WPE II Paper

Crowd Simulation: Implementation on Geometry, Animation and Behavior

Erdan Gu

Advisor: Norm Badler
Human Modeling and Simulation Center,
Computer Information and Science Department,
University of Pennsylvania
Abstract

Crowd simulations are becoming more popular in entertainment industry and interactive applications. However, the task of rendering and animating crowds of virtual humans in real-time is challenging. In this paper, we describe recent progress on crowd simulation research on three different levels, namely, geometry, animation and behavior. These three levels deal with implementation issues from low level to high level, equivalently, from rendering to controlling of the avatars. At the geometric rendering level, three acceleration techniques are introduced: culling, geometric level of detail and image-based impostor. Culling algorithm basically discards invisible parts of objects in the screen. But it is not sufficient since one of its limitations to displaying numerous virtual humans lies in the number of polygons that can be processed by the rendering pipeline. Thus, geometric level of detail (LOD) seems to be the most common answer to this problem. Each simulated human has multiple different resolutions ranging from million polygons down to only dozen polygons in the application. Another solution relying on the intrinsic temporal coherence of the animation, termed image-based impostors, can visualize thousands of virtual avatars within the view of the observer. At the animation level, animation LOD and fast collision detection techniques, required to deal with crowds of virtual humans, are presented. Animation LOD can be employed to reduce the computational cost of updating the behavior of characters that are less important. Consequently, a scalable but less accurate method for collision detection, “height map”, is used when the interference between objects might not be noticeable, while the hierarchal collision detection is introduced to decide for the inter-human collision and easily adapted into different detail levels. For behavior, Reynolds’ Boids pioneered the behavioral animation work, Bouvier, Brogan and Hodgins have used particle systems and dynamics for modeling the motion of groups with significant physics, and rule-based behavioral systems are also exploited to simulate human behaviors. Then, ViCrowd, a flocking systems-based hierarchical model is presented in detail to exemplify a behavioral system to simulate crowd behaviors. Finally, the paper closes on a summary of these three implementation levels, and we describe a viable framework for future work.

Keywords: crowd simulation, crowd rendering, culling, impostor, level of detail, collision detection, flocking, behavioral animation
1 Introduction

The wide use of computer graphics in games, entertainment, medical, architectural and cultural applications, has led it to becoming a prevalent area of research. At the current stage of technology, a user can interactively navigate through complex, polygon-based scenes rendered with sophisticated lighting effects and high quality antialiasing techniques. The presence of virtual human (agent) with which users can interact greatly increases virtual environment fidelity, are also becoming more popular.

An accepted definition of crowd is that of a group of individuals in the same physical environment, sharing a common goal. Crowd Simulator System is used to produce motion and simulate the behavior for a collection of groups of characters. And crowd and group simulations have advanced dramatically in recent years, revolutionizing the motion picture, game and multimedia industries. For example, in the movie Titanic (fig.1), extensive use was made of virtual people, whose movements were generated from a library of precaptured motions. Such technology can be used in situations where it is dangerous for real people to perform the actions, such as falling over 50 feet off a ship, or to reduce the complexity and expense of handling large numbers of human extras.

![Fig.1 Virtual people on the virtual ship in “Titanic” (courtesy of [1])](image)

All these applications try to satisfy two contradictory goals: speed and realism. Speed is usually required to enable the user to interact with the software. Some applications lay the emphasis on speed and interaction, such as arcade game, where players obviously expect the system to respond immediately to their actions. Response time is also critical in VR applications where avatars may interact with the environment or with other users. On
the other hand, some other applications stress visual accuracy, and our acceptance of virtual characters in terms of realism has greatly improved over the past few years. For example, today’s virtual humans embody real participants in collaborative virtual environments more faithfully than ever, and are capable of conveying emotions through facial animation. It is computationally so demanding for real time application. Therefore, it remains very hard to strike the right compromise between realism and animation speed.

In this paper, we describe recent progress on crowd simulation research on three different levels, namely, geometry, animation and behavior. These three levels deal with implementation issues from low level to high level, equivalently, from rendering to controlling of avatars. The rest part of the paper is organized as follows. The next section focuses on geometric representation of human and scene, and introduces three acceleration techniques for rendering purposes: culling, geometric level of detail and image-based impostor. At the animation level, animation LOD and fast collision detection technique, required to deal with crowds of virtual humans, are presented in section 3. For behavior, section 4 describes several significant efforts, such as Reynolds’ Boids, to simulate and interact with virtual actors. Then, ViCrowd, a flocking systems-based hierarchical model will be presented in detail to exemplify a behavioral tool to simulate crowd behaviors. Finally, the paper closes on a summary of these three implementation levels, and we describe a viable framework for future work.

