University of California
Los Angeles

Door and Doorway Etiquette for Virtual Humans

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Wenjia Huang

2014

Abstract of the Dissertation

Door and Doorway Etiquette for Virtual Humans

by

Wenjia Huang

Doctor of Philosophy in Computer Science

University of California, Los Angeles, 2014

Professor Demetri Terzopoulos, Chair

In the context of realistic autonomous human animation, we introduce a novel framework for simulating a variety of nontrivial, socially motivated behaviors that underlie the orderly passage of pedestrians through doorways, especially the common courtesy of opening and/or holding doors open for others, an important etiquette that has been overlooked in the graphics literature to date. Creating self-animating virtual humans that can emulate this common social activity requires serious attention to the interplay of visual perception, navigation in constrained doorway environments, manipulation of a variety of door types, and high-level decision making based on social considerations.

To address this complex human simulation problem, we take an artificial life approach to modeling autonomous pedestrians, proposing a layered architecture comprising mental, behavioral, and motor layers. The behavioral layer is a coupled composite of two stages: (1) a decentralized, agent-based strategy for dynamically determining the well-mannered ordering of pedestrians around doorways, and (2) a state-based model that directs and coordinates a pedestrian's interactions with the door and synthesizes various door holding actions with the support of a flexible procedural motion model at the motor layer. The mental layer comprises a

Bayesian network decision model that selects appropriate door holding behaviors by considering both internal and external social factors pertinent to pedestrians interacting with one another in and around doorways.

Our framework addresses the various door types in common use and supports a variety of doorway etiquette scenarios with efficient, real-time performance.

The dissertation of Wenjia Huang is approved.

Joseph Teran

Glenn Reinman

Song-Chun Zhu

Demetri Terzopoulos, Committee Chair

University of California, Los Angeles

2014

To my mother and father.

Table of Contents

List of Figures

List of Tables

Acknowledgments

First, I am very grateful to my advisor, Professor Demetri Terzopoulos. His great experience and vision in the human simulation domain inspired and drove my PhD research forward. Beyond the high ambition for and perfectionism towards research that I learnt from him, I am very thankful for his patience, continuous guidance, and encouragement about my research. I could never have expected to achieve the work reported in this thesis without all his positive influence.

I would like to thank Professors Song-Chun Zhu, Glenn Reinman, and Joseph Teran for serving on my Thesis Committee and offering their helpful insights regarding my work. I would also like to thank former Committee member Professor Eddo Stern for providing valuable suggestions from the video game perspective. My thanks also go to Professor Petros Faloutsos for providing essential suggestions on the animation aspects of my initial work.

I acknowledge the important contributions of several of my colleagues: Billy Hewlett directed my initial research in this field during my visit to the MAGIX Lab in the summer of 2008. Mubbasir Kapadia and Shawn Singh created the Steersuite software package, which I integrated into my work and which greatly facilitated my research. Mubbasir also collaborated on my initial technical paper writing effort. Dr. Wei Shao helped to clarify my questions on his Autonomous Pedestrians and allowed me to peruse his software, which served as a great reference. Former MS student Miri Kim did some preliminary work with me on synthesizing door opening and entering motions for virtual humans.

My gratitude also goes to my other lab colleagues, namely Weiguang Si, Lap-Fai Yu, Kresimir Petrinec, Gabriele Nataneli, Brian Allen, Masaki Nakada, Eduardo Poyart, Sharath Gopal, Matthew Wang, Gergely Klar, Jingyi Fang, Chenfanfu Jiang, Xiaolong Jiang, Xiaowei Ding, Garett Ridge, Konstantinos Sideris, Diana Ford, and Alexey Stomakhin, for sharing their knowledge and ideas with me.

The research reported herein was supported in part by National Science Foundation Award No.0905671 and by a Dissertation Year Fellowship from UCLA.

On a personal level, I am deeply grateful to my parents, for their unconditional love, trust, and support, which made it possible for me to reach this far.

Vita

2009	B.E., Digital Media Technology Zhejiang University Hangzhou, China.
2010–2012	Research Assistant Computer Science Department University of California, Los Angeles Los Angeles, California
2012	Teaching Assistant Computer Science Department University of California, Los Angeles Los Angeles, California
2011	Development Intern USC Institute for Creative Technologies Los Angeles, California

Publications

W. Huang, M. Kapadia, and D. Terzopoulos (2010). "Full-Body Hybrid Motor Control for Reaching." Proceedings of the *Third International Conference on Motion in Games*, Zeist, the Netherlands, November, 2010.

CHAPTER 1

Introduction

Autonomous virtual humans are finding broad applicability in the entertainment industry and beyond. Increasingly higher demands are being placed on their realism—not just fidelity in their appearance and movement, but also in their behavior and social interaction. This is evident in lifelike non-player characters in many recently released video game franchises such as Grand Theft Auto^{TM}, Assassin's Creed^{TM}, and The Elder Scrolls^{TM}. Most existing work on autonomous virtual humans addresses either reactive behaviors in the absence of social considerations (e.g., avoiding physical collisions), or social behaviors with very loose motor interactions (e.g., gesturing in conversation). However, simulating behaviors with significant social and complex motor interactions, is a much more challenging problem to solve.

Despite the large body of literature on human animation, computer graphics researchers have not yet given serious consideration to the common door as a mechanism that evokes complex body movements and rich social interactions highly worthy of simulation. For example, in social simulation video games, such as "The Sims" series (published by Electronic Arts, Inc.), the level of sympathy one feels for the game characters would be enhanced if they could open and hold doors for others. In some AAA first-person shooter games, it would be more engaging if your allied leader can hold the door open while secretly signaling you to follow him closely.

Doors are ubiquitous impediments in our daily lives (Figure 1.1). Since a

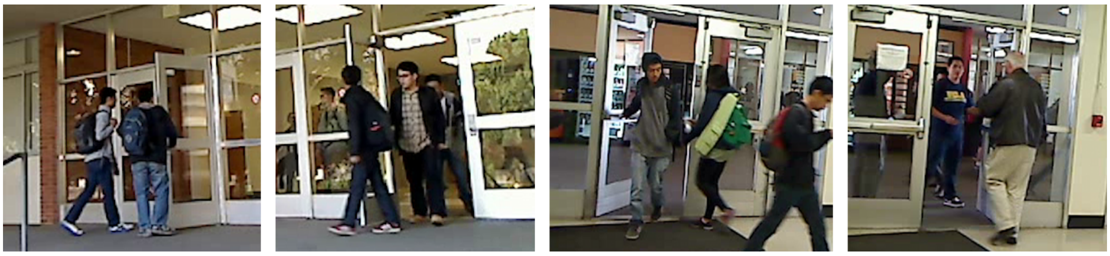


Figure 1.1: Door and doorway etiquette in real life.

doorway is a shared resource, usually allowing just one person or perhaps two people to pass at once, it induces interesting social situations when several people wish to pass through from the same or from opposite sides. The related social rules, the customary code of polite behavior known as “door(way) etiquette”, are broadly observed across different cultures.

There are several different types of doors in common use, each with particular interaction modalities, such as sprung hinged doors that open when pulled and/or pushed and close automatically when released, double doors that comprise two sprung hinged doors, and revolving doors. Using the sprung hinged door as an example, a standard social norm is to hold the door open for the convenience of others. Usually, a well-mannered person will hold the door open for someone following closely behind, or hold the door open in order to allow others to pass through the doorway first, as gentlemen will often do for ladies. However, these rules are not rigidly fixed and they will vary depending on dynamic factors related to a person’s character and frame of mind, such as their kindness and their sense of haste, as well as the state of their environment, such as the distance to the follower. Door-holding behaviors are rich and somewhat unpredictable, hence they are of great interest in the context of human simulation and animation.

Even a casual examination of real-world video footage (Figure 1.1) readily reveals that interacting with doors also involves nontrivial motor tasks. When a person opens and passes through a door, intricate stepping and manipulation actions are observed—they approach the door in an orderly manner, avoiding collisions with others while adhering to precedence, achieve a convenient location and orientation relative to the door, reach out for the door handle and pull the door open, and finally enter the doorway while perhaps holding the door open for others. If another person is following not far behind, one might hesitate and hold the door until the follower reaches it, or step out of the way to allow the follower to pass through the doorway first. Considering the diversity of highly adaptive movement exhibited throughout the entire doorway negotiation procedure, the potential complexity of the human simulation problem at hand is daunting.

The Artificial Life (ALife) approach to human simulation regards virtual humans from a decentralized, ego-centric perspective, as autonomous agents, modeling them comprehensively at the motor, perceptual, behavioral, and cognitive levels (Terzopoulos, 1999; Funge et al., 1999). It has been applied successfully to multi-human simulation (Shao and Terzopoulos, 2007; Pelechano et al., 2007; O'Sullivan and Ennis, 2011). The ALife architecture can also support probabilistic decision-making and social interactions that require an awareness and consideration for others (Yu and Terzopoulos, 2007). In this context, this dissertation provides the first complete solution for simulating orderly human behaviors and proper etiquette near doors and doorways (Figure 1.2).

1.1 Contributions

Espousing the ALife approach, we develop a hybrid architecture where each character is an autonomous, self-animating individual modeled with motor, behavioral, and mental functional layers. Our approach relies on egocentric perception as the basis for acquiring the environmental information necessary to drive individual decision-making, yielding rational behavior subject to social considerations in the presence of other individuals. Our novel system supports continuous, realtime animation performance with dynamic multi-character interactions mediated by

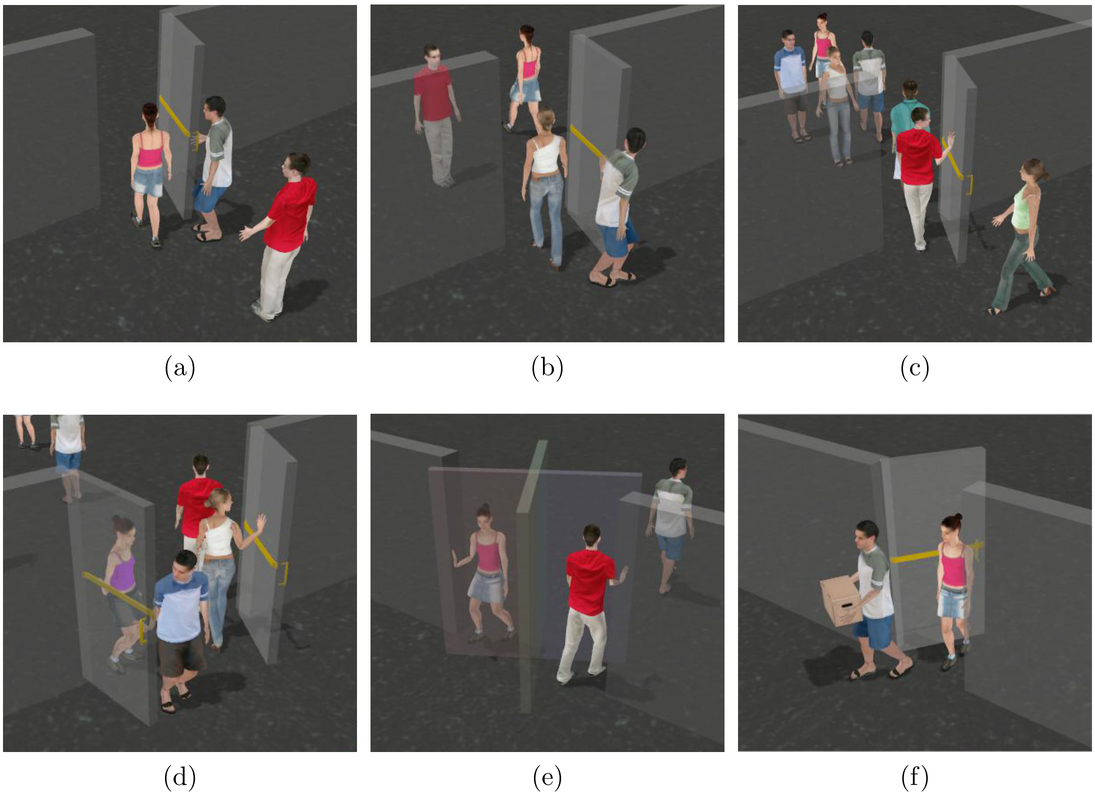


Figure 1.2: Simulated door and doorway etiquette scenarios involving three common door types—a single sprung door (a)–(c), a double sprung door (d), and a revolving door (e)—and involving a character carrying an object (f).

nontrivial environmental mechanisms—doors—which substantively influences the initiation and evolution of the character interactions.

At a high level, our agent-based simulation framework supports the decentralized perception of doors and other pedestrians in the vicinity of doorways, the design of orderly negotiation strategies for passing through doorways, and a state-based model for achieving versatile and stable door interaction generation driven by social considerations, which are encoded as Bayesian networks that govern appropriate door-holding action selection under dynamic conditions. At a lower level, we achieve plausible and robust motion generation by data-driven and procedural motion approaches. Despite their heterogeneity, these techniques are well integrated within the ALife architecture to systematically address our complex human simulation problem.

Our state-based door interaction model provides a robust approach to synthesizing interactions between doors and pedestrians, including the roles of door-opener and follower. It drives motion generation and supports different door holding motions. This model works in conjunction with the doorway ordering model so that while some pedestrians are performing door interactions, others are approaching the doorway in an orderly manner.

Our doorway ordering model is decentralized. Each pedestrian can sense other nearby pedestrians. We devise ordering rules that determine which pedestrians other pedestrians will follow, which we refer to as “leaders”. Leaders can change depending on the dynamic situation, subject to the constraint that they are not yet holding the door for specific followers. We introduce the notion of a “critical motion phase” to handle this complexity.

Our decision model for initiating door holding behaviors is based on Bayesian networks. The design of relevant factors is partially based on psychological studies (Santamaria and Rosenbaum, 2011). The values of the factors in the decision model vary dynamically and they determine action selection appropriate to the current situation.

Our layered and modular functional design architecture enables easy and clean extensibility, mainly by augmenting specific functional layers while modestly modifying other layers or leaving them untouched. For example, the support of various different doors, from single doors to double doors to revolving doors, and the animation of more complicated doorway scenarios, including introducing characters carrying objects, pushing baby strollers, and friends walking shoulder to shoulder. To handle such scenarios, we augment the motor and behavioral layers of the characters, and easily incorporate the new social factors into their mental layers, without disturbing the prior implementation of these layers. Our layered, modular framework also enables us to easily add gaze behaviors, which crucially enhances the fidelity of human behavior simulation.

As we change the type of door and its settings, as well as the number and various types of pedestrians approaching the doorway, our multi-human simulation system can generate automatically and in real time a broad variety of convincing animations demonstrating proper door(way) etiquette (Figure 1.2).

1.2 Overview

The remainder of the dissertation is organized as follows: Chapter 2 reviews related work. Chapter 3 describes our framework and its design principles. Chapter 4 presents the technical details of our behavioral synthesis models that support doorway ordering and door interactions. Chapter 5 presents our decision model and constituent factors that determine behavior generation. Chapter 6 presents the details of our motion generation method for accommodating the door interaction model. Chapter 7 presents our experiments and results. Chapter 8 concludes the thesis and discusses avenues for future work.

Appendix A fills in additional details about our step-based locomotion and steering system. Appendix B presents our work on gaze and attention modeling and animation. Appendix C presents our work on hybrid full-body motion control for reaching. Appendix D presents an application of our human simulator to visual surveillance in the context of "Virtual Vision".

CHAPTER 2

Related Work

No doubt due to its intricateness, the topic of human behavior around doors and doorways has only casually been addressed in the computer graphics literature, and there exists only some partially relevant literature from psychology and robotics/AI.

2.1 Agent-based Multi-Human Simulation

As opposed to so-called "crowd simulation", where multitudes of simple human characters are typically visualized from a distance, our work on doorway etiquette makes a much needed contribution to autonomous agent-based, multi-human simulation (see, e.g., (Shao and Terzopoulos, 2007; Pelechano et al., 2007; O'Sullivan and Ennis, 2011)), an endeavor whose objective includes the automatic animation of the *detailed* socially-motivated behaviors of smaller groups of people in urban environments. Even though detailed body kinematics, locomotion, perception, reactive behavior, and cognition is addressed in a distributed, agent-centric manner, the collective exhibits the natural characteristics of human crowds, as is the case in the real world. In this context, some recent work has incorporated psychological models for generating diverse steering behaviors (Guy et al., 2011; Durupinar et al., 2011) as well as probabilistic decision models for producing some nontrivial social behaviors (Yu and Terzopoulos, 2007). In principle, agent-based models are amenable to extension with additional behaviors that support doorway etiquette, but as we shall see, this is not easy to achieve in practice.

2.2 Relevant Psychological Theories

At the highest level, our work is informed by psychological findings, although quantitative psychological research on specific human social behaviors around doors is by no means extensive. Based on data analysis, Santamaria and Rosenbaum (2011) proposed a shared-effort model and studied two factors that could influence door etiquette. They found that close proximity between pedestrians yields a higher probability of holding the door open regardless of the number of followers; thus, we include the distance factor in our model. The influence of gender in door holding—that door-holders offer courtesies to females at a higher rate than to males—is also quantifiable (Webster et al., 2007). Finally, the variability of personality, such as kindness to others, is well documented (Pervin, 1996), as is the effect of urgency, so we include both in our decision model.

2.3 Multiple Character Interactions

Generating motions around doorways requires cooperative movements between multiple characters. On the motion level, graphics researchers have focused on the animation of interactions between multiple human characters (Kwon et al., 2008; Shum et al., 2008), exploring motion analysis and synthesis approaches to achieving cooperative multi-character movements, among them physically-based optimization (Liu et al., 2006), and motion editing (Kim et al., 2009). Lau and Kuffner (2005) deal simultaneously with steering behaviors and some simple reactive behaviors by building an abstract state-space, which facilitates motion planning and modification. Their work has inspired us to develop a state-based model to deal with the complex steering and cooperation tasks that arise as people order and coordinate themselves in the spatially constrained albeit highly dynamic doorway environment; however, we must also take into account the fact that people exhibit diverse behaviors driven by perception and social norms in the presence of others.

Some doorway passing ordering tasks for crowds have been approached through proxy agents (Yeh et al., 2008) and situation agents (Schuerman et al., 2010) that provide influence over other agents such that they can cooperate in order to pass through openings more efficiently, but these efforts ignore the existence of door mechanisms and the potentially involved manual interactions with them that form an important aspect of door etiquette. Based on the motion analysis of people approaching and opening doors (Egges, 2008), prior work includes highly limited results involving hinged (Lo and Zwicker, 2008) and revolving (Lee et al., 2009) doors, but they are included merely as “eye candy”—aside from collision avoidance, the rich social interactions mediated by doors and doorways have thus far been overlooked.

2.4 Object/Door Manipulation

Research on animating manipulation tasks generates motions that involve interactions with objects. Unfortunately, optimal motion planning subject to constraints (Yamane et al., 2004; Bai et al., 2012) is computationally expensive. However, the motion of the lower-body around a door is highly constrained, and we have found that parameterizing the motions in step space (van Basten et al., 2010; Singh et al., 2011b; Choi et al., 2003) offers an efficient approach to tackling this problem. We considered extensions to synthesize full-body motions under constraints (van Basten and Egges, 2011; Heck et al., 2006), but this was insufficient for dynamic situations involving interference from multiple pedestrians; hence, we devised our own motion synthesis solution that can achieve the required behaviors.

On the motion and control level, the door manipulation problem has been investigated in the robotics literature (Chitta et al., 2010; Arisumi et al., 2009; Nagatani and Yuta, 1995), but it has focused on enabling individual robots to open a door and mechanically pass through it in a non-human-like manner.

2.5 Planning Actions

In the domain of Artificial Intelligence, the problem of planning in the real world considers both scheduling constraints (e.g., timing of motions) and resource constraints (e.g., doorways allowing only a limited number of agents to negotiate at the same time) (Currie and Tate, 1991). More specifically, the problem of people holding doors or collaborating to pass through doorways in various ways resembles the multi-agent planning problem (Weiss, 1999). Through symbolic reasoning, these methods can dynamically generate plans to execute a task, but symbolic action generation fails to account for the inherent constraints and coordination required at the motion generation level, which largely simplifies the complexity of the problem.

CHAPTER 3

Framework

Conventional approaches to creating autonomously self-animating characters focus on motion synthesis, specifically on achieving natural motions for tasks involved a single character or for multi-character interactions. By contrast, the Artificial Life approach (Terzopoulos, 1999; Funge et al., 1999) incorporates motion synthesis as the lowest layer of a hierarchical character control architecture. An important benefit is that one can incorporate at the high level a mental model to purposefully direct the low-level motion generation. Moreover, it is easy to extend the autonomous character's abilities by including additional behaviors in the intermediate behavioral layer. We have adopted an artificial life architecture in tackling our problem. Figure 3.1 presents an overview of our framework.

When an autonomous pedestrian wishes to pass through a doorway, it will perceive the door and employ its locomotion/steering system to approach it. If the character perceives other pedestrians to the front of it, its ordering model will dynamically form a doorway-passing order. When the character's turn comes, its state-based door interaction model will be enabled, supported by a set of lower-level motion generation models. A high-level decision model influenced by social factors determines the door holding behavior during the door interaction phase and it affects passing order during the doorway phase. After passing through the doorway, the pedestrian will walk away, avoiding collisions with oncoming traffic.

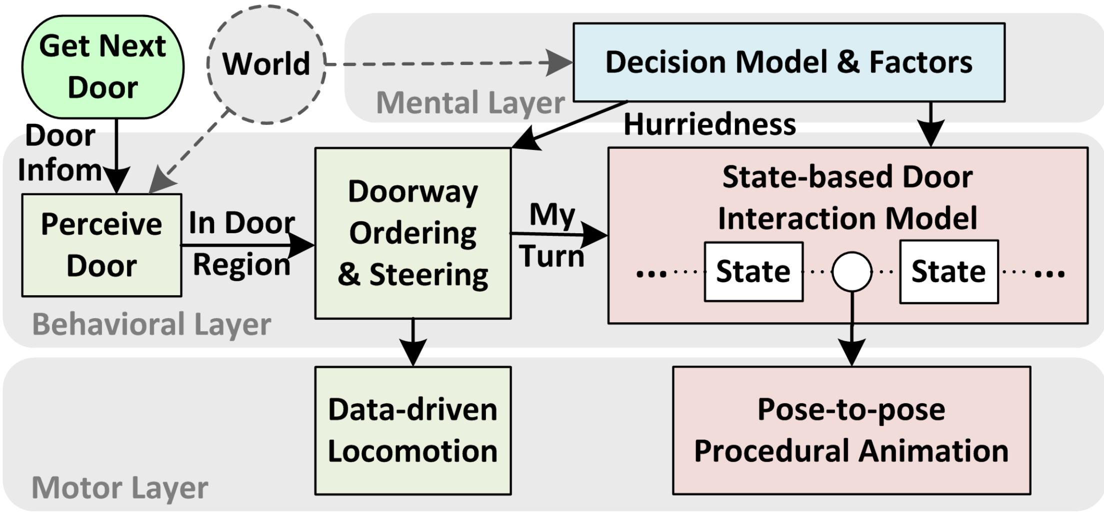


Figure 3.1: Our system comprises two phases, (i) perceiving and approaching the door (green) and (ii) interacting with the door (red). A high-level factor and decision model governs both phases (blue).

