Recognizing Customers' Mood in 3D Shopping Malls Based on the Trajectories of Their Avatars

Anton Bogdanovych[1], Mathias Bauer[2], and Simeon Simoff[1]

[1] School of Computing and Mathematics, University of Western Sydney, Australia
{A.Bogdanovych, S.Simoff}@uws.edu.au
[2] Mineway GmbH, Science Park 2, Stuhlsatzenhausweg, 66123 Saarbruecken, Germany
mbauer@mineway.de

Abstract. This paper proposes a method to assess the cognitive state of a human embodied as an avatar inside a 3-dimensional virtual shop. In order to do so we analyze the trajectories of the avatar movements to classify them against the set of predefined prototypes. To perform the classification we use the trajectory comparison algorithm based on the combination of the Levenshtein Distance and the Euclidean Distance. The proposed method is applied in a distributed manner to solving the problem of making autonomous assistants in virtual stores recognize the intentions of the customers.

Keywords: 3D virtual worlds, Trajectory recognition, Avatar, e-Commerce.

1 Introduction

Starting as game-oriented technology 3D Virtual Worlds became one of the few successful online businesses that are making money on the Web [1]. The popularity of Virtual Worlds grows and the demand for them to be applied to a wider range of domains (i.e. Electronic Commerce, Tourism, Museums) becomes more and more explicit. Another trend associated with the growth of Virtual Worlds is the demand for mixed societies, which can be populated by both humans and autonomous computational agents. This demand is stimulated by the desire of businesses establishing in Virtual Worlds to provide customers with adequate assistance and, at the same time, save on human resources by employing autonomous agents.

A particularly interesting case of E-Commerce environments that would benefit from the presence of autonomous shopping assistants represent 3D Shopping Malls. 3D Shopping Malls are online stores located in various Virtual Worlds (e.g. Second Life). Due to the fact that all the participants in 3D Virtual Worlds share similar embodiment and the environment allows for full observation of customer actions – 3D Virtual Worlds offer a potentially better platform for development of intelligent shopping assistants than form based interfaces [2].

In 3D Virtual Worlds the range of possible actions is much wider. Every mouse or keyboard event can be associated with a number of attributes that play a role in triggering this event, and those attributes can be easily analyzed and extracted from the environment. Every movement of an avatar[1] can be precisely described by a set of

[1] Avatars are graphical representations of humanoids in Virtual Worlds.

J. Filipe and J. Cordeiro (Eds.): ICEIS 2009, LNBIP 24, pp. 745–757, 2009.

transformation vectors and each of those vectors can be easily related to the surrounding objects, helping to build a mathematical model of the training data. Moreover, analyzing the movements of the human can reveal information about his/her cognitive state [3], which is difficult to figure out in form-based interfaces.

On the example of museum visitors [4] it was shown the choice of the correct assistance strategy required by an individual is highly dependent on his/her cognitive state. Some aspects of the cognitive state can be directly recognized from the trajectory of movement of this individual. In [3] it is described how motion-based information can play an important role in recognizing various aspects of the cognitive state of the user including a degree of user's commitment to a goal and the goal itself. The paper suggests using the standard CAPRI algorithm for mining the association rules from the movement profile of the users, which in our case may be used for selecting an appropriate assistance style, and determining the relevance of a particular piece of information the assistant is about to present to the user. Another idea proposed in [3] is collecting a number of motion sequences in a certain area and then determining clusters of similar motions, where each cluster corresponds to a specific value of the cognitive state.

This paper extends the work presented in [3] and applies it to the domain of 3D Shopping Malls. Here we provide a details of the clustering algorithm, outline the method for calculating the distance between motion sequences and conduct a set of experiments for validating this method. It is also shown how this clustering relates to recognizing the cognitive state of the customers visiting virtual shops. The suggested clustering method is based on the Levenshtein Distance and the Euclidean Distance measures, so it can capture the geometrical features of the motion sequences. While the same problem could have been solved using classical data mining methods, using geometrical features is quite important for the domain of 3D E-Commerce as such information can be used for relating the motion sequence to a particular location (product).

Another contribution of our work is "Distributed User Modeling" that is a technique for introducing the distributed data mining into the domain of 3D E-Commerce.