2. Geometric Rendering Level

The requirement for real-time frame rate means that only a limited number of polygons can be displayed by graphics engines in each frame. Therefore, the high polygon count often has to be reduced in order to achieve acceptable display rates. There exist various techniques to speed up the rendering geometrical scene in crowd simulation application. They roughly fall into three categories: culling, image-based impostor and geometric level of detail. They all have in common the idea of reducing the complexity of the scene while retaining its visual characteristics.

2.1 Culling

In real crowd simulation application, two separated problems are often considered:
visualization of large-scale static environments, and visualization of animated crowds and traffic. Both are expensive to render, containing thousands of polygons. Although thousands of polygons can be displayed and visualized in a real-time frame rate, delays appear between frames for a larger number of polygons decreasing the quality of the visualization and the ability of the user to walk through.

To address this problem, culling algorithm [2] is proposed, the basic idea of which is to discard as much as possible parts of objects that are not visible in the final rendered image. Visibility culling occurs when object lies outside the viewing frustum, while occlusion culling means that it is occluded by other parts of the scene. In rendering huge data sets which require real-time feedbacks, visibility culling is problematic. Occlusion culling can perform more appropriate geometric simplification for densely occluded architectural models. Airey et al.[3], Teller and Sequin[4], Luebke and Georges [5] used spatial subdivision and attempted to precompute visibility relationships within a complex building scene. The central idea of these approaches is to predict the visible part of a scene for the next few frames, thus reducing the number of primitives which must be rendered. This is achieved using intelligent memory management and viewer motion prediction.

One major drawback of occlusion culling algorithms is that they usually require a specific organization of the entire geometry database: the scene is typically divided into smaller units or cells. This makes occlusion culling perform poorly for individual, deforming, moving objects. Because of this restriction, it is no use trying to apply on virtual humans if they are considered individually. Nevertheless, occlusion culling might be contemplated at a higher level if virtual humans are placed in a densely occluded world, as could be the case when simulating a human crowd for instance.

Tecchia suggested a new occlusion algorithm, dynamic culling [6], based on binary tree merging. When the viewpoint gets closer to the ground, buildings become very effective occluders. Tecchia’s method used the discretization and the properties of urban scenes to quickly cull away not only the invisible static geometry but also avatars. First they built a KD-tree of the scene geometry, using as partitions only planes that coincide with tile edges (given by the 2D grid). Then for each frame they built an occlusion tree from the current viewpoint and merged it with the KD-tree, marking its leaves, and indirectly the tiles, as visible or hidden. Finally, the state of the occupied cell was checked
before each avatar was processed any further. Fig. 2 is the result of culling in a highly populated urban scene.

2.2 Level of Detail
Culling can be a very efficient acceleration in urban scenes, but still many polygons would need to be rendered. Geometric level of detail (LOD) attempts to reduce the number of rendered polygons by using several representations of decreasing complexity of an object. For each frame, the appropriate model or resolution is selected. The selection criterion, termed geometric LOD resolver, is usually the size of the object relative to the distance to the viewer.

The major hindrance to using geometric LOD is related to the problem of multi-resolution modeling, that is to say the automatic generation from a 3D object of simpler, coarser 3D representations that bear as strong a resemblance as possible to the original object. To overcome this difficulty, Aubel et al. [7] constructed virtual actors by the B-spline patches. They associated each body part (arm, leg, pelvis, etc.) with a B-spline patch. It is then quite straightforward to generate, for each body part, similar meshes of increasing coarseness (fig. 3). Note that the body extremities can be cleverly replaced with simple textured geometry for the lowest resolution, thus dramatically reducing the number of triangles. By relying on implicit surfaces, they avoided resorting to complicated algorithms and automatically generated multiple look-alike resolutions of a model.

Another acceleration techniques, image-based impostors technique, were also used to

Fig.2 A closer view of populated urban environments (courtesy of [6])


substantially reduce scene complexity in their work. It would be further described in Section 2.3.

