the hue, the more certain or crisp the value contained in that region is [12]. The number of hue groups used in the mapping of values (*cardinality*) can indicate the level of precision in the values. The less precise solution has fewer variations in hue values, while a more precise solution has a smoother shaded appearance.

For fuzzy data, it can also be used to categorise the membership of a particular data point. In the example in Fig. 8, we see that the membership of different fuzzy terms can be illustrated by different hues. The region where the colours overlap indicates intuitively the location where these membership functions share areas of the domain.

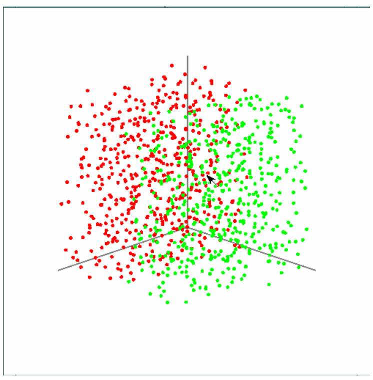


**Fig. 8** Hue differences (shown as grey shades) as an indication of the membership degree for a fuzzy membership function.

*Luminance* may also be used to signify categories and highlight differences within scalar data. The cardinality of the luminance feature can be varied to show the precision of the data in a similar manner to the cardinality of the hue space. Luminance can also be used to directly indicate the Degree of Fulfilment (DOF) value for a single membership function.

The *size* of objects can be used to indicate the scalar component of vector information. An example of this is the variation of the size of error bars surrounding a datum point, to indicate its imprecision [20]. *Transparency* can be used to show underlying structure, but in this context can be used to show the fuzziness of the data by mapping the possibility of the fuzzy variable to the transparency. *Texture* may be applied to objects to indicate the level of precision, ambiguity or fuzziness in the spatial location upon an object or upon a spatial location. Both *glyphs* and *icons* can create a problem and a possibility, as they allow the representation of data using an object or shape etc. This leaves an unending list of possible glyphs to use with regards to visualisation of fuzzy information. *Particles* could be used to represent the fuzziness of a region or an object by varying the space between them, and the colour of the particles themselves (refer to Fig. 8). *Blurring* or *depth of field* effects from spatial frequency components being removed in the image plane can be used to show the indistinct nature of data points [9].

## 5. Higher Spatial Representations

The visual features listed above are usually spatially arranged to form a coherent display in graphic forms, which enable the perception of various patterns in the data. We can combine the use of such visual features for denoting the imprecision in data with a number of common representation methods employed to display spatial data in higher dimensions: 2D, 3D, parametric, dynamic, metaphors and multimedia sensors.

***2D Representations***
2D graphs of various forms can be used to encode the colours and shapes into a display on a Cartesian system, in order to show the spatial relationships of values. These graphs may not necessarily be related to a spatial location. Some examples of graphs are: histograms, bar charts, tree diagrams, time histories of 1D slices, maps, iconic and glyph-based diagrams. The structure and inter-relationships of rules may be illustrated by graphs, trees and flowcharts.

Variation in intensity or colour may be used to encode another dimension on a 2D graph which indicate the degree of imprecision or fuzzy membership functions of the data displayed. Graphs may also be used to represent the fuzzy membership functions or alpha-cuts of a fuzzy set.

Another common technique is to project data for reduction of dimensionality (e.g. Principle Component Analysis) and display results on a scatter plot. However, although this technique provides a high level analysis of the most significant components of the data, it has a drawback due to the loss of information during the process.

Other techniques such as multi dimensional scaling and parallel coordinates [2] provide ways to display multi-dimensional fuzzy data in 2D without losing any information. For multi-dimensional scaling, the authors introduced an algorithm to generate 2D view of a set of fuzzy rules which minimizes the inter-point distances. The rule set is then visualized as a 2D scatter plot, where different grey scales denote different classes and the size of each square denoting each class indicates the number of examples and hence the importance of the class. For the parallel coordinates approach, n Cartesian coordinates are mapped into n parallel coordinates and an n-dimensional point becomes a series of (n-1) lines connecting the values on n parallel axes.

