Jurnal Teknologi Maklumat & Multimedia **9**(2010): 1 - 13

Autonomous Virtual Agent Navigation in Virtual Environments Using Dempster Shafer Approach and Fuzzy Logic

JAFREEZAL JAAFAR AND ERIC MCKENZIE

ABSTRAK

This paper presents a solution for behavioural animation of autonomous virtual agent navigation in virtual environments. We focus on using Dempster-Shafer's Theory of Evidence in developing visual sensor for virtual agent. The role of the visual sensor is to capture the information about the virtual environment or to identify which part of an obstacle can be seen from the position of the virtual agent. This information is required for virtual agent to coordinate navigation in virtual environment. The virtual agent uses fuzzy controller as a navigation system and fuzzy alpha-level for the action selection method. The testing was divided into two parts namely navigating in complex environment using different degrees of uncertainty and measuring the effectiveness of proposed action selection method to coordinate the behaviours by comparing with Fuzzy Behaviour Fusion (FBF) method. The aim of the testing was to evaluate the performance in terms of robustness and quality of path generated by the virtual agent. The result clearly demonstrates that the path produced is reasonably smooth even though there is some sharp turn and not diverted too far from the potential shortest path. This indicates the strength of our method, where more reliable and accurate paths produced during navigation task.

Keywords: Agent, Navigation, Dempster Shafer, Fuzzy Logic.

ABSTRAK

Artikel ini menerangkan satu penyelesaian untuk animasi tingkah laku dalam pengemudian perlakuan animasi agen maya secara autonomi dalam persekitaran maya. Kami memfokus kepada penggunaan teori pembuktian Dempter-Shafer's dalam membangunkan pengesan maya untuk agen maya. Fungsi pengesan visual adalah untuk mendapatkan maklumat tentang persekitaran visual atau, untuk mengenal pasti bahagian mana halangan boleh dilihat daripada kedudukan agen visual tersebut. Maklumat ini diperlukan untuk mengkoordinasikan pengemudian agen maya dalam persekitaran maya. Agen maya menggunakan kawalan kabur sebagai sistem pengemudian dan aras alfa kabur sebagai teknik pemilihan tindakan seterusnya (membuat keputusan). Pengujian telah dibahagikan kepada dua bahagian iaitu pengemudian di persekitaran kompleks menggunakan darjah kekaburan yang berbeza dan pengujian kedua ialah untuk mengukur keberkesanan teknik membuat keputusan untuk memilih tindakan seterusnya dalam mengkoordinasi perlakuan pengemudian serta membandingkan dengan kaedah Gabungan Tingkah laku Kaburn. Tujuan pengujian adalah untuk menilai prestasi berdasarkan keteguhan and kualiti lintasan yang dijana oleh agen maya. Keputusan pengujian telah menunjukkan lintasan yang dijana boleh dikatakan lancar walaupun terdapat sedikit lencongan tajam, namun masih tidak tersasar jauh dari potensi lintasan terpendek. Ini menunjukkan kekuatan kaedah yang digunakan yang mana telah menjana lintasan yang lebih boleh dipercayai dan tepat semasa pengemudian.

Kata kunci: Agen, Pengemudian, Dempster Shafer, Fuzzy Logic.

INTRODUCTION

Navigation is the process where people control their movement using environment cues and artificial aids such as maps so that they can achieve their goal without getting lost (Darken and Sibert 1993). Autonomous virtual agent navigation in a virtual environment can be described as the ability of a virtual agent to move purposefully without user intervention. The navigation task may be decomposed into three sub-tasks: mapping and modelling the environment; path planning and selection; path following and collision avoidance (Wan and Tang 2003). Virtual agent navigation can occur in known and unknown environments. For a known environment, the virtual agent will have knowledge about the environment and can generate the navigation path. The methods used are based on optimization and computational intelligence. In contrast, in an unknown environment in which the virtual agent does not have any knowledge about the environment, the navigation path is generated according to user specifications and the virtual agent cannot be prepared ahead of time (Li and Lie et al. 1999).

BACKGROUND

The basic problem of navigation is moving from one place to another by the coordination of planning, sensing and control. The challenge is generating an optimal traversing sequence through the user-specified locations of interests and computation of a collision free path. (Li and Lien et al. 1999) had shown an example of a path traversing through all user-specified locations. In order to navigate in an unknown environment, a virtual agent needs to deal with the environment in a timely manner.

Approaches such as discrete grid based (Bandi and Thalmann 2000), central path computation (Chaudhuri, Khandekar, et al. 2004) and roadmap with tactical information approaches (Rook and Kamphuis 2005) have been used for collisions free path planning. For example (Stilman and Kuffner 2004) studied navigation among static and movable obstacles. The planner takes advantage of the navigational structure through state-space decomposition and a heuristic search. The planning complexity is reduced to the difficulty of the specific navigation task, rather than the dimensionality of the multi-object domain.

Inspired by studies in human behaviour, Lamarche and Donikian (2004) proposed a general model to simulate the navigation process inside indoor and outdoor environments. Techniques such as set hierarchy, regular graph, artificial potential field and corner graph have been used but are only suitable for 2D environments. One of the reasons is those algorithm require high computational resource in 3D environments. For a 3D environment, navigation mesh and waypoint graph techniques are very popular. A navigation mesh technique is a representation that covers the walkable surface of the world with convex polygons (Tozour 2003). Waypoint is a set of points in the 3D environment with reachability links between them, where we can place a waypoint at any point in 3D space. The disadvantages of these two techniques are large memory usage, and they require a powerful processor. Even though some of these techniques have been used in computer games, it is still not clear that these approaches have been used in autonomous navigation in virtual environments (Salomon and Garber et al. 2003).

