AN AGENT-BASED ARCHITECTURE FOR LARGE VIRTUAL LANDSCAPES

Bruno Fanini

Department of Computer Science, University of Bologna (Italy) – bruno.fanini@gmail.com

KEY WORDS: Agent-Based Model, Crowd Simulation, Artificial_Intelligence, Cultural Heritage, Real-time, Algorithms

ABSTRACT:

This paper presents an agent-based architecture for real-time crowd simulation into large reconstructed landscapes and virtual worlds, with the purpose of adding dynamic entities and enhance environment comprehension through a user-crowd interaction model. The system is based on the open-source *OpenSceneGraph* library adding to the existing scene an artificial life layer, provided with a set of customizable rules and parameters to influence social behaviour of agents. Separate modules enable the configuration of areas affordance, geo-morphology impact, attitudes and hotspots able to magnetically attract agents towards valuable buildings, areas or resources in terms of cultural heritage and automatically evolve smart paths using pheromone trails and simulating some basic human movement aspects. The rendering of large number of agents, relationships' evolution, collisions and social influences are addressed with level-of-detail techniques, scene-graphs algorithms and spatial partitioning of the crowd allowing a scalable approach in different scenarios with paging support for large datasets, locally o remotely located. Informative contents and arbitrary media are stored inside hotspots, collected by agents during their exploration and reproduced on demand when the user interacts with a nearby local crowd. Evolutionary aspect of the model thus leads to an adaptive moving *informative crowd*, plunging into the virtual world dynamic entities to be used as tour-guides enabling the user navigation itself to be softly linked with, or modify surrounding environment for historical reconstructions, didactic purposes or demos.

1. INTRODUCTION

Reconstructed landscapes within Virtual Heritage usually focus on 3D visualization of buildings such as monuments, old towns and architectural representations. The general approach is to provide the user with a friendly interface up to explore and navigate the virtual world, this is the case of recent projects like Virtual Rome (Pescarin et al., 2009). Interactive navigation is one of the main factors for direct impact on the final user and on the quality and the level of ancient landscape comprehension. The lack of visual clues and content rarefaction in large reconstructed scenarios have been discussed in other projects (Fanini, 2009) and solutions have been proposed to address disorientation issues and unsatisfying navigation sessions. On the other hand, the relatively old lack of visual realism is nowadays fulfilled hardware-side with recent 3D graphics technologies and software-side with advanced frameworks to organize and manage very complex scenes, optimizing the load on the final pipeline and on memory footprint. The final result can often reach photo-realistic levels but most of the times there is a lack of dynamic elements, such as crowds or animated contents (Ulicny, Thalmann, 2002). The presence of animated entities, such as a real-time 3D crowd that populate the virtual world, has the main goal of providing a switchable and highly parametrisable simulated life layer, that reacts to user actions and enhances the overall landscape comprehension with interactive behaviours. The system aims to consider virtual environments of any dimension, from small local areas to huge paged territories, although the main focus is on large datasets, locally or remotely stored: these goals introduced large-scale issues to deal with and the underlying needs to study various and efficient solutions along with the study of solid architecture model.

Agent-based Models (usually ABMs) are representations of systems composed by several interacting actors (agents) all of whom can take actions and make decisions within the environment: they are discrete independent entities with the fundamental capability to make decisions. The attributes of agents can be continuous measurements like age or wealth, or discrete categories. The behaviour of these entities can range from primitive reactive decisions to very complex adaptive Artificial Intelligence or social attitudes. Approaches can vary from completely reactive, like performing an action when triggered to do so by some external stimulus (e.g., actions of another agent or the user itself to be illustrated later) to goal-directed, through seeking of a particular goal. In applications of ABMs to social processes, agents represents people and inter-agent relationships represent social interaction processes (Macal, North, 2008) with the goal of credibly modeling these features on a certain reasonable level of abstraction. In this perspective, *movement* is the basic feature commonly implemented in an Agent-based crowd: to implement a valid and realistic model for the presented architecture, recent studies (Murrieta Flores, 2009) have been considered to properly implement A.I. factors and simulate the basics of human movement, influenced by many variables and parameters. A division was made into two main groups: the first one contains independent variables that can include terrain geomorphology, physics, effort, the second one contains social factors that involve local avoidance, collisions, data sharing and transmission, hotspot routes, flock cohesion and other variables that are dealing with a social radius among agents.

