Simulating Movement Interactions between Avatars & Agents in Virtual Worlds using Human Motion Constraints

Sahil Narang* Andrew Best† Dinesh Manocha‡

University of North Carolina, Chapel Hill
http://gamma.cs.unc.edu/pedvr

Abstract

We present an interactive algorithm to generate plausible movements for human-like agents interacting with other agents or avatars in a virtual environment. Our approach takes into account high-dimensional human motion constraints and bio-mechanical constraints to compute collision-free trajectories for each agent. We present a novel full-body movement constrained-velocity computation algorithm that can easily be combined with many existing motion synthesis techniques. Compared to prior local navigation methods, our formulation reduces artefacts that arise in dense scenarios and close interactions, and results in smoother and plausible locomotive behaviors. We have evaluated the benefits of our new algorithm in single-agent and multi-agent environments. We investigated the perception of a single agent's movements in dense scenarios and observed that our algorithm has a strong positive effect on the perceived quality of the simulation. Our approach also allows the user to interact with the agents from a first-person perspective in immersive settings. We conducted a study to investigate the perception of such avatar-agent interactions, and found that interactions generated using our approach lead to an increase in the user's sense of co-presence.

1 Introduction

The problem of generating realistic movement and behavior of human-like agents is important for many virtual reality applications, such as training simulators, entertainment and games, treatment of psychological disorders etc. Many such applications also enable the user to actively participate in an immersive virtual environment by embodying a virtual avatar i.e. a perceptible digital representation whose behaviors reflect those executed by a specific human being [18, 31]. Prior studies have established that human-like agents can elicit social responses [4], enhance the user's sense of presence in the virtual world [25], and generate plausible interactions between agents and avatars [20].

One of the major challenges is to generate plausible movement and behavior for each virtual agent as it interacts with other agents and avatars in the scene. The naturalness of the interaction is governed by the trajectory of each agent, as well as the full-body animation or actions [24]. Studies have shown that many full-body movements, such as shoulder motions, gestures, or gaze, can significantly impact the perceived naturalness and the sense of presence [9,15,21].

There is extensive work on human motion simulation and collision-free navigation in computer animation, crowd simulation, and robotics. Each human is represented as an articulated agent with tens of degrees-of-freedom (DOFs). Most human animation systems tend to use motion capture (Mocap) data, which is mainly used to compute the movement of a single human and is not well suited to simulate a large group of human-like agents at interactive rates, especially in dense scenarios.

*e-mail: sahil@cs.unc.edu
†e-mail:best@cs.unc.edu
‡e-mail:dm@cs.unc.edu

Prior interactive simulation algorithms decompose the movement interaction problem into 2D velocity computation or path planning for simple 2D agents, followed by 3D human motion synthesis. There is a large body of work [12, 35, 36] that uses simple 2D representations for each agent (e.g., a disc) and computes planar collision-free trajectories. This is followed by generating full-body animation for each human along a trajectory as a post-process [39]. This two-step decomposition overcomes the computational complexity of simulating high-dimensional agents for interactive applications. However, these methods do not account for many kinematic and dynamic stability constraints that are inherent to human locomotion which can result in artefacts especially when simulating movement interactions in dense spaces.

Main Results: We present *Body Aware Movement*, or BAM, a novel velocity computation algorithm for interactive multi-agent simulation that takes into account high-dimensional human motion constraints and computes collision-free trajectories for each agent. Our approach is designed to generate plausible full-body motion for multiple human-like agents in a virtual environment and to simulate the movement interactions with other agents and avatars (Section 3). Overall, BAM offers the following benefits over prior 2D velocity computation methods:

- BAM reduces the dimensionality mismatch between 2D navigation and high-DOF motion synthesis by deriving full-body motion constraints from captured data and established principles of bio-mechanics, and mapping them efficiently to the 2D velocity plane.
- BAM is general and can be easily integrated with many existing full-body animation or simulation methods.
- Our approach accounts for the presence of a tracked real user in an immersive virtual environment, and generates collision-free and plausible avatar-agent interactions.
- BAM can be easily parallelized on multiple cores and used to simulate hundreds of 2D agents at interactive rates. Moreover, it can simulate and render movement interactions of 60+ full-body agents at VR friendly rates.

We have integrated our system with the Unreal game engine to render the agents in real-time (Section 5). We demonstrate the benefits of our algorithm in generating agent-agent and avatar-agent interactions in many challenging scenarios. We have conducted an extensive user evaluation which highlights the benefits of BAM over prior methods in both immersive & non-immersive settings:

- **Evaluating Single-Agent & Agent-Agent Movement Interactions in Non-Immersive Settings:** We conducted two within-subjects user studies in non-immersive settings i.e. using a monitor display, which compared BAM to prior interactive methods [12, 36]. The first evaluated the plausibility

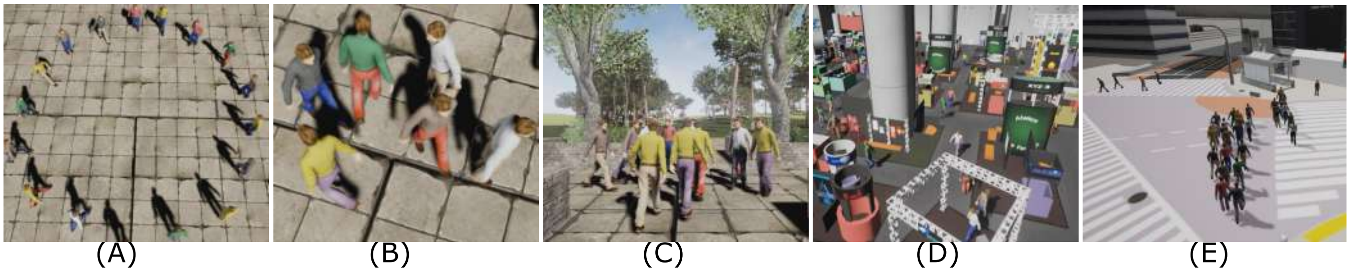

Figure 1: **Benchmarks:** Our algorithm can generate plausible agent-agent & avatar-agent interactions in complex scenarios including: (a-b) Antipodal circle scene with 17 human-like agents, (c) Dense 4-way Crossing, (d) Tradeshow with 50 agents and (e) Shibuya crossing with 50 agents. We can simulate and render tens of full-body agents at 60 fps on a multi-core desktop PC.

