The 5th International Symposium on Virtual Reality, Archaeology and Cultural Heritage
VAST (2004)
K. Cain, Y. Chrysanthou, F. Niccolucci, N. Silberman (Editors)

Adaptive Crowd Behaviour to Aid Real-Time Rendering of a Cultural Heritage Environment

G. Ryder, P. Flack and A.M. Day

School of Computer Sciences, University of East Anglia, Norwich, NR4 7TJ, UK

Abstract

In current city visualisations crowds are being included to increase realism in the scene. With the self-steering nature of crowds it is traditionally difficult to control the number of humans that could be in view at any one time. While rendering speedups have been successfully applied for many years, this paper takes another approach with the aim to keep a steady frame rate. We attempt to influence the crowd dynamics to maintain the frame rate, without this becoming apparent to the user. We show how this work can be applied to a virtual reality tour of a medieval town.

Categories and Subject Descriptors (according to ACM CCS): I.3.8 [Computer Graphics]: Applications I.2.9 [Robotics]: Autonomous vehicles

1. Goal

The work presented in this paper is associated with the EPOCH project [EPO04], to recreate the medieval town of Wolfenbuettel and populate it with a number of interactive avatars at interactive frame rates by limiting the movement of the crowd. Our current work extends the CHARISMATIC [HFDA03] project in researching rapid methods for modelling and rendering large scale urban environments. Figure 1 shows a typical scene with a small number of high-resolution avatars.

Figure 1: *Example screenshot of the Wolfenbuettel scene with an avatar guide*

2. Background

Rendering scenes of complexity beyond which can be visualised in real-time in their native form, has been the focus of work for many years. Funkhouser and Séquin[FS93] presented a framework which has the ability to render objects in the static scene at different complexity and rendering methods. While the sizes of scenes attempted and the bottlenecks of rendering have changed, the basic concepts of their work have much validity in scene rendering today.

The main focus of the work is adapting the quality of the image to maintain a consistent frame rate. This is a common idea to a number of the methods that focus on real-time speedups. However for dynamic objects there are other issues and possible solutions.

There has been significant work on reducing the rendering overhead for displaying a crowd of virtual humans. Image based rendering has shown promise at visualising crowds of a significant size (over 1000 people) [TC02, ABT99, LTC01]. Figure 2 shows an example screenshot from our implementation of image based rendering for avatars. These methods are suitable for large crowds that are maintained at a distance to the camera. There are significant memory overheads, along with a very limited animation set allowed for each character. Geometry based Level of Detail (LOD) systems can provide increased visual fidelity; including avoiding the aliasing problems associated with image based methods. However a combined approach, using geometry for near the camera and impostors for those further away, removes many of the problems associated with each [ABT00].

Figure 2: *Example of Image Based rendering, using a normal map to allow for dynamic relighting*

Maintaining a consistent frame rate cannot be solved by reducing the rendering cost of each individual human. Each human has the ability to move freely about the scene and thus it is possible that at any time an unacceptably high number of Virtual Humans (VHs) will be close to the camera. It is possible that instead of using fixed distance switching for a LOD solution, these boundaries could be moved based upon the number of people within view. This will lead to a representation being viewed closer than originally intended, possibly resulting in lower quality results. Thus with standard rendering techniques alone it is not possible to maintain refresh rate, image quality and also a navigable path through the scene. In this paper we examine the adjustment of crowd dynamics as a solution to this problem.

Path planning for crowds is traditionally based upon robot locomotion schemes. Robot locomotion problems can be split into the following groups:

- Roadmap approaches, an example of which is [KL94]
- Cell Decomposition, Bandi and Thalmann [BT98] present an implementation for humans
- Potential Field, Reif and Wang [RW95] extend to include social forces
- Rule based, [Rey99]

In addition to those spawned from the robot locomotion field, there are other possible solutions. Chenny [Che04] presented a velocity field based solution, which allows a designer to assign directions of travel for a crowd through a scene. However, the limiting nature of the constraints of the method does not make it suitable for crowd movements within a complex city (for example crowd paths cannot cross).

