

Scalable Pedestrian Simulation for Virtual Cities

Soteris Stylianou
Department of Computer
Science
University of Cyprus
Nicosia, Cyprus
ssotos@cs.ucy.ac.cy

Marios M. Fyrillas
Department of Mechanical
Engineering
Frederic Institute of
Technology
Nicosia, Cyprus
eng.fm@fit.ac.cy

Yiorgos Chrysanthou
Department of Computer
Science
University of Cyprus
Nicosia, Cyprus
yiorgos@ucy.ac.cy

ABSTRACT

Most of the common approaches for the pedestrian simulation, used in the Graphics/VR community, are bottom-up. The avatars are individually simulated in the space and the overall behavior emerges from their interactions. This can lead to interesting results but it does not scale and can not be applied to populating a whole city. In this paper we present a novel method that can scale to a scene of almost any size. We use a top-down approach where the movement of the pedestrians is computed at a higher level, taking a global view of the model, allowing the flux and densities to be maintained at very little cost at the city level. This information is used for stochastically guiding a more detailed and realistic low level simulation when the user zooms in to a specific region, thus maintaining the consistency.

At the heart of the system is an iterative method that models the flow of avatars as a random walk. People are moved around a graph of nodes until the model reaches a steady state which provides feedback for the avatar low level navigation at run time. The Negative Binomial distribution function is used to model the number of people leaving each node while the selected direction is based on the popularity of the nodes through their preference-factor. The preference-factor is a function of a number of parameters including the visibility of a node, the events taking place in it and so on.

An important feature of the low-level dynamics is that a user can interactively specify a number of intuitive variables that can predictably modify the collective behavior of the avatars in a region; the density, the flux and the number of people can be selectively modified.

Categories and Subject Descriptors

I.3.7 [Computer Graphics]: Animation

General Terms

Algorithms

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

VRST'04, November 10-12, 2004, Hong Kong.

Copyright 2004 ACM 1-58113-907-1/04/0011 ...\$5.00.

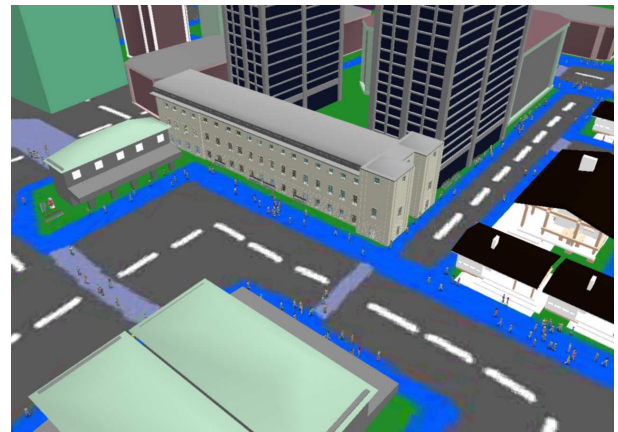


Figure 1: The addition of avatars with seemingly correct behavior greatly improves the appearance of an urban model.

Keywords

Avatars, Pedestrian simulation, Animation

1. INTRODUCTION

Large virtual urban models, such as those of cities, have been available for several years and are now often used in Virtual Reality applications. However, we rarely see them populated with more than a few avatars which has an impact on their realism and consequently can limit their usefulness. Traditionally one of the main reasons for their sparse population had been the prohibitive cost of rendering the virtual humans. However, recently, the use of modern graphics hardware and acceleration techniques (image and point based rendering [23, 28], visibility culling [4] and levels of detail [15]) have helped in populating urban models. Nevertheless, the realistic simulation of avatar behavior can still be an issue.

Most of the common approaches used for avatar simulation are bottom-up. The avatars are individually simulated, step by step at each frame requiring a super-linear number of operations to move them around. Even if some of the methods can lead to visually pleasant results, they simply do not scale to the required numbers for populating a whole city. In this paper we describe a novel top-down approach for describing the high level behavior of avatars in a city that

is both scalable and requires minimal user input. By high level we refer to the decisions of the avatars on where they want to go and what they want to do, how long to spend in an area and how fast they should move. Other issues such as avoiding collision, being momentarily distracted by a shop window, or greeting a friend are all part of the low level behavior.

The proposed technique operates in two levels, the macroscopic (or *high*) level and the microscopic (or *low*) level. At the macroscopic level we have a global view of the model. Since large populated scenes are very complex environments the particular actions of most individuals are of little importance. What is significant is the collective behavior, i.e. the average (ensemble) behavior of a large number of people, see Figure 1. Thus, we do not attempt to simulate specifically each and every avatar but rather we employ a statistical approach. On the other end, at the microscopic level, where the viewer might be down at the ground looking at a few avatars, the detail and individuality are important.

