HANDLING SPATIAL VAGUENESS IN VIRTUAL AGENT CONTROL

Spyros Vosinakis, Nikos Pelekis, Yannis Theodoridis and Themis Panayiotopoulos
Department of Informatics, University of Piraeus, 80 Karaoli & Dimitriou str., Piraeus, Greece
{ spyrosv, npelekis, ytheod, themisp }@unipi.gr

Keywords: Fuzzy Logic, Computer Animation, Virtual Reality, Virtual Agents

Abstract: Virtual agents are an important element of virtual environments that enhance their believability and autonomy. Nowadays, there are a number of architectures and languages for designing virtual agents and scripting their control process, but the majority of them cannot cope with the vagueness that characterizes a designer's description of a location or direction using natural language, and, thus, fail to demonstrate complex spatial behavior in dynamic environments. We propose a framework for handling spatial vagueness in virtual agent control, which uses binary fuzzy relations to represent vague location descriptions and a fuzzy rule-based system for the agent control. We present a prototype implementation and a case study, in which the proposed framework is successfully used for a virtual agent's locomotion in a dynamic environment.

1 INTRODUCTION

Virtual Agents are autonomous entities in synthetic environments that aim to enhance the believability of virtual reality applications (Aylett and Luck, 2000). To satisfy this need for autonomy, they are equipped with mechanisms for monitoring changes in the environment (sensors), for executing actions upon it (effectors), and for taking decisions about their actions.

One problem in current agent control architectures, which is a consequence of their need to combine both an abstract representation and a concrete model of the virtual world, is *spatial vagueness*, i.e. the inability to translate abstract descriptions of locations and areas into crisp coordinates as needed by the control mechanism. Most agent control languages in the literature represent locations as specific points in space, i.e. as 2D or 3D coordinates. In some cases, locations can be determined by areas predefined by the designer, or they can be dynamically assigned to the position of another object or agent (Arafa and Mamdani, 2003). Such representations are adequate for simple instructions such as 'go to the door' or 'look at John'. However, in the case of dynamic and non-deterministic environments, the agent's locomotion and spatial behavior in general (gaze, object placement, etc.) cannot be based on predefined locations or simply on the position of another object.

Fuzzy logic is an effective means of representing vague and imprecise knowledge, and has already been used successfully in the field of robotics (Benreguieg et al, 1997). It can, therefore, be used for representing vague descriptions of locations and areas, and for building a virtual agent control mechanism that can operate with vague data.

In this paper, we present a framework for handling spatial vagueness in virtual environments, which could extend existing agent control languages and increase their autonomy and adaptability. We define vague locations as a representation scheme that supports linguistic descriptions of locations and propose an agent control architecture that is using a fuzzy rule-based system to take low-level decisions concerning the agent's spatial behavior. We have implemented a language for scripting the agent's control mechanism using vague locations and we present a case study of an agent operating in a dynamic environment using the proposed architecture.

2 VAGUE LOCATIONS

The problem of spatial vagueness (Hazarika and Cohn, 2001) is that people think and reason about locations and areas in a qualitative manner in contrast to software agents operate with crisp coordinates. A designer may wish to use regions instead of coordinates as execution parameters, e.g. 'on the table', 'in the room', etc., and in some cases it may also be needed to represent locations in terms of vague regions such as 'near the wall', 'in front of the table', etc.

We define a *Vague Location L* as a binary fuzzy relation in the Cartesian plane:

$$L = \{((x,y), \mu_L(x,y)) \mid (x,y) \in \Re^2\}$$

The degree of membership μ_L of each pair (x,y) in the vague location represents the plausibility of a point at (x,y) belonging to the location represented by L.

