

Autonomous Pedestrians

Wei Shao and Demetri Terzopoulos

New York University, Courant Institute, Media Research Lab[†]

Abstract

*We address the difficult open problem of emulating the rich complexity of real pedestrians in urban environments. Our artificial life approach integrates motor, perceptual, behavioral, and cognitive components within a model of pedestrians as **individuals**. Our comprehensive model features innovations in these components, as well as in their combination, yielding results of unprecedented fidelity and complexity for fully autonomous multi-human simulation in a large urban environment. We represent the environment using hierarchical data structures, which efficiently support the perceptual queries of the autonomous pedestrians that drive their behavioral responses and sustain their ability to plan their actions on local and global scales.*

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation; I.6.8 [Simulation and Modeling]: Types of Simulation—Animation

1. Introduction

“Forty years ago today at 9 a.m., in a light rain, jack-hammers 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 (Fig. 2), which had opened to the public in 1910, in order to make way for the Penn Plaza complex and Madison Square Garden, was “a tragic loss of architectural grandeur”. Although state-of-the-art computer graphics enables a virtual reconstruction of the train station with impressive geometric and photometric detail, it does not yet enable the automated animation of the station’s human occupants with anywhere near as much fidelity. Our research addresses this difficult, long-term challenge.

In a departure from the substantial literature on so-called “crowd simulation”, we develop a decentralized, comprehensive model of pedestrians as autonomous *individuals* capable of a broad variety of activities in large-scale synthetic

urban spaces. Our artificial life approach to modeling humans spans the modeling of pedestrian appearance, locomotion, perception, behavior, and cognition. We deploy a multitude of self-animated virtual pedestrians within a large environment model, a VR reconstruction of the original Penn Station (Fig. 1). The environment model includes hierarchical data structures that support the efficient interaction between numerous pedestrians and their complex virtual world through fast (perceptual) query algorithms and support pedestrian navigation on local and global scales.

We continue with a review of related work in Section 2. Section 3 briefly reviews our virtual environment model. In Section 4, we present our autonomous pedestrian model, mostly focusing on its (reactive) behavioral and (deliberative) cognitive components. Section 5 presents results comprising long term simulations with well over 1000 pedestrians and reports on performance. Finally, Section 6 concludes the paper and discusses future work.

2. Related Work

Human animation is an important and challenging problem in computer graphics [BPW93]. 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 attention of CG researchers [ALA*01, MT01]. The topic has also been of some interest

[†] www.mrl.nyu.edu/~{weishao,dt}



Figure 1: A large-scale simulation of a virtual train station populated by self-animated virtual humans. From left to right are rendered images of the main waiting room, concourses, and arcade.



Figure 2: Original Pennsylvania Station in New York City.

in the field of artificial life [BA00], as well as in architecture and urban planning [Lov93, SE01] where graphics researchers have assisted in visualizing planned construction projects, including pedestrian animation [FMS*00, MH03].

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). The seminal work of [Rey87] on behavioral animation is certainly relevant to our effort, as is its further development in work by other researchers [TT94, TDB*02, LRBW04]. Behavioral animation 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. Low-level crowd interaction models have been developed in the sciences [GM85, HM95, BJTG98, Sch02] and by animation researchers [GKM*01, LMM03, SGC04, UdHCT04, LD04] and also in the movie industry by Disney and many other production houses, most notably in recent years for horde battle scenes in feature films (see www.massivesoftware.com).

While our work is innovative in the context of behavioral animation, it is very different from so-called “crowd animation”. As the aforementioned literature shows, animating large crowds, where one character algorithmically follows another in a stolid manner, is relatively easy. We are uninterested in crowds *per se*. Rather, the goal of our work is to develop a comprehensive, self-animated model of *individual* human beings that incorporates nontrivial human-like abili-

ties suited to the purposes of animating virtual pedestrians in urban environments. Our approach is inspired most heavily by the work of [TT94] on artificial animals and by [FTT99] on cognitive modeling for intelligent characters that can reason and plan their actions. We further develop their comprehensive artificial life approach and adopt it for the first time to the case of an autonomous virtual human model that can populate large-scale urban spaces. In particular, we pay serious attention to deliberative human activities over and above the reactive behavior level.

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. Recently, [LD04] proposed a suitable structuring of the geometric environment together with reactive navigation algorithms for pedestrian simulation. While this part of our work is conceptually similar, our methods differ. We have devoted considerable effort to developing a large-scale (indoor) urban environment model, which is described in detail elsewhere [ST05], and which we summarize next.