The remainder of the paper is structured as follows. Section 2 explains the concept of cognitive state, discusses how it can be assessed on the basis of a trajectory and outlines the testing scenario. Section 3 presents our distributed user modeling approach for solving the problem with analyzing the trajectory of the avatars not directly controlled by the assistant agent. In Section 4 we outline the algorithm that we use for discovering some aspects of the customer's cognitive state on the basis of his/her avatar's trajectory. Section 5 demonstrates the results of the experiments we have conducted. Finally, Section 6 presents some concluding remarks and the direction of future work.

2 Assessing Cognitive State

Cognitive state is broad term used in different disciplines. Below is the definition that most accurately reflects what is understood by the cognitive state in this paper.

Definition. ***Cognitive State*** *is the state of a person's cognitive processes*[2]. *In DAI*[3] *cognitive state is usually associated with intentions, beliefs and desires of an individual* [5].

[2] http://www.dictionary.com

[3] Distributed Artificial Intelligence.

It is not the goal of this paper to analyze different aspects of the cognitive state and provide a comprehensive study on how they can be learned. Instead, our goal is to show the potential that 3D E-Commerce domain provides in analyzing it. Therefore, the presentation here is limited to analyzing only one aspect of the cognitive state, namely the mood of the customer inside a virtual shop.

The mood aspect of the cognitive state has received particular attention in ethnography [4]. Based on the mood of the people visiting art expositions in museums researchers identified four distinct categories of visitors, briefly summarized in [4].

1. The *ant visitor* spends a long time to observe all exhibits, stops frequently and usually moves close to walls and exhibits, avoiding empty spaces.
2. The *fish visitor* moves preferably in the center of the room, walking through empty spaces. Fish visitors do not look at details of exhibits and make just a few or no stops; most of the exhibits are seen but for a short time.
3. The *grasshopper visitor* only sees the exhibits which comply with grasshopper's interests. These personal interests and pre-existing knowledge about the contents of the exhibition guide the grasshopper. The grasshopper quickly crosses empty spaces, but the time spent on observing selected exhibits is quite long.
4. The *butterfly visitor* frequently changes the direction of visit, usually avoiding empty spaces. The butterfly sees almost all the exhibits, stopping frequently, but times vary for each exhibit.

The problem of assisting a customer in a virtual shop is very similar to the problem of assisting a visitor of an art exhibition in a museum. In fact, most of the existing virtual shops in the domain of 3D E-Commerce have a similar set up as an art exhibition. Many of such shops, due to the high price of 3D modeling, have the pictures of the products located alongside the walls of the store.

2.1 The Poster Shop Scenario

In our experiments we have used a poster shop, where the goods offered for sale were various graffiti posters placed on the walls of the virtual room.

Due to a close similarity with the art exhibition domain we decided to use the same 4 mood state values: "Ant", "Grasshopper", "Butterfly" and "Fish" to represent Buyer's behavior in the shop room. However, in contrast to the art exhibition in our scenario only one room was used for conducting the experiments. Another difference with the art gallery scenario from [4] is that we need to analyze the trajectories of the visitors dynamically, meaning that it is not acceptable to wait until a participant exits the room to be able to classify their mood. In our case mood classification is required to be completed before the participant approaches the assistant. Having these limitations requires a slight modification of the legends behind the behaviors presented in [4].

As adapted to our scenario, in the "Ant" state the main task of the user is considered to be walking around the room and absorbing the visual information presented there. In the "Grasshopper" state a user is focused on particular items in the room (graffiti posters) and requires more information about them. In the "Butterfly" state the user is experiencing the problem with either navigating the Virtual World or with the presented information. Being in "Fish" state means that the user is focused on some task outside the shop area, wishes to pass through the room as quickly as possible and doesn't want to be distracted from the main activity.

The mood of the visitors of the graffiti poster shop affects the way they perceive information. For the assistance purposes knowing it is very important to determine which strategy to select and to understand whether any assistance is required at all. Clearly the person in the "Fish" state would not be very excited about hearing the information about graffiti posters and the person in "Butterfly" state experiencing navigational problems would first like to know how to solve the problem and only then may express some interest in the poster exhibition.