In the macroscopic level we estimate the flows and distributions of avatars using the layout of the urban environment, the functionality of each space and any other related information provided by the designer, such as an event taking place somewhere. The flows and distributions are computed at start-up using an iterative method, which usually takes a few seconds to converge. Furthermore, it allows the user to refine the behavior within any region through the use of some intuitive parameters. These will affect primarily the immediate region and to a lesser extent the entire model. The user can have precise control on the number of people in a region with the use of two tools: the preference factor and the probability distribution of each node. The former can simulate the addition of an attractor or an event or a public transport hub and will change the flow of people towards the node. On the other hand, the latter can simulate the change in usage of a node such as a transformation from a business street to a commercial street and will leave the influx of people mostly unchanged but modulate the residence time of each passing avatar in the region, thus increasing (or decreasing) their overall number.

The microscopic level uses the information computed above to guide the avatars at run time. Here, the characteristics of each individual can be taken into account. Avatars will be given the appropriate speed depending on whether they are young or old, in a rush or not. They will be instructed to take a direct path through an area or wander around for a while, depending on whether they are in a hurry to their work or if they are casual passers, and so on. However, these decisions are weighted based on the solution of the high level in such a way as to maintain the distributions and flows.

The two-level simulation provides several advantages over the purely avatar based approaches. For example, the low level behavior needs only be employed for avatars within the visible part of the city - or a small superset of it. In the non-visible areas, the avatars can be culled away since they can be easily regenerated from the high level data if a specific area is to come into view. The savings in the processing cost can be invested for a more realistic low-level simulation. It is scalable in both the number of avatars and the extent of the urban model, since the on-line requirements are almost negligible. The pre-processing is in the order of seconds for a model with hundreds of roads and junctions, as the one shown in the results.

1.1 Contribution

In brief these are some of the most important contributions of the proposed approach compared to previous methods:

Scalability The macroscopic simulation can be computed in a matter of seconds at start-up and be used at minimal cost at run-time

Efficiency The microscopic simulation needs only be run for a small superset of the visible regions, therefore it can be made as smart and as complicated as necessary

Ease of use If an automatic method for partitioning the model into regions is used, then it can provide an initial, fairly realistic, solution with no user intervention

Control The user can precisely control the density or the flow of pedestrians in each region, with the use of two intuitive parameters.

2. RELATED WORK

Generating realistic behavior for virtual humans has been the subject of considerable previous research in the wider Computer Graphics community [1]. A large part of this work concentrates on producing detailed realistic behavior [26] that can be useable in scenes with just a handful of avatars but would be inefficient to be supported in applications with large populations of virtual humans, such as a city.

Reynolds [21] in his classic paper presented one of the first crowd behaviors, 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 large set of individuals, each going their own way. It can also become costly. Bouvier and Cohen [2] 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 behavior within the system. The model is scalable but the behaviors are quite simple and cannot easily account for the individual characteristics of avatars.

Some researchers used a more sociological approach [18]. To enable more human-like reactions, agents are specified by a level of dominance, a level of relationship, an emotional status and are guided by seeking and flocking laws. The work of Thomas and Donikian [25] and of Kallmann and Thalmann [12] store information in the environment to help guide the avatars. In [25] 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.

There is relevant work in other fields. Zacharias [30] 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 [10] 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 [13], where agents are programmed with short range vision to guide themselves around obstacles, and the fire-evacuation program EXODUS [7].

Concepts from Fluid Mechanics have been widely used to model pedestrian flow dynamics and traffic flow [9, 14]. The

Cellular Automaton (CA) model has also been widely used [6, 5, 29] because is simple and flexible and provides qualitative insight on the transition between laminar flow and start-stop waves. Related to this model is the lattice gas models with biased random walkers [22, 17]. Both, CA and lattice gas models, can be understood to represent the flow in the Lagrangian frame of reference where each avatar's movement is explicitly computed. However, none of the above is really applicable to real-time graphics. They either too slow and static or have difficulties in accounting for the characteristics of the individual avatars.

An interesting idea that appears recurrently is the addition of visual fields to the agents. Adding a complete visual system [24] is hard and expensive however, there have been a number of approaches trying to utilize the visual field for movement in a complex environment[19]. 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 [11]. In [11] 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 [20] 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 [27] demonstrated that using an EVA and simple natural movement rules, human-like behavior can be produced inside a building.

To summarize, the proposed method is probably the first top-down approach in the Computer Graphics literature that can take on models of such scale. Although large scale top-down methods have been proposed in other fields they are geared towards predicting accurate flows rather, rather than producing a realistic, user controlled, real-time VR simulation.

