Virtual Clones: Data-Driven Social Navigation

Doron Friedman and Peleg Tuchman

The Advanced Virtuality Lab
Sammy Ofer School of Communications
Herzliya, Israel
doronf@idc.ac.il

Abstract. Millions of participants inhabit online virtual worlds such as SL[1] and engage in a wide range of activities, some of which require the performance of tedious tasks. Our goal is to develop a virtual proxy that would replace human participants in online virtual worlds. The proxy should be able to perform simple tasks on behalf of its owner, similar to the way the owner would have performed it. In this paper we focus on the challenge of social navigation. We use a data-driven approach based on recording human participants in the virtual environment; this training set, with a machine learning approach, is then used to control an agent in real time who is performing the same social task. We evaluate our method based on data collected from different participants.

Keywords: Navigation, behavioral cloning, imitation learning, proxy, data-driven approach.

1 Background

One of the old dreams of robotics researchers is to be able to send a robot to replace them in the less exciting chores in life. As people spend increasing amounts of time in online virtual worlds, our long-term goal is aimed towards replacing ourselves with virtual proxies that will be able to act on our behalf in virtual worlds. Specifically, in this paper we discuss how a virtual clone can learn to clone spatial navigation characteristics of its owner.

Social virtual environments are now becoming widely popular. Millions of people spend time in multi-user online games and virtual worlds, where they not only play but also engage in various social activities together. Of particular interest is SecondLife (SL), a generic platform that enables a virtual world constructed completely by its citizens. The citizens engage in a diverse range of activities (rather than gaming): socialization, romance, education, buying and selling of virtual goods, and many more [1]. There is scientific evidence that spatial behavior in virtual worlds has social aspects that are similar to real-world behavior at least to some extent [2, 3].

Bots in virtual worlds such as SL are avatars that are controlled by software rather than by a human operator, and have the appearance of any SL participant. Over the last few years we have developed the AVL bot platform, which was also used in the

[1] http://www.SecondLife.com

H. Högni Vilhjálmsson et al. (Eds.): IVA 2011, LNAI 6895, pp. 28–34, 2011.

study described in this paper. A proxy with basic functionality was already scheduled to appear in a real-world conference about conflict resolution in virtual worlds[2].

A data-driven approach where an agent learns from recorded expert data has been termed behavioral cloning by Anderson et al.[4]. They apply behavioral cloning in the context of an intelligent tutoring system embedded in flight simulators, which can adapt to specific students and their needs. As their goal is different than ours it is difficult to compare the technique or the results. Sammut and his colleagues have applied behavioral cloning to various problems, most notably capturing student models from flight simulation in order to learn to fly a plane [5, 6]. Their method is symbolic, which makes it difficult to compare with ours.

3 A Method for Social Navigation

When humans use space in a social context they are sensitive to the interpersonal distance and gaze direction in ways that are not completely understood by psychology, and are difficult to formalize. We focus on a simple example where a participant has to approach another participant in a socially acceptable way as if about to open conversation, in an unconfined open space. While this is a deliberately-selected simple problem, we note that it contains many of the challenges in modeling social spatial behavior. First, there are individual differences in the preferred proximity among people; this has been studied extensively in social psychology since the 1960s [7, 8], and more recently in virtual worlds [2, 3]. Another issue is how to approach a person who is standing with their back to you, and yet another issue is whether you walk straight, in a curved trajectory, or in a jittery trajectory. Gaze direction further interacts with all these factors. To our knowledge there is no attempt to automatically address this problem; the video games industry uses manually crafted trajectories with the aid of waypoints for non-player characters. Our approach is deliberately straightforward: we capture the navigation strategy of a virtual-world participant in the form of a training set of state-action couples. The participant's proxy can then use this training set by comparing its current state with states in the training set, and taking actions accordingly. The trajectories of the human participants are sampled over time. We convert each trajectory t into a set of state-action couples:

$$S_t = \{(s_1,a_1), \ldots, (s_k,a_k)\}; \quad (1)$$

For each participant we collect all such trajectory sets into a union set S of all the state-action couples in all the trajectories. Thus, our training set "forgets" which sample was taken from which trajectory. In this simple task the state can be modeled with only three parameters: ρ, θ, capturing the relative distance and direction of the agent from the target in polar coordinates, and α, capturing the relative difference in gaze angles between the agent and the target. The actions in our case are: move forward, move backward, rotate left, rotate right, move forward and rotate left, move forward and rotate right, move backwards and rotate left, and move backwards and rotate right.

