Large-Scale Animation of Autonomous Pedestrians in Urban Environments

Anonymous

Abstract
We address the difficult open problem of emulating real pedestrians in urban environments. Our approach, which dramatically advances the state of the art of pedestrian animation in terms of its scope, scale, robustness, and efficiency, is one of comprehensive artificial life modeling, integrating motor, perceptual, behavioral, and cognitive components. Our major contribution in this paper is a sophisticated new model of autonomous pedestrians that includes innovations in each of these components, as well as in their combination, yielding results of unprecedented fidelity and complexity for fully autonomous multi-human animation in large-scale virtual worlds. To support a variety of natural interactions among pedestrians in an extensive and highly dynamic environment, we represent the latter using hierarchical data structures that efficiently execute the perceptual queries of autonomous pedestrians to sustain their behavioral responses and to enable them to plan their actions on global and local scales. We present a virtual human animation system that can automatically synthesize a rich variety of activities involving multitudes of autonomous pedestrians populating a realistically modeled train station environment. Our autonomous virtual human animation system has applications both inside and outside the field of computer graphics.

CR Categories: 1.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation; 1.6.8 [Simulation and Modeling]: Types of Simulation—Animation.

Keywords: Virtual humans, virtual worlds, pedestrian simulation, behavioral animation, perceptual modeling, cognitive modeling, artificial life.

1 Introduction

“Forty years ago today at 9 a.m., in a light rain, jackhammers began tearing at the granite walls of the soon-to-be-demolished Pennsylvania Station, an event that the editorial page of The New York Times termed a "monumental act of vandalism" that was "the shame of New York."”

(Glenn Collins, The New York Times, 10/28/03)

The demolition of New York City’s original Pennsylvania Station, which had opened to the public in 1910 (Figure 2), in order to make way for the Penn Plaza complex and Madison Square Garden, was “a tragic loss of architectural grandeur”. Fortunately, standard computer graphics techniques enable a virtual reconstruction of the train station with stunning geometric and photometric detail, at least in principle. Unfortunately, they have not yet enabled us to reconstruct and animate the station’s human occupants with anywhere near as much fidelity. In this paper, we address this latter challenge.

To this end, we develop an entirely autonomous pedestrian model that requires no centralized, global control whatsoever and is capable of performing a variety of activities in synthetic urban spaces, such as a virtual Penn Station (Figure 1). Our virtual human model incorporates the modeling of pedestrian appearance, locomotion, perception, behavior, and cognition. We deploy a multitude of these self-animating pedestrians within a large-scale environment model, which includes hierarchical data structures that support the efficient interaction between numerous pedestrians and their complex virtual world through fast (perceptual) query algorithms and navigation on local and global scales.

The next section reviews related work. Section 3 presents our virtual environment model, describing its graph map, grid map, and perceptual query components. Section 4 presents our autonomous pedestrian model, describing its appearance and motor emulation components but, more importantly, developing its behavioral modeling, cognitive modeling and control components. Section 5 presents a selection of results including long term simulations involving well over 1000 pedestrians and reports on performance. Section 6 briefly describes ongoing applications of our pedestrian simulator outside computer graphics. Section 7 concludes the paper and discusses future work.

2 Related Work

Psychologists and sociologists have been studying the behavior and activities of people for many years. Closer to home, pedestrian simulation has recently begun to capture the imagination of CG researchers [Ashida et al. 2001; Muse and Thalmann 2001].
The topic has also been of some interest in the related field of artificial life [Blue and Adler 2000], as well as in architecture and urban planning [Lovas 1993; Schreckenberg and Sharma 2001] where graphics researchers have assisted in visualizing planned construction projects, including pedestrian animation [Farenc et al. 2000; Metoyer and Hodgins 2003]. Our work is applicable to all of these areas and to others, such as computer vision.

For a survey of issues relevant to human simulation in general, see the book by [Badler et al. 1993]. The seminal work of Reynolds [1987] on behavioral animation is highly relevant to our approach. It has given impetus to an entire industry of applications for distributed (multiagent) behavioral systems that are capable of synthesizing flocking, schooling, herding, etc., behaviors for lower animals, or in the case of human characters, crowd behavior. Behavioral animation has been applied heavily in the movie industry by Disney and many other production houses—most notably in recent, large-scale battle scenes in feature films [Massive, 2003]. Low-level crowd interaction models have been developed in the sciences [Batty et al. 1998; Gipps and Marksjo 1985; Helbing and Molnar 1995; Schadschneider 2002] and they have also been explored by animation researchers [Goldenstein et al., 2001]. Our work makes innovations at the behavioral level, but its scope extends to higher-level, cognitive modeling of human intelligence.

In pedestrian animation, the bulk of prior research has focused on synthesizing natural locomotion (a problem that we do not consider in this paper) and on path planning (one that we do). A pioneering effort on autonomous navigation is that by Noser et al. [1995]. Metoyer and Hodgins [2003] develop an interesting model for reactive path planning in which the user can refine the motion by directing the characters with navigation primitives. We prefer to have our pedestrians navigate entirely on their own, as normal biological humans are capable of doing. To this end, we adopt a comprehensive artificial life approach to addressing the problem of pedestrian animation. Our approach was inspired most heavily by the work of Tu and Terzopoulos [1994] on artificial animals and by Funge et al. [1999] on cognitive modeling for intelligent characters that can reason and plan their actions. We further develop this approach and adopt it for the first time to the case of virtual humans occupying large-scale urban environments.

3 Virtual Environment Model

The interaction between a pedestrian and his/her environment plays a major role in the animation of autonomous virtual humans in synthetic urban spaces. This, in turn, depends heavily on the representation and (perceptual) interpretation of the environment. Organizing the virtual environment into a simple list of geometric objects might suffice for modeling small worlds with only a few objects and pedestrians, but it is grossly inadequate when attempting to synthesize a large urban space populated by numerous pedestrians, such as a busy train station.

