Modeling and Simulation of Pedestrian Behaviors in Crowded Places

WEE LIT KOH SUIPING ZHOU
School of Computer Engineering
Nanyang Technological University
{KOHW0005, ASSPZhou}@ntu.edu.sg

Pedestrian simulation has many applications in computer games, military simulations and animation systems. A realistic pedestrian simulation requires a realistic pedestrian behavioral model that takes into account the various behavioral aspects of a real pedestrian. In this paper, we describe our work on such a model that aims to generate human-like pedestrian behaviors. To this end, various important factors in a real-pedestrian’s decision-making process are considered in our model. These factors include a pedestrian’s sensory, attention, memory and navigational behaviors. In particular, a two-level navigation model has been proposed to generate realistic navigational behavior. As a result, our pedestrian model is able to generate various realistic behaviors such as overtaking, waiting, side stepping and lane-forming in a crowded area. The simulated pedestrians are also able to navigate through complex environment given an abstract map of the environment.

Categories and Subject Descriptors: I.6.5 [Simulation and Modeling]: Model Development—modeling methodologies; I.6.8 [Simulation and Modeling]: Types of Simulation—animation

General Terms: Design, Performance
Additional Key Words and Phrases: Pedestrian simulation, behavioral modeling, autonomous agents

1. INTRODUCTION

It is such a common sight to see pedestrians strolling along a street or squeezing through a crowded area that it is sometimes easy to forget about the complexity involved in synthesizing realistic pedestrian behaviors. In real-life, pedestrians may influence how other pedestrians will react with their own walking behaviors. They need to adopt a variety of maneuvers such as following, overtaking and evading to be able to reach their destinations. At the same time, different pedestrians have different personalities which may also influence their navigation choices. Pedestrians also have their own preferences such as comfortable walking speed, and their own habits, such as the continuous use of the same route to reach the same destination.

Equally important are that pedestrians may have their own subjective judgement, e.g., what is considered as fast by one pedestrian may be considered as slow by another. Pedestrians also behave differently in different social context and different social groups. As a result, people may have different expectations on how other people around them will move in response to their own movements. Similarly, people in different cultures may have different social norms that will influence the way one moves around the environment. For example, people in some cultures may prefer walking on the left (or right) side of a lane. For a realistic pedestrian simulation, it is therefore important to notice the differences among individual pedestrians.
and understand how a person’s navigation behavior is affected by various factors.

Pedestrian simulations have many applications in computer games, military simulations and animation systems. For example, automated generation of animated pedestrians in a virtual scene can be used to enhance the realism of the virtual environment [Ulicny and Thalmann 2002]. Pedestrian simulations can also be used to study the design, safety and egress of buildings such as airports and office floors [Still 2000]. In these applications, the realism of the pedestrian’s navigation behavior will affect the accuracy and reliability of the simulation results, e.g., the total time required to evacuate from a building.

In pedestrian simulations, the pedestrians are usually represented by autonomous agents whose movements are driven by a navigation model. Thus, one of the key challenges to the modeling and simulation of realistic pedestrian behavior is on the development of a navigation model that mimics the decision making process of a pedestrian. In particular, factors that affect a pedestrian’s navigation choices may come from different domains such as the pedestrian’s age and patience. Hence it is desirable to develop a behavioral model to incorporate the influences of all these factors on pedestrian navigation behavior. However, such a model may be too computationally costly to be applied in any practical applications. Thus, our approach to this challenge is to develop an extensible navigation model that can be customized to demonstrate different navigation behaviors on demand.

In this paper, we proposed a two-level navigation model to describe the movements of a pedestrian. First, the macro-level navigation model is used to compute a path (not necessary shortest) to a destination based on various influences that may affect route choice such as the agent’s knowledge about the environment. Second, the micro-level navigation model is used to compute new steering parameters for the agents. Important pedestrian navigation behaviors like collision avoidance and overtaking are generated by the micro-level navigation model.

In Section 2, we will describe related work and give an overview about our pedestrian representation in Section 3. Then, in Section 4, we will describe the pedestrian navigation model for generating realistic navigation behaviors. We will describe how various physical and psychological factors that affect the navigation behaviors of a pedestrian are considered in our proposed model in Section 5. We will present some navigation behaviors demonstrated by our pedestrian model in Section 6 and conclude this paper in Section 7.

2. RELATED WORK

Pedestrian navigation in a virtual environment can be achieved using social forces [Helbing et al. 2005], [Helbing and Molnár 1995], [Teknomo 2006], flow based approaches [Hoogendoorn and Bovy 2001], [Hughes 2002], cellular automata [Blue and Adler 2001], [Burstedde et al. 2001], [Kirchner 2002], and agent based approaches [Feurtey 2000], [Musse and Thalmann 2001], [Reynolds 1999]. However, few existing work covers both the steering and path-finding considerations of a pedestrian. Instead, existing pedestrian navigation models usually have a more thorough solution to macro-level navigation challenges due to the abundance of related work on path-planning. For example, Shao and Terzopoulos [2005] used a Quadtree approach to recursively divide the environment into smaller regions and represents
each region as a node in a graph. The graph is then used to search for a shortest route to the agent’s destination. While the macro-level approach is similar to ours, Shao used a rule based approach to model the reactive behaviors of a pedestrian. However, it is difficult to describe all the alternatives that are available to a pedestrian using these rules. For this rule-based approach, the realism of the generated behaviors depends greatly upon the skills of the modelers to specify these rules.

Lamarche and Donikian [2004] proposed a two-level approach to achieving realistic pedestrian movements. Similar to our method, Lamarche recursively divides the environment into smaller blocks to represent the areas that are accessible to an agent. Collision avoidance is achieved by predicting the time to collision and the type of collision, e.g., rear or front collision. Based on the expected type of collision, several collision avoidance strategies are proposed and evaluated. Then the best strategy is used for the agent to determine its new speed and orientation. One problem with this approach is that the model focuses on avoiding physical contacts between different agents and does not give sufficient consideration to the different factors that may influence a pedestrian’s navigation behavior, e.g., a pedestrian may even choose not to avoid the collision with others in certain situations.

