Fuzzy Finite State Machines in Crowd Simulation.

Leonel Antonio Toledo Diaz*, Isaac Rudomin*,
*Barcelona Supercomputing Center, Barcelona, Spain
E-mail: {leonel.toledo1, isaac.rudomin}@bsc.es

Fig. 1. An example of Barcelona City. created using Unity's 3d plug -in

***Keywords*—Crowd Simulation, GPU, Fuzzy Logic, FSM.**

I. Extended Abstract

Large crowds of pedestrians are a common phenomenon in big cities, this field of research studies and reproduces this kind of situations in virtual environments. Complex interaction within the agents is desired but it requires modeling their internal state. Through the internal state of the agents we can include reactive behaviors with multiple and changing objectives. The individual (personality) and social (group membership) properties of the agent can also be modified due to external or internal changes, such as environment, communication and mood. The inclusion of this features, and the internal state of the agents, change the behavior of the crowd. One of the main problems when simulating a crowd is to define individual traits for all the agents within the simulation. It is possible to define parameters that individually describe each of the characters involved in the simulation, however this approach turns to be unfeasible as the crowd grows bigger. We propose a method that combines finite state machines with fuzzy logic to represent concepts such as fast or slow which its definition may vary from agent to agent.

A. Fuzzy Finite state machines for crowds in urban environments.

We present a method for simulating crowds in urban environments, to create these scenarios, we use Unitys WRLD plug-in as shown in figure 1. We use the city of Barcelona in our simulation. In a similar fashion as van Essens [1] and Thomnsen [2], where 3D maps are produced using the most relevant features.

Figure 2 shows how we create the urban environment using the previously discussed techniques, we use WRLD3D plug-in which gives us detailed information about geographic locations, in this case we construct the simulation using Barcelona as a reference. Once the environment is created we incorporate the crowd into the simulation, our goal is to make the simulations as complex as possible, to reach that goal we consider two different techniques that we combine; first, we collect real data from GPS traces that describe routes that pedestrians take within the city, this trace includes information about the latitude, longitude, elevation and the time when the sample was taken, and our agents can follow the given routes and be animated accordingly. Second, we consider autonomous characters that can navigate the environment. We include simple behaviors such as patrolling, following, avoiding obstacles or pedestrians just to state a few. This behavior is controlled by finite state machines in which each agent has the freedom to decide how to change states accordingly. Nevertheless, pedestrian behavior cannot be modeled realistically using deterministic models, thats why we incorporate fuzzy logic into the simulation, this way we can create different profiles for each character, and work with concepts such as fast or slow inside the simulation, what is true for an agent might not work in the same way for other. To decide whether a character is moving fast or slow and simulate properly we use a shared library of parameters that all characters inherit from, we can manually tweak each of the variables for any given character or randomly assign values. This allows us to create two different profiles for all the elements in the simulation, the first profile is focused in information such as vision range, maximum speed, weight, turn speed, to state some. The second profile is oriented towards how each character understands fuzzy concepts such as fast or slow, this way even if the members of the crowd have the same physical profile they might behave very different according to their fuzzy parameters. One of the main advantages of this method is that all agents have access to this knowledge and without any changes to the script we can achieve a lot of variety in the crowd behavior.

B. Results

Using this approach, we have been able to simulate both complex urban environments, as well as agent behaviour within the city. In terms of performance, we have achieved a balance between render and simulation, for instance we are capable of simulating one thousand characters in scenes with a little more than 4 million polygons and 3 million vertices. For a scene similar as the one shown in figure 2, we use 1115 draw calls and 39 batched draw calls, 300 megabytes of RAM memory is required and 112 megabytes of video memory. Each frame takes about 28 milliseconds to render, which allow us to have simulation at interactive frame rates (more than 30 fps). On average, each frame has 18,000 objects in viewable space and a total of 43000 total objects in the whole simulation. This outperforms known techniques such as [3] Millans comparison between impostors and point based render, in terms of memory consumption.

Fig. 2. A crowd simulated within the city in Unity

C. Conclusions and Future Work

We present a robust approach for urban crowd simulation that can run at interactive frame-rates. Which is powerful enough to handle large environments in real time without compromising visual quality and the simulation of individual behaviors. At the current stage the system has proven to be successful in achieving meaningful diversity in terms of how characters react in different situation. Nevertheless, this is not enough, significant efforts must be done to further optimize the system, for instance LOD techniques are desirable to further expand the size of the crowd [4]. Visual variety is also considered, at this point, every character in the simulation uses the same mesh.

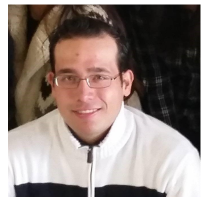
REFERENCES

[1] R. van Essen, *Maps Get Real: Digital Maps evolving from mathematical line graphs to virtual reality models*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 3–18.

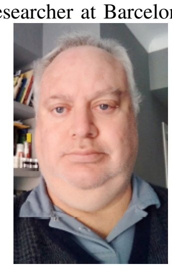
[2] A. Thomsen, M. Breunig, E. Butwilowski, and B. Broscheit, *Modelling and Managing Topology in 3D Geoinformation Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 229–246.

[3] I. Rudomín, B. Hernández, O. De Gyves, L. Toledo, I. Rivalcoba, and S. Ruiz, "GPU Generation of Large Varied Animated Crowds," *Computación y Sistemas (CyS)*, vol. 17, no. 3 (Special Issue) Supercomputing: Applications and Technologies, pp. 365–380, 2013.

[4] L. Toledo, O. De Gyves, I. Rivalcoba, and I. Rudomn, "Hierarchical Level of Detail for Varied Animated Crowds," 2014.



Leonel Toledo recieved his Ph.D from Instituto Tecnolgico de Estudios Superiores de Monterrey Campus Estado de Mxico in 2014, where he was a full-time professor from 2012 to 2014. He was an assistant professor and researcher and has devoted most of his research work to crowd simulation and visualization optimization. He has worked at the Barcelona Supercomputing Center using general purpose graphics processors for high performance graphics. His thesis work was in Level of detail used to create varied animated crowds. Currently he is a researcher at Barcelona Supercomputer Center.



] Isaac Rudomin is a senior researcher at the Barcelona Supercomputer Center, which he joined in 2012. His focus is on crowd rendering and simulation including generating, simulating, animating, and rendering large and varied crowds using GPUs in consumer-level machines and in HPC heterogeneous clusters with GPUs. Previously, Isaac was on the faculty at Tecnologico de Monterrey Campus Estado de Mexico (from 1990 to 2012). He finished his Ph.D. at the University of Pennsylvania under Norman Badler on the topic of cloth modeling.