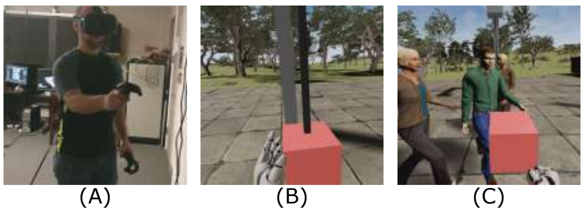

Figure 2: **Avatar-agent interactions:** Our approach provides the user with an (A) immersive room-scale VR experience from a first person perspective using the HTC Vive. The tracked movement of the user is mapped to a virtual avatar. We conduct a user study where the user is tasked with (B) moving objects (red box) in a virtual environment. (C) Virtual agents account for the presence of the avatar, and compute smooth, collision-free full-body movements.

of a single agent's movement with static and dynamic obstacles while the second study evaluated the plausibility of multi-agent interactions in dense spaces. In both studies, participants indicated significant preference for BAM in terms of both navigation and motion plausibility over prior methods.

- **Evaluating Multi-agent & Avatar Movement Interactions in Immersive Settings:** We evaluated the perceptual benefits of BAM in simulating multi-agent and avatar movement interactions in immersive settings using the HTC Vive and a 3.8 x 3.8 m tracking area. We found a significant preference for BAM compared to prior methods [12,36] in terms of perceived collision avoidance behaviors, as well as co-presence in the virtual environment.

2 Related Work

In this section, we give a brief overview of prior work in generating and evaluating passive agent-agent interactions and avatar-agent interactions.

2.1 Agent-Agent Interactions

There is extensive prior work on simulating 2D interactions and collision avoidance for multiple agents.

2D Interactions & Collision Avoidance: 2D multi-agent collision avoidance techniques can be broadly classified as macroscopic models and microscopic models. Macroscopic models [35] compute the aggregate motion of the agents by generating fields based on continuum theories of flows. Microscopic models, also called agent-based models, compute trajectories for each individual agent by decomposing the trajectory computation problem into two phases: global planning and local navigation. The global planners [16] compute a collision-free path through the environment considering only static obstacles. The local navigation algorithms [12, 30, 34, 36] adapt the local motion of each agent to avoid collisions with dynamic obstacles and other agents. The 2D trajectories generated by these approaches serve as input for full body motion synthesis or animation. There is extensive literature in computer graphics and animation on generating human like motion [39] including data-driven [28], procedural [6], and physics-based methods [11]. In most cases, the computed 2D trajectories do not account for human motion constraints, and may lead to collisions and other artefacts in the motion synthesis stage.

Non-locomotive Interactions: There is considerable work on synthesizing natural-looking interactions between virtual characters. Many of these methods rely on spatial discretization [10, 29, 41], or may not provide collision-free guarantees [14] for navigation-based behaviors. In contrast, our approach is more suitable for navigation-based behaviors and can compute collision-free trajectories for hundreds of agents in dense environments.

2.2 Avatar-Agent Interactions in VR

Prior research on avatar and single agent interactions in VR has shown that both behavior and appearance have a strong effect on the user's sense of presence in immersive virtual environments [1]. In fact, non-interactive virtual agents can negatively impact the user's sense of presence [32]. There is extensive work in embodied conversation agents (ECA) [38], in which a single animated anthropomorphic agent interacts with the user. There is also prior work in the context of interactive multi-agent simulations. This includes studying navigation behaviors and proxemics in controlled simulations [7, 18, 27], and modeling approach behaviors to enable face-to-face avatar-agent interactions [26]. Recent studies have also highlighted the role of collision avoidance on the part of virtual agents and its effect in increasing the sense of perceived realism, and overall presence in the simulation [15, 33]. However in both of these studies, collision avoidance is treated as a binary variable which is either enabled or disabled. Instead, we compare our proposed navigation algorithm BAM with prior collision avoidance methods and find that the perceived plausibility of the resulting interactions varies considerably with respect to the underlying collision avoidance algorithm.

2.3 Human Motion Constraints

Human locomotion consists of cyclic events generated by complex intersegmental co-ordination among different muscle groups. It is constrained by kinematic limits [5, 17] as well as postural and dynamic stability constraints [19, 40]. Some physics-based methods seek to account for many of these human motion constraints while computing full body movement for agents but are not suitable for interactive applications [23]. There are also interactive methods that make simplifying assumptions to reduce the complexity. However, these methods may not generate collision-free motion in dense spaces [28], or may use footstep based planning [2, 3] and impose strict constraints in motion synthesis to follow the generated footsteps which can impact the plausibility of the resulting motion. In

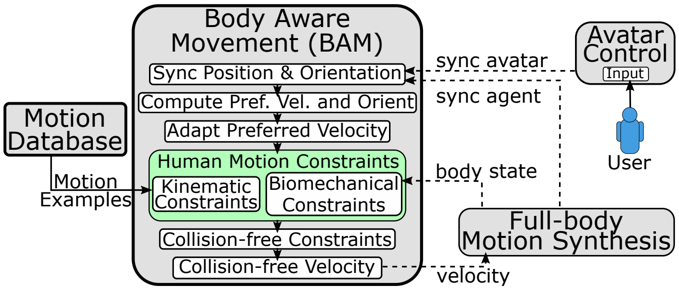

Figure 3: **Overview:** We highlight various components of our interactive algorithm to generate plausible collision-free movements for multiple human-like agents. The core of our approach is a 2D velocity computation algorithm called Body Aware Movement (BAM) which takes into account human motion constraints while efficiently computing feasible 2D trajectories. We couple BAM with a full-body motion synthesis algorithm and enable a tracked user to interact with the agents via their virtual avatar.

contrast, our approach (BAM) is less restrictive and can be combined with many existing motion synthesis techniques.

3 Overview

In this section, we introduce the notation and terminology used in the rest of the paper and give an overview of our approach.

3.1 Notation and Assumptions

We denote a scalar variable n with lower case letters, a vector $\mathbf{x}$ with a bold face lower case letter, a set $\mathcal{C}$ of entities with an upper case calligraphic letter. Each agent i in the simulator has an associated skeletal mesh, that is used for full-body motion synthesis. Each configuration $\mathbf{q}_i$ of the skeletal mesh is defined using the degrees-of-freedom (DOFs), including the 6-DOF root pose and the joint angles represented using n-dimensional vector space. We define the simulator state $\mathcal{S}$ as the union of all entities in the scene, including obstacles in the environment and the overall state space $\mathcal{Q} = \cup_i \mathbf{q}_i$.

