Congestion-Aware Optimization of Pedestrian Paths

Srihari Narasimhan
narasisi@ipvs.uni-stuttgart.de
IPVS, Universität Stuttgart
Universitätsstr. 38, D-70569 Stuttgart, Germany

Hans-Joachim Bungartz
bungartz@in.tum.de
Institut für Informatik, Technische Universität München
Boltzmannstr. 3, D-85754 Garching bei München, Germany

Abstract

Today, more and more simulation tasks with a traditionally non-geometric background need to be embedded into some geometric context in order to provide spatial context to non-spatial data. This holds especially for graph-based applications in some location aware context. As an example, one might think of a large theme park or a large commercial center where the customers shall be provided with some navigation and scheduling information such as where to go and when – either a priori or even in real time via some mobile device. In this paper, we present an approach to embed an event-driven simulation tool for pedestrian traffic into a 3D geometric environment in order find an optimal path through the graph-based model by considering the congestions caused due to the pedestrian movements.

1 Introduction

In a large commercial center, the shortest path between two points is not necessarily the fastest path. This is due to the fact that the shortest path does not take into consideration the congestions that are created by the pedestrians in the buildings. In such a scenario, pedestrian traffic can be expected all over the building and especially more in the main pathways and around the service counters. Now, when a customer wishes to go to some location, he would either look in the map to find the easiest or the shortest path or he would look for direction boards, which point to his destination. However, the customer would then encounter locations where there are large pedestrian congestions. To avoid such time-consuming situations, the customer would need an effective navigation system, which would not only guide him to his destination but also find the fastest path by identifying the congestions in the region and by simulating the time required to transit the congested paths. This paper proposes an approach to embed the queuing system into the geometry-based model and to find the fastest path by simulating the pedestrian traffic.

After a short introduction on simulation techniques of pedestrian paths in Section 2, this paper describes embedding of queuing systems into the geometric model in Section 3. Section 4 discusses a sample scenario in an architectural model. The whole system is then simulated and a comparison of path lengths and the time taken to cross the paths is made in Section 5. This is then followed by the conclusions where the future possible directions of research are presented.

2 Discrete-Event-Based Pedestrian Simulation

The pattern of a pedestrian congestion is similar to a 'Queuing System' where the pedestrians wait in the congestion to transit a specific path. One standard technique to model a queuing system or a network of queuing systems is the so-called Discrete Event Simulation [5, 6], where pedestrians are entities in the queue and the movements of these entities into and out of the queue are recorded as events. For this purpose, each path where possible pedestrian movement exists is identified from the given model. Each individual path is modeled as a queuing system wherein the transition time of the pedestrian is simulated. The model consists of a queue and several service units. The number of service units of the queuing system is decided based on the number of pedestrians the specific path can accommodate or in other words, the capacity of the path. The pedestrian arriving in this path first joins the queue. If there is no service unit available to serve the pedestrian, the pedestrian waits in the queue until there is a free service unit. This means that there is congestion in the path and the pedestrian needs extra time to transit the path. Once there is a service unit available, the pedestrian leaves the queue and enters the service unit. This means that there is enough space to accommodate the pedestrian in the path. The service time at the service counter is same as the time needed to cross the path. The service time is calculated generating the approximate walking speed and dividing it by the distance of the path. The walking speed is assumed to have a mean value μ of 1.38 m/s with a standard deviation σ of 0.37 m/s [3].

Since there are several paths in a building and since each path is modeled as a queuing system, the paths are interconnected to form a queuing network. So as the customer chooses his path through the building, he transits all the paths thereby waiting in all the queues along his path. So the arrival time of each queue he enters is same as the departure time of his previous queue. This method precisely estimates the time required for a pedestrian to reach his destination. In case of a congestion, it is also possible to estimate the total time required to cross a path or in other words, the sojourn time (service time + waiting time) of each queue at frequent time intervals. By calculating the difference in the sojourn time at time t and the actual time needed to cross the path, the additional time that is needed to cross the path is determined. So it can be decided if the path is fast enough to reach the destination or if a rerouting is necessary.

3 Geometry Embedding of Queuing Systems

In order to identify the paths in an architectural model, the possible paths where the pedestrian movement exists are extracted automatically from the CAD model of the building using the Pathscan program as in [1] and stored as a graph. Pathscan is a flexible tool that reads a CAD model of an architectural building, scans the model along the z-buffer and identifies the paths where the pedestrian can move and creates a graph of the paths. The flexibility of Pathscan allows the user to add new paths, modify or even remove the existing paths based on the users requirements. Once the graph is finalized in the Pathscan, the graph can be exported to a universal XML file that can be read by other applications.