We represent the virtual environment by a hierarchical collection of maps. As illustrated in Fig. 3, the model comprises (i) a topological map which represents the topological structure between different parts of the virtual world. Linked within this map are (ii) perception maps, which provide rel-

event information to perceptual queries, and (iii) path maps, which enable online path-planning for navigation.

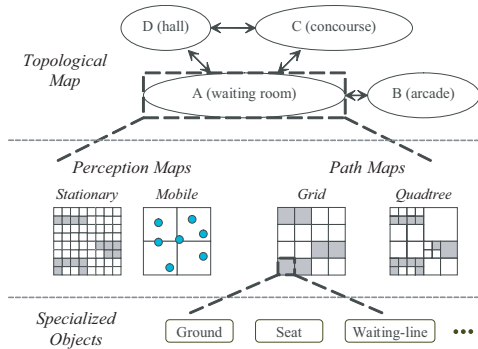


Figure 3: Hierarchical environment model.

In the topological map, nodes correspond to environmental regions and edges 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 (e.g., ground, walls, benches). The representation assumes that the walkable surface in a region may be mapped onto a horizontal plane without loss of essential geometric information. Consequently, the 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 keeps track of mobile objects, usually other pedestrians. 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 $0.2 \sim 0.3$ 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 perception process will be discussed in more detail in Section 4.2.

The path maps 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 of the area covered 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 (i.e., how far 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 (cf. [BMS04]). Our simulations with numerous pedestrians indicate that the quadtree map is used for planning about 94% of their paths. The remaining 6% of the paths are planned using the grid path 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.

Our environment model is efficient enough to support the real-time (30fps) simulation 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 [ST05].

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 architected as a hierarchical artificial life model. Progressing through levels of abstraction, our model incorporates appearance, motor, perception, behavior, and cognition sub-models. The following sections discuss each of these components in turn.

4.1. Human Appearance, Movement, & Motor Control

As an implementation of the low-level appearance and motor levels, we employ a human animation software package called *DI-Guy*, which is commercially available from Boston Dynamics Inc. It provides textured human characters with basic motor skills, such as standing, strolling, walking, running, sitting, etc. [KCR98]. *DI-Guy* characters are by no means autonomous, but their actions may be scripted *manually* using an interactive tool called *DI-Guy Scenario*, which we do *not* use. *DI-Guy* also includes an SDK that allows external C/C++ programs to control a character's basic motor repertoire. This SDK enables us to interface *DI-Guy* to our extensive, high-level perceptual, behavioral, and cognitive control software, which will be described in subsequent sections, thereby achieving fully autonomous pedestrians.

Emulating the natural appearance and movement of human beings is a difficult problem and, not surprisingly, *DI-Guy* suffers from several limitations. The character appearance models are insufficiently detailed. More importantly, *DI-Guy* characters cannot synthesize the full range of motions needed to cope with a highly dynamic urban environment. With the help of the *DI-Guy Motion Editor*, we have modified and supplemented the motion repertoire, enabling faster transitions, which in turn enables our pedestrians to deal with busy urban environments.

Moreover, we have implemented a motor control interface between the kinematic layer of *DI-Guy*, and our behavioral

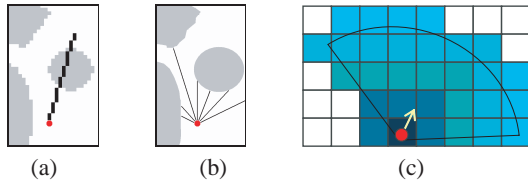


Figure 4: Perception. (a-b) Sensing stationary objects, (a) by examining map cells along the rasterized eye ray, while (b) perceiving the broader situation by shooting out a fan of eye rays (rasterization not shown). (c) Sensing mobile objects by examining (color-coded) tiers of the sensing fan.

controllers. The interface accepts motor control commands from behavior modules, and it verifies and corrects them in accordance with the pedestrian’s kinematic limits. It then selects an appropriate motion sequence or posture and calls upon the kinematic layer to apply the update to the character. Our seamless interface hides the details of the underlying kinematic layer from our higher-level behavior routines, enabling the latter to be developed more or less independently. Hence, in principle, any suitable low-level human animation API can easily replace DI-Guy in our future work.

4.2. Perception

An autonomous and highly mobile virtual human must have a perceptive regard of its environment. Our environment model (Section 3) efficiently provides accurate perceptual data in response to the queries of autonomous pedestrians.

Sensing ground height. In the static object perception map, each map cell contains the height functions of usually a single and sometimes a few ground objects, such as the floor, stairs, etc. The highest object at the desired foot location of a pedestrian is returned in constant time and it is processed within the pedestrian’s motor layer, which plants the foot at the appropriate height.