3.1 Behavioral Subgoals and Challenges

The task of walking through doorways involves the distinguishable subgoals of walking towards the door, possibly manipulating the door while walking through the doorway, and finally walking away from the doorway. The task of walking focuses on natural locomotion and efficient steering towards the door. More specifically, steering must deal with navigating towards the door while avoiding collisions with other pedestrians, and at the same time figuring out a doorway passing order. Locomotion must deal with the lower-body constraints for turning and transitioning from walking to stoping, which usually requires certain foot-adjusting space and number of completing steps. These two specific issues have been addressed together by prior work.

The door manipulation task focuses on natural, collision free door manipulations. A greater challenge is to keep the lower body moving naturally, in particular, performing locomotion, while the upper body manipulates the door. Therefore, this task requires generating full-body motions subject to constraints.

Even through the two tasks would at first seem unrelated, there is an inherent overlapping in regard to the decision-making and consequent motion when the two tasks are performed consecutively. As one person is manipulating the door, the subsequent person to pass through the doorway will be decided at some point, and that person will perform initial follow-up motions (such as carefully moving closer to the door in order to prepare to take over the door manipulation task). If this decision occurs too early, the result will be unresponsive to situations in which another person (say, in a hurry) cuts in and passes through next, or if the decision occurs too late, the next person to pass will perform motions in a less fluent manner, which will lead to a rather robotic animation.

CHAPTER 4

Behavioral and Motor Synthesis

The doorway ordering and door interaction models are connected, so we present their details together in this section. For simplicity, we will use the most common sprung hinged door in our technical description, and then adapt our methods to other door types, including the revolving door and double door.

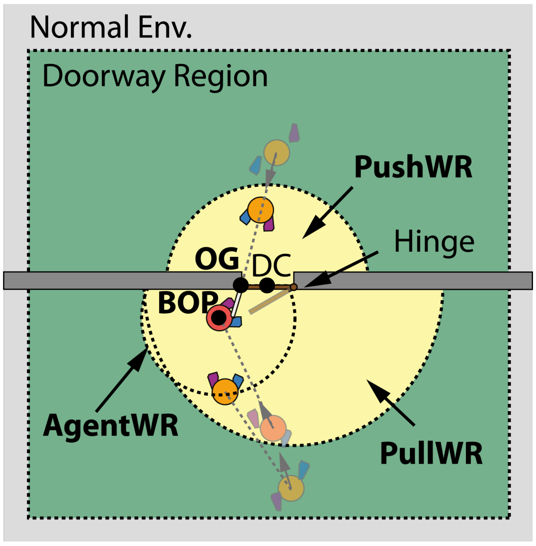
4.1 The Doorway

4.1.1 Door Perception

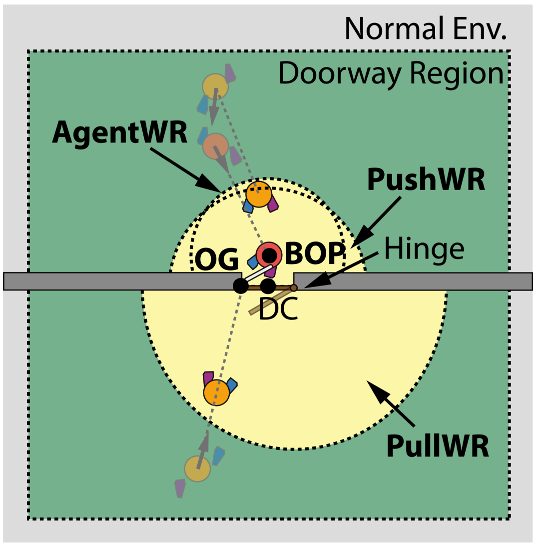
Following Shao and Terzopoulos (2007), the environment of the autonomous pedestrians is abstracted as a 2D gridmap which encodes the static obstacles, in our case the walls. When we assign to a pedestrian a target goal that lies on the far side of a doorway, it will first engage in global path planning to determine whether it must pass through a doorway. The pedestrian will autonomously walk towards the door through the use of its perception/locomotion/steering system, which is an integration of the work of Shao and Terzopoulos (2007), van Basten et al. (2010) and Singh et al. (2011a).

4.1.2 Doorway Ordering

When the pedestrian enters the doorway region (Figure 4.1), it will prepare to pass through the doorway using an agent-based doorway ordering process. This decentralized process ensures that the order is formed dynamically and can



(a)



(b)

Figure 4.1: (Top-down view) The waiting regions in the door-pulling (a) and door-pushing (b) cases. The Opposing Goal (OG) is the position towards which the opposing agent should proceed and wait. The Best Open-door Position (BOP) is the position that the door-opener should approach. The Pushing-side Waiting Region (PushWR) is a circle around the Door Center (DC) that prevents the pushing-side agents from approaching too close to the door thereby blocking the pulling-side agents. The Pulling-side Waiting Region (PullWR), centered at the door hinge, maintains the distance of the pulling-side agents from the door hinge, thus preventing collision with the door panel when the door is open. The Agent Waiting Region (AgentWR) is a circle centered at the door-opener, which prevents the waiting agents from getting too close to the door-opener.

naturally adapt to unexpected changes in the perceived environmental situation. For example, if a hurried pedestrian cuts in front of others, they can adapt by giving way.[1] In fact, the passing order is subject to change up to and until a critical motion phase associated with the door-opener role (which is embedded in the state-based door interaction model described in Section 4.2). If the door-opener has already engaged in that phase of door manipulation, the follower will commit to passing next.[2]

[1]By contrast, using a centralized global queue to maintain the order of pedestrians will not adequately deal with dynamic situations. Even if a queue is continually updated, it lacks the egocentric perspective of each of the agents in their environment.

[2]For the time being, we do not enable a committed follower to change their mind about passing next. Doing so could give rise to more intricate scenarios where another pedestrian

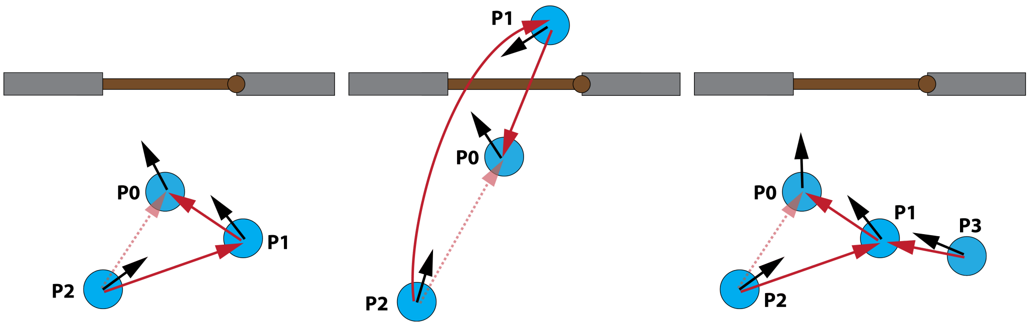


Figure 4.2: (Top-down view) Agent-based doorway ordering process (assuming no effect from waiting time). (a) P_2 selects P_0 as the initial leader, but then switches to P_1 since P_1 is closer to P_0. (b) P_2 modifies its leader from P_0 to P_1 on the opposite side since P_1 is closest to P_0. (c) It is therefore possible that two pedestrians, P_2 and P_3, will have the same leader P_1, which might create conflict when initiating the follower role w.r.t. the door interaction model.

Pedestrians should wait in natural positions so as not to impede other pedestrians currently passing through the doorway. We define several waiting regions as shown in (Figure 4.1), whose measurements are given in Section 4.4.3. If the door-opener is situated on the pulling side of the door, its followers on the same side should stop if the door-opener enters the doorway Region(PullWR ∪ AgentWR), and its opposing-side follower should stop if it enters Region(PushWR). If the door-opener is situated on the pushing side of the door, the corresponding waiting regions are Region(PushWR ∪ AgentWR) and Region(PullWR). If there are multiple pedestrians waiting at one side, each pedestrian must wait behind its self-selected leader, typically by defining AgentWR for its leader.

More specifically, the ordering algorithm works as follows (Figure 4.2(a)(b)): Each pedestrian will sense its environment, identify all pedestrians to the front (on the same side of and closer to the door), and choose the closest pedestrian as initial "leader". Subsequently, the leader selection process proceeds according to Algorithm 1. In this algorithm, "compete" means to evaluate who has the advan-

takes over and passes next. This merits future work.

Algorithm 1: Leader Selection.

if *I have an initial leader and any other pedestrians on the same side of the door choose the same initial leader* **then**
  compete with pedestrians on the same side of the door and select a new leader.
  if *the new leader is a follower or door-opener* **then**
    compete with pedestrians on the opposite side of the door to select the ultimate leader.
  end
else if *I have no initial leader and there is an opposing follower/door-opener* **then**
  choose the opposing follower/door-opener as initial leader, and compete with pedestrians on the same side as the leader (who may also choose this initial leader) to select the ultimate leader.
else
  choose the initial leader as the ultimate leader, or choose no leader.
end

tage over the pedestrian and choose the last one as a new leader. The evaluation is based on the distance to the leader and the waiting time: $E(dis_{leader}, t_{wait})$. Also in the algorithm "door-opener" refers to a pedestrian that has committed to interacting with the door. After the leader is selected using Algorithm 1, the pedestrian will decide its doorway behavior according to Algorithm 2.

Note that it is still possible for two agents to choose the same leader (Figure 4.2(c)), which does happen in reality. This will not cause a conflict when following the leader, but when it comes to committing to the follower role for triggering the door interaction model, it is possible for more than one pedestrian to assume this role. A mutex prevents more than one pedestrian from initiating the door interaction model at the same time, which corresponds to the situation that as one person observes another person taking the intended action first, they will abort their planned action and replan based on the new situation.

Algorithm 2: Decide doorway behavior.

```
if I have chosen a leader then
    if the leader is a door-opener and has entered its critical phase then
        commit to being a follower and approach the leader until close
        enough to the door (entered either PullWR or PushWR), and then
        initiate the door interaction model.
    else
        if the leader is on the same side then
            approach the leader.
        else
            /* the leader is on the opposite side                */
            approach the door.
        end
    end
else
    /* I do not have a leader                                    */
    if I am sufficiently close to the door then
        assume the role of door-opener and initiate the door interaction
        model.
    else
        approach the door.
    end
end
```

4.2 State-Based Door Interaction Model

We approach the problem of synthesizing motions that support multi-character interactions with doors based on the general ideas of Lau and Kuffner (2005), who synthesize motions based on abstract behavior models with states. They associated each state with a set of candidate segments of motion data, and the resulting motion is synthesised on-line while the characters interact with their environments. Due to the difficulty and workload of motion capturing people manipulating doors, and the challenge of highly constrained full-body motions in the narrow doorway environment, we have built a procedural motion model with inverse kinematics (IK), together with a compatible design of the door controller (will be described in Section 6.1), which can robustly support our state-based model.[3]

4.2.1 Door Interaction States and Transitions

The door-holding procedure is based on discretized motion steps, which enables collaboration in passing control of the door among multiple pedestrians. The door is considered the core component of the procedure, which implicitly guides and sets constraints during the interaction process. Perusing videos, we found there are typically two roles in interacting with a sprung hinged door, the opener and the follower. Furthermore, we identify and divide the states based on critical motion points viewed from the video that are important in synchronizing cooperative motions between the holder and follower. Subsequently, transitions are added to enable the interaction. Figure 4.3 illustrates the structure of our state-based door interaction model. In the following, we will describe the most prominent states along with the associated key poses.

Reaching and Opening Door Phases (Prepare H): When the door opener is close enough to the BOP (Figure 4.1), the pedestrian will enter the *OpenerInit* state, and switch from the doorway locomotion to the procedural motion model. The next goal is to move to the BOP and assume a convenient full-body pose for opening the door. It is not trivial to determine the BOP. Based on our observations of real-world videos, humans tend to stop at the gap of the door with a facing angle that can permit their handling-side arm to reach the door handle or push bar. We have manually defined the BOP and target body orientation, together with the target setting for the end-effector of the handling-side arm, such that the hand is at the door handle or push bar. This information is stored as a key pose and our procedural motion model (described in the next section) will synthesize motion

[3] As is at times evident in our demonstrations, this non-ideal solution produces motion artifacts. However; better motion models can straightforwardly be adapted within framework, so long as they support the abstraction of actions (states).

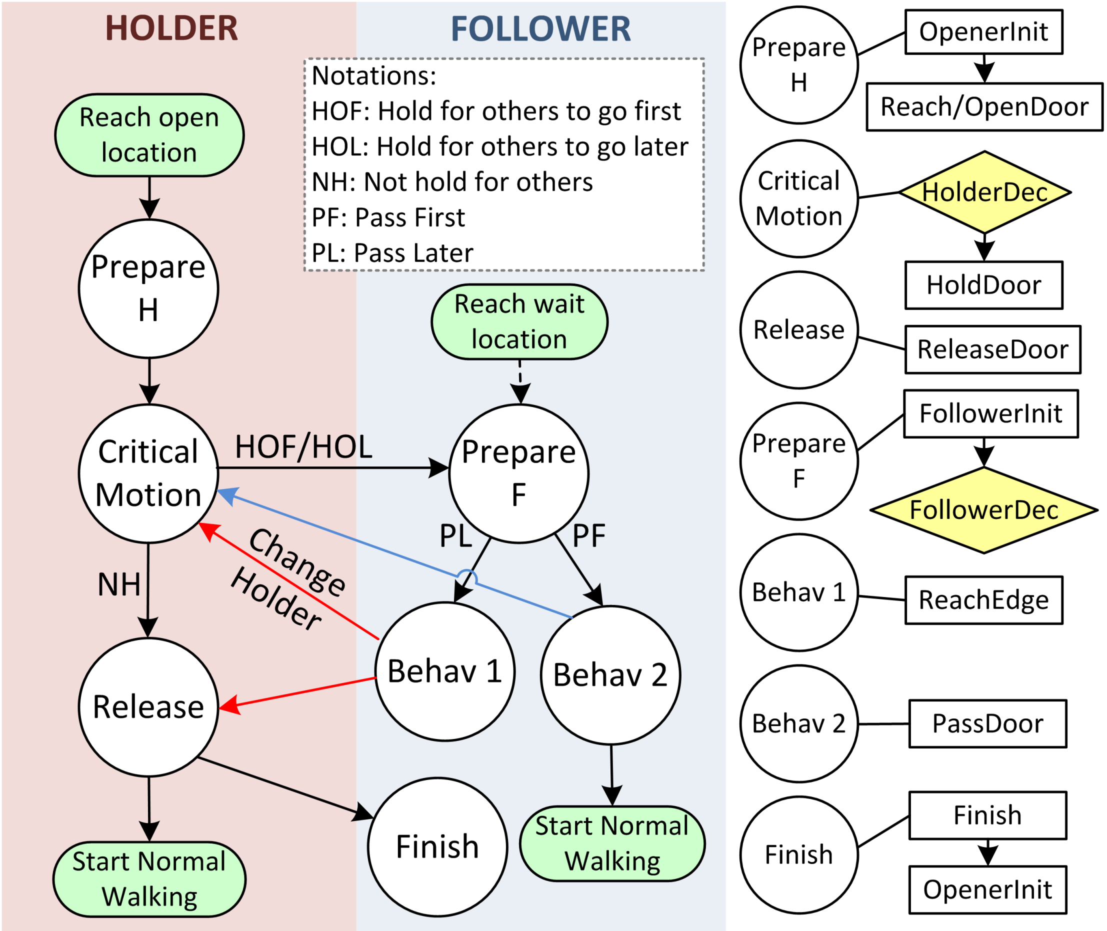


Figure 4.3: Structure of the state-based model for door interaction at two levels of abstraction, providing a intuitive and general view showing the interaction structures between holder and follower, as well as a detailed view indicating the specific state(s) of the motion phases implemented in each interaction state. The white circles represent interaction states. The white squares represent states of motion phases. The yellow diamonds represent decision states in which characters will choose different door-holding behaviors or following behaviors, and the green ovals represent states in which characters will switch between data-driven locomotion and the procedural motion model. A follower can become a holder by virtue of the red routine, at which point a PL follower takes control of the door, and then the former holder releases the door and the follower becomes the holder. A HOF holder can continuously hold the door for another pedestrian to pass through, according to the blue routine.

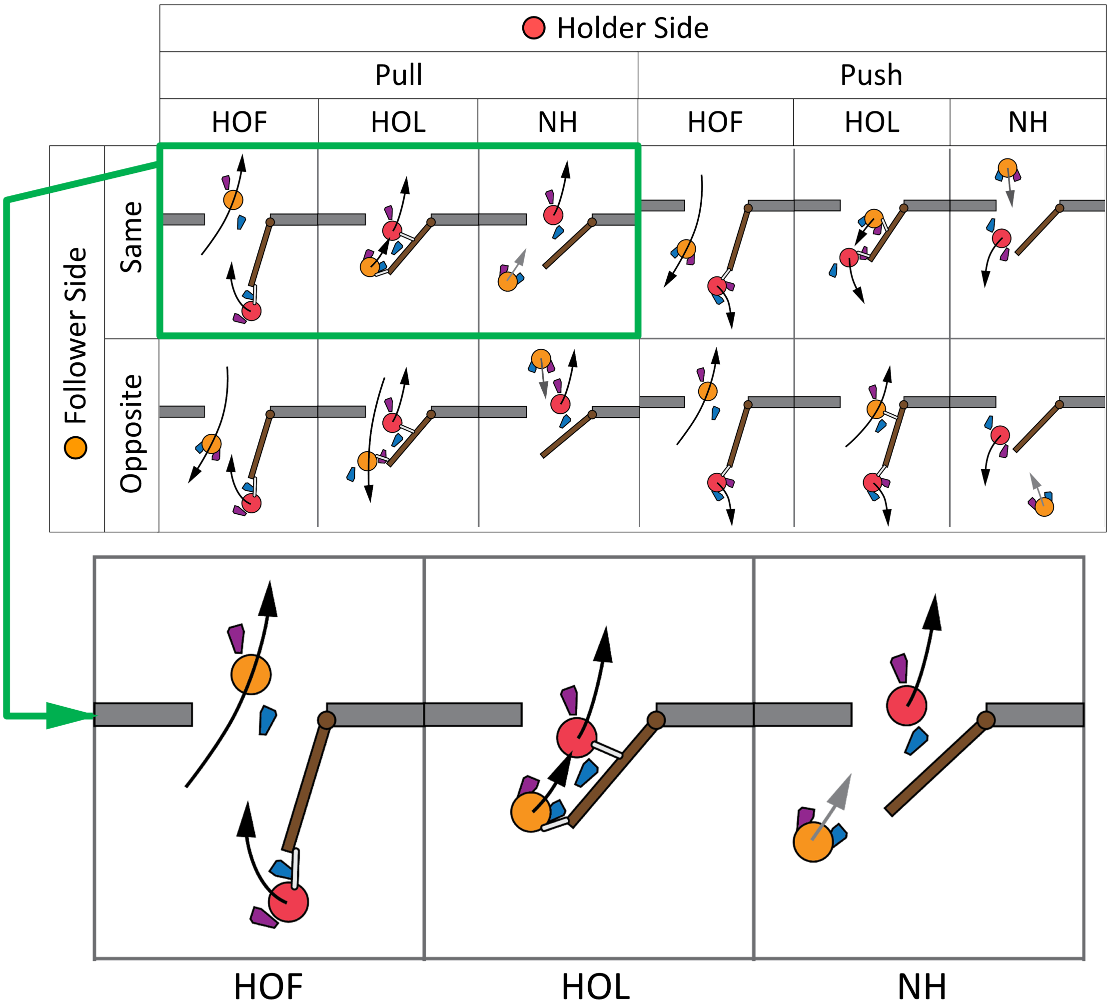


Figure 4.4: (Top-down view) All possible door holding behaviors with an opener (red) and a follower (yellow).

between the current pose and target pose (*ReachDoor* state). After completing the motion, the pedestrian will be in good start pose to open the door.

