Automatic High Level Avatar Guidance Based on Affordance of Movement

Despina Michael and Yiorgos Chrysanthou

Department of Computer Science, University of Cyprus

Abstract

As virtual cities become ever more common and more extensive, the need to populate them with virtual pedestrians grows. One of the problems to be resolved for the virtual population is the behaviour simulation. Currently specifying the behaviour requires a lot of laborious work. In this paper we propose a method for automatically deriving the high level behaviour of the avatars. We introduce to the Graphics community a new method adapted from ideas recently presented in the Architecture literature. In this method, the general avatar movements are derived from an analysis of the structure of the architectural model. The analysis tries to encode Gibson's [7] principle of affordance, interpreted here as: pedestrians are more attracted towards directions with greater available walkable surface. We have implemented and tested the idea in a 2x2 km^2 model of the city of Nicosia. Initial results indicate that the method, although simple, can automatically and efficiently populate the model with realistic results.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Animation

1. Introduction

During the past few years the modelling of virtual cities has become widespread in games, entertainment and architectural applications. Populating such models, with autonomous characters, is a problem that is still under research. A system that can realistically simulate the behaviour of pedestrians and at the same time have a good visual quality real-time rendering can help the user to extract conclusions with observation of virtual world, and use these conclusions to optimize the sector for which the model has developed.

The behaviour simulation of the avatars can be separated into high-level behaviour, involved mainly with what the avatars want to do and what direction they want to follow; and the more low level processes such as path planning or interaction with other avatars (including collision avoidance) [21]. In this paper we will mainly be concerned with the high level decisions. We have implemented a complete system with 3D avatar rendering, collision avoidance and a very effective path following method to allow us to test our ideas but our main contribution nevertheless remains the high level guidance.

The main aim of the research presented here is to provide a method that would allow us to take any 3D urban model, whether representing a reconstruction of a real city or something entirely computer generated [14], and simulate the behaviours of the inhabiting entities with no user intervention. We want the result to be fairly realistic, which for the model of a real scene, that would mean that the resulting densities of pedestrian should match to a certain degree the observed ones, but also for an imaginary model the densities should be such that would match the expectations of an expert architect. Therefore we based our ideas on work coming from Architecture [24]. We use a perceptual model based on Gibson's principle of "affordance" [7] which can be interpreted as: a person is more likely to walk in directions that afford more space for movement. It is interesting to note that although it is a perceptual approach, and thus each decision depends on what the avatar is looking at the time, we can pre-process the geometric structure of the scene and compute information which can be used to guide the decisions of the avatars essentially eliminating any on-line cost.

The strength of our approach lies in the following attributes:

- It is automatic, requiring no user input.
- It is very fast, the on-line decisions just involves a stochas-

tic selection from a set of pre-computed values at the road junctions.
- It gives fairly realistic results. We can not claim to have absolute reproduction of the real pedestrian flow, that is not our goal, but the method gives the right "feel" to a virtual scene.
- Ideas from this approach can be extended to apply to other urban groups such as cars, bicycles, etc.
- In the same framework, we could encode places of interest or attractors (monuments, bus stops, benches, "jugglers") for a more fine tuned behaviour.

2. Previous Work

There is a large body of research on human behaviour simulation in virtual environments [3]. Here we are only concerned with behaviour that is applicable to large populations of virtual humans, as would be required for a city.

Reynold [17] in his classic paper presented one of the first group behaviours, the Boids. This method tackles mainly the problem of a group of entities moving together, pursuing a common goal. Although a very successful approach, it is not directly applicable to a set of individuals, each going their own way. Bouvier and Cohen [4] implemented a simulation of a crowd based on a particle system. The human "particles" are assigned a state and a reaction function that govern their behaviour within the system. The model is scalable but the behaviours are quit simple. S.R. Musse and D. Thalmann [13] use a more sociological approach. To enable more human-like reactions, each agent is specified by a level of dominance, a level of relationship and an emotional status and is ruled by seeking and flocking laws.