3 Distributed User Modelling

In the above presented scenario the Assistant agent has to be able to asses the cognitive state of the Buyer approaching the Assistant to select an appropriate assistance strategy. As shown above, the trajectory of the Buyer could reveal some of the aspects of the cognitive state. Analyzing a trajectory of another avatar, however, is a very challenging task associated with a number of problems. The Assistant agent must constantly observe every Buyer that enters the room and translate the movements of the Buyers into arrays of landmarks. Without being able to directly acquire this information from the system, analyzing the cognitive state becomes nearly impossible.

Obtaining landmarks and their precise coordinates just from observation of the movements of one avatar through the field of view of another avatar is a very challenging task. This task could be simplified by letting the system provide every agent with detailed information about all the other agents (including the positions of the corresponding avatars at a given time). Such simplification, however, increases computational load and raises privacy concerns.

To be able to make the trajectory recognition achievable and, at the same time, ensure the privacy of participants we propose the following decentralized solution to the problem. Each autonomous agent only observes its principal and dynamically updates the user model of the principal (in our particular case the user model only constitutes the cognitive state). When some other agent (i.e. the Assistant) needs to obtain some information from the user profile of this agent (Buyer), instead of trying to observe the behavior of its avatar and use sophisticated modeling techniques, it simply sends a direct request to the agent "responsible" for the corresponding avatar. If the other agent agrees to share the information it will reply with the relevant part of its user profile.

Such decentralized solution is feasible because the duality (agent/principal) is a general feature of Virtual Institutions technology employed for the development of our scenario. In Virtual Institutions [2] every participant is integrated into the system via such architecture. The proposed approach can significantly reduce the amount of computations and the size of the stored data. It also permits easier control over privacy (e.g. if a participant doesn't want to be observed he/she just prohibits the agent to share personal information with others or may even select which aspects of the user profile can be shared and which aspects are private). The decentralized approach also helps to easily use the characteristics of the user profiles of the surrounding agents in the system as the attributes for other kinds of machine learning. The Virtual Institutions architecture

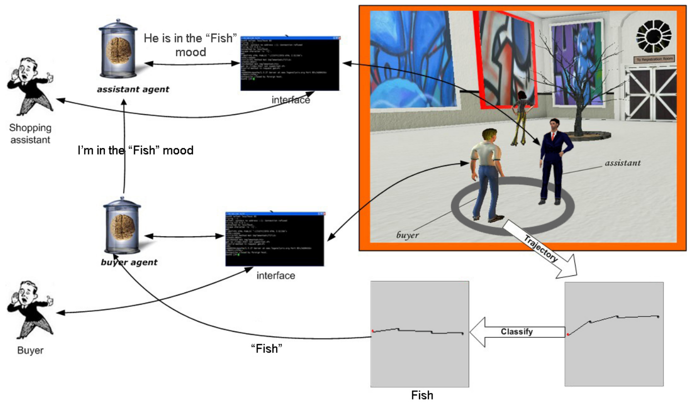

Fig. 1. Distributed User Modeling

supports that an agent that observes the behavior of its principal may directly communicate with the agents attached to the avatars currently visible to it and ask them to share a particular part of their profile. The elements of the obtained user profile can then be used as the input for a classifier in implicit training of the assistant agent [6].

Figure 1 graphically presents the idea of distributed user modeling. It outlines the case of a human controlling the avatar marked as “Buyer” and an autonomous agent controls the avatar of the “Assistant”. In the beginning of the scenario outlined in the figure the Assistant representative agent notices a Buyer approaching its avatar. To be able for the Assistant to detect that it was approached the notion of *audibility distance* is used. The audibility distance represents the radius of the imaginary circle surrounding the avatar (audibility zone), which is used to determine the range in which everything that is said by any other avatar will be heard, while outside of this range nothing can be heard. Audibility distance is a very useful concept for facilitating social interactions and providing a natural way to filter the communications. Another purpose of the audibility zone in our system is that the fact of entering the audibility zone of one avatar by another avatar means that the first avatar was approached by the second one.

Once approached, the Assistant requests the description of the Buyer’s cognitive state from the Buyer representative agent. In its current form the state corresponds to the label describing the mood of the human. This label is acquired by the Buyer representative agent through comparing the current approaching trajectory of the Buyer with the set of predefined prototypes and extracting the label of the prototype that has the closest match. In the scenario outlined in the figure this label is “Fish”.