Sensing static objects. The visual sensing computation shoots out a fan of line segments, with length determining the desired perceptual range and density determining the desired perceptual acuity (Fig. 4 (a)-(b)). Grid cells on the perception map 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.

Sensing mobile objects. To sense mobile objects (mostly other humans), a pedestrian must first identify nearby pedestrians within the sensing range. The range here is defined by a fan as illustrated in (Fig. 4(c)). On the mobile object perception map, the cells wholly or partly within the fan are divided into “tiers” based on their distance to the pedestrian. Closer tiers are examined earlier. Once a predefined number (currently set to 16) of nearby pedestrians are perceived, the sensing is terminated. This is motivated by the fact that, at

any given time, people usually pay attention only to a limited number of other people, usually those that are most proximal. Once the set of nearby pedestrians is sensed, further information can be obtained by referring to finer maps, estimation, or simply querying some pedestrian of particular interest. Given the sensing fan and the maximum number of sensed pedestrians, sensing is a constant time operation.

4.3. Behavioral Control

Realistic behavioral modeling, whose purpose is to link perception to appropriate actions in an autonomous virtual human, is a big challenge. 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 [TT94], 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. We have developed six key reactive behavior routines, each suitable for a different set of situations in a densely populated and highly dynamic environment (Fig. 5). Given that a pedestrian possesses a set of motor skills, such as standing still, moving forward, turning in several directions, speeding up and slowing down, etc., these routines are responsible for initiating, terminating, and sequencing the motor skills on a short-term basis guided by sensory stimuli and internal percepts. The details of the six routines, denoted Routines A–F, are provided in the Appendix.

Several remarks regarding the routines are in order: Obviously, the fail-safe strategy of Routine E suffices in and of itself 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 a short time.

A remaining issue is how best to activate the six reactive behavior routines. Since the situation encountered by a pedestrian is always some combination of the six key situations that are covered by the six routines, we have chosen to activate them in a sequential manner (Fig. 6), giving each the chance to alter the currently active motor control command, comprising speed, turning angle, etc. 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

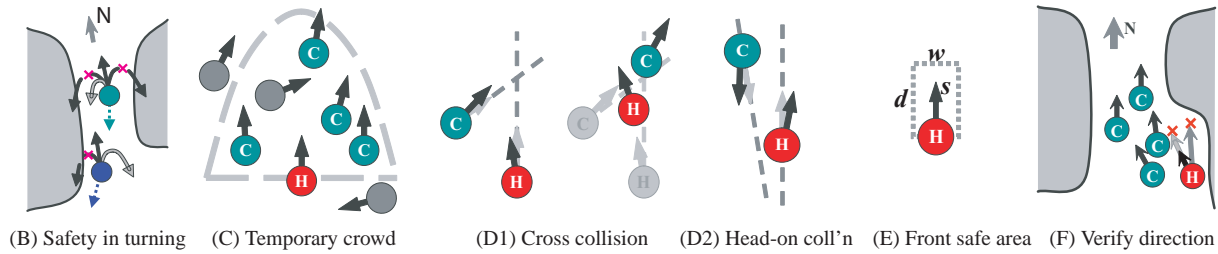


Figure 5: Reactive behaviors. (B) Pedestrians choose best turning curves (light gray) to turn southward. (C) Pedestrians within H 's front parabola traveling in similar directions as H (labeled C) are in H 's temporary crowd. (D1) To avoid cross collision, (left) H slows down and turns toward C while C does the opposite until collision is cleared (right). (D2) To avoid head-on collision, both pedestrians turn slightly away from each other. (E) The dotted rectangle defines H 's front safe area; w and d depend on H 's bounding box size and d is determined by H 's current speed s . (F) Confronted by static and dynamic threats, H picks obstacle-free direction (light gray arrow) and slows down (black arrow) to let others pass before proceeding.

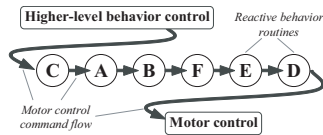


Figure 6: Sequencing of the reactive behavior routines in the best permutation order "C-A-B-F-E-D".

overriding 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 simulations (longer than 20 minutes in virtual time) in the Penn Station environment with different numbers of pedestrians (333, 666, and 1000), exhaustively evaluating the performance of all 720 possible permutation orderings. The best permutation of the six routines, in the sense that it results in the fewest collisions while reasonable progress is still maintained in navigation, 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. A pioneering effort on autonomous navigation is that by Noser et al. [NRTMT95]. Metoyer and Hodgins [MH03] propose a 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.

As we must deal with online simulations of numerous pedestrians within large, complex environments, we are confronted with many navigational issues, such as the realism of paths taken, 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, including moving forward, turning (in place or while moving), proceeding toward a target, and arriving at a target (see [Rey99] for details).