The work of Thomas and Donikian [22] and of Kallmann and Thalmann [11] has some relation to our work since in that they too store information in the environment to help guide the avatars. In [22] the agents are guided by using a combination of visibility graphs that link points of interest, and Voronoi diagrams that provide routes around those points. Some of the ideas do begins to address the interaction between humans and objects as described by Gibson. However, compared to our approach, it is more complex and not entirely automatic.

Zacharias [26] provides a comprehensive review of many techniques suitable for transportation planning. Some of these could also be relevant for real-time agent simulations. Helbing and Molnar's [8] granular physics agents move according to predetermined flows in simple systems, but have attractive emergent properties, such as forming lanes when two sets of people walk in opposite directions. Other systems of note include PEDFLOW [12], where agents are programmed with short range vision to guide themselves around obstacles, and the fire-evacuation program EXODUS [6]. The addition of visual fields to agents is a recent phenomenon, thus although most humans would seem to use vision as there primary guiding mechanism, agent models have used gravitational fields or fixed routes between points to manoeuvre. The problem has been to find a way of adding a complete visual system to agents, and as a result either the environment is simplified [20] or the available routing is simplified [22]. Despite this, there have been a number of approaches trying to utilise the visual field for movement in a complex environment, going back to Penn and Dalton [15]. Such approaches tend to follow a form of "natural movement", that is, general, non-goal specific movement, based on the assumption that people follow their line of sight [9]. They applied analytic techniques and discovered that the numbers of people found in urban environments appeared to follow some form of natural movement. Following that, Penn and Turner [16] considered using an 'exosomatic visual architecture' (EVA). They applied a grid over the environment and precalculated the visibility connections between all pairs of locations in the grid, and used this as a look-up table to find possible visible next destinations for agents. Turner and Penn [24] demonstrated that using an EVA and simple natural movement rules, human-like behaviour can be produced inside a building.

3. Affordance Based Guidance of the Avatars

Our method builds on the ideas presented in the paper of Turner and Penn [24], where Gibson's theory of affordances [7] is revisited. According to Gibson the agent perceives the contents of the environment and uses affordances within it to guide its actions without reference to superior representational models. He asserts that the available walkable surface affords movement. To quote from [24]: "When engaging in natural movement, a human will simply guide herself by moving towards further available walkable surface. The existence of walkable surface will be determined via the most easily accessible sense, typically her visual field".

We have implemented the above idea in practice as a simple rule set that makes avatars more likely to walk in directions where they have the "larger" views. Of course humans are influenced by many other factors and hence this approach cannot hope to provide a complete model of all human pedestrian behaviour. However, as can be seen by the results, the rule set is simple but effective.

Ideally each avatar should be given synthetic vision and be allowed to take decisions in every step [23]. This is not a scalable approach though. In our case, since we are dealing with pedestrians that walk on specified lanes - pavements, roads (we will discuss open spaces such as parks later) - we can assume that they will change direction only at the junctions and thus only there do they need to have vision to guide their decisions. In fact we can pre-compute the visibility information at the junctions and store it with the model. Any other deviation in between the junctions, for example to greet a friend or avoid a collision will be taken care by the more low level simulation.

The system works in two steps. The preprocessing, which computes the visibility at the junctions as mentioned above, and the on-line guidance of the avatars. We will now see the two steps in some more detail.

3.1. Preprocessing the Environment

As a very first step we need to analyze the model, to identify the street junctions and connect them with their neighbours, in the form of a graph. In our particular implementation this was trivial since the city data came from a GIS system which also provided the centrelines of the roads. If this data is not available it is still possible to compute the junctions from the road geometry, although with a more involved method [5].

At the centre point of each junction we have to compute the available walkable surface in each direction. For this we make use of the visibility polygon (or viewshed). The visibility polygon is a star shaped polygon centred at the viewpoint, with edges the visible parts of the objects and whose interior intersects no object. There are well known methods in Computational Geometry for the computation of such structures, even if the scene is dynamic [10]. We can see an example of the visibility polygon of a junction in Figure 1.

