Abstract
We address the difficult open problem of emulating real pedestrians in urban environments. Our comprehensive artificial life modeling approach to this problem integrates motor, perceptual, behavioral, and cognitive components. Our contribution is a sophisticated new model of individual 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 a large urban environment. To support a variety of natural interactions between pedestrians and their environment, we represent the latter using hierarchical data structures that efficiently support the perceptual queries of pedestrians that sustain their behavioral responses and to enable them to plan their actions on global and local 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, 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". Although standard computer graphics techniques enable a virtual reconstruction of the train station with stunning geometric and photometric details, 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.

In contrast to the substantial literature on so-called "crowd animation", our goal has been to develop a comprehensive model of individual pedestrians that is autonomous, decentralized, and capable of performing a broad variety of activities in large-scale synthetic urban spaces. Our artificial life approach to modeling humans incorporates the modeling of pedestrian appearance, locomotion, perception, behavior, and cognition. We deploy a multitude of self-animating virtual pedestrians within a large-scale environment model, a VR reconstruction of the original Penn Station (Figure 1), which 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. For a survey of issues relevant to human simulation in general, see the book by [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 imagination of CG researchers [ALA01, MT01]. The topic has also been of some interest 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 [FMS00, MH03]. Our work is applicable to all of these areas and others, such as computer vision.
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 [Mas05].

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. The goal of our work is to develop a comprehensive, self-animating model of individual human beings that incorporates nontrivial human abilities 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 this comprehensive artificial life approach and adapt it for the first time to the case of an autonomous virtual human model suitable for large-scale urban spaces. In particular, we pay serious attention to deliberative human activities 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 the development of a large-scale urban environment model, which is described in detail elsewhere [Ano05], and which we summarize next. We represent the virtual environment by a hierarchical collection of maps. As illustrated in Figure 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 relevant information to perceptual queries, and (iii) path maps, which enable online path-planning for navigation.

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 (ground, walls, benches for example). The representation assumes that the walkable surface in a region may be mapped onto a horizontal plane without loss of essential geometric information.
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 [Ano05].

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 character model similar to the one advocated by [FTT99] for an aquatic “merman” and dinosaurs. 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.

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. [KCR98] [Bos04]. DI-Guy provides a variety of textured human models together with a set of basic motor skills, such as strolling, walking, sitting, etc., implemented using conventional IK and motion library techniques. DI-Guy is designed for users to script the actions of characters manually in space and time and it provides a tool called DI-Guy Scenario (which we do not use) to facilitate this. It also provides an SDK that allows external C/C++ programs to control a character’s basic motor repertoire. We build our own control programs at the motor, perceptual, behavioral, and cognitive levels.

Emulating the natural appearance and movement of human beings is a highly challenging problem and, not surprisingly, DI-Guy suffers from several limitations. First, the character models are insufficiently detailed. Second, and more important, characters cannot synthesize the full range of motions needed in a busy 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 pedestrians to deal with highly dynamic urban environments.

Moreover, we have implemented a motor control interface between the kinematic layer of DI-Guy, and our higher-level behavioral controllers to be described in the next section. The interface accepts motor control commands from behavior modules, and 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. This is a seamless interface that hides the details of the underlying
kinematic layer from our higher-level behavior modules, enabling the latter to be developed more or less independently. Hence, any other suitable low-level 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 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 the desired perceptual acuity (Figure 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 (Figure 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 pre-defined 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, especially 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 interest. Given the sensing fan and the maximum number of sensed pedestrians, sensing is a constant time operation.

![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 tiers (color-coded) of the sensing fan.](image)

4.3. Behavioral Control

Realistic human behavioral modeling, whose purpose is to link perception to appropriate actions in an autonomous 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 [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 (Figure 5), each suitable for a different set of situations in a densely populated and highly dynamic environment. Given that a pedestrian possesses a set of motor skills, such as standing still, moving forward, turning in different 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, Routines A–F, are provided in the Appendix which we encourage the reader to review at this time.

Several remarks regarding the 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 threat passes and then proceed in the direction that is free of static obstacles. 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 (Figure 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

© The Eurographics Association and Blackwell Publishing 2005.
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 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) 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. A pioneering effort on autonomous navigation is that by [NRTMT95]. [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, including 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 and lead to lengthy delays. In our experience, available techniques, such as queuing by [Rey99], self-organization by [HM95] and global crowd control techniques by [MT01] cannot completely tackle the problem, as it involves highly dynamic two way traffic and requires quick and flexible response 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 people. Second, all connecting passageways between two places should be used in balance. The two behaviors are detailed next.

Passageway navigation. Inside a narrow passageway in real life, if all pedestrians are traveling in the same direction, they will tend to spread out in order to see further ahead and maximize their pace. However, people will compromise and, if necessary, form opposing lanes, once oncoming traffic is observed, to maximize the throughput in both directions. Our virtual pedestrians employ 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 (Figure 7(a)). Restricting H’s travel direction within a safety fan defined by the boundaries (as shown in the figure) guarantees that it 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 (Figure 7(b)). The search starts from H’s current direction and continues clockwise. If 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 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 (Figure 7(d)). Finally, Routine C described in the previous section is used to maintain a safe separation between ongoing pedestrians. By altering their crowding factor \( w_i \) based on the observation of oncoming traffic, pedestrians can spread out or draw tightly to accommodate the change of situation (Figure 7(c)).

