Learning Crowd Behavior

Panayiotis Charalambous
University of Cyprus
totis@cs.ucy.ac.cy

Yiorgos Chrysanthou
University of Cyprus
yiorgos@cs.ucy.ac.cy

Abstract

We present here our ongoing work on learning crowd behavior. The steering behavior of crowds from various video sources is tracked and databases of examples are generated. These examples contain various stimuli (metrics) that could affect the persons behavior. These databases are used to learn rules for crowd steering in an agent based framework using regression algorithms and more specifically, decision trees. Various simulations are run and the statistics measured at the simulation stage are compared to those of the original video to determine which stimuli affect an agents behavior the most.
Keywords: crowd simulation, machine learning, computer animation

1 Introduction

Rule based crowd simulation algorithms generally use empirically selected stimuli to determine when and how an agent should react. Stimuli in general consist of events in the environment that influence behavior such as the visual perception of an agent, its neighbor's state and actions and nearby obstacles. In order to simulate a particular crowd behavior such as leader following, path following etc., a set of rules is defined that takes into account these particular sensory inputs. Usually, researchers select beforehand what should be taken as input for these rules, such as the center of mass of the agents temporal crowd, their moving direction etc. Our purpose in this work is to evaluate these metrics using real world input and determine which are the most important. Obviously, not all of the possible input stimuli can be tested, but we hope with this work to cover a large portion. Also, in this work we only consider steering behaviors, but higher level social behavior patterns are our next target.

2 Previous Work

Although methods for animating a single human character advance in leaps and bounds, automatically animating a believable pedestrian crowd remains a challenge. A lot of algorithms and techniques have been developed that fall into two broad categories: rule based [1] and data driven [2, 3]. Most of these approaches model the behavior of a single virtual character in accordance to its environment. Some interesting work on interacting small crowds can be seen in [4, 5].

Rule-based methods can produce realistic results for specific situations. In order to do so, many finely tuned specific rules are required something that is difficult to define and often requires a lot of trial and error. Given a new situation, a new set of rules needs to be defined. Reynolds [6] proposed one of the earliest rule-based approaches for animal crowds. He later expanded this to include additional behaviors such as seek, pursue and evade [1].

In [2] and [3], videos from real captured data are used o generate examples of crowd behavior. Individual characters are recognized on the videos, and their trajectories are tracked. This approach generates a number of example situations and actions for the characters. An example database is created, and is queried at simulation time to get the best action. In [4], videos of martial artists fighting each other are recorded and used to generate believable fight sequences between two characters. This approach creates very realistic results but requires a lot of examples to cover the whole spectrum and is difficult to apply in very involved situations.

Lately, researchers started working on evaluating crowd behaviors, such as the work done by [7]. In this work, various metrics such as number of collisions, time and effort efficiency, are used for evaluation as part of a benchmark framework.

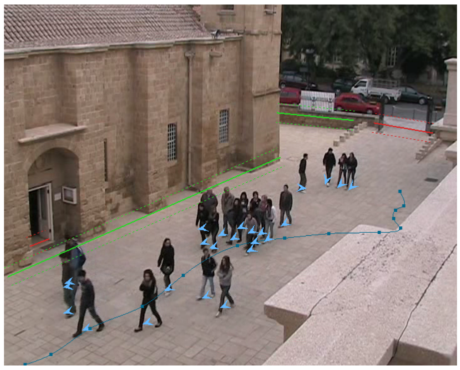

Figure 1: Tracking crowds from video.

3 Learning Crowd Behavior

3.1 Overview

In our still ongoing work, some experiments are being done to determine which of a set of sensory inputs affect an agent's behavior the most and how. Videos of real and simulated crowds (from existing rule based systems such as [1]) were recorded and the steering behavior (trajectories) of the people in them was tracked using our own manual crowd tracker. These trajectories were sampled and at each sampled point, the surrounding stimuli of each individual and his current speed and direction were recorded. These stimuli consist of statistics of nearby people's steering behavior: speed, acceleration, orientations etc. These samples were stored in a database as a set of examples. In addition, various statistics regarding speed, rotations, flow, collisions, people distribution, collisions etc. were calculated for the entire duration of the video and used for the evaluation of the experiments.