The main question at this point of the implementation was: which of these factors are ”*mechanizable*” and what level of realism are we going to reach without loosing efficiency?

2. ARCHITECTURE DESIGN

The core architecture is modeled to be scalable and highly parameterizable. Being the main focus on large landscapes and very complex scenarios, the adopted framework was *OpenSceneGraph*: a highly efficient, portable and open-source library that is based on scene organization concepts and optimized workload using graphs. These concepts provides a rock solid base to develop an efficient crowd simulation and related scene-graph algorithms to maximize efficiency of the whole system and provide the user with a final scalable architecture. Large crowd simulation systems in facts, represent an interesting and complex research domain that throws up several challenges (Kota, Bansal, Karlapalem, 2007), including architecture approach, crowds phenomena to model, scalability issues and efficiency on large-scale environments.

2.1 OpenSceneGraph

OpenSceneGraph is an open-source high performance 3D graphics tool-kit, used by application developers in fields such as visual simulation, games, virtual reality, scientific visualization and modelling. Entirely written in standard C++ and OpenGL, it runs on all Windows platforms, OSX, GNU/Linux, IRIX, Solaris, HP-Ux, AIX and FreeBSD operating systems. It is now well established as the world leading scene graph technology, used widely in the vis-sim, space, scientific, oil-gas, games and virtual reality industries. In the virtual heritage context, many projects have been developed using this tool-kit for its flexibility and scalability: there is no specific support for crowd simulations, thus the presented system wants to be a proposal of a base architecture for agent-based crowds adaptable onto large landscapes, with the main focus on cultural heritage application.

2.2 The Agent

The basic object within crowd simulation is the agent, this unit encapsulates a series of rules that can be customized impacting factors and variables to produce certain behaviours: this single unit has independent parameters and an initial setup that let the structure evolve during time.

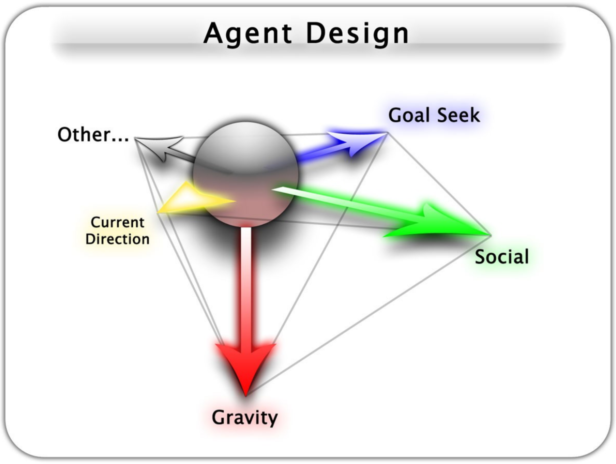

Figure 1. Agent design and weighted influences.

As shown in the scheme (Fig.1) a single particle has many rules bundled into a set of vectors and weights that can evolve over time: each rule can be individually weighted. Single values should be considered how hard the agent will try to respect a given rule. If the agent meets more than one conflicting condition at the same time, it will try to fulfill all the rules according to the respective weight of each. The model adopted to render agents is an extension of the *Boid Model* (Reynolds, 1987), this classical model is based on three rules:

1. *Avoidance*: Steer to avoid local collisions
2. *Alignment*: Steer towards average direction and speed of neighbors
3. *Cohesion*: Boids tends to gather nearby the local crowd center

Additional rules, behaviours, evolving memory and environment maps have been developed to extend and enhance the whole design to be automatically adaptable to a wide set of scenarios and let the agents evolve in a smart way. A given agent will try as much as it can to comply with each of the rules he is given to finally compute its direction and behaviour.