Pull vs Push: There are typically two door opening situations: the first is pulling the door handle from the outside, the second is pushing the door bar from the inside. In the pulling case, the character must stand outside the circle formed by the opening door, so that the door will not collide with the character's body when opened. In the pushing case, the character must stand within the circle, so that when pushing and moving from inside, the character will pass through the

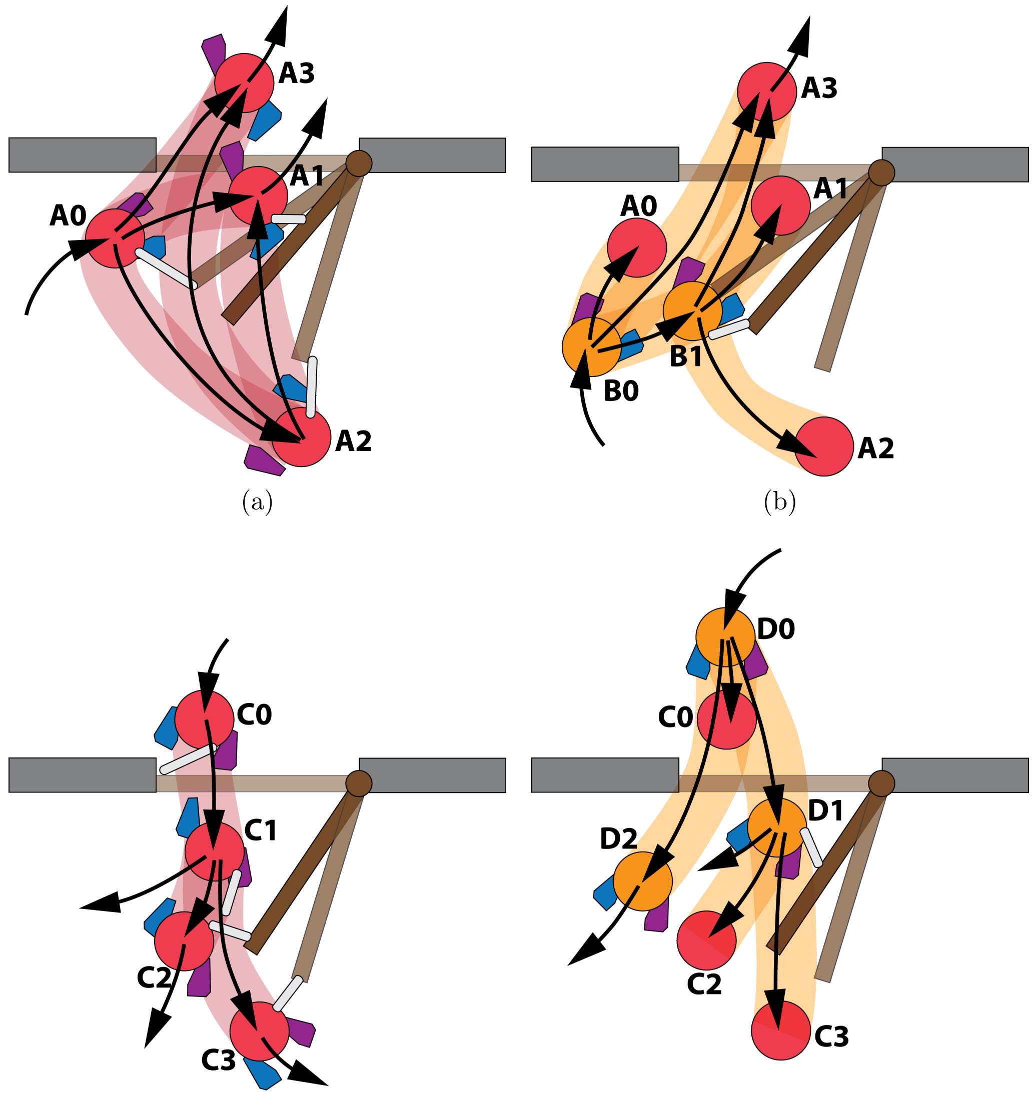
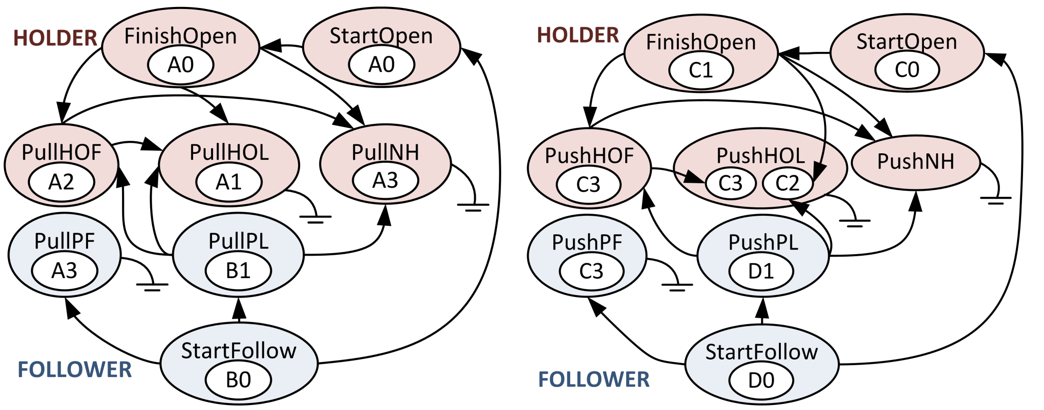


Figure 4.5: (Top-down view) The key poses and possible transitions between key poses occurring during door state transitions. The first line shows the door pulling case, while the second line shows the door pushing case. The red circles denote opener poses and the yellow circles denote follower poses.



(a) (b)

Figure 4.6: The relationships between key poses and different holding behaviors. The red ovals represent holder behaviors. The blue ovals represent follower behaviors. The white ovals inside denote key poses as indicated in Figure 4.5. The character can transition its behavior from follower to holder.

doorway rather than colliding with the walls on either side (refer to the BOP in both cases in Figure 4.1). To simplify the problem, we define the handling hand sides as the right hand for pulling the door and the left hand for pushing the door. There are multiple possible handling strategies, but we focus on the typical one to demonstrate the related behaviors.

The open door phase (*OpenDoor* state) is provided with a partially defined target pose that includes, in the pulling case, the spine rotation angle, with the constraint of the hand holding the handle, while in the pushing case, we have additional goals for the root, since the character must move forward in order to push the door open. Given the goal pose, the procedural motion model will generate motions between the reaching and opening door poses. The door opens automatically and the handling side hand follows the handle or bar by acquiring its location from the door controller. After the character achieves the goal pose, it will transition to the door holding phase (*HolderDec* state).

If the previous character did not hold the door, then as the next character reaches the door, the door might not have fully closed. In that case, the next opener can interrupt the door closing procedure and continue to open it, which is more realistic.

Door-Holding Phase: Thus far, the opener does the sole work of handling the door. Meanwhile, if the current follower is ready (reached the waiting region (Figure 4.1)), it will be committed, transition to procedural motion mode, and move to a position closely behind the opener that is convenient to take over the door. At this stage, the holder needs to commit to its decision to engage in door holding behavior (*HolderDec* state). This is the aforementioned critical motion phase used in doorway ordering so as to prevent further changes in the follower role. The holding behaviors are summarized in Figure 4.4. Basically, the holder has 3 different holding behaviors:

1. Hold the door for others to pass later (HOL). The opener will maintain the holding pose until the follower has reached the door and holds it. After the opener releases the door, the follower will take over control of the door.

2. Hold the door for others to pass first (HOF). The opener will maintain the holding pose while standing aside so that another agent can pass through the doorway. After the follower passes through, the opener will either continue to hold the door for the next character or release the door.

3. Not hold the door for others (NH). The opener will not wait until the follower reaches the door, but will release the door after it opening it to the minimal angle permitting passage.

After the opener achieves the holding pose (*HoldDoor* state), the state transition is under the follower's control (*FollowerInit* state). If the follower has completed the transition to the procedural motion mode, it needs to respond to the holder's behaviors with corresponding follower behaviors (*FollowerDec* state):

1. Pass later (PL) is responding to the holder's behavior of holding the door for others to pass later (HOL). After the follower reaches the door (*ReachEdge* state), the state will transition to *ReleaseDoor* and the holder will release the door. Then, the follower will become the holder by transitioning back to *HolderDec* of *Critical Motion Phase* (red routines in Figure 4.3).

2. Pass first (PF) is in response to the holder's behavior of holding the door for others to pass first (HOF). After completing the motion, the follower will transition to the normal locomotion/steering mode, and the state will transition back to *HolderDec* of *Critical Motion Phase* (blue routine in Figure 4.3). Then, the holder will make another door holding decision for the next follower.

Releasing and Taking Over Door Phase: In the door releasing phase (*ReleaseDoor* state), if there is a PL follower, the transition works as mentioned before; if there is no follower or the follower decides to not hold the door, the next state will be *FinishDoor*, which will trigger the door closing motion. If the follower detects this state, it will become the opener, transition to *OpenerInit*, and restart the door opening procedure. In both cases, the opener will record the current procedural motion pose and prepare to blend to data-driven locomotion control. Our design has the advantage of reusing states and it will not be affected by the number of followers in the procedure.

4.2.2 Human Motion Generation

At the motion level of our door interaction framework, we have developed a flexible procedural motion model which synthesizes motions between two key poses that are specified by two states. Reviewing the door interaction procedure, the related key poses and possible transitions between them are shown in Figure 4.5. During the holding door procedure, given that the follower might become the holder, the next key pose will be decided as shown in Fig 4.6. Given two key poses, a procedural motion model will generate motions between them, which enables continuous motion transitions under the environmental constraints. Section 6.3 presents the details of our pose-to-pose procedural animation method and Section 6.4 presents the details of our step-based procedural locomotion method.

In certain poses, such as when holding the door, pedestrians need to pause. For the sake of realism, it is necessary to keep them subtly moving. To this end, we employ Perlin Noise (Perlin, 1995). Moreover, head motion and eye movements are critical to realistic social behaviors; thus, we included an attention-driven Head/Eye system whose details are presented in Section 6.2.

4.3 Other Door Types

Our door behavior synthesis framework can accommodate door types other than the basic sprung doors, including revolving doors and sprung double hinged doors.

4.3.1 Revolving Door

A typical revolving door comprises 4 panels and a person can push one of them to pass through the doorway. The door control is modeled in 4 phases (Figure 4.7(a)), and it provides an additional function to identify the best entry phase for either side of the door given its current rotation (angle).

An important feature of the revolving door is that people entering the doorway from both sides can interact with the door at the same time. Usually one person will initiate the motion and the door can accommodate at most 2 followers at the same time, one from the same side and one from the opposite side. The opposite side follower can start the door interaction at any time without needing to wait until the pusher (holder) has completed the critical motion, whereas the same side follower needs to wait, which is similar to the sprung door situation when

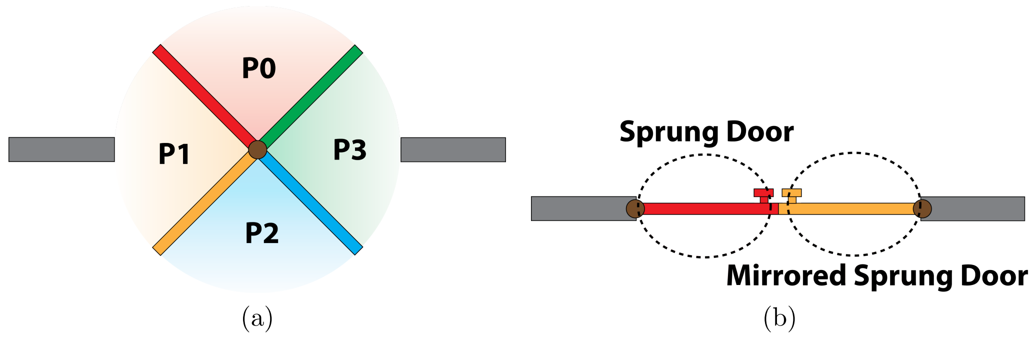


Figure 4.7: (a) Revolving door design with 4 phases. (b) Double door design as 2 juxtaposed sprung doors.

holding the door for others to pass later. These characteristics necessitate partial alterations in both doorway ordering and door interaction.

Regarding doorway ordering, we allow at most 2 holders and 2 followers at the same time. Pedestrians will prefer to select a leader from the same side. Subsequently, we allow at most 2 door interaction models running at the same time, each of which deals with a pair of interactions between one holder and one follower. Initially no one is handling the door, and one agent will be the initial opener with at most 2 followers; as the system runs, the opposite side follower will become the holder and will choose its own follower, potentially from the same side; thus 2 door interaction models are now running simultaneously. As there is only one possible behavior of pushing and passing through the door, this yields a simplification of the door interaction model (Figure 4.8). On the motion level, extra effort is placed on enabling the follower to catch up to the moving door and time a suitable entry point. Interesting behaviors emerge, such as in the case where the agent cannot catch up and must wait for the subsequent door opening.

4.3.2 Double Hinged Door

The double door introduces an interesting situation for doorway etiquette as it presents two doors from which to choose. In North America, people will prefer to pass through the right-hand door. However, this is not compulsory, and an

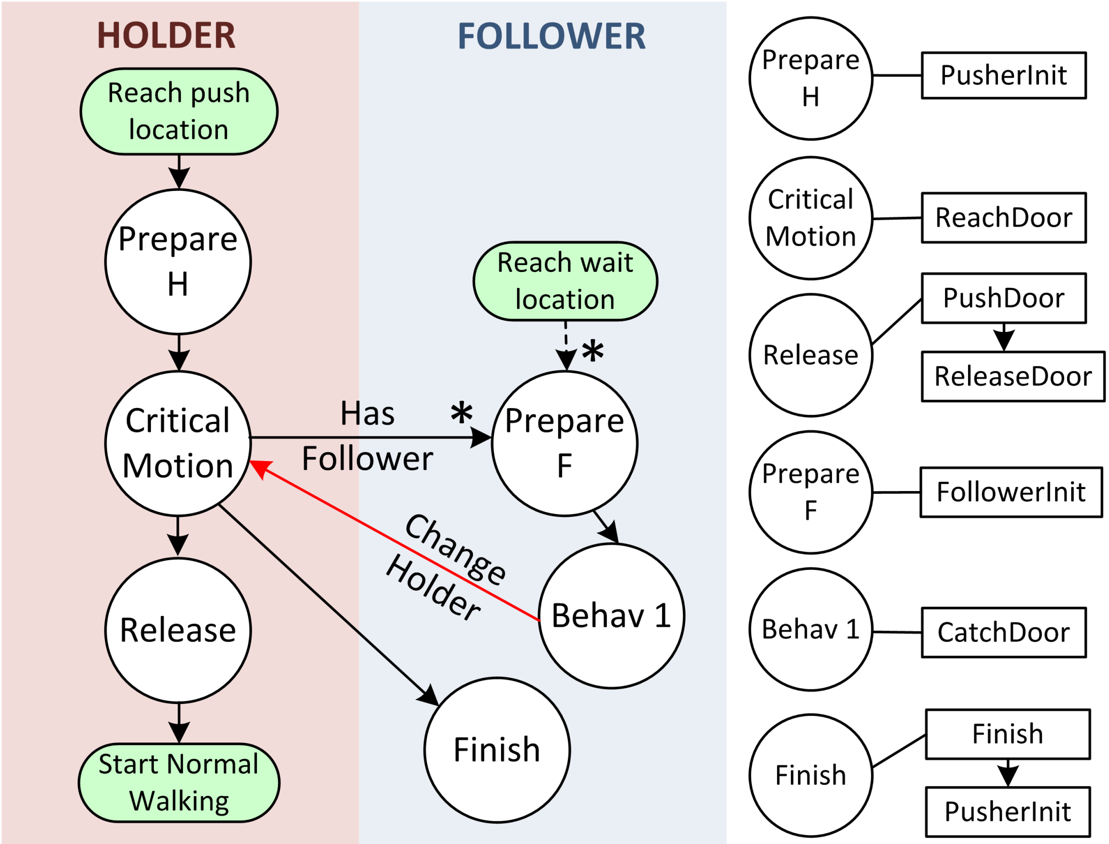


Figure 4.8: Revolving door adoption of the state-based door interaction model. Two stars: if the follower is on the same side, the *Prepare F* state needs to wait after the *Critical Motion* state of the holder. If the follower is on the opposite side, the *Prepare F* state can start as long as the follower has reached the wait location.

interesting phenomenon we observed from video is that, if there is a door-holder, the follower will tend to choose the held door, regardless of whether it is the right or left door. Moreover, if either door becomes congested with traffic, the other door will be put to use. To simulate both phenomena, at the doorway behavior level, pedestrians must observe the traffic loads of both doors.

The door control and door interaction model is simply a composition of two basic sprung doors, one a mirror image of the other (Figure 4.7(b)), and there is no need for modification. The major modification is in finding the leader (Algorithm 3).

Algorithm 3: Leader selection (double door case).

```
if there is no same-side pedestrian closer to the door than me then
    modify the leader by considering opposite-side pedestrians
    (Algorithm 4)
else if there is one same-side pedestrian before me then
    choose an initial leader (the one furthest from the door but before me)
    and modify the leader by considering opposite-side pedestrians
    (Algorithm 4)
else if there are two same-side pedestrians before me then
    choose an initial leader
    if I have not arrived within the waiting region then
        take the initial leader as the ultimate leader
    else
        modify the leader by considering opposite-side pedestrians
        (Algorithm 4)
    end
else
    /* more than two same-side pedestrians */
    choose an initial leader and take it as the ultimate leader.
end
```

Since the double door requires additional consideration on choosing a passing side, doorway behavior generation needs to take this into account. The leader selection procedure involves an extra procedure, Algorithm 4, for modifying the potential leader by considering opposite-side pedestrians and simultaneously choosing the passing side, under the condition that no more than 2 same-side pedestrians are closer to the door. The initial evaluation in the algorithm for determining whether there is any other pedestrian prior to me is based on $E(dis_{leader}, t_{wait})$. The algorithm will return either a leader or the selected passing side. If a leader is returned, the pedestrian will adopt the leader's passing side. Finally, the doorway behavior is decided according to Algorithm 2.

Algorithm 4: Leader modification considering opposite-side pedestrians and/or choosing the passing side (double door case).

```
if there is no opposite-side pedestrian prior to me then
    if I have an initial leader that is already a follower/door-opener, or no
    door is free then
        take the initial leader as the ultimate leader.
    else
        prefer the free door (if any) rather than the right-side door.
    end
else
    /* there is any opposite-side pedestrian prior to me      */
    if my initial leader is a follower/door-opener and I rank 3rd among
    same-side and opposite-side pedestrians then
        if there is a free door then
            choose the free side.
        else
            follow the pedestrian prior to me, which should be from the
            opposite side.
        end
    else
        if I have an initial leader then
            take the initial leader as the ultimate leader.
        else
            take the pedestrian prior to me as the ultimate leader, which
            should be on the opposite side.
        end
    end
end
```

4.4 Other Behaviors

Our door behavior synthesis framework is readily extensible to support scenarios involving additional, possibly conflicting, motor tasks, such as a pedestrian carrying a box, or pushing a baby stroller, as well as additional doorway ordering behavior scenarios, such as overtaking someone and offering to open the door for the box carrier to pass first, or multiple pedestrians walking shoulder to shoulder as friends towards a door and casually deciding their doorway passing order.

4.4.1 Pedestrians Manipulating Large Objects

Carrying a big box or pushing a baby stroller while trying to open a door and pass through the doorway is not a simple problem. However, since our framework largely separates the behavior and motor layers, the task boils down to supplementing the set of skills in the motor layer. It is rather easy to support the motor skill of carrying a box. Consider the skill of manipulating a baby stroller. This requires a motion controller. We have crafted a procedural motion model with control functions to work properly in conjunction with our pose-to-pose procedural animation system. Additionally, it becomes necessary to expand the doorway regions to support the new doorway behaviors (see Section 4.4.3).

4.4.2 Connected Doorway Behaviors

Interesting scenarios that involve personal connections between pedestrians can be readily simulated by augmenting the doorway behavior module. The modifications relate to deciding doorway behavior, while the perception and the leader selection procedure remain unchanged. In scenarios of a follower overtaking a leader to offer door-opening assistance, the follower will walk at an increased speed towards the door instead of towards the leader, ultimately resulting in the follower becoming the leader by virtue of dynamic doorway ordering. In scenarios of friends designated as such and walking together, shoulder to shoulder, no matter who is the leader or follower, both will walk towards the door while adjusting their speed and proximity to accommodate one another, resulting in a casual passing order.

4.4.3 Measurements of the Door Waiting Regions

Table 4.1 specifies the measurements of the waiting regions for various scenarios.

	PullWR	PushWR	AgentWR
Simple Door	3.3	2.5	1.5
Revolving Door	3.0	3.0	1.5
Double Door	3.5	2.5	1.5
Carry Stroller	3.3*1.2	N/A	1.5*1.2

Table 4.1: Measurements of the door waiting regions (WR) for different situations (in meters). The waiting regions for the double door are centered at the middle of the two sprung hinged doors. The N/A is due to the fact that in the stroller-pushing scenario, we used the sprung hinged door and only simulated the door pulling-side case.

CHAPTER 5

Social Factors and the Decision Model

Different people will have differing behaviors in the same situation. We adopt a probabilistic model for decision making in the context of door-holding behaviors, which is inspired by the work of Yu and Terzopoulos (2007). In particular, we build a Bayesian network to decide door-holding behaviors (Figure 5.1 with Table 5.1). Unlike the prior work, our Bayesian network makes its decisions based on the probability rather than a utility. From the modeling and computational perspective, it gives equally sound results. There are four random variables or factors in the network. The value of each variable is between 0.0 and 1.0, with *Effort* and *Care* evaluated dynamically at runtime, while *Kindness* and *Rush* are statically assigned from the start. They are introduced as follows:

5.1 Social Factors

Effort: Based on the study of Santamaria and Rosenbaum (2011), the door-holding problem is regarded as a minimum shared-effort model. As a person holds the door open for someone else, others will tend to hold the door open

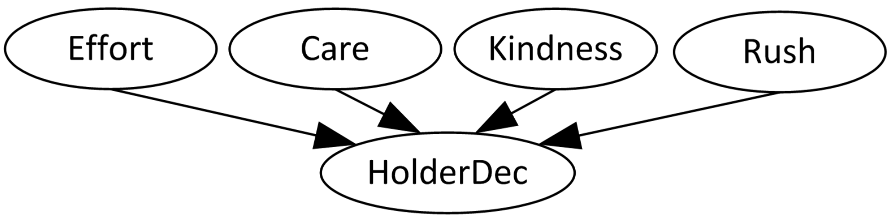


Figure 5.1: Bayesian network for door-holder behavior decision making.

Rush=t						Rush=f					
Effort	Care	Kindness	HOL	HOF	NH	Effort	Care	Kindness	HOL	HOF	NH
t	t	t	0.7	0.1	0.2	t	t	t	0.2	0.8	0.0
t	t	f	0.6	0.1	0.3	t	t	f	0.7	0.2	0.1
t	f	t	0.7	0.0	0.3	t	f	t	0.9	0.1	0.0
t	f	f	0.5	0.0	0.5	t	f	f	0.6	0.1	0.3
f	t	t	0.1	0.0	0.9	f	t	t	0.2	0.1	0.7
f	t	f	0.1	0.0	0.9	f	t	f	0.5	0.0	0.5
f	f	t	0.1	0.0	0.9	f	f	t	0.3	0.0	0.7
f	f	f	0.0	0.0	1.0	f	f	f	0.1	0.0	0.9

Table 5.1: Conditional Probability Table (CPT) for the door-holder behavior decision network. (Each value in the CPTs is chosen by experience under each condition, a combination of binary settings for all 4 factors.)

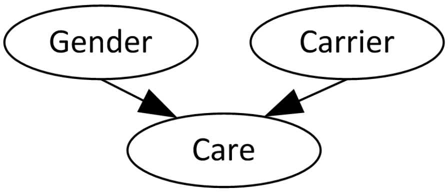


Figure 5.2: Bayesian network for determining the value of the *Care* variable.

Gender	Carrier	Care
F	t	1.0
F	f	1.0
M	t	1.0
M	f	0.0

Table 5.2: Conditional Probability Table (CPT) for the network of variable *Care*.

for this person at other times. Therefore the total energy any person expends in passing through doors will be minimized. In their study, these researchers found that the closer the distance from the follower to the holder, the larger the probability of holding the door, regardless of whether there are 1 or 2 followers. Thus, we incorporate a distance factor to evaluate the value of the *Effort* factor. In summary, the factor *Effort* describes how beneficial it is to hold the door for others in the current situation. The *Effort* factor is dynamically evaluated as a function of the current distance d to the follower. For $d < 1.0m$, the function value is 1.0; for $d > 4.0m$, the value is 0.0; for $1.0m \leq d \leq 4.0m$, the function value is linearly interpolated between 1.0 and 0.0.