Our implementation is built upon a rule based solution, due to the desire to alter each person's "driving" forces on a frame by frame basis. Craig Reynolds presented simple rule driven particles [Rey87, Rey99] for the simulation of flocks and crowds through emergent behaviour. The behaviour of each particle is unpredictable over a moderate time scale. Crowd behaviour for a large crowd can be a significant overhead for the system. Boids in its native form is $O(n^2)$ (where 'n' is the number of people in the scene), however with use of spatial data structure this overhead can be reduced to almost $O(n)$. Due to ease of implementation and generation of satisfactory crowd dynamics for testing our methods, the Boids framework [Rey99] was chosen for our test bed. However, we believe the concepts are transferable to other frameworks.

Lamarche and Donikian [LD04] present a method for automatically pre-processing a complex scene and producing real-time paths through the scene. They focus on fast, low cost path finding with collision avoidance and reaction. Through spatial subdivision they generate a roadmap which they search during runtime to allocate paths. These paths are then adapted with local driving rules to handle social rules. An advantage of their method over many is they can handle indoor and outdoor navigation in one system. The run-time overheads of their results indicate that this method shows good promise, though little mention of the perceived visual quality of the results is given.

Tecchia et al. [TLCC01] present a framework for developing behaviour rules. They split the problem into four layers:

- Inter-collision detection layer for collision between the people
- Collision detection layer for collision of the people against the scene
- Behaviour layer for local rules governing a person
- Callback layer for including any pre-scripted behaviour for a particular point in the scene

The end result is a framework very similar to our own, though more formally specified, with a detailed graphics user interface. However, little mention is made of any rules on which to base the crowd movement, beyond elementary test rules.

Group based behaviour and path planning has also been the focus of much work. Kamphuis and Overmars [KO04] present an extension to allow a group of people to be assigned a path (generated by a standard path finding algorithm). While our framework handles each VH as an individual entity, it would be an interesting extension to handle the fact that people travel in sub-groups of varying size within a crowd.

Our implementation uses OpenSG [Ope04] which provides:

- State Sorting
- Multi-threading support (allowing clustering)
- Optimized geometry handling (stripes etc)
- Platform independence
- View volume culling
- Open source implementation for ease of extendibility

A converse strategy to ours for creating crowds in a scene is presented by Ulincy et al [UdHCT04]. They present a graphical interface, "a brush metaphor", in which the user paints the position (and attributes) of the crowd in a scene.

The main advantage of the method is the natural system of placing static people within the scene. There still needs to be in place a behaviour system for the crowd, to move the entities once the simulation starts. The paper mentions that their framework has a simple waypoint driven simulation. Therefore it would be an interesting extension to combine our work with the intuitive interface of the paintbrush metaphor.

3. Framework

In this section we present how we classify the Virtual Humans (VHs) in a scene relative to the viewer. Each VH in the scene is classified twice (see Figure 3 & Figure 4). Firstly, each VH is classified depending on if they are located inside one (the Outer) or both of the view volumes. Secondly, the world space is divided into circular regions based on the distance from the camera, with each VH within one of these regions.

3.1. View Frustum Classifications

Two viewing areas are maintained in our framework. The first, or inner, area is a 2D triangle closely linked to the view frustum used for rendering. It differs in two respects from the viewing volume of the graphics engine. Firstly the sides are always vertical in the Y-axis. This calls for a navigation system where the camera is prevented from rolling (only yaw and pitch are permitted). Secondly, we do not maintain near, top or bottom sides. The "Personal Space" region replaces the concept of a near plane, while top and bottom are not required given the nature of the scenes.

The second volume is identical to the first, except as shown in Figure 3, the apex point is behind the viewer, causing the left and right edges to be outside the viewing volume. This additional area is maintained in order to act as a boundary layer between those VHs that are invisible and have no inter-human collision testing, and those VHs that are visible and thus have full collision detection. When in the boundary zone the humans are not visible, but they are subject to collision detection. This is required to avoid situations where two or more virtual humans enter the viewing volume in the same place, are subject to collision detection and are forcibly separated within view (Figure 5). Outside the outer volume, navigation is still maintained on the VHs, but they are not subject to collision detection, nor are they rendered.