For agent based approaches, it has become increasingly common to include psychological factors in the representation of individual pedestrians [Braun et al. 2003], [Pelechano et al. 2007]. This trend of modeling increasingly realistic pedestrian behavior is expected to continue mainly due to the continuous increase in the computational power of modern computers. For example, Sakuma et al. [2005] proposed an agent model that considers the memory and vision capabilities of a pedestrian. Pedestrian navigation is then achieved using potential fields, as first proposed by Reynolds [1999]. For the decision to avoid collision, Sakuma proposed the use of a two stage personal space. When agents are found in the inner personal space, these agents should be avoided immediately. On the other hand, agents found in the outer personal space represent agents that should be avoided without making sudden changes in velocity. The inclusion of individual memory and vision will help enhance the diversity of the generated behavior among different agents. This is an important property which we will also address in this paper.

In representing a pedestrian using an autonomous agent, it is important for each agent to have their own sensory inputs and representation of the environment. This will allow agents to react according to the situation in their local area. Pan et al. [2005] proposed one such framework to model pedestrian behaviors in an egress simulation. In the proposed framework, Pan describes an agent to have age, gender, sensory inputs and navigation decision modules. In particular, the movement of a pedestrian is described using steering behaviors such as random walk, collision avoidance and target following [Reynolds 1999]. Higher level navigation behaviors are achieved by combining several basic steering behaviors. However, Pan’s model is focused on the egress scenario where the primary goal of an agent is to reach an exit or safe zone as soon as possible. Thus, the resultant behaviors may not be generally suitable for other situations. For example, the herding effect, where pedestrians will attempt to move towards a general direction where most of the other pedestrians are moving, is hardly seen for commuters in a underground passage way.

Another important distinction between Pan’s framework and our proposed model
is that our navigation model does not rely on a pre-determined sequence of steering rules to execute a more complex navigation behavior. Instead, our proposed model changes the steering parameters of the agents according to some pre-defined steering constraints. From a modeler’s point of view, these constraints are quite natural and easy to specify. Complex navigation behaviors such as lane forming or overtaking in a crowded area naturally emerge as a result of these constraints. That is, in our model, we never specify the rules for these behaviors. Thus, there is also no need to specify when to switch between different navigation behaviors. In addition, our model is more flexible in that it allows different navigation behaviors to be easily combined. For example, an agent can be avoiding one agent while overtaking another agent seamlessly.

Treuille et al. [2006] proposed a potential field approach to model the personal preferences of individual agents for path planning. This preference is modeled as a discomfort field that pushes the agents away from a path that they are uncomfortable or unfamiliar with. The computational cost of Treuille’s model is relatively low, thus it can support tens of thousands of agents on a single workstation with a reasonable frame rate. However, it is mainly suitable for modeling a large homogeneous pedestrian gathering.

In contrast to Treuille’s approach, our approach aims to generate a large heterogeneous pedestrian gathering where the movement of individual agents depends upon individual differences such as their habit and personality. Therefore, our main contribution towards the modeling and simulation of pedestrian behaviors can best be regarded as a bridge between physio-sociological studies of real pedestrian behaviors such as the works of Baddeley [1997], Gwynne et al. [1999], Posner and Petersen [1990], and Rao and Georgoff [1999], and the models for generating automated agent movement and animation such as works of Thalmann et al. [1999], Schadschneider et al. [2002] and Burstedde et al. [2001].

3. PEDESTRIAN REPRESENTATION

In order for the agents to demonstrate realistic navigation behaviors, an accurate representation of a pedestrian is needed. The proposed pedestrian representation aims to mimic how a real pedestrian collects and uses information about the environment for navigation. To this end, our agent is equipped with individual and limited sensory and memory capabilities. The modeling of an agent’s perception about the environment will allow for individual route-plans based on the agent’s own knowledge about the environment, e.g., agents may not always know the shortest path to the destination. Limited sensory capabilities allow for more realistic behaviors, e.g., an agent will not attempt to avoid another agent who is outside its sensory range. In this paper, some parameters of an agent such as its sensory range are initialized randomly within a range. However, it is possible to base the initialization on some empirical studies about human behaviors.

Figure 1 summarizes the agent representation of a pedestrian based on our observations on real pedestrians. During navigation, real pedestrians need to sense the changes in the environment to plan their paths and steer through the environment to avoid both structural obstacles and other moving objects. Therefore, we have introduced the sight, sound and touch modules to model a person’s sensory system.
The agent’s sensory systems are implemented as external sensors that are attached to the agent. These sensors will allow the agent to be aware of its surroundings and other agents.

Figure 2 shows the proposed sensory model for the agent. The agent’s sight has controlling parameters, $\text{FOV}$, $r_{\text{vision}}$ and $n$. $\text{FOV}$ and $r_{\text{vision}}$ represent how wide and how far the agent can see, respectively. $n$ divides the vision cone into $n$ sections, and hence $n + 1$ discrete rays are needed to sample the objects in the environment. When moving, an agent will emit a sound which may be picked up by other agents. An agent’s hearing capability is controlled by $r_{\text{sound}}$ which determines how far away the agent is able to hear. In particular, the sound sensor is useful for detecting agents approaching from behind or around a corner which will otherwise be invisible to the agent. The agent’s touch sense has been modeled as a bounding box around the agent’s 3D model. The sensory module of the agent is used to capture the distance, speed and orientation of neighboring objects in the environment. This information is then passed to the agent for the purpose of navigation.

In addition to the senses, the agent also has an invisible personal space to represent the space which the agent considers as private. The modeling of a personal space is based on the observation that pedestrians will feel uncomfortable to touch
shoulder to shoulder when navigating through the environment. Instead, pedestrians will try to keep a distance between themselves and those around them. We will further elaborate the effect of personal space on the simulation in Section 6.3.