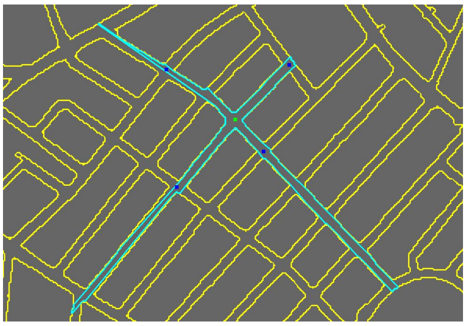

Figure 1: *The visibility polygon of the junction (green square) is shown with a cyan line. The blue squares are the immediate neighbour junctions*

Once we have the viewshed we can use it to compute the visible area in the direction of each of the junctions' neighbours. We estimate the visible area by taking the part of the viewshed that lies within the view-frustum when looking towards the direction of the specific neighbour. In the example of Figure 2, we used a view-frustum of 110 degrees. That portion of the viewshed is then triangulated and its total area is computed as the sum of the triangle areas.

At the end of this process, all we need to store at each junction is the percentage of walkable (or visible) area that corresponds to each direction, and the actual vector that indicates the direction.

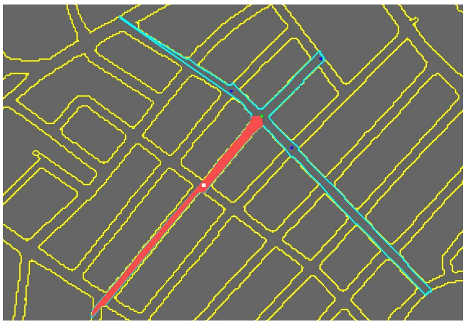

Figure 2: *Red area defines the visibility of green junction towards white junction*

3.2. On-line Avatar Guidance

For each frame, to move an avatar a step forward we need first to test for collision, both against the static environment and against the other avatars. Then, depending on whether the avatar is along a road or at a junction, we either follow the curve of the road or possibly take a decision on where to turn.

To accelerate the process we discretize the ground plane of the environment into cells which are stored into an array [19] we call the "Guidance Map". Each cell of the Guidance Map can have the value of either "Building", "Junction", or "Road". Any cell flagged as "Building" is inaccessible and the avatars need to avoid colliding with it. "Junction" cells, are those cells that lie within a small radius of the centre of a junction. For all "Road" cells, we keep the direction of the road at that point, which is derived by the direction of its centreline at that part of the road. "Junction" and "Road" cells can also have "Pedestrian" flag to indicate if an avatar is occupying them.

While the avatars are in "Road" cells, they walk along the road and, assuming there is no collision, they adopt the direction of the respectively cell. When they come up to a "Junction" cell, they take a decision on changing their direction. This decision depends on the percentages of the visibilities computed at the preprocessing. The greater the visibility towards a neighbour, the greater the probability that the neighbour will be chosen as the next target. At this point, some kind of memory is needed. Avatars need to remember which junction they have just come from and that is not considered to be a candidate, even if it has the greatest visibility of all neighbour junctions. Once the decision is taken, and while the avatars are in "Junction" cells, they change their direction gradually, taking into account their current direction and the direction towards the start of the next road. When avatars reach the "Road" cells of the new road, they continue their journey by just following the directions stored in them, as described above.

Irrespective of whether an avatar is on a "Road" or a

"Junction" cell, collision detection needs to be performed for each step. We examine all cells lying in a region about 2 meters in length, in front of the avatar. If an imminent collision is detected, the avatar's direction is changed with a small rotation in order to avoid the obstacle. After the obstacle is avoided, the avatar gradually goes back to following the "Road" vector again.

4. Rendering of the Avatars

There is a great deal of literature on rendering of complex static models [1], however not much has been published regarding the rendering of densely populated environments. The standard techniques employing level-of-detail and occlusion culling are usually not adequate. There are many situations when even after culling, a large number of avatars might be still visible and even simplified versions of a human model will be expensive in the required numbers. Alternative representations based on images [19, 18, 2] or points [25] show more promise.

Our system is based on the work of Tecchia et. al. [18]. For the rendering of each avatar a single image is used. The images are computed and stored as compressed textures at preprocessing and selected on-line based on the viewing position. Per pixel lighting can also be used for dynamic shading, if the GPU capabilities allow it.