Once the label is assigned it is sent back to the Assistant representative agent. Before starting a conversation, the Assistant already knows that it shouldn’t bother the visitor with additional information about the posters and if asked directly, its responses have to be very short and precise.

Notice that it is also possible that the Assistant avatar is controlled by the human. In this case the corresponding autonomous agent will inform the human about the mood of the approaching participants and the human will be able to use this information for selecting the right strategy, and will train the agent accordingly.

4 Trajectory Comparison

The cognitive state of the Buyer is estimated on the basis of its approaching trajectory. In this paper we do not intend to prove the connection between the movement of the avatars and the cognitive state of the humans generating these movements, but rely on the outcomes of the research presented in [4] to make this link. We use the adapted versions of the four trajectories presented in [4] as the basis for the trajectory comparison.

Each of the four trajectory prototypes is stored in the classification list. In order to classify the approaching trajectory of an avatar we compare it with every trajectory in the classification list and identify the most similar one from the list. The label associated with the resulting trajectory is considered to be the result of the classification.

Technically, the trajectories are specified as arrays of landmarks. Each of the landmarks corresponds to a position of an avatar in a given moment. The position is permanently updated by the system every 50 Ms, so the information about avatar's velocity is easily reconstructed from the distance between two neighboring landmarks. This simple representation allows efficient trajectory classification. To increase the performance of the classification on the first step of the algorithm the irrelevant landmarks (noise) are removed using the approach presented in [7].

After this, a combination of Levenshtein Distance and Euclidean Distance algorithms is applied to compare the analyzed trajectory with each trajectory stored in the classification list. As the result of the comparison, the trajectory from the classification list with the lowest distance value is selected and the corresponding text value is extracted to be used as a behavior label for the cognitive state of the human.

4.1 Levenshtein Distance

The Levenshtein Distance is the algorithm normally used to measure the distance between two strings. It determines the minimum number of operations needed to transform one string into another given string, where possible operations are insertion, deletion, or substitution of a single character [8]. The steps of the Levenshtein Distance Algorithm [8] are presented in Table 1. Here *s* and *t* are the two strings being compared, *n* – the length of string s and *m* – the length of string t.

In order to apply the Levenshtein Distance to the trajectory comparison we propose the following modifications to the original algorithm. Firstly, instead of comparing the Strings we compare arrays of landmarks. So each of the $s[i]$ and $t[j]$ will be a point in a 3-dimensional system of coordinates – (x_i, y_i, z_i) and (x_j, y_j, z_j), respectively.

Second change is the replacement of the *cost* assessment model. In the original algorithm the cost can be seen as the actual distance between two characters. This cost model is very simple and is equal to "0" if the two characters are similar and is equal to "1" if the characters are different.

In our case we are dealing with arrays of landmarks instead of characters. Each landmark has a unique coordinate in the 3-dimensional space and, therefore, instead of just having "0" and "1" we can employ a more appropriate distance measurement technique, namely, Euclidean Distance.

The Euclidean Distance between two points in a 3D space is calculated as follows:

$$D_{euclid} = \sqrt{(p_1.x_1 - p_2.x_2)^2 + (p_1.y_1 - p_2.y_2)^2 \times (p_1.z_1 - p_2.z_2)^2} \qquad (1)$$

Table 1. The Steps of the Levenshtein Distance Algorithm

Step	Description
1	Set n to be the length of s (first string). Set m to be the length of t (second string). If n = 0, return m and exit. If m = 0, return n and exit. Construct a matrix containing 0..m rows and 0..n columns.
2	Initialize the first row to 0..n. Initialize the first column to 0..m.
3	Examine each character of s (i from 1 to n).
4	Examine each character of t (j from 1 to m).
5	If s[i] equals t[j], the cost is 0. If s[i] doesn't equal t[j], the cost is 1.
6	Set cell d[i,j] of the matrix equal to the minimum of: a. The cell immediately above plus 1: d[i-1,j] + 1. b. The cell immediately to the left plus 1: d[i,j-1] + 1. c. The cell diagonally above and to the left plus the cost: d[i-1,j-1] + cost.
7	After the iteration steps (3, 4, 5, 6) are complete, the distance is found in cell d[n,m].

Here D_{euclid} is the Euclidean distance, p_1 and p_2 – the landmarks for which the distance is measured and (x_1, y_1, z_1) and (x_2, y_2, z_2) – coordinates of p_1 and p_2.