3. OVERVIEW

The method starts by reading in a three-dimensional urban model. This is analyzed to identify the different walkable regions, which are classified into roads and junctions/open spaces. All of these regions are treated the same, with each represented by a node in the city graph (Figure 2).

The high level layer operates on the graph to find the steady-state distribution and flows of the avatars in the model. This is an iterative method, computed at pre-processing but can also be updated at run-time whenever the user introduces a change that can influence the solution. Such a user interaction might be the addition/removal of a pedestrian crossing or road, the modification of the parameters related to a node, the triggering of an event such as a concert or a terrorist alert.

Once the high level information is computed then the individual avatars are introduced into the system. The low-level simulation navigates the avatars stochastically while the high-level indicates which direction the avatar should take once it has reached a node. The individual characteristics of an avatar can be accounted for in the low level by adding the appropriate bias to the stochastic selection.

4. MODELLING PEDESTRIAN BEHAVIOR

As mentioned above, our system is made up of a network of nodes that describe the entire walkable area of a city. To distinguish between people leaving and arriving at a node we establish two connections (streams) from node to node. Each stream allows people to move only in one direction (Figure 2), either fore or back. For example, $\Phi(t, 1, 2)$ denotes the number of people that during the t -th time step have crossed from node 1 to node 2 and similarly, $\Phi(t, 2, 1)$ denotes the number of people that have crossed from node 2 to node 1. Pedestrians entering a node have to take two de-

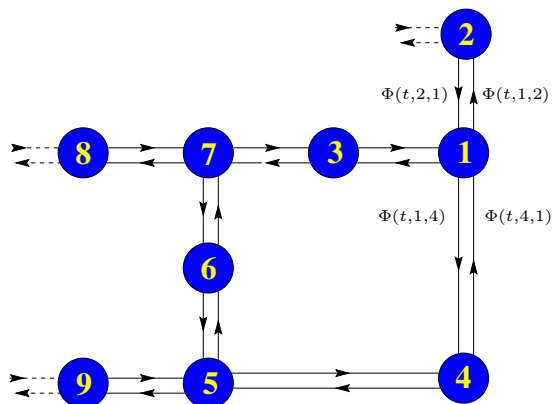
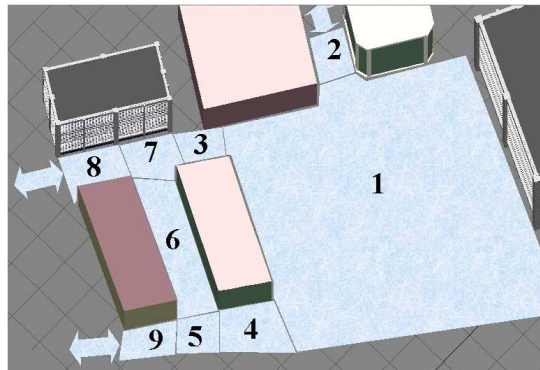


Figure 2: An example on how a nodal diagram is constructed. At the top we see the partitioned three-dimensional environment while at the bottom we see its graph. Nodes 2, 8 and 9 are boundary nodes. In the truncated system they represent sources and sinks of avatars.

isions; when they should leave and where they should go. The former is decided based on a probability distribution function $f(x)$ (Equation 1) and the latter on the preference factor PR (Equation 4).

4.1 When to leave a node

An avatar entering a node has to spend a minimum amount of time in that node, depending on its length, even if he walks straight to the nearest exit. Avatars may also stop at attractions or wander around in an open space, thus increasing the total time they spend in a node. These two observations are related to the random walk problem [3]. The random walk phenomenon is also related to the Brownian motion in physics and it describes the motion of molecules

in gas which causes diffusion. In one dimension, Brownian motion can be regarded as a series of jumps of the same size and with an equal probability of moving forward or backward. The probability that a particle will be at a certain position after N jumps is obtained by the Binomial Distribution. Alternatively, the probability that a particle will be within an identified region during a certain period of time is related to the Negative Binomial Distribution. In general, the Negative Binomial Distribution is associated with problems that deal with the length of time before some event occurs; for example the length of time a customer must wait in line until receiving service or the length of time until a piece of equipment fails [16].

The Negative Binomial distribution is used in the random walk problem to describe the probability of an avatar exiting a node after a certain period of time or, alternatively, after a certain number of steps (time-steps). We chose to use this distribution in our simulation to decide when avatars should leave a node, since it relates best to our problem.