Care: We introduce a *Care* variable to encode how much the follower is assessed to require assistance. It is accessed from two other factors: *Gender* and *Carrier* (Figure 5.2 with Table 5.2). Gender differences are significant in doorway etiquette. In cultures of most western countries, well-mannered men would, in most cases, prefer to hold open the door for women, a phenomenon which was studied by Webster et al. (2007). Therefore, we include a *Gender* factor that takes into account the gender of the follower. In most friendly cultures, people would offer opening door for those who need assistance; e.g., when carrying an object. Therefore we include a *Carrier* factor that indicates whether or not the follower is carrying an object.

Personality: Personality (Pervin, 1996) determines the inner psychological differences of people, and it can result in completely different behaviors in the same situation. We include a *Kindness* factor to determine whether a character is willing to hold the door for others.

Hurriedness: If people are in a rush, they will tend to hold the door open less frequently, since door-holding takes extra time and causes delay, which can be critical in hurried situations such as building evacuations. The *Rush* factor decides how hurried the door-holder is.

Values for *Effort* and *Gender* will be acquired from the state of the environment, while *Kindness* and *Rush* will be initially assigned to each person.

5.2 Decision Model for Holding Door Behaviors

Given the above variables, the structure of the network is shown in Figure 5.1. Its output comprises the 3 different holding behaviors (actions), which include holding the door for others to pass first (HOF), holding the door for others to pass later (HOL), and not holding the door (NH). Given all the values for the factors of the current state, the decision network will calculate the probabilities (utilities) of all the actions, and the action with maximal probability will be chosen in accordance with the Maximum Expected Utility Principle (MEU) (see (Russell and Norvig, 2003), Chapter 16.5). The Conditional Probability Table (CPT) (Table 5.1) for setting the network is designed based on our experience, which reflects human decision making under different conditions.

5.3 Variability in Doorway Behavior

The *Rush* factor is also used to vary the speed of pedestrians in the doorway. A rushed pedestrian is able to walk faster and possibly surpass agents not in a rush, which results in quicker passing. Additionally, another important social trait that we added is for the follower to speed up if it is too far from the door-holder.

CHAPTER 6

Motion Generation

6.1 Door Control Functions

Door opening is naturally effected by humans applying force at a relatively constant point on the door. We have created a procedural door controller which is not physically-based, but which provides flexible and stable human interaction. We build two functions for human interaction. The first function is the door rotation angle, determined by the relative (pos_{rel}) and absolute (pos_{abs}) positions of the hand on the door. It is employed in the door-holding phase when the character needs to move their body while holding the door, so it is better to let the hand position determine the angle of the door: $f_{ang}(pos_{abs}, pos_{rel})$. The second function is to obtain the absolute position of the hand given the door rotation angle (ang_{curr}) and the relative position (pos_{rel}) of the hand on the door: $f_{pos_{abs}}(ang_{curr}, pos_{rel})$. It is employed in the opening phase, when the door must be opened gradually to a certain angle, while the hand is fixed at a relatively constant position on the door.

The door has automatic opening and closing motion procedures that can be triggered, and it has a bouncing effect, which occurs when the person releases the door—he/she will apply extra force in order to conserve momentum, which results in the door opening to a larger angle and then closing back. We create a function to mimic this effect: $f_{bounce}(ang_{bounce}, p_{stop})$, where ang_{bounce} is the extra angle the door will open, and p_{stop}, when provided, the door will stop closing if any part of the door board hits this position, and it is useful for the character to hold the door after it bounces back.

6.2 Attention-Driven Head/Eye Motion

Visual attention, as indicated by the head and eye movements, is crucial to the realism of synthetic human characters, especially those movements that have behavioral significance. Human visual attention is closely connected to perception and, due to the limited field of view and foveal acuity of the eyes, it determines what visual information people can sense in their environment. Active visual attention control is responsible for acquiring useful information for completing a task. Given the holding door task, we add head/eye motions in the following situations:

1. Characters need to pay attention to their manipulating side hand to locate the handle of the door in order to reach it with their hand.

2. If they are considerate, they will voluntarily rotate their heads to look back and check if another character is following and, if so, hold the door. While the follower is passing through the doorway, the holder will look at the follower.

6.3 Pose-to-Pose Procedural Animation

A full-body pose is comprised by the joint angles of some critical joints. A basic joint pose is a transformation configuration of a joint: $P = (p, \{r_q, (\theta_x, \theta_y, \theta_z)\})$, where P is a pose, p is position, and the rotational component can be represented either by a quaternion r_q or by rotation angles θ_x, θ_y, θ_z, around the $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ axes of the world coordinate system. A full-body pose is composed of some joint poses: $P_f = (P_{rt}, P_{lh}, P_{rh}, P_{sp})$, where P_{rt}, P_{lh}, P_{rh}, P_{sp} are the joint poses for the root, left hand, right hand, and spine. Given a current pose and a target pose, the procedural animation system will generate motions $G(P_f^c, P_f^t)$, where P_f^c and P_f^t are the current and target full-body poses. The root transformation will trigger the step-based procedural locomotion (described in the next section). Animating the spine requires gradually rotating the root joint until the target rotation is achieved.

The arm motion animation takes into account several constraints and is controlled by IK. For fast performance, we have adopted an analytical IK model (Kallmann, 2008) comprising 7-DOFs for the human linkage. Given a target pose, the arm end-effector will move gradually towards the target along the shortest path, until the arm reaches the target or the length limit. Since the body is possibly being moved by the root transformation, the current IK target configuration must be updated according to the updated setting of the shoulder frame, which is the coordinate system upon which the IK is based. This will ensure that the arm always moves towards the target without incorrect backwards movement due to dramatic root transformation. Taking the right hand as an example, the current absolute position of the end effector is $p_{rh}^c = r_{rs}^c {p'}_{rh}^l + p_{rs}^c$, where p_{rh}^c is the current world position of hand, r_{rs}^c is the current world rotation of right shoulder joint, and ${p'}_{rh}^l$ is the last frame's relative position of the hand w.r.t. the shoulder frame. p_{rs}^c is the current world position of the shoulder. Then the hand's next world target position can be calculated as $p_{rh}^n = p_{rh}^c + \delta d$, where $\delta d = ||p_{rh}^t - p_{rh}^c|| s \delta t$, with p_{rh}^t as hand's target position, s as the hand speed, and δt as the time step. If the target position p_{rh}^n is not reachable, we use p_{rh}^c as the target position. This method is used for animating reaching. In some other situations, the hand must follow the track of an object, such as when the character pulls or pushes the door open. We add another hand control that can dynamically update the target pose of the hand between frames.

6.4 Step-Based Procedural Locomotion

Between two key poses, the character might undergo positional displacements of the root. Due to the highly constrained environment and the complexity of handling the door task, target root positions are defined for the characters to reach in each phase. An automatic step generation procedure will generate reasonable steps connecting the current to the target root position. The step-based procedural locomotion can generate full-body locomotion: $L(P_{rt}^{c}, P_{rt}^{t}, f, cs)$. f is the current foot configuration: $f = (P_{lf}, P_{rf}, sw, sd)$, where P_{lf} and P_{rf} are the current joint poses for the left and right foot, sw is the current swing foot side, sd indicates whether the current pose is standing or in the middle of walking; if standing, the first step will be generated as a special case. $cs = (sd, bk)$ denotes constraints on the target step, where sd indicates whether the last step is to a standing pose or just a pause in the middle of walking, and bk indicates whether the steps are generated as backwards steps, which, when set to true, the step generation procedure works basically the same way, but the root orientation keys will be reversed when synthesizing full-body locomotion.

The step generation works as follows (Figure 6.1(a)) in the case of current step is in the middle of walking (sd = false): The methods mainly decides the position of the feet. The foot orientation is decided by the orientation of the root after steps are generated. We calculate the target root direction $\mathbf{d}_{tgt} = ||\mathbf{p}_{rt}^{tgt} - \mathbf{p}_{rt}^{curr}||$, where $\mathbf{p}_{rt}^{curr}$ is the current root position and $\mathbf{p}_{rt}^{tgt}$ is the target root position. The right side direction is $\mathbf{d}_r = \text{rot}(-\pi/2, \mathbf{y})\mathbf{d}_{tgt}$, and the left side direction is $\mathbf{d}_l = -\mathbf{d}_r$. The next (right) step position is calculated and then constrained by C_{s_1} for different side-step distances:

$$\mathbf{p}_{rf_1} = \mathbf{p}_{rt}^{curr} + \alpha \mathbf{d}_{tgt} L_{step} + \mathbf{d}_r W_{step}, \tag{6.1}$$

$$\mathbf{p}_{rf_1} = \mathbf{p}_{lf_0} + ||\mathbf{p}_{rf_1} - \mathbf{p}_{lf_0}|| C_{s_1}, \tag{6.2}$$

where the parameter α is set to 1.5. We can map the first step in the target root direction, and then calculate the second (left) step constrained by C_{s_2} for same side step distance:

$$m_{rf_1} = (\mathbf{p}_{rf_1} - \mathbf{p}_{rt}^{curr}) \cdot \mathbf{d}_{tgt}, \tag{6.3}$$

$$\mathbf{p}_{lf_1} = \mathbf{p}_{rt}^{curr} + \mathbf{d}_{tgt}(L_{step} + m_{rf_1}) + \mathbf{d}_l W_{step}, \tag{6.4}$$

$$\mathbf{p}_{lf_1} = \mathbf{p}_{lf_0} + ||\mathbf{p}_{lf_1} - \mathbf{p}_{lf_0}||C_{s_2}. \tag{6.5}$$

Similarly, we map the second foot in the target root direction as m_{lf_1}. The next step will be calculated as

$$\mathbf{p}_{f_{next}} = \mathbf{p}_{rt}^{curr} + \mathbf{d}_{tgt}(L_{step} + m_{f_{prev}}) + \mathbf{d}_s W_{step}, \tag{6.6}$$

where $\mathbf{d}_s$ is chosen between $\mathbf{d}_r$ and $\mathbf{d}_l$. If $L_{step} + m_{last} > L_{tgt}$, where $L_{tgt} = \text{len}(\mathbf{p}_{rt}^{tgt} - \mathbf{p}_{rt}^{curr})$, we generate the last step as

$$\mathbf{d}_1 = ||\mathbf{p}_{rt}^{tgt} - \mathbf{p}_{f_{last}}||, \tag{6.7}$$

$$l_1 = W_{step}/\sin(\text{ang}(\mathbf{d}_1, \mathbf{d}_{tgt})), \tag{6.8}$$

$$\mathbf{p}_{f_{end}} = \mathbf{p}_{rt}^{tgt} + \mathbf{d}_1 l_1. \tag{6.9}$$

We must add an extra turning support step if the line between p_{rt}^{curr} and p_{rt}^{tgt} goes behind the current supporting foot (Figure 6.1(c)). If the current step is standing, the first step is generated closer to the initial foot position (Figure 6.1(d)) by setting α to 1.0 in (6.1). If the last step is required to be standing, and given the normalized right direction of the target facing direction as $\mathbf{d}'_r$, the last two foot positions are re-generated (Figure 6.1(d)):

$$\mathbf{p}_{lf_{end}} = \mathbf{p}_{rt}^{tgt} - \mathbf{d}'_r W_{step}, \tag{6.10}$$

$$\mathbf{p}_{rf_{end}} = \mathbf{p}_{rt}^{tgt} + \mathbf{d}'_r W_{step}. \tag{6.11}$$

Based on the generated steps, procedural full-body locomotion will be synthesized. We create a walking motion profile with feet, root, and hand keys. The root will have keys in the two-foot support phase, and its orientation will decide the orientation of the adjacent foot keys. Both hands have keys in the two-foot support phase, and have the front extreme key when the same side foot is at the back, and back extreme key when the same side foot is at the front. Finally, the root has keys in the middle of the single foot support phase, when the root reaches the highest height in the walking cycle, and it has keys in the two-foot support phase when the root reaches the lowest height. While animating the motion, the feet, root, and hands are synchronized for coordinated full-body locomotion.

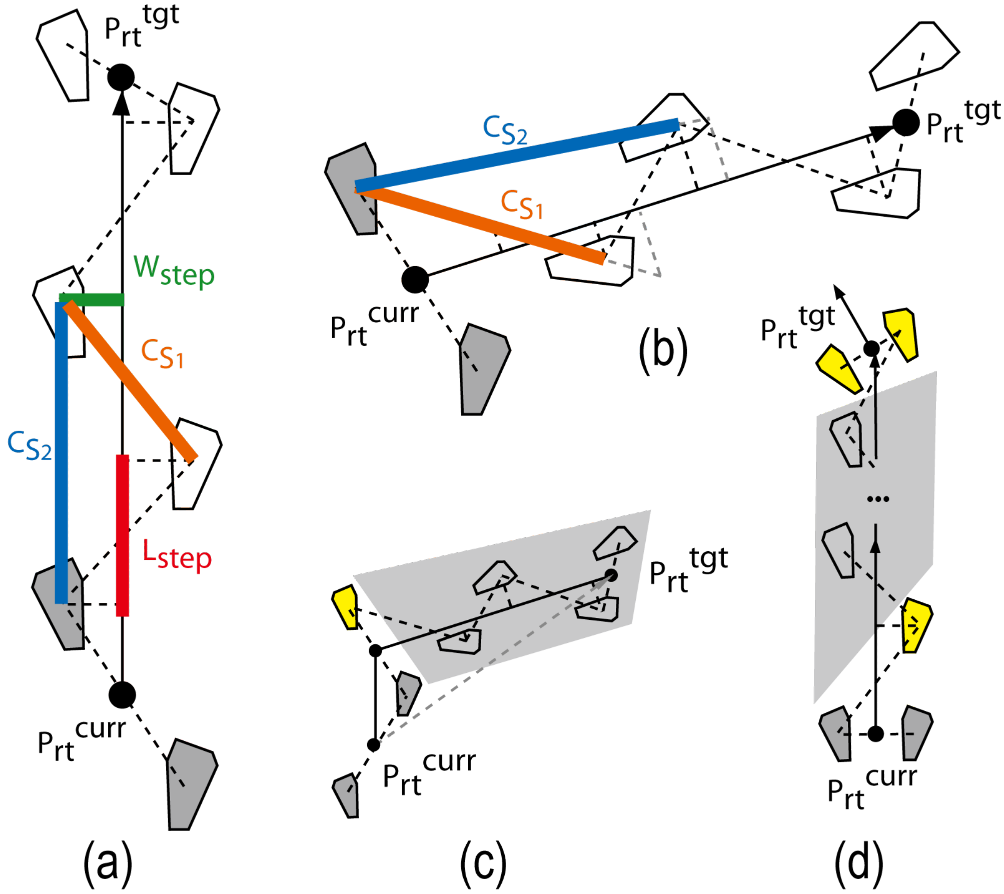


Figure 6.1: Step generation. The gray footprints indicate the initial foot positions and the white and yellow footprints are the generated steps (a) A step sequence is generated between p_{rt}^{curr} and p_{rt}^{tgt}. L_{step} and W_{step} are values for generating steps, and C_{S_1} and C_{S_2} are constraints for distance between the left and right foot positions and the distance one foot can step, respectively. (b) When a large turn is required, constraints C_{S_1} and C_{S_2} will efficiently constrain the foot positions to valid configurations. (c) An auxiliary step is generated if the target direction is behind the current support foot. (d) From a standing pose, the first step will be generated to close the distance; the target step will be regenerated with the target facing direction, if it is required to be a standing pose.

CHAPTER 7

Results

Our efficient model is capable of real-time performance. Our simulator runs on a 3.33GHz Intel Xeon computer with 12 GB memory and an NVIDIA Quadro FX 4800 graphics card. In a 4-door, 16-pedestrian scenario, with the simple door type, the performance is about 50 fps including rendering. Integrated with a complex building model shown in Figure 7.1, the performance is about 25 fps, as the geometric complexity of the models dominates the simulation time. Next, we will present results of 3 different types of doors with the sprung door serving as the baseline case.

7.1 Sprung Hinged Door Example

Our system can synthesize multiple characters passing through the door in a continuous manner, while generating cooperative door manipulation and holding behaviors between them. It can generate same-way door-passing behaviors, both from the pull and push sides of the door (Figure 7.2 and Figure 7.3), and opposing-way door-passing behaviors (Figure 7.5). Different holding behaviors are generated and paired with corresponding following behaviors, and they are well-synchronized both spatially and temporarily.

It is interesting to observe that if two groups of pedestrians arrive from opposite directions at nearly the same time, an alternating passage pattern will result, where the groups will alternate in passing several pedestrians at a time (Fig-

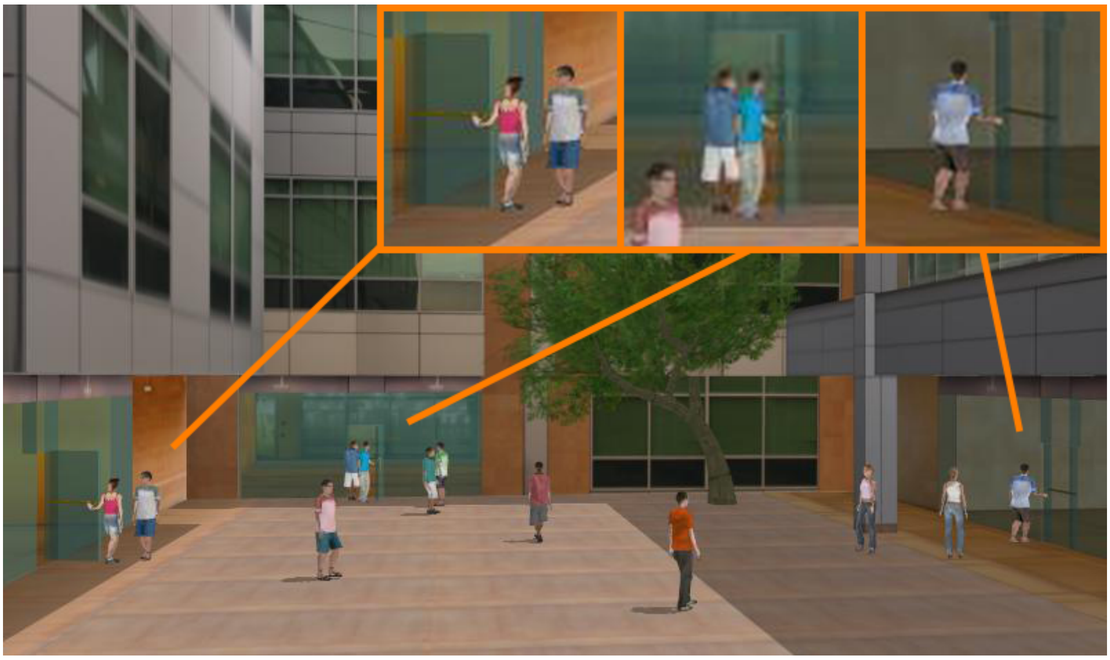
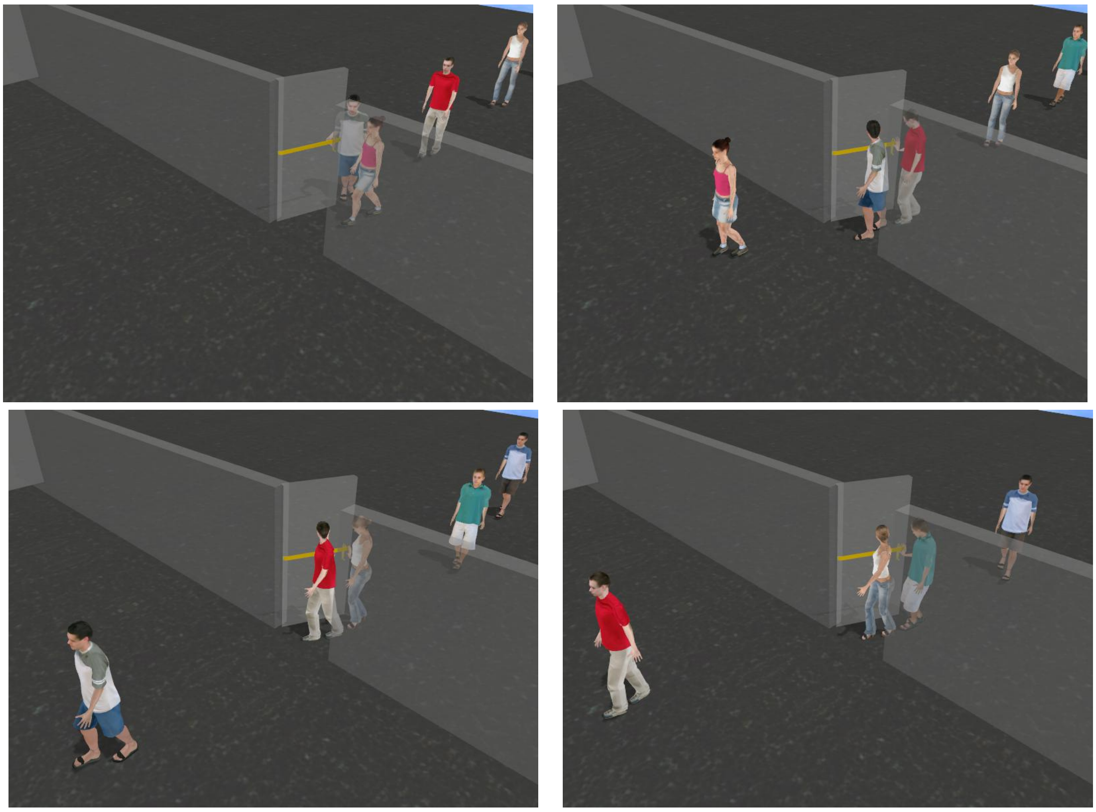


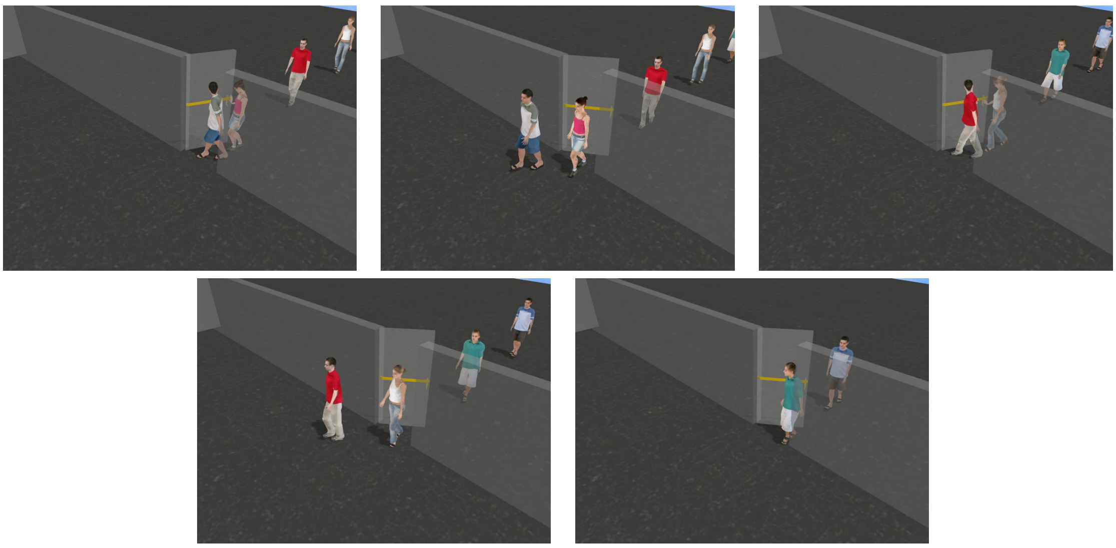
Figure 7.1: A multiple door scenario with 4 doors and 16 pedestrians.

ure 7.5(b)). However, if the group from one side of the door arrives significantly earlier than that from the other, the later group will wait until the earlier one has finished passing (Figure 7.5(a)), which actually happens in real life.