2.3 The System

The main Model System controls and manages all the agents (the entire crowd), using an input set of several factors, rules and spatially localized behaviours introduced before. The whole structure can accept any complex OSG paged model or simple scene-graph representing the *virtual world*: a graph locally or remotely stored, able to be modified or animated during the simulation (Fig.2).

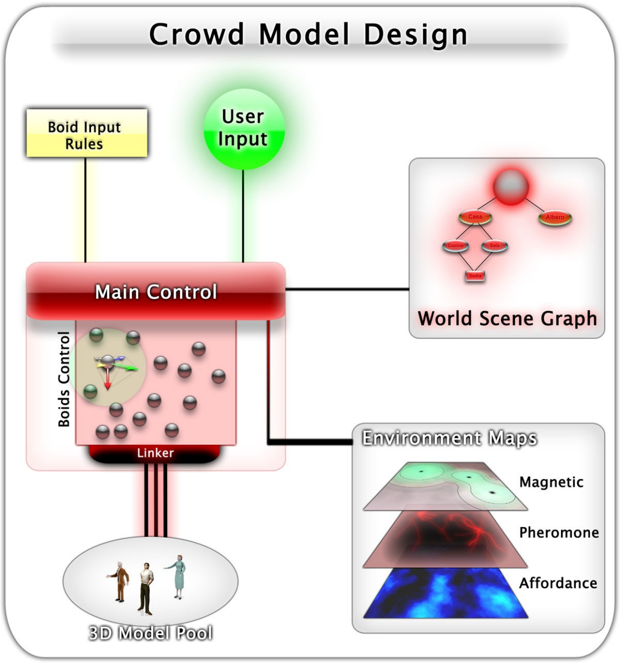

Figure 2. System Design.

The main unit handles a user-defined number of agents, providing a local or remote *pool* that consists of a simple directory containing different typologies (i.e. 3D models) to be automatically instantiated into the crowd, OpenSceneGraph plugins allow loading several formats (*3ds, dxf, obj* and others) apart from native OSG formats (complete list: http://www.openscenegraph.org/projects/osg/wiki/Support/UserGuides/Plugins).
Given the strict link between crowd engine and interactive user navigation upon which is based the main concept of the system, the user has also an influencing role: artificial intelligence of the agents evolves also depending on user input actions, position and attitudes to be shown later.

3. ENVIRONMENT MAPS

The architecture provides a conceptual unit that consists of a set of *environment maps* acting as layers connected with the landscape and simulated crowd: some of them are used to track particular evolutions of the model while others are directly provided by the user. They are used to define valuable hotspots, their importance and goals (magnetic map), to furnish terrain affordances layer, or to develop automatic paths (pheromone map) with the general purpose of adding constraints and influence artificial intelligence behaviour.

3.1 Affordance Map

There are usually different typologies of terrain in a landscape context: these differences influence the speed and the energy used by an agent to walk through a certain surface. For instance, a complex environment often contains vegetation areas and that coverage impacts human agility and the cost of choosing certain paths in the reconstructed world: this map provides a way to define how the terrain type affects the movement in a given spatial point or even to reproduce many terrain conditions that are able to influence agents.

3.2 Magnetic Map

Considering a virtual heritage context and content rarefaction issue introduced before, many solutions have been proposed: the basic concept mostly derives from another project study (Fanini, 2009) where sets of points are able to smoothly influence user navigation up to increase the overall quality of exploration. As a result from many tests, an interactive user exploring a large virtual world tends to focus his attention on dynamic elements and thus attitude to navigate towards certain areas. In this scenario, a magnetic net can be defined and customized to smoothly influence the artificial intelligence of the entities towards nearby important hotspots in terms of cultural heritage value, and evolve proper internal networks during the session. The purpose of this map is to provide a layer upon the model that is easily integrable with underlying multi-informative virtual world, consisting of a set of hotspots defining values for monuments, buildings, areas or everything that is conceptually critical to understand and properly comprehend the current landscape. An attractor node of the net has an internal repulsive kernel to avoid "black-hole" collapsing trap and releases a magnetic force depending on radius, strength and distance from the agent (Fig.3).