Information sensed by a pedestrian includes the topology of the environment and positional data of other pedestrians. However, not all the information gathered by a pedestrian is expected to be accurate. In fact, this information is affected by factors like personality, personal experiences and subjective judgements. The sensory estimation error component is hence used to introduce such human sensing error into agent’s sensory inputs. In this paper, the sensory error is simply modeled as a random value in a pre-defined range which is chosen to reflect the different tendencies of the individuals in sensing spatial information. For example, a negative error range is used to represent a person who tends to under-estimate the distance between two objects.

The agent’s attention component is introduced to model a pedestrian’s attention system. Pedestrians are not expected to notice everything that they sense in the environment, i.e., an object in the environment may be seen but not be consciously processed by a pedestrian if he/she is occupied by other objects. Instead, the human sensory system will actively select objects of interest to focus on [Posner and Petersen 1990]. For example, a pedestrian who does not pay attention to the surroundings may collide with objects in the environment. In our simulation, we model attention by giving each agent a specific number of attention points to start with. Each time an agent needs to track the movement of a new target, a specific number of attention points are subtracted from this pool. When the number of attention points reaches zero, the agent can no longer track objects that enter its detection range. Objects of interest are selected for conscious processing based on their relative positions to the sensing pedestrian. That is, nearer objects will have a higher chance of being consciously processed than an object further away. Attention points can be reclaimed if a tracked object is ignored by the agent.

Information sensed by the pedestrians is stored in the sensory and visual memory part of the human memory [Baddeley 1997]. The sensory memory acts as a temporary buffer between what is sensed by a person’s sensory systems and what is perceived by the person. Only the information that is less likely to change and necessary for the planning of a person’s path such as the position of different land-
marks, are transferred and stored in the visual memory. Our pedestrian model stores dynamic information such as the position of neighboring agents in the sensory memory and stores static information such as structural obstacles in the visual memory. Information from the sensory memory is fed to the visual memory to aid the agent in identifying the current stage of its journey.

An abstract map of the environment is maintained by an agent and stored as part of the visual memory in our model. This abstract map is used by a pedestrian’s path planner, and when combined with the pedestrian’s goal, produces a rough path for the agent to follow. The rough path produced by the path-planner is represented as a list of waypoints in our macro-level navigation module.

Based on the information gathered about other pedestrians in the environment and a rough plan to reach the destination, an agent will then determine the speed and orientation needed to navigate through the environment. In our model, the task of computing the new steering parameters for the agent is done by the steering computer. Its output will then be used to drive the agent’s animation and move the agent through the environment. Both the steering computer and the animation controller form our micro-level navigation module. Together, the micro and macro-level navigational modules constitute the core modules for generating realistic navigation behaviors for our autonomous agents.

4. AGENT NAVIGATION MODEL

Our agent navigation model consists of a macro-level model and a micro-level model. The macro-level navigation model is used to compute a path for an agent to reach a destination based on various influences that may affect the agent’s route choice. For example, the agent’s limited knowledge about the environment or personal habit may determine the route that is used to reach a destination. The micro-level navigation model is used to compute new steering parameters for the agent to move along the path as determined by the macro-level navigation model. An important consideration of the micro-level model is how to steer an agent in a
realistic manner in complex environments with many moving agents. Figure 4 shows how the abstract map of an agent is derived and how the macro and micro navigation models are coupled.

Note that the rough path generated by the macro-level navigation model is in the form of a sequence of way points from an agent’s current location to its destination. It does not specify how an agent should move from one way point to the next way point along the rough path. The exact steering behavior of an agent between two way points along the rough path is dynamically determined by the micro-level navigation model according to the movements of other agents in the environment and the agent’s own preferences. The relation between the macro-level navigation model and the micro-level navigation model in fact reflects how real pedestrian make navigational decisions. As an example, consider how a person describes the way to office from home. It is unlikely that the person knows exactly how many meters to move and along which exact directions. Instead, the person is more likely to describe the way in terms of a number of key points and their relations, e.g., “first, I will go to the tall building at Street $S_1$, then from there, I will go to the subway station at Street $S_2$, ..., finally I reach my office at X.” In each stage along this rough path, the person needs to consider the movement of other objects and other factors to determine how to move step by step in terms of moving direction and speed.

As shown in Figure 3, when agents are moving on a simple and open area, e.g., public squares 3(a) and corridors 3(d), there is no need to use a path planner. In this case, agents will be able to reach their destinations using only the micro-level navigation model. However, in more complex environments such as Figure 3(b) and 3(c), agents may need to make twists and turns in order to reach the final destination. In such cases, the macro-level navigation model is needed to help the agents to navigate through the environment.

In this section, we will outline our macro-level model (Section 4.1) and micro-level model (Section 4.2).

4.1 Macro-Level Navigation Model

The macro-level navigation model is used by our agent to plan a rough path to reach its destination in a complex environment. In particular, the environment is
represented as an abstract map that outlines the accessible and inaccessible regions. The completeness of this abstract map depends on several factors, e.g., whether a pedestrian is a new or frequent visitor of this space. With a different abstract map, the underlying path-finding algorithm will generate a different rough path and therefore results in different route choices for different agents. Depending on the scenario being represented, different pedestrian behaviors can be modeled when the abstract map is absent or incomplete. For example, an agent may try to seek the nearest map to find its orientation and complete its abstract map or the agent may choose explore the environment.

The model is also responsible for determining the goal point of the agent in a virtual environment. For example, if an agent wishes to visit a group of friends having lunch before going back to office, the agent will insert sub-goals in the macro-navigation model at the meeting point and entrance to the office building. For behavioral diversity, several meeting points can be defined in the virtual environment giving the agent a choice to proceed directly to the area’s exit or to visit one of the meeting points first.