Our simulated humans perform reasonable behaviors based on the current state of the world. If the follower is too far from the door-holder, the door will not be held. If the follower is a lady, the holder is more likely to hold the door and even to let the follower pass first. However, a hurried pedestrian will exhibit behaviors through the doorway that are consistent to a hurried state of mind, walking quickly, and geing less apt to hold the door open to allow a follower to pass first. When a group of pedestrians approaches the door, the result will usually be that the hurried ones will pass through the doorway earlier than unhurried ones (Figure 7.4), as expected. Given pedestrians that are generally polite and unhurried, a sequential door-holding pattern emerges and efficient flow results (Figure 7.2(a)). By contrast, pedestrians that are not kind and/or in hurry result in a flow that is continually interrupted as some pedestrians neglect to hold the



(a)



(b)

Figure 7.2: Sprung hinged door simulations (pulling side).

Figure 7.3: Sprung hinged door simulation (pushing side). (a) Continuous passing sequence generated by generally kind and unhurried pedestrians. (b) Frequently interrupted passing sequence by less kind but hurried pedestrians.

(a) (b) (c)

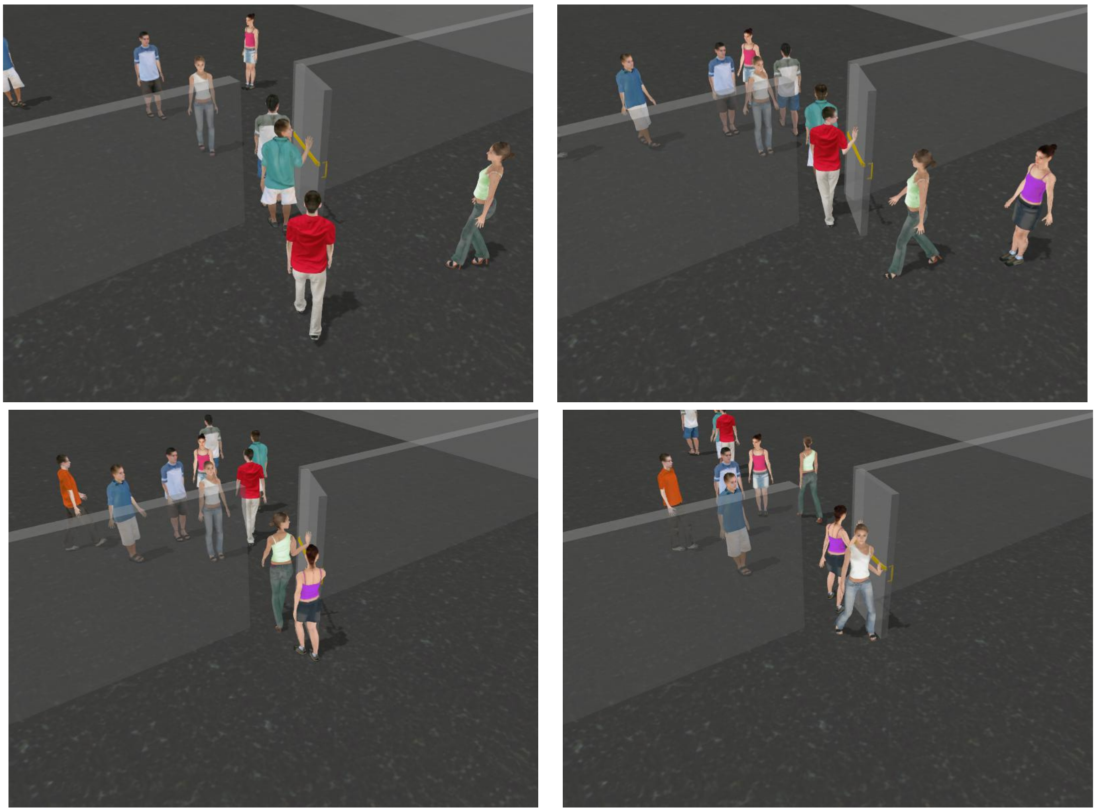
Figure 7.4: With similar initial starting positions (a), whether a pedestrian (circled in yellow) is hurried (b) or not (c) will yield different passing orders.

door for others (Figure 7.2(b)).

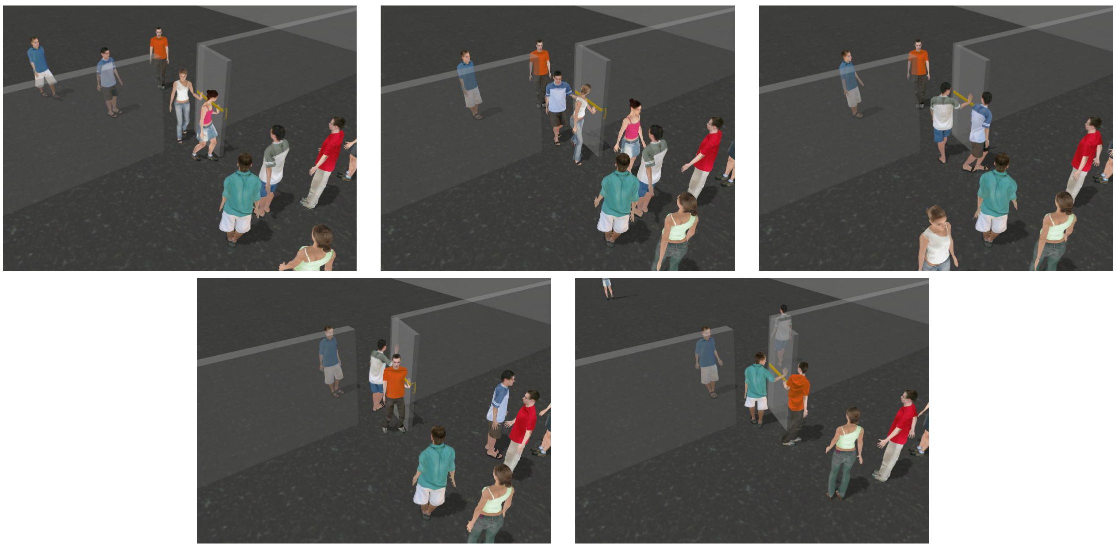
7.2 Other Door Types

7.2.1 Revolving Door Example

In the revolving door case, pedestrians approaching from the same side can form a continuous flow (Figure 7.6(a)). With pedestrians from opposite sides, the simultaneous flow of both sides is ensured (Figure 7.6(b)). Furthermore, based on when they arrive at the door, pedestrians can dynamically choose to speed up to catch the opening door or slow down and wait for the next opening.



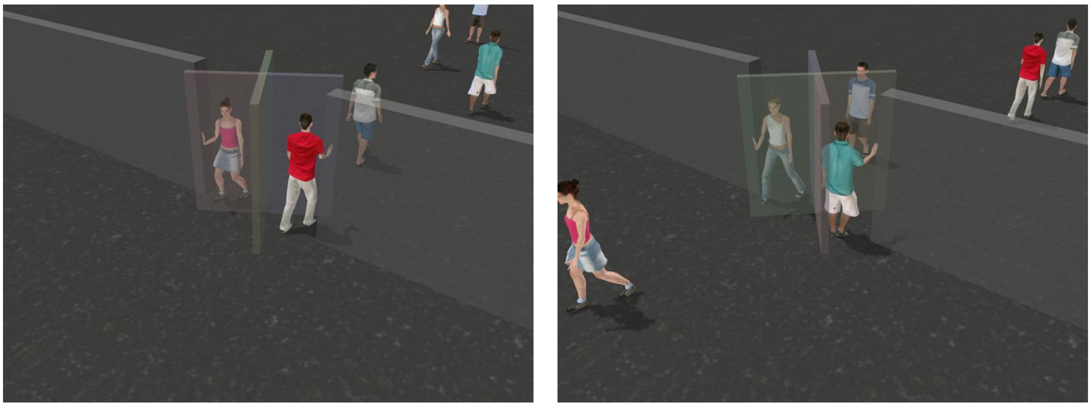
(a)



(b)

Figure 7.5: Sprung hinged door simulations (opposite sides). (a) The significant earlier arrival of one group results in its complete passage while the other group waits. (b) A similar arrival of two groups results in the alternate passing of a few pedestrians at a time from each side.

(a)



(b)

Figure 7.6: Revolving door simulations.

7.2.2 Double Door Example

In the double door example, both features of preferring the right-hand door and the dynamic preference of the holder's side can be observed. In normal situations, pedestrians will choose the right-hand door and a two-way flow will result (Figure 7.7). If one door is in relatively heavy use while the other is clear, then the later arrivers will start using the less congested door, even if it is not the right-hand one. Moreover, if there is only one holder of one door with the other door clear, the next follower will select the holders side regardless of whether it is the left or right side (Figure 7.8).

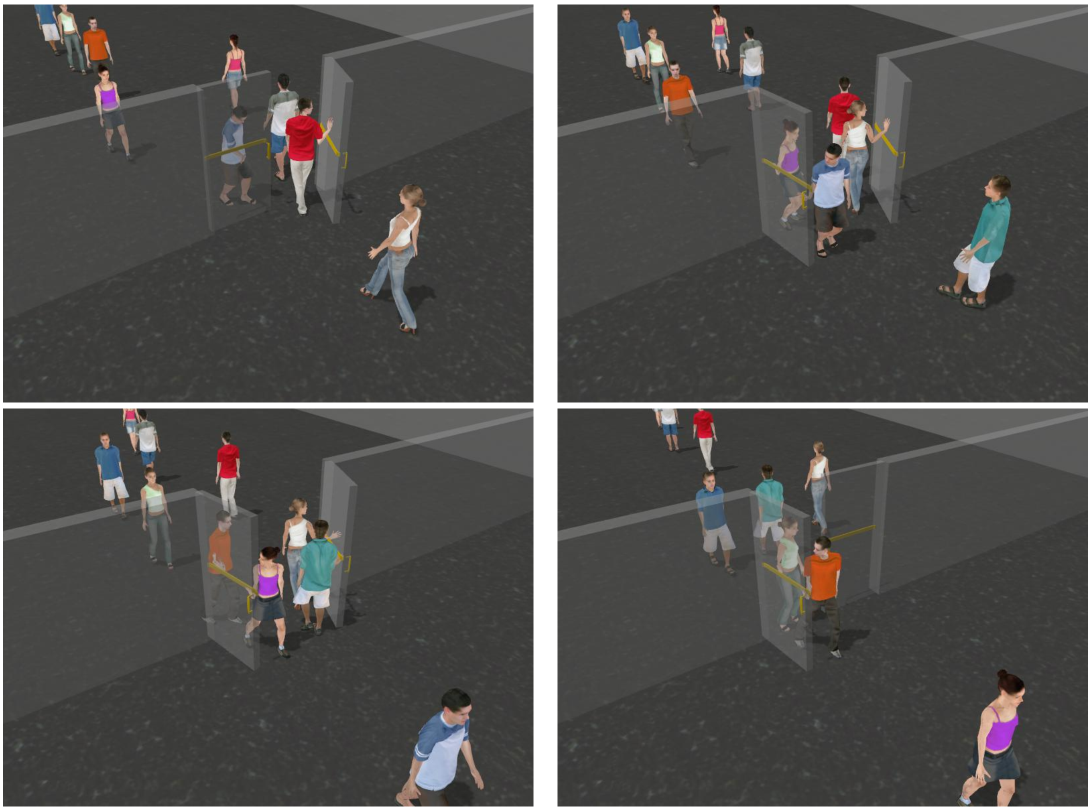


Figure 7.7: Double door simulations (right side passing preference).

7.3 Additional Behavior Variability Examples

In scenarios of a pedestrian carrying a box or pushing a baby stroller, the follower of a carrier/pusher will try to overtake in order to open the door first and offer door-holding assistance (Figure 7.9). In scenarios of two pedestrian friends walking shoulder to shoulder, they do not exhibit leader/follower dominance while approaching the door, until they arrive at the proper region for interacting with the door, resulting in a casual passing order (Figure 7.10).

7.4 Multiple Door Scenario

We built a multiple-door scenario involving 4 sprung hinged doors and 16 pedestrians, as mentioned earlier. The 3D environment model is a reconstruction of

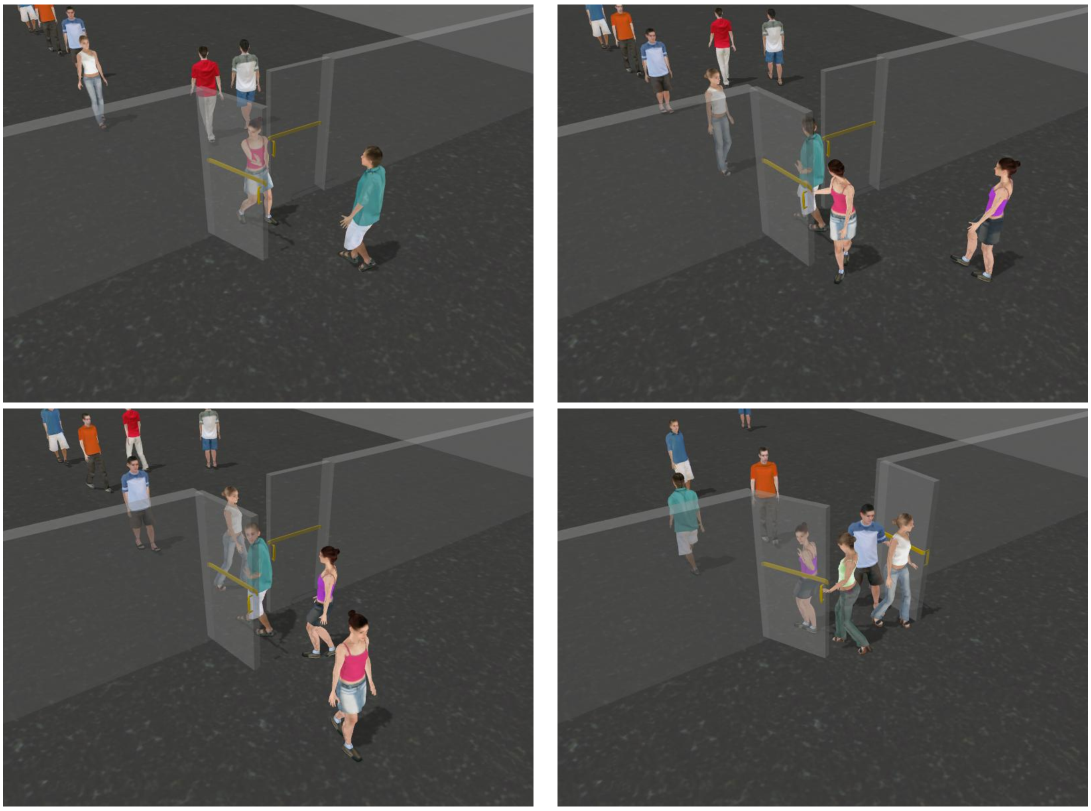


Figure 7.8: Double door simulations (dynamically selected passing side).

(a)

(b)

Figure 7.9: Simulations of a follower overtaking and opening the door for someone needing assistance: (a) baby stroller pusher scenario, (b) box carrier scenario.

(a)

(b)

Figure 7.10: Simulation of friends walking together, shoulder to shoulder, which casually results in different passsing orders ((a) vs (b)).

Winston Chung Hall at the University of California, Riverside. The required initial setting of the system is simple. We define the spatial information (position and orientation) of each door in text files, together with the locations and dimensions of obstacles (walls) in the environment and the initial and target positions of all pedestrians. As we run the system, the pedestrians will autonomously walk towards their destinations, while deciding whether to pass through a door and dynamically performing door manipulation/holding behaviors. After a pedestrian passes through a door, it can proceed to pass through a subsequent door (e.g., the pedestrian with the purple shirt in Figure 7.11), which is automatically determined by the pedestrian's initial position and final destination.

As we casually fly the virtual camera through the environment, we can observe a highly dynamic process as pedestrians approaching a door randomly come across other pedestrians, but they still competently figure out the passing order while showing concern for each other with a collaborative door manipulation strategy (note, e.g., the cutting-in behavior shown at the top of Figure 7.12). If we reinitialize pedestrians that arrive at their destinations back to their initial settings, the system can continue to run infinitely, continuously producing a variety of interactions around doorways in a non-repetitive fashion (bottom of Figure 7.12).

7.5 Crowded Scenarios

To evaluate the scalability of the system, we tested a scenario of 12 pedestrians approaching from the pulling side of the door (Figure 7.13(a)). The result reveals that all the pedestrians can pass through the door successfully. As current door handlers finish passing through the door, later-arriving pedestrians toward the back of the pack who are waiting their turns to proceed through the doorway will consistently advance closer to the door.

In a second test scenario, 14 pedestrians, approach the door from opposite sides as two groups of 7 (Figure 7.13(b)). The result is still plausible, as pedestrians evaluate the most likely next pedestrian to pass through the doorway, and systematically pass from the same side or from alternate sides.

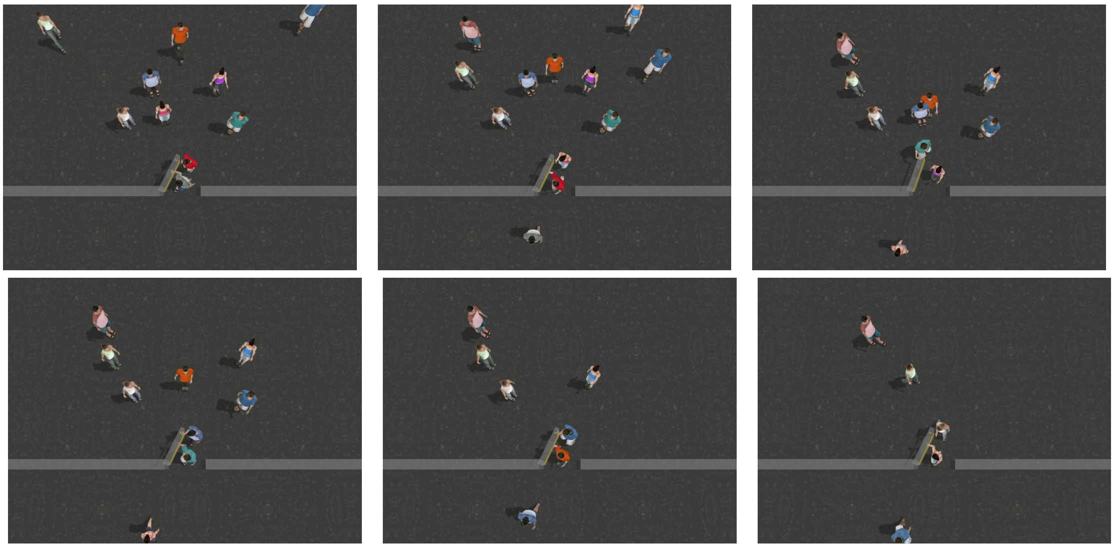
We succeeded in making the same-side passage more efficient by introducing a "second" follower role, which will pass after the follower. A pedestrian will commit to the second follower role if its leader is the follower and it enters the corresponding door waiting region. As a result, it can initiate the door interaction model and start moving to the convenient location for taking over the door, which will result in a more timely preparation for interacting with the door. Also during door interaction, we enabled the follower to start the door-reaching motion (*ReachEdge*) once the door-holder starts the door holding motion (*HoldDoor*), which also results in the follower appearing to have anticipation and performing the appropriate motions earlier. Finally, we gave the pedestrians a preference to hold doors for others to pass later. These changes resulted in a more efficient passage of the aforementioned scenario of 12 pedestrians approaching from the

Figure 7.11: Multiple door scenario snapshots.

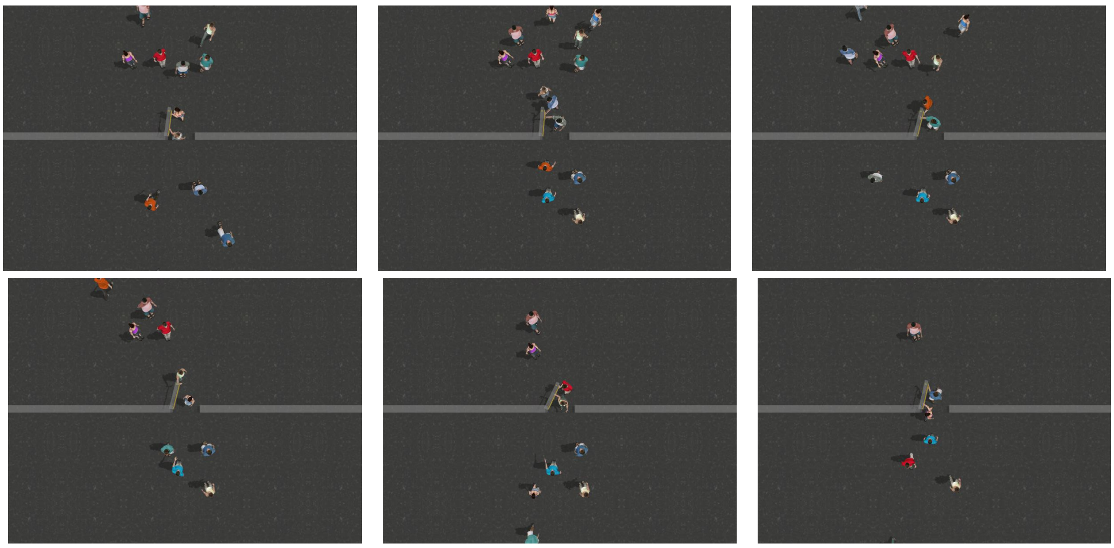
Figure 7.12: Multiple door scenario snapshots (continued).

pulling side of the door (Figure 7.14).