We define affine transformations (i.e. translation, rotation and scale) on vague locations as follows: Let L be a vague location with membership function μ_L. Let also M be a 2x2 affine transformation matrix. Then L' is the transformation of L by M iff:

$$\forall(x',y') | \mu_{L'}(x',y') = \mu_L(x,y) \; [x \; y]^T = [x' \; y']^T M^{-1}$$

Vague Locations can be further combined in more complex linguistic expressions using fuzzy operators and quantifiers, such as AND, OR, NOT and VERY. We propose plausible representations for distance and direction relations (Vazirgiannis, 2000) as vague locations. As stated in (Gapp, 1994), the interpretation of spatial relations depends strongly on the reference object's size. Furthermore, people do not account for every detail of the reference object when applying spatial relations (Landau and Jackendoff, 1993), and therefore, an oriented bounding rectangle approximation can be used. Based on the above, the definition of vague locations from spatial relations can take place on a prototype object, which will then be scaled, translated and rotated according to the geometric properties of the reference object. We use an object with a bounding rectangle of size 1 x 1 located at the origin and oriented towards the positive Y-axis as the prototype on which all spatial relations are defined.

Concerning the spatial relation 'near', we define N as the near value, a distance below which a coordinate is assumed to be near the reference object, and F as the far value, a distance beyond which a coordinate is assumed to be far from that object. The prototype NEAR membership function for that object is defined as follows:

$$\mu_{NEAR}(x,y) = \begin{cases} d \le N, & 1 \\ N < d < F, & 1 - \dfrac{d-N}{F-N} \\ d \ge F, & 0 \end{cases}$$

where d is the Euclidean distance of the point (x,y) from the origin, thus: $d = \sqrt{x^2 + y^2}$. Fig. 1 shows a possible vague location 'near object A'.

A plausible interpretation of the spatial relation 'far' is to define its membership function as the equivalent to the NOT NEAR function, so $\mu_{FAR}(x,y) = 1\text{-}\mu_{NEAR}(x,y)$.

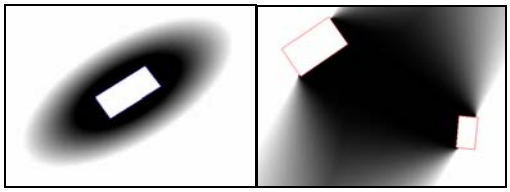

Figure 1: Vague Locations of 'near A' and 'between A and B'.

Concerning direction relations 'front of', 'behind', 'left of' and 'right of', we propose the membership function FRONT, which is a plausible interpretation of the 'front of' spatial relation of the prototype reference object. It is defined as follows:

$$\mu_{FRONT}(x,y) = \begin{cases} y \le 0, & 0 \\ y > 0; x < -0.5, & \dfrac{y}{\sqrt{(x+0.5)^2 + y^2}} \\ y > 0; -0.5 \le x \le 0.5, & 1 \\ y > 0; x > 0.5, & \dfrac{y}{\sqrt{(x-0.5)^2 + y^2}} \end{cases}$$

Spatial relations can be interpreted in various ways (Olivier and Tsujii, 1994), according to the frame of reference. We use the *intrinsic* and *deictic* interpretation of direction relations. In the intrinsic interpretation the reference frame is centered at the reference object and adopts its intrinsic orientation, whilst in the deictic interpretation, it is centered at the observer and adopts his orientation. The vague location of a directional relation for an actual reference object in the environment is calculated by scaling the prototype FRONT location by the reference object's size, by translating it to the reference object's position and by rotating it to the appropriate direction denoted by the type of the directional relation.

Concerning the vague location 'between objects A and B' (Fig. 1), we adopt the following approach: the Vague Location D_FRONT_OF_A

is calculated, which is the deictic 'front of' using object A as reference object and B as observer. Then, the respective location D_FRONT_OF_B is calculated by swapping reference object and observer. The final Vague Location BETWEEN_A_AND_B is the combination of these two locations using the AND connective: $\mu_{BETWEEN\_A\_AND\_B}(x,y) = \min(\mu_{D\_FRONT\_OF\_A}(x,y), \mu_{D\_FRONT\_OF\_B}(x,y))$.