Artificial intelligence techniques, for example neural networks (Lozano & Molina 2002), genetic algorithms (Velagic, Lacevic et al. 2006) and reinforcement learning (Ho-Sub, So-Joeng et al. 2000) have been used. Wang (2002) presented a multi-agent based evolutionary artificial neural network (ANN) for general navigation. The virtual creature explores unknown environments as far as possible with obstacle avoidance. Through constant interaction with the environment, the virtual agent systems co-decide and consult with each other for the move decision. Lozano and Molina (2002) have integrated attention and navigation skills in a 3D virtual agent. They divided their neural model into two main phases. First of all, the environment categorization phase, online pattern recognition and categorization of the virtual agent current input sensor data is carried out by an adaptive resonance driven self organizing neural network. Then, the model must learn how and when to map the current short term memory state into navigation and the attention of the virtual agent. However the majority of 3D virtual agents focus on low cost global techniques to solve navigation problems and attention is less frequently considered in virtual worlds.

The reactive virtual agent (Piaggio, Sgorbissa, et al. 1997) is capable of carrying out autonomous navigation. The virtual agent extends the artificial potential field approach, used for trajectory formation, to environment exploration and symbolic feature detection. The virtual agent's capabilities range from obstacle avoidance to maze navigation, carried out autonomously or under the supervision of higher cognitive levels. Other methods by Salomon et al. (2003) have been used in a known environment. On the other hand, in an unknown environment, methods such as sensor based control in a study by Wan and Tang (2003) use Adaptive Dynamic Points of Visibility (ADPV) for moving virtual agents in dynamical unconfigured environments.

ARCHITECTURE

The navigation system can be divided into three main components, namely are the fuzzy navigator, virtual agent and the environment, as in Figure 1. The main component of the navigation system is the virtual agent itself. The virtual agent should be able to make its own decisions; it does not require any information about the virtual environment; and does not require any training or learning before the navigation task.

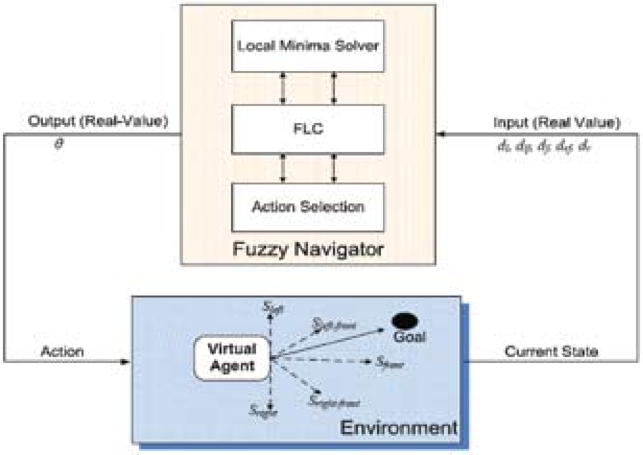

FIGURE 1. Navigation system

The fuzzy navigator is the main engine for the virtual agent. It comprises of three main components:

- Fuzzy Logic Controller (FLC) - using a behaviour-based architecture which comprises of Path-Planning Behaviour (PP), Goal-Seeking Behaviour (GS) and Obstacle-Avoidance Behaviour (OA).
- Local Minima Solver (LMS) - responsible for helping the virtual agent escape from dead-ends.
- Fuzzy Action Selection (Decision Making) Mechanism (Fuzzy-ASM) - to make the final decision in • selecting the possible action required by the virtual agent to reach the goal.

The fuzzy navigator receives input from the visual sensor and produces the final action needed to be executed by the virtual agent. Each component in the fuzzy navigator is integrated and works independently.

A. *Visual Sensor*

The main information between environment and virtual agent is retrieved using a visual sensor. This visual sensor differs from vision systems in robotics, since all information about pattern recognition and noisy images can be ignored (Kuffner 1999). The visual sensor captures the information about the virtual environment or identifies which part of an obstacle can be seen from the position of the virtual agent as in Figure 2. Also, the visual sensor only identifies whether a square (cell) in the vision range is occupied by an obstacle or not. The assumption made is that all objects are opaque.

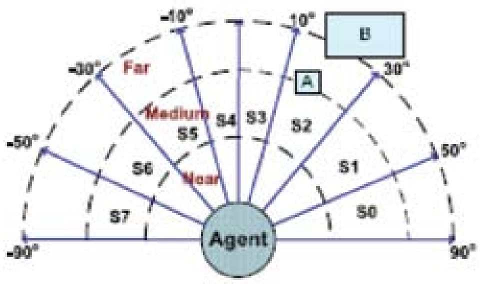

FIGURE 2. Example of Vision Field and Sensor's Region based on location

The visual sensor field of the vision range is 180° . The vision field is divided into eight main sectors which are represented as $S0, S1, S2, S3, S4, S5, S6$ and $S7$. Hence, there is a probability that the cells located in the proximity may be occupied. Cells well inside the vision field sector are likely to be empty. An occupancy grid is essentially a data structure that indicates the certainty that a specific part of space is occupied by an obstacle. It is a representation of an environment as a two-dimensional array. Each element of the array corresponds to a specific square on the surface of the actual world, and its value shows the certainty that there is some obstacle there.