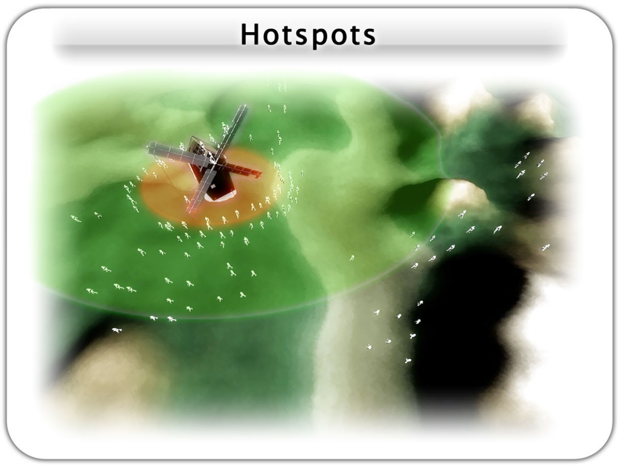

Figure 3. Agents visiting an hotspot (green radius) with repulsion of internal radius (red).

The hotspot node contains or can be loaded with information or media about the current area or position with a cultural relevance. On the other hand, nodes of the net can be also defined as repulsive allowing the customization of constraints for artificial intelligences up to avoid certain areas (such as poor-detailed areas for example) or to be combined with others nodes. Hotspots can also contain coordinates about the next node to visit: this allows definition of complete walk-paths to be gently pushed into agents' behaviour. In this context, when a agent has its visit terminated, it has information about the next hotspot that will influence its behaviour according to the other internal rules and relative weights and will also influence other surrounding idle agents with no target: information will be shared and they will be socially influenced in a certain measure to pursue the same target. Every node of the net can be moved or modified at run-time: the system allows complete customization of the configuration and immediate reaction of the crowd model, useful for demo purposes or within historical reconstruction contexts and teaching purposes.

3.3 Pheromone Map

In Nature, the behavior of some insects is to initially wander randomly and upon finding valuable resources return to their home while laying down pheromone trails. If the other mates find such path, they tend to follow the trail, reinforcing it as pheromone evaporates reducing the attractive strength (Dorigo, Gambardella, 97). A general path-finding algorithm was developed and tested from this case study and is currently implemented in this architecture. The main model keeps a structure that maps agents movements within environment and let it evolve with time: resources are mapped into the hotspots defined in the magnetic-map and the more it takes for an agent to travel down a inter-hotspot path and back again, the more time the pheromone trails have to evaporate. A short path gets marched faster and in that case pheromone density remains high thus evolving the agent paths towards optimal solutions. The main concept is using the virtual environment as a communication channel: the system can initialize the Pheromone map from scratch and let the boids evolve smart paths. The tool can also apply to the rendering of overlays representing agent-paths generated by the input rules, factors and parameters given to the model and study discordances or similarities with human-developed paths in ancient reconstructed landscapes.

4. CROWD AND USER INTERACTION

The single agent unit has been developed with the possibility to attach a proper reaction that can be triggered on user interaction or attitude. In a virtual heritage context, the study focused on the importance of hotspots such as archaeological sites, reconstructed monuments or even entire valuable areas scattered across the virtual landscape. The module is tightly joined with magnetic map and differentiation of agent typologies up to provide a "*moving informative crowd*": every agent contains collected information about the different hotspots or areas visited by far and every agent typology has different information context.