**Passageway selection.** Real people are usually motivated enough to pick out the best option from several available access facilities, depending on both personal preferences and the real-time situation in and around those facilities. 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 at the level of each individual, has a global effect of balancing the loads at different passageways.

With the above 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 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 (Figure 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 foot step 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. 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 an increasingly important factor 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, employing reactive behaviors for static obstacle avoidance (Routines A and B) 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. 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 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 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.

![Figure 9: Action selection in a pedestrian commuter.](image)

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 [FTT99] is responsible for creating and executing plans suitable for an autonomous agent. Such a model must be able to make reasonable global navigation plans in order for a pedestrian to travel deliberatively (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 in the behaviors “perception-guided navigation” and “arrival at target” in section 4.3.2.

On the other hand, the pedestrian also needs the ability to decide, in a dynamic environment such as a train station, whether and when a new plan is needed. These all 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. It exploits the topological map at the top level of the environment model (Figure 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 [Ano05]), 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 described in Section 4.3.2.

Coupling cognitive control to behavioral control is necessary in order to increase the realism of our pedestrian model. To this end, every pedestrian maintains a stack of goals with the top one being the current goal. The stack is accessible to both the deliberative, cognitive layer and the 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 decomposed into several parts, allowing the cognitive controller to handle those within its ability (such as planning a path), and the behavioral controller to handle the remaining 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 (say, thirsty & vending machine nearby, then push “plan to get a drink”). This usually interrupts the execution of the current task and typically invalidates it at the same time. When it is time for the interrupted task to resume, a new plan is often needed. Intuitively, then, the goal stack remembers “what needs doing”, the cognitive controller decides “how to do it” at a higher, abstract level, the mental state variables dictate “why it should be done”, and the behavior controller decides “how to do it” at a lower, concrete level and attempts to “get it done”.

5. Results

Our pedestrian animation system 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) x 150(w) x 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 the objects.

In our simulation experiments, we populate the virtual station with five different types of characters: commuters,
tourists, performers, policemen, and patrolling soldiers. With everyone 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 an Intel Xeon 2.8GHz machine with 1GB memory. The total length of every test is 20 minutes in virtual world time. Figure 10 shows how the computational load increases with the number of pedestrians in the simulation. 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 as they are not controllable by us and turn out to be very time-consuming. 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 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.

5.2. Animation Examples

We will now describe several representative simulations that demonstrate specific functionalities. (Note to the referees: These simulations are provided as additional material.) To help place the animation scenarios in context, Figure 12 shows a top-down plan view of the station. Note that the station has a complete ceiling, as is shown in Figure 1.

Following an Individual Commuter. In the first animation, we follow a commuter, in order to clearly make the point that we have implemented a comprehensive pedestrian model that makes every pedestrian a complete individual with an interestingly broad behavioral and cognitive repertoire. The camera follows 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 for 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’s time for our subject to proceed to his train platform. He passes 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 other pedestrians, but our subject has no time to watch the performance and descends stairs to the train platform.

Pedestrian Activity in the Train Station. The second simulation, which includes over 600 autonomous pedestrians, demonstrates various pedestrian activities that are typical for a train station from various different perspectives. In addition, some behaviors that are not shown in the first animation are included here, such as 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 several rather narrow staircases. The camera focuses on the main waiting room area first and then moves to concourse area afterward, in order to visualize some of the various interesting activities that are occurring simultaneously in different parts of the station.
6. Conclusion and Future Work

We have developed a sophisticated, large-scale human animation system driven by a comprehensive model of individual pedestrians that combines behavior, perception, and cognitive control components. 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 individual activities. Although motion artifacts are still conspicuous in our animation results due to the limitations of the 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 close 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 pedestrian “friends” that move together in small groups.

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 (Figure 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 before proceeding.

Routine B: Static obstacle avoidance in a complex turn. When a pedestrian needs to make a turn that cannot be finished in one step, he 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 (Figure 5(a)). 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 $Ci$, this pedestrian, a directed repulsive force (cf. [HM95]) given by $f_i = r_i(d_i/d_i^0)/(d_i - d_{min})$ is exerted on $H$, where $d_i$ is the vector separation of $Ci$ 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 “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 $a_l \Delta t$ 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 those pedestrians not in one’s temporary crowd, a pedestrian $H$ estimates its own velocity $v$ and the velocities $v_i$ of nearby pedestrians $Ci$. Two types of threats are considered here. By intersecting its own linearly extrapolated trajectory $T$ with the trajectories $T_i$ of each of the $Ci$, pedestrian $H$ identifies potential collision threats of the first type: cross-collision (Figure 5(e)). In the case where trajectories of $H$ and $Ci$ are almost parallel and will not intersect imminently, 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. Among all collision threats, $H$ will pick the most imminent one $C^*$. 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 $H$ 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 $C^*$.

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 his front 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 will be 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, oc-

© The Eurographics Association and Blackwell Publishing 2005.
casionally 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.

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


© The Eurographics Association and Blackwell Publishing 2005.