3 A FUZZY RULE-BASED CONTROLLER

An agent is usually equipped with a visual sensor that updates its internal model of the environment based on the objects that are in its field of view, and with a number of effectors that can execute actions sequentially or in parallel. Some of these effectors may utilize spatial behaviors, e.g. in the case of anthropomorphic agents there can be effectors such as gaze direction, body orientation, pointing, target-based navigation, etc. Spatial vagueness exists at both the perception the action levels. We propose a fuzzy rule-based mechanism for the low-level decision process of virtual agents in dynamic environments that operates using vague locations. The proposed architecture is presented in Fig. 2.

All sensor data are stored in the agent's memory, which contains the known objects and their property values. The agent's effectors operate using crisp positions. They have an equal number of fuzzy rule sets assigned to them, and they receive crisp input after a complete fuzzyfication - evaluation - defuzzification loop. Fuzzy rule sets contain condition – action rules that are defined by the designer using vague locations. The condition part of a rule may be a simple or a compound condition.

The *fuzzification* process is executed in two steps. Initially, all vague locations are recalculated using the actual geometric properties of the objects they refer to, as described in section 2. Then, the condition of each rule is examined and a truth value is assigned to it.

During the *rule evaluation* process, a fuzzy inference takes place. The truth value of each condition defines the proportion of the respective rule taking part in the final output. We adopt the PRODUCT inference method, according to which, the vague location defined in the action part of each rule will be scaled down by the truth value of the respective condition. Thus, if $t \in [0,1]$ is the truth value of the condition of a rule and L is the vague location of the action part, the output location L' of this rule will be defined as: $\mu_{L'}(x,y) = \mu_L(x,y) \cdot t$

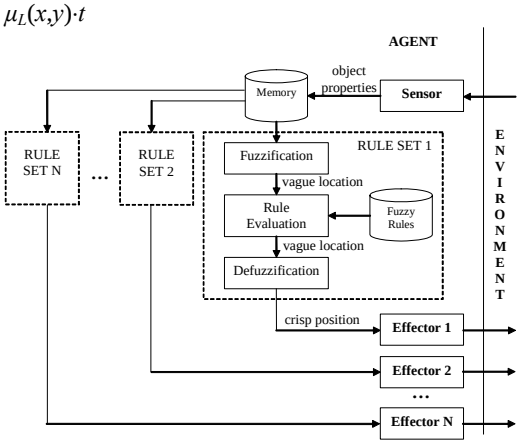

Figure 2: Architecture of the fuzzy rule-based controller

The next step in the evaluation process is the composition of the rules into the resulting vague location. We use the MAX composition method: the membership value of each coordinate in the vague location L is the maximum value of all respective membership values of the same coordinate in the output locations of the rules. Thus, if $L_1, L_2, \ldots, L_N$ the output locations of the rules, the resulting location L will be defined as:

$\mu_L(x,y) = \max(\mu_{L1}(x,y), \mu_{L2}(x,y), \ldots, \mu_{LN}(x,y))$

The final step is the defuzzification process, in which a crisp result is produced from the resulting vague location. Using the CENTROID defuziffication method on a vague location L, a crisp position (x,y) can be calculated as follows:

$$x = \frac{\iint x \cdot \mu_L(x,y)\,dx\,dy}{\iint \mu_L(x,y)\,dx\,dy}, \quad y = \frac{\iint y \cdot \mu_L(x,y)\,dx\,dy}{\iint \mu_L(x,y)\,dx\,dy}$$

The crisp output position is used by the effector as argument to execute the agent's actions.

4 CASE STUDY

We have created a prototype implementation of the control mechanism described in the previous section and connected it to the locomotion process of a virtual agent in a 3D environment. Additionally, we have designed and implemented a scripting language for defining the fuzzy rule set of the agent using vague locations. The representation of spatial relations as vague locations is processed after projecting the 3D model of the reference

object(s) on the ground plane and calculating their oriented bounding rectangle. The complete system has been implemented in Java and Java3D.