In theory the values of the Euclidean Distance can vary between "0" and infinity, practically the distance is always limited by the dimensions of the space where the measurement is taking place. The cost value in the original Levenshtein Distance algorithm is required to be normalized (take values in the [0,1] interval). Therefore, instead of using pure distance value we use the following equation:

$$cost = \frac{D_{euclid}}{\sqrt{R.width^2 + R.height^2 + R.depth^2}} \tag{2}$$

Here $cost$ is the value that should be used instead of "1" on the step 5 of the Levenshtein Distance algorithm. And $R.width$, $R.height$, $R.depth$ are the dimensions of the room used for trajectory classification. In case the trajectory comparison is required to be done outside any of the rooms, R should correspond to the space where the experiment is conducted with $width, height, depth$ being the dimensions of this space.

5 Experiments

To verify the proposed method for trajectory recognition and classification of the cognitive state as well as to test its accuracy we have conducted a series of experiments.

The aim with the assessment of the cognitive state was not to predict the actual cognitive state of the user, e.g. whether a user really was expressing fish browsing style or was rather a grasshopper. Instead, we wanted to prove that the trajectory recognition method based on the combination of Levenshtein Distance and Euclidean Distance is an appropriate trajectory clustering technique.

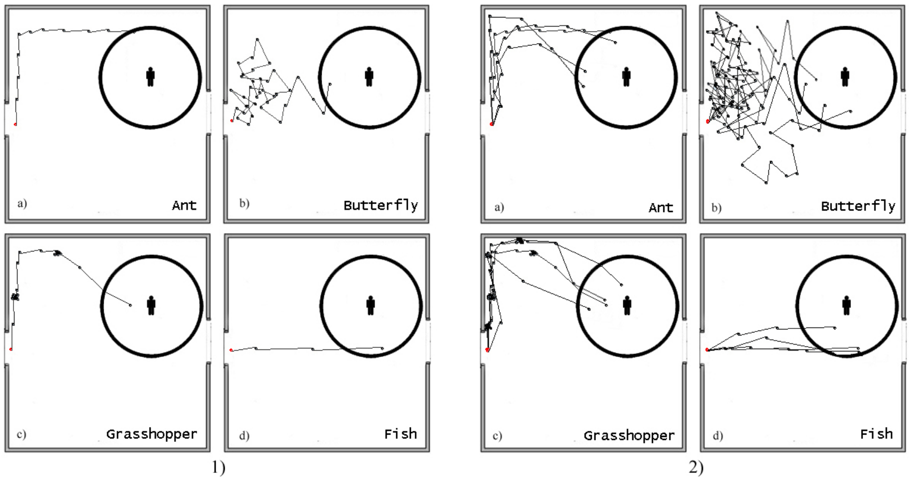

Fig. 2. Trajectories Used for Training and Experiments

5.1 Design of Experiments

For our experiments we implemented the scenario outlined in Figure 1. Test subjects playing the buyer role entered the poster shop room and the assistant had to identify their mood. The Virtual World consisted of the 3 rooms, only one of which was used for our experiments. The schematic representation of the room is shown in Figure 1.

To asses the mood of the buyer each individual agent was observing the trajectory of its principal and comparing the current trajectory with each of those in the classification list. The classification list contained 4 prototypes as displayed in Figure 2. 1.

Here each trajectory is shown as a set of landmarks connected by lines. Each of the landmarks corresponds to the position of the avatar at the moment of measurement. Each new measurement was conducted every 50 Ms. The schematic representation of the poster shop is added to show the context under which the trajectories were obtained. Each of the landmarks is projected onto the corresponding position in the shop room and the label describing the trajectory is shown in the bottom right corner of the room. The figure consists of 4 duplicates of the poster shop marked as “a)”, “b)”, “c)” and “d)”. The trajectories on each of those duplicates corresponds to one class of the cognitive state as marked in the picture. The black solid figure present in each copy of the poster shop room represents the autonomous agent associated with the assistant and the circle around it represents the audibility zone.