5. Results

To test the method we run a set of experiments with a simple model of central Nicosia. As we see in Figure 3, the town is separated into the old part, at the top of the image, which lies within the Venetian walls and has small narrow streets, many of which come to dead ends due to the barracks dividing the town, and the new part which has a longer, wider and better connected network.

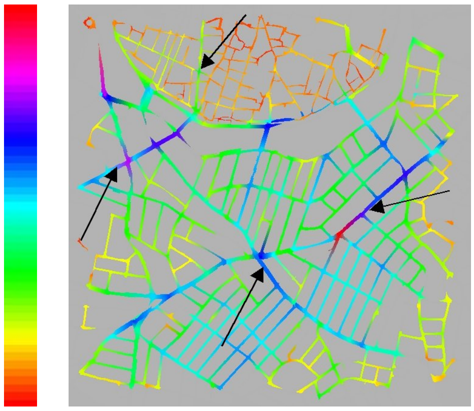

Figure 3: *Initialising the avatars based on the overall viewshed size at each junction* [9]

We compared the method described in the paper (*affordance*) with one that chooses a random direction with equal likelihood for every avatar that reaches a junction (*random*). Initially, for both methods, the avatars are distributed randomly in the streets of the whole model. We let the simulation run over a certain number of frames while counting how many avatars go through each junction at every step. We use colour to show the densities at the junctions. Roads values, are shown by interpolating along junctions values at either end. Orange and yellow is used for the lower end of the scale while darker blue and red is for the higher end.

In Figure 4(left), we see the results of the *random* simulation after 500 frames. The *affordance* method looks similar at that stage since they both start from a random distribution. If we give them enough time to converge though, we begin to see the difference. The *random* method appears to break the town into two sectors the old and the new. The distribution within each sector come out fairly even for most roads, although there is a big difference between the two sectors. The reason for this is probably the difference in connectivity of the two parts as described earlier. For the *affordance* method the results are much better. We can see that certain streets have a lot more density than others and these seem to match quite closely the main shopping streets of the town as it stands today. In Figure 3 we show with the black arrows the 4 main streets of the particular part of town.

One improvement we can do is the following: instead of starting off the avatars at random positions and wait for hundreds of frames for them to converge we can compute the visibility polygon at the junctions and use the overall area of that to guide us for specifying the distributions directly from the first frame. That is, near junctions with larger visibility polygon we place more avatars. This idea follows some of the theories described in [9]. We can see that actually the results of running the *affordance* for 5000 frames (Figure 4(right)) and that of using the visibility polygon at the junctions (Figure 3) are not too different.

In Figure 5 we show the results of two runs of the *random* method with 5000 frames. In the left we employed a quarter of the avatars than we did in the right. We can see that the number of avatars does not change the results in a significant way. This is true for both methods.

We implemented the 3D system and run it on a Pentium 4, 1.8GHz CPU, with 256MB RAM and a GeForce4 Graphics Card with 64MB of memory. For the scene with 10000 moving avatars it took a bit less than a second per frame. This time includes the rendering, the high level behaviour and the collision processing. However, by far the greatest cost (about 80% of the total time) was the collision avoidance.

For new results see project web page at http://www.cs.ucy.ac.cy/Projects/create/papers/