We have devoted considerable effort to the development of a large-scale urban environment model, which is described in detail elsewhere [Anonymous 2005] (note to reviewers: this paper is provided as supplementary material), and which we review next. We represent the virtual environment by a hierarchical collection of maps that support efficient environmental information storage and retrieval. As Figure 3 illustrates, the model comprises (i) a topological map which represents the topological structure between different parts of the virtual world. Linked within the topological map are (ii) perception maps, which efficiently provide relevant information in response to the perceptual queries issued by pedestrians, and (iii) path maps, which enable the pedestrians to perform efficient goal-directed navigation and planning of routes.

In the topological map, nodes correspond to environmental regions and edges between nodes represent accessibility between regions. A region is a bounded volume in 3D-space (such as a room, a corridor, a flight of stairs or even an entire floor) together with all the objects inside that volume (for example, ground, walls, ticket booths, benches, vending machines, etc.). The representation assumes that the walkable surface in a region may be mapped onto a horizontal plane without loss of essential geometric information (the sizes of objects and distances between locations). Consequently, a 3D-space may be adequately represented within the topological map by several 2D, planar maps, thereby enhancing the simplicity and efficiency of environmental queries.

The perception maps include grid maps that represent stationary environmental objects on a local, per-region basis, as well as a global grid map that represents mobile objects, usually other pedestrians. Each of these uniform grid maps store information within each of their cells that identifies all of the objects occupying that cellular area. The typical cell size of the grid maps for
stationary object perception is 0.2–0.3 meters. Each cell of the mobile grid map stores and updates identifiers of all the agents currently within its cellular area. Since it serves simply to identify the nearby agents, rather than to determine their exact positions, it employs cells whose size is commensurate with the pedestrian’s visual sensing range (currently set to 5 meters).

The path maps support efficient online path planning. They include a quadtree map which supports global, long-range path planning and a grid map which supports short-range path planning. Each node of the quadtree map stores information about its level in the quadtree, the position in the world of the area represented by the node, the occupancy type (ground, obstacle, seat, etc.), and pointers to neighboring nodes, as well as information for use in path planning, such as a distance variable (indicating how far the area represented by the node is from a given start point) and a congestion factor (the portion of the area of the node that is occupied by pedestrians). The quadtree map supports the execution of several variants of the A* graph search algorithm, which are employed to compute quasi-optimal paths to desired goals. It is used for planning about 94% of the paths. The remaining 6% of the paths are planned using the path grid map, which also supports the execution of A* and provides detailed, short-range paths to goals in the presence of obstacles, as necessary. A typical example of its use is when a pedestrian is behind a chair or bench and must navigate around it in order to sit down.

Excluding DI-Guy motion synthesis and character/environment rendering time, our environment model is efficient enough to support the real-time (30fps) animation of about 1400 pedestrians on a 2.8GHz Xeon PC with 1GB memory. For the details about the construction and update of our environment model and associated performance statistics regarding its use in perception and path planning, we refer the reader to [Anonymous 2005].

4 Autonomous Pedestrian Model

Like real humans, our synthetic pedestrians are fully autonomous. They perceive the virtual environment around them, analyze environmental situations, make decisions and behave naturally. Our autonomous human characters are structured in accordance with a hierarchical character model similar to the one advocated by Funge et al. [1999]. Progressing through levels of abstraction up the “modeling pyramid”, our model incorporates appearance, motor, perception, behavior, and cognition sub-models. The following sections discuss each of these components in turn. Of course, our pedestrian-specific sub-models differ from those in Funge et al.’s work, which involved an aquatic “merman” and dinosaurs.

4.1 Human Appearance, Movement, & Motor Control: “Augmented DI-Guy”

As an implementation of the low-level appearance and motor levels of the pyramid, we employ a human animation package called DI-Guy, which is commercially available from Boston Dynamics Inc. [Koechling et al. 1998]. DI-Guy provides a variety of textured geometric models that portray different people. These character models are capable of basic motor skills, such as strolling, walking, jogging, sitting, etc., implemented using conventional IK and motion library techniques. DI-Guy is intended as an application that enables users to script the actions of human characters in space and time. To facilitate this task, it provides an interactive scripting interface called DI-Guy Scenario, which we do not use in our work. However, it also provides an SDK that enables each character’s motor repertoire to be controlled by external user-specified C/C++ programs. By augmenting DI-Guy with our own control programs at the motor, perceptual, behavioral, and cognitive levels, we have successfully developed a fully self-animating virtual human model that is capable of synthesizing a relatively rich variety of autonomous behaviors and actions associated with pedestrians in urban environments.

Emulating the natural appearance and movement of human beings is a highly challenging problem and, not surprisingly, DI-Guy suffers from several limitations. In particular, the DI-Guy geometric human models are insufficiently detailed to provide pleasing renderings of humans for close-up viewing. We have made no attempt to ameliorate this situation as the DI-Guy appearance code remains a proprietary “black box”. More importantly, although the most advanced DI-Guy characters have reasonably broad action repertoires, they cannot synthesize the full range of motions needed in a busy urban environment that produces frequent close encounters between independently locomoting pedestrians. Specifically, DI-Guy characters are limited in how tightly they can execute turns and how quickly they can change locomotion gaits and speeds, sometimes requiring up to a couple of seconds before completing a transition, depending on the exact instant within the locomotion cycle. Fortunately, the most recent release of the software package provides a Motion Editor interface that enables users to modify and supplement the motion repertoires of DI-Guy characters. We have used this interface to decompose DI-Guy motions, which typically span one entire cycle of a gait, to more elementary motions that include only single steps and other motion primitives. This enables the DI-Guy-based pedestrians to make faster transitions so they can deal with more highly dynamic urban environments.