The trajectory in Figure 2.1 a) corresponds to the case when a buyer enters the shop not aware of its content. Once inside the room the buyer moves along the wall with a moderate speed checking out the presented posters. Here we wanted to present the case of a curious browser, who has no specific interests or knowledge about the presented products and wants to make a sound decision by browsing through all the posters presented there. This trajectory was associated with the label “Ant”. This trajectory is characterized by the monotonic speed of the user movement along the walls.

In Figure 2.1. b) we present the case of the participant randomly walking around expressing a high degree of confusion. Such a trajectory is typically generated by novice participants who are not yet quite familiar with controlling their avatars. Label "Butterfly" is assigned to this trajectory. The key characteristic of this movement class is that a participant frequently changes the movement direction and returns to the location close to initial point a number of times.

Figure 2.1 c) illustrates the case of the visitor of the room who has particular interest in some posters and no interest in the others. From the distance between landmarks it is clear that the browsing speed is not constant. In front of two pictures the human stopped for a while and then headed very fast straight towards the exit. This trajectory is labeled as "Grasshopper". On the picture the groups of landmarks placed closely together do not represent very short movements but are due to the fact that during recording a landmark is added after a constant interval of time even if there was no movement produced.

Finally, Figure 2.1 d) illustrates the case when a participant has no interest in buying any posters and shortly after entering the room quickly walks towards the exit (using the poster shop as a corridor) in order to continue with some activities in the next room. The label we use for this behavior is "Fish". The key characteristics here: the high speed of movement, which can be recognized from the distance between the landmarks and the fact that the participants moves in the middle of the rooms, away from its walls.

As it is clearly seen on the pictures, each of the trajectories was recorded from the moment a human entered the room until the moment the corresponding avatar approached the assistant within the audibility distance. In our experiments once the avatar entered the audibility zone of the assistant agent the recording of the trajectory was finished, the currently recorded sequence of landmarks was compared with each of the prototypical 4 sequences and the trajectory with the lowest Levenshtein Distance was selected as the class describing the cognitive state of the human. Then, the corresponding label associated with this trajectory was sent by the autonomous agent of the buyer to the autonomous agent of the Assistant to inform it about the mood of the buyer. This information was further used by the assistant to decide whether to offer help (the help should not be offered to "Fish") and what kind of assistance is required (i.e. the "Butterfly" participant needs a different type of assistance than "Ant" or "Grasshopper").

To validate the trajectory recognition method we conducted a series of experiments with the set of 50 different movement sequences executed in the poster shop. The human operator playing the "Buyer" role was told about the distinct characteristics of each of the 4 classes presented earlier and then was asked to produce 10 different movement patterns for each of the classes so that those patterns would match the given descriptions and at the same time would be distinct. Each of the 10 patterns would end with buyer approaching the assistant. The result of the classification of buyer's trajectory by the assistant agent was printed in the chat window.

Table 2 outlines the results of the conducted experiments. In this table the "**Nr**" column shows the experiment number and the "**Result**" column stores the label printed in the chat window as the result of the classification. The columns marked as "D_{ant}", "$D_{butterfly}$", "D_{fish}", "$D_{grasshopper}$" store the value of the Levenshtein distance between the trajectory from the experiment and the prototypical "Ant", "Butterfly", "Grasshopper" and "Fish" trajectories correspondingly. The "**Correct**" column shows whether the behavior the test subject was intended to demonstrate was correctly recognized.

To give an impression about the movement patterns expressed by the operator Figure 2.2 outlines the results of the 16 out of first 50 experiments we have conducted. We show only 16 (4 per each of the classes) to avoid overcrowding the picture with unnecessary data. The recorded trajectories exemplify the series of movements which begin when the operator driving the avatar of the "Buyer" agent entered the poster shop room and end at the moment this avatar entered the audibility zone of the Assistant agent. For presentation purposes the trajectories are projected onto the schematic representation of the poster shop room. Each of the recordings is classified into one of the four classes as in Figure 2.1. For simplicity of understanding we placed all of the trajectories having the same class into the same part of the figure, which allows for a better comparison.

The 4 trajectories presented in Figure 2.2 a) correspond to the experiments 1–4 in Table 2. In Figure 2.2 b) the experiments 11–14 are outlined. Figure 2.2 c) and Figure 2.2 d) show the experiments 21–24 and 31–34 correspondingly.