The visual sensor in Wang and McKenzie (1999) has been modified by using Dempster-Shafer evidence theory (Shafer 1976). Whenever the virtual agent moves, it catches new information about the environment and updates the map. To facilitate building an occupancy map (Velagic, Lacevic et al. 2006) of the environment, a grid representing the whole space needs to be constructed. Every discrete region of the map (each cell) may be in two states, *Empty* is E and *Full* is F. Then, a frame of discernment, κ, is defined by the set $\kappa = [E, F]$, where E and F represent the possibility that a cell is *Empty* or *Full*. The advantage of this technique is that the building of occupancy maps is well suited to path planning and obstacle avoidance (Young-Chul, Sung-Bae et al.).

Review of Young-Chul, Sung-Bae, et al. (2002) Use of Dempster-Shafer's Theory of Evidence
A basic probability assignment is a function m: $\kappa = [0,1]$ where γ is a set of all subsets of κ. In our case, $\gamma = 2^{\kappa} = \{ø, \{E\}, \{F\}, \{E, F\}\}$. The state of each cell is described by assigning a basic probability number to each label γ. For each cell (i, f) in the grid, it is required that:

$$m_{i,j}(ø) = 0 \quad (1)$$

$$\sum\left[\left\{m\right]_{i,j}\right\}(A) = m_{i,j}(\phi) + m_{i,j}\{E\} + m_{i,j}\{F\} + m_{i,j}\{E,F\} \quad (2)$$

Every cell in the environment is initialized as follows:

$$m_{i,j}\{E\} = m_{i,j}\{F\} = 0 \quad (3)$$

$$m_{i,j} = \{E, F\} = 1 \tag{4}$$

Then, the virtual agent moves and scans the environment. If n cells exist in the vision field sector, the basic probability assignment for the vision field sector is as follows:

$$m_{i,j}(F) = \frac{1}{n}, m_{i,j}(E) = 0, \forall \text{ cells}(i,j) \in \text{sector} \tag{5}$$

$$m_{i,j}(F) = 0, m_{i,j}(E) = 0, \forall \text{ cells}(i,j) \notin \text{sector} \tag{6}$$

By adding subscripts S and M to basic probability masses m, we can describe the basic probability assignment of the sensor as equations (8) and (9):

$$\kappa = 1 - m_M(E)m_S(F) - m_M(F)m_S(E) \tag{7}$$

$$m_M \oplus m_S(E) = \frac{m_M(E)m_S(E) + m_M(E)m_S(E,F) + m_M(E,F)m_S(E)}{\kappa} \tag{8}$$

$$m_M \oplus m_S(F) = \frac{m_M(F)m_S(F) + m_M(F)m_S(E,F) + m_M(E,F)m_S(F)}{\kappa} \tag{9}$$

However, the number of states can be reduced to two ($m_{i,j}(E)$, $m_{i,j}(F)$) assuming that $m_{i,j}(\emptyset) = 0$ and applying equation (2). The state (0,0) means total ignorance, and so $m_{i,j}(E,F) = 1$. When the virtual agent is sure about cell occupancy, $m_{i,j}(F) = 1$, the other labels are made equal to zero. On the other hand, $m_{i,j}(E) = 1$ when the virtual agent is sure that the cell is empty.

The input value θ of the virtual agent, which is a real number normalized in the interval [0,1], then results from a weighted sum of all the points in the visual field.

$$\theta = \sum_x \left[2^{-2d(x)}\right] u(x)) \tag{10}$$

summed over all $d(x)$ in visual field where is the distance of a point x from the current position of the virtual agent, and $u(x)$ indicates the availability of the point x. Since the visual sensor is related to availability of spaces in the visual field, it is independent of specific environments and objects. The result is that the occupancy of cells is increased. This process will be carried out until the virtual agent reaches the goal.

B. Fuzzy Controller

A Fuzzy Associative Memory (FAM) is used as a process of encoding and mapping the input fuzzy sets to the output fuzzy set (Kosko 1992). The fuzzy controller is based on our proposed method (Jaafar and McKenzie 2006 2007). Consider a set of fuzzy rules, $R = \{R_1, R_2, \ldots, R_i, \ldots, R_k\}$ where R_m is the m^{th} rule of the fuzzy controller. The rule R_m is given as follows:

$$\begin{aligned}&IF\ X_1\ is\ A_1^m\ AND\ X_2\ is\ A_2^m\ AND\\ &\ldots AND\ X_n\ is\ A_n^m\ THEN\ Z\ is\ C_n^m\end{aligned} \tag{11}$$

The following fuzzy relation will implement R_i:

$$R_m(X_1, X_2, \ldots, X_n, Z) = \begin{bmatrix} A_1^m(X_1) \wedge A_2^m(X_2) \wedge \\ \ldots \wedge A_n^m(X_n) \end{bmatrix} \rightarrow C_n^m(Z) \tag{12}$$

Where $X_1, X_2, \ldots, X_n$ are input variables which are the sensor data of the virtual agent, $A_1^m, A_2^m, \ldots, A_n^m$ are the input fuzzy sets, C_n^m is the output fuzzy set, Z is the output variable, n is the dimension of the input vector and m is the number of fuzzy sets.