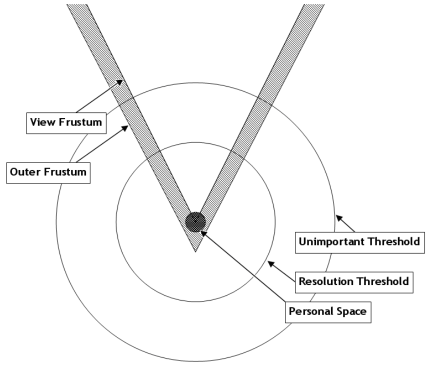

Figure 3: *Threshold distances and viewing volumes*

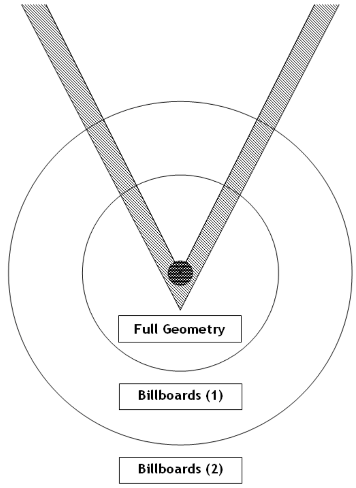

Figure 4: *Full geometry vs. billboard zones*

3.2. Level of Detail Control

In both Figure 3 and Figure 4 the three threshold distances used for grouping the VHs are shown. The inner-most distance (dark circle) represents the personal space of the user, which the VHs are prohibited from entering. This is maintained both to avoid graphical errors (artefacts with the near camera clipping plane) and to emulate real-life, where people cannot occupy the same physical space.

The second threshold distance is the boundary between 3D geometry VHs (high detail) and image based billboard VHs (low detail). The value chosen for the boundary distance is based on a combination of factors:

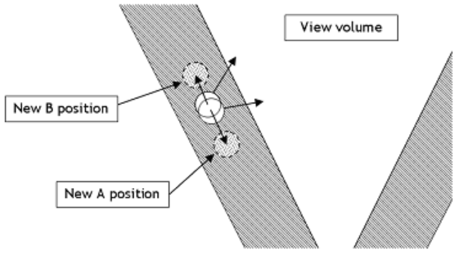

Figure 5: *Humans A and B are entangled and will be in view next frame. They are forcibly separated before that happens, ensuring that the separation does not place either within the view volume instantly*

- Minimising the number of geometry avatars
- Avoiding aliasing artefacts due to the billboard projected size being larger than the original texture
- Minimise "popping". At certain viewing angles a noticeable 'pop' may happen on switching between representations due to the billboard being an approximation of geometry for this viewing angle

As the number of VHs increases (in a general scene increasing both the number of low detail and high detail VHs), the number of billboards being occluded increases, allowing for a closer threshold distance.

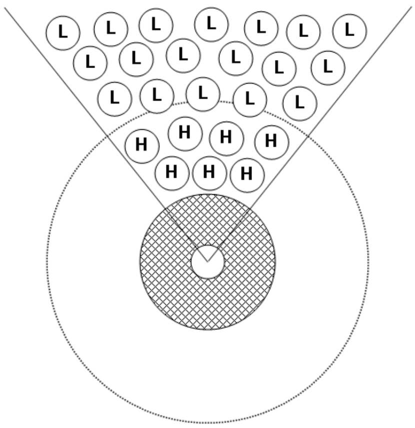

Figure 6: *Dense crowd*

In a dense crowd situation close to the viewing position, depicted in Figure 6, the second and third distance thresholds can be significantly reduced. In this case some, or all, of the popping and aliasing artefacts are hidden by the occlusions caused by the near VHs. Likewise in a sparse crowd situation, depicted in Figure 7, the boundaries can be pushed back to allow a consistent number of high resolution VHs.

We proposed two different methods for controlling the

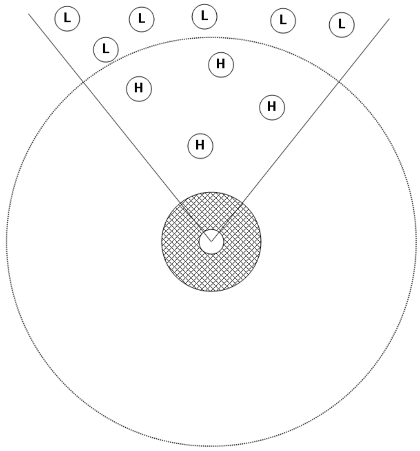

Figure 7: *Sparse crowd*

placements of "resolution boundary". The first option is to allow a fixed number of high detail representations and resize the boundary to maintain this number.

The second option is to adjust the number of high resolution humans based on the frame rate, thus taking into account the overall amount of detail within the scene. This method is beneficial when the complexity of each individual VH is significantly lower than the background scene complexity. Also this allows for additional refinement of the scene when camera speed is low and when high speed is required, the fidelity can be sacrificed.