One of the goals of the experiments was to highlight the benefits of using geometrical features of the training data for classification. Experiments 41–45 and 46–50 were testing the hypothesis that using the Euclidean Distance as the cost in the comparison will help capturing the specifics of each particular shop and allow for an easy way of expressing the location based preferences. In the experiments 41–45 the test subject was demonstrating the "Ant" behavior, but instead of moving along the upper side of the room was asked to move using similar movement style, but in the bottom part of the room. The same was done for the "Grasshopper" behavior in the experiments 46–50. Using classical data mining methods (which do not take geometrical characteristics into account) for this case would most likely result no difference between moving in the upper part of the room or in its bottom part, both would be classified as being identical. In the 3D E-Commerce domain, however, such a situation is often not acceptable. In particular, for the "Grasshopper" case the posters in front of which users stops make a big difference for making a decision on what kind of information a user might require. Note that for the experiments 41–50 the "**Correct**" column in the results table tells whether the intended behavior ("Ant" and "Grasshopper") is recognized or not.

5.2 Discussion of Results

As shown in Table 2, two of the presented four classes were correctly identified by the system in all cases. These classes are "Ant" and "Fish". Out of 10 experiments per each of those classes 10 were recognized correctly making it 100% classification accuracy.

The "Butterfly" trajectory was also detected very accurately with the precision of 80%, where 8 out of 10 generated examples were recognized correctly. The motion in vertical direction in one of the misclassified trajectories was very low, which became the main reason why this pattern was classified as "Fish". In another misclassified example the operator approached the initial position only twice (while in the prototypical trajectory it happened 4 times) and, therefore, this pattern was also classified as "Fish".

The recognition of the "Grasshopper" trajectory showed the worst precision of only 60%. It proved to be too similar to the "Ant" behavior with all of the 4 misclassified examples being recognized as "Ant".

To gain a detailed understanding for the reasons of misclassification we conducted 50 more experiments with "Grasshopper" and "Butterfly" classes. These experiments

Table 2. Experiments

Nr	D_{ant}	$D_{butterfly}$	D_{fish}	$D_{grasshopper}$	**Result**	**Correct**
1	0.46	3.58	2.03	0.70	ant	y
2	0.45	3.13	2.02	0.96	ant	y
3	0.63	3.13	2.02	0.96	ant	y
4	0.50	3.26	1.62	1.07	ant	y
5	0.68	3.07	2.60	1.42	ant	y
6	0.47	3.15	2.05	1.07	ant	y
7	0.56	3.02	1.87	0.82	ant	y
8	0.65	2.81	1.86	1.41	ant	y
9	0.55	2.83	1.97	0.96	ant	y
10	0.80	3.16	2.79	1.29	ant	y
11	6.05	3.80	3.58	6.22	fish	**n**
12	5.94	2.99	5.11	5.58	butterfly	y
13	3.23	2.57	2.39	3.85	fish	**n**
14	7.04	3.90	5.86	6.03	butterfly	y
15	4.41	2.69	3.65	4.44	butterfly	y
16	4.92	3.19	4.04	5.09	butterfly	y
17	4.54	2.58	3.95	4.46	butterfly	y
18	6.84	3.27	4.36	6.03	butterfly	y
19	6.37	3.28	5.51	5.73	butterfly	y
20	5.66	3.06	6.77	5.10	butterfly	y
21	0.67	3.99	4.82	1.27	ant	**n**
22	1.05	6.67	4.82	0.91	grasshopper	y
23	0.94	3.66	5.01	0.82	grasshopper	y
24	1.36	3.99	6.04	1.34	grasshopper	y
25	1.48	4.02	6.57	1.45	grasshopper	y
26	1.21	4.21	5.98	1.26	ant	**n**
27	1.27	4.17	6.71	1.21	grasshopper	y
28	1.05	4.03	5.68	1.11	ant	**n**
29	1.44	4.96	8.04	1.22	grasshopper	y
30	0.98	4.35	7.26	1.14	ant	**n**
31	2.46	3.37	0.34	4.72	fish	y
32	2.98	5.12	0.50	5.98	fish	y
33	2.89	4.85	0.38	5.89	fish	y
34	3.42	6.73	0.70	7.02	fish	y
35	2.56	3.95	0.36	5.12	fish	y
36	2.89	5.28	0.25	6.01	fish	y
37	3.04	4.54	0.62	5.81	fish	y
38	2.79	4.98	0.26	5.69	fish	y
39	2.72	4.08	0.35	5.47	fish	y
40	3.69	7.53	0.79	7.56	fish	y
41	5.24	8.05	2.64	8.52	fish	**n**
42	5.83	7.91	3.30	8.73	fish	**n**
43	5.52	7.20	2.82	8.28	fish	**n**
44	6.83	8.25	3.98	9.34	fish	**n**
45	8.07	9.32	4.94	10.44	fish	**n**
46	9.86	11.17	7.37	12.50	fish	**n**
47	8.51	9.84	6.07	11.21	fish	**n**
48	8.71	10.07	6.23	11.36	fish	**n**
49	7.25	8.70	4.83	9.93	fish	**n**
50	6.30	7.93	3.87	8.98	fish	**n**