Given a minimum number of steps r an avatar has to take before exiting a node, and the possibility p that an avatar will wander inside the node rather than head towards the exit (at each step), we can then ask what the probability is for a pedestrian to leave a node after N movements. In the definition of movements we include both taking a step that brings him closer to the exit and a step that does not. The latter might be the case when moving towards an attraction to the side or just standing somewhere for time equal to the duration of a step. We will call these *wandering* steps. The probability of taking $r - 1$ steps and x wandering steps followed by a single step signifying the exit from the node, is:

$$f(x) = \binom{x+r-1}{r-1} p^r (1-p)^x, \quad (1)$$

where $\binom{x+r-1}{r-1}$ is the binomial coefficient. In this formulation is implied that the length of the node is $r - 1$ steps, that the number of total movements $N = x + r$ and that a wandering step has the same duration as a forward step. In Figure 3 we show the distribution function for different values of p but set value of r .

Instead of considering the distribution with respect to steps taken by a pedestrian, it is convenient to express the results in terms of time-steps. If we assume that the average time required for a pedestrian to take a step is $\delta t_p = \delta l_p / V_p$ (δl_p is the average length of a step and V_p is the average velocity of a pedestrian), then multiplying N by δt_p in Figure 3 would show the probability distribution of pedestrians leaving a node having entered in previous time steps. In particular, Figure 3 shows the percentage of pedestrians that would leave a node assuming that they have entered at the zeroth time-step. Evidently, there would be no pedestrians leaving the node in less than r time-steps. Note that as the p becomes smaller the distribution goes closer to a Gaussian.

The calculation of the total number of pedestrians that would leave a node in a given time-step is a convolution of the people that had entered the node from its neighbors during the previous time-steps. We distinguish between people arriving from different neighbors by assigning them in separate banks. This allows us to control the number of avatars returning back to the direction they came from. Currently we have it set to 0%.

Each bank maintains the people that have arrived from a certain neighbor over a number of time-steps. Calculating the number of people leaving is done per bank. Using the probability density function f (Equation 1), the number of pedestrians leaving the bank i can be calculated by a time convolution of the pedestrians that have previously arrived in the bank. Finally the convolution of people leaving from all banks is summed up.

If the number of pedestrians arriving at a node during the t -th time-step is denoted by $\Phi_{in}(t, i)$ then the number of people leaving the node during the t -th time step, denoted by $\Phi_{out}(t, i)$ can be obtained by:

$$\begin{aligned} \Phi_{out}(t, i) &= \\ & \sum_{m=r(i)}^{\infty} \Phi_{in}(t-m, i) \times \binom{m-1}{r(i)-1} p(i)^{r(i)} (1-p(i))^{m-r(i)} \\ & \equiv [\Phi_{in}(i) * f(i)], \end{aligned} \quad (2)$$

where $r(i)$ and $p(i)$ are different for each node. Evidently, the equation is consistent with the steady-state solution where we expect that $\Phi_{out} = \Phi_{in}$ for each node. The number of people present in each node at time t , denoted as $N(t, i)$ is calculated by a simple balance around a node:

$$N(t, i) = N(t-1, i) + \Phi_{in}(t, i) - \Phi_{out}(t, i). \quad (3)$$

4.2 Where to go

To determine which direction an avatar should go next, we assign a weight to each neighbor of the current node, the Preference Factor ($PR(i)$), which signifies the preference of a pedestrian to visit a particular node as compared to the other neighboring nodes. The $PR(i)$ can be arbitrarily complex encompassing all the factors that influence the decision of a pedestrian on choosing where to go; factors like what attractions there are, how well connected the road is, how directly it will take them (not many turnings) to popular destinations, or even how crowded the destination is. We have opted to use only two components for this implementation, the Visibility $VIS(i)$ and the Functionality $FUN(i)$:

$$PR(i) = \alpha(i)VIS(i) + FUN(i). \quad (4)$$

The former is based on ideas from the work of [11] and [8], i.e. that the longer a road is the more busy it gets. The magnitude is computed by performing a viewshed calculation from the center of the node and is then modulated using the global factor α . The latter is related to the commerciality/attractiveness of the node and is an input to the model or can be specified interactively by the user.