In section 7.1, we shall illustrate how the visualisation of fuzzy rules by parallel coordinates provided by these authors could be improved to make it more intuitive for users by judicious choice of visual features.

***3D Representation***
A 3D volume has spatial regions mapped to a location in n-dimensional space. The features of the volume partitions could be modified to indicate the precision of the data within the volume (e.g. varying intensity, colour saturation, texture, opacity). These techniques may be used to show classification boundaries in fuzzy classification methods. A 3D height-field (expressed as surface) could also be used to represent fuzzy membership functions of data displayed in 2D graphs. To visualize hierarchical information, a cone tree method was introduced by Fujiwara et al. [7] to represent a tree structure.

***Parametric representations***
Different parameters could be used to highlight or suppress various factors in an interactive manner. This method may also be performed in a non-interactive manner as a movie, using fixed temporal effects.

This is useful from a computer human interfaces perspective as the imprecision in the data could be visualised over a number of perceptual feature dimensions to reinforce various combinations, and to allow interaction as another form of visualisation technique.

***Dynamic representations***

Various visual features discussed in the last section could be used to modify the animation to display object behaviour over time, e.g. using motion blur levels, flickering etc. to represent the precision of the measurements of the object motion in a plane crash simulation.

***Metaphors***

As humans can perceive the effects of certain common phenomena at a very fast speed, abstract representations may be used as metaphors to represent data that is not easily visualised. For example, expressions on human faces can be used to represent the quality of the results, where a happy / sad expression indicates good / bad quality.

***Multimedia sensors***

Haptic and audio feedback can be used to indicate precision, imprecision, eg. mapping mouse location to a form of sound that is noisy and incoherent in imprecise regions, and coherent and tonal in regions that are precise.

## 6. Visualisation of Fuzzy Rules

In the last section, we mentioned how parallel coordinates and multiscaling can be used for representing fuzzy data and fuzzy rules. This Section reviews other methods that explicitly cater for fuzzy rules. These methods fall into two categories: to model the causal flow between elements in a fuzzy system; and to represent fuzzy clustering or classification of data. The first category aims to facilitate the understanding of the complex nature of a rule-based fuzzy control system, while the second category helps to analyse the effectiveness of fuzzy clustering algorithms.

### 6.1 Visualising Causal Fuzzy Relationships

Dickerson et al. [4] used fuzzy cognitive map in the form of a graph to encode relationships in a complex interacting system (e.g. relationships between proteins in the expression of genetic information). This technique is useful for encoding expert information which is commonly present in fuzzy control systems.

The 3D cone graph mentioned in the last Section was later used by Gershon [8] to produce a 3D flowchart to represent hierarchical rule structure in a rule-based program to facilitate its understanding. In section 7.2, we shall show how these techniques can be extended to visualizing fuzzy rules, where appropriate visual features can be integrated to the cone tree structure to express the degree of uncertainty in each rule (e.g. each node is displayed with different degree of opacity). The propagation of fuzzy rules can also be visualized by using a truncated double cone structure.

### 6.2 Visualising Fuzzy Data Cluster

Berthold and Holve [2] mapped an N-dimensional fuzzy rule to a point in 2D. The 2D function was constructed so that the error values between two rules are minimized. Rules are represented as squares with different grey intensities to distinguish the clusters. The size of each square indicates the number of data points and the importance of the rule.

Cox et al. [3] used 2D and 3D plots with various thresholds to represent convex hulls of data point clusters. Glyphs with different shape and size are used for the data points. The precision of the representation of the cluster membership depends on the accuracy of the projection of points into the convex hulls. The authors also presented another method of using the colour hues from red (hot) to blue (cold) to represent the membership values of data points. However, although using colour plots to highlight a single cluster is effective for cluster separation, to do so for all cluster at once would cause confusion. Colour hues were also used by Lowe et al. [15] to represent belief values in the form of a flame to facilitate decision making in an anaesthetic monitoring system.

To show the classification boundaries resulted from fuzzy classification methods, Nurnberger et al. [16,17] used 3D illuminated surfaces. These surfaces can clearly demonstrate the effects of t-norm and t-conorm operators on the resultant surfaces. However, they do not show internal details for each class. These details can be easily revealed by incorporating transparency.