The weighted sum C for each individual membership can be defined by using *minmax* aggregation (Ross 2004) operators as given below:

$$\begin{aligned} C &= \sum_{i=1}^{k} U_m C'_m \\ &= \sum_{i=1}^{k} U\left(\left[A_1^m(X_1) \wedge A_2^m(X_2) \wedge \ldots A_n^m(X_n)\right] \Rightarrow C_m(Z)\right) \end{aligned} \tag{13}$$

The non-negative weight U_i summaries the strength of the m^{th} FAM entry and $n \times m$ is the number of rules in the system. In order to relate the n^{th} fuzzy set of the m^{th} fuzzy rule, the fuzzy implication model using the Mamdani min operator (L.-X. Wang 1997) interprets the logical rules for rule firing. We obtain the final defuzzification response for a κ output membership function $U_c(Z)$ is defined as:

$$U_c(Z) = \max_{\substack{m=1 \\ X=U}}^{k} \left[\min_{m=1}^{n} \begin{bmatrix} \mu_{A_n^m}(X_n), \\ \mu_{R_m}(X_1, X_2, \ldots, X_n, Z) \end{bmatrix} \right] \tag{14}$$

Equations (12) and (14) are used to derive the FAM model and the output fuzzy system respectively.

C. Action Selection Method

This work is inspired by the ranking method of Huang (1989), Mabuchi (1988) and Yuan (1991) and uses α-level and fuzzy subtraction operations to calculate the area of a new fuzzy number, which is produced by the comparison of two fuzzy numbers. If there are m fuzzy numbers, then $m(m-1)/2$ pairs of fuzzy numbers must be compared to determine overall rank. Our proposed method will reduce the redundancy of calculating $m(m-1)/2$ pairwise comparisons to m pairwise comparisons by the fuzzy subtraction operation.

In general, when comparing m different fuzzy numbers produced by each behaviour (OA, GS, PP) the height and common maximizing and minimizing barriers are used. Let $\mu_{\tilde{X}}(x)$ be the membership function of a fuzzy number, $\tilde{X}$ (behaviour output), defined on R. Unlike convexity, n assumptions about the normality of $\mu_{\tilde{X}}(x)$ are made.

Based on (Choobineh and Li 1993), the loci of the left or right spreads and the maximum and minimum barriers of the α-cut of the fuzzy number, $\tilde{X}$, are $\mu_{\tilde{X}_\alpha}^{L}(x)$ and $\mu_{\tilde{X}_\alpha}^{R}(x), 0 \le \alpha \le h_{\tilde{X}}$, respectively, where $h_{\tilde{X}}$ is the height. If $\tilde{X}_n$ is denumerable or connected, then:

$$\begin{aligned} &\mu_{\tilde{X}_\alpha}^{L}(x) = \min\left\{x \middle| x \in \tilde{X}_\alpha\right\}, 0 \le \alpha \le h_{\tilde{X}}, \text{ and} \\ &\mu_{\tilde{X}_\alpha}^{R}(x) = \max\left\{x \middle| x \in \tilde{X}_\alpha\right\}, 0 \le \alpha \le h_{\tilde{X}}. \end{aligned} \tag{15}$$

The height, maximizing and minimizing barriers are set to:

$$\begin{aligned} h_{\tilde{X}}(x) &= \max\left\{\mu_{\tilde{X}_i} \middle| i = 1, 2, \ldots, m\right\}, \\ c &= \min_{\alpha}\left\{\mu_{\tilde{X}_{i\alpha}}^{L}(x) \middle| i = 1, 2, \ldots, m; 0 \le \alpha \le h_{\tilde{X}}\right\}, \\ d &= \max_{\alpha}\left\{\mu_{\tilde{X}_{i\alpha}}^{R}(x) \middle| i = 1, 2, \ldots, m; 0 \le \alpha \le h_{\tilde{X}}\right\}. \end{aligned} \tag{16}$$

Based on Huang (1989) and equation (16), that is, $h_{\tilde{X}}(x)$ is the maximum value of the height of all m fuzzy numbers.

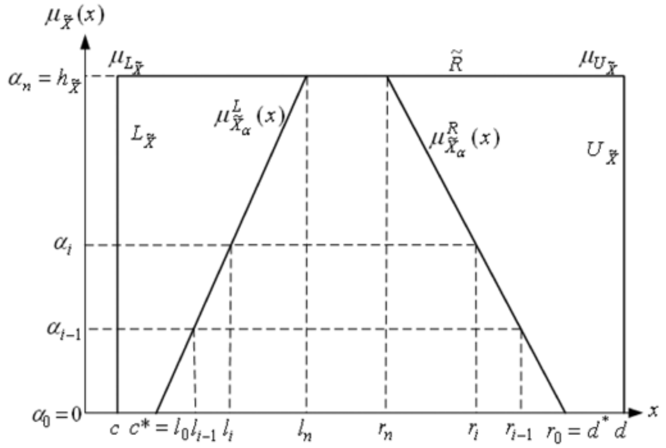

FIGURE 3. Trapezoidal fuzzy number

The variables c and d are at the minimum value of the left spread and the minimum right spread of all fuzzy numbers, respectively. To simplify the fuzzy subtraction between the fuzzy number $\tilde{X}$ and referential rectangle $\tilde{R}$, at α_i level, interval subtraction is used:

$$\begin{aligned}\tilde{X}_{\alpha i}\langle - \rangle \tilde{R} &= [l_i, r_i][-][c, d] \\ &= [l_i - d, r_i - c], \quad i = 0,1,2,\dots,n\end{aligned} \tag{17}$$

then, the behaviour weight, W of equation (17) becomes:

$$W\left(\tilde{X}_j, \tilde{R}\right) = \frac{\sum\limits_{i=0}\left(r_i - c\right)}{\sum\limits_{i=0}^{n}\left(r_i - c\right) - \sum\limits_{i=1}^{n}\left(l_i - 1\right)}, \text{ and } n \to \infty \tag{18}$$

where n is the number of the α – level and as n approaches to ∞, the summation becomes the area measurement. In equation (18), $\sum\limits_{i=0}^{n}\left(r_i - c\right)$ is a positive value and $\sum\limits_{i=0}^{n}\left(l_i - 1\right)$ is a negative value. Here, the denominator represents the total area as n approaches ∞. In addition, if all of the aggregated fuzzy numbers are normal and within the unit interval, then $h_{\tilde{X}} = 1$, c = 0, d = 1, and equation (18) becomes:

In our case, the behaviour weight value W from equation (14) was used. For every W, we used the minimax (maximin) criterion, which selects the lowest value from each behaviour as δ_1; and then selects the highest value from each behaviour as δ_2. The index of optimism (Chen and Yu 1997), σ, was used to represent the level of uncertainty of the virtual environment. When selecting one particular action from a range of possible actions, the selection was based on the Hurwicz criterion (Arnold, Grossl et al. 2002) which is defined as:

$$\begin{aligned}\eta &= \sigma \cdot \left(\min_{i=1}^{m} W_{ijo}\right) + (1-\sigma)\cdot\left(\max_{i=1}^{m} W_{ijo}\right) \\ &= \max_{j\in J}\left(\sigma\cdot\left(\min_{i=1}^{m} W_{ij}\right) + (1-\sigma)\cdot\left(\max_{i=1}^{m} W_{ij}\right)\right) \\ \text{where } \eta &= \begin{cases}\sigma = 0 \to \text{Maximin criterion} \\ 0 < \sigma < 1 \to \text{Compromise opinion} \\ \sigma = 1 \to \text{Maximax criterion}\end{cases}\end{aligned} \tag{20}$$

Based on the above discussion the following algorithm was used. Let $\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_j, \ldots, \tilde{X}_m$ be m arbitrary bounded fuzzy numbers produced by each behaviour.

Step 1: Set the height $hg(x)$, common maximizing barrier d and minimizing barrier c for referential rectangle $\tilde{R}$.

Step 2: Determine the subtracted interval numbers $[l_i - d, r_i - c]$, $i = 0,1,2,\ldots,n$ by calculating the n α – level s for each fuzzy number $\tilde{X}_j <-> \tilde{R}$, $j = 0,1,2,\ldots,m$.

Step 3: Determine the behaviour weight, W for each $\tilde{X}_j$, by equation (19).

Step 4: Repeat Steps 2 and 3, for every j, $j = 0,1,2,\ldots,m$ and the m behaviour weights for m fuzzy numbers are obtained.

Step 5: For every W, use the minimax (maximin) criterion, which selects the lowest value from each behaviour as δ_1 and selects the highest value from each behaviour as δ_2.

Step 6: Determine the index of optimism σ. The final action is selected based on the Hurwicz criterion using equation (20).

RESULTS

Experiments were also conducted to observe the effect of using different degrees of optimism, sigma, by the virtual agent to navigate in complex environments. Figure 4 shows the result of the experiment conducted in a cluttered environment using different degrees of optimism, σ, which are (a), $\sigma = 0.9$ and (b) $\sigma = 0.4$. The environments contain different sizes of obstacle and narrow passages. The virtual agent in Figure 4(a) has produced a shorter path compared to the virtual agent in Figure 4(b). However the number of steps is higher compared to Figure 4(b). The main reason is that the virtual agent is required to go through a narrow passage in order to produce the shortest path. In Figure 4(b), the virtual agent has made a sharp turn and high number of time steps at this point. As a result the virtual agent take a big turn to the wider passage before turning and reaching the goal. Time steps at the rest of the path are consistent since there is no complex obstacle to avoid. The results show that the decision process by the virtual agent is affected by the degree of optimism. Using a higher value of \sigma makes the virtual agent enter the narrow passage compare to a low value of \sigma which makes the agent prefer to select the wider passage. However the number of decisions and steps might vary depending on the complexity of the environment.

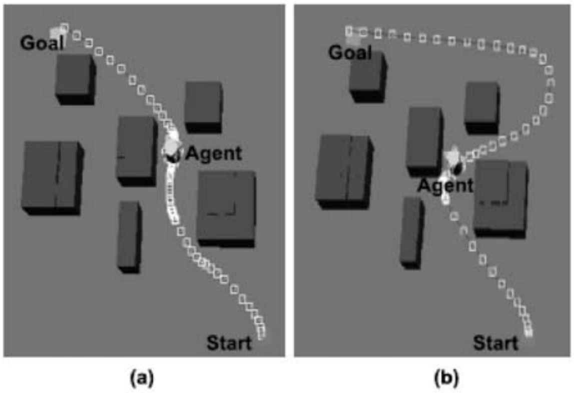

FIGURE 4. Different Degrees of Optimism (a) = 0.9 and (b) = 0.4

Further experiments with complex environments have been conducted. The environments contain a combination of maze and cluttered obstacles and three random goals have been selected. The degree of optimism, $\sigma = 0.5$, was used for the first trial. Unfortunately this value did not give a very promising result as in Figure 5. The virtual agent had successfully reached the goal, but paths produced are long with many sharp turns and a high number of time steps.

Based on result in Figure 4, using a higher value of σ will give a better result. A new value of σ = 0.8 has been selected. Figure 6(a) and (c) produce smooth and short paths compared to the results in Figure 5(a) and (c). In Figure 6(b), the virtual agent follows a similar path compared to Figure Figure 5(b) but with a small number of sharp turns. From the figures, we also notice that the virtual agent does not take the narrow path at X. One probable is that the passage is too narrow and might require a higher value of σ. However having a higher value of σ, the virtual agent might follow a longer and unsafe path.