Moreover, we have implemented a motor control interface between the low level kinematic layer of DI-Guy, and the higher-level behavioral controllers described in the next section. This interface accepts motor control commands from behavior modules, selects the appropriate direction, speed, acceleration, and gait for the pedestrian in accordance with its kinetic limits, and updates the posture and position of the pedestrian using the low level kinematic control mechanism. As motor control commands issued by the higher-level behavior system are not necessarily physically achievable, our motor control layer is responsible for verifying and correcting them (e.g., speeds or accelerations of big magnitude are trimmed). Given a desired locomotion speed, the motor controller will choose the gait that has the closest speed and, if necessary, call upon the kinematic layer to make as smooth as possible a transition from the current gait to the new one.

Hence, our motor control layer is a seamless interface that hides the details of the underlying kinematic layer from the higher-level behavior modules, enabling the latter to be developed more or less independently. Despite its remaining limitations, we have found that for the time being our “Augmented DI-Guy” low-level humanoid model usefully supports our research into autonomous human animation through perceptual, behavioral, and cognitive modeling, as we shall describe in subsequent subsections. It is important to note, however, that our higher-level modeling abstractions are designed to be more or less independent of the lower levels; hence, any other suitable low-level API can easily replace DI-Guy in our future work. As case in point, in addition to Augmented DI-Guy, we have implemented a simpler kinematic layer for testing our higher level algorithms, where each pedestrian is represented by a simplistic mobile bounding box.
4.2 Perception

An autonomous and highly mobile virtual human must have a keenly perceptive regard for its environment. Our environment model, reviewed in Section 3, efficiently provides accurate perceptual data in response to the perceptual queries of an autonomous pedestrian.

To ensure that a pedestrian’s feet touch the ground in a natural manner, especially when climbing stairs or locomoting on uneven ground, the pedestrian must query the environment model to sense the local ground height so that the feet can be planted appropriately. Each grid map cell contains the height functions of usually a single and sometimes a few ground objects, such as the floor, stairs, etc. The object of greatest height at the desired foot location is returned in constant time and processed within the pedestrian’s motor layer, which plants the foot at the appropriate height.

The visual sensing computation shoots out a fan of line segments, whose length reflects the desired perceptual range and whose density reflects the desired perceptual acuity, rasterizing each segment into the grid map (Figure 4(a-b)). Grid cells along each line are interrogated for their associated object information. This perceptual query takes time that grows linearly with the length of each line times the number of lines but, most importantly, it does not depend on the number of objects in the virtual environment. Without the help of grid maps, the necessary line-object intersection tests would be time consuming given a large, complex virtual environment populated by numerous pedestrians.

Similarly, a mobile grid map (Figure 4(c)) with larger cell size is employed to sense mobile objects, usually other pedestrians. Note that for the purposes of avoiding collisions, the pedestrian needs to identify other nearby pedestrians, rather than determine their exact positions. The larger cell size does not necessarily decrease the final accuracy; since once the pedestrian obtains general information about nearby pedestrians, it can resolve them better by referring to finer grid maps. Hence, the purpose of the map is to efficiently provide the set of the nearby pedestrians within a sensing range \( R \). To this end, all the cells wholly or partly within a circle of radius \( R \) centered at the pedestrian’s current position are examined and the pedestrians occupying those cells will be included into the pedestrian’s “nearby-agent” set. We set the cell size equal to the sensing range \( R \) such that the number of relevant cells reduces to 9, the center cell and its 8-connected neighbors. This makes the query a very small, constant-time operation.

4.3 Behavioral Control

Realistic human behavioral modeling, whose purpose is to link perception to appropriate actions in an intelligent virtual human, is an enormous hurdle. Even for pedestrians, the complexity of any substantive behavioral repertoire is high. Considerable literature in psychology, ethology, artificial intelligence, robotics, and artificial life is devoted to the subject. Following Tu and Terzopoulos [1994], we adopt a bottom-up strategy that uses primitive reactive behaviors as building blocks that in turn support more complex motivational behaviors, all controlled by an action selection mechanism.

4.3.1 Basic Reactive Behaviors

Reactive behaviors appropriately connect perceptions to immediate actions. Given that a pedestrian possesses a set of motor skills, such as sitting down and rising up, standing, moving forward, turning in different directions, speeding up and slowing down, etc., reactive behaviors are responsible for initiating, terminating, and sequencing these motor skills on a short-term basis guided by sensory stimuli and internal percepts. The timely and accurate perceptual interpretation of afferent sensory data is crucial for meaningful behavioral response. For our synthetic pedestrians, we have developed six key reactive behavior routines, each suitable for a different set of situations related to the navigation of a densely populated and highly dynamic environment. We will first explain each of these behavior routines in detail, and then explain how they are combined.

![Figure 4. Perception](image)

(a) Sensing stationary objects, (a) by examining grid map entries along the rasterized eye ray, while (b) perceiving the broader obstacle situation by shooting our a fan of eye rays. (c) Sensing mobile objects by examining 9 cells covered by the sensing circle (blue pedestrians are sensed; gray are not).

![Figure 5. Reactive behaviors](image)

(a) Safety in turning

Initially heading north, the two pedestrians want to turn southward. They use RB_turn and pick the best turning curves.

(b) Temporary Crowd

Pedestrians that are within \( H \)'s front cut-off parabola and are traveling along similar directions as \( H \) (marked with \( C \) here) are considered to be in \( H \)'s “temporary crowd.”

(c) Avoid cross collision

Left: \( H \) and \( C \) both detect a cross collision ahead. \( H \) slows down and turns left while \( C \) does the opposite. Right: One second later, the potential collision is cleared.

(d) Face-to-face collision

\( H \) and \( C \) travel in opposite directions along nearly-parallel trajectories. To avoid collision of two bodies, they turn slightly away from each other.