## 7. Improvement on Current Methods

We now present two cases where we have extended and improved the capabilities of existing visualization methods for fuzzy data and fuzzy rules. In the first case, we extend the parallel coordinates approach to 3D graphs and 3D polygonal surfaces. In the second case, we integrate different visual techniques to cone trees to allow different types of information to be perceived.

### 7.1 Extension of Parallel Coordinates

Fig. 6 shows an illustration of method developed by Hall and Berthold [11], for representing multidimensional fuzzy rules using 2D parallel coordinates. Two rules are illustrated from their Iris data example, where four features are used for classification (sepal length, sepal width, petal length and petal width). They used the thickness or grey intensity of lines to indicate the fuzziness of points. One drawback is that it is difficult to visually distinguish fine grades of grey level on single lines. Another drawback is that it is not possible to perceive the core and support of a fuzzy set simultaneously.

One way of addressing those limitations on visual perception (discussed in Section 5) is to use a 3D representation where the parallel coordinates are displayed on the x-y plane, and the fuzzy set membership functions are displayed as z-coordinates. Different alpha-cuts of fuzzy rules can be identified by applying horizontal cutting planes. Fig. 9 illustrates of the new 3D parallel visualisation showing the membership functions from Fig. 9, with a superimposed alpha cut plane. The separation of classes based on the confidence of decision can be highlighted by using filled polygons with texture or using colour. The filled polygons highlight membership functions, which classify with a high degree of confidence.

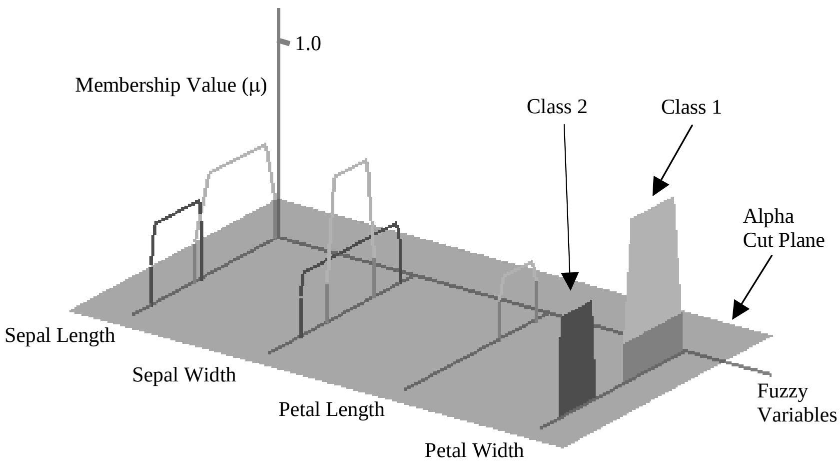



**Fig. 9** New 3D parallel coordinates visualisation showing the membership functions from **Fig. 6**, with a superimposed alpha cut plane.

The perception of different classes can be facilitated further by introducing surfaces with illumination and texture. Fig. 10 demonstrates a visualisation of the same Iris data by lit and textured surfaces. Note how the alpha cutting of the membership function for Rule 2 on the Petal Length dimension is now easily perceived.

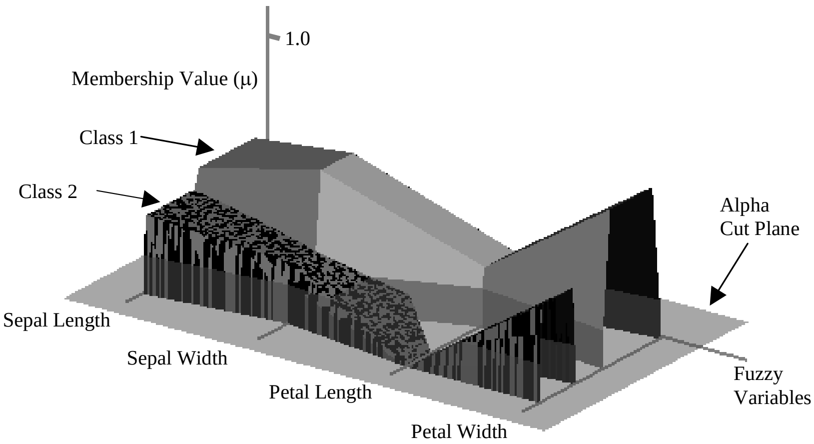



**Fig. 10** Visualisation of the Iris data with lit and textured surfaces showing the same Iris data. Note how the alpha cutting of the membership function for Rule 2 on the Petal Length dimension is easily perceived.