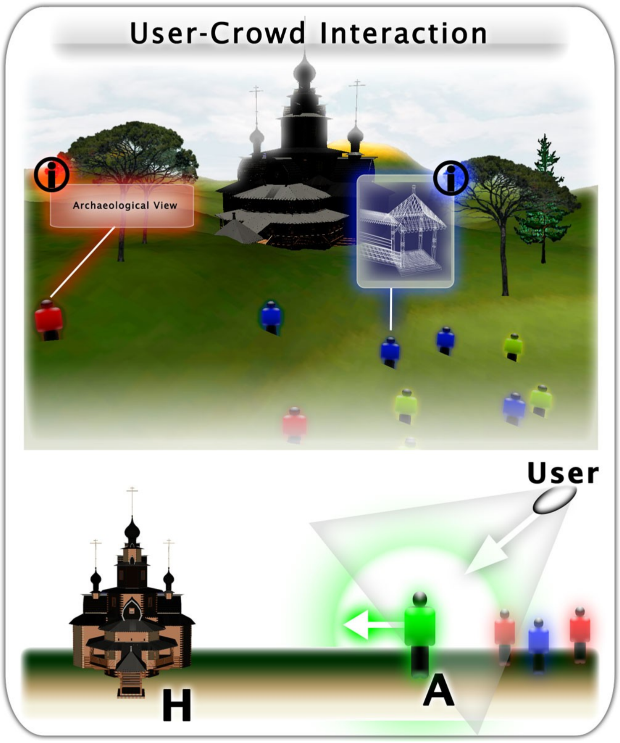

Figure 4. Example with simplified agents providing user with different informative contexts such as archaeological (red) and architectural (blue). Users can link their navigation to an agent (A) and follow it towards hotspots (H) or agent-developed paths.

A typical scenario is to define different ontologies such as archaeologist, architect, geologist and others to provide user with different information about the same hotspot when he clicks or interacts with a agent visiting the area (Fig.4). A typical pop-up web page or any arbitrary media can be attached as well as a direct action on surrounding environment. Agents' behaviour is also influenced with parametrizable strength by user movements around the virtual landscape: this feature allows the crowd to track and follow user exploration and assist him with hotspot information when in proximity of cultural valuable areas defined in the magnetic-map, reducing crowd dispersion onto large landscapes. Agent is thus studied to collect hotspot information and reproduce contents when user interaction occurs and to provide user a guide: an implemented feature permits user navigation to be softly linked with the agent, allowing automated tours onto the paths developed by agents themselves.

5. ALGORITHMS DISCUSSION AND RESULTS

System efficiency and performance are among the final goals: within large agent-based crowd simulations there are a number of problems and scalability issues to be handled and possibly solved. One of the main goals of the project is to maintain abstraction about environment and geometries, thus no pre-computation algorithms for trajectories were implemented or static environment assumptions: direct and real-time manipulation of terrain, virtual world and magnetic hotspots is allowed. OpenSceneGraph provides a database pager that addresses some of the issues about visualization of large terrain datasets: the virtual world is organised into a quad-tree hierarchical structure and tiles are loaded on demand into the graph and unmounted when not necessary to maintain a good memory footprint. The main issues about the crowd development regarded the large amount of intersections agents have to face to query terrain geometries for basic movements and subsequent geomorphology reactions, this was addressed using OpenSceneGraph provided KD-Tree structure (k-dimensional trees): it is a space partitioning data structure for organizing points in a k-dimensional space and speed up in this context the intersection visitors on the scene-graph. Performance gain with enabled KD-Trees is more than 10 times faster and their building at the very beginning is very fast and without any noticeable hit even in a paged terrain context, where tiles are loaded and mounted on the current graph from disk. The crowd is kept by the main model as a node containing children, each one consisting of a transform node representing the agent position and attitude being updated. The pure 3D geometry of the agent is loaded from the 3D pool consisting of a local or remote directory containing different models. These models represent typologies and they are instantiated once and linked to agent transform nodes, maintaining a light memory footprint: for example if the pool has a single model (a single typology) every agent unit has a reference to the same geometry. The rendering of a massive crowd is also addressed using level of detail nodes that switch between different representations of a given typology (high, medium and low resolution).