showed that for the "Grasshopper" class the irregularities in speed (expressed by different distance between landmarks in "Grasshopper" class) had higher importance for the classification than the fact of a participant stopping. The trajectories with similar number of stops as in the "Grasshopper" class but with relatively constant movement speed in between stops had a very high risk of being classified as "Ant", while movement patterns with clear speed irregularities were more likely to be classified as "Grasshopper".

For the "Butterfly" class our initial hypothesis that the amplitude of the vertical movement as well as the number of returns to the initial position are the key factors in misclassification proved to be correct. The misclassified trajectories had the lowest value of the Levenshtein distance when compared with the trajectory from the "Fish" class, as the generated examples were too distinct, so the example with the fewest number of landmarks had the lowest distance value.

Confirming our original hypothesis, none of the experiments with numbers 41–50 have resulted in "Correct" classification. The use of geometrical features of the trajectories resulted in the classification being very sensitive to the actual positions of the posters. This indicates that our method can potentially distinguish between items located in different positions even if the same movement style is used to approach them.

6 Conclusions

In this paper we have showed how the trajectory of avatar's movement can be used to assess some aspects of customer's cognitive state. The selected method has proved to be capable of accurate trajectory classification with only one training example required per each class to be recognized. Furthermore, the use of Euclidean distance measure provides a possibility to make the classification position sensitive, which is a highly desired feature in the domain of 3D E-Commerce. Although, we base our initial assumption that a trajectory is tightly connected with the cognitive state of the human on the existing research and also find this fact intuitively right, in the future we plan to obtain more supporting evidence through additional experiments. Therefore, we are planning to further extend our system and conduct tests where the buyers will be able to evaluate how good the actual shopping mood was recognized, not only the trajectory. We also plan to use the presented method for implicit training of shopping assistants.

References

1. Hunter, D., Lastowka, F.G.: To kill an avatar. Legal Affairs (2003), `http://www.legalaffairs.org/issues/July-August-2003/feature_hunter_julaug03.msp`
2. Bogdanovych, A.: Virtual Institutions. PhD thesis, University of Technology, Sydney, Australia (2007)
3. Bauer, M., Deru, M.: Motion-Based Adaptation of Information Services for Mobile Users. In: Ardissono, L., Brna, P., Mitrović, A. (eds.) UM 2005. LNCS, vol. 3538, pp. 271–276. Springer, Heidelberg (2005)
4. Chittaro, L., Ieronutti, L.: A visual tool for tracing users behavior in virtual environments. In: AVI 2004: Proceedings of the working conference on Advanced visual interfaces, pp. 40–47. ACM Press, New York (2004)

5. Konolige, K., Pollack, M.E.: A representationalist theory of intention. In: Bajcsy, R. (ed.) Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI 1993), Chambéry, France, pp. 390–395. Morgan Kaufmann publishers Inc., San Mateo (1993)
6. Bogdanovych, A., Simoff, S., Sierra, C., Berger, H.: Implicit Training of Virtual Shopping Assistants in 3D Electronic Instituions. In: Proceedings of e-Commerce 2005 Conference, pp. 50–57 (2005)
7. Perng, C.S., Wang, H., Zhang, S.R., Parker, D.S.: Landmarks: A new model for similarity-based pattern querying in time series databases. In: Proceedings of the 16th International Conference on Data Engineering, pp. 33–42 (2000)
8. Levenshtein, V.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady 10, 707–710 (1966)