### 7.2. Truncated Double Cone Structure for Visualising Fuzzy Rule Propagation

There are a number of ways to extend the visualization capabilities of the 3D cone trees. To illustrate our ideas, we now present various ways to achieve four purposes:

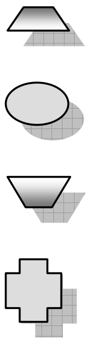
- representing fuzzy data at tree nodes
- representing fuzzy rules including degree of likelihood and important weights
- highlighting dominant rules to create a "pop-out" effect
- illustrating the propagation of fuzzy rules (aggregation, min-max, de-fuzzification)

We introduce a truncated double cone structure to facilitate the visualization and analysis of fuzzy rule propagation. The shape reflects the nature of the propagation starting from a small number of rules to a wider network of rules, then contracting its size during the de-fuzzification stage (Fig. 11). Each set of rules is displayed as a networkon the surface of the double cone. The relationships between sets of rules are represented as connecting lines located within the double cone, which can be made visible by using transparency. This double cone has the following properties:

- Its structure is composed of concentric internal cones and parallel layers occupying the cone from top to bottom;
- Typically the middle is wider than the top and bottom due to the network of rules being resident in the middle;
- The double cone can be rotated to show desired rules;
- All the layers can expand to fit extra rules variables etc. (shown by arrows in diagram);
- The cone can be rotated around to examine the rules (shown by arrows in diagram);
- Rules migrate through the concentric cones depending on arbitrary criteria, eg. the rules/variables that have the largest DOF (Degree of Fulfillment) are situated on the outer shell;
- The whole structure can be flattened out into a plane for normal perusal/printing;
- It can be modified to bring the most important rule (or a particularly selected rule) to the front of the cone for greater emphasis.

In Fig. 11 - Fig. 13, we use the following representations to denote different entities and operators in the tree structure:

- Upright *Trapezoids* for fuzzified variables into the system;
- Inverted *Trapezoids* for defuzzification operators;
- *Circles* for set operations – union and intersection, etc.;
- *Pluses* for aggregation operators;



- Fuzzified input variable

- Fuzzy Operator

- Defuzzification Operator

- Aggregation Operator

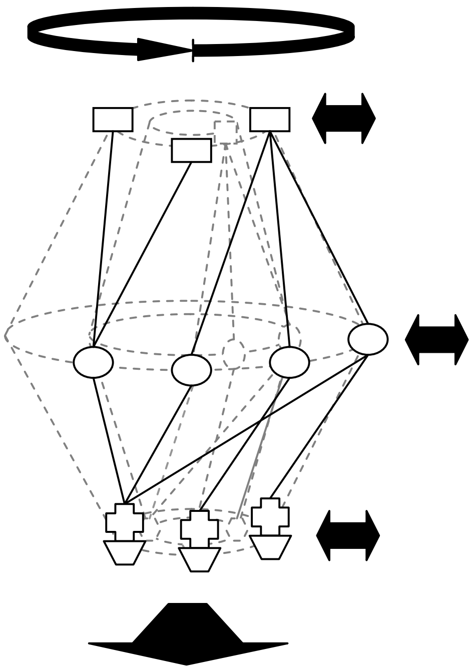


**Fig. 11** Truncated double cone for visualizing fuzzy rules and their propagation.

Fig. 12 and Fig. 13 show two example network diagrams of fuzzy rules on the double cone surface, where

- Icons have appropriate data superimposed onto the location;
- Lines use size, transparency or colour to indicate fulfilment values;

- The output visual feature of the operator mimics the operator itself, eg. output of Centroid defuzzification is the average thickness of the input lines;
- Visual features  are scaled to have the highest and lowest values at the extremes, eg. the highest value is the thickest possible line to induce a popout effect by creating a large enough difference.