The database of examples is used as the input of a regression algorithm (we started with decision trees, we plan on using some others also) that selects the best action (change of speed and direction) depending on new input that might not be part of the database. A series of simulations were run and in each of them different sets of time varying stimuli were selected to generate new inputs to the regression algorithm. This regression algorithm returns the appropriate action to be taken (speed change and rotation).

The same statistics that are calculated for the original input videos are recorded and the two are compared. The best matches were used to refine the original stimuli to achieve better results. The result of this study is twofold: (a) to identify the most important factors that affect good steering behaviors and therefore help in the design of better rule based systems and (b) a flexible data driven simulation framework.

In the following sections, a more detailed look at the whole procedure is described.

3.2 Tracking

Various real life crowds were recorded with a simple consumer grade camera from an elevated position (Figure 1). Autodesk's ImageModeler was used to estimate the configuration of the real world camera. Using our own manual crowd tracker, the trajectories of all the people in the crowd were tracked and their coordinates were transformed from pixel space to world space using the previously estimated camera configuration. In addition to videos, some simulations from other rule based systems, like OpenSteer [8], were run and were used as proof of concept.

3.3 Statistics Gathering

A set of statistics that characterize the crowd behavior in each video is generated. These statistics are calculated per agent, per frame and for the entire duration of the video (Table 1). In their simplest form, they include speed, rotation and collision statistics. For example his speed profile over time is recorded. From this speed profile average and variance values for speed are estimated and recorded. Also, a histogram of the all the people's speed is generated which describes the input video.

Statistics	Average	Variance
Speed	y	y
Acceleration	y	y
Rotation Speed	y	y
Distance Traveled	-	-
Displacement	-	-
...	...	...
Neighbors Count	y	y

Table 1: Some statistics that are gathered per input person.

These statistics are used for two purposes:

- for crowd generation: these information can provide the simulator with number of agents, starting speeds, etc.
- for behavior evaluation and help in the selection of the best input stimuli

It's worth mentioning here, that for different videos of crowds, different statistics are generated and therefore different simulations are generated. For example, statistics from a video of sparse crowds in a market will be vastly different from a video of a parade. It is expected, that for each input case, a different set of input stimuli will be selected as the best but at the same time there will be an overlap between them.

3.4 Example Generation

Using as input the trajectories of all the people in the videos, a series of state-action pairs are generated. A state consists of a set of time varying stimuli, mostly steering behavior of nearby agents and an action is a pair of acceleration and angular change applied on the agent.

The agent perceives the environment using a circular region around itself (Figure 2). This circular region can be divided into a series of arcs using the local coordinate system of the agent. These arcs are not necessarily of the same size. Whenever the system requests that a new example should be generated, each agent finds which of the nearby agents at the moment are geometrically inside each arc. Using information held by each agent, a series of metrics are generated that correspond to the state of the agent. More specifically, each agent can produce metrics for itself and per arc (neighbors). These metrics are discussed in the following paragraphs.

Self Values

Since each agent knows its history, it can produce various metrics that characterize its own behavior, such as total distance traveled during a time window, displacement, average and variance values for speed, acceleration, rotation etc. Also, metrics such as distance and angle from a predetermined goal can be used to learn how an agent moves toward a goal. Since it is difficult to know beforehand from a video the goal of a person, its final position can be assumed as the goal.

Neighbor Values

Using the neighbors that lie in each arc, the agent can calculate some statistics for each one. For example, the average speed of all the agents in that region at that particular moment, their center of mass, distance weighted speed, formation area (i.e. the minimum circle enclosing all of these agents divided by the arc area) etc.