The third threshold distance represents not a level-of-detail threshold, but a point beyond which the behaviour of the crowd does not concern us - the "Unimportant Threshold" - see Figure 3. At this distance the VHs are far enough away that they cannot change to high-resolution in the near future, thus they are not to be considered for path alteration.

Additionally, not shown in the diagrams, a fourth boundary may be included to represent the far clipping plane of the graphics engine. How this is implemented and where it falls in the rendering pipeline is dependent on the graphics engine / scenegraph structure used (i.e. it may be provided automatically). While geometry is not sent to the graphics card for objects beyond this distance, navigation for these humans must be maintained. Although, the fidelity of the dynamics for these crowd members can be reduced to the most basic of path following, i.e. ignoring inter-person collision detection and avoidance.

4. Dynamics Modification

As previously stated, the number of high-resolution VHs is limited in order to maintain an acceptable frame rate. Once this threshold number has been reached, the behaviour of the crowd must be adapted to deviate the low-resolution VHs away from the high-resolution area. The Boids framework has the ideal solution for this in 'Obstacles'.

We are concerned with maintaining a maximum polygon count for all VHs. When all high-res VHs are the same polygon count (unrealistic simplification) the polygon budget is determined by:

$$(N_H P_H) + (N_L P_L) \leq P_{budget}$$

Where P_H is the number of polygons for a high-resolution VH; P_L the number of polygons for a low-resolution VH; N_H is the number of high-resolution VHs in view; N_L the number of low-resolution VHs in view. In the general case, where there are a different number of polygons in each VH, the polygon budget is determined by:

$$\left(\sum_{i=1}^{N_H} P_H(i)\right) + (N_L P_L) \leq P_{budget}$$

4.1. Camera Obstacles

We introduce an obstacle that moves with the camera to cover the visible high resolution area. However, all of the VHs in the scene are not subject to this obstacle as this would cause no high-resolution VHs. When the number of high-resolution humans falls below the threshold we deactivate the obstacle for the low-resolution avatars.

Our experiments with the size and number of obstacles (only circular shaped obstacles are available in the current framework) have shown that two overlapping obstacles covering the frustum immediately in front of the camera give the most aesthetically pleasing and numerically accurate results. Figure 8 shows the arrangement. Obstacle 2 overlaps the secondary frustum, which has the effect of pushing VHs away when the camera turns and thus preventing them from getting too close. This does not change our original definition of the high/low-resolution threshold, as the regions covered by the obstacles closely approximate the high-resolution region, with the uncovered parts being outside the view volume.

4.2. Scene Obstacles

In addition to the obstacles that move with the camera, there are obstacles to represent the fixed geometry in the scene. These are typically rectangles slightly larger than the buildings they represent. Only a few centimetres of clearance are required to stop the VHs touching the walls. Other obstacles are easily included e.g. circles for trees, lamp posts, etc.

4.3. Special Cases

Additional constraints have been added to the framework to handle cases when the user is moving too quickly.

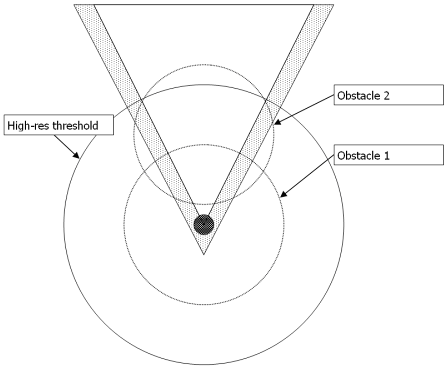

Figure 8: *Obstacles*

4.3.1. High Speed Rotation of the Camera

The "Outer frustum" is present to avoid this problem. The maximum angular velocity is naturally high, so this can cause a human outside both view areas to move inside the main view in a small amount of time. In an extreme situation the human would be outside the outer area in one frame and inside the inner area in the next. If unchecked, this would cause problems as two humans in collision could potentially appear close to the camera. To prevent this, a method of separation is used such that a pair of humans that find themselves entangled (Figure 5) is instant, and designed so that after separation, neither human is within the inner volume. The maximum speed of rotation and gap between the frustums is set to ensure all virtual humans are caught in the gap for at least one frame when the user turns.