[2] See video: http://www.youtube.com/watch?v=1R4Eo3UDT9U

In order to introduce some consistency into the proxy's behavior some memory capabilities were required. We append a fourth parameter – the previous action – to the state vector, which now becomes

$$s = (\rho_t,\ \theta_t,\ \alpha_t,\ a_{t-1}). \tag{2}$$

We construct a different model for each participant, from all the points in all the trajectories for that participant. The model can then be used to generate trajectories online as follows. The agent begins in a random initial state in terms of position and gaze direction. The current state of the world is translated into a state vector

$$s' = (\rho_t,\ \theta_t,\ \alpha_t,\ a_{t-1}) \tag{3}$$

as described above, and the training set is now queried for a state-action couple *(s, a)* s.t. $|s\text{-}s'|$ is minimal. We have evaluated several distance metrics, and concluded that the Manhattan metric

$$d(\bar{x},\bar{y}) = \sum_i |x_i - y_i| \tag{4}$$

provides a better separation among as compared with other metrics.

The state vector is comprised of different types of measurements. We normalized the distance, and for the rotation angles we used

$$\mathrm{d}(\theta,\theta') = \frac{1+\cos(|\theta-\theta'|)}{2} \tag{5}$$

for θ the agent rotation and θ' the training set rotation. Similarly, we used

$$\mathrm{d}(\alpha,\ \alpha') = \frac{1+\cos(|\alpha-\alpha'|)}{2} \tag{6}$$

for α the agent's gaze direction and α' the training set gaze direction.

The distance between two different actions is either 0 if the actions are equal or 1 otherwise. In order to adjust the contribution to that of the other state parameters this was weighted by a factor proportional to the mean contribution of the distance parameter.

We have also tested a weighted k-nearest neighbor approach; we find the set of k nearest state-action couples $\{(s_1,a_1),\dots,\{s_k,a_k)\}$ in the training set s.t. the distances between the state vectors s_i and the agent's state vector s is minimal. If $b_1,\dots,b_n$ are the actions that can be taken by the proxy agent, we choose b s.t.

$$b = \underset{b_j}{\arg\max} \sum_{i=1}^{k} e^{-|s-s_i|\delta(a_i,b_j)}, \quad \delta(a,b) = \begin{cases} 1, \\ 0, \end{cases} \tag{7}$$

The training set in our case includes up to approximately 50000 points per participant. This size can be processed in real time, but larger samples sizes might need to be reduced, e.g., by applying vector quantization (VQ) to the training set.

In practice, the samples we record from participants are sparse and cover only a small part of the state space, which results in cases where the nearest sample is very far; in such cases the proxy navigation may fail to reach the target within reasonable time. If there are not enough samples of input trajectories for a given participant we can generalize from the existing samples as follows. We choose *N* start points with uniform distribution around the target. For each start point we execute the proxy as in Section 3.3, generate a trajectory and check whether it has reached the target successfully within a predefined number of steps. If the target was reached we add the samples to the training set, if it was not reached we discard of the samples.

4 Evaluation

The trajectories generated by the proxy were first evaluated in a simulated abstract 2D environment; this is useful for debugging and manual inspection. Later on we have visualized the same trajectories in OpenSim: this is an open-source project similar to SL, which allows us to host the server locally, and hence avoid significant latency. A more systematic evaluation would require a display of the resulting behavior in the virtual environment, and subjective rating by naïve viewers; such as study is beyond the scope of this paper. A companion video illustrates some of the results[3].

Data was collected using an automated tool implemented in SL. Participants had to locate another avatar and approach it. Once the approach was completed they were teleported to another random point in the same region. In each new run the gaze directions of both the participant's avatar and the target were randomly modified.

The evaluation reported here is based on three participants: two females (ages 24 and 33) and one male (age 24). The female participants completed 98 and 20 trajectories correspondingly and will be referred to as F1 and F2, and the male subject completed 50 trajectories, and will be referred to as M1.Each trajectory includes a few hundred data points that are later on used in the training set as state-action vectors.

5 Results

Figure 1 shows two trajectories recorded from human participants as examples. First we show that individual differences in participants' trajectories may be detected. This is necessary to establish that our approach allows for maintaining these individual differences in the virtual proxies.

Manual inspection of the preferences of the different participants revealed clear differences. For example, Figure 2 shows some differences evident in the navigation tactics used by F2 and M1.