Several existing methods can be used to represent regions of the environment that are accessible by the agent, for example, corner graphs, quadtrees and waypoints [Millington 2006]. We have used a hybrid quadtree based approach to build an accessibility graph of the environment. Unlike a normal quadtree model, we have added waypoints to the center and edges of the region bounded by a quadtree leaf. These waypoints are then connected together with their neighbors to represent the plausible routes between all reachable parts of the environment. Using only the region centers as waypoints for navigation allows only four moving directions: forward, reverse, left and right. However using our approach, the agent’s path will have an added diagonal degree of freedom, which increases the smoothness of the generated path, especially when navigating around sharp corners. In this paper, we have used the A* algorithm as the primary graph search algorithm [Hart et al. 1968] although other path-finding algorithms can also be used.

Figure 5 shows the path generated using different waypoint connection approaches. In particular, Figure 5(a) shows the path generated using our Hybrid Quadtree approach. As compared to a standard implementation of a Quadtree (Figure 5(b)), our approach results in less orientation changes when navigating around irregular corners, which is more realistic. Similarly, as compared to Figure 5(c), our approach can help to minimize the cutting of the agent’s 3D model with the walls of the environment when navigating around sharper corners.

The heuristics used in our implementation of A* includes the pedestrian density ($\rho$) and the diagonal distance ($d_p$) from the agent’s current position to the destination. Pedestrian density is used as a heuristic to encourage agents to avoid congestions and seek alternative routes. Diagonal distance is used because it does not have a square root component and is hence more computationally efficient than the manhattan distance heuristics. The diagonal distance ($d_p$) from an agent’s current position ($x, y$) to a destination point ($p_x, p_y$), is given by

$$d_p = \max(|x - p_x|, |y - p_y|)$$  \hspace{1cm} (1)
The heuristic for the path planner is given by

\[ h(d_p, \rho) = w_d d_p + w_\rho \rho \]  

(2)

where \( w_d \) and \( w_\rho \) are the weights representing the importance of each parameter. Although additional considerations can be added to the path-finding heuristics to model different behaviors and scenarios, having more considerations tends to diminish the effects of existing factors. Therefore, customizing the heuristics according to the required scenario rather than considering an exhaustive list of factors will provide better controls over the generated behaviors.

As we have discussed before, the macro navigation component is used to plan a rough path for an agent to move from a starting location to a destination. Similar to real pedestrians, an agent cannot predict all possible situations. In case an agent encountered some unexpected difficulties that make the previous path plan invalid, the agent needs to do the re-planning (for instance using an \( A^* \) algorithm) which may simulate the improvisation. The micro navigation component is used to steer the agent to move along the planned path while avoiding dynamic objects such as other agents. Only during critical failures of the rough path will path re-planning be executed. Otherwise, the micro-level navigation model will handle the failures. Critical failures in the rough path can be for example, a previously accessible path being completely blocked.

4.2 Micro-Level Navigation Model

Pedestrian navigation is an interactive and dynamic process. The movement of a pedestrian is affected by the movement of other pedestrians in the environment. In turn, a pedestrian’s own movement will also affect the movement of other neighboring pedestrians. However, these mutual influences in navigation are localized and do little to affect the overall path plan for a pedestrian. For example, pedestrians do not replan their path just because they are about to collide with the pedestrian in front of them. We therefore term this interactive steering to minimize physical contacts with other pedestrians in the environment as reactive planning. Reactive planning therefore consists of two main challenges, collision prediction and collision avoidance. Furthermore, physical factors such as the pedestrian’s age and habits, for example, the pedestrian’s comfortable walking speed, also influence the way in which the pedestrian navigates.
which they avoid other pedestrians or move in the environment. In this section, we will describe the basic framework to achieve pedestrian-like agent steering behavior.

### 4.3 Collision Prediction

When an agent is first initialized, it will be given a random number of attention points. The amount of attention points will be used to determine how many foreign objects the agent is capable of tracking at any one time. When a target agent first enters an agent’s vision, touch or hearing range, the target agent will be ranked according to its relative position and speed to determine if the target agent is to be given attention. The higher the risk of collision, the more attention points are allocated by the agent to obtain information about the target agent. Attention points are then reclaimed when the agent finds no risk of collision with the target agent or when the target agent leaves the agent’s sensory range.

The central idea of the agent’s micro-level navigation model lies in the relative frame and the duality property of the relative frame. We will discuss the relative frame in this section and the duality property in Section 4.4. A relative frame provides a mathematical framework by which the movement of two agents can be better analyzed. Let the source agent be the agent that is avoiding a target agent. Then the symbols and parameters used to describe a relative frame can be summarized in Table 6. A relative frame allows us to treat a target agent as a stationary object and focus on the steering parameters of the source agent for collision avoidance. Note that the assignment of the source and target is relative, i.e., the source agent is also a target agent from the target agent’s point of view.

Figure 7 shows the relative frame between a source agent and a target agent. The collision zone is defined as a region of space that the source agent should avoid to prevent collision with the target agent, i.e., collision is predicted some time in the future if

\[ \theta_{r_{\text{min}}}(t) \leq \theta_r(t) \leq \theta_{r_{\text{max}}}(t) \]  

and if the two agents do not change their current speed and orientation. We have used the agent’s personal space for collision prediction instead of its actual geometry because we observe that real pedestrians tend to keep the integrity of this invisible personal space during navigation. For example, pedestrians do not walk pass each other rubbing shoulders when there is obviously ample space to move around.