Once we compute the number of pedestrians that will leave a node, the percentage of pedestrians allocated to each neighboring node is calculated based on the ratio of the PR factor of each neighboring node divided by the sum of the PR factors of all neighboring nodes. There are two possible scenarios: The pedestrians are allocated in each node irrespective of the node visited in the past or whether this node is not considered as a possible candidate. To allow for both scenarios, we incorporate the memory factor μ which assumes values between zero and one: zero for no memory and one for full memory. It is instructive to use an example from Figure 2. Consider the number of people leaving node 1 for node 2:

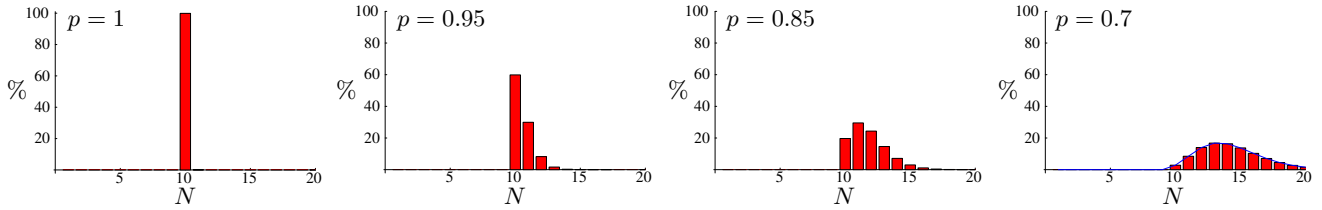


Figure 3: The Negative Binomial Distribution for $r = 10$ and different values of p . When $p = 1$ the avatar simply takes steps towards the exit of the node.

$$\Phi(t, 1, 2) = \frac{(1 - \mu)PR(2)}{PR(2) + PR(3) + PR(4)} [\Phi_{in}(1) * f(1)] + \mu \left(\frac{PR(2)}{PR(2) + PR(3)} [\Phi(4, 1) * f(1)] + \frac{PR(2)}{PR(2) + PR(4)} [\Phi(3, 1) * f(1)] \right). \quad (5)$$

This form of the equation can be shown to be consistent with Equation 2 by summing the total number of people leaving node 1.

5. THE SIMULATION

The first thing that needs to be done is to set the r and p for each node. In our case we used the length of the node for defining r , and the width for defining p . The reasoning for the latter is that if the road is narrow, it is probably just a passage and people will just walk through it, while if it is a wide avenue or a square then they are more likely to wander around. The preference factor for most of our experiments was defined based on the viewshed of the node, borrowing the idea from [20]. Of course the above assumptions were just a convenient way of setting the various parameters without user intervention. For more realistic results specific information can be supplied to the system from the designer - information such as where the main attractors are, what are the shopping streets etc.

On start-up of the pre-processing phase we allocate avatars in the nodes assuming that they have just entered the node. At each time-step, we estimate the number of avatars that leave each node using the convolution function (Equation 2). The infinite summation is truncated at the time-step which the cumulative distribution is 98%, i.e. 98% of the people that entered at past time-steps have left the node. Subsequently, the avatars are distributed to the neighboring nodes according to the preference factor of each node (Equation 5). The iterations are repeated until the number of avatars in each node fluctuates within 1%.

Once the high level information is computed then we add the actual avatars into the system (note that the high level is purely computed using the graph and the node parameters). We can either directly initialize the avatars according to the computed distributions or let them enter from the “dangling edges” of the graph. These edges are the only roads that lead people in and out from our model. Typically, most virtual city models represent only the city center, thus these “dangling” edges would be the avenues that lead the people in and out, hence people can be added through such roads (edges). In order to make it look more realistic the number of people added is proportional to the computed flux.

5.1 Adjusting the Flows and Densities

The user can control the simulation by either modifying the distribution curve at each node through the value of p , or by manipulating the PR factor by changing α and FUN . These can be set through an intuitive point-and-click interface.

Each parameter has a different effect on the dynamics of the simulation and the steady-state solution. Changing the p value of a node would primarily affect the total number of avatars in that node and as a secondary effect would cause a change in the influx/outflux of people in the node. This can be justified by considering the steady-state solution $N(i) = r/p \Phi_{in}$. Decreasing p , which implies that an avatar spends more time in a node, would result in an increase in the number of avatars in the node while the influx/outflux would not be altered significantly.

On the contrary, increasing the PR factor would result in an increase in the influx of avatars which would of course translate into an increase in the number of avatars in the node. An increase in PR accompanied with the right increase in p could result in increasing the flow but not the overall number in the node at any given moment.

Another scenario that it is interesting to model is the case of a significant event, i.e. a concert, an open market, etc., taking place in a street (node). Here, we would expect that the increase of the PR factor of the main street (node) would result in an increase in PR of the neighboring nodes. Hence, we allow a proportional increase of the PR factor to propagate radially outwards from the main node until we achieve the expected accumulation of people.

6. AVATAR NAVIGATION

The low level navigation takes place only within nodes. We did not make the provision to perform any path planning based on the assumption that the user would not be as much interested in following a single avatar around town but rather on the overall (collective) crowd behavior. When each agent enters into a new node then he is randomly assigned an exit direction, based on the ideas mentioned earlier. He is also assigned the time that he should spend within the node. The time depends on the high-level solution computed at preprocessing but also depends on the characteristics of each individual agent. A tourist, a shopper or a mother with young children is likely to spend more time rather than a worker.