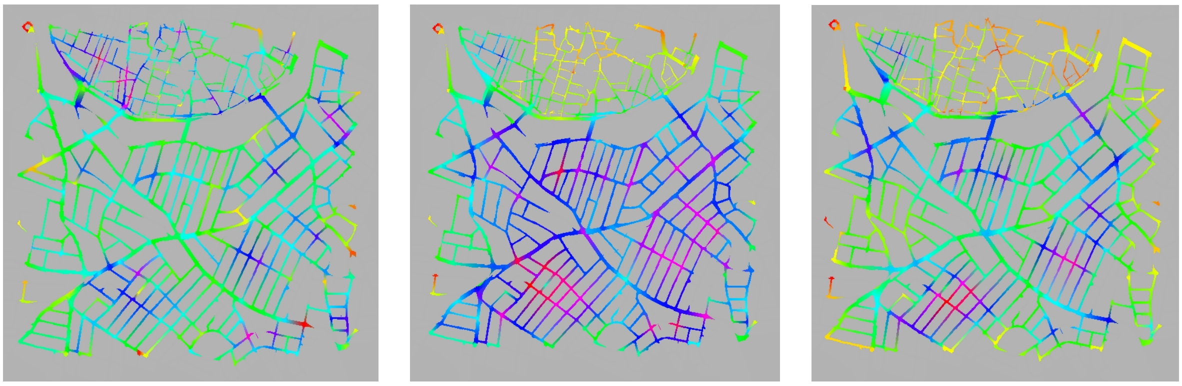

Figure 4: *Traffic counts of 10000 avatars after 500 frames (left), after 5000 frames in the random selection method (middle) and after 5000 in our method (right)*

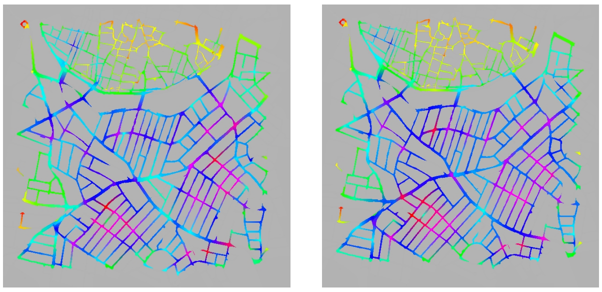

Figure 5: *Running the methods with a different number of avatars only changed the intensity (i.e. the absolute numbers) but not the relative distribution. In this example we see the random method with 2500 (left) and 10000 (right) avatars over the same number of frames*

Figure 6: *A screen shot of the system running with the 3D avatars*

6. Discussion and Conclusion

In this paper we described a new, for Computer Graphics, idea that allows us to take any urban model and populate it with minimal user intervention, as far as the high level behaviour is concerned. Although this is a very preliminary implementation we still observe realistic pedestrian movements and high speed.

The method is easily extended to dynamic environments. When there is a change in the environment, for example by adding a new road or adding a road block, we can quickly update the stored information. At the point where the change has happened, we can compute the visibility polygon, and check which of the neighbouring junctions lie within it. For these ones the visibility information is no longer valid and we would need to recompute it. Everything else remains unchanged. Since we can compute a visibility polygon in a matter of a few milliseconds (<5) and a change is likely to affect only junctions in the close vicinity, we expect the whole process to run almost unnoticed by the user.

To make the results more realistic and take in account some of the other factors affecting human behaviour, the system could be extended with the addition of attractors. Pedestrians will then take their decision for the next junction not only based on visibility, but also based on wether an attractor is visible from their current position and the characteristics of the attractors.

There still remains the issue of populating open spaces that have no junctions where the decisions can be taken. In such cases we intend to use a method closer to that of [24] where the visibility information is computed for every point in the 2D space and a new decision is taken every few steps. This of course has the problem that it is very difficult to prevent the avatars from taking wondering paths. A finer tuning will be required.

Acknowledgements

We would like to thank Alasdair Turner of the Bartlett School of Architecture for sharing with us his expertise in pedestrian movement, Franco Tecchia and Celine Loscos for letting us use the impostor based crowd rendering and Soteris Stylianou for his help with our implementation. This work was partly financed by the EU IST-2001-34231 CREATE Project.