We project the geometric representation of each skeletal mesh in $\mathbb{R}^n$ space to the $\mathbb{R}^2$ plane and bound it with a tightly fitted circle of radius r_i where r_i is equal to half of the shoulder width of the skeletal mesh. Each skeletal mesh with 6-DOF root joint $\mathbf{q}_i^{rt}$ is represented in the 2D multi-agent simulator by a disc of radius r_i positioned at $\mathbf{p}_i$, where $\mathbf{p}_i$ is simply the projection of the root joint $\mathbf{q}_i^{rt}$ on the 2D plane. The multi-agent navigation algorithm generates trajectories that correspond to the XY-projection of the 6-DOF root joint $\mathbf{q}_i^{rt}$ of the associated skeleton. These collision-free trajectories are represented as 2D time varying functions representing the position $\mathbf{p}_i(t)$ and velocity $\mathbf{v}_i(t)$.

3.2 Human Motion Constraints

Human locomotion requires complex control and coordination between multiple limb and body segments. It is characterized by the periodic movement, or 'gait cycle', of each foot from one position to the other, in conjunction with sufficient ground reaction forces [19, 40]. The gait cycle can be divided into six distinct periods that comprise the stance and swing phase [37]. For a gait cycle starting with the right foot leaving the ground, the first three phases i.e. initial double support, single limb stance, and second double support comprise the left stance phase; while the next three phases corresponding to initial swing, mid swing and terminal swing comprise the left swing phase. Despite its complexity, the full-body human motion corresponds to smooth, energy-efficient trajectories that satisfy several constraints, including:

- **Kinematic Motion Constraints:** Human movement is constrained, at the anatomical level, by the limits of joint rotations and accelerations [5, 17]. These constraints limit the set of spatial configurations, and consequentially the resulting trajectory.
- **Stability and Balance Constraints:** Stability, or control of balance, describes the dynamics of body posture to prevent falling. Balance while standing, or 'static stability', can be achieved by keeping the body's Center of Gravity (COG) within the 'base of support', i.e. the area defined by the ground contact points. However, the walking motion requires moving the COG outside the base of support and yet preventing the body from falling, a condition described as 'dynamic stability'. During gait initiation, one voluntarily initiates a forward fall to accelerate the COG ahead of the base of support. It is imperative that the swing foot is placed such that the COG returns to within the base of support, and thus prevents the fall [40].
- **Collision-free:** Humans are adept at finding energy-efficient collision-free paths, even in dense crowds.

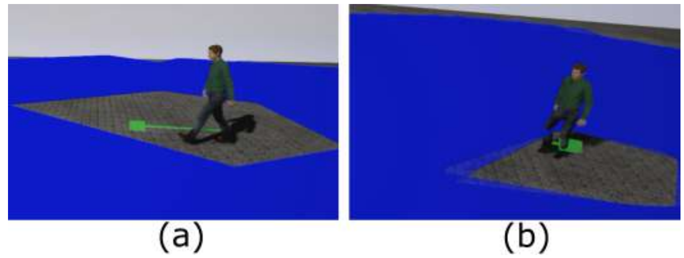

Figure 4: **Human Kinematic Constraints from Captured Data:** We derive kinematic constraints by analyzing a database of human motions and formulate them as half-plane velocity constraints, depicted in blue. (a) These constraints reflect the asymmetry in human motion during forward gait and (b) limit implausible velocities during turning.

Prior work in robotics on modeling these kinematic and stability constraints is limited to non-interactive applications and may not generate plausible motion. Instead of exactly modeling these constraints using the first principles, our velocity computation algorithm (BAM) accounts for these constraints in a manner that can generate plausible full-body motion and movement interactions for multiple agents (Section 4).

3.3 Human Motion Synthesis

Our goal is to reduce the dimensionality mismatch between 2D velocity computation and high-DOF motion synthesis that exists in prior two-step decomposition methods. Existing motion synthesis methods offer tradeoffs between naturalness and control of the synthesized motion [39] and often, infeasible 2D trajectories can result in artefacts which are magnified in immersive settings. BAM accounts for many human motion constraints and seeks to produce feasible 2D trajectories. It can be easily integrated with any existing motion synthesis technique capable of following a root velocity. However, the choice of the motion synthesis algorithm can significantly impact the perception of the virtual agents.

3.4 Simulating Multi-agent & Avatar Interactions

Figure 3 highlights our overall velocity computation and full-body motion algorithm and the various components. At every time-step, we first synchronize the 2D position and orientation of the agent with the root joint of the corresponding skeletal mesh. We use a Behavioral Finite State Machine (BFSM) to map the time and simulator state into a goal position $\mathbf{g}_i$ for agent i. Next, we employ a navigation mesh to plan a collision-free path with respect to static obstacles in the environment. The global planner maps the simulator state and the agent's goal position into a instantaneous preferred

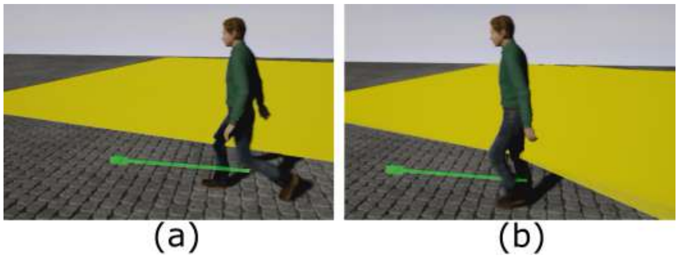

Figure 5: **Dynamic Stability Constraints** We compute half-plane velocity constraints (yellow) to account for the constrained motion of the center of gravity (COG) during a gait cycle [40]. (a) During gait initiation, as the left limb lifts off the ground, the COG is constrained by the stance foot. (b) As the left limb moves forward, the constraint is relaxed to allow the COG to move forward and away from the right limb.

velocity, $\mathbf{v}_i^{pref}$, and preferred orientation, o_i^d. By definition, the preferred velocity does not take into account the other agents and local conditions. We use a "social forces" based method [12] to adapt the preferred velocity to local dynamic conditions. The *adapted preferred velocity*, $\mathbf{v}_i^{pref*}$, is used to simply influence the agent's plan but does not account for human motion constraints (Section 3.2), nor does it provide sufficient collision avoidance guarantees. We then take into account interactions between an agent and an avatar, as well as agent-agent interactions.

3.4.1 Avatar-Agent Interactions

The user embodies a virtual avatar and is provided with a first person perspective of the virtual environment rendered via a head-mounted display (Figure 2). Our framework is agnostic to the specific input method used to track the user's movement in the virtual environment, and maps the input to the user's avatar. The embodied avatar is free to move around in the virtual environment populated with virtual agents. The virtual agents treat the avatar as a "special agent" i.e. they do not assume perfect reciprocity in collision avoidance.