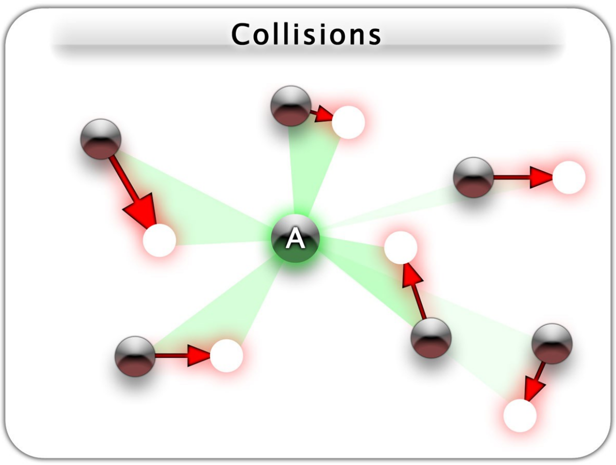

Figure 5. Collision prediction algorithm.

Collision avoidance within such a huge population is often a challenge: the algorithm developed into the system predicts other surrounding agents movements (using directions and speed) and pushes the collision reaction into the set of influences with a priority based on distances (Fig.5). A basic implementation of a crowd model consisting of ***n*** agents would grow in complexity as the order of $O(n^2)$ among the computation of social influences (collision avoidance, local flock cohesion, average speed, etc...).

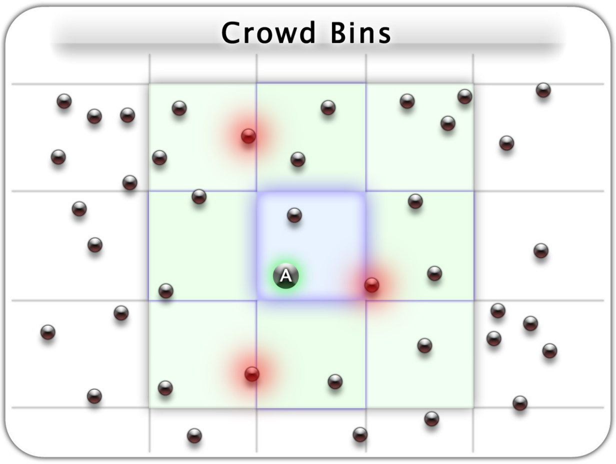

Figure 6. Using bins to quickly access neighbourhood

To reduce the load for large amount of agents, the system computes a spatial partitioning of the population into "*bins*" that results in a faster social computations as the number of neighbors taken into account is sightly reduced (Fig.6) with a considerable gain with large number of agents (Fig.7).

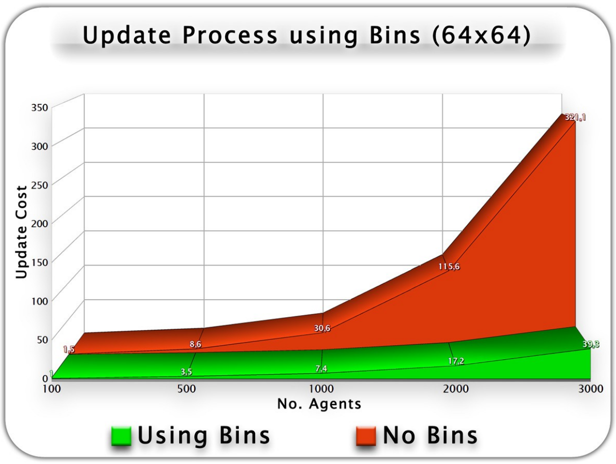

Figure 7. A cost comparison between standard implementation (red) and spatial partitioning with bins (green).