The graph resulting from the Pathscan tool contains a list of nodes and their coordinates in the geometry model and the edges connecting the nodes. With this data, a list of paths along with other attributes such as exact dimensions of the path, special paths such as stairs or ramps, etc., is generated to embed the queuing system into the graph. As described in the previous section, each path is modeled as a queuing system. On simulating the model, the precise time needed to transit the various paths is calculated. The resulting system gives an overview about the congestions that occur across the geometric model. The following sections demonstrate how a geometry model is converted to a graph of pedestrian paths and how pedestrian move across the building and cause congestions.

4 Model and Scenario

In order to demonstrate the simulation and optimization of pedestrian paths, we first need an architectural model. For this purpose, we take the CAD model [4] of the new computer science building of Universität Stuttgart finished in 2003 as the reference model. The CAD model is then viewed with the Pathscan tool to generate the pedestrian paths along the building. Other path attributes such as the type (stairs, walkways, ramps, etc.) are automatically detected. Since it is desired that the pedestrian use only specific paths and not all possible paths (such as emergency exits, private rooms, service areas, etc.) the paths are refined by altering the graph generated by Pathscan by adding attributes to the paths. Figure 1 illustrates the parsing of the model and creation of the graph.

As shown in the figure, the final graph that is needed for the pedestrian simulation is extracted. Now, the pedestrian or customer needs to visit a specific destination. So it is necessary to define attributes to the nodes in order to differentiate between rooms, pathways, doors, etc. This is illustrated in Figure 1.d. Now that we have the graph, we try to simulate the movement of pedestrians. For this purpose, we need statistical data on how many people would visit the building, the destinations they would visit in the building, the time they would need at each destination and so on. Since the objective is to demonstrate the embedding of queuing systems into a geometric context to optimize the pedestrian paths, we build a flexible model with the required statistics as parameters to the model. We then imagine a hypothetical but quite a realistic scenario for generating the values for the parameters.

Figure 1.a: VRML model of the computer science building

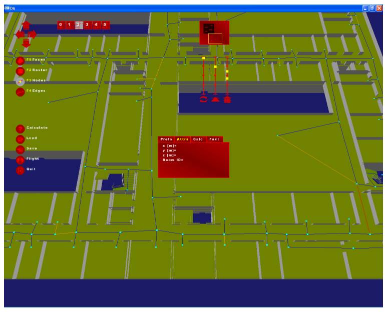

Figure 1.b: Screen shot of the Pathscan program showing the graph of pedestrian paths

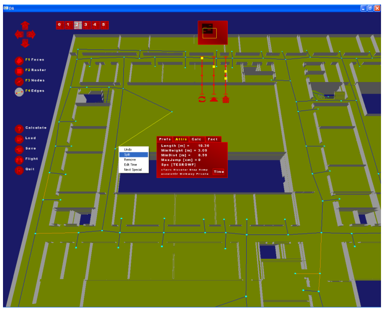

Figure 1.c: Pathscan tool used to alter the edges of the graph

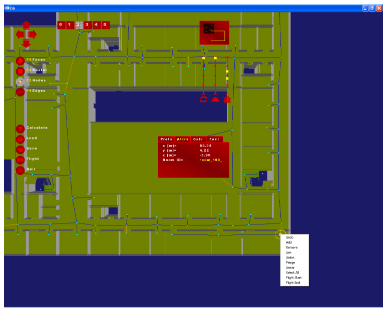

Figure 1.d: Pathscan program to specify the room numbers of the destinations. The same method also used to define node attributes such as door, pathway, exits, etc.

Hypothetical Scenario

We already know the list of destinations where a customer would possibly go to execute his tasks. We assume that each customer decides to visit n destinations at a random sequence. We also define different probability distributions to each destination so that the destinations chosen by each customer is well distributed. Initially, the customers arrive at exponentially distributed time. The subsequent arrival times are the departure times of the previous queue. In case of joining a queue after executing the task at the destination, the arrival time will be the sum of the departure time and the sojourn time at the destination. The service times at the queues are normally distributed resulting in an M|G|n queuing model. The path from one destination to an other is the shortest path in terms of distance which is determined by a Dijkstra's shortest path function for weighted graphs.

5 Simulation

Once the sample scenario is decided, a queuing system for each path is modeled and all queuing systems are interconnected to form a queuing network. The model is simulated for a time period of 8 hours with 1000 customers visiting the building. The model always chooses the shortest path for a customer irrespective of the delays caused due to the pedestrian congestion. On the other hand the simulation also calculates the additional time needed to cross the path in case of congestion for each path. So it is now possible to find the fastest path instead of the shortest path by replacing the weights of the graph with the extra time needed instead of the distances. Figure 2.a visualizes the congestion caused at each path.