[3] See video http://www.youtube.com/watch?v=jmfGS1omgdk

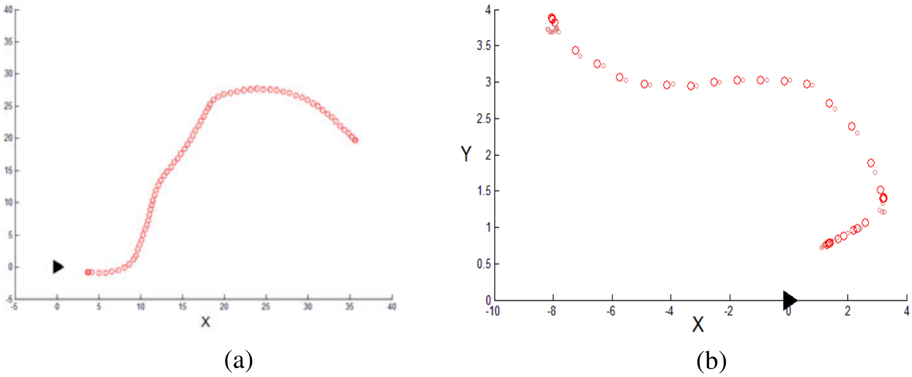

Fig. 1. Examples of recorded trajectories when the starting point is in front of the target (a) and behind it (b). The target is shown by a black arrow indicating its gaze direction.

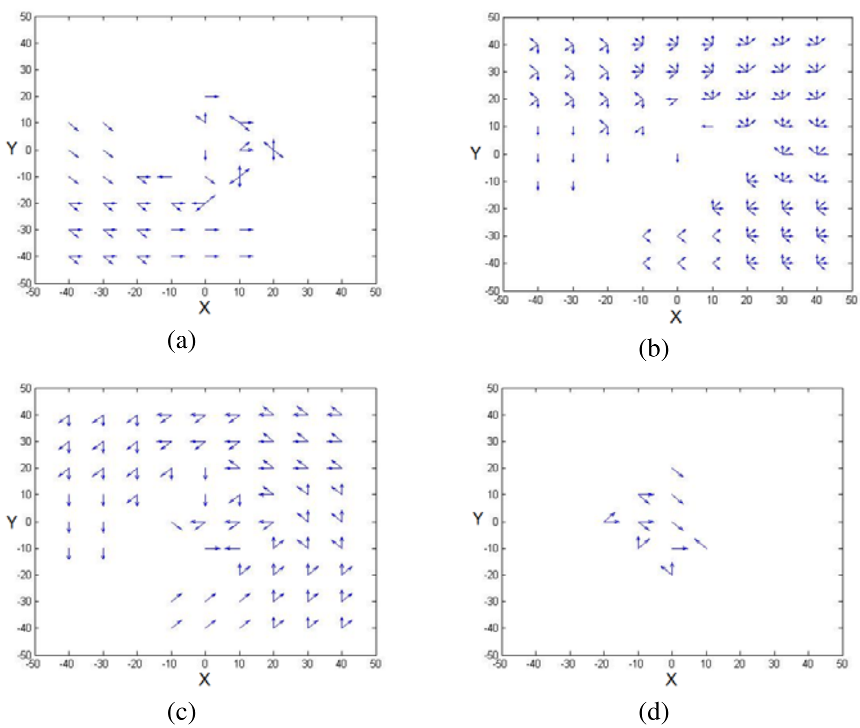

Fig. 2. A visualization of the navigation tactics used by different participants. The state space was quantized and normalized such that the target is always in (0,0) and facing to the right. For each position of the participant relative to the target we show an arrow if and only if the action indicated below was the most frequent action taken by participant when it is facing in the arrow's direction. (a) Participant F2 rotating left. (b) Participant M1 rotating left. (c) Participant F2 moving forward and rotating left. (d) Participant M1 moving forward and rotating left.