6. CONCLUSIONS

The presented architecture provides a layer upon large landscapes to furnish a customized and highly parameterizable generation of agent-based crowds. The developed architecture is suited for virtual heritage exploration with users allowed to tune rules and weights of artificial intelligences up to simulate proper human movement through that *mechanizable* factors. Separate modules of the model allow to enable or disable various features depending on local system capabilities or user needs in a particular context, such as affordance maps or pheromone trails to build optimal paths along with geo-morphology and terrain shape influences. The use of open-source framework OpenSceneGraph provided a solid base to develop an architecture that efficiently adapts to large terrain datasets: database paging techniques and graphs optimization together with a clean and a well-studied toolkit have been important factors for the final choice. The agent-based layer initializes a crowd that through many simulations and tests has shown the ability to automatically adapt to the reconstructed world, providing a set of magnetic hotspots that map valuable areas or buildings thus influencing agent behaviours, they can be also linked to generate customized magnetic paths. The main concept of the hotspot node is to contain different information, contents and arbitrary media relative to the current place that are collected by agents during their visit and reproduced on demand when user interacts with the crowd. Evolution of the model leads to a *moving informative crowd* that can supply information collected from hotspots placed across the surface and thus plunge dynamic elements into the virtual world that can be used as tour-guides or entities able to modify surrounding environment for historical reconstructions, didactic purposes or demos. The system concept and algorithms used, from collisions to social evolutions, adapt both to small and large landscapes, although the main aim is to provide a solid support for huge datasets: the use of crowd spatial partitioning and other algorithms allows to generate a high number of agents with interactive frame-rate, within the challenge of paged data and providing good scalability also thanks to the overall system modularization.

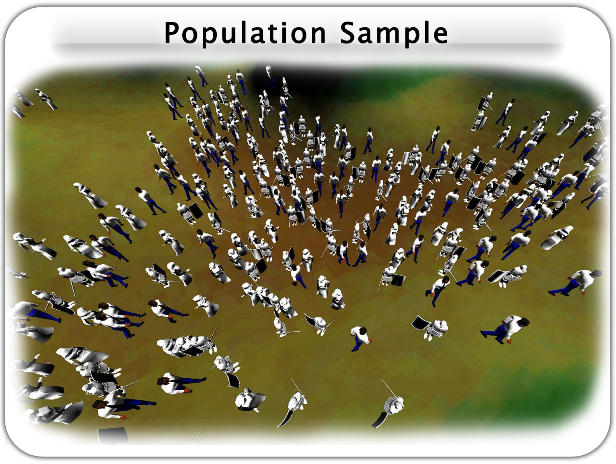

Figure 8. Part of a crowd with 1000 agents with 3 different typologies at 60 FPS.

Future optimizations and user-friendly interfaces will be able to allow user to visually setup the model rules and parameters. Integration of the model into a web context using some recent plugins (such as OSG4web used for Virtual Rome) and the upcoming support in OpenSceneGraph for document rendering such as *html* and *pdf* inside the 3D world are other possibilities, although a wide set of virtual-reality applications of the system are reasonable veins.

7. REFERENCES

S. Pescarin, A. Palombini, V. Vassallo, L. Calori, C. Camporesi, B. Fanini, M. Forte, 2009. *Virtual Rome*. Proceedings CAA 2009, Williamsburg (VA).

P. Murrieta Flores, 2009. *Traveling in a Prehistoric Landscape: Exploring the Influences that Shaped Human Movement.* Proceedings CAA 2009, Williamsburg (VA).

B. Fanini, 2009. *ViRo*. IV ArcheoFoss Conference, Rome (Italy).

C. Loscos, D. Marchal, A. Meyer 2003. *Intuitive Crowd Behaviour in Dense Urban Environments using Local Laws*.

C. Reynolds, 1987. *Flocks, Herds, and Schools: A Distributed Behavioral Model*. ACM Siggraph '87 Conference Proceedings, Anaheim, California.

M. Macal, J. North, 2008. *Agent-based Modeling and Simulation*. Proceedings of the Winter Simulation Conference.

B. Ulicny, D. Thalmann, 2002. *Crowd Simulation for Virtual Heritage*.

M. Dorigo L. Gambardella, 1997. *Ant colonies for the traveling salesman problem*.

R. Kota, V. Bansal, K. Karlapalem, 2007. *System Issues in Crowd Simulation using Massively Multi-Agent Systems*. Eighth International Workshop on Multi-Agent-Based Simulation (MABS) at AAMAS '07, Honolulu (USA).

8. ACKNOWLEDGEMENTS

Many thanks to the OpenSceneGraph community for the software, support and knowledge sharing.