3.4.2 Agent-Agent Interactions

The BAM algorithm is used to generate constraints on human motion based on the current state of the skeletal mesh. The motion constraints, combined with collision-free constraints, are solved to yield a feasible 2D velocity for the agent. Finally, we synthesize the full body motion for the agent.

4 BAM: Our Velocity Computation Algorithm

In this section, we present our novel 2D navigation algorithm that takes into account the current state of the skeletal mesh and many human motion constraints to generate 2D trajectories that are feasible for full-body motion synthesis.

4.1 Full-body Kinematic Constraints from Mocap

Prior work in kinematic analysis of human gait shows that gait parameters can vary not only across speeds and subjects, but even from trial to trial [5]. However, analysis across a large population of healthy adults suggests that, for a given speed, these parameters have limited variability and can be easily bounded [17]. To that end, we analyze a motion database with a broad sampling of human motions, and derive the bounds on kinematic constraints in the velocity space. Let each motion in the database be defined based on the average scalar speed v^f, turning rate ω^t, and strafing rate ω^s. We begin by first mapping the motion examples, where $\omega^s = 0$, to velocity space. For example, the motion $\mathbf{m} = \{v^f, \omega^t, 0\}$ can be mapped to the 2D velocity $\mathbf{v}^{motion} = \{v^x, v^y\}$ as:

$$\mathbf{u} = \{cos\omega^t, sin\omega^t\} \tag{1}$$

$$\mathbf{v}^{motion} = \frac{\mathbf{u}}{\|\mathbf{u}\|}.v^f, \tag{2}$$

where $\mathbf{v}^{motion}$ represents the average velocity of $\mathbf{m}$.

We wish to limit the set of feasible velocities to the space that is specified by the motion database. We formulate the space using half-plane constraints that can be combined with the other constraints and efficiently solved. It is likely that the wrapping polygon for the set of vertices is non-convex and thus, the corresponding half-plane constraints will cull velocities supported by the database. We overcome this by first computing a convex hull of the set of motion examples to yield a clockwise ordered set of n vertices $\mathcal{V} = \{\mathbf{v}_0^{motion}, \mathbf{v}_1^{motion}, ..., \mathbf{v}_{n-1}^{motion}\}$. Next, we compute half plane constraints for each each edge of the convex hull (Figure 4) and denote the set by $\mathcal{C}^{motion}$. For two consecutive vertices $\mathbf{v}_i^{motion} = \{v_i^x, v_i^y\}$ and $\mathbf{v}_{i+1}^{motion} = \{v_{i+1}^x, v_{i+1}^y\}$, the half-plane constraint, C_i^τ, can be defined by the point $\mathbf{p}$ and direction vector $\mathbf{d}$ given as:

$$\mathbf{p} = \mathbf{v}_i^{motion}, \tag{3}$$

$$\mathbf{d} = \frac{\mathbf{v}_i^{motion} - \mathbf{v}_{i+1}^{motion}}{\|\mathbf{v}_i^{motion} - \mathbf{v}_{i+1}^{motion}\|}. \tag{4}$$

By considering the convex hull of extreme motions in the parameterized space, we have likely included feasible velocities that are not contained within the motion database. These include the velocities, where the character is turning behind i.e., $|\omega^t| > 90$. We address this issue by dynamically adding half-plane constraints, if the preferred velocity, $\mathbf{v}^{pref}$, suggests a turn of more than 90 ° from the current orientation of the character. Based on this formulation, we account for the asymmetry in the human motion i.e., turning motion is more restrictive than forward motion.

4.2 Dynamic Stability Constraints

During gait initiation, lifting the swing limb greatly narrows the base of support, causing the Center of Gravity (COG) to move towards the stance limb. This causes lateral instability which is countered by means of an anticipated postural adjustment (APA) [19], wherein the center of pressure (COP) preemptively moves towards the swing limb. The lateral displacement of the COP is a result of the momentary loading of the swing limb. The COG moves closer to the base of support delineated by the stance foot. After unloading the stance limb, the COG accelerates forward and away from the stance limb towards the future position of the swing limb and traverses the medial border of each support foot [40].

We account for such dynamic stability in our 2D velocity computation algorithm by restricting the movement of the center of gravity of the articulated agent. We use the full-body motion synthesis algorithm to determine the stance leg, St, and swing leg, Sw, of the character at every time-step. Let $\mathbf{St}^{toe}$, and $\mathbf{Sw}^{heel}$ denote the positions of the toe joint of the stance limb, and heel joint of the swing limb respectively. During gait initiation, as the body transitions from double support to single support, we formulate a half-plane constraint H_{cog} that limits the set of feasible landing positions of the swing foot and prevents the COG from falling laterally which can cause dynamic instability [19, 40]. The spatial half-plane constraint H_{cog} is given as:

$$H_{cog} = \{\mathbf{p} | (\mathbf{p} - \mathbf{Sw}^{heel}).\mathbf{n}^b \geq 0\}, \tag{5}$$

where $\mathbf{n}^b$ denotes the normal to the vector $\mathbf{St}^{toe} - \mathbf{Sw}^{heel}$, outward with respect to the root position $\mathbf{q}^{rt}$.

During the gait initiation phase, this constraint limits the COG to remain within the base of support defined by the stance limb. As the swing limb approaches the mid swing point in the gait cycle (Section 4.1), the constraint H_{cog}, by definition, relaxes uniformly and allows the less restrictive forward movement of the COG [40]. The constraint is prevented from relaxing further during the terminal swing phase, which allows for greater maneuverability in the agent's trajectory. The spatial constraint H_{cog} can be easily mapped to a constraint in velocity space and is added to $\mathcal{C}^{motion}$ to yield the set of full body motion constraints, as shown in Figure 5.

4.3 Preferred Orientation

The 2D velocity computation algorithm sets the desired forward facing vector $\mathbf{f}_i^d$ for agent i as:

$$\mathbf{f}_i^d = \begin{cases} \frac{\mathbf{v}_i^{pref}}{\|\mathbf{v}_i^{pref}\|}, & \text{if } \frac{\mathbf{v}_i^{pref}}{\|\mathbf{v}_i^{pref}\|} \cdot \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} \geq 0, t_i^{strafe} < t^{strafeLim} \\ \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|}, & \text{otherwise} \end{cases}$$

where $\mathbf{v}_i^{pref}$ denotes the initial preferred velocity, $\mathbf{v}_i$ is the collision-free velocity. This formulation yields lateral movement, also called *strafing*, when $\mathbf{f}_i^d.\mathbf{v}_i \neq 0$. We track the contiguous time that the agent has been strafing t_i^{strafe} and limit it to a predefined threshold $t^{strafeLim}$. Finally, we set the desired orientation o_i^d to the angular representation of the unit vector $\mathbf{f}_i^d$.