More recently, some more work on subdivision surfaces [8] provides visually better approximations for increasing and reducing the detail of characters. Many of the problems associated with other curved surface representations such as NURBS can be avoided by using subdivision schemes, which behave in a way similar to polygonal meshes. Character skins can be used almost directly with some subdivision schemes while others require modification in order to get a good representation of the original mesh. Another advantage of subdivision surfaces is that only a low-resolution mesh is necessary as the starting point. Fig. 4 shows examples of some subdivision schemes that can be used: Linear, Butterfly and

Fig. 3 Multi-resolution virtual human, left: mesh; right: textured (courtesy of [7])

Fig. 4 Two Iterations of the linear, butterfly and loop subdivision (courtesy of [9])
2.3 Image-based Impostors

In a crowded square, a user might visualize thousands of virtual avatars as well as view the surrounding environment details. Culling is not sufficient to reduce the number of polygons to display as well as to take care of the real-time animation of avatars. Even the additional use of level of details techniques results in too many polygons to display. Considering these limitations, image-based impostor approach seems more suitable when the viewer is at a certain distance from the virtual humans, and potentially has in view a very large number of them.

Current graphics systems rarely take advantage of temporal coherence during animation. Yet, changes from frame to frame in a static scene are typically very small, which can be exploited. Relying on the intrinsic temporal coherence of the animation, image-based impostors reuse parts of the frame buffer content over several frames, thus avoiding rendering for each frame the whole scene from scratch.

Tecchia and Chrysanthou [10] used fully precomputed images (fig. 5), in which each human is represented with a single adaptive impostor, in order to minimize geometric complexity. However, in that approach, the required texture memory is excessive, resulting in a simulation that includes only few types of avatars. Then, they used aggressive optimization, removing all unused space in the impostor image set, to reduce in this way the memory requirements of about 3/4. Fig. 6 illustrates the rendering of impostor using compressed texture that lies on the bottom right. To enhance the crowd variety without increasing the memory usage, they can selectively address and modulate the base color of
different regions on each impostor image using multiple rendering passes. Furthermore, to minimize the popping effect when changing views, they chose the appropriate size and displacement that fit best for walking humans. Besides, computing and displaying the shadows of moving humans greatly improves the visual realism of the scene. Figure.7

![Image](https://example.com/image.jpg)

**Fig.7 visualization of crowds of virtual human (courtesy of [10])** shows the visualization of thousands of different humans in real time using the optimization techniques presented in their paper.

Tecchia and Chrysanthou proposed a scalable but less accurate method that shows results with only few individuals replicated many times due to the excessive texture requirements. An alternative approach employed by Aubel et al. is the use of animated impostors, where textures are generated on demand. In this case, they used a single, alpha-matted, textured quadrilateral (or billboard) for impostor. The texture that is mapped onto the quadrilateral needs to be refreshed from time to time because of the mobility of the virtual human or camera motion, and regularly enough for the quadrilateral not to be visually distracting. This process is done in three steps: firstly, an off-screen buffer is set up that will receive the snapshot; then, the virtual human is taken on the new posture and the camera is placed accordingly; finally, the actor is rendered and the result is copied into texture memory. Even in the worst cases (very high texture refreshment rate), impostors prove not to be slower than rendering the actual 3D geometry. Fig.8 shows that there are twenty walking virtual human, two football teams moving in the simulation. And the left part is rendered by full geometry while the right is represented by impostors.
Impostors appear to be one of the most promising techniques. As a summary of this section, we characterize image-based impostor technique here. On one hand, it is a software technique that is not very demanding in terms of hardware; simulation considerations can be easily integrated (quasi-constant frame rate, rendering priorities etc.); impostors can (and should) be combined with other acceleration techniques, such as LOD. On the other hand, the weaknesses are the following: texture memory usage; small animations that improve the visual realism (eg. breathing, fidgeting) are sacrificed. This technique is ideal for visualizing large crowds in real-time, but the representations are not detailed enough for closer viewing, being simply flat textures. In addition, severe popping effects occur when switching from the impostor to a full geometrical model.

3. Animation Level

The term ‘level-of-detail’ has been widely used in relation to research on levels of detail in geometric models. As Carlson and Hodgins pointed out [11], these geometric techniques are also relevant to systems that vary the detail of simulation. Essentially, this means that one may switch between animation levels of varying computation complexity. Therefore, we introduce the notion of animation Level Of Detail. Exact collision detection using standard methods for every moving entity is very expensive and is probably not necessary for crowd simulation. Accordingly, adaptive collision detection will perform more properly for animation LOD.
3.1 Animation LOD

Animation is about controlling or specifying the motion of objects. The classical problem is “insufficient animator leverage”, which means that the animator has to (tediously) specify all motion in great detail. Consider, for example, the workload involved in the Pixar production “Toy Story” where the main character contained 700 DOFs (including the facial animation controls). It is computationally prohibitive, especially for real time application, though this yields very realistic results. Animation LOD is introduced here for describing such a technique, namely, the animator can switch animation levels in term of realism for varying computation complexity. This ensures that we do not do any needless and wasteful computations for objects that are not asserting high detail motion.