**Routine A: Static obstacle avoidance:**

If there is a nearby obstacle in the direction of locomotion, a set of lateral directions within a specified angular extent (currently set to 90 degrees) to the left and right are tested for obstacles and the less cluttered direction is chosen. If neither alternative direction seems better than the current heading, the pedestrian will slow down and perform the aforementioned test again using larger lateral search angles (currently set to 150 degrees). This behavioral response mimics that of a real person upon encountering a tough (set of) obstacle(s); i.e., slow down while turning the head to look around before proceeding.
Routine B: Static obstacle avoidance in a complex turn:
A pedestrian employs this routine whenever it needs to make a turn that cannot be finished in one step (Figure 5(a)). In this routine, turns with increasing curvatures are considered in both directions, starting with the side that permits the smaller turning angle, until a collision-free turn is found. If the surrounding space is too cluttered, the curve is likely to degenerate causing the pedestrian to stop and turn on the spot. The turn test is implemented by checking sample points along a curve with interval equal to the distance of one step of the pedestrian moving with the anticipated turn speed.

Routine C: Maintain separation in a moving crowd:
For a pedestrian $H$, other pedestrians are considered to be in $H$’s “temporary crowd” if they are moving in a similar direction to $H$ and are situated within a parabolic region in front of $H$ defined by $y = -(4/R) x^2 + R$, where $R$ is the sensing range, $y$ is oriented in $H$’s forward direction and $x$ is oriented laterally (Figure 5(b)). To maintain a comfortable distance from each individual $H_i$ in this temporary crowd, a directed repulsive force (cf. [Helbing and Molnar 1995]) given by $f_i = r_i (|d_i| / |d_i| - d_{min})$ is exerted on $H$, where $d_i$ is the vector separation of $H_i$ from $H$, and $d_{min}$ is the predefined minimum distance allowed between $H$ and other pedestrians (usually 2.5 times $H$’s bounding box size). The constant $r_i$ is $H_i$’s perceived “repulsiveness” to $H$ (currently set to -0.025 for all pedestrians). The repulsive acceleration due to $H$’s temporary crowd is given by $a = \sum_i f_i / m$, where $m$ is the “inertia” of $H$. The acceleration vector is decomposed into a forward component $a_f$ and a lateral component $a_l$. The components $\Delta a_f$ and $\Delta c_i a_l$ are added to $H$’s current desired velocity. The crowding factor $c_i$ determines $H$’s willingness to “follow the crowd”, with a smaller value of $c_i$ giving $H$ a greater tendency to do so (currently 1.0 $\leq c_i \leq 5.0$).

Routine D: Avoid oncoming pedestrians:
In a crowded public area, such as a plaza, where pedestrians are moving in various directions, the likelihood of collisions increases dramatically. To mitigate the situation, pedestrian $H$ estimates its own velocity $v$ and the velocities $v_i$ of nearby pedestrians $C_i$. Two types of threats are considered here. By intersecting its own linearly extrapolated trajectory $T$ with the trajectories $T_i$ of each of the $C_i$, pedestrian $H$ identifies potential collision threats of the first type: cross-collision (Figure 5(c)). In the case where trajectories of $H$ and $C_i$ are almost parallel and will not intersect immensly, a face-to-face-collision (Figure 5(d)) may still occur if their lateral separation is too small; hence, $H$ measures its lateral separation from oncoming pedestrians. The details are given in by Algorithm 1 in the Appendix.

Note that each pedestrian evaluates the possibility of collision with respect to the size of its own bounding box $B(H)$ and the size of the bounding boxes $B(C_i)$ of the nearby pedestrians. Bounding box sizes vary from 0.3 to 0.6 meters for different pedestrians, depending on body size. Among all collision threats, $H$ will pick the oncoming pedestrian $C^*$ that poses the most imminent one. If $C^*$ poses a face-to-face-collision threat, $H$ will turn slightly away from $C^*$. If $C^*$ poses a cross-collision threat, $H$ will estimate who will arrive first at the anticipated intersection point $p$. If pedestrian $H$ determines that it will arrive sooner at $p$ than $C^*$, it will increase its speed and turn slightly away from $C^*$; otherwise, it will decrease its speed and turn slightly towards $C^*$ (Figure 5(c)). This behavior will continue for several footsteps, until the potential collision has been averted.

Routine E: Avoid dangerously close pedestrians:
This is the fail-safe behavior routine, reserved for emergencies due to the occasional failure of Strategies C and D, since in highly dynamic situations predictions have a nonzero probability of being incorrect. Once a pedestrian perceives another pedestrian within his forward safe area (Figure 5(e)), it will resort to a simple but effective behavior—brake as soon as possible to a full stop, then try to turn to face away from the intruder, and proceed when the way ahead clears.

Routine F: Verify new directions relative to obstacles:
Since the reactive behavior routines are executed sequentially as explained momentarily, motor control commands issued by Routines C, D or E to avoid pedestrians may counteract those issued by Routines A or B to avoid obstacles, thus steering the pedestrian towards obstacles again. To avoid this, the pedestrian checks the new direction against surrounding obstacles once more. If the way is clear, it proceeds. Otherwise, the original direction issued by either the higher-level path-planning modules or by Routine A, whichever was executed most recently prior to the execution of Routine F, will be used instead. However, occasionally this could lead the pedestrian toward future collisions with other pedestrians (Figure 5(f)) and, if so, it will simply slow down to a stop, let those threatening pedestrians pass, and proceed.