4.3.2. Camera Collisions

As the user navigates through a scene it is inevitable that they attempt to navigate through a VH. The approach we adopt is that on colliding with a VH the movement of the camera is slowed or stopped. Changing the path of the camera, i.e. overriding the user's 'wrong' input is potentially annoying for the user and would yield non intuitive results for the navigation system. Another option we consider is the lesser extreme, where there is time to detect the impact and avert it by allowing the VH to move out the way.

If the user attempts to move through a dense crowd it is unrealistic to expect free navigation. The dynamics of how we move through would require a high detail simulation, outside the scope of this work. We are aiming for a visually satisfactory result, not a physically correct one. We modulate the user's speed depending on the crowd density, effectively slowing the user in proportion to the density of the crowd. This allows the crowd enough time to disperse away from the desired walk path.

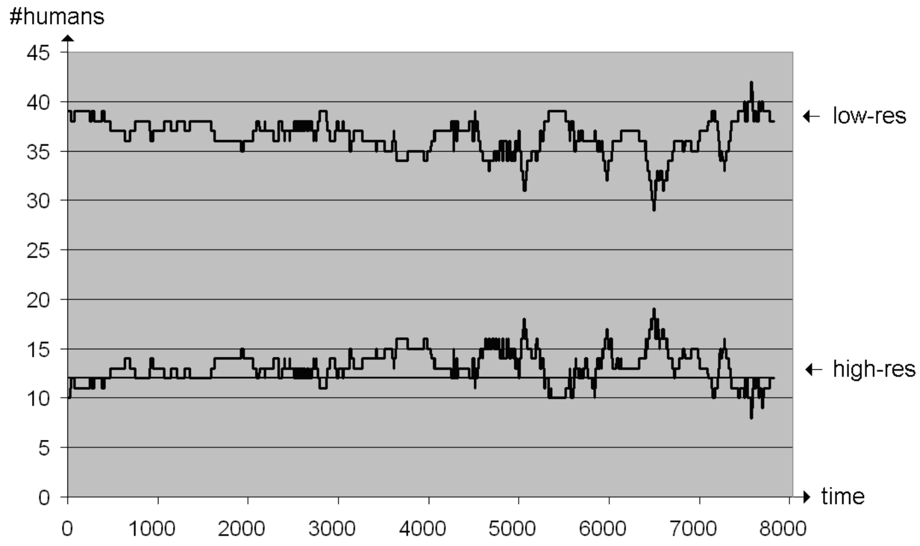

Figure 10: *Walkabout in lightly populated scene. Navigation speed is twice the maximum speed of the VHs.*

Figure 9: *Screenshot of a densely packed crowd*

5. Results

Figure 9 shows an example of a densely packed crowd scene within the Wolfenbuettel model. In Figure 10 results for a typical walkabout in a lightly populated scene are shown. 50 VHs are in the scene, with a fixed high-resolution limit of 12 (24% of total). The user's walking speed is approximately twice that of the VHs, resulting in the high-resolution limit being exceeded several times. In figure 11, the walking speeds are equal, resulting in very few limit exceptions.

Figure 12 shows what happens when turning anti-clockwise in a very densely populated scene (far beyond the required number for most populated scenes). The positions of the obstacles can be clearly seen, and it is clear that the low-resolution VHs are being pushed. The group circled in the figure are potentially in view. Very densely populated scenes do tend to suffer with VHs bunching together on the periphery of the obstacles.

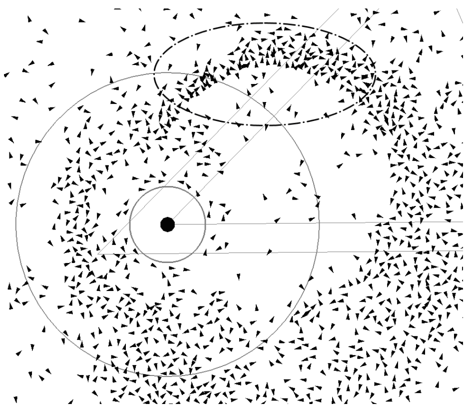

Figure 12: *Turning anti-clockwise*

6. Conclusion

The methods presented in this paper have shown promise in the test situations. The results indicate a more consistent number of people within the high resolution boundary. Thus in the full virtual tour framework it translates to a more consistent polygon count due to the avatars in the scene.