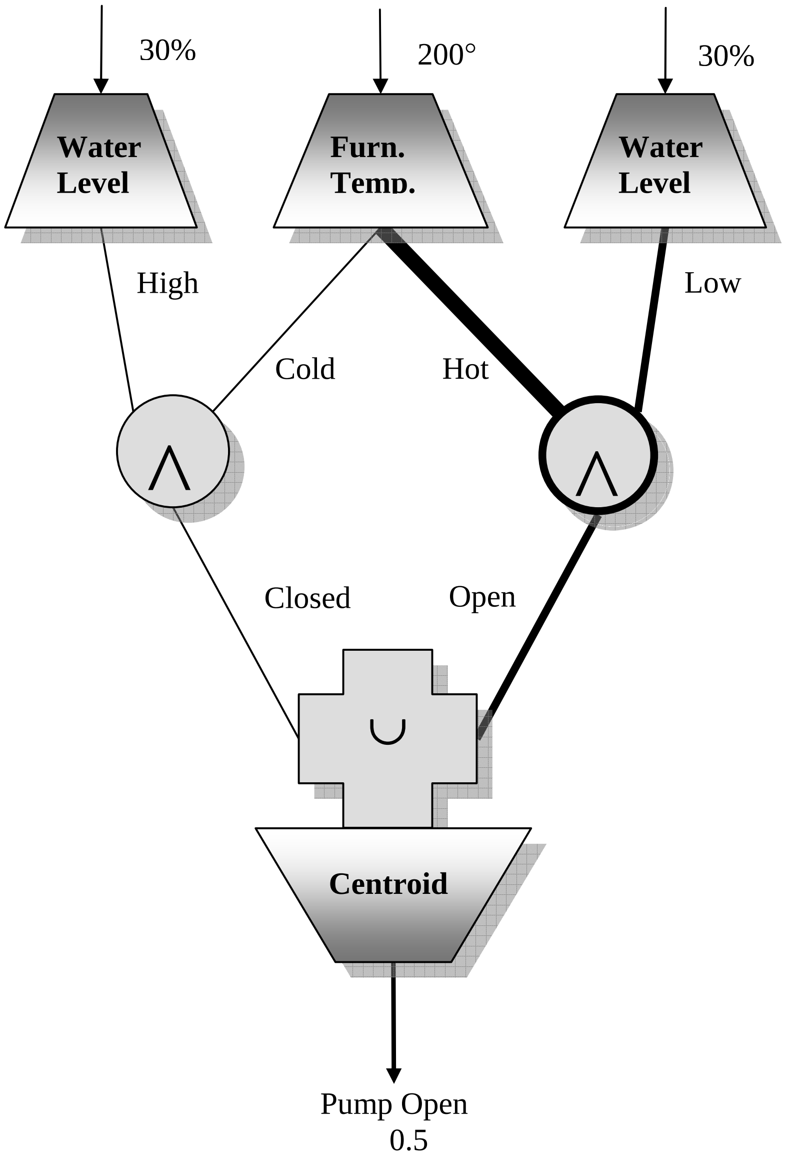


**Fig. 12** Example network diagram on the surface of the cones. The size and intensity of lines illustrate the popout effect.