Since the steering behavior of a person does not rely only on the current stimuli, the previous mentioned metrics are extended to account for time. For example, the time varying average speed of these agents is the sum of the average speeds of the agents per input frame over a time window. This time window is usually less than 2 seconds, but can easily be modified for experiment purposes. Some experiments using only the closest neighbor were done also, since this is a very common approach used by researchers.

We note here that the number of arcs should be kept as small as possible since the regression algorithm would need lots of data to learn because of the curse of dimensionality.

At the time of writing, the agent can track other moving agents, but it can easily be extended to handle static objects.

4 Simulation and Evaluation

Using as input the statistics that were gathered from the videos, a number of agents is generated with characteristics matching those in the original videos. Using the exact same metrics and

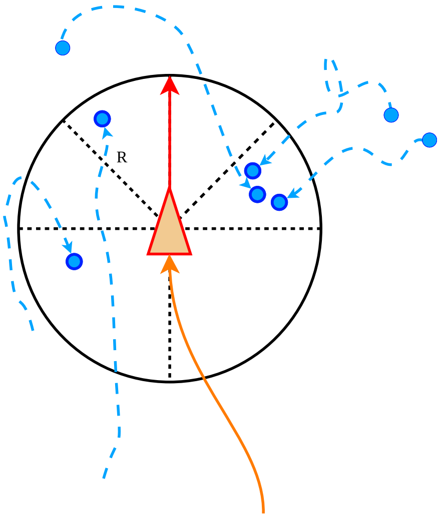

Figure 2: Agent's perception

perception that were used in the learning phase, and the already learned decision tree, the simulation is run. At runtime, the agent perceives the environment and estimates this metrics. These metrics are then given as input to the decision tree and new acceleration and rotation values are returned and applied on the agent.

The same statistics that were gathered from the videos are collected here also. At the time of writing this is still in the development phase. By having the statistics (basically the histograms that were mentioned in section 3.2) from the original videos and the statistics from the simulations the two are compared (using k-nearest) and the closest match indicates which stimuli generate the closest match to the original video. This does not mean that the same stimuli can generate the best results for all the input videos. If this is the case, then the same stimuli can be used for all crowd simulation algorithms, but this is not expected.

The main idea here is to generate a number of simulations with different metrics, perception, look back time and then determine for each video which of these are the most important.

5 Conclusions

Concluding, we would like to mention here that this method can be used to learn social behavior and not only steering behaviors. By using higher level stimuli, such as the people's social behavior (talking, waving, shouting) we believe that some realistic social behavior patterns will emerge. Also, some metrics that are mentioned in other papers (like [7]), could be used in out experiments.

References

[1] Craig Reynolds. Steering behaviors for autonomous characters. In *Game Developers Conference 1999*, 1999.

[2] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. *Computer Graphics Forum (Proceedings of Eurographics)*, 26(3):655–664, September 2007.

[3] Kang Hoon Lee, Myung Geol Choi, Qyoun Hong, and Jehee Lee. Group behavior from video: a data-driven approach to crowd simulation. In *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 109–118, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

[4] Taesoo Kwon, Young-Sang Cho, Sang Il Park, and Sung Yong Shin. Two-character motion analysis and synthesis. *Visualization and Computer Graphics, IEEE Transactions on*, 14(3):707–720, May-June 2008.

[5] Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion graphs. *ACM Trans. Graph.*, 21(3):473–482, 2002.

[6] Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34, New York, NY, USA, 1987. ACM.

[7] Shawn Singh, Mubbasir Kapadia, Petros Faloutsos, and Glenn Reinman. Steerbench: a benchmark suite for evaluating steering behaviors. *Comput. Animat. Virtual Worlds*, 20(5–6):533–548, 2009.

[8] Craig Reynolds. Opensteer, October 2004. `http://opensteer.sourceforge.net`.