Some remarks regarding the above routines are in order: The intuition behind the strategy of Routine-F is that whenever a pedestrian is faced with threats of collision from both static and mobile obstacles, it is a bad idea to steer toward the former, as that will definitely lead to a collision. Instead, the pedestrian should hesitate until the mobile collision threats pass and then proceed in the direction that is free of static obstacles. Obviously, the fail-safe strategy of Routine E suffices per se to avoid nearly all collisions between pedestrians. However, our experiments show that in the absence of Routines C and D, Routine E makes the dynamic obstacle avoidance behavior appear very awkward—pedestrians stop and turn too frequently and they make slow progress. As we enable Routines C and D, the obstacle avoidance behavior looks increasingly more natural. Interesting multi-agent behavior patterns emerge when all the routines are enabled. For example, pedestrians will queue to go through a narrow portal. In a busy area, lanes of opposing pedestrian traffic will tend to form spontaneously after some time passes, since this pattern enables more pedestrians to make faster progress. These self-organizing patterns of movement, which the virtual pedestrians are not explicitly programmed to form, resemble those of real human crowds in urban environments.

![Flow of control and sequencing of the reactive behavior routines](image)

Figure 6. Flow of control and sequencing of the reactive behavior routines in the best permutation order “C-A-B-F-E-D”.

A remaining issue is how best to activate the aforementioned 6 reactive behavior routines. Since the situation encountered by a pedestrian is always some combination of the six key situations that are covered by the behavior routines, we have chosen to activate the routines in a sequential manner, as illustrated in Figure 6, giving each the chance to alter the currently active motor control.
command, comprising speed, turning angle, etc.\textsuperscript{1} For each routine, the input is the motor command issued by its predecessor, either a higher-level behavior module (possibly goal-directed navigation) or another reactive behavior routine. The sequential flow of control affords later routines the advantage of overwriting motor commands issued by earlier routines, but this may cause the pedestrian to ignore some aspect of the situation, resulting in a collision. The problem can be mitigated by finding a "best" permutation ordering for the six routines. We have run many extensive (>20 minute long in virtual time) simulations in the Penn Station environment with several different numbers of pedestrians (333, 666, and 1000), exhaustively evaluating the performance of all 720 possible permutation orderings. The best permutation of the reactive behavior routines (in the sense that it results in the fewest collisions) that we have found is C-A-B-F-E-D.

4.3.2 Navigational and Motivational Behaviors

While the reactive behaviors enable pedestrians to move around freely, almost always avoiding collisions, navigational and motivational behaviors enable them to go where they desire, which is crucial for pedestrians. As we must deal with online simulations of numerous pedestrians within large, complex environments, we are confronted with many navigational issues, including the realism of paths taken, and the speed and scale of path planning, and pedestrian flow control through and around bottlenecks. We have found it necessary to develop a number of novel navigational behavior routines to address these issues. These behaviors rely in turn on a set of conventional navigational behavior routines. The latter include moving forward, wandering about, turning in place, steering towards a target, proceeding towards a target, and arriving at a target (see [Reynolds 1999] for details).

In the Penn Station environment, stairways from the concourse to train platforms are less than 1.8 meters in width, which allows only two pedestrians to advance comfortably side by side. The two major portals connecting the main waiting room and the concourses are less than 2.8 meters wide, making it difficult for four pedestrians to pass simultaneously. In a space that is expected to accommodate the hustle and bustle of hundreds or even thousands of pedestrians, these narrow passageways can easily become bottlenecks that will cause queuing crowds to accumulate. The crowds will hinder opposing traffic, exacerbating the situation, and pedestrians may experience lengthy delays.

To our knowledge, available techniques cannot completely tackle the problem. The queuing behavior introduced by Reynolds is not well-suited to situations involving two-way traffic, because in narrow passageways oncoming pedestrians often cause one another to slow down or stop. In the cramped space, this quickly leads to blockage. Self-organization, as discussed by Helbing and Molnar [1995] and by us below, yields lanes of opposing traffic which increases throughput, but it takes time and space to manifest itself. A crowd will quickly grow beyond control in narrow passageways, well before self-organization can help. Global crowd control techniques, such as those proposed by Musse and Thalmann [2001], are useful for directing a given group of agents traveling around as a whole, but they lack the flexibility to handle highly dynamic groups of autonomous pedestrians.

\textsuperscript{1} Alternatively, it may be possible to obtain better results by activating the reactive behavior routines in parallel instead of sequentially, or by developing an alternative set of key routines.

---

Figure 7. Passageway navigation. (a) Two imaginary boundaries (dashed lines) and the safety fan. (b) Search for safe direction interval when blocked by oncoming people. (c) Spread out if no oncoming people are observed. (d) Typical flow of pedestrians in a passageway – big flows on the side with small unblocking streams intermingle in the middle.

We have determined that the application of two behavioral heuristics maximizes pedestrian flow in bottleneck situations. First, pedestrians inside a bottleneck should move with traffic while trying not to impede oncoming traffic. Second, all connecting passageways between two places should be used in balance. These two behaviors are detailed next:

**Passageway navigation:**

Inside a narrow passageway, real people demonstrate different behavior patterns under different conditions. If all people are traveling in the same direction, they will tend to spread out in order to see further ahead and maximize their pace. However, if people are in busy two-way traffic, they will compromise and quickly form opposing lanes of pedestrian traffic to maximize the throughput.

When inside a passageway, our pedestrians employ a similar behavior. First, to maintain a safe distance from the walls as a pedestrian navigates a passageway, two imaginary boundaries are computed parallel to the walls and offset from them by about half the pedestrian’s bounding box size (Figure 7(a)). Space between these boundaries is considered to be safe. Hence, restricting the travel direction of the pedestrian within a safety fan, as shown in the figure) guarantees that stays clear of the walls. Second, if a pedestrian detects that its current direction is blocked by oncoming pedestrians, it will search within the safety fan hoping to find a direction interval big enough to get through (Figure 7(b)). The search starts from the pedestrian’s current direction and continues clockwise. During the search, the pedestrian attempts to steer to the right of every blocking on-comer, testing whether there is enough room to get through. If the search is successful, the pedestrian will proceed in the safe direction found. Otherwise, it will slow down and proceed in the rightmost direction within the safety fan. This strategy allows non-blocking traffic to intermingle without resistance. However, in a manner that reflects the preference of real pedestrians in many countries, a virtual pedestrian will tend to squeeze to the right if it is impeding or impeded by oncoming traffic (Figure 7(d)). Finally, ongoing pedestrians apply Routine C described in the previous section behavior in order to maintain a safe separation between one another in the moving crowd (Figure 7(c)). In the event that no more oncoming traffic is observed, pedestrians decrease their crowding factor \( c_i \) to spread out, allowing faster walkers to overtake slower ones. Upon encountering oncoming traffic, they will increase their crowding factor, which will draw the crowd more tightly, making more space for oncoming pedestrians.