4.4 Collision-free Velocity Computation

We use reciprocal velocity obstacle [36] to formulate collision avoidance constraints $\mathcal{C}_i^{collision_\tau}$ for the planning time τ. The intersection of half-plane constraints ($\mathcal{C}_i^{collision_\tau} \cap \mathcal{C}_i^{motion}$), yields the set of feasible velocities for agent i. Similar to [36], we use linear programming to find a new collision-free 2D velocity $\mathbf{v}_i$ from this set that minimizes the deviation from the adapted preferred velocity $\mathbf{v}_i^{pref*}$. Overall, this results in 2D trajectories that are amenable to full body motion synthesis and plausible simulation.

5 Implementation and Results

We highlight the results of our approach on several challenging benchmarks and discuss benefits over prior approaches including ORCA [36], Powerlaw [12] and Smartbody [28].

5.1 Metrics for Trajectory Evaluation

We leverage commonly used quantitative metrics to evaluate the trajectories generated by simulation algorithm. First, we use interval penetration depth to measure agent-agent collisions. Second, we compute a smoothness score for each simulation, measured by the complement of the average acceleration over all agents and frames. More details on these metrics are provided in the supplemental document.

5.2 Comparisons

We have compared the performance of our approach with prior methods. These include comparisons between BAM and prior 2D navigation algorithms (using two-step decomposition); and comparison of BAM with prior coupled human-agent simulation algorithms and systems.

Two-step Decomposition Methods: We compare BAM to prior 2D navigation methods that are based on two-step decomposition, including a social-forces based model, Powerlaw [12], and a velocity optimization model, ORCA [36]. The simulation time step was consistent across all three methods and was lower than 0.03 seconds for all benchmarks (Section 5.3). In each case, BAM results in fewer collisions and smoother trajectories. This is due to the fact that BAM accounts for many human motion constraints (Section 4) in 2D velocity computation. Moreover, BAM can automatically generate many emerging behaviors including commonly observed emergent behaviors such as lane formations, arching at bottlenecks, etc.

Coupled Approach: We compare our approach to Smartbody [28], an animation system that couples a 2D steering algorithm and a motion-blending-based technique. Smartbody prioritizes naturalness of the synthesized motion and is prone to collisions in medium to high density scenarios (Table 1). Moreover, it can lead to noisy trajectories in dense crossings, as is evident from the results in the supplemental document.

5.3 Benchmarks

We demonstrate the performance of our approach on five benchmark scenarios.

Antipodal Circle: In this benchmark, 17 agents are placed on the circumference of the circle with diametrically opposite goals Figure 1(a-b). This causes congestion at the center with a risk of head-on collisions. Agents simulated with BAM slow down appropriately as they approach the congested center, and smoothly navigate around each other resulting in fewer collisions, and smoother trajectories compared to prior techniques, as depicted in Table 1.

Crossing Flow: In this benchmark, two populations, each with 10 agents, cross each other orthogonally. Agents with BAM slow down appropriately as they approach the congested intersection, sidestep and find gaps to avoid each other. Thus, BAM algorithm results in fewer collisions as compared to ORCA and Powerlaw.

Bidirectional Flow: In this benchmark, two groups of agents approach each other at an angle of 180°. ORCA agents abruptly change velocities to avoid collisions leading to noisy trajectories. BAM agents attempt to smoothly navigate past each other which leads to slightly higher number of collisions (Table 1). Compared to Powerlaw, both BAM and ORCA agents depict emergent behaviors such as lane formation.

Tradeshow: We simulate a tradeshow scenario which is challenging due to the high number of obstacles and narrow passages (Figure 1(d)). Agents can be seen smoothly avoiding collisions with one other in the narrow passages, forming lanes and and often sidestepping to avoid each other.

Shibuya Crossing We simulate a busy street crossing, where agents are probabilistically assigned goal positions and must use the pedestrian walk lanes to navigate (Figure 1(e)). Agents simulated with BAM depict lane formation behaviors (depicted in the video). Our algorithm can simulate and render 50 agents at approx. 50-60 fps.

5.4 Performance

We have implemented our algorithm in C++ on an desktop PC with Intel Xeon E5-1620 v3 4-core processor, 16 GB of memory and Windows 10 OS. BAM can generate collision-free 2D trajectories for 100's of agents at interactive rates (see supplemental document), and the computation cost scales approx. linearly with the number of agents. Moreover, the full body motion synthesis system coupled with BAM is similar in performance to other data-driven approaches [28]. More details on the animation system are provided in the supplemental document. Overall, we can simulate and render 60+ full-body agents at 60+ fps. Our current implementation is not optimized and can be easily parallelized on multiple cores.

6 User Evaluation

We conducted a series of user evaluations to demonstrate the perceptual benefits of BAM compared to prior techniques in simulating agent-agent interactions in non-immersive settings, as well as avatar-agent interactions in immersive settings.

6.1 Single Agent & Agent-Agent Interactions in Non-Immersive Settings

We conducted two studies to evaluate the navigation and motion plausibility of simulations generated using BAM. The first study

Benchmark	Num. Agents	Collisions				Trajectory Smoothness			
		SB	ORCA	PL	BAM	SB	ORCA	PL	BAM
Crossing Flow	20	0.016	0.001	0.00003	0.00006	0.911	0.894	0.881	0.876
Bidirectional Flow	14	0.0	0.003	0.007	0.001	0.739	0.776	0.792	0.826
Antipodal Circle	17	0.079	0.012	0.060	0.009	0.765	0.819	0.86	0.851

Table 1: **Comparing BAM with prior algorithms:** We evaluate our 2D velocity algorithm, BAM, with prior two-step decomposed methods such as ORCA, and Powerlaw(PL), and a prior coupled method, Smartbody (SB). Both ORCA and Powerlaw are coupled with a motion blending based synthesis and simulated at a fixed time step. We compare (a) the number of agent-agent collisions, measured using interval penetration depth averaged over all frames and agents, and (b) the smoothness of the root joint, measured as the complement of averaged acceleration, over all frames and agents. Our algorithm, BAM, accounts for the full body pose during 2D planning leading to fewer collisions and smoother fully body motion. In a few cases, BAM has lower smoothness score due its prioritization of collision avoidance.