### 4.4 Collision Avoidance Constraint

The collision avoidance constraint is used to restrict the movements of the source agent so as to avoid a predicted collision with a target agent. In particular, if a collision has been predicted, then from condition (3), the collision can be avoided if the source agent selects a new relative velocity, \( \nabla_r(t + \Delta t) \), that satisfies the condition

\[ \neg (\theta_{r_{\text{min}}}(t) \leq \theta_r(t) \leq \theta_{r_{\text{max}}}(t)) \]  

Since condition (4) represents a range of possible values of \( \theta_r(t + \Delta t) \) for the source agent to avoid a collision, a specific \( \theta_r(t + \Delta t) \) needs to be selected. The choice of \( \theta_r(t + \Delta t) \) is called the collision avoidance strategy of the agent. In this paper, we have randomly chosen \( \theta_r(t + \Delta t) \) to be either \( \theta_{r_{\text{min}}}(t) \) or \( \theta_{r_{\text{max}}}(t) \), which models the
<table>
<thead>
<tr>
<th>Symbol(s)</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_s(t)$, $s_g(t)$</td>
<td>speed of source and target agent at time $t$, respectively.</td>
</tr>
<tr>
<td>$\vec{v}_s(t)$, $\vec{v}_g(t)$</td>
<td>velocity of source and target agent at time $t$, respectively.</td>
</tr>
<tr>
<td>$\vec{v}_r(t)$</td>
<td>relative velocity between the source and target agent at time $t$.</td>
</tr>
<tr>
<td>$\vec{P}_s(t)$, $\vec{P}_g(t)$</td>
<td>displacement of source and target agent at time $t$, respectively.</td>
</tr>
<tr>
<td>$\theta_s(t)$, $\theta_g(t)$</td>
<td>orientation of source and target agent at time $t$, respectively.</td>
</tr>
<tr>
<td>$\theta_r(t)$</td>
<td>relative orientation between the source and target agent at time $t$.</td>
</tr>
<tr>
<td>$r_s$, $r_g$</td>
<td>personal space of the source and target agent, respectively.</td>
</tr>
<tr>
<td>$r_r$</td>
<td>relative personal space between the source and target agent.</td>
</tr>
</tbody>
</table>

Fig. 6. Summary of symbols used.

minimum effort of the agent to avoid a potential collision. However, the choice of $\theta_r(t + \Delta t)$ can also be selected based on empirical studies about pedestrians.

We can also observe from Figure 7 that the corresponding relative speed, $s_r(t + \Delta t)$, does not have a direct effect on the source agent’s ability to avoid a collision. As a result, there exists more than one combination of $s_s(t + \Delta t)$ and $\theta_s(t + \Delta t)$ that is able to satisfy a specific $\theta_r(t + \Delta t)$. This is illustrated in Figure 8. This property naturally reflects the fact that pedestrians can choose to change their personal space.
speed, orientation or both when avoiding other pedestrians.

However, there is an important relation between the choice of $s_s(t + \Delta t)$ and $\theta_s(t + \Delta t)$. This relation is called the duality property in this paper, and is given by the following equation:

$$s_s(t + \Delta t) \sin(\theta_s(t + \Delta t) - \theta_r(t + \Delta t)) = s_g(t) \sin(\theta_g(t) - \theta_r(t + \Delta t))$$  \hspace{1cm} (5)

where $s_s(t + \Delta t)$, $\theta_s(t + \Delta t)$, $s_g(t)$ and $\theta_g(t)$ are the speed and orientation of the source and target agent, respectively. $\theta_r(t + \Delta t)$ is the required direction of the relative velocity between the two agents to avoid a collision. The duality property naturally reflects how people attempt to avoid collision in daily life. For example, some people may attempt to steer clear of a collision by changing their orientation while minimizing changes to their speed. Other people may use short bursts of speed but minimize their deviation from the current path to avoid collision. As illustrated in Figure 9, the duality property can be easily proved. It is worth to note that in equation 5, $s_g(t)$ and $\theta_g(t)$ are used, which reflects the fact that when the source agent is deciding its new speed and orientation for the next step, it has no knowledge about the target agent’s new steering parameters in the next step.

To determine the desired speed and orientation for the source agent to avoid the target agent, we have assumed that pedestrians prefer to minimize changes in their orientation when avoiding structural obstacles or other pedestrians in this paper. To this end, we need to derive the constraints on the source agent’s steering

![Fig. 8. Multiple choices of $s_s(t + \Delta t)$ and $\theta_s(t + \Delta t)$](image)

![Fig. 9. The Duality Property.](image)
parameters in terms of $\theta_s(t+\Delta t)$. If instead, it is required to model pedestrians who prefer to minimize changes in their speed, then the derived constraints should be written in terms of $s_s(t+\Delta t)$.

Given the selected $\theta_r(t+\Delta t)$, we need to first find the required change in the source agent’s orientation for collision avoidance. Then the speed of the agent can be found using the duality property as shown in Equation 5. Let

$$\omega_s = \theta_s(t+\Delta t) - \theta_s(t)$$

Then we have the following collision avoidance constraint ($l_1$) on the source agent’s $\theta_s(t+\Delta t)$:

If $\sin(\theta_g - \theta_r) < 0$,

$$(k_1 < \omega_s < k_1 + \pi) \land (k_2 - \pi < \omega_s < k_2)$$

otherwise,

$$(k_1 - \pi < \omega_s < k_1) \land (k_2 < \omega_s < k_2 + \pi)$$

where $k_1 = \theta_r(t+\Delta t) - \theta_s(t)$ and $k_2 = \theta_g(t) - \theta_s(t)$. In the case of $\sin(\theta_g(t) - \theta_r(t)) = 0$, then the velocity of the target agent is collinear with the relative velocity between the source and target agent. This is a trivial case and hence not discussed in this paper. We will now give the proof for $l_1$. Note that we have sometimes omitted the notion of $t$ in the proof for simplicity.