Also in Figure 5(b) and Figure 6(b), notice that the virtual agent does not produce the shortest path. The virtual agent moves forward to the goal even though there are a walls and a dead-end. Then the virtual agent makes a left turn to escape from dead-end and follow the wall toward the goal. The virtual agent tried to reach the goal by moving straight ahead towards the goal by having a high value for Goal-Seeking behaviour. The virtual agent starts to switch to Path-Planning behaviour and Obstacle-Avoidance behaviour when it encounters an obstacle and needs to make a turn to reach the goal. This shows that the virtual agent has imitated how a human might make decisions during a navigation task in an unknown environment by making a good assumption that the path to the goal is ahead of them even though they cannot see the goal.

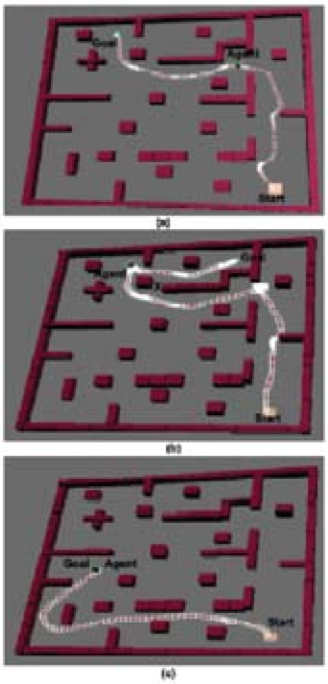

FIGURE 5. Navigating in combination of cluttered and maze environment (0.5)

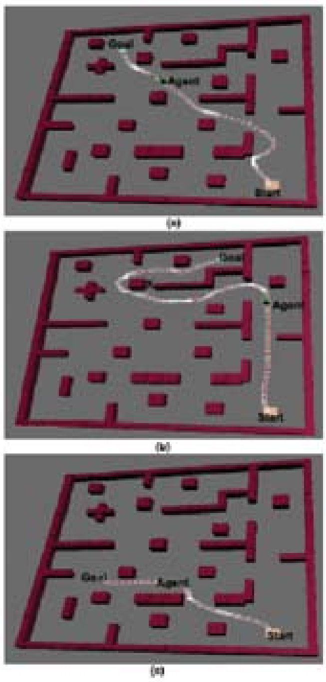

FIGURE 6. Navigating in combination of cluttered and maze environment (σ = 0.8)

Four test cases have been used which are the virtual agent being moved from the same start point to different target points as in Figure 7 to Figure 10 (Test Case 1, 2, 3 and 4). Figure 10 shows the example of the path produced by the virtual agent in Test Case 4.

Figure 10(a) is the path produced by Cang's Method and Figure 10(b) shows the path produced by our Fuzzy-ASM. The path produced by the Fuzzy-ASM is shorter than Cang's method even though the smoothness of the path is similar.

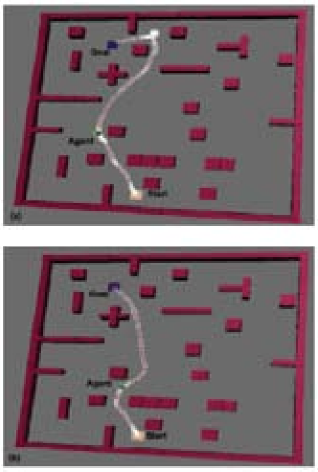

FIGURE 7. Test Case 1

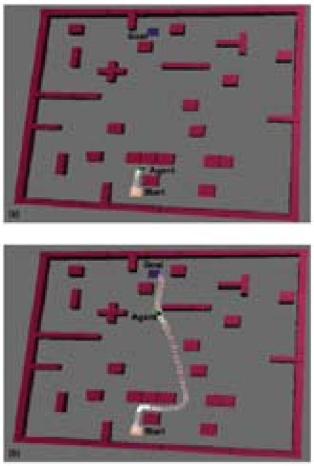

FIGURE 8. Test Case 2

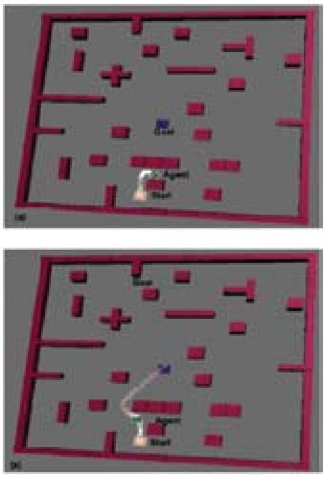

FIGURE 9. Test Case 3

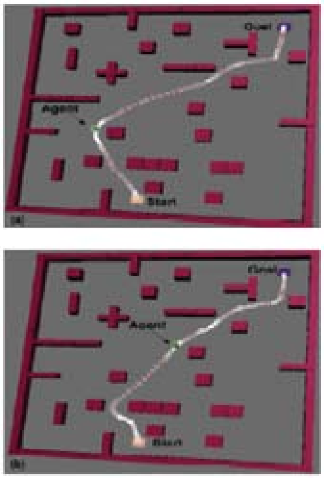

FIGURE 10. Test Case 4

Further testing has also been conducted with nine different goal locations. Figure 11 shows the result of (a) Time (t_n), (b) Distance (d_t) and (c) Decisions (k) taken by the virtual agent for all nine goal locations. The results show that Fuzzy-ASM has taken less time and a shorter distance to complete the task. The average percentages of Δt_n and Δd_t are 16% and 17.4%, respectively. When we compare the number of decisions made by each method, Fuzzy-ASM has made fewer decisions. The average number of decisions is 8.04% less than Wang's method. Fewer decisions leads to a faster and more reliable decision making process.