In our current implementation the low level behavior is simplified; avatars perform collision avoidance and adjust their speed. The nodes are mostly convex and the portals between them are usually mutually visible, thus it is fairly easy for the agents to navigate. An agent is directed between portals by assigning its next destination to be a

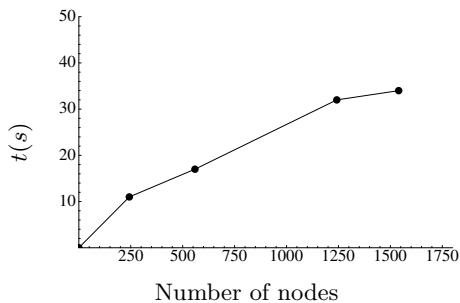


Figure 4: The method is fairly scalable in the number of nodes in the system, taking a maximum of 34 seconds to reach steady-state for the biggest scene we tried.

random point from the next portal. This eliminates the unwanted effect of all the agents walking in a straight line and on the same path when walking between portals. Collision avoidance also helps in producing more fuzzy paths. As a future work, the behavior would be enriched such that a certain avatar that spends a long period of time in the node would spend it by either sitting at a coffee shop or window shopping or talking to people etc.

7. RESULTS

We tested the system thoroughly to make sure that our assumptions lead to realistic results. We observed that indeed the simulation converged and the number of people in each node matched the expected $r/p \Phi_{in}$.

One of the issues we wanted to test is the sensitivity of the final result of the simulation with regards to the initial positions of the avatars. We initialized our simulation with different distributions at startup, such as spreading the avatars equally, or starting them all up in one corner. Our conclusion was that the resulting steady state does not depend on the initial distribution. If the node parameters and the connectivity are kept constant then the resulting steady state is always the same. It does make some difference though in the time required for convergence. The further the initial distribution is from the final result, the longer it takes to converge.

The method proved to be not only fast but also scalable. In Figure 4 we can see the required time for converging at steady-state. It ranges from 11-34 seconds with a more or less linear increase with respect to the number of nodes. The time needed for re-convergence, once changes are introduced to the system, depends on how large the change is and of how many nodes the system consists. Typically it is in the order of a few seconds, during which the simulation still runs on the background. The iterations for the high-level re-convergence are interleaved with the rendering so it does not stall.

At the moment the number of people placed in the system does not affect the balance between the nodes. An important reason for this is that we have not yet implemented congestion factors for the nodes.

We have run the system on a variety of city models and our results are qualitatively realistic. In Figures 1 and 5 we see images of a relatively small, hand-built model, while in Figure 8 we have a much bigger procedurally generated



Figure 5: An example scene simulated with our method.



Figure 6: A close up of the Nicosia network showing the partition into nodes.

model. The latter is based on the GIS data of the city of Nicosia and it is arguably more interesting, not only because it has a much larger network (more than 2000 nodes) but it also gives us an opportunity to test our method against the ground truth.

Starting from the GIS data, the nodes were automatically created. The roads were traced by following along their centerlines and checking the width. At the places where the road opens wider than a certain amount then we assume that we are either entering a junction or a square and thus mark the end of a node by placing a portal. Although it is a simple method the result is satisfactory, as can be seen in the close-up of Figure 6. The preference factor, the r and the p for each node were again computed automatically as described in Section 5. It is encouraging to see that our results are close to reality as the method has identified most main streets of the city giving them a higher flow of avatars. These are shown in Figure 8 with light blue. In the same figure on the right we can see the effect of adding attractors to a street. After obtaining the initial solution, the user decided to modify it by placing shops on one of the roads in the Old Town. That had the effect of increasing the flow of

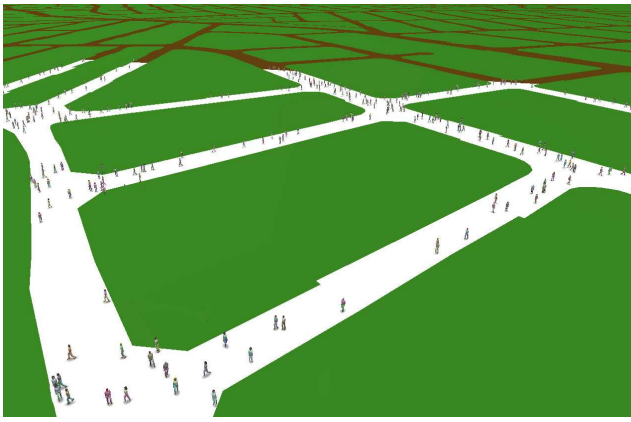


Figure 7: The low level navigation can be restricted to only the visible nodes (in white), while the rest (in dark brown) is maintained by the high level system.

avatars in that specific road and in its immediate neighbors. The change in the rest of the network is less profound.