**Proof:**

Using the duality property,

$$\vec{v}_s - \vec{v}_g = \vec{v}_r$$

∴ $s_s \sin \theta_s = s_g \sin \theta_g = s_r \sin \theta_r$

Using the duality property,

$$\frac{\sin(\theta_g - \theta_r) \sin \theta_s}{\sin(\theta_s - \theta_r)} - \sin \theta_g = \frac{s_r}{s_g} \sin \theta_r$$

Equation 9 can then be simplified as,

$$\frac{\sin(\theta_s - \theta_g)}{\sin(\theta_s - \theta_r)} = \frac{s_r}{s_g}$$

Given $s_r(t) > 0$, $s_g(t) > 0$ and $s_s(t) > 0$, therefore

$$\sin(\theta_s - \theta_g) \sin(\theta_s - \theta_r) < 0$$

Using the same argument, then according to Equation 5, both $\sin(\omega_s + \theta_s(t) - \theta_r(t+\Delta t))$ and $\sin(\theta_g(t) - \theta_r(t+\Delta t))$ must have the same sign. That is,

$$\sin(\theta_s - \theta_r) \sin(\theta_g - \theta_r) > 0$$

From Equations 11 and 12, it will then be trivial to show that $l_1$ is true.

In addition to the collision avoidance constraints, pedestrians may have their own comfortable walking speed and destinations to reach. As a result, having collision avoidance constraints alone may not be sufficient to describe a very rich set of pedestrian navigation behaviors. Therefore, more constraints on the agent’s steering parameters are needed to generate more realistic navigation behaviors.
4.5 Other Constraints

We will now describe two other basic constraints, speed constraints and orientation constraints. Speed constraints can be used to model a pedestrian’s maximum attainable speed or comfortable walking speed. Speed constraints take the form of

\[ s_{\text{min}}(t) \leq s(t) \leq s_{\text{max}}(t) \] (13)

where \( s_{\text{min}}(t) \) and \( s_{\text{max}}(t) \) represents the minimum and maximum speed of the source agent respectively. The speed constraint can be used to restrict the speed of the agent for the modeling of different pedestrians. e.g., old pedestrians will be expected to have a smaller \( s_{\text{max}}(t) \) than a younger pedestrian. However, in the case of a predicted collision, the form as described by Equation 13 cannot be directly combined with the collision avoidance constraint, i.e., Equations 7 and 8. This is because Equation 13 is in the speed domain while the collision avoidance constraints are in the orientation domain.

In order to effectively combine speed constraints with orientation constraints, the duality property can be used to rewrite the agent’s speed in terms of its orientation. Specifically, the speed constraint of the agent can be re-written in terms of \( \omega_s \) as follows:

If \( \sin(\theta_g - \theta_r) > 0 \),

\[ \frac{k_3}{s_{\text{min}}(t)} \leq k_4 \leq \frac{k_3}{s_{\text{max}}(t)} \] (14)

otherwise,

\[ \frac{k_3}{s_{\text{max}}(t)} \leq k_4 \leq \frac{k_3}{s_{\text{min}}(t)} \] (15)

where \( k_3 = s_g(t) \sin(\theta_g(t) - \theta_r(t + \Delta t)) \) and \( k_4 = \sin(\omega_t + \theta_s(t) - \theta_r(t + \Delta t)) \).

Equation 15 can then be easily solved based on the values of \( k_3 \), \( s_{\text{min}}(t) \) and \( s_{\text{max}}(t) \) to obtain the final speed constraints in terms of \( \omega_s \).

Orientation constraints can be used to represent a pedestrian’s desire to stay on course in order to reach a location in the environment or to describe a pedestrian’s tolerance to changes in orientation during navigation. For example, to model a pedestrian’s unwillingness to change orientation during navigation, it is possible to restrict the value of \( \omega_s \) to a small range. Orientation constraints for the source agent can be written in terms of \( \omega_s \) as:

\[ \theta_{\text{min}}(t) - \theta_s \leq \omega_s \leq \theta_{\text{max}}(t) - \theta_s \] (16)

where \( \theta_{\text{min}}(t) \) and \( \theta_{\text{max}}(t) \) represents the orientation limits of the source agent.

5. BEHAVIORAL MODELING

In this section, we describe how various factors affecting pedestrian navigation can be considered by our navigation model using different constraints for the micro-level navigation model and heuristics for the macro-level navigation model. All these behaviors can be generated using the constraints that we have discussed in Sections 4.1 and 4.2. From a behavior modeler’s point of view, these constraints are quite natural and easy to specify. Complex navigation behaviors such as lane forming or overtaking in a crowd naturally emerge as a result of these constraints.
5.1 Comfortable Walking Speed

Pedestrian movement can be affected by the person’s age and other physical capabilities. For example, Le pointed out the differences in human gait cycle and its relationship with human locomotion [Le et al. 2003], i.e., how much distance a person can travel with each step. In addition to differences in the maximum attainable speed, different people may have different comfortable walking speeds.

According to Bohannon, there is no significant difference in the average comfortable walking speed for pedestrians aged between 20 and 50 while there is a small difference for those aged above 50 [Bohannon 1997]. There are however significant differences in the maximum comfortable walking speed for pedestrians in all different age groups. For example, an average male in his 20s will have a mean and maximum comfortable walk speed of 139cm/s and 253cm/s, respectively. On the other hand, an average female in her 50s will have a mean and maximum comfortable walking speed of 139cm/s and 201cm/s, respectively.

In our proposed model, a pedestrian’s comfortable walking speed can be modeled by introducing a speed constraint on an agent’s steering parameters. For example, by introducing a speed constraint $120 \leq s(t) \leq 140$ and combining it with the collision avoidance constraint, the source agent will prefer to move with a speed between 120cm/s to 140cm/s even when it tries to avoid other agents.

5.2 Time Constraint

Pedestrian movement may also be influenced by situational factors like time constraints. For example, a pedestrian who is late for work usually wants to move faster to reach the place of work. In such cases, it is common for the pedestrian to also lose patience in waiting behind slower moving pedestrians, i.e., an impatient pedestrian may overtake a slower pedestrian in front.