**Passageway selection:**

In most urban environments there exist several options for transiting from one location to another. Real people are usually motivated enough to weigh the options and choose a route that promises to maximize their pace. Their choice depends both on personal preferences and on the real-time situation in and around
those access facilities. Similarly, our pedestrians will assess the situation around stairways and other passageways, pick a preferred one based on proximity and density of pedestrians near it, and proceed toward it. As crowd density is always changing, however, our pedestrian may be motivated to modify its choices too frequently. Hence, our selection routine dictates that the pedestrian maintain its original choice unless a significantly more favorable traffic density condition develops in a different passageway. For the details, see Algorithm 2 in the Appendix. This behavior, although executed at the level of an individual, has a global effect, balancing the loads at different passageways.

When pedestrians employ the above passageway navigation and passageway selection behaviors, we are able to increase the number of pedestrians within the Penn Station model from under 400 to well over 1000 without any long-term blockage in bottlenecks.

Visually-guided navigation among static obstacles is an important behavior for pedestrians. As illustrated by Figure 8, the following behavioral routine accomplishes this task on a local scale:

**Perception-guided navigation among static obstacles:**
Given a path P (the global planning of paths will be explained in the next section), a “farthest visible point” p on P—i.e., the farthest point along P such that there is no obstacle on the line between p and the pedestrian’s current position—is determined and set as an intermediate target. As the pedestrian progresses toward p, it may detect a new farthest visible point that is even further along the path. This enables the pedestrian to approach the final target in a natural, incremental fashion. On each foot step during navigation, motor control commands issued by the navigation behavior routine are verified sequentially by the entire set of reactive behavior routines in their aforementioned order so as to keep the pedestrian safe from collisions.

**Detailed “arrival at target” navigation:**
Before a pedestrian arrives at a target, a detailed path will be needed if small obstacles intervene that are not resolved on ordinary path maps. Such paths can be found on a fine-scale grid path map. Unlike the previous situation, a pedestrian will follow the detailed path more strictly as it approaches the final target, because accuracy becomes an increasingly important factor in the realism of the navigation as the distance to the target diminishes. Additionally, reactive behaviors that are used to avoid stationary obstacles (Routines A and B) must now consider that some part of an obstacle may also be a part of the target or may be very close to the target; e.g., the front part of a vending machine, or the back of a bench in which the pedestrian wants to sit. Employing the routines indiscriminately will cause the pedestrian to avoid the obstacle as well as the target, and thereby hinder or even prevent the pedestrian from reaching the target. One possible solution is to remove the target portion of the obstacle from the perception map before applying the routines, but this becomes difficult to manage in an online simulation. We choose a simpler alternative—temporarily disable Routines A and B and accurately follow the detailed path, which will avoid obstacles. Note that the other reactive behaviors, Routines C, D, and E, remain active, as does Routine F, which will continue to play the important role of verifying that modified motor control commands never lead the pedestrian into obstacles.

**4.3.3 Other Interesting Behaviors**

The previously described behaviors comprise an essential portion of the pedestrian’s behavioral repertoire. To make our pedestrians more interesting, however, we have augmented the repertoire with a set of non-navigational behavior routines including, among others, the following:

- Select an unoccupied seat and sit down
- Approach a performance and watch
- Meet with friends and chat
- Queue at a vending machine and make a purchase
- Queue at ticketing areas and purchase a ticket

In this last behavior, for example, a pedestrian joins the queue and stands behind its precursor pedestrian until it comes to the head of the queue. Then, it will approach the first ticket counter associated with this queue that becomes available. Note that these non-navigational behaviors depend on the basic reactive behaviors and navigational behaviors to enable the pedestrian to reach its targets in a collision-free manner.

**4.3.4 Mental State and Action Selection**

Each pedestrian maintains a set of internal mental state variables, which encodes the pedestrian’s current physiological, psychological or social needs. These variables include tiredness, thirst, curiosity, the propensity to get attracted by performers, the need to acquire a ticket, etc. When the value of a mental state variable exceeds a specified threshold, an action selection mechanism chooses the appropriate behavior to fulfill the need. Once a need is fulfilled, the value of the associated internal state variable begins to decrease asymptotically to zero.

We classify pedestrians in the virtual train station environment as commuters, tourists, law enforcement officers, performers, etc. Each pedestrian type has an associated action selection mechanism with behavior-triggering thresholds associated with mental state variables set accordingly. For instance, law enforcement officers on guard will never attempt to buy a train ticket and commuters will never act like performers. As a representative example, Figure 9 illustrates the action selection mechanism of a commuter.
4.4 Cognitive Control

At the highest level of autonomous control, a cognitive model [Funge et al., 1999] is responsible for creating and executing plans suitable for autonomous pedestrians. Whereas the behavioral substrate described in Section 4.3 is reactive, the cognitive modeling layer makes the pedestrian a deliberative autonomous agent.

The realism of pedestrian animation depends on the execution of rational actions at different levels of abstraction and at different spatio-temporal scales. At a high level of abstraction and over a large spatio-temporal scale, a pedestrian must be able to make reasonable global navigation plans that can enable it to travel deliberatively (and with suitable perseverance) between widely separated regions of the station, stay from the end of the arcade through the waiting room and concourses to a specific train platform. Such plans should exhibit desirable properties, such as saving time. During the actual navigation, however, the pedestrian must have the freedom to decide whether or not, and to what extent, to follow the plan, depending on the real-time situation. Priority must be given to the local (in both space and time) situation in the dynamic environment in order to keep the pedestrian safe while still permitting progress towards a long-range goal. In our autonomous pedestrian model, we couple the global plans and local reactive behaviors to make navigation as natural as possible.