When the characters at the lowest level of detail are far from the viewer, key-framed animations, chosen at random and changed at different intervals, are used to animate virtual humans. When the viewer focuses on these characters, actions that are more meaningful are then chosen. Animation LOD means, applying more sophisticated behavior, it requires generating more realistic motions that do not cycle and become more accurate as the character increases in importance.

![ALOHA inverse kinematics test-bed](image)

**Fig.9 The ALOHA inverse kinematics test-bed (courtesy of [12])**

In the simple case, predefined forward kinematics controls from motion capture data, combining interpolation techniques, to obtain a high-speed, low-detail animation. The
character has the ability to interact with dynamic changing environment, thus, motion capture data suffers since it can only contain pre-recorded sequences, and, in its basic form, can not easily adapt in real time. Inverse kinematics (fig.9) is a useful tool to solve for the joint angles of an articulated model given the positions of the root and end-effector and the lengths of the links to overcome the disadvantages of motion capture. Once a certain movement is rated with a low importance by the LOD resolver, the inverse kinematics will be allocated less iteration with which to resolve the joint angles. But it is almost certain that inverse kinematics will not provide a good practice solution, and will not necessarily produce convincing motion. However, the LOD resolver should only approve of such measures in cases where the viewer will not perceive them since such techniques will sacrifice simulation accuracy and smoothness for speed.

On the other hand, if a high level of realism is required, physical based animation, which used dynamic simulation to script the motion, occurs. ALHOA framework [12] uses a dynamic constraint based technique [13] for such simulations. Since greater accuracy implies greater computational complexity, this technique will be reserved for animation that is highly rated by the LOD resolver.

3.2 Collision Detection

Traditional approaches [14] for highly detailed collision detection with virtual humans will perform polygon level checks between humans and objects. Although it might be appropriate for offline simulations and for smaller, less populated scenes, this obviously becomes computationally infeasible for a large scene with many virtual humans interacting and being simulated accurately at the same time.

In Tecchia’s work on populated urban scene, a method [15], fast enough to run in real-time with tenths of thousands of particles, was proposed. The overall idea of the algorithm is to create a discrete representation of the static part of the model (the height map) and use it to detect collisions of moving particles with the environment. This map stores the height at each point in the environment. For every frame of the simulation, before moving a particle to its new position, its current elevation is checked against that stored in the height-map for the target-position. It will not only prevent the human from walking through building but also detect and adjust their elevation on the model without
having to query the geometry database. It can trade off small errors in exchange of greater speed and scalability. But its current implementation only detects interference with the static parts of the environment (i.e. no inter-particle collision), thus only works for collision avoidance. Thus, collision avoidance often gives the appearance that the crowd is too sparse, and therefore unconvincing. Attending a concert or sporting, real people bump up against each other, or are squashed together, or even if the crowd is not too tight, people who know each other will be walking along, chatting, holding hands, or occasionally touching. Therefore, we might seek other methods for more precise and involved cases.

There are many techniques to detect interference between geometric objects. Many of them use hierarchical data structures, for example, hierarchical bounding boxes [16], spheres trees [17] and BSP trees [18]. In ALHOA framework, a hierarchy of rigid bounding volumes based on Line Swept Spheres (LSS) is employed (fig. 10). Line swept spheres (LSS) are the volumes generated by sweeping a sphere across a line segment. A hierarchy of LSS nodes is an efficient way to model characters for collision detection as LSS’s provide the best combination of ease of computation and tight-boundedness in the case of virtual humans. If the characters are modeled based on hierarchical transforms based on LSS’s then these transforms can easily be used to update the positions of nodes in the collision detection hierarchy, while the rest of the environment can be modeled on more general volume representations using heterogeneous collision primitives such as spheres, planes and boxes.

Moreover, it needs collision responses, based on behavioral rules, in order to achieve a more chaotic, less military look to the crowds. For example, people may react differently if they bump into or touch a stranger. An optimal solution framework [19] was developed by O’Sullivan et al. for the inclusion of physically correct responsive objects within the
virtual space. This work is primarily concerned with the provision of a feasible real time level of detail hybrid impulse/constraint based solution.