The quality of the trajectories generated by the proxy depends on the number of samples. First, we measured the success rate of each proxy by checking 324 trajectories from uniformly sampled starting points in terms of initial conditions of ρ, θ, and α. For M1 and F1, for which we had 50 and 98 trajectories correspondingly, the success rate was 92% and 88% correspondingly. In some occasions the proxy would fail from reaching the social distance from the target within a given time. In order to obtain a robust proxy behavior we deploy generalization as explained in Section 3.4. After running a few dozen random trajectories per model the proxy reaches a 100% success rate on our test sample. The proxy for participant F2, which was based on a small number of input trajectories (20), had a lower rate of successful trajectories (70%). In addition, we manually inspected the trajectories generated by the three different proxies. We selected random start points, and had all three proxies start from each start point, using the training set without generalization as a basis for their operation. Figure 5 shows some examples. We note that the trajectories for F1 and M1 seem both acceptable and believable, while the trajectories from F2, with the small training set, were often unrealistically intricate (see Figure 3), and, worse, tended to get within unacceptable proximity of the target (recall that in social situations people always keep some minimum interpersonal distance). Also, note that the algorithm is deterministic; the reason that each proxy generates a different trajectory is that each is based on a different training set. This preserves our intention that the proxies will have an individual behavior, based on their owner, yet consistent.

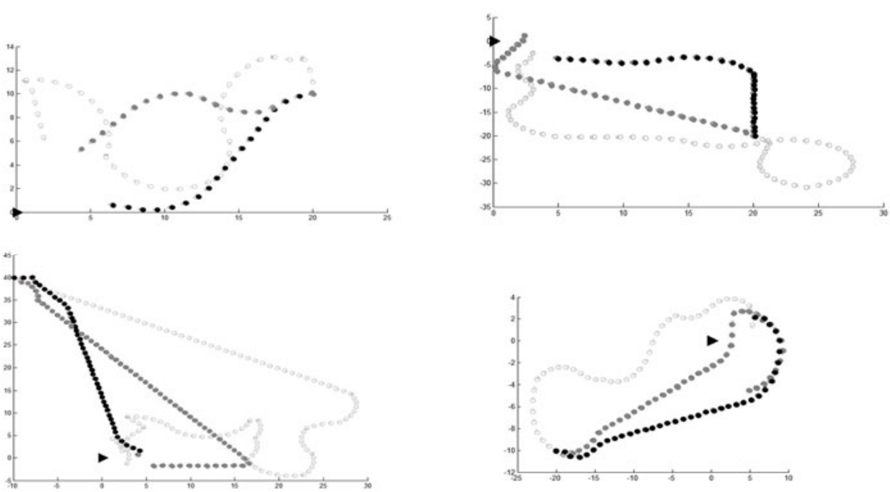

Fig. 3. Sample trajectories of three proxies modeled from three participants starting at the same conditions: M1 (Black), F2 (Empty), and F1 (Gray)

7 Discussion

This paper describes our work in progress on virtual proxies in virtual worlds. These are different than other type of virtual agents in that they are intended to be clones of individual users. We show that using a simple k-nearest neighbor approach can be a good starting point for a virtual clone, when tested on a simple instance of the problem of social navigation. In the future, we hope to extend this method and evaluate its performance in larger-scale challenges. Naturally, the method will be put to the test in a more realistic space, with obstacles and additional agents that move around and change their gaze. In addition, we are working on extending this method to data tracked from a full body rather than just 2D positions. Clearly, more complex state representations may be required, and with them more advanced methods for generalization and abstraction.

Acknowledgements. This project is supported by EU FP7 project 248620 Beaming. The authors would like to thank Daniel Korn for implementing the record & replay facilities in OpenSim that were used for the evaluation of trajectories. The authors would also like to thank Chani Vaintraub for useful comments.

References

1. Boellstorff, T.: Coming of Age in Second Life. Princeton University Press, Princeton (2008)
2. Friedman, D., Steed, A., Slater, M.: Spatial Social Behavior in Second Life. In: Proc. Intelligent Virtual Agent, pp. 252–263 (2007)
3. Yee, N., Bailenson, J.N., Urbanek, M., Chang, F., Merget, D.: The unbearable likeness of being digital; The persistence of nonverbal social norms in online virtual environments. Cyberpsychology and Behavior 10, 115–121 (2007)
4. Anderson, C.W., Draper, B.A., Peterson, D.A.: Behavioral Cloning of Student Pilots with Modular Neural Networks. In: Proceedings of the Seventeenth International Conference on Machine Learning, pp. 25–32 (2007)
5. Isaac, A., Sammut, C.: Goal-directed Learning to Fly. In: ICML, pp. 258–265 (2003)
6. Morales, E.F., Sammut, C.: Learning to fly by combining reinforcement learning with behavioural cloning. In: Proceedings of the Twenty-first International Conference on Machine Learning, Banff, Alberta, Canada, pp. 76–83 (2004)
7. Hall, E.T.: The Hidden Dimension. Doubleday, New York (1966)
8. Hall, E.T.: The Silent Language. Doubleday, New York (1959)