Our tests also show that the success rate for the Fuzzy-ASM is higher than Wang's method, as shown in Figure 12. Success rate refers to the percentage of test runs (total of 25 runs) for each test where the virtual agent successfully reached the goal. In test 1 to test 4, the fuzzy ASM had a 100% success rate. Wang's method starts to decrease at test 2. The lowest success rate is 90% compared to Wang's method at 70%. This suggests that the Fuzzy-ASM is more reliable.

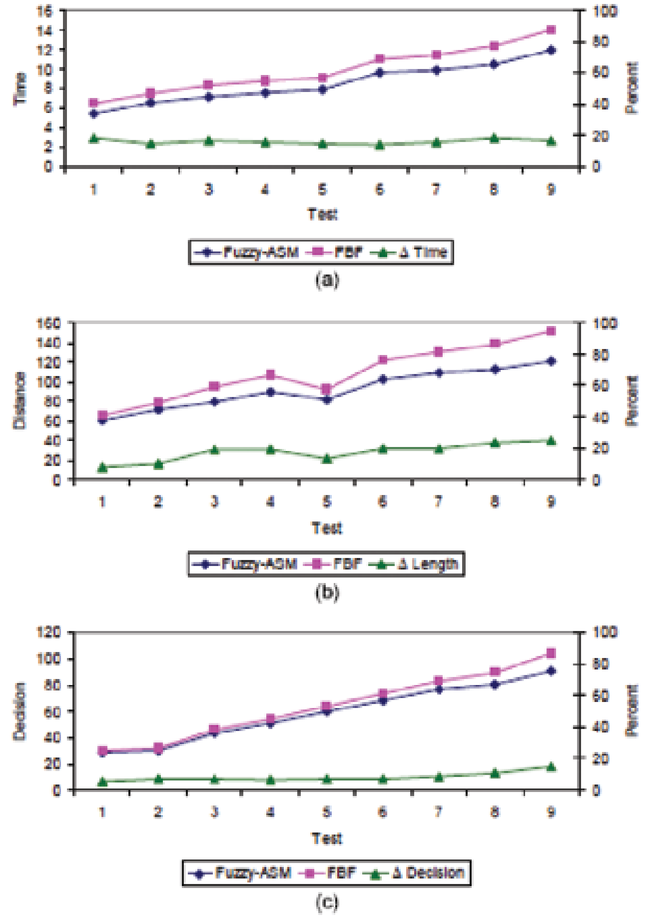

FIGURE 11. (a) Time, (b) Distance and (c) Decisions

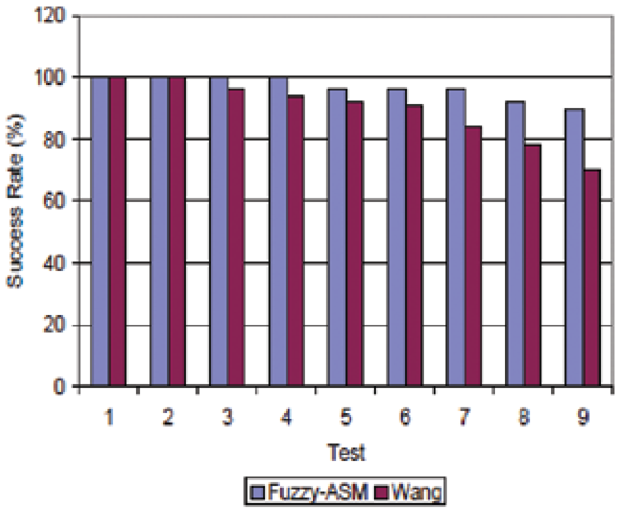

FIGURE 12. Test Success Rate

CONCLUSION

Our visual sensor had shown how information is captured in virtual environment and how it identifies which part of an obstacle can be seen from the position of the virtual agent. Also, the visual sensor only identifies whether a square (cell) in the vision range is occupied by an obstacle or not. This information is critical for virtual agent in coordinating its navigation task. The visual sensor is also easy to integrate with our fuzzy controller. The evaluation results showed that the virtual agent had deviated with minimum distance when avoiding the obstacles. The results also clearly demonstrated the mapping of inputs to output with a smooth path in a navigation task. This presents a natural way of dealing with a virtual environment without having to use complex mathematical model.

REFERENCES

Arnold, B. F., Grossl, I. et al. 2002. The minimax, the minimin, and the Hurwicz adjustment principle. *Theory and Decision, 52*(3), 233-240.

Bandi, S., and Thalmann, D. 2000. Path finding for human motion in virtual environments. *Computational Geometry: Theory and Applications,* 15(1-3): 103-127.

Cang, Y., and Danwei, W. (2000). *Novel Behavior Fusion method for the navigation of mobile robots*. Paper presented at the 2000 IEEE International Conference on Systems, Man and Cybernetics, Nashville, USA. 3526-3531

Chaudhuri, P., Khandekar, R., et al. 2004. *An efficient central path algorithm for virtual navigation*. Paper presented at the International Computer Graphics, 2004, Crete, Greece. 188-195

Chen, T. Y., and Yu, Y. T. 1997. *On the Criteria of Allocating Test Cases under Uncertainty.* Paper presented at the Fourth Asia-Pacific Software Engineering and International Computer Science Conference, Hong Kong. 405-413

Choobineh, F., and Li, H. 1993. An Index for Ordering Fuzzy Numbers. *Fuzzy Sets and Systems,* 54(3): 287-294.