The implemented system has been used to set up a case study, in order to assess the functionality of the proposed framework. We have designed an agent to operate as a guide in a virtual exhibition. The 3D environment contains the static geometry (walls, doors, etc.), a number of exhibits represented as 3D objects, the agent and the user's avatar. The goals of the agent are:

- To avoid collision with static objects or the user
- To stand besides the exhibit that the user is currently looking at, and present it to him.

The first test was the collision avoidance using the vague location framework. We used the following set of rules:

```
1: if nearest in front_of *me
      and left_of *me
      and near *me
   then move_to right_of *me
2: if nearest in front_of *me
      and right_of *me
      and near *me
   then move_to left_of *me
```

According to these rules, if the object that is nearest to the agent is in front of the agent and at a close distance, then it is probably going to obstruct its navigation. In this case the agent is ordered to move to the left if the obstacle is located to its right, and the vice versa.

The second goal involved dynamic positioning of the agent. It had to move itself to a place near the exhibit, in order to present it to the user, but not in a position that blocks the user's view. Therefore, the agent should not be in front of the exhibit as seen by the user. Based on these requirements, the final rule was defined as follows:

```
3: if "exhibit" in front_of "avatar"
      and near "avatar"
   then move_to very near "exhibit"
      and not d_front_of "exhibit"
      asb "avatar"
```

The condition of this rule tests whether an exhibit is located in front of the user ("avatar") and near him. In that case, the user is probably studying the exhibit. If this rule fires, the agent is ordered to move to a place that is close to the exhibit (using the spatial relation 'near' and the fuzzy quantifier 'very'), but not in the region defined by the deictic relation 'd_front_of' with the exhibit as a reference object and the user as an observer.

6 CONCLUSIONS

We have presented a framework for handling spatial vagueness in virtual agent control using binary fuzzy relations in the Cartesian plane to represent vague locations. We have proposed plausible interpretations of spatial relations using vague locations and presented a fuzzy rule-based control mechanism that operates in dynamic environments with linguistic descriptions of locations. The main advantages of the proposed approach are the functionality it offers a designer to define complex locations at both the perception and action level of an agent, and the ability of the control process to demonstrate adaptive behavior in dynamic environments.

ACKNOWLEDGEMENTS

This research is partially supported by the Pythagoras EPEAEK II Programme of the Greek Ministry of National Education and Religious Affairs, co-funded by the European Union.

REFERENCES

Arafa, Y., Mamdani, A., 2003. Scripting embodied agents behaviour with CML: character markup language. In Proc. of International Conference on Intelligent User Interfaces, pp.313-316.

Aylett, R., Luck, M., 2000. Applying artificial intelligence to virtual reality: Intelligent virtual environments. In Applied Artificial Intelligence, 14 (1), pp.3-32.

Benreguieg, et al, 1997. Fuzzy navigation strategy : Application to two distinct autonomous mobile robots. In Robotica, 15, pp. 609-615.

Gapp, K.-P., 1994. Basic meanings of spatial relations: Computation and evaluation in 3d space. In Proc. of AAAI94, Seattle, WA, pp.1393-1398.

Hazarika, S., Cohn, A.G., 2001. A Taxonomy for Spatial Vagueness: An Alternative Egg-Yolk Interpretation. In Proc. of SVUG'01.

Landau, B., Jackendoff, R., 1993. What and where in spatial language and spatial cognition. In Behavioral and Brain Sciences, 16 (2), pp.217-265.

Olivier, P., Tsujii, J.-I., 1994. Quantitative perceptual representation of prepositional semantics. In Artificial Intelligence Review, 8, 147-158.

Vazirgiannis, M., 2000. Uncertainty handling in Spatial Relationships. In Proc. of the 2000 ACM symposium on Applied computing, pp.494-500.