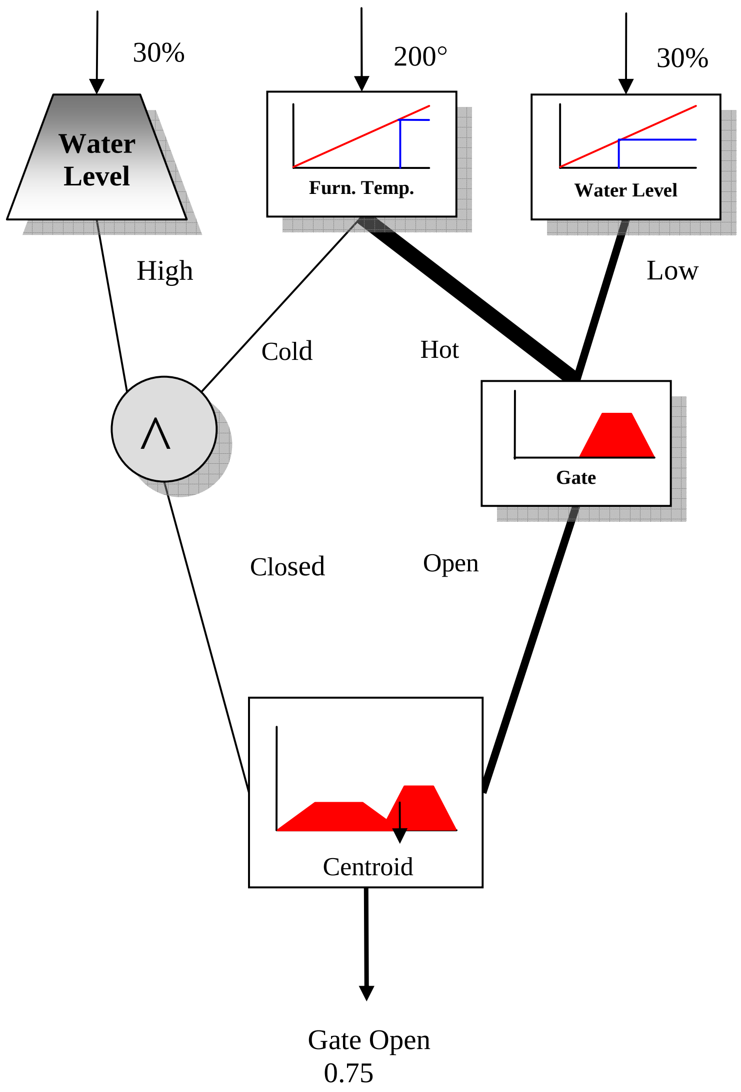



**Fig. 13** Example network diagram on the surface of the cones. The data values in the system are illustrated within the icons for a part of the network.

## 8. Conclusion and Future Work

We have presented a holistic approach to constructing a visualization system for fuzzy systems based on a requirement analysis. We discussed the motivations behind the use of a multi-agent approach to develop a framework for visualization of fuzzy systems which is user- and task-oriented. This framework is based on our design of fundamental ontologies which underlie the structure and requirements of these systems. We have also presented the structure and activities of these agent classes and how they would be implemented.

A case study has been investigated to provide visualization support for a fuzzy model which had been developed to predict electricity spot prices [3]. A critical review of current methods for visualizing fuzzy data and fuzzy relationships was also provided. In addition, we demonstrated how two of these methods can be extended and improved by judicious choice of visual features and their integration. In particular, we introduce 3D parallel coordinate approach for displaying fuzzy rules, and a truncated double cone structure as a new visualization representation for fuzzy rules propagation. The double cone representation has potential for further development to cater for layers of fuzzy rules using both horizontal and concentric layers of the double cone. We will continue to explore the usefulness of such schemes in depth. On-going work also includes the implementation of agent classes to extend fully their capabilities and the evaluation of this framework further using case studies of fuzzy systems in different application domains.