Although our approach seems reasonably scalable, there are some issues with the generated results. Due to the limited agility of our pedestrian locomotion and steering system, the pedestrians cannot squeeze too close to each other and demonstrate natural behavior in very tight situations. In the opposing-side doorway passing scenario, a few collisions can occur between pedestrians who finish passing through the doorway and those waiting on the opposite side. In fact, in highly crowded scenarios, waiting pedestrians need to have the ability to yield way to those coming through the doorway in front of them, perhaps by stepping to the side or even backward. This will require a more advanced pedestrian locomotion/steering system.



(a)



(b)

Figure 7.13: Crowded scenarios. (a) 12 pedestrians approach the pull side of the door. (b) 14 pedestrians approach the door from opposite sides in two groups of 7.

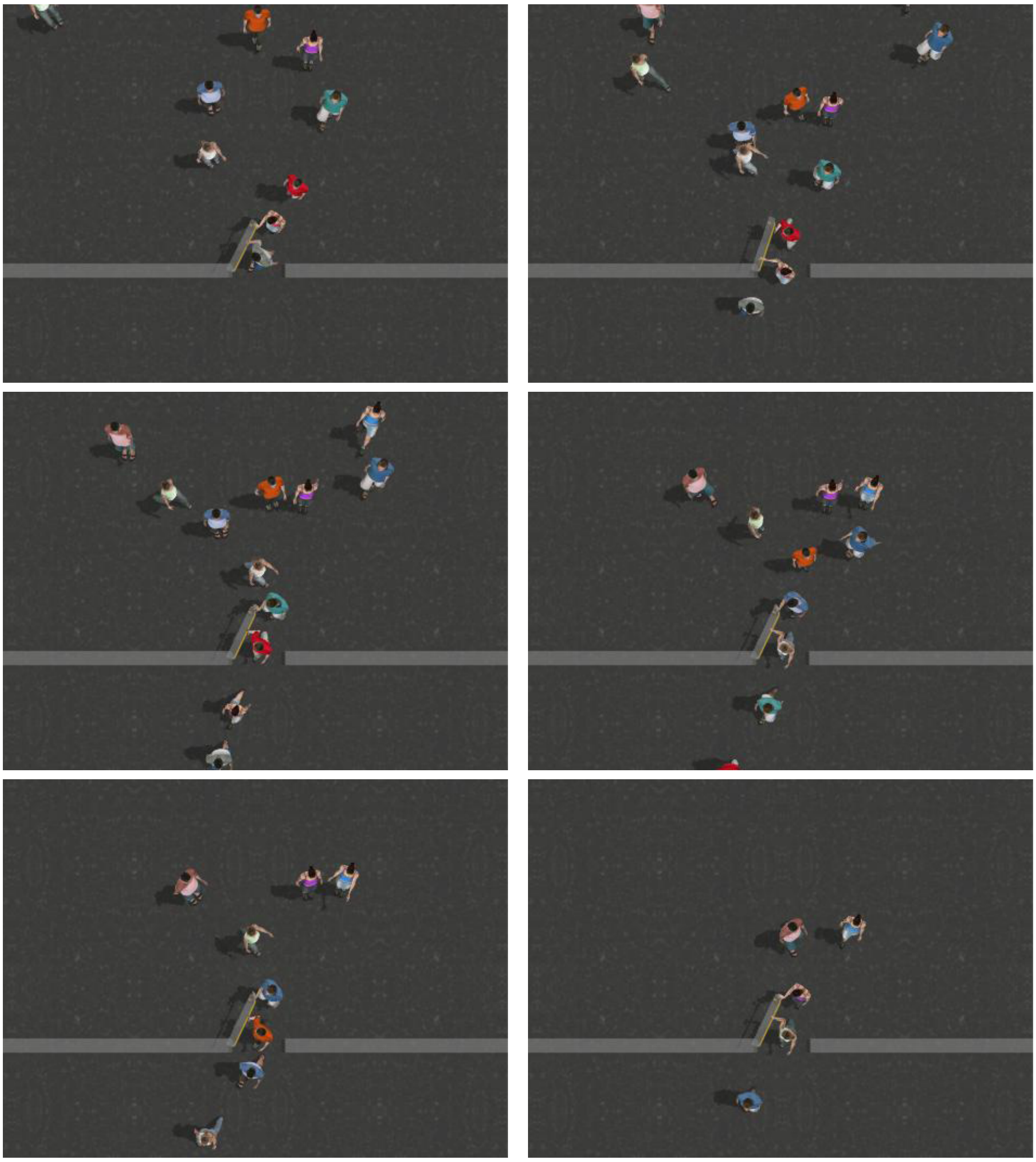


Figure 7.14: Crowded scenarios (followers keep closer): 12 pedestrians approach the pull side of the door.

CHAPTER 8

Conclusion

8.1 Summary

We have introduced a framework for multi-human simulation that is capable of synthesizing convincing door(way) etiquette at a social level. In particular, our simulator can synthesize cooperative door-holding behaviors that have not previously been the subject of study in computer animation. Our general framework can support the simulation of different types of doors, including sprung doors, revolving doors, and double doors. Our efficient model generates continuous, dynamic, and diverse results in real time, making it practical to create nontrivial multi-door and multi-character scenarios.

8.2 Limitations and Future Work

Density of People: Our system was designed to simulate uncrowded scenarios, say 5 to 6 pedestrians waiting simultaneously at each side of a sprung hinged door, and similarly for the revolving door and double door cases. These conditions elicit social behaviors that exhibit more respect to others, and a variety of cooperative actions. In scenarios involving many more people, the doorway will become congested and individuals will tend to overlook etiquette in favor of passing through the doorway as quickly as possible. We have tested our simulator with two extreme cases of 12 same-side and 14 opposing-side pedestrians attempting to pass at the same time. These denser scenarios sometimes lead to deadlocks that our current steering system cannot resolve. Indeed, some pedestrians may have to step backwards in order to yield to others coming through, which would require a more precise, flexible, and adaptive locomotion and steering system.

Alternative Motion Generation Models: Our procedural motion generation model can achieve adequate control for the tasks at hand, but it cannot adequately match the naturalness that data-driven motion synthesis methods are capable of. The use of data-driven methods (Wampler et al., 2010; Kwon et al., 2008; Shum et al., 2008) or physics-based models (Coros et al., 2009) in this highly constrained and dynamic environment is a rich topic for future work. More work on motion planning (Bai et al., 2012; Yamane et al., 2004) for this purpose is also worthy of consideration. Additional motion repertories can be incorporated in the motion generation phase in order to augment the diversity of the motion repertoire, or to define new behaviors at the decision layer from which to select.

Coordination of Decisions and Motions: Another interesting topic would be the automatic coordination between decisions and motions. An example is how to determine whether it is too late to change one's chosen follower and subsequently alter one's door holding behaviors. We have defined the critical motion phase to afford our system responsiveness to such changes of situation. However, it would be interesting to have such a method—especially given a more dynamic and powerful motion method with a larger repertoire of motions—that can dynamically analyze the current state of motion and identify whether it is too late to change a decision. Tackling this question would also be of great benefit in simulating other high-level human behaviors that require coordination between perception, decision, and action.

Training of the Bayesian Network: On the decision level, it will be preferable to acquire real-world data with which to train our Bayesian network model, or to train it using analyzed data from user evaluations of simulation results.

Social Groups: We demonstrated simulation results for a pair of friends passing though doors together. An interesting generalization would be to include additional social connections between pedestrians, such as pedestrians forming small groups whose cooperative behaviors transcend egocentric considerations. In such scenarios, one polite member of the group may choose to pass in front of the others and hold the door open for all of them. Meanwhile, other pedestrians would recognize them as a group and refrain from interfering with their collective progress. Furthermore, gestures could be incorporated to signal and reinforce these and related behaviors.

Similar Social Interactions: Other social interactions with supporting motor actions are similar in nature to door(way) etiquette, in that they involve spatial and temporal coordination of movement among multiple people, perception and assessment of the current situation, and decision making to initiate motor actions with the proper timing. An example is choosing a seat: As a group of people simultaneously enter a space with seats, they keep moving as they decide which seat to take while being aware of and considerate to others. Another example is passing bread around the table during dinner, which requires dynamically assessing whether one wants to handle the bread basket first or allow someone else to take the initiative, while continuously performing other social interactions between multiple people. The simulation of these social situations can be addressed by an approach similar to that presented in this thesis.

APPENDIX A

Step-Based Locomotion and Steering System

Two important aspects of pedestrian simulation is locomotion and steering. Locomotion is a motor control problem that aims to produce basic human movement skills, such as walking and running. By contrast, steering deals with the control of locomotion with the aim of navigating to targets while avoiding collisions. Locomotion and steering are usually addressed independently in the human animation literature, but their integration becomes crucial in our application to autonomous pedestrians, which is a nontrivial task that requires the proper coordination and adjustment of locomotion and steering.

A.1 A Data-Driven Step-Based Locomotion System

The locomotion control problem has been extensively studied in the literature. In general, there are three main approaches (Multon et al., 1999). The first is procedural animation, which generates locomotion by following some predefined routines. This method requires light computation, but it cannot reflect the naturalness of motion. The second is physical simulation, which ensures the physical plausibility of motion, but it is computationally expensive and also hard to control. The third, and the one upon which our method relies, is the data-driven approach. We built a motion database and generated new motions from it. This method maintains the naturalness of the motion and has fast performance. Its drawback is that it is not dynamic and not reactive to perturbations in the environment.

A.1.1 Step Space Parameterizations

The "step space" (van Basten et al., 2010) is a recent work that parameterizes walking motions into individual steps with related parameters. The benefit of employing a step space comes from two characteristics. First, one is able to search for a sequence of step clips that have consecutive features. Second, when one is concatenating two consecutive steps, one can align them according to some step space metrics that can mostly reduce foot mismatches. Both characteristics working together largely reduce the artifacts that most other data-driven methods suffer from, such as foot skating, and it only requires a little blending between the switching points. For these reasons, we parameterize the walking motions of our pedestrians in the step space.

A.1.2 Control Interface

However, varying the step space parameters is not an intuitive way of providing locomotion control commands; consequently, it is not a reasonable interface to be controlled directly by high-level character AI. In our application, it is unnecessary to search for a whole sequence of steps to reach a global goal, such as a distant target. It is more practical to have a short sequence of steps corresponding to an short intentional motion period, such as "I want to make a sharp turn now". In contrast, type/speed/angle control (Park et al., 2002) is a reasonable interface to be accessed by AI, since it maintains a human-oriented global view of locomotion control.

Inspired by these considerations, we build an intermediate module between the AI and step animation system, which converts the inputs of type/speed/angle to step-space-based animations. This architecture ensures the maintenance of natural global control and the naturalness inherent in the motion data.

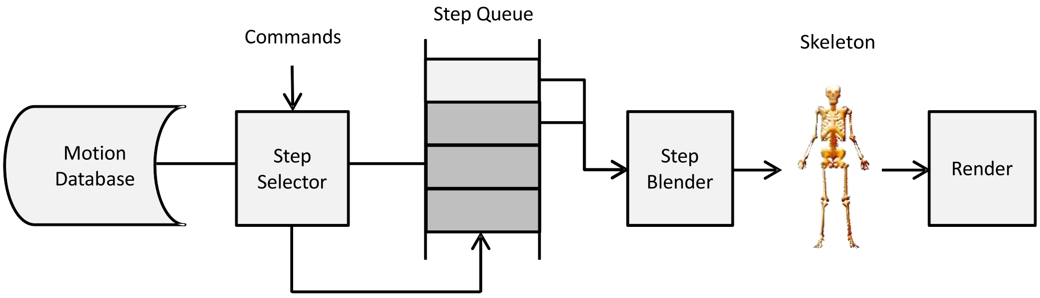


Figure A.1: System overview and main components.

A.1.3 System Overview

The animation system is constructed as shown in Figure A.1. We build a motion database that contains walking at different speeds, turning at different angles, and transitions between walking and standing.

The full-body motion data is organized as motion clips of sequences of steps associated with walking at various different speeds and turning angles. A step queue stores the steps that are to be performed. After a step is completed, it is dequeued. When only one steps remain in the queue, a step selector will permit input of a walking speed/angle command. It will then select from the database a new motion clip that best satisfies the command and enqueue the clip's effective consecutive steps (usually 1 to 3 steps) and their parameters. A step blender generates a full-body pose at each frame during the transition between clips by easing-out the current motion clip while easing-in the new motion clip. The blending occurs between the clips of the first two steps in the step queue, which is during a swing phase of the same leg in the two clips, and ensures that after completing the first step, the pedestrian will be in the start pose of the second step.

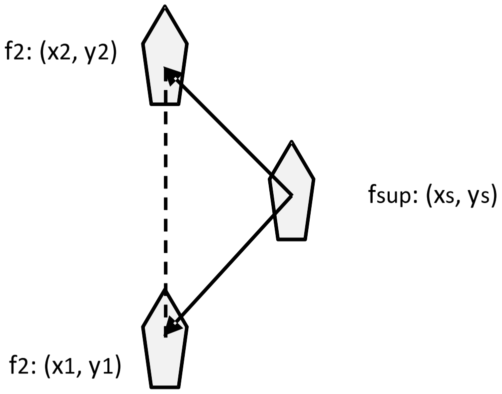
The locomotion controller must achieve a good balance between responsiveness and naturalness; that is, the pedestrian should respond as quickly and as naturally as possible to any new walking speed/angle command. Since we can have multiple steps in the step queue at any time, whenever a significant walking command is issued we allow the step queue to discard as many unexecuted steps as possible. Given our step concatenation scheme, the first 2 steps in the queue cannot be discarded, nor can a step sequence that corresponds to a large turning angle, as doing so can induce blending artifacts.

Motion Database Construction: We tried several ways to obtain motion data. Initially we tried CMU's motion capture database, but found their data is not well cleaned, especially suffering from artifacts in the foot motions. We also considered using Kinect technology to capture motion data, but it has similar problems of noise in data, and cleaning the data would cost much time. On the other hand, we found that 3dsMax provides the functionality to automatically generate walking animations given footstep placements. The resulting motions look good, albeit not fully natural. This is satisfactory, however, given our intended application in behavioral and social animation.

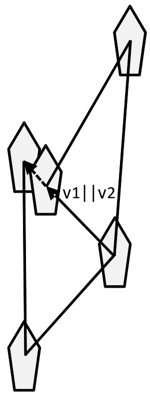
Since the motions involve different types, speeds, and angles, it is impossible to synthesize all possible motions in the continuous motion space; however, we discretize the motions to cover the motion space as evenly as possible.

The Step Space: We parameterize the step space in a similar manner to van Basten et al. (2010). Each step is a half walk cycle, starting from one double support phase to the next double support phase, and it is parameterized by 3 parameters (Figure A.2(a)): the support foot position and the two successive placements of the swing foot.

Concatenating and Blending Steps: Suppose that S_a and S_b are two selected consecutive steps. They are concatenated by first aligning the vector $v_1 = p(f_2) - p_{sup}$ (from the supporting foot position p_{sup} of S_a to the position of the second



(a)

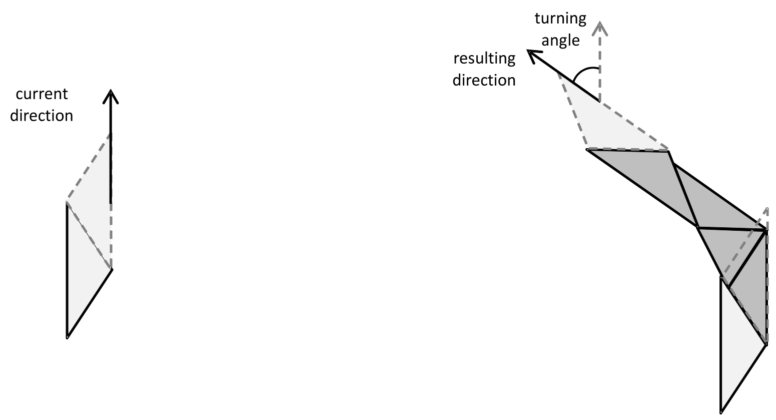


(b)

Figure A.2: (a) Step space parametrization. (b) Align steps by aligning v1 and v2.

placement $p(f_2)$ of S_a), with the vector $v_2 = (p_{sup} - p(f_1))$ (from the position of the first placement $p(f_1)$ of S_b to the supporting foot position p_{sup} of S_b) (Figure A.2(b)). Suppose $prev(S_b)$ is the previous step of S_b in its original recording. Then $prev(S_b)$ is on the same side as S_a. We ease-out from S_a and ease-in to $prev(S_b)$ by linear interpolation. The blending is done during the swing phase, so there is no footskating whatsoever.

Angle of Steps: The turning angle is defined over a sequence of steps. We design a method to evaluate the angle. Given the current step, we temporarily append one more normal straight step, called the "direction verification" step. Then the direction of the current step is defined as the vector along the swing foot of the appended step (Figure A.3(a)). The turning angle of a sequence of steps is calculated dynamically given the current step. The method is to append the sequence of steps to the current step, and also append one more normal straight step to obtain the resulting direction. Thus, the turning angle of the step sequence is the angle between the current direction and the resulting direction (Figure A.3(b)).



(a) (b)

Figure A.3: (a) Decide current direction. (b) Decide turning angle of a sequence of steps.

A.1.4 Results

In response to the input commands, the character can walk, turn, change speed, and stop naturally. But there are also some issues. Some delays are inevitable between command issuance and the character's actual motion responses, because the character must wait for a proper transition point to another motion. Furthermore, the turning angle is not accurate. This will turn to be a problem for characters in highly constrained environments. These issues cannot be resolved completely, but in the context of our autonomous pedestrians, we can decide how much improvement this system requires in order to satisfy the requirements of our application.

A.2 Rule-Based Steering System

Regarding steering, we mainly integrated the Shao and Terzopoulos (2007) work and the Steersuite software with PPR AI, which was developed in our lab at UCLA (Singh et al., 2011a).

A.2.1 PPR AI

PPR AI is composed of 3 phases, Planning, Prediction, and Reaction. In the planning phase, the algorithm finds collision free paths and local goals for the agent. This phase is further separated into 3 stages: long-term, mid-term, and short-term planning. Long-term planning first finds a global path to the target using the A^* algorithm. In the mid-term planning stage, a set of local target points are sampled along the global path. The sample point that is closest to the agent is then used as a secondary target, and a local path is found from the current location of the agent to this secondary target. Finally, during the short-term planning stage the farthest visible point on the local path is computed and used as the current local target. In the prediction phase, the character will sense the agents around it, and plan proper behaviors in advance to eliminate possible collisions in the future. Finally, in the reactive phase, the agent will use a local sensing model to identify the surrounding obstacles and make local behavior adjustments in order to produce the final steering command.

A.2.2 Shao's Reactive Rules

In contrast, Shao and Terzopoulos adopted quad-trees to model the environmental search space, and evaluated various search algorithms for path-finding. At the reactive level, they proposed 6 reactive rules for collision avoidance. In addition, they included a higher-level set of passageway rules that are specially designed for passageway scenarios. The result yields human-like queueing behaviors in restricted corridors, which optimizes the pedestrian throughput.

A.2.3 Integration

Combining and assimilating the ideas of both PPR AI and Shao's system, we implemented a steering system that is based on Steersuite and uses the planning part from PPR AI plus reactive rules similar to Shao's system. We also found that, for different environments, open space versus passageways, the best results are generated using different steering rule sets, and our pedestrians apply the steering rules that are most appropriate for the particular environment that they enter.

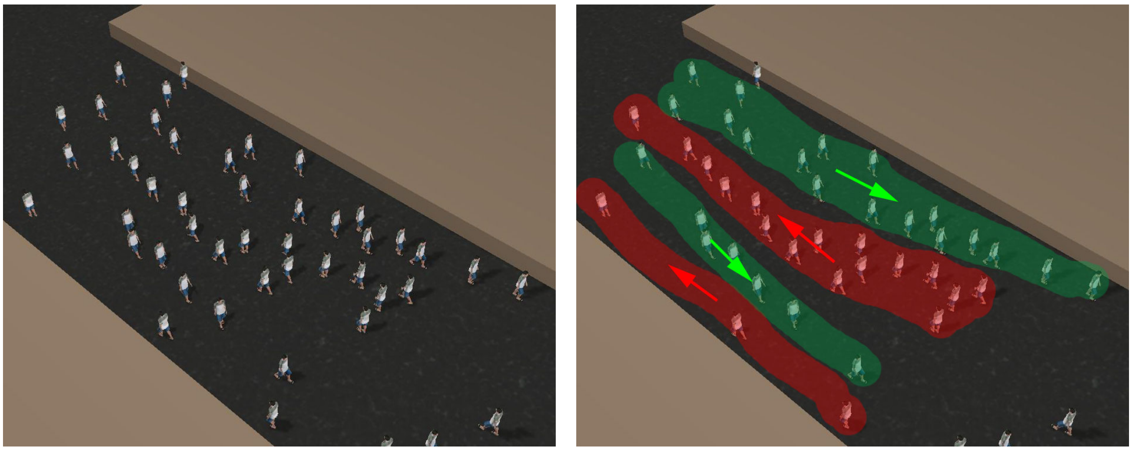
A.3 Connecting the Steering System and Locomotion System

There remains the challenge of interfacing locomotion with steering to ensure both responsiveness and naturalness. Singh et al. (2010) mentioned that the step space might be the best interface. Through experiments, we found that steering and locomotion should be improved simultaneously to ensure the flexibility of motions for pedestrians. Our system was improved iteratively. Some improvements include the following:

1. Locomotion: Adding start walking in different angles; adding turn in place;
2. Steering: Tuning the turning amount and speed in the steering command; slowing down when approaching obstacles.