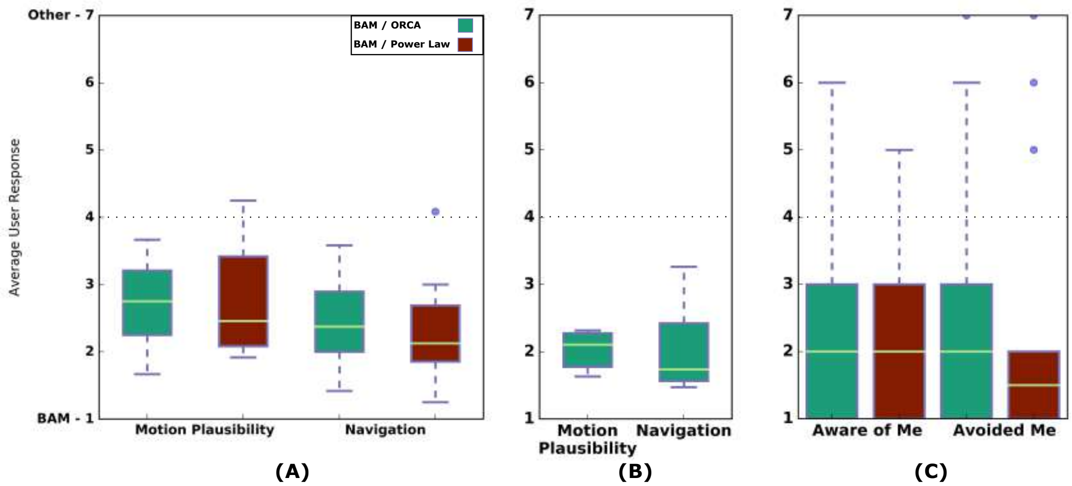

Figure 6: **Participant Preferences in User Studies**: (A) Participants clearly preferred our approach BAM compared to baseline methods, ORCA [36] and Powerlaw [12], in terms of both motion plausibility and navigation in the single agent non-immersive study. Mean preferences were 2.71, 2.75, 2.45 and 2.29 respectively where a rating of 1 indicates a strong preference for BAM, 7 indicates a strong preference for the other method and 4 indicates no preference. (B) Preference for BAM was even stronger in the non-immersive multi-agent study. Mean preferences were 2.02, and 2.04 respectively. (C) In the immersive study, participants significantly preferred BAM on questions of awareness and avoidance on part of the virtual agents, indicating higher co-presence. The mean preferences were (L-R) 2.44 ± 1.59, 2.50 ± 1.90, 2.25 ± 1.95, and 2.44 ± 1.46.

seeks to evaluate the movement plausibility of a single agent navigating in a dense scene with dynamic obstacles whereas the second study was designed to evaluate agent-agent interactions.

Experiment Goals & Expectations: We hypothesize that in both studies, agents simulated with BAM will exhibit fewer oscillations, fewer collisions, smoother trajectories and overall plausible motion and navigation behaviors compared to prior methods. Therefore, in both cases, participants will strongly prefer our approach to the baseline navigation algorithms.

Experimental Design: Both studies were conducted based on a within-subjects, paired-comparison design. Participants were shown two pre-recorded videos in a side-by-side comparison of our method BAM and one of set of baseline navigation algorithms. The order of scenes and the positioning of the methods was counterbalanced. The single-agent study depicted two viewpoints for every scene (close-up and top-down) whereas the multi-agent study depicted the perspective of an avatar in the crowd.

Comparison Methods: The single-agent study compared our algorithm BAM with two prior 2D navigation techniques: a velocity-optimization method, ORCA [36], and a social forces-based method, Powerlaw [12]. The agent-agent study compared BAM with ORCA only. All three 2D navigation methods were coupled with a motion-blending full-body animation approach which strictly follows the computed 2D trajectory.

Environments Two scenarios were used for the single agent study. Each comprised of one agent navigating to a goal in an environment with dynamic obstacles that avoided each other, but not the full-body agent. The scenarios were the *Antipodal circle* with 20 obstacles, and the *Crossing Flow* with 8 obstacles (Section 5.3) moving back and forth, orthogonal to the agent's preferred direction. The multi-agent study also comprised of two scenarios, each designed to increase the probability of agent-agent interactions. The first scenario, *Four-way crossing*, comprised of 4 groups, each with 6 agents, initialized at the ends of two perpendicular narrow passages meeting in the center. The second scenario, *Dense Unidirectional Flow*, comprised of 100 agents walking in a winding corridor.

Metrics: In both studies, participants indicated their preference for a method using a 7-point Likert scale, with 1 indicating strong preference for the method presented on the left, 7 indicating strong preference for the method presented on the right, and 4 indicating no preference. Participants responded to several questions, each representing an aspect of navigation or motion plausibility.

6.1.1 Results

The single-agent and multi-agent studies were taken by 12 and 19 participants respectively. For both studies, we tested each dimension of our questionnaire independently for reliability using the Cronbach's alpha test and and found good values, indicating that the questions were capturing the same aspect of the simulation. Moreover, we performed one-sample t-test comparing the mean of each dimension with a hypothetical mean of 4 (no preference) and found that the preferences were significant in each of the two-way comparisons. Overall, participants showed strong preference for BAM compared to both baseline methods on each dimension. Figure 6(A-B) details the preference values obtained for each dimension and comparison.

Single Agent Interactions Good reliability was determined for the navigation dimension ($\alpha = 0.845$) and the motion plausibility dimension ($\alpha = 0.857$). For the ORCA comparison, our responses demonstrate significant difference from the hypothetical mean on motion plausibility ($t(11) = -8.543, p < 0.001$) and navigation ($t(11) = -7.683, p < 0.001$). The Powerlaw comparison showed significant differences on both motion plausibility ($t(11) = -10.348, p < 0.001$) and navigation ($t(11) = -12.56, p < 0.001$).

Multi-agent Study Acceptable reliability was determined for the navigation dimension ($\alpha = 0.799$) and the motion plausibility dimension ($\alpha = 0.759$). Both dimensions were found to be significant, navigation ($t(18) = -10.715, p < 0.001$) and motion plausibility ($t(18) = -10.364, p < 0.001$). Overall, participants show strong preference for BAM on both dimensions, 2.04 ± 0.77 and 2.02 ± 0.81 respectively (Figure 6).