## References


1. Berkan, R.C. and Trubatch, S.L.: Fuzzy System Design Principles, IEEE Press, NY (1997).
2. Berthold, M., and Holve, R.: Visualizing high dimensional fuzzy rules, in Proceedings of Fuzzy Information Processing Society: NAFIPS. 19th International Conference of the North American, Dept. of Electr. Eng. & Comput. Sci., California Univ., Berkeley, CA, USA (2000) 64-68.
3. Brown, R. and Pham, B., Visualisation of Fuzzy Decision Support Information: A Case Study, IEEE International Conference on Fuzzy Systems, St Louis, USA, May (2003).
4. Cox, Z., Dickerson, J.A., and Cook, D.: Visualizing Membership in Multiple Clusters after Fuzzy C-means Clustering, in Proceedings of Visual Data Exploration and Analysis VIII (2001) 60-68.
5. Dickerson, J.A., Cox, Z., Wurtele, E.S., and Fulmer, A. W.: Creating metabolic and regulatory network models using fuzzy cognitive maps, in Proceedings of IFSA World Congress and 20th NAFIPS International Conference, Joint 9th, Dept. of Electr. Eng, Iowa State Univ., Ames, IA, USA, 4 (2001) 2171-2176.
6. Farwowski, W., and Mita, A. (Eds.) Applications of fuzzy set theory in human factors (1986).
7. Fujiwara, Y., Shirashi, M., Nakagawa, D., and Okada, S.: Visualization of the Rule-based Program by a 3D Flowchart, in Proceedings of 6th International Conference on Fuzzy Theory and Technology (JCIS), NC, USA, (1998) 250-254.
8. Gershon, N.: Visualization of an imperfect world, Computer Graphics and Applications, IEEE, 18, (1998) 43-45.
9. Gershon, N. D.: Visualization of fuzzy data using generalized animation, Visualization '92, Proceedings, Mitre Corp., McLean, VA, USA, (1992) 268-273.
10. Goodchild, M.F., Montello, D.R., Fohl, P., and Gottsegen, J.: Fuzzy spatial queries in digital spatial data libraries, in Fuzzy Systems Proceedings, IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on, Nat. Center for Geogr. Inf. & Anal., California Univ., Santa Barbara, CA, USA, 1, (1998) 205-210.
11. Hall, L., and Berthold, M.: Fuzzy Parallel Coordinates, in Fuzzy Information Processing Society, NAFIPS. 19th International Conference of the North American, Atlanta, GA, USA, (2000) 74 -78
12. Jiang, B.: Visualisation of Fuzzy Boundaries of Geographic Objects, Cartography: Journal of Mapping Sciences Institute, Australia, 27, (1998) 31-36.
13. Keller, P., and Keller, M.: Visual Cues. Piscataway, USA: IEEE Press (1993).
14. Kosara, R., Miksch, S., and Hauser, H.: Focus+Context Taken Literally, IEEE Computer Graphics and Applications, 22, (2002) 22-29.
15. Lowe, A., Jones, R. and Harison, M.: The graphical Presentation of Decision Support Information in an Intelligent Anaesthesia, Artificial Intelligence in Medicine, 22, (2001) 173-191.
16. Nurnberger, A., Klose, A., and Kruse, R.: Discussing cluster shapes of fuzzy classifiers, in Fuzzy Information Processing Society, 1999. NAFIPS. 18th International Conference of the North American, Fac. of Comput. Sci., Magdeburg Univ., Germany, (1999) 546-550.
17. Nurnberger, A., Klose, A., and Kruse, R.: Analyzing borders between partially contradicting fuzzy classification rules, in Fuzzy Information Processing Society, NAFIPS. 19th International Conference of the North American, Fac. of Comput. Sci., Magdeburg Univ., Germany, (2000) 59-63.

18.Pham B., and Brown, R. Analysis of Visualisation Requirements for Fuzzy Systems, Proc. GRAPHITE 2003 Conference, (International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia) Melbourne , (2003) 181-187.
19. Pham B., and Brown, R.: Multi-agent Approach for Visualisation of Fuzzy Systems, International Conf. Computational Science ICCS2003, Melbourne, Australia and St. Petersburg, Russia, (2003) 995-1004.
20.Robertson, G.G., Mackinlay, J.D., and Card S.K. Cone Trees: Animated 3D Visualization of Hierarchical Information, Proc. CHI, (1991) 189-193.
21.Thomas, A.: Contouring Algorithms for Visualisation and Shape Modelling Systems, in Visualisation and Modelling, R. Earnshaw, J. Vince, and R. Jones, Eds. San Diego, USA: Academic Press, (1977) 99-175.
22.Tufte, E.: The Visual Display of Quantitative Information. Cheshire, USA: Graphics Press (1983).
23.Wandell, B.: Foundations of Human Vision, 1st ed. Sunderland, USA: Sinauer (1995).
24.Zadeh, L.A.: Toward a Theory of Fuzzy Information Granulation and its Centrality in Human Reasoning and Fuzzy Logic, Fuzzy Sets and Systems, 90, 2, (1997) 111-127.
25.Zenik, L., and Pham. B.: Fuzzy Models in Evaluation of Information Uncertainty in Engineering and Technology Applications, Proc. the 10th IEEE International Conference on Fuzzy, Australia (2001) 972-975.