We populated the Nicosia model with 7,000 avatars but the resulting frame rate was far from real-time, down to 1-2 frames per second. The main cause for the slowdown was the low level behavior and in particular the collision avoidance. The cost of the high level instructions is almost negligible at run-time since the costly part is performed at pre-processing. However, in most cases, especially when the viewer is standing at ground level, only a small fraction of the pedestrians in a city can be seen. With our system it is easy to switch off both the low level simulation and the rendering for the non-visible nodes and maintain consistency using the high level (see Figure 7). By doing that, only the visible areas are populated saving valuable resources that can be used to improve the detailed behavior and rendering of the relevant regions and hence achieving a high frame rate.

8. CONCLUSION AND FUTURE WORK

In this paper we presented a system for specifying the high level behavior of avatars in a large urban setting. The main contributions of this system are (a) the ease of use, as it can derive the initial densities and flows of the avatars automatically based only on the geometry configuration of the model and (b) the precise user-control allowing the user to easily modify the flows and densities in the desired regions by adjusting some intuitive parameters.

Our evaluation has shown that the system is relatively fast to converge and respond. It does not require the simulation of the avatars over the entire network to correctly maintain the densities. Avatars need only be generated within the visible part of the city and are easily re-generated from the simulation densities when a node becomes visible again.

The flexibility of the model allows for the variation of the number of people in the city or the simulation of a popular or unpopular event, even the alteration of the urban structure. These can all be accomplished at interactive rates.

Areas where there is room for improvement include the implementation of congestion and the analytic estimation of the steady-state. As mention in the results, we have not implemented the congestion factor into the system. The

consequences of congestion would necessitate the dynamic modification of both the probability density function (pdf) and the preference factor for each node. The pdf has to be modified because pedestrians would spend more time in a congested node and the preference factor has to be modified to account for the fact that pedestrians are less likely to visit a congested node.

Acknowledgements

This work was partially supported by the CREATE project (EU FP5 IST-2001-34231) and the Frederick Research Center. We wish to thank Franco Tecchia and Celine Loscos of University College London for providing us with the crowd rendering module. We also wish to thank Timotheos Motwary for the helpful discussions.

9. REFERENCES

- [1] R. Ayelett and M. Cavazza. Intelligent virtual environments. State-of-the-art report at Eurographics 2001, 2001.
- [2] 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.
- [3] S. Chandrasekhar. *Noise and Stochastic Processes*. Dover, 1943.
- [4] F. Durand, G. Drettakis, J. Thollot, and C. Puech. Conservative visibility preprocessing using extended projections. *Proceedings of SIGGRAPH 2000*, pages 239–248, July 2000.
- [5] M. Fukui and Y. Ishibashi. Traffic flow in 1d cellular automaton model including cars moving with high speed. *J. Phys. Soc. Jpn*, 65(6):1868–1870, June 1996.
- [6] M. Fukui, H. Oikawa, and Y. Ishibashi. Flow of cars crossing with unequal velocities in a two-dimensional cellular automaton model. *J. Phys. Soc. Jpn*, 65(8):2514–2517, August 1996.
- [7] 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.
- [8] J. J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, Boston, MA, 1979.
- [9] D. Helbing, H. J. Herrmann, M. Schreckenberg, and D. E. Wolf, editors. *Traffic and Granular Flow '99*. Springer, Berlin, 2000.
- [10] D. Helbing and P. Molnár. Self-organization phenomena in pedestrian crowds. In F. Schweitzer and H. Haken, editors, *Self Organization of Complex Structures: From Individual to Collective Dynamics*, pages 569–577. Gordon & Breach, Amsterdam, 1997.
- [11] 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.
- [12] 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.