Let us now make a comparison of the fastest path and the shortest path for the above-mentioned scenario. Let us assume that at some time t during the day, a customer is on his way to his destination. We choose the source and destination such that the shortest path passes through the main pathway where the congestion probability is higher. There are two staircases, one at the north end and one at the south end. The source is located between the staircases, close to the south side staircase. The destination is located one floor below in the north side of the building. The shortest path obviously is the path towards the north side staircase which is about 82 meters. On simulating the scenario, we find that the time it takes to reach the destination is about 128 seconds. On the other hand, the fastest path takes just about 112 seconds and the distance it covers is 100.2 meters. Figure 2.b shows the source and destination and differentiates the shortest and fastest path.

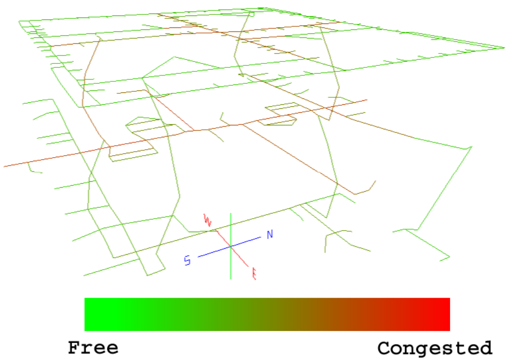

Figure 2.a: The graph of pedestrian paths in the architectural model showing the congestion at each path

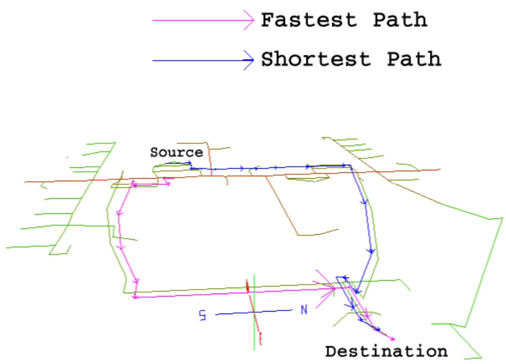

Figure 2.b: Shortest and fastest path from the source to the destination

From the obtained values, one can note that there is a significant difference in the shortest and fastest path. Considering the fact that the customer might not be interested in saving a few seconds or a few meters to reach is destination, the results for one particular customer and one particular path might not be significant. But the key lies in navigating many customers to many destinations and efficiently scheduling the visits to each destination for which this approach of embedding queuing system into a geometry context is necessary.

6 Outlook

It is interesting to note that the embedding of the queuing systems into the geometry structure results in a completely different approach of pedestrian navigation. This paper bridges the gap between traditional pedestrian simulation methods and the standard path search algorithms and thereby provides a method for the development of an effective navigation system that is suited for pedestrians. A similar approach to detect bottlenecks in the architectural model by integrating evacuation planning is proposed in [2]. Another interesting aspect to work on is the scalability of this approach by connecting the CAD models of several buildings and the pedestrian simulations in each building. In a theme park for example, when a visitor wishes to visit as many attractions as possible, this approach can help to efficiently schedule the visits based on waiting time in the queues as well as navigate between the different attractions. Apart from this, several other parameters and functionalities such as planning a visit to several locations, scheduling the visits in an optimal sequence, predicting and simulating the occurrence of congestions, etc., are still an open avenue of research.

References

[1] *Thomas Drexl*, Entwicklung intelligenter Pfadsuchsysteme für Architekturmodelle am Beispiel eines Kiosksystems (Info-Point) für die FMI in Garching, Diploma thesis, Technische Universität München, 2003, München, Germany.

[2] *Ralf-Peter Mundani, Hans-Joachim Bungartz, and Stefan Giesecke*, Integrating Evacuation Planning into an Octree-Based CSCW Framework for Structural Engineering, In Proc. of the 22nd CIB-W78 Conference on Information Technology in Construction, 2005, Dresden, Germany.

[3] *Kardi Teknomo*, Microscopic Pedestrian Flow Characteristics: Development of an Image Processing Data Collection and Simulation Model, Ph.D. Dissertation, Tohoku University, March 2002.

[4] *Stefan Giesecke, Matthias Stier and Sven Grumbein*, Pfadsuche in Architekturmodellen und Stereoprojektion, Softwarepraktikum, Universität Stuttgart, 2004.

[5] *Jerry Banks, John S. Carson II, Barry L. Nelson and David M. Nicol*, Discrete-Event System Simulation, Prentice Hall Intl. Series in Industrial and Systems Engineering, 2000.

[6] *Kevin Watkins*, Discrete Event Simulation in C, McGraw Hill Intl. Series in Software Engineering, 1993.