Darken, R. P., and Sibert, J. L. (1993). *A toolset for navigation in virtual environments*. Paper presented at the Proceedings of the 6th annual ACM symposium on User interface software and technology, Atlanta, Georgia, United States. 157-165

Ho-Sub, S., So-Joeng, Y., et al. 2000. *Fuzzy reinforcement function for the intelligent agent to process vague goals*. Paper presented at the Annual Conference of the North American Fuzzy Information Processing Society - NAFIPS, Atlanta, GA, USA. 29 - 33

Huang, C. C. 1989. *A Study on The Fuzzy Ranking and Its Application on The Decision Support System*. Phd Thesis, Tamkang University, Taiwan.

Jaafar, J., and McKenzie, E. 2006. *Behaviour Coordination of Virtual Agent Navigation Using Fuzzy Logic*. Paper presented at the International Conference on Fuzzy Systems (Fuzz-IEEE 2006), Vancouver, Canada. 1139-1145

Jaafar, J., and Mckenzie, E. 2007. *A reactive architecture for autonomous agent navigation using fuzzy logic*. Paper presented at the Artificial Intelligence and Soft Computing 2007, Palma de Mallorca, Spain. 584-039

Kosko, B. 1992. *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. New Jersey: Prentice-Hall inc.

Kuffner, J. J. 1999. *Autonomous Agent for Real-Time Animation*. PhD Thesis, Stanford University, Stanford, USA.

Lamarche, F., and Donikian, S. 2004. *Crowd of Virtual Humans: a New Approach for Real Time Navigation in Complex and Structured Environment*. Paper presented at the Eurographics 2004, Grenoble, France. 509-518

Li, T.-Y., Lien, J.-M., et al. 1999. *Automatically generating virtual guided tours*. Paper presented at the '99 Computer Animation Conference, Geneva, Switz. 99-106

Lozano, M., and Molina, J. 2002. *A neural approach to an attentive navigation for 3D intelligent virtual agents*. Paper presented at the 2002 IEEE International Conference on Systems, Man and Cybernetics, Hammamet, Tunisia. 5-10

Mabuchi, S. 1988. An approach to the comparison of fuzzy subsets with an alpha-cut dependent index. *IEEE Transactions on Systems, Man and Cybernetics,* 18(2): 264-272.

Piaggio, M., Sgorbissa, A., et al. 1997. Autonomous robot navigation using a reactive agent. In M. Lenzerini (Ed.), *AI*IA 97: Advances in Artificial Intelligence* (pp. 96-105): Springer.

Rook, M., and Kamphuis, A. 2005. *Path Finding using Tactical Information*. Paper presented at the Eurographics/ACM SIGGRAPH Symposium on Computer Animation (2005), Los Angeles, USA. 1-2

Ross, T. J. (2004). *Fuzzy Logic with Engineering Application* (2nd ed.). West Sussex: John Wiley & Sons.

Salomon, B., Garber, M., et al. 2003. *Interactive navigation in complex environments using path planning*. Paper presented at the 2003 Symposium on Interactive 3D graphics, Monterey, California. 41-50

Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton, NJ: Princeton University Press.

Stilman, M., and Kuffner, J. J. 2004. *Navigation Among Movable Obstacles: Real-time reasoning in complex environments*. Paper presented at the 4th IEEE-RAS International Conference on Humanoid Robots, 2004, Santa Monica, USA. 322-341

Tozour, P. 2003. Search Space Representations. In S. Rabin (Ed.), *AI Game Programming Wisdom 2*. USA: Charles River Media Inc.

Velagic, J., Lacevic, B., et al. 2006. A 3-level autonomous mobile robot navigation system designed by using reasoning/search approaches. *Robotics and Autonomous Systems,* 54(12): 989-1004.

Wan, T., and Tang, T. (2003). *An intelligent vehicle model for 3D visual traffic simulation*. Paper presented at the International Conference on Visual Information Engineering (VIE 2003): 206-209

Wang, F. (2002). *Study of an Adaptive and Multifunctional Computational Behaviour Generation Model for Virtual Creature*. PhD Thesis, University of Edinburgh, Edinburgh, UK.

Wang, F., and McKenzie, E. 1999. *A multi-agent based evolutionary artificial neural network for general navigation in unknown environments*. Paper presented at the Third Annual Conference on Autonomous Agents, Seattle, United States. 154-159

Wang, L.-X. 1997. *A Course on Fuzzy System and Control*. London: Prentice-Hall International Inc.

Young-Chul, K., Sung-Bae, C., et al. 2002. *The Dempster-Shafer approach to map-building for an autonomous mobile robot with fuzzy controller.* Paper presented at the 2002 AFSS International Conference on Fuzzy Systems, Calcutta, India. 40-46

Yuan, Y. 1991. Criteria for evaluating fuzzy ranking methods. *Fuzzy Sets and Systems,* 43(2): 139-157.

Jafreezal Jaafar
Computer and Information Sciences Department
Universiti Teknologi PETRONAS
Malaysia
jafreez@petronas.com.my

Eric McKenzie
Institute of Perception, Action and Behaviour
School of Informatics
University of Edinburgh
ram@inf.ed.ac.uk