The extension to the Boids framework presented has introduced very little performance overhead. However in a system which relies on pre-processing the scene to achieve real-time results, consideration would be needed to handle the dynamic objects we use to influence crowd behaviour.

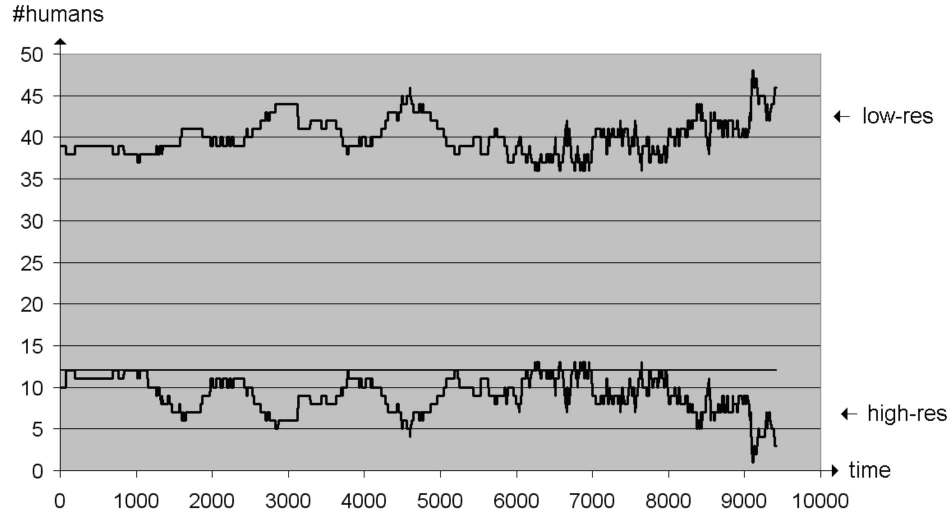

Figure 11: *Walkabout in lightly populated scene. Navigation speed is equal to the maximum speed of the VHs.*

7. Future Work

In Figure 13 an extreme case scenario is shown, the user is walking into a dead-end alleyway, with the high-resolution threshold cutting off the exit for the VHs. In this example the VHs are unable to move away from the user. As the user moves further up the alleyway, the VHs become more compressed and thus a high proportion of them will become high resolution. Short of culling or merging VHs, no simple solution presents itself to prevent this problem. The layout of scenes needs to be carefully considered with exit paths provided.

Presently the virtual humans have a limited understanding of their environment; they do not understand differences between, for example, roads and pavements. Extensions to the VH's understanding of the environment, similar to Marchal and Loscos [ML02] would aid the realism.

In our system people are always on the move. This is unrealistic of natural crowd behaviour and thus an aim is to add some loitering rules to the framework. However, when a VH is in a static location, we lose the option of deviating them away from the camera to maintain our low:high resolution mix. Thus we would have to return them to an automotive state.

Additionally, the behaviour of the crowd is not as realistic as we would have liked when they are being affected by the camera obstacles (see Figure 12). This is due to our decision to use the OpenSteer framework and is one feature we would change in any future implementations. The ideal solution is motion prediction of the virtual humans, which would allow for detection collisions earlier and provide more realistic motion.

We have assumed all scenes are flat, or sufficiently flat, to be able to ignore the problems associated with hills and other contours. The path planning does not take the incline as a factor and thus unnatural paths may be generated. Adding rules for ascending and descending different gradient slopes is a possible extension.

7.1. Acknowledgements

EPOCH - Excellence in Processing Open Cultural Heritage EU Framework 6 network

CHARISMATIC - Cultural Heritage Attractions featuring Real-time, Interactive Scenes and Multi-functional Avatars as Theatrical, Intelligent Characters EU Framework 5 project.

References

[ABT99] AUBEL A., BOULIC R., THALMANN D.: Lowering the cost of virtual human rendering with structured animated impostors. *WSCG, Plzen, Czech Republic* (1999).

[ABT00] AUBEL A., BOULIC R., THALMANN D.: Real-time display of virtual humans: Levels of detail and impostors. *IEEE Transactions on Circuits and Systems for Video Technology* (2000).

[BT98] BANDI S., THALMANN D.: The use of space discretization for autonomous virtual humans (video session). In *Proceedings of the second international conference on Autonomous agents* (1998), ACM Press, pp. 336–337.