6.2 Multi-Agent & Avatar-Agent Interactions in Immersive Settings

We evaluate the ability of our approach to simulate plausible movement interactions between agents and the user's avatar in immersive settings. In this study, participants used an HTC Vive HMD and motion controllers and physically walked in a 3.8*x*3.8 m obstacle-free space. Their physical movement was mapped to their virtual avatar to ensure congruent proprioceptive sensations in the real and virtual world, at least in terms of motion. They used motion controllers to manipulate virtual objects, and were also provided haptic feedback during collisions with agents.

Experiment Goals and Expectations: Our hypothesis was two-fold. First, we hypothesized that that the perceived realism of agent-agent interactions with BAM highlighted by the non-immersive studies will stand true in immersive settings even if the user is merely a passive observer. Second, we hypothesized that BAM will result in more plausible avatar-agent interactions, leading to higher co-presence. Henceforth, we refer to the former as the *agent-agent condition*, and the latter as the *avatar-agent condition*.

Experimental Design: This study was conducted based on a within-subjects, paired-comparison design. For each condition, participants first interacted with two simulations back to back with a fixed exposure time. The two simulations were generated using BAM and one of two baseline methods, in counterbalanced order. After these two simulations, participants answered a set of questions before moving on to the next set of two simulations comparing BAM with the other baseline method.

Comparison Methods: Similar to the non-immersive studies (Section 6.1), we evaluated BAM against both baseline methods, ORCA and Powerlaw.

Metrics: The questions were derived from commonly used presence questionnaires to address aspects of presence, co-presence, motion and navigation plausibility. The response format was similar to the one used in Section 6.1.

Scenarios: In the agent-agent condition, participants were asked to be stationary and observed the Antipodal Circle from a distance. Each method was run three times before moving onto the next method. In the avatar-agent condition, participants were tasked with picking a colored block and dropping it on a matching colored tile on the virtual floor. The blocks and the matching tile were separated by approximately 5.25 m, inducing the participant to physically walk back and forth in the real world. The direct path of the participant was orthogonal to a unidirectional flow, increasing the probability of interactions with virtual agents.

6.3 Results

The immersive user-study was taken by 18 participants, 13 male, with a mean age of 23.06 ± 4.41 years. Analysis across dimensions was not consistently significant across baseline methods, therefore we reserve our discussion to 2 questions for which significance was observed for both compared methods. For each question, a one-sample t-test was performed against a hypothetical mean of 4 (no preference). On the question "In which simulation were the virtual characters more aware of your presence?", the comparison against ORCA was shown to be significant ($t(15) = -3.930, p < 0.001$) as well as Powerlaw ($t(15) = -3.162, p = 0.006$). On the question "In which simulation were the virtual characters avoiding you more?" comparisons against ORCA ($t(15) = -4.283, p = 0.001$) and Powerlaw ($t(15) = -3.591, p = 0.003$) were significant. Overall, participants show a strong preference for BAM on each question, as detailed in Figure 6.

6.4 Discussion

We performed a series of studies to evaluate the benefits of BAM under a variety of conditions. In non-immersive conditions, participants clearly preferred BAM over prior methods when evaluating a single agent's navigation and motion plausibility. Preference for BAM was even stronger in the case of multi-agent interactions.

Preference for BAM was also evident in the case of avatar-agent interactions in virtual reality. Participants reported the virtual agents avoided them more, and felt that the agents were more aware of their presence as compared to simulations with prior methods. These results indicate that the underlying 2D navigation algorithm can significantly impact the user's experience when interacting with virtual agents. This is a new finding which extends previous studies that explore the impact of collision avoidance as a binary condition [15, 33]. While these responses were statistically significant, responses to other questions such as naturalness of motion were not consistently significant across the two compared baseline methods. Similarly, responses for the agent-agent condition were also not consistently significant. This can be due to a number of reasons. First, our approach so far has been focused only on collision avoidance which may not be the socially appropriate response in close interactions. Second, we had only 18 participants and there was significant variation in their level of prior experience with VR. Moreover, the full-body motion generated by the coupled animiation system is prone of artefacts such as foot-skating. The fact that we observed significant differences consistently in the avatar-agent condition, and not in the agent-agent condition merits further investigation. Overall, responses were largely in favour of our navigation algorithm as depicted in Figure 6.

7 Conclusion, Limitations & Future Work

We present an interactive approach to generate plausible movement for human-like agents. Unlike prior interactive methods that use a two-step decomposition approach, we present a new velocity computation algorithm (BAM), which takes into account human motion constraints. Our approach can be easily combined with many interactive human motion synthesis methods. It can be easily parallelized on multi-core processors and is suitable for large-scale interactive crowd simulation [22]. We have also evaluated its perceptual benefits by performing multiple user studies with single-agent and multi-agent environments.

Our approach has some limitations. Given the overall goal of interactive performance, our BAM algorithm may not be accurately account for all human motion constraints. It may be possible to generate more natural looking motions using data-driven or physics-based simulation algorithms, but they tend to be more computationally expensive. Moreover, the overall results depend on the choice of the coupled motion synthesis algorithm and its ability to generate natural looking animation.

There are many avenues for future work. Besides overcoming the limitations, we would like to improve the fidelity as well as performance of the human motion synthesis algorithm. It could be useful to evaluate the benefits of the movements generated by our algorithm in terms of realistic human perception of crowds, adding different gestures [8, 21, 25], and also motion styles based on high level attributes such as personality. Our 2D algorithm, BAM, can also be combined with other human motion synthesis algorithms [29, 41]. We would also like to conduct additional quantitative evaluations of the human motion, including energy usage etc. [13]. In addition, we would like to explore the integration of more complex dynamical stability constraints with BAM. Finally, we would like to extend our algorithm to shared virtual environments with multiple avatars and social VR. [22]

ACKNOWLEDGMENTS

This work is supported in part by ARO grants W911NF16-1-0085 and W911NF-17-1-0181, and Intel.

REFERENCES

[1] J. N. Bailenson, K. Swinth, C. Hoyt, S. Persky, A. Dimov, and J. Blascovich. The independent and interactive effects of embodied-agent appearance and behavior on self-report, cognitive, and behavioral markers of copresence in immersive virtual environments. *Presence: Teleoper. Virtual Environ.*, 14(4):379–393, Aug. 2005.

[2] A. Beacco, N. Pelechano, M. Kapadia, and N. I. Badler. Footstep parameterized motion blending using barycentric coordinates. *Computers & Graphics*, 47:105–112, 2015.

[3] G. Berseth, M. Kapadia, and P. Faloutsos. Robust space-time footsteps for agent-based steering. *Computer Animation and Virtual Worlds*, 2015.