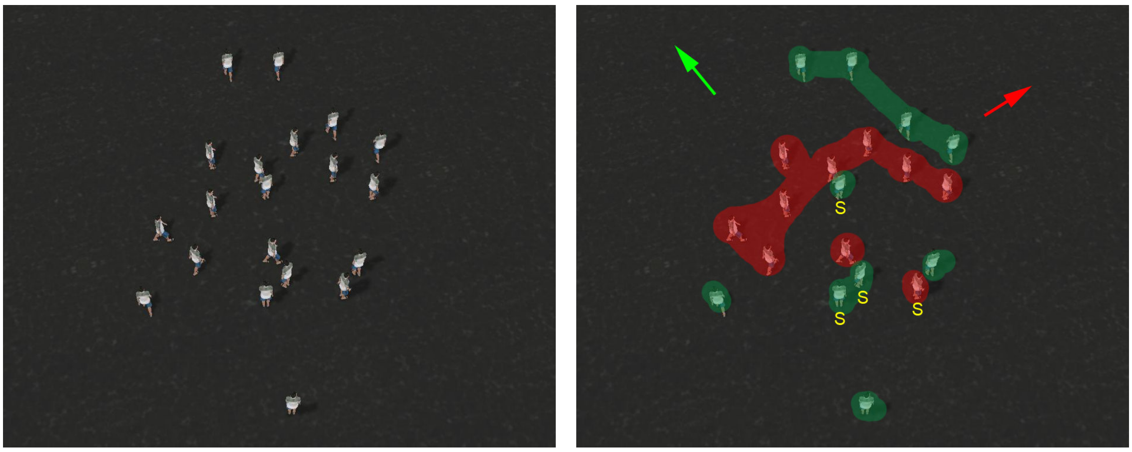
A.4 Results

Our results demonstrate human-like behaviors at both individual and global levels. In a corridor scenario (Figure A.4(a)), alternating lanes of pedestrians form when two groups of pedestrians move past each other (Figure A.4(b)), while each pedestrian is independently adjusting their speed and turning angle based on steps. In a crossing scenario (Figure A.5(a)), pedestrians will predict the locomotion direction of other nearby pedestrians, and adjust their motions accordingly; and if failed, stop and wait for others to pass, in a polite manner (Figure A.5(b)).



(a) (b)

Figure A.4: (a) Corridor scenario. (b) Alternate lane formation in the corridor scenario.



(a) (b)

Figure A.5: (a) Crossing scenario. (b) Some pedestrians stop (marked with “S”) in the crossing scenario.

APPENDIX B

Gaze and Attention Modeling and Animation

Perception is necessary for acquiring environmental information. Attention furthermore focuses perception on the most relevant environmental information. Human perceptual and attentional processes govern the motions of the eyes, head, and possibly other body parts. To better support the perception task, eye movements do not occur independently, but usually in conjunction with other body movements. We regard the gaze motions themselves and their supporting body movements as gaze behaviors in general. Gaze behaviors have previously been explored in the context of crowd simulation. The gaze behavior system for our autonomous pedestrians is based on the work of Grillon and Thalmann (2009), but we extend their approach such that gaze can affect other body motions, specifically the steering behaviors of the pedestrians. Our novel contribution significantly improves the realism of the simulated gaze behaviors.

B.1 Gaze Behaviors

We consider natural gaze behaviors not only as some independent natural motions of the head and eyes, but also an indication of the hidden attentions and behaviors of the agent. Basically, most human behaviors are led by perception, which is the result of gaze behaviors. But since human attention can be diverted by salient visual targets and the feedback of gaze is not limited to perceptual information, but also the classification of the current situation (e.g., being attracted, very limited perception, and the attention of other agents), these gaze behaviors can

	Gaze Type	Description
a	Monitoring	While walking, an agent will look at the horizon or at the destination and occasionally glance at the ground. When crossing the road, an agent will glance at the light more frequently when it is yellow than when it is green, anticipating an imminent change.
	Motion in the Periphery	The peripheral motion sensor of an agent will detect moving objects in the periphery and attend to them.
	Spontaneous Looking	When there are no deliberate or exogenous events vying for attention, attention is drawn to targets that are likely to be interesting.
b	Interruption	Events occurring in the environment that will attract the attention of an agent, and immediately cause the agent to shift its gaze towards them.
c	Visual Search	If an agent is unsure about the location of a gaze target, it will generate a sequence of intermediate positions that move the eye from its current position towards the target position.
	Reaching and Grasping	While reaching and grasping, an agent will generate eye movements toward the relevant grasp site. We implement this gaze type to synthesize natural full-body motions for door and doorway behavior simulation.

Table B.1: Gaze types. (a) Steering related gaze types. (b) Interruption gaze. (c) Other important gaze types.

affect the behaviors of other body parts, such as those responsible for steering. Automatically synthesized head/eye motions can add much lifelike liveliness to pedestrian characters, which can substantially improve the fidelity of pedestrian simulations.

B.1.1 Gaze Types

Chopra-Khullar and Badler (1999) proposed a computational framework for generating visual attention behaviors, which dealt with various types of gaze behaviors pertinent to most common human activities. We model our system based on the portion that is relevant to the steering task: 'monitoring', 'motion in the periphery', and 'spontaneous looking' (Table B.1(a)). Additionally, we include the gaze type of 'interruption' (Table B.1(b)) to deal with visually attractive exogenous events. Furthermore, there are two other interesting and prominent gaze types with which we do not deal in the context of the steering task, but which are worth mentioning: 'visual search' and 'reaching and grasping' (Table B.1(c)).

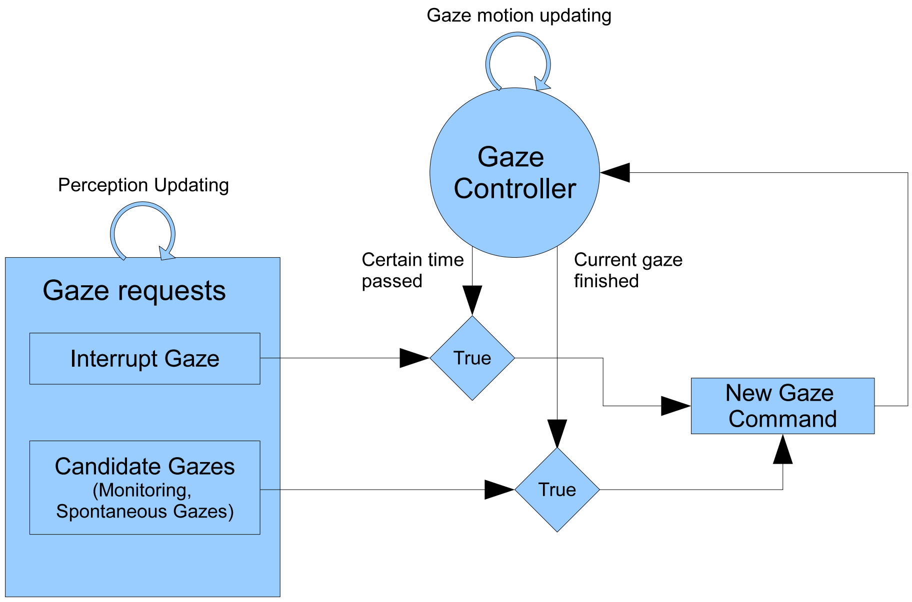


Figure B.1: Overview of the gaze manager.

B.1.2 Attention Manager

Since there are a few different types of gaze behaviors, we need a selection and scheduling mechanism for generating gazes. Following the work of Pennock (2005), we designed a gaze manager, an overview of which can be seen in Figure B.1, based on the following rules of gaze selection:

1. Gazes should be chosen based on the type, which determines their basic priorities.

2. In general, a new gaze can be performed after the last gaze is finished. But the current gaze can also be interrupted if the requested new gaze is an interruption type and if the minimal gaze time has passed.

3. After a target is gazed, it should not be gazed again for a certain amount of time. There should be a mechanism for tracking this.

4. An agent will be able to decide when to abandon the current eye behavior

and initiate a subsequent one based on the confidence of locating the visual target.

B.2 Saliency Map

The concept of a Saliency Map originated in the field of Computer Vision, and it has been applied within the computer animation domain in recent years. Peters and O'Sullivan (2003) modeled the visual attention of a virtual human based on the saliency map. Some graphics researchers avoid the image processing required to compute a conventional saliency map, instead building an abstract saliency map at significantly reduced computational cost. Kim (2006) designed a computational model of attention based on calculating the benefits and costs of objects to look at. Grillon and Thalmann (2009) proposed a method for extracting interest points to gaze at with consideration to proximity, relative speed/orientation, and periphery.

To model attention, we investigated the use of abstract Saliency Maps to track which targets to look at, and applied this idea in a conversational agent (Hartholt et al., 2009). If certain keywords are mentioned during a dialogue, the character will use the map to decide whether to look at the related objects, whereas, in the idle state, the character will look at objects around it, which is decided by a biased random probability model based on their priorities in the map.

B.3 Head/Eye Controller

The head/eye controller realizes the gaze behaviors according to a set of parameters. Following the work of Pennock (2005) and Lee et al. (2007), we designed the parameters as indicated in Table B.2.

The gaze-affected joint range is an interesting phenomenon we would like to investigate. It should vary based on the importance of the visual targets and the limits of the joints. Grillon and Thalmann (2009) realized the gaze motion by

Type	Description
Gaze Behavior type	Includes interrupt gaze, monitoring gaze (peripheral vision is included in this type), and spontaneous gaze.
Target	The target can be an object, a point or an angle.
Gaze motion type	Includes glance and focus. Glance means the character will look at the target once, and then look back in the original direction. Focus means the target will be tracked continuously.
Duration	If the gaze motion type is set to glance, it defines the time length for which the agent keeps looking at the target.
Speed	The speed of the gaze motion. It includes slow, normal, and fast.
Joint affected	Define the joint range to be affected by the gaze. The joints that can be affected include eyes, head, spine, and foot. If the range is set to include foot, the agent will bend its original trajectory towards the visual target while looking at it.
Priority	Defines the importance of the gaze.

Table B.2: Defined gaze parameters.

an optimized dedicated gaze Inverse Kinematics solver. A displacement map is computed and propagated over a automatically defined time period.

Additionally, there are some other topics related to simulating gaze motions. Expressive gaze was investigated for adding emotional variations into gaze motions, which added much liveliness to the motions (Queiroz et al., 2008). Lee and Terzopoulos (2006) developed a biomechanical model of the neck for synthesizing natural head motion.

Saccadic movements (Lee et al., 2002) are fast eye movements that impart liveliness to the eyes. They obey natural positional and timing patterns. In general, a saccade towards a certain visual target follows the pattern of rotating the eyes to the direction of the target, followed by some fast eye movements of small angles around the target, and then a period of fixation. By following this concept, we developed a simple model in the aforementioned conversational agent project and obtained good results.

B.4 Incorporating Attention Into Steering Behaviors

Given the many research efforts in crowd simulation, head/eye motion generation, and attention modeling, there remains a vacuous area about how attention can affect the behaviors of other body parts. In the first place, we want to simulate how attention can affect steering behaviors. In general, the following are some interesting facts to simulate:

1. As mentioned in Sec. B, the gaze behaviors related to steering tasks includes monitoring and peripheral vision. These gaze behaviors ensure agents acquire the necessary perceptual information to make steering decisions. The literatures reveals that a walking agent will look at each agent in front it repetitively, and after a certain period, it will look at the destination. More importantly, if there is a imminent collision threat to the front, the agent should look at it more frequently, or even stare at it until the threat is eliminated.

2. In more lifelike scenarios, the agents have not only steering tasks, but they can also be distracted by other things in the environment. One example is when there is an attractive event happening the agents passing by will attend to it, which might lead to bending their original travel trajectories towards the position of the event. Meanwhile, they will pay less attention to the steering task so that they may not predict the motion of other agents very well, which could result in more stops.

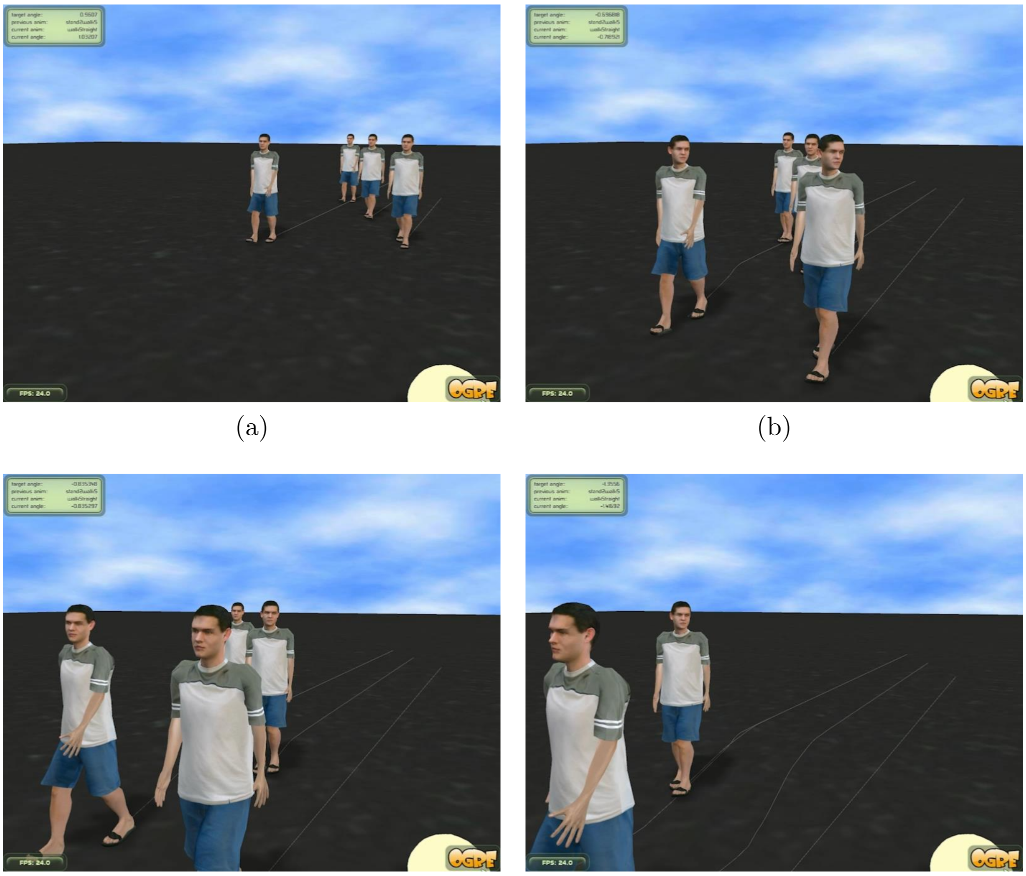
3. When there is a large angle between the traveling direction and the attended direction, the agent tends to slow down to avoid possible collisions. On the other side, if a pedestrian finds a threatening agent in front is attending to some other events and unaware of the pedestrian it will avoid this threat more proactively than normal, such as turning in larger angles.

4. When turning around a sharp corner, since perception is limited, the agent will slow down. If two agents do not see each other due to occlusion, this should result in a collision.

5. As mentioned by Pennock (2005), when a walking agent is passing from behind, it should be glanced at sideways, which is a behavior frequently observed in crowds.

B.5 Results

We implemented the gaze motion generation with gaze behavior selection, and integrated it with steering. Figure B.2 shows that the steering behavior will be affected as characters attend to and gaze at an interesting point in the environment.



(a) (b) (c) (d)

Figure B.2: As characters attend to an interesting visual target at the lower-right corner, they slightly bend their steering trajectory towards it while slowing down a little. After they pass the target and stop attending to it, they resume their normal walking trajectory and speed.

APPENDIX C

Hybrid Full-Body Motion Control for Reaching

Object manipulation is an important motor ability for virtual humans, but for the most part it has been overlooked in pedestrian simulation. The motor control logic should ensure that the hand accurately reaches the grasp target, while the whole body motion appears natural, and the performance is fast enough for the purposes of multi-human simulation. We have proposed a method for achieving these requirements (Huang et al., 2010), and this appendix is based mainly on that work.

C.1 Coordinated Whole-Body Reaching Motion

A fundamental research problem in motion planning for virtual characters (and robots) is to control the body in order to achieve a specified target position. It occurs frequently in game scenarios when characters must reach to grasp an object. There are three issues that may need to be addressed in generating realistic reaching motion: (1) correctness of hand placement, (2) satisfaction of the constraints on the character's body, and (3) naturalness of the whole-body movement. An ideal reaching motion requires the character's hand to follow a smooth, collision-free trajectory from its current position to the target position while satisfying the constraints on the entire body. Inspired by prior work (Kallmann and Marsella, 2005; Monheit and Badler, 1991; Kulpa and Multon, 2005), we consider balancing, coordinating, and variation for reaching motion that takes into account the motion of the arms, spine, and legs of an anthropomorphic character.

In particular, we develop a hierarchical control strategy for controlling a virtual character. We use analytical IK to control the hands of the character, thus providing fast and accurate control of the end effectors (hands). A stepping motion controller, coupled with a non-deterministic state machine, controls the lower limbs to move the character towards a target that is beyond its reach. The spine controller ensures the balance of the character by accounting for its center of mass, and uses rotation decomposition for simple and robust control. Finally, we develop a novel controller-scheduling algorithm that generates the coordination strategies between multiple controllers.

C.2 Hierarchical Control Structure

The character has a hierarchical skeletal structure. The spine is at the highest level of the hierarchy and it determines the position and orientation of the entire character. The head, arms, and legs are at the second level of the hierarchy. The arms and legs are controlled using analytical IK, facilitating easy control of the joint angles and specifying joint limits. The four controllers that we employ deal with different sets of hierarchical skeleton components. The arm controller enables the arm to perform a reaching motion. The spine controller controls the spine and decomposes rotations into swing and twist for simple and robust control. The head controller determines the direction in which the character's head is facing. The leg controller controls the legs and the root joint of the spine by generating stepping movements.

C.2.1 Arm

Arm motion is a fundamental aspect of reaching, as it determines the ability and accuracy of the motion of the end-effector (hand) in reaching a desired position.

The end-effector is required to follow a smooth trajectory from its current position to the desired position, while obeying joint constraints and providing natural-looking motions (we do not deal with hand grasping motions). Modeling the trajectories of hand motion using Bezier curves (Faraway et al., 2007) can yield satisfactory results, but this requires pre-captured hand motion data. We employ an ease-out strategy in which the hand trajectory is the shortest path to the target and the speed of the hand starts relatively high and gradually decreases as the hand reaches the target. This results in the hand slowing down naturally as it nears the target position and avoids "punching" the target.

C.2.2 Spine

The spine of the character plays a crucial role during reaching as the character needs to bend and twist its upper body to reach targets at different positions. The spine controller determines the upper-body position and orientation of the character (six degrees of freedom). Our four-joint spine model enhances the control of the character, resulting in more fluid and natural animations.

Spine Rotation Decomposition: Previous work (Kallmann, 2008) introduced swing-and-twist orientation decomposition for arm joints, enabling simple and robust control. Inspired by this work, we similarly decompose the orientations of the spine joints. The advantage of this decomposition method is that it separates the spine motion into two primitive orientations, each of which can be governed by a separate control module in order to achieve robust control.

Center of Mass Adjustment: Consider a reaching motion when bending the upper body down to pick up an object. To balance the character, the hip must move backwards in order to maintain the COM within the support polygon of the character. Similarly when reaching a high target, the hip must move forward to maintain balance. In our framework, we adjust the position of the sacral joint, which determines the root of the spine, to generate visually appealing balancing motion during spine swing.

C.2.3 Leg

When synthesizing reaching motion, the focus tends to be on arm control and stepping motion is often neglected. Humans frequently step in the direction of the target in order to assist their reaching action. In doing so, the target falls within the convenient reaching range; i.e., it is neither too close nor too far from the actor. Some research has considered this topic in the context of motion planning (Kallmann et al., 2003). Yoshida et al. (2006) considered stepping as a dynamic stability problem. We address the problem as an intentional behavior, employing a leg controller that uses a probabilistic state machine to determine the stepping motion of the virtual character.

Stepping Motions: Two kinds of stepping motions can occur during the reaching process—half stepping or full stepping. Half stepping involves only one foot step forward while the other remains at its original position. Full stepping involves both feet moving forward to assist the reaching motion. One can observe that when the step length is small, people usually perform half stepping, while a large step length often compels full stepping. However, this is also subject to individual "random" factors, which is considered in our implementation. A probabilistic state machine is used to determine the stepping motion of a character that assists its reaching motion.

Leg Flexibility: The flexibility of the legs is an interesting motion feature that manifests itself during a reaching action. In order to pick up an object from the ground, the character needs to choose between stooping or squatting to lower its upper body. One factor in producing this motion is the flexibility of the legs. If the flexibility is low, a stoop is preferable in the reaching motion, otherwise if the flexibility is high, a squat is preferred. The squat motion involves lowering the position of the root of the spine, while the stoop motion does not. A stoop range is computed as the ratio of the length of the arm and a user-defined "flexibility parameter. If the vertical distance between the shoulder and the target exceeds this range, the root is triggered to squat down in order to reach the target.

C.3 Motion Controller Scheduling for Coordinated Reaching Motions

The order of movement of the arm, legs, and spine often varies with each situation and for different individuals. Hence, there exist multiple valid orderings of the individual control strategies that produce realistic animation. For example, a character may first choose to minimize the distance to the target in the horizontal space (stepping motion), then twist the body to achieve a comfortable reaching angle (spine twist rotation), then minimize the distance in the vertical space (spine swing rotation), and finally position the hand to touch the object. This ordering of controllers may easily be changed in a different situation.

To address the non-uniqueness, we devise a motion controller scheduling strategy that selects a sequence of one or more controllers to animate a virtual character for a particular reaching task. An overview of the motion controller scheduling algorithm is presented in Figure C.1. First, we identify four states in which a character may find itself during any given reaching task. In each state, one or multiple controllers are triggered to perform stepping, arm, head, or spine motions in an effort to reach the target. These states are evaluated in sequence, implicitly ordering the selection of the controllers.

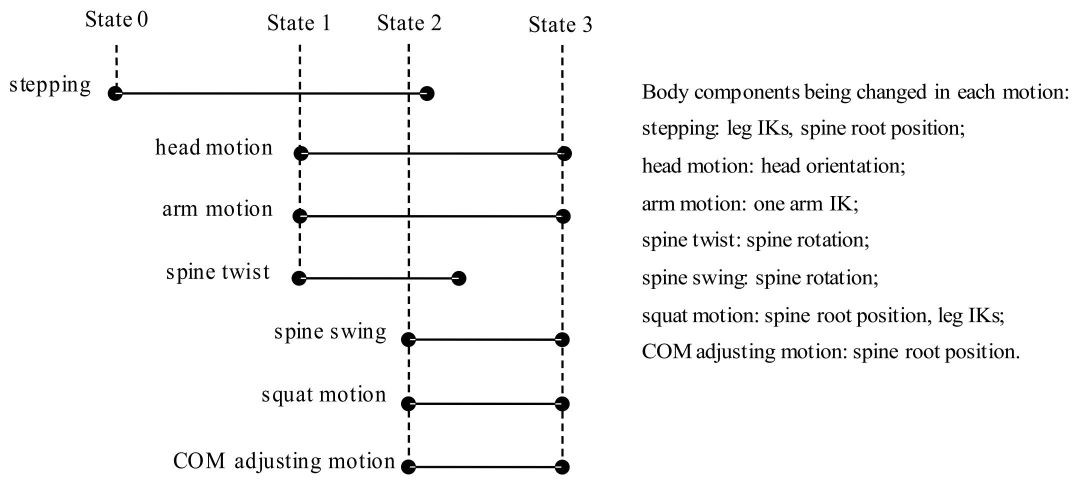


Figure C.1: Motion controller scheduling.