4. Behavior Level

Typical computer animation models only the shapes and physical properties of characters, whereas behavioral animation seeks to model the behaviors of characters. The goal of behavior animation is for such simulated characters to handle many of the details of their actions, and hence their motions. These behaviors include a whole range of activities from simple path planning to complex "emotional" interactions between characters.

History reveals a great amount of interest given towards understanding and controlling behavior of crowds of people over past decades. The researches on behavioral modeling have mainly focused on various aspects of human control, such as particle systems, flocking systems and behavioral systems. Table 1, modified from [20], characterizes these techniques in different aspects, like the possible number of individuals to be simulated,

<table>
<thead>
<tr>
<th>Method</th>
<th>Particle system</th>
<th>Flocking system</th>
<th>Behavioral system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure</td>
<td>Nonhierarchical</td>
<td>Levels: flock, agents</td>
<td>Hierarchy</td>
</tr>
<tr>
<td>Participants</td>
<td>Many</td>
<td>Some</td>
<td>Few</td>
</tr>
<tr>
<td>Intelligence</td>
<td>None</td>
<td>Some</td>
<td>High</td>
</tr>
<tr>
<td>Physics-based</td>
<td>Yes</td>
<td>Some</td>
<td>No</td>
</tr>
<tr>
<td>Control</td>
<td>Force fields, global tendency</td>
<td>Local tendency</td>
<td>Rules</td>
</tr>
</tbody>
</table>

Table 1. Methods comparison
their autonomy level and decision-making ability, etc.

Recent research into crowd simulation has to a large extent been inspired by the flocking work of Craig Reynolds [21]. He revolutionized the animation of flocks of animals, in his case birds (or “Boids”), by adapting some ideas from particle systems, where each individual bird is a particle. In fact, the birds (or ‘boids’) maintain proper position and orientation within the flock by balancing their desire to avoid collisions with neighbors, to match the velocity of neighbors and to move towards the center of the flock. In recent work, Bouvier [22], Brogan and Hodgins [23] used particle systems and dynamics respectively for modeling the motion of groups with significant physics.

Furthermore, rule-based behavioral systems are well known and often exploited to simulate human behaviors. Zeltzer [24] presented a classification of levels of interaction and abstraction required in different applications. Noser and Thalmann [25] have described a L-system animation to model autonomous agents able to learn using synthetic vision and perception issues. Badler et al. [26] created a task planner and a biomechanical model for virtual humans illustrated with autonomous soldiers. Unuma et al. [27] modeled human figure locomotion with emotions. More recently, Musse and Thalmann [28] presented a flocking systems-based hierarchical model--ViCrowd. In next section, we will describe this model in detail to exemplify a behavioral system to simulate crowd behaviors.

4.1 ViCrowd System

ViCrowd system addresses two main issues: crowd structure and crowd behavior. Considering crowd structure, ViCrowd model deals with a hierarchy composed of crowd, groups and agents. Concerning crowd behavior, virtual agents are endowed with different levels of autonomy (LOA) including scripted, reactive and guided behavior.

The overview of ViCrowd architecture is presented below. Firstly, three categories of information: knowledge, beliefs and intentions, are dealt with in order to characterize the crowds. Knowledge represents the information of the virtual environment. Group knowledge concerns the memory of groups related to the past experiences as well as perception related to agents and groups. Beliefs describe the internal status of groups and individuals. Finally, intention represents the goals of the crowd and groups of agents. The crowd information together with perceptions, innate and sociological behaviors and
The treatment of events are processed by “Behavioral Motor” (BM) in order to achieve the groups and individuals low-level behaviors (fig. 12). In (1), the scripted behaviors are defined before the simulation, (2) describes the process from which one is able to interact and to send orders to the virtual crowd. In (3), the crowd information described in (1) and (2) is distributed among groups. In (4), the group information describes the knowledge, intentions and beliefs associated with other information like events and sociological behaviors (5). In (6), the BM is represented concerning the process where the group and individual low-level behaviors are generated, e.g. current goal, current speed, current emotional status, etc. The behaviors generated for groups/individuals are (7):

- Internal status of individuals and groups which deals with the way of walking, walking speed and repertory of basic actions.
- Goals (motion, action).

The novel idea of ViCrowd paper is to present the possibility of generating simulations and controlling motion with different degrees of autonomy. Another contribution is that the sense of crowd behavior and presence have been achieved in ViCrowd system where families and groups of friends can be observed, comparing this group-based approach with the individual approach. Musse and her colleague are currently investigating the
simulation of crowd behaviors in panic and emergency situations \[29\] and modeling the crowd motion as well as the sociological behavior of the people in these conditions \[30\].

\section{Conclusion and Future Work}

This paper has presented recent progress on crowd simulation research on three different levels, namely, geometry, animation and behavior. These three levels deal with implementation issues from low level to high level, equivalently, from rendering to controlling of avatars.

The method to culling polygon could be a very efficient acceleration in urban scenes, but thinking of, for example, a crowded square, a user might visualize thousands of virtual avatars as well as view the surrounding environment details. Culling is not sufficient to reduce the number of polygons to display as well as to take care of the real-time animation of avatars. Even the additional use of level of detail techniques results in too many polygons to display. Considering these limitations, image-based impostor approach seems more suitable when the viewer is at a certain distance from the virtual humans, and potentially has in view a very large number of them.

At the animation level, the movements themselves can be simulated at adaptive levels of details regarding to LOD solver. Animation LOD can be employed to reduce the computational cost of updating the behavior of characters that are less important. Consequently, a scalable but less accurate collision detection method “height map” is used when the interference between objects might not be noticeable, while hierarchal collision detection, is introduced to decide for the inter-human collision and easily adapted into different detail level.
For behavior, Reynolds’ Boids, a distributed behavior model for simulating flocks formed by actors endowed with perception skills, pioneered the behavioral animation work. Bouvier, Brogan and Hodgins have used particle systems and dynamics for modeling the motion of groups with significant physics. And rule-based behavioral systems are also exploited to simulate human behaviors. More recently, ViCrowd model proposed by Musse et al. presents different degrees of autonomy for control, and the sense of crowd behavior and presence were achieved in the model where families and groups of friends can be observed.

5.1 A Viable Framework of Crowd Simulation

Before the end of this paper, we attempt to describe a viable framework for future work that will amalgamate the research efforts on three implementation levels, geometry, animation and behavior into a scalable system for crowd simulation. The expected system should be robust enough to deal with realistic real time characters in a dynamically changing environment, from the conversation between 2-3 people, even to densely populated environment, as well as keeping a consistently high and smooth frame rate.

A high level view of this framework reveals four major modules (fig. 14): heuristic LOD resolver, geometry controller, animation controller and behavior controller.

**Heuristic LOD resolver:** A more generic LOD resolver is interested to be investigated in all levels, so that truly polymorphic LOD control can be achieved. An important criterion for heuristic resolver is that viewer perception of any degradation in the simulation remains
minimal. Several factors would be taken into account [31]: the complexity of the environment and the distance to the viewer, image-space size, position in the viewing plane, distance to the line of sight, motion blur. Objects can be assigned different priorities since they may play not an equal role in the simulation.

**Geometry controller:** The geometric controller encapsulates all the information necessary for rendering the object’s appearance in the scene. Occlusion culling firstly is used as preprocessing for highly occluded architectural environment. Then, full geometric model, replaced by a lower resolution one, combining with 2D impostor, would allows further reduction of the number of polygons to be required based on important heuristics. We would aggregate groups of virtual humans, or the scene into billboard impostor. The impostor is broken into several layers, each rendered with a different depth value.

**Animation controller:** The motion controller is responsible for handling any animation at any level of detail specified by the LOD resolver, to provide a scalable framework. A lower level motion capture technique is employed when the system deems that that level of detail is sufficiently low enough. Forward kinematics and inverse kinematics are used to supplement the required realism of our motion. For control over the simulation of great accuracy, physical based animation could be used instead.

**Behavior controller:** We will follow ViCrowd hierarchal structure, agent, group and crowd to build our crowd model. Comparing three categories of information in ViCrowd system, our behavior controller has ability to manage some more complicated animation request. It is the duty of the behavior controller to register animation requests, to store necessary information by associated memory, to update knowledge, thus enhance decision rule set.
Reference

in complex virtual building environments”, in Computer Graphics (1990 Symposium on Interactive 3D
In ACM Siggraph Technical Sketch, August 2001.
[7] A. Aubel, R. Boulic and D. Thalmann. “Real-time Display of Virtual Humans: Level of Details and
Impostors”, IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on 3D
1988, pp. 179-188.
detection system for large-scale environments, Proceedings of ACM Interactive 3D Graphics