This behavior can be naturally modeled by increasing the minimum speed, i.e., an impatient agent will have a relatively higher minimum speed. For example, by creating a speed constraint of $80 \leq s(t)$ and combining it with a collision avoidance constraint, the agent will attempt to overtake any agent moving slower than 80 cm/s. This agent will appear to be more impatient than another agent with a minimum speed constraint of $40 \leq s(t)$. Note that the agent does not have to follow a predefined sequence of speed changes in order to overtake a slower moving agent nor does it switch to a routine that contains instructions for overtaking. Instead, the combination of constraints forces the micro-level navigation model to generate a new set of steering parameters at each update cycle. That is, the overtaking behavior emerges as a result of these constraints. We will further demonstrate this behavior in 6.1.

5.3 Different Route Choices

Different people may have different knowledge about the environment and as a result may use different routes to reach their destinations. In particular, not all pedestrians have the knowledge about the shortest possible route to a destination. Furthermore, a pedestrian’s route choice may also be influenced by a variety of different psychological factors like personal habits and preferences.

Different route choices can be modeled using our proposed visual memory rep-
presentation. Since each agent has an independent abstract representation of the
environment for path-finding, partial knowledge about the environment can be
modeled by introducing some missing regions of the environment in the agent’s
visual memory. Similarly, personal preferences can be added as a heuristic cost
for the path-finder when evaluating a route. In this way, it is possible to model
the different route choices of individual agents based on the psychological factors
considered by a pedestrian during navigation.

5.4 Social Norm
In a densely packed metropolis like Tokyo, London and New York, it is common
to see hundreds or even thousands of people packed into small places like a train
station. The pedestrians in these cultures can also be observed to self-organize
themselves into lanes to expedite their own movement through a crowded area.
Furthermore, there exists social norms for people traveling in a particular direction
to keep to their own side of the lane (e.g., keeping left or right). It is therefore
important for a realistic pedestrian simulation to achieve the same behavior during
the simulation of a large gathering of pedestrians. Social norms can be modeled by
introducing additional constraints on an agent’s steering parameters. For example,
an orientation constraint will keep the agent moving on the left or right side of a
corridor according to the social norm being modeled. We will further demonstrate
this behavior in Section 6.3.

5.5 Group Movement
In real-life, pedestrians can be found to move in groups such as with family, friends
or colleagues. The composition of the group will then influence how the group as a
whole will react to the movement of other pedestrians or groups, such as whether
the group will give way and avoid other pedestrians or whether the group will tend
to stay together when moving around in the environment. Furthermore, factors
like social status and social context will affect the way a group will move. For
example, a bodyguard may need to both match the speed and maintain proximity
with the important person. Speed matching can be easily achieved by introducing
a speed constraint on the body guard (agent) to dynamically match its speed with
the speed of the important person (agent). To maintain the distance between the
body guard and the important person, whenever they become too far apart, the
body guard can be set to deliberately neglect potential collisions with other agents
in order to keep up with the important person. We will demonstrate this behavior
in Section 6.2.

6. IMPLEMENTATION AND RESULTS
Our pedestrian model has been prototyped using the A7 Game Studio game engine
for rendering and basic framework operations. The prototype was programmed
using C++ and compiled using Microsoft Visual C++ Express 6.0. The prototype
has been bench marked using a Pentium 1.66GHz Dual Core computer with 1GB
RAM. We have tested the performance of the prototype with various scenarios and
produced some demo videos. In this section, we will summarize some of our major
results.
6.1 Basic Navigation Behaviors

In this section, we will demonstrate the basic navigation capabilities of the agents using only our micro-level navigation module. The demonstrated behaviors may look trivial, but they are important building blocks for more complex navigation behaviors such as lane cutting and lane formation. It is worth noting that all the behaviors demonstrated by our agents are actually emergent behaviors. That is, our agents are not given rules to recognize and react to the various pedestrian configurations. Instead, these behaviors emerge solely based on the constraints that we have given to these agents. For example, we set constraints that limit the agent’s speed and the need to change its current orientation. We do not explicitly specify a lane for the agent to follow.

Figure 10 shows some of the common pedestrian behaviors that are demonstrated by our agents. Figure 10(a) shows the avoidance behavior which is one of the most common responses for pedestrians when a collision is predicted. The overtaking behavior as shown in Figure 10(b), occurs when one of the front agents moves slower than the minimum acceptable speed of the agent walking behind. In contrast, Figure 10(c) shows the following behavior where the group in front is traveling faster than the minimum acceptable speed of the agent walking behind. Note that the decision to overtake or to follow is made according to the goals and personality of the agent which can be easily incorporated into our model by the various constraints.

Figure 10(d) demonstrates the separation behavior as two or more agents move-
19

Fig. 11. Group movements.

ing to avoid collision with another agent. The demonstrated behavior naturally emerges as a result of each agent following their own constraints while avoiding other agents in their neighborhood. This behavior is a another natural behavior that is commonly seen in pedestrian navigation.

6.2 Group Movement

Group movement is an interesting area of research for pedestrian simulation. Pedestrians as social creatures will react differently when they are alone or in groups. In this section we will illustrate two interesting pedestrian behaviors that we have produced in our experiments.

In the first experiment, two groups of pedestrians are generated. One group is moving from left to right (left group) and the other from right to left (right group). Figure 11(a) shows how the two groups will finger across each other to reach their destination. This behavior can be easily observed in many real-life situations.

In the second experiment, the left group is given a higher social status than the right group and it is assumed that the difference in social status is apparent to both groups. In real-life, a group’s social status can be easily derived by observing their costumes and social etiquette. From Figure 11(b) we can observe that the right group begins to scatter around the left group when the two groups meet. This behavior can be seen for example in an entourage of some important peoples (left group), such as city mayors, mafia bosses or celebrities. This group of people will not split from the important figures and expect all other pedestrians to give way to them. As a result, they tend not to change their headings to avoid other
pedestrians. This behavior can also be used to model the movement of social groups with different social status through the streets of a city. Groups with higher social status will have the right of way over groups with a lower social status and will hence maintain their current headings. Groups with lower social status will have to give way because of their lower social status.