C.4 Results

Figure C.2 illustrates a diverse set of reaching motions synthesized for a full-body character using our method. For targets beyond its reach, the character chooses appropriate stepping motions and spine motions and it integrates them with head and hand motions to achieve coordinated, graceful whole-body movement. Diversity in the motion is achieved using a probabilistic model for stepping along with user defined parameters such as leg flexibility. Coordination and diversity are integrated in the motion synthesis by the modular design of the motion controllers with the systematic scheduling algorithm.

More spine configurations and stepping with additional steps should be considered in future work. As in (Yamane et al., 2004), motion capture data can be coupled with controllers to improve the naturalness of the synthesized motion. Moreover, our method should be extended to consider collision avoidance as well as to handle dynamic targets. A more intelligent control mechanism such as a planner would be required to handle space-time goals.

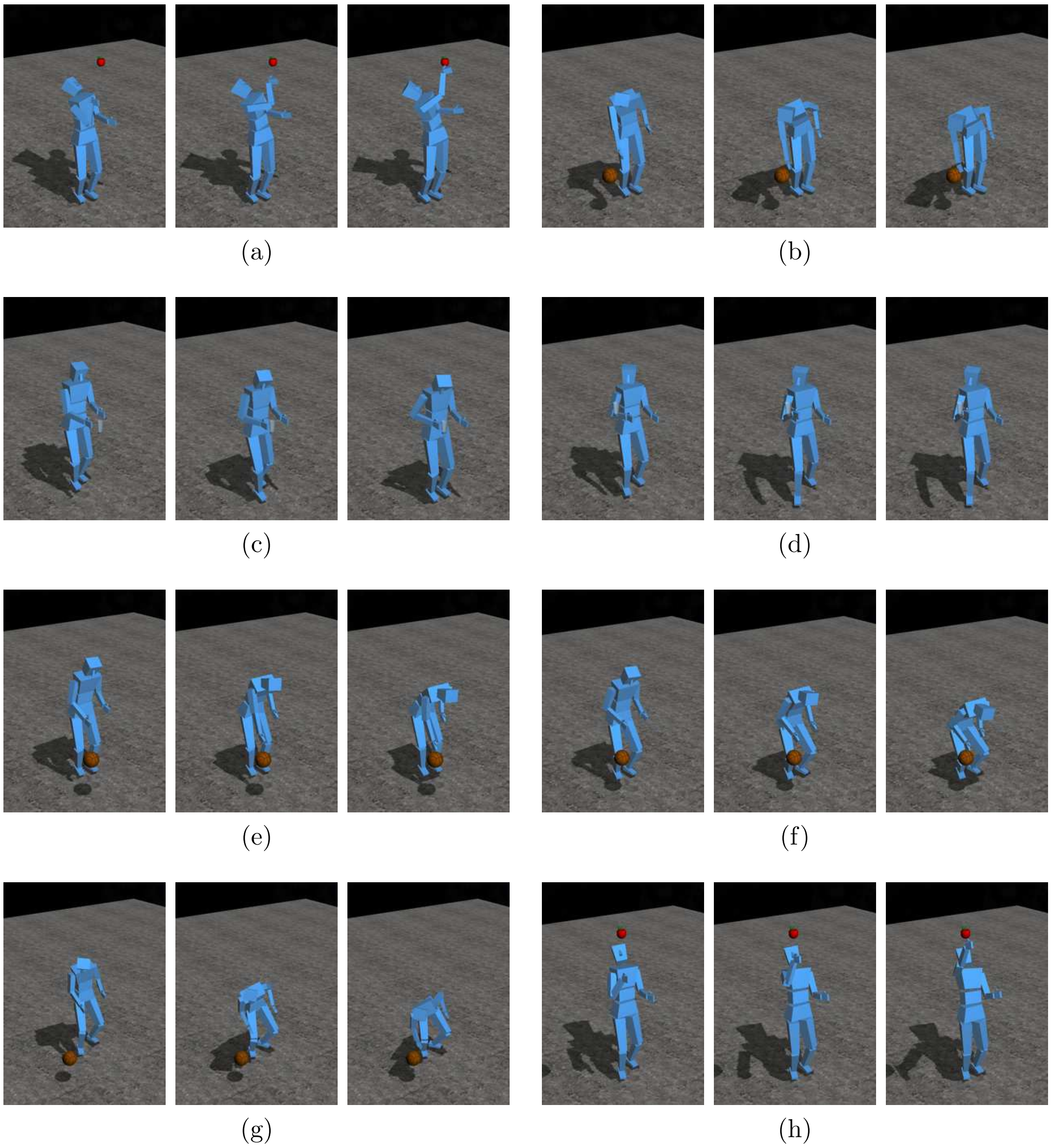


Figure C.2: A diverse set of reaching motions synthesized using our method. (a)–(b) Swing and twist spine motion: (a) The character reaches an apple to the upper-left; (b) the character reaches a basketball to the lower-right. (c)–(d) The character performs full stepping and half stepping to reach a glass. (e)–(f) The character with different leg flexibilities: (e) The character has a lower leg flexibility than in (f) as it stoops to reach a low target; (f) the character squats to reach the same low target. (g)–(h) Combined results: (g) The character steps and squats to reach a low basketball; (h) the character steps and swings up to reach a high apple.

APPENDIX D

Virtual Vision Application

An interesting application of multi-human simulation in urban environments is "Virtual Vision", a synthesis of computer graphics, human simulation, and computer vision technologies (Terzopoulos, 2003). Virtual vision has been employed to prototype and experiment with several autonomous CCTV (Closed Circuit Television) surveillance systems (Qureshi and Terzopoulos, 2008). In this appendix we present the application of our pedestrian simulation system in the implementation of a new virtual vision prototype.

D.1 Environment Model

Using photos and blueprints of Winston Chung Hall on the campus of the University of California, Riverside, we accurately reconstructed a textured 3D environmental model of a large part of the second floor of that building (Figure D.1).

D.2 Populating the Environment With Autonomous Pedestrians

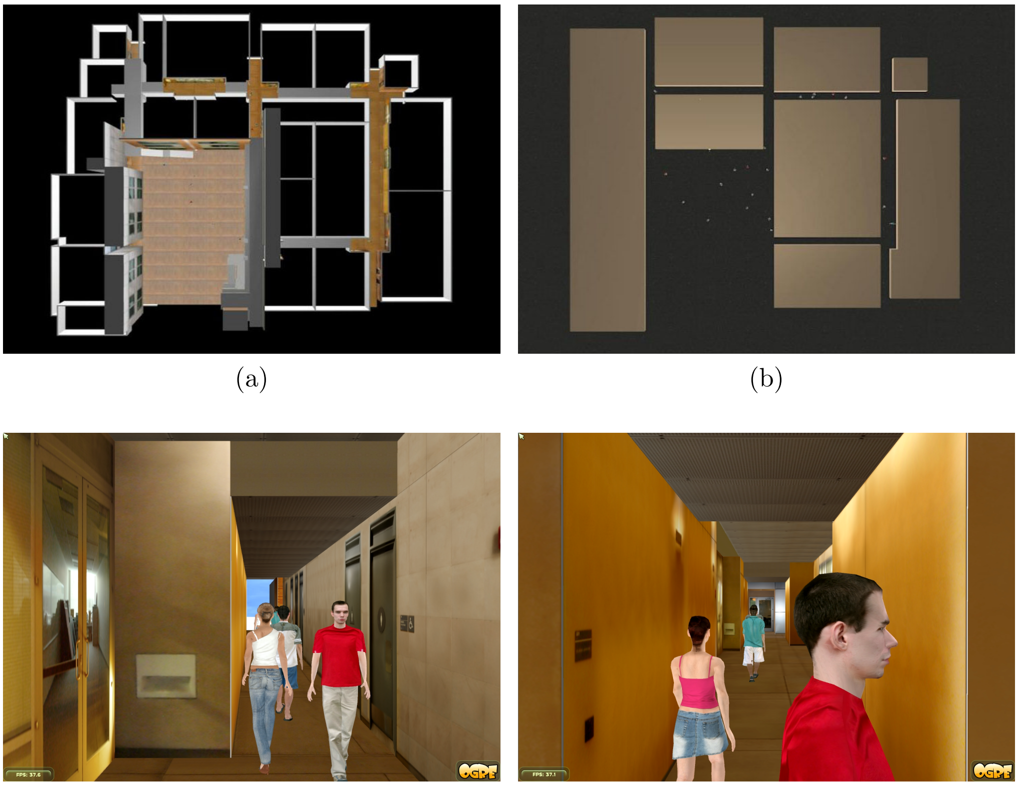
Our autonomous pedestrian system requires little labor to populate the virtual building. The first step is to discretize the free space in our 3D environment model (Figure D.2(a)) using grids and to specify the obstacles in the environment (Figure D.2(b)). The second step is to instantiate pedestrians at different initial locations and give them different target destinations. The system is then ready to run.

Figure D.1: Building model for virtual human simulation.

Pedestrians will automatically navigate through the environment to reach their destinations, performing appropriate locomotion and steering to avoid collisions with one another and with obstacles (Figure D.2(c)–(d)). To deal with the narrow corridors, the autonomous pedestrians employ special rules to handle the highly constrained situation, and we annotate the narrow corridors to distinguish them from the open, central common space.

D.3 Virtual Video Surveillance System

We implemented a multi-camera virtual vision surveillance system with the following features: There are a total of 32 virtual PTZ (pan/tilt/zoom) video surveillance cameras strategically situated in the virtual building environment. Our system produces three CCTV display panels, each tiled with synthetic video



(a) (b)

(c) (d)

Figure D.2: Buidling model populated with autonomous pedestrians.

streams captured by approximately 10 of the cameras (Figure D.3). The large video stream shown at the center of each display panel is the one that the surveillance operator has currently selected (indicated by the red border) and the associated camera can be interactively panned, tilted, and zoomed using the slider controls displayed below the central video stream. Using the mouse, the operator can click on one of the surrounding video streams to select it. The selected video stream will then become the expanded central stream and the associated video camera will become interactively controllable by the operator.

Figure D.3: CCTV panel tiled with video camera streams. The pan, tilt, and zoom parameters of the currently selected camera (bordered by red) can be interactively controlled using the three sliders situated below the central, larger-format video stream displayed from that camera.

Bibliography

Arisumi, H., Chardonnet, J.-R., and Yokoi, K. (2009). Whole-body motion of a humanoid robot for passing through a door: Opening a door by impulsive force. In *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IROS'09, pages 428–434, Piscataway, NJ, USA. IEEE Press. 9

Bai, Y., Siu, K., and Liu, C. K. (2012). Synthesis of concurrent object manipulation tasks. *ACM Trans. Graph.*, 31(6):156:1–156:9. 9, 60

Chitta, S., Cohen, B., and Likhachev, M. (2010). Planning for autonomous door opening with a mobile manipulator. In *International Conference on Robotics and Automation*, Anchorage, Alaska. 9

Choi, M. G., Lee, J., and Shin, S. Y. (2003). Planning biped locomotion using motion capture data and probabilistic roadmaps. *ACM Trans. Graph.*, 22(2):182–203. 9

Chopra-Khullar, S. and Badler, N. I. (1999). Where to look? Automating attending behaviors of virtual human characters. In *Proceedings of the Third Annual Conference on Autonomous Agents*, AGENTS '99, pages 16–23, New York, NY, USA. ACM. 72

Coros, S., Beaudoin, P., and van de Panne, M. (2009). Robust task-based control policies for physics-based characters. In *ACM SIGGRAPH Asia 2009 Papers*, SIGGRAPH Asia '09, pages 170:1–170:9, New York, NY, USA. ACM. 60

Currie, K. and Tate, A. (1991). O-Plan: The open planning architecture. *Artificial Intelligence*, 52(1):49–86. 10

Durupinar, F., Pelechano, N., Allbeck, J., Gudukbay, U., and Badler, N. I.

(2011). How the ocean personality model affects the perception of crowds. *IEEE Comput. Graph. Appl.*, 31(3):22–31. 7

Egges, A. (2008). Opening doors in motion analysis research. In Egges, A., Kamphuis, A., and Overmars, M., editors, *Proceedings of the International Conference on Motion in Games*, pages 188–199, Berlin, Heidelberg. Springer-Verlag. 9

Faraway, J., Reed, M., and Wang, J. (2007). Modeling 3D trajectories using Bezier curves with application to hand motion. *Applied Statistics*, 56:571–585. 81

Funge, J., Tu, X., and Terzopoulos, D. (1999). Cognitive modeling: Knowledge, reasoning and planning for intelligent characters. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 29–38, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co. 3, 11

Grillon, H. and Thalmann, D. (2009). Simulating gaze attention behaviors for crowds. *Comput. Animat. Virtual Worlds*, 20:111–119. 71, 74

Guy, S. J., Kim, S., Lin, M. C., and Manocha, D. (2011). Simulating heterogeneous crowd behaviors using personality trait theory. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '11, pages 43–52, New York, NY, USA. ACM. 7

Hartholt, A., Gratch, J., Weiss, L., and Team, T. G. (2009). At the virtual frontier: Introducing gunslinger, a multi-character, mixed-reality, story-driven experience. In Ruttkay, Z., Kipp, M., Nijholt, A., and Vilhjlmsson, H., editors, *Intelligent Virtual Agents*, volume 5773 of *Lecture Notes in Computer Science*, pages 500–501. Springer Berlin Heidelberg. 74

Heck, R., Kovar, L., and Gleicher, M. (2006). Splicing upper-body actions with locomotion. *Computer Graphics Forum*, 25(3):459–466. 9

Huang, W., Kapadia, M., and Terzopoulos, D. (2010). Full-body hybrid motor control for reaching. In *Proceedings of the Third International Conference on Motion In Games*, MIG'10, pages 36–47, Berlin, Heidelberg. Springer-Verlag. 79

Kallmann, M. (2008). Analytical inverse kinematics with body posture control. *Comput. Animat. Virtual Worlds*, 19(2):79–91. 39, 81

Kallmann, M., Aubel, A., Abaci, T., and Thalmann, D. (2003). Planning collision-free reaching motions for interactive object manipulation and grasping. *Computer Graphics Forum (Proc. Eurographics'03)*, 22(3):313–322. 82

Kallmann, M. and Marsella, S. (2005). Hierarchical motion controllers for real-time autonomous virtual humans. In *Proc. 5th Int. Conf. on Intelligent Virtual Agents*, pages 243–265, Kos, Greece. 79

Kim, M., Hyun, K., Kim, J., and Lee, J. (2009). Synchronized multi-character motion editing. In *ACM SIGGRAPH 2009 Papers*, SIGGRAPH '09, pages 79:1–79:9, New York, NY, USA. ACM. 8

Kim, Y. (2006). *A computational model of dynamic perceptual attention for virtual humans*. PhD thesis, University of Southern California, Los Angeles, CA, USA. AAI3238341. 74

Kulpa, R. and Multon, F. (2005). Fast inverse kinematics and kinetics solver for human-like figures. In *5th IEEE-RAS Int. Conf. on Humanoid Robots*, pages 38–43. 79

Kwon, T., Cho, Y.-S., Park, S. I., and Shin, S. Y. (2008). Two-character motion analysis and synthesis. *IEEE Transactions on Visualization and Computer Graphics*, 14(3):707–720. 8, 60

Lau, M. and Kuffner, J. J. (2005). Behavior planning for character animation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '05, pages 271–280, New York, NY, USA. ACM. 8, 18

Lee, J., Marsella, S., Traum, D., Gratch, J., and Lance, B. (2007). The rickel gaze model: A window on the mind of a virtual human. In *Proceedings of the 7th international conference on Intelligent Virtual Agents*, IVA '07, pages 296–303, Berlin, Heidelberg. Springer-Verlag. 74

Lee, S.-H. and Terzopoulos, D. (2006). Heads up! Biomechanical modeling and neuromuscular control of the neck. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, pages 1188–1198, New York, NY, USA. ACM. 75

Lee, S. P., Badler, J. B., and Badler, N. I. (2002). Eyes alive. *ACM Trans. Graph.*, 21:637–644. 75

Lee, Y., Lee, S. J., and Popović, Z. (2009). Compact character controllers. *ACM Trans. Graph.*, 28(5):169:1–169:8. 9

Liu, C. K., Hertzmann, A., and Popović, Z. (2006). Composition of complex optimal multi-character motions. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '06, pages 215–222, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association. 8

Lo, W.-Y. and Zwicker, M. (2008). Real-time planning for parameterized human motion. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '08, pages 29–38, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association. 9

Monheit, G. and Badler, N. I. (1991). A kinematic model of the human spine and torso. *IEEE Computer Graphics and Applications*, 11(2):29–38. 79

Multon, F., France, L., Cani-Gascuel, M.-P., and Debunne, G. (1999). Computer animation of human walking: a survey. *The Journal of Visualization and Computer Animation*, 10(1):39–54. 62

Nagatani, K. and Yuta, S. (1995). An experiment on opening-door-behavior by an autonomous mobile robot with a manipulator. In *Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 45–50 vol.2. 9

O'Sullivan, C. and Ennis, C. (2011). Metropolis: Multisensory simulation of a populated city. In *3rd IEEE International Conference on Games and Virtual Worlds for Serious Applications (VS-Games)*, pages 1–7. 3, 7

Park, S. I., Shin, H. J., and Shin, S. Y. (2002). On-line locomotion generation based on motion blending. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '02, pages 105–111, New York, NY, USA. ACM. 63

Pelechano, N., Allbeck, J. M., and Badler, N. I. (2007). Controlling individual agents in high-density crowd simulation. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '07, pages 99–108, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association. 3, 7

Pennock, C. (2005). Gaze control for autonomous pedestrians. Master's thesis, Computer Science Department, New York University. 73, 74, 77

Perlin, K. (1995). Real time responsive animation with personality. *IEEE Transactions on Visualization and Computer Graphics*, 1(1):5–15. 26

Pervin, L. A., editor (1996). *The Science of Personality*. Oxford University Press, USA. 8, 35

Peters, C. and O'Sullivan, C. (2003). Bottom-up visual attention for virtual human animation. In *Computer Animation for Social Agents*, pages 111–117. 74

Queiroz, R. B., Barros, L. M., and Musse, S. R. (2008). Providing expressive gaze to virtual animated characters in interactive applications. *Comput. Entertain.*, 6:41:1–41:23. 75

Qureshi, F. and Terzopoulos, D. (2008). Smart camera networks in virtual reality. *Proceedings of the IEEE*, 96(10):1640–1656. 86

Russell, S. J. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Pearson Education, 2nd edition. 36

Santamaria, J. and Rosenbaum, D. (2011). Etiquette and effort: Holding doors for others. *Psychol Sci*, 22(5):584–8. 5, 8, 33

Schuerman, M., Singh, S., Kapadia, M., and Faloutsos, P. (2010). Situation agents: Agent-based externalized steering logic. *Comput. Animat. Virtual Worlds*, 21(34):267–276. 9

Shao, W. and Terzopoulos, D. (2007). Autonomous pedestrians. *Graph. Models*, 69(5-6):246–274. 3, 7, 14, 67

Shum, H. P. H., Komura, T., and Yamazaki, S. (2008). Simulating interactions of avatars in high dimensional state space. In *Proceedings of the 2008 Symposium on Interactive 3D Graphics and Games*, I3D '08, pages 131–138, New York, NY, USA. ACM. 8, 60

Singh, S., Kapadia, M., Hewlett, B., Reinman, G., and Faloutsos, P. (2011a). A modular framework for adaptive agent-based steering. In *Symposium on Interactive 3D Graphics and Games*, I3D '11, pages 141–150, New York, NY, USA. ACM. 14, 67

Singh, S., Kapadia, M., Reinman, G., and Faloutsos, P. (2011b). Footstep navigation for dynamic crowds. In *Symposium on Interactive 3D Graphics and Games*, I3D '11, pages 203–203, New York, NY, USA. ACM. 9

Singh, S., Kapadia, M., Reinmann, G., and Faloutsos, P. (2010). On the interface between steering and animation for autonomous characters. In *Workshop on Crowd Simulation, Computer Animation and Social Agents*, Saint-Malo, France. 69

Terzopoulos, D. (1999). Artificial life for computer graphics. *Commun. ACM*, 42(8):32–42. 3, 11

Terzopoulos, D. (2003). Perceptive agents and systems in virtual reality. In *Proc. ACM Symposium on Virtual Reality Software and Technology*, pages 1–3, Osaka, Japan. 86

van Basten, B. and Egges, A. (2011). Flexible splicing of upper-body motion spaces on locomotion. *Computer Graphics Forum*, 30(7):1963–1971. 9

van Basten, B. J. H., Peeters, P. W. A. M., and Egges, A. (2010). The step space: Example-based footprint-driven motion synthesis. *Comput. Animat. Virtual Worlds*, 21(34):433–441. 9, 14, 63, 65

Wampler, K., Andersen, E., Herbst, E., Lee, Y., and Popović, Z. (2010). Character animation in two-player adversarial games. *ACM Trans. Graph.*, 29(3):26:1–26:13. 60

Webster, H. J., Hish, E., JuraleeSmith, Speck, L., Braun, C., and Zutter, K. E. (2007). Allow me: Gender stereotypes and equality in door-holding. *Radford University Undergraduate/Graduate Research Forum*. 8, 35

Weiss, G., editor (1999). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press, Cambridge, MA, USA. 10

Yamane, K., Kuffner, J. J., and Hodgins, J. K. (2004). Synthesizing animations of human manipulation tasks. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 532–539, New York, NY, USA. ACM. 9, 60, 84

Yeh, H., Curtis, S., Patil, S., van den Berg, J., Manocha, D., and Lin, M. (2008). Composite agents. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '08, pages 39–47, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association. 9

Yoshida, E., Kanoun, O., Esteves, C., and Laumond, J.-P. (2006). Task-driven support polygon humanoids. In *6th IEEE-RAS Int. Conf. on Humanoid Robots*. 82

Yu, Q. and Terzopoulos, D. (2007). A decision network framework for the behavioral animation of virtual humans. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '07, pages 119–128, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association. 3, 7, 33