[4] J. Blascovich, J. Loomis, A. C. Beall, K. R. Swinth, C. L. Hoyt, and J. N. Bailenson. Immersive virtual environment technology as a methodological tool for social psychology. *Psychological Inquiry*, 13(2):103–124, 2002.

[5] N. A. Borghese, L. Bianchi, and F. Lacquaniti. Kinematic determinants of human locomotion. *Journal of physiology*, 494(3):863–879, 1996.

[6] A. Bruderlin and T. Calvert. Interactive animation of personalized human locomotion. In *Proc. of Graphics Interface*, pp. 17–23, 1993.

[7] J. Bruneau, A. H. Olivier, and J. Pettré. Going through, going around: A study on individual avoidance of groups. *IEEE Transactions on Visualization and Computer Graphics*, 21(4):520–528, April 2015.

[8] C. Ennis, C. Peters, and C. O'Sullivan. Perceptual effects of scene context and viewpoint for virtual pedestrian crowds. *ACM Transactions on Applied Perception (TAP)*, 8:10, 2011.

[9] L. Hoyet, A.-H. Olivier, R. Kulpa, and J. Pettré. Perceptual effect of shoulder motions on crowd animations. *ACM Trans. Graph.*, 35(4):53:1–53:10, July 2016. doi: 10.1145/2897824.2925931

[10] K. Hyun, M. Kim, Y. Hwang, and J. Lee. Tiling motion patches. *IEEE Trans. Vis. Comput. Graph*, 19(11), 2013.

[11] S. Jain, Y. Ye, and C. K. Liu. Optimization-based interactive motion synthesis. *ACM Trans. Graph.*, 28(1):10:1–10:12, Feb. 2009.

[12] I. Karamouzas, B. Skinner, and S. J. Guy. Universal power law governing pedestrian interactions. *Physical Review Letters*, 113(23):238701, 2014.

[13] I. Karamouzas, N. Sohre, R. Narain, and S. J. Guy. Implicit crowds: Optimization integrator for robust crowd simulation. *ACM Transactions on Graphics (TOG)*, 36(4):136, 2017.

[14] M. Kim, K. Hyun, J. Kim, and J. Lee. Synchronized multi-character motion editing. In *ACM Transactions on Graphics (TOG)*, vol. 28, p. 79, 2009.

[15] M. Kyriakou, X. Pan, and Y. Chrysanthou. Interaction with virtual crowd in immersive and semi-immersive virtual reality systems. *Computer Animation and Virtual Worlds*, 2016. CAVW-16-0002.R1.

[16] S. LaValle. *Planning Algorithms*. Cambridge, 2006.

[17] J. L. Lelas, G. J. Merriman, P. O. Riley, and D. Kerrigan. Predicting peak kinematic and kinetic parameters from gait speed. *Gait —& Posture*, 17(2):106 – 112, 2003. doi: 10.1016/S0966-6362(02)00060-7

[18] J. Llobera, B. Spanlang, G. Ruffini, and M. Slater. Proxemics with multiple dynamic characters in an immersive virtual environment. *ACM Trans. Appl. Percept.*, 8(1):3:1–3:12, Nov. 2010.

[19] M.-L. Mille and L. Mouchnino. Are human anticipatory postural adjustments affected by a modification of the initial position of the center of gravity? *Neuroscience letters*, 242(2):61–64, 1998.

[20] A. Nagendran, R. Pillat, A. Kavanaugh, G. Welch, and C. Hughes. A unified framework for individualized avatar-based interactions. *Presence: Teleoperators and Virtual Environments*, 23(2):109–132, 2014.

[21] S. Narang, A. Best, T. Randhavane, A. Shapiro, and D. Manocha. PedVR: Simulating gaze-based interactions between a real user and virtual crowds. *Proc. of ACM VRST*, 2016.

[22] S. Narang, T. Randhavane, A. Best, and D. Manocha. Fbcrowd: Interactive multi-agent simulation with coupled collision avoidance and human motion synthesis. Technical report, University of North Carolina-Chapel Hill, 2016. Available at http://gamma.cs.unc.edu/pedvr.

[23] C. Park, A. Best, S. Narang, and D. Manocha. Simulating high-dof human-like agents using hierarchical feedback planner. In *Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology*, pp. 153–162. ACM, 2015.

[24] N. Pelechano, J. M. Allbeck, M. Kapadia, and N. I. Badler. *Simulating Heterogeneous Crowds with Interactive Behaviors*. CRC Press, 2016.

[25] N. Pelechano, C. Stocker, J. Allbeck, and N. Badler. Being a part of the crowd: towards validating vr crowds using presence. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*, pp. 136–142, 2008.

[26] T. Randhavane, A. Bera, and D. Manocha. F2fcrowds: Planning agent movements to enable face-to-face interactions. *Presence: Teleoperators and Virtual Environments*, 26, 2017.

[27] F. A. Sanz, A. H. Olivier, G. Bruder, J. Pettré, and A. Lécuyer. Virtual proxemics: Locomotion in the presence of obstacles in large immersive projection environments. In *2015 IEEE Virtual Reality (VR)*, pp. 75–80, March 2015.

[28] A. Shapiro. Building a character animation system. In J. Allbeck and P. Faloutsos, eds., *Motion in Games*, vol. 7060 of *Lecture Notes in Computer Science*, pp. 98–109, 2011.

[29] H. P. Shum, T. Komura, M. Shiraishi, and S. Yamazaki. Interaction patches for multi-character animation. In *ACM Transactions on Graphics (TOG)*, vol. 27, p. 114, 2008.

[30] S. Singh, M. Kapadia, P. Faloutsos, and G. Reinman. An open framework for developing, evaluating, and sharing steering algorithms. In *Proceedings of the 2nd International Workshop on Motion in Games*, pp. 158–169, 2009.

[31] M. Slater, C. Guger, G. Edlinger, R. Leeb, G. Pfurtscheller, A. Antley, M. Garau, A. Brogni, and D. Friedman. Analysis of physiological responses to a social situation in an immersive virtual environment. *Presence: Teleoperators and Virtual Environments*, 15(5):553–569, 2006.

[32] M. Slater, B. Lotto, M. M. Arnold, and M. V. Sanchez-Vives. How we experience immersive virtual environments: the concept of presence and its measurement. *Anuario de psicología*, 40(2), 2009.

[33] N. Sohre, C. Mackin, V. Interrante, and S. J. Guy. Evaluating collision avoidance effects on discomfort in virtual environments. In *Virtual Humans and Crowds