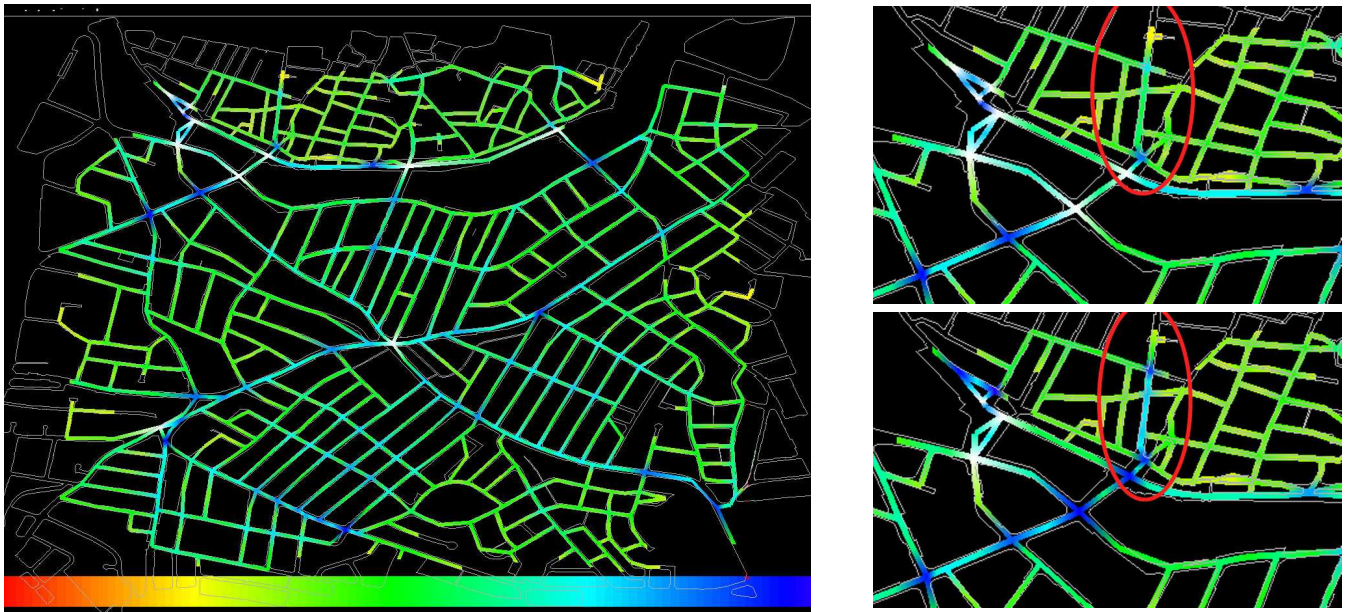


Figure 8: The figure on the left shows the flow of people calculated by our method for the city of Nicosia with all parameters set automatically (the magnitude of the flow is indicated by the legend; blue indicates larger flow). In the two close-ups on the right we compare the flows computed by setting the parameters automatically, shown on the upper figure, and the flows when the user adds some attractors on a specific road (indicated within a red circle), shown on the lower figure.

- [13] 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*, 28:327–341, 2001.
- [14] M. Lighthill and G. Whitham. On kinematic waves: I. flood movement in long rivers; ii. theory of traffic flow on long crowded roads. *Proc. R. Soc. London, A*, 229:281, 1955.
- [15] D. Luebke. A developers survey of polygonal simplification algorithms. *IEEE Computer Graphics and Applications*, 21(3):24–35, May 2001.
- [16] W. Mendenhall and T. Sincich. *Statistics for engineering and the sciences*. Prentice Hall, 1995.
- [17] M. Muramatsu, T. Irie, and T. Nagatani. Jamming transition in pedestrian counter flow. *Physica A*, 267:487–498, 1999.
- [18] 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.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] Y. Tajima, K. Takimoto, and T. Nagatani. Scaling of pedestrian channel flow with a bottleneck. *Physica A*, 294:257–268, 2001.
- [23] F. Tecchia, C. Loscos, and Y. Chrysanthou. Image-based crowd rendering. *IEEE Computer Graphics and Applications*, 22(2):36–43, March–April 2002.
- [24] D. Terzopoulos. Visual modelling for computer animation: Graphics with a vision. *Computer Graphics*, 33:42–45, 1999.
- [25] G. Thomas and S. Donikian. Modelling virtual cities dedicated to behavioural animation. *Computer Graphics Forum*, 19(3):C71–C80, 2000.
- [26] X. Tu and D. Terzopoulos. Artificial fishes: Physics, locomotion, perception, behavior. In *SIGGRAPH '94*, pages 43–50, July 1994.
- [27] 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.
- [28] M. Wand and W. Straßer. Multi-resolution rendering of complex animated scenes. *Computer Graphics Forum*, 21(3), 2002. Eurographics 2002.
- [29] B. Wang, Y. Woo, and P. Hui. Mean field theory of traffic flow problems with overpasses and asymmetric distribution of cars. *J. Phys. Soc. of Jpn*, 65(8):2345–2348, August 1996.
- [30] J. Zacharias. Pedestrian behavior and perception in urban walking environments. *Journal of Planning Literature*, 16(1):3–18, 2001.