4.4.1 Coupling Cognitive Control to Reactive Behavior

The integration of planning and behavior control at different levels increases the realism of our pedestrian model. From a global point of view, it provides pedestrians with rational long-term plans. However, it also allows pedestrians to move in a natural, environmentally-adaptive manner while carrying out the plan.

The pedestrian maintains a stack of goals with the top one being the current goal. If the current goal is beyond the scope of the behavioral controller, it is handled by the cognitive controller. The stack is updated by both the cognitive controller and the behavioral controller. The behavioral controller can insert directives according to the internal mental state and environmental situation (say, thirsty & vending machine nearby, then push “plan to get a drink”). The behavioral control mechanism will retrieve goals from the stack, decompose them into intermediate ones and initiate behaviors appropriate to the satisfaction of these goals. For example, a pedestrian whose tiredness variable exceeds a threshold will look for seats, select one that is unoccupied, plan a path to it, approach and sit down to take a rest. Once achieved, goals are removed from the stack.

Intuitively, the goal stack remembers "what needs doing", the cognitive controller decides "what to do", the mental state variables dictate "why it should be done", the behavior controller attempts to "get it done".

4.4.2 Global Path Planning

Path planning has been an active research subject in robotics and it has also been applied to interactive games. Different applications put emphasis on different properties of planned paths and thus choose different techniques to attack the problem. Our pedestrians employ a method that is well-suited to the virtual urban environment representation. We detail our path search algorithms, which are based on the well-known A* graph search algorithm, elsewhere [Anonymous 2005]. Generally speaking, our algorithms are very efficient, but they provide rough paths—i.e., paths that are either jagged (grid path maps) or containing many spikes (quadtree path maps)—as opposed to smooth, spline-like paths. Consequently, a pedestrian uses those rough paths only as a navigational guide and retains the freedom to locomote locally as natural a manner as possible. Despite the complexity of situations with which a virtual pedestrian must deal, the important cognitive ability of planning global paths to targets can be readily implemented on top of the behavioral substrate of Section 4.3:

Plan global path to target:

Global planning directs a pedestrian to proceed through intermediate areas and finally reach the ultimate destination. Global planning exploits the topological map at the top level of the environment model (Figure 3). As was discussed in Section 3, the topological map represents the connectivity between different regions of the train station environment (the main waiting room, arcade, upper and lower concourses, train platforms, ticketing areas, etc.) in the form of a graph. By applying the search algorithms detailed in [Anonymous 2005] within the path maps associated with each region, a path is planned from the current location to the boundary or portal between the current region and the next region on the global path. The process is repeated in the next region, and so on, until it can take place in the target region to terminate at the target location. Although the extent of the path is global, the processing is primarily local. Depending on the situation on the way, a pedestrian acts differently, either going directly, or selecting a passage, or planning a path, as is detailed by Algorithm 3 in the Appendix and more detail in [Anonymous 2005].

5 Results

Our pedestrian animation system, which consists of about 150,000 lines of C++ code, enables us to run long-term simulations of pedestrians in a large-scale urban environment, specifically the Penn Station environment, without manual intervention. Our Penn Station model, which can be bounded by a box of 200(l) x 150(w) x 20(h) cubic meters, is huge and complex with hundreds of architectural and non-architectural objects. The raw textured geometric model is available in OpenFlight file format (we obtained it from Boston Dynamics, Inc.) and it includes floors walls and ceilings, stairs, doorways, columns, ticket booths, platforms, tracks, and lamps. We augmented it with fountains, benches, vending machines and tables, signs, etc., and even added some litter on the floor. We manually divided the entire 3D space into 43 regions.

At run time, our hierarchical world model requires approximately 90MB of memory to accommodate the station and all the objects.

In our simulation experiments, populate the virtual station with five different types of characters. Most of them are classified as commuters. In addition, there are tourists, performers, policemen, and patrolling soldiers. With everyone guided by his/her own autonomous control, they act out their volition, coordinate with other strangers based on both observation and personal preferences, and at the same time follow simple common-sense rules. These autonomous characters imbue the virtual train station with liveliness, social (dis)order, and a realistically complex dynamic.

5.1 Performance

Figure 10 shows how the computational load increases with the number of pedestrians in the station. These tests are run on an Intel Xeon 2.8GHz machine with 1GB memory. The total length of every test is 20 minutes in virtual world time. The time we measured here is the time spent only on computation of our algorithms—environment model update, perceptual query, motor control, behavioral control, and cognitive control. Rendering time and DI-Guy-specific character motion control are not included. The
The small quadratic term is probably due to the fact that as the number of pedestrians increases, the number of proximal pedestrians also increases, but with a much smaller factor.

Figure 11 analyzes the computation time for various parts of the simulation based on multiple experiments with different numbers of pedestrians ranging from 100 to 1000 on the aforementioned PC. Again, the time for rendering and DI-Guy-specific character motion control are excluded.

The exit to the station has three separated stairways and pedestrians automatically choose to take different stairways based on the perceived crowd density and distance. Each of the stairways is narrow, allowing only four people to walk side by side. A pedestrian steers to the right when in a narrow passageway that is busy with two-way traffic, such as the stairways. Later in the animation, such behavior can be observed more clearly on the even narrower stairways leading down to the train platforms.

The scenario in the second simulation in the arcade is similar to the first example, but this time the pedestrians perceive the arcade as a corridor that can conveniently accommodate only two major opposing traffic flows. Once oncoming people are observed, a pedestrian will prefer to head to the right. As a result, the crowds squeeze past each other to their respective right. As soon as the oncoming crowd is cleared, the pedestrians tend to spread out again in order to maximize their view and pace.