References

1. D. G. Aliaga and D. Manocha. Interactive walkthroughs of large geometric datasets. In *ACM SIGGRAPH*, page Course 18, July 2000. 4
2. A. Aubel, R. Boulic, and D. Thalmann. Lowering the cost of virtual human rendering with structured animated impostors. In *Proceedings of WSCG 99*, Plzen, Czech Republic, 1999. 4
3. R. Ayelett and M. Cavazza. Intelligent virtual environments. State-of-the-art report at Eurographics 2001. 2
4. E. Bouvier, E. Cohen, and L. Najman. From crowd simulation to airbag deployment: particle systems, a new paradigm of simulation. *Journal of Electronic Imaging*, 6(1):94–107, 1997. 2
5. X. Décoret and F. Sillion. Street generation for city modelling. In *Architectural and Urban Ambient Environment*, 2002. 3
6. E. R. Galea. Simulating evacuation and circulation in planes, trains, buildings and ships using the EXODUS software. In M. Schreckenberg and S. Sharma, editors, *Pedestrian and Evacuation Dynamics*, pages 203–226. Springer-Verlag, Heidelberg, Germany, 2001. 2
7. J. J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, Boston, MA, 1979. 1, 2
8. D. Helbing and P. Molnár. Self-organization phenomena in pedestrian crowds. In F. Schweitzer and H. Haken, editors, *Self Organization of Complex Strutures: From Individual to Collective Dynamics*, pages 569–577. Gordan & Breach, Amsterdam, 1997. 2
9. B. Hillier, A. Penn, J. Hanson, T. Grajewski, and J. Xu. Natural movement: or configuration and attraction in urban pedestrian movement. *Environment and Planning B: Planning and Design*, 20:29–66, 1993. 2, 4
10. S. Hornus and C. Puech. A simple kinetic visibility polygon. 3
11. M. Kallmann and D. Thalmann. A behavioral interface to simulate agent-object interactions in real-time. In *Computer Animation*, pages 138–146. Computer Society Press, 1999. 2
12. J. Kerridge, J. Hine, and M. Wigan. Agent-based modelling of pedestrian movements: The questions that need to be asked and answered. *Environment and Planning B: Planning and Design*, 28(3):327–341, 2001. 2
13. S. R. Musse and D. Thalmann. A model of human crowd behavior: Group inter-relationship and collision detection analysis. In *Workshop of Computer Animation and Simulation of Eurographics '97*, pages 39–52, Budapest, Hungary, 1997. 2
14. Y. I. H. Parish and P. Müller. Procedural modeling of cities. In *ACM SIGGRAPH 01*, pages 301–308, Los Angeles, CA USA, 2001. 1
15. A. Penn and N. Dalton. The architecture of society: Stochastic simulation of urban movement. In N. Gilbert and J. Doran, editors, *Simulating Societies: The Computer Simulation of Social Phenomena*, pages 85–125. UCL Press, London, 1994. 2
16. A. Penn and A. Turner. Space syntax based agent models. In M. Schreckenberg and S. Sharma, editors, *Pedestrian and Evacuation Dynamics*, pages 99–114. Springer-Verlag, Heidelberg, Germany, 2001. 2
17. C. W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. In M. C. Stone, editor, *Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21, pages 25–34, July 1987. 2
18. F. Tecchia, C. Loscos, and Y. Chrysanthou. Image-based crowd rendering. *IEEE Computer Graphics and Applications*, 22(2):36–43, March-April 2002. 4
19. F. Tecchia and Y.Chrysanthou. Real-time visualisation of densely populated urban environments: a simple and fast algorithm for collision detection. In *Eurographics UK*, Apr. 2000. 3, 4
20. D. Terzopoulos. Visual modelling for computer animation: Graphics with a vision. *Computer Graphics*, 33:42–45, 1999. 2
21. D. Thalmann. Challenges for the research in virtual humans. Autonomous Agents 2000, Barcelona, Spain, June 2000. 1
22. G. Thomas and S. Donikian. Modelling virtual cities dedicated to behavioural animation. *Computer Graphics Forum*, 19(3):C71–C80, 2000. 2
23. X. Tu and D. Terzopoulos. Artificial fishes: Physics, locomotion, perception, behavior. In *SIGGRAPH '94*, pages 43–50, July 1994. 2
24. A. Turner and A. Penn. Encoding natural movement as an agent-based system: an investigation to human pedestrian behaviour in the built environment. *Environment and Planning B: Planning and Design*, 29, 2002. 1, 2, 5
25. M. Wand and W. Straßer. Multi-resolution rendering of complex animated scenes. *Computer Graphics Forum*, 21(3), 2002. Eurographics 2002. 4
26. J. Zacharias. Pedestrian behavior and perception in urban walking environments. *Journal of Planning Literature*, 16(1):3–18, 2001. 2