6.3 Dense Pedestrian Gathering

Modeling a dense pedestrian gathering offers unique challenges because it allows for more complex navigation behaviors to surface as a result of the navigation choices that are made by individuals within the gathering. In real-life, pedestrians in a crowded area tend to self-organize into lanes in order to reach their destination faster and easier. A dense pedestrian gathering can be seen in a commuter’s hub during peak hours or at the entrance of a stadium in the event of a popular football match. Congestion appears when there is a large difference in throughput at both ends of an area. In this experiment, we model a scenario in a commuter’s hub where the input and output flow from the area is high but relatively constant such that no visible congestion has occurred. The resulting lane formation phenomenon is then demonstrated in Figure 11.

Figure 11(c) demonstrates an ad-hoc lane formation behavior. It can be observed that temporarily formed openings are used by the other agents using the following behavior until the gap eventually closes and a new lane is formed elsewhere in the crowded area. This behavior is common in trade fairs and exhibitions where pedestrians form ad-hoc lanes in order to ease their navigation through a crowded area.

Figure 11(d) shows the voluntary lane formation behavior. It can be observed that the lanes can be maintained for a relatively long period after they are formed as compared to the simulation in Figure 11(c). This behavior is common in crowded train stations or underground passes where lanes are voluntarily formed to improve the flow of pedestrians. It is also common to see various pedestrian signs in these locations advising pedestrians to keep left (or right) and follow the pedestrian flow.

Figure 12 shows the effect of personal space on the distribution of agents for pedestrian simulations of varying density. Note that the lane formation behavior occurs at lower pedestrian density when the personal space of an individual agent is relatively large (compare Figure 12(a) and Figure 12(d)). It can also be observed that agents with a larger personal space generate more realistic results in scenarios with a lower density of pedestrians. On the other hand, agents with smaller personal space generate more realistic results in scenarios with a higher density of pedestrians. (compare Figure 12(c) and 12(f)). This suggests that pedestrians have a smaller personal space in more crowded areas than in less crowded areas.

6.4 Complex Environments

Figure 13 shows a scenario where the macro-level navigation module is used together with the micro-level navigation module to help the agents find their way around structural obstacles. In Figure 13(a), we have agents moving from left to right while in Figure 13(b), we have agents moving in different directions. This scenario demonstrates that the navigation module of our agent is capable of avoiding odd shaped structural obstacles and other moving agents at the same time.
This capability is important in pedestrian simulations with urban or indoor environments.

Figure 14 shows a shopping mall scenario that we have implemented to demonstrate both the path finding and congestion avoidance behaviors. Unlike the scenario shown in Figure 13, agents in this scenario are given three distinct roles. They are the shoppers, the store attendants and the passers-by. Shoppers will enter the area with a list of shops to visit and merchandizes to purchase. Shoppers will then visit the different stores to browse for merchandizes and store attendants will patrol the shop attending to shoppers. Passers-by will just pass through the area, entering from one corner of the mall and exiting from another. Figure 14 shows
agent taking detours when part of the mall becomes overly congested ($t=20$). This behavior will help ensure an even distribution of agents throughout the scenario. In contrast, without the congestion avoidance behavior, agents may be found to crowd in one side of the environment. This behavior can for example be used to simulate people gathering around a performance show area.

6.5 Performance Analysis

In order to analyze the run-time performance of our proposed model, the scenario shown in Figure 13 has been used to benchmark the rendering period of the simulation. The environment is represented by approximately 1000 polygons and each agent is represented using approximately 150 polygons. In order to reduce the overhead for running the simulation, free-lists have been used to help manage the memory requests for the simulation [Johnstone and Wilson 1999].

Figure 15 shows the time needed to compute the movement of all agents and to render each frame. The left axis shows the rendering period and the right axis shows the number of agents in the scene. The rendering period is obtained by counting the number of milliseconds from the start of an update cycle to its end. It takes approximately 150ms to update 200 agents.

Figure 16 shows the number of actions taken by all agents in the environment for each frame. For example, at $t = 1000$, a total of ten path replanning actions and fifty five steering adjustments to avoid collision were made.

When an agent undertakes an avoidance steering behavior, it may deviate from...
its current planned path. However, waypoints are represented as an area, rather than a single point. This allows agents to continue along their current path without

ACM Journal Name, Vol. V, No. N, Month 20YY.
having to strictly align themselves with the target waypoint or to replan their paths. In this way, we aim to reduce the number of computationally expensive path-replanning actions and to better reflect the way a real pedestrian moves around the environment. That is, path replanning is performed by the macro-level navigation module while steering adjustments are performed by the micro-level navigation module.

7. CONCLUSION AND FUTURE WORK

We have adopted an agent based approach to the modeling of pedestrians and proposed a navigation model based on observations about real pedestrians. The proposed navigation model is able to combine various considerations used by pedestrians during navigation by formulating collision avoidance, speed and orientation constraints. In particular, we discussed how physical, psychological and situational factors can be modeled using different steering constraints. As a result, the navigation model is able to demonstrate some interesting individual and group navigation behaviors. We have also demonstrated that our proposed navigation model is able to naturally demonstrate lane formation behaviors in a scenario with higher pedestrian density without explicit instructions.

However, some important challenges still exist for the simulation of a realistic scenario with high pedestrian density. For example, pedestrian behaviors such as queuing and pushing are still not part of our current pedestrian navigation module. It will also be interesting to study how a more realistic human attention system can be modeled. In particular, how real humans select objects to include in their attention system can be incorporated into our attention model to reduce the computational costs of deriving an agent’s candidate list of objects for collision detection. The agent’s abstract map of the environment can also be refined by cross-referencing works on mental maps in the field of social psychology. Finally, our proposed model generates pedestrian behaviors based on known social norms. It will be interesting to allow agents to develop (for instance, using some evolutionary models) their own social norms by providing them with certain cultural knowledge. We will look into these issues in our future work.

REFERENCES


ACM Journal Name, Vol. V, No. N, Month 20YY.