Figure 12. Plan view of Penn Station, revealing the concourses (left), the main waiting room (center), and the arcade (right).
virtual cameras generate synthetic video feeds that emulate those generated by real surveillance cameras monitoring public spaces (Figure 13). Unlike the real world, however, we can readily reconfigure the multi-camera layout in the virtual station environment. A second advantage is that we do not violate the privacy of any real people in our virtual world. A third advantage is that in the virtual world we can potentially experiment with complex and/or dangerous scenarios that cannot be attempted safely or repeatedly in real life. For example, we can potentially simulate the reactions of crowds in crisis situations, such as the presence of an armed terrorist, and see if we can design suites of computer vision algorithms that, through an automated analysis of the virtual video streams, can detect such situations and issue appropriate alerts. Possibilities abound, but a complete description of our virtual vision research is well beyond the scope of this paper.

7 Conclusion and Future Work

We have developed a sophisticated, large-scale human animation system that combines behavior, perception, and cognitive simulation algorithms for autonomous pedestrians. Coupled to our hierarchical environmental modeling framework, our novel system efficiently synthesizes numerous self-animating pedestrians performing a rich variety of activities in a large-scale indoor urban environment.

Our animation results speak to the robustness of our system and its ability to automatically generate large quantities of intricate animation of pedestrians autonomously carrying out various activities. For now, the most serious shortcomings—those which yield the most annoying artifacts in our animations—are attributable to the DI-Guy low-level, real-time human animation API. First, despite the steps that we have taken to improve the motion transitions in the DI-Giy characters, limitations remain in how quickly they can speed up, slow down, change gaits, and make tight turns, so their bodies can sometimes find it difficult to react to the fast dynamics of a complex, hectic environment that produces frequent close encounters between pedestrians. To make matters worse, some DI-Guy characters are rather limited in terms of their motion repertoires and, although the best of them can do plenty, they cannot synthesize the range of motions/actions that we would like to have. On the bright side, DI-Guy and competing human animation APIs are continually improving.

In future work, we plan systematically to expand the behavioral and cognitive repertoires of our autonomous virtual pedestrians to further close the gap between their abilities and those of real people. We would also like to extend their immediate perception beyond the current range of about 5 meters. Given the quality of the behaviors and actions that we are now seeing, the lack of head movement and manipulation skills in our virtual pedestrians has begun to become objectionable. It is our intention to develop a satisfactory set of reactive and deliberative head motion behaviors for our virtual pedestrian model and to imbue it with useful manipulation skills. For instance, it hasn’t escaped our notice that our train station simulations would be more realistic if some virtual pedestrians toted luggage.

Acknowledgements

The research reported herein was supported in part by grants from the Defense Advanced Research Projects Agency (DARPA) of the Department of Defense and from the National Science Foundation (NSF). We thank Dr. Tom Strat of DARPA for his generous support and encouragement. We also thank <name> for his invaluable contributions to the implementation and visualization of the Penn Station environmental model and its integration with the DI-Guy software API.

References

ANONYMOUS. 2001. Environmental modeling for autonomous virtual pedestrians. Submitted to the 2005 SAE Symposium on Digital Human Modeling for Design and Engineering. In review. Note to reviewers: A copy of this manuscript has been provided in this submission as supplementary material.


Appendix: Algorithms

if $T$ and $T_i$ intersect then
  let $p$ be the anticipated point of intersection
  let $t$ be $H$'s travel time to $p$
  let $t_i$ be $C_i$'s travel time to $p$
  safe-$dt = (B(H) + B(C_i)) / \min(|v|, |v_i|)$
  if $|t - t_i| < safe-$dt$ then
    return "$C_i$ is a cross-collision threat"
if $T$ and $T_i$ almost parallel then
  if the lateral separation $< B(H) + B(C_i)$ then
    return "$C_i$ is a face-to-face-collision threat"
else return "$C_i$ is not a collision threat"

Algorithm 1. Avoid oncoming pedestrians.

Given a list of candidate passageways $P_i$, together with the density $D_i$ (unit: $\text{#pedestrians/m}^2$) and proximity $T_i$ (unit: meter) of the entrance of $P_i$ closer to the pedestrian:

1. Let $S_i = T_i + 25D_i$, a value that reflects $P_i$'s situation. // Smaller $S_i$ indicates a better situation.
2. Let $m$ be the index of the passageway that has the smallest $S_i$
3. Let $c = m$ if none is picked previously
4. If ($P_c$ is NULL) then
   pick $P_m$ // no previous choice, so pick the best
else if ($S_m < S_c - 5.0$) then
   pick $P_m$ // a significantly better one found, so pick it
else
   pick $P_c$ // stick to the previous choice

Algorithm 2. Passageway selection.

Direct agent $H$ to go to target $T$
1. Let $R_T$ be the region where $T$ is in
2. Let $R_H$ be the region where $H$ is in
3. If ($R_T \neq R_H$) then
   a. Use the "path-to-via" information to find $R_T$'s neighbor regions $R_i$ that lead to $R_T$
   b. If all $R_i$'s are reachable only from passageways then
      initiate behaviors "passageway selection" and "passageway navigation"
else
   pick any of the $R_i$'s that are reachable not from a passageway, and let $T$ be the boundary between $R_0$ and that picked $R_i$
4. Now that $T$ must be in $R_0$, test whether there is any obstacle between $T$ and $H$
5. If obstacle exists between $T$ and $H$ then
   If $T$ and $H$ is close then
      use behavior "arrival"
else
      use behavior "path navigation"
else
   go directly toward $T$
6. Use necessary reactive behaviors routines to verify the motor control issued by navigation behavior routines

Algorithm 3. Go to a target. Refer to [Anonymous 2005].