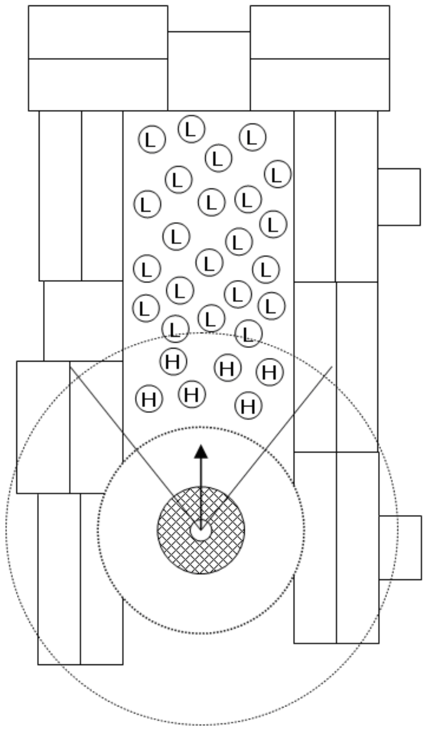

Figure 13: *Trapped*

[Che04] CHENNEY S.: Flow tiles. In *Eurographics/SIGGRAPH Symposium on Computer Animation* (2004), Boulic R., Pai D. K., (Eds.), Eurographics Association.

[EPO04] EPOCH: http://www.epoch-net.org/. European Research Network on - Excellence in Processing Open Cultural Heritage (EPOCH), 2004.

[FS93] FUNKHOUSER T. A., SéQUIN C. H.: Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques* (1993), ACM Press, pp. 247–254.

[HFDA03] HAVEMANN S., FELLNER D., DAY A. M., ARNOLD D. B.: New approaches to efficient rendering of complex reconstructed environments. In *Proceedings of the 4th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage (VAST-03) incorporating the 1st EUROGRAPHICS Symposia on Graphics and Cultural Heritage* (2003), Arnold D., Chalmers A.,, Niccolucci F., (Eds.), pp. 185–194.

[KL94] KAVRAKI L., LATOMBE J.-C.: Randomized preprocessing of configuration space for fast path planning. In *In Proc. IEEE Int. Conf. on Robotics and Automation* (1994), pp. 2138–2145.

[KO04] KAMPHUIS A., OVERMARS M.: Finding paths for coherent groups using clearance. In *Eurographics/SIGGRAPH Symposium on Computer Animation* (2004), Boulic R., Pai D. K., (Eds.), Eurographics Association.

[LD04] LAMARCHE F., DONIKIAN S.: Crowd of virtual humans: a new approach for real time navigation in complex and structured environments. *Computer Graphics Forum 23*, 3 (2004), 509–518.

[LTC01] LOSCOS C., TECCHIA F., CHRYSANTHOU Y.: Real-time shadows for animated crowds in virtual cities. In *Proceedings of the ACM symposium on Virtual reality software and technology* (2001), ACM Press, pp. 85–92.

[ML02] MARCHAL D., LOSCOS C.: *Simulating pedestrian crowd behaviour in virtual cities*. Tech. rep., UCL, July 2002.

[Ope04] OPENSG: http://www.opensg.org/. 2004.

[Rey87] REYNOLDS C. W.: Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics 21*, 4 (1987), 25–34.

[Rey99] REYNOLDS C.: Steering behaviors for autonomous characters, 1999.

[RW95] REIF J., WANG H.: Social potential fields: A distributed behavioral control for autonoomous robots. In *International Workshop on Algorithmic Foundations of Robotics (WAFR)* (1995), K. Goldberg D. Halperin J.-C. L., Wilson R., (Eds.), pp. 431–459.

[TC02] TECCHIA F., CHRYSANTHOU C. L. Y.: Visualizing crowds in real-time. *Computer Graphics Forum 21*, 4 (November 2002), 753–765.

[TLCC01] TECCHIA F., LOSCOS C., CONROY R., CHRYSANTHOU Y.: Agent behaviour simulator (abs): A platform for urban behaviour development. GTEC'2001, January 2001.

[UdHCT04] ULICNY B., DE HERAS CIECHOMSKI P., THALMANN D.: Crowdbrush: Interactive authoring of real-time crowd scenes. In *Eurographics/SIGGRAPH Symposium on Computer Animation* (2004), Boulic R., Pai D. K., (Eds.), Eurographics Association.