In the Penn Station environment, large regions are connected by narrow portals and stairways, some of which allow only two or three people to advance comfortably side by side. These bottlenecks can easily cause extensive queuing, leading to lengthy delays. In our experience, available techniques, such as queuing in [Rey99], self-organization in [HM95], and global crowd control in [MT01] cannot tackle the problem, as it involves highly dynamic two way traffic and requires quick and flexible responses from pedestrians. In our solution, we employ two behavioral heuristics. First, pedestrians inside a bottleneck should move with traffic while trying not to impede oncoming pedestrians. Second, all connecting passageways between two places should be used in balance. The two behaviors are detailed next.

Passageway navigation. In real life, if all pedestrians are traveling in the same direction inside a narrow passageway, they will tend to spread out in order to see further ahead and maximize their pace. However, once oncoming traffic is encountered, people will tend to form opposing lanes to maximize the two-way throughput. Our virtual pedestrians incorporate a similar behavior. First, two imaginary boundaries are computed parallel to the walls with an offset of about half the pedestrian H 's bounding box size (Fig. 7(a)). Restricting H 's travel direction within a safety fan defined by the boundaries, as shown in the figure, guarantees that H stays clear of the walls. Second, if H detects that its current direction is blocked by oncoming pedestrians, it will search within the safety fan for a safe interval to get through (Fig. 7(b)). The search starts from H 's current direction and continues clockwise. If the search succeeds, H will move in the safe direction found. Otherwise, H will slow down and proceed in the rightmost direction within the safety fan. This

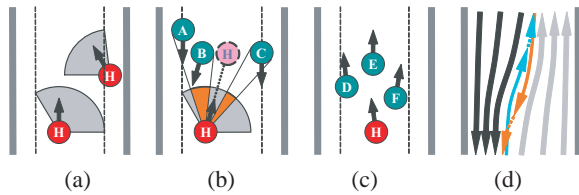


Figure 7: Passageway navigation. (a) Two imaginary boundaries (dashed lines) and the safety fan. (b) Pedestrian H searches for a safe direction interval when confronted by oncoming traffic. (c) Spread out when no oncoming traffic is observed. (d) Typical flow of pedestrians in a passageway—big flows on the sides with small unblocking streams intermingling in the middle.

strategy allows non-blocking traffic to intermingle without resistance. However, in a manner that reflects the preference of real people in many countries, a virtual pedestrian will tend to squeeze to the right if it is impeding or impeded by oncoming traffic (Fig. 7(d)). Finally, Routine C (see the Appendix) is used to maintain a safe separation between oncoming pedestrians. By altering their crowding factor w_i based on the observation of oncoming traffic, pedestrians can spread out or draw tightly to adapt to the situation (Fig. 7(c)).

Passageway selection. People are usually motivated enough to pick the best option from several available access routes, depending on both personal preferences and the real-time situation in and around those routes. Likewise, our pedestrians will assess the situation around stairways and portals, pick a preferred one based on proximity and density of pedestrians near it, and proceed toward it. They will persist in the choice they make, unless a significantly more favorable condition is detected elsewhere. This behavior, although executed independently by each individual, has a global effect of balancing the loads at different passageways.

With the above two passageway 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 another important behavior for pedestrians. The following two behavioral routines accomplish 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 H 's current position—is determined and set as an intermediate target (Fig. 8). As H 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. During navigation, motor control commands for each footstep are verified sequentially by the en-

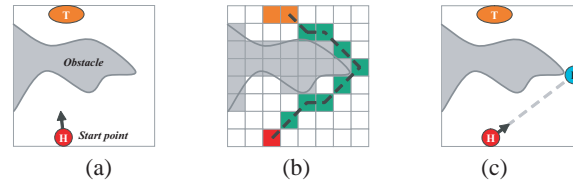


Figure 8: Perception-guided navigation. (a) To reach target T , (b) pedestrian H will plan a jagged path on a path map (either grid or quadtree), (c) pick the farthest visible point (blue circle marked F) along the path and proceed toward it.

tire 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. Such paths can be found on a fine-scale grid path map. The pedestrian will follow the detailed path strictly as it approaches the target, because accuracy becomes increasingly important in the realism of the navigation as the distance to the target diminishes. As some part of an obstacle may also be a part of the target or be very close by, indiscriminately employing reactive behaviors for static obstacle avoidance—Routines A and B (refer to the Appendix)—will cause the pedestrian to avoid the obstacle as well as the target, thereby hindering or even preventing the pedestrian from reaching the target. We deal with this by temporarily disabling the two routines and letting the pedestrian accurately follow the detailed path, which already avoids 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 aspect 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 the 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, the pedestrian will approach the first ticket counter associated with this queue that becomes available. Space limitations preclude the detailed specification of the above behaviors in this paper. Note, however, that these non-navigational behaviors depend on the basic reactive behaviors and navigational behaviors to enable the pedestrian to reach targets in a collision-free manner.

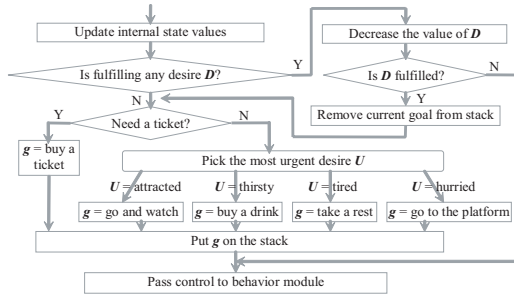


Figure 9: Action selection in a pedestrian commuter.

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 be attracted by performances, 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 appropriately set behavior-triggering thresholds associated with mental state variables. 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, Fig. 9 illustrates the action selection mechanism of a commuter.

4.4. Cognitive Control

At the highest level of autonomous control, a cognitive model [FTT99] is responsible for creating and executing plans, as is necessary for a deliberative human agent. Such a model must be able to make reasonable global navigation plans in order for a pedestrian to travel purposefully and with suitable perseverance between widely separated regions of the environment. 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, as we discussed when explaining the behaviors in Section 4.3.2. On the other hand, in a highly dynamic environment such as a train station, the pedestrian also needs the ability to decide whether and when a new plan is needed. These decisions require a proper coupling between the behavioral layer and cognitive layer. Before we describe the coupling mechanism, we will explain the global path planning strategy.

Global path planning directs a pedestrian to proceed

through intermediate regions and finally reach the ultimate destination. To do this, it exploits the topological map at the top level of the environment model (Fig. 3). Given a pedestrian's current location and a target region, this map provides a set of optimal neighboring regions where the pedestrian can go. By applying path search algorithms within the path maps associated with each region, the pedestrian can plan a path from the current location to the boundary or portal between the current region and the next. The process is repeated in the next region, and so on, until it terminates at the target location. In this way, although the extent of the path is global, the processing is primarily local. Our path search algorithms (detailed in [ST05]), which are based on the well-known A* graph search algorithm, 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 plans only as a navigational guide and retains the freedom to locomote locally in as natural a manner as possible, as was described in Section 4.3.2.

Coupling cognitive control to behavioral control increases the realism of our pedestrians. To this end, every pedestrian maintains a stack of goals, the top one being the current goal. The goal stack is accessible both to the deliberative, cognitive layer and to the underlying reactive, behavioral layer. If a goal is beyond the scope of the behavioral controller (for example, some task that needs path planning), it will be further decomposed into subgoals, allowing the cognitive controller to handle those subgoals within its ability (such as planning a path) and the behavioral controller to handle the others by initiating appropriate behavior modules (such as navigation on a local scale). The behavioral controller can also insert directives according to the internal mental state and environmental situation (e.g., if thirsty & vending machine nearby, then push "plan to get a drink"). This usually interrupts the execution of the current task and typically invalidates it. When it is time for the interrupted task to resume, a new plan is often needed. Intuitively, the goal stack remembers "what needs doing", the mental state variables dictate "why it should be done", the cognitive controller decides "how to do it" at a higher, abstract level, and the behavior controller determines "how to do it" at a lower, concrete level and ultimately attempts to "get it done".

5. Results

Our pedestrian animation system, which comprises about 50,000 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. The entire 3D space of the Penn Station ($200(l) \times 150(w) \times 20(h)m^3$), which contains hundreds of architectural and non-architectural objects, is manually divided into 43 regions. At run time, our environment model requires approximately 90MB of memory to accommodate the station and all of its associated objects.

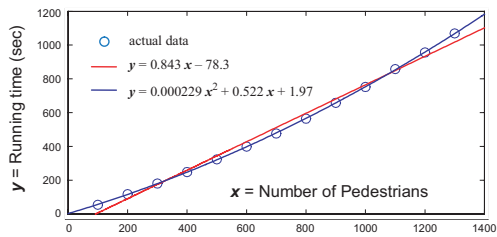


Figure 10: Pure simulation time vs. number of pedestrians.

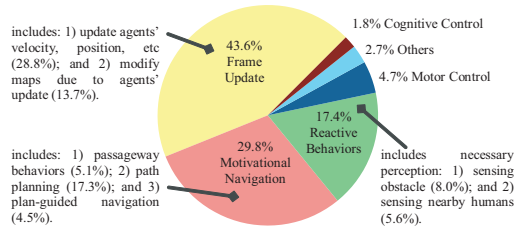


Figure 11: Computational loads of the system's parts.

In our simulation experiments, we populate the virtual station with five different types of pedestrians: commuters, tourists, performers, policemen, and patrolling soldiers. With every individual guided by his/her own autonomous control, these autonomous pedestrians imbue the virtual train station with liveliness, social (dis)order, and a realistically complex dynamic.

5.1. Performance

We have run various simulation tests on a 2.8GHz Intel Xeon system with 1GB of main memory. The total length of each test is 20 minutes in virtual world time. Fig. 10 indicates the computational load increase with the number of pedestrians in the simulation. The simulation times reported include only the requirements of our algorithms—environment model update and motor control, perceptual query, behavioral control, and cognitive control for each pedestrian. The figure shows that real-time simulation can be achieved for as many as 1400 autonomous pedestrians (i.e., 20 virtual world minutes takes 20 minutes to simulate at 30fps). Although the relation is best fit by a quadratic function, the linear term dominates by a factor of 2200. The small quadratic term is likely due to the fact that the number of proximal pedestrians increases as the total number of pedestrians increases, but with a much smaller factor. Fig. 11 breaks down the computational load for various parts of the simulation based on experiments with different numbers of pedestrians ranging from 100 to 1000 on the aforementioned PC. Fig. 12 tabulates the frame rates that our system achieves on the aforementioned PC with an NVIDIA GeForce 6800 GT AGP8X 256MB graphics system. Due to the geometric complexity of the Penn Station

# of Pedestrians	0	100	200	300	400	500
Simulation only	n/a	64.4	32.2	23.0	16.9	12.3
Rendering only	21.0	12.5	9.2	7.6	6.0	5.4
Simulation+Rendering	21.0	10.5	7.2	5.7	4.4	3.8

Figure 12: Frame rate (in frames/sec) for pedestrian simulation only (including DI-Guy), rendering only (i.e., static pedestrians), and both simulation and rendering, with different numbers of pedestrians.

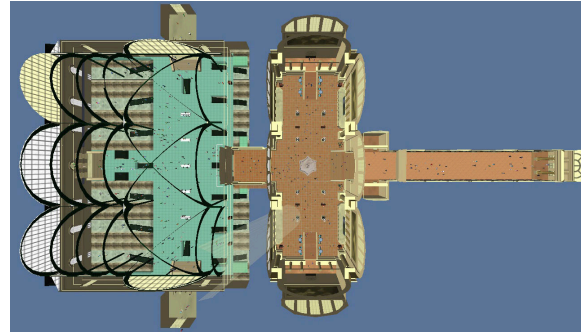


Figure 13: Plan view of the Penn Station model with the roof not rendered, revealing the 2-level concourses and the train tracks (left), the main waiting room (center), and the long shopping arcade (right).

model and numerous pedestrians, rendering times dominate pedestrian simulation times.

5.2. Animation Examples

We will now describe several representative simulations that demonstrate specific functionalities. To help place the animation scenarios in context, Fig. 13 shows a plan view of the Penn station model.

Following an Individual Commuter. As we claimed in the introduction, an important distinction between our system and existing crowd simulation systems is that we have implemented a comprehensive human model, which makes every pedestrian a complete *individual* with a richly broad behavioral and cognitive repertoire. We can therefore choose a commuter and, in a typical animation, follow our subject as he enters the station, proceeds to the ticket booths in the main waiting room, and waits in a queue to purchase a ticket at the first open booth. Having obtained a ticket, he then proceeds to the concourses through a congested portal. Next, our subject feels thirsty and spots a vending machine in the concourse. He walks toward it and waits his turn to get a drink. Feeling a bit tired, our subject finds a bench with an available seat, proceeds towards it, and sits down. Later, the clock chimes the hour and it is time for our subject to get up and proceed to his train platform. He makes his way

through a somewhat congested area by following, turning, and stopping as necessary in order to avoid bumping into other pedestrians. He passes by some dancers that are attracting interest from many other pedestrians, but our subject has no time to watch the performance and descends the stairs to his train platform.

Pedestrian Activity in the Train Station. A routine simulation, which includes over 600 autonomous pedestrians, demonstrates a variety of pedestrian activities that are typical for a train station. We can interactively vary our viewpoint through the station, directing the virtual camera on the main waiting room, concourse, and arcade areas in order to observe the rich variety of pedestrian activities that are simultaneously taking place in different parts of the station (Fig. 1). Some additional activities that were not mentioned above include pedestrians choosing portals and navigating through them, congregating in the upper concourse to watch a dance performance for amusement, and proceeding to the train platforms using the rather narrow staircases.

6. Conclusion and Future Work

We have developed a sophisticated human animation system whose major contribution is a comprehensive model of pedestrians as highly-capable individuals that combines perceptual, behavioral, and cognitive control components. Incorporating a hierarchical environmental modeling framework, our novel system efficiently synthesizes numerous self-animated pedestrians performing a rich variety of activities in a large-scale indoor urban environment.

Our results speak to the robustness of our system and its ability to produce prodigious quantities of intricate animation of pedestrians carrying out various individual and group activities. Although motion artifacts are at times conspicuous in our animation results due to the limitations of the low-level DI-Guy software, our design facilitates the potential replacement of this software by a better character rendering and motion synthesis package should one become available.

In future work, we plan systematically to expand the behavioral and cognitive repertoires of our autonomous virtual pedestrians to further narrow the gap between their abilities and those of real people. It is also our intention to develop a satisfactory set of reactive and deliberative head motion behaviors for our virtual pedestrian model and to model “families” of pedestrians that move together in small groups. We will also pursue new applications of our simulator to archaeology, computer vision, and other fields.

Appendix A: The Basic Reactive Behavior Routines

Routine A: Static obstacle avoidance. If there is a nearby obstacle in the direction of locomotion, lateral directions to the left and right are tested until a less cluttered direction is found (Fig. 4(b)). If a large angle (currently set to 90°) must be swept before a good direction is found, then the pedestrian will start to slow down, which mimics the behavior of a real person upon encountering a tough

array of obstacles; i.e., slow down while turning the head to look around, then proceed.

Routine B: Static obstacle avoidance in a complex turn. When a pedestrian needs to make a turn that cannot be finished in one step, it will consider turns with increasing curvatures in both directions, starting with the side that permits the smaller turning angle, until a collision-free turn is found (Fig. 5(B)). 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 (Fig. 5(C)). To maintain a comfortable distance from each individual C_i in this temporary crowd, a directed repulsive force (cf. [HM95]) given by $f_i = r_i(d_i/|d_i|)/(|d_i| - d_{min})$ is exerted on H , where d_i is the vector separation of C_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 C_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 $a_f \Delta t$ and $a_l w_i \Delta t$ are added to H 's current desired velocity. The crowding factor w_i determines H 's willingness to “follow the crowd”, with a smaller value of w_i giving H a greater tendency to do so (currently $1.0 \leq w_i \leq 5.0$).

Routine D: Avoid oncoming pedestrians. To avoid pedestrians not in one's *temporary crowd*, a 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* (Fig. 5(D1)). In the case where the trajectories of H and C_i are almost parallel and will not intersect imminently, a *head-on collision* (Fig. 5(D2)) may still occur if their lateral separation is too small; hence, H measures its lateral separation from oncoming pedestrians. Among all collision threats, H will pick the most imminent one C^* . If C^* poses a *head-on 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 H determines that it will arrive sooner, it will increase its speed and turn slightly away from C^* ; otherwise, it will decrease its speed and turn slightly towards C^* (Fig. 5(D1)). 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 Routines C and D, since in highly dynamic situations predictions have a nonzero probability of being incorrect. Once a pedestrian perceives another pedestrian within its front safe area (Fig. 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 (see Section 4.3.1), 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 (Fig. 5(F)) and, if so, it will simply slow down to a stop, let those threatening pedestrians pass, and proceed.

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 Mauricio Plaza-Villegas for his invaluable contributions to the implementation and visualization of the Penn Station model (which was distributed to us by Boston Dynamics Inc.) and its integration with the DI-Guy software API.

References

- [ALA*01] ASHIDA K., LEE S., ALLBECK J., SUN H., BADLER N., METAXAS D.: Pedestrians: Creating agent behaviors through statistical analysis of observation data. In *Proc. IEEE Conf. on Computer Animation* (Seoul, Korea, 2001), pp. 84–92.
- [BA00] BLUE V., ADLER J.: Cellular automata model of emergent collective bi-directional pedestrian dynamics. In *Proc. Artificial Life VII* (August 2000), pp. 437–445.
- [BJTG98] BATTY M., JIANG B., THURSTAIN-GOODWIN M.: Local movement: Agent-based models of pedestrian flow. *Center for Advanced Spatial Analysis Working Paper Series 4* (1998).
- [BMS04] BOTEVA A., MÜLLER M., SCHAEFFER J.: Near optimal hierarchical path-finding. *Journal of Game Development 1*, 1 (2004), 7–28.
- [BPW93] BADLER N., PHILLIPS C., WEBBER B.: *Simulating Humans: Computer Graphics, Animation, and Control*. Oxford University Press, 1993.
- [FMS*00] FARENC N., MUSSE S., SCHWEISS E., KALLMANN M., AUNE O., BOULIC R., THALMANN D.: A paradigm for controlling virtual humans in urban environment simulations. *Applied Artificial Intelligence 14*, 1 (2000), 69–91.
- [FTT99] FUNGE J., TU X., TERZOPOULOS D.: Cognitive modeling: Knowledge, reasoning and planning for intelligent characters. In *Proc. SIGGRAPH 99* (1999), pp. 29–38.
- [GKM*01] GOLDENSTEIN S., KARAVELAS M., METAXAS D., GUIBAS L., AARON E., GOSWAMI A.: Scalable nonlinear dynamical systems for agent steering and crowd simulation. *Computers & Graphics 25*, 6 (2001), 983–998.
- [GM85] GIPPS G., MARKSJO B.: A micro-simulation model for pedestrian flows. *Mathematics and Computers in Simulation 27* (1985), 95–105.
- [HM95] HELBING D., MOLNAR P.: Social force model for pedestrian dynamics. *Physical Review 51*, 5 (1995), 4282–4286.
- [KCR98] KOECHLING J., CRANE A., RAIBERT M.: Applications of realistic human entities using DI-Guy. In *Proc. Spring Simulation Interoperability Workshop* (Orlando, FL, 1998).
- [LD04] LAMARCHE F., DONIKIAN S.: Crowd of virtual humans: A new approach for real time navigation in complex and structured environments. *Computer Graphics Forum 23*, 3 (2004), 509–518.
- [LMM03] LOSCOS C., MARCHAL D., MEYER A.: Intuitive crowd behaviour in dense urban environments using local laws. In *Theory and Practice of Computer Graphics* (2003), IEEE, pp. 122–129.
- [Lov93] LOVAS G. G.: Modeling and simulation of pedestrian traffic flow. In *Proc. European Simulation Multiconference* (1993).
- [LRBW04] LOYALL A. B., REILLY W. S. N., BATES J., WEYHRAUCH P.: System for authoring highly interactive, personality-rich interactive characters. In *SIGGRAPH/EG Symposium on Computer Animation* (2004), pp. 59–68.
- [MH03] METOYER R., HODGINS J.: Reactive pedestrian path following from examples. In *Computer Animation and Social Agents* (2003), pp. 149–156.
- [MT01] MUSSE S., THALMANN D.: Hierarchical model for real time simulation of virtual human crowds. *IEEE Transactions on Visualization and Computer Graphics 7*, 2 (2001), 152–164.
- [NRTMT95] NOSER H., RENAULT O., THALMANN D., MAGNENAT-THALMANN N.: Navigation for digital actors based on synthetic vision, memory and learning. *Computers and Graphics 19*, 1 (1995).
- [Rey87] REYNOLDS C. W.: Flocks, herds, and schools: A distributed behavioral model. *Proc. SIGGRAPH 87, Computer Graphics 21*, 4 (July 1987), 25–34.
- [Rey99] REYNOLDS C. W.: Steering behaviors for autonomous characters. In *Proc. Game Developers Conf.* (1999), pp. 763–782.
- [Sch02] SCHADSCHNEIDER A.: Traffic flow: A statistical physics point of view. *Physica A313* (2002), 153–187.
- [SE01] SCHRECKENBERG M., (EDS.) S. S.: *Pedestrian and Evacuation Dynamics*. Springer-Verlag, 2001.
- [SGC04] SUNG M., GLEICHER M., CHENNEY S.: Scalable behaviors for crowd simulation. *Computer Graphics Forum 23*, 3 (2004), 519–528.
- [ST05] SHAO W., TERZOPOULOS D.: Environmental modeling for autonomous virtual pedestrians. In *SAE Symposium on Digital Human Modeling for Design and Engineering*. (Iowa City, IA, 2005).
- [TDB*02] TOMLINSON B., DOWNIE M., BERLIN M., GRAY J., LYONS D., COCHRAN J., BLUMBERG B.: Leashing the alphawolves: Mixing user direction with autonomous emotion in a pack of semi-autonomous virtual characters. In *SIGGRAPH/EG Symposium on Computer Animation* (2002), pp. 7–14.
- [TT94] TU X., TERZOPOULOS D.: Artificial fishes: Physics, locomotion, perception, behavior. In *Proc. SIGGRAPH 94* (July 1994), pp. 43–50.
- [UdHCT04] ULICNY B., DE HERAS CIECHOMSKI P., THALMANN D.: Crowdbush: Interactive authoring of real-time crowd scenes. In *SIGGRAPH/EG Symposium on Computer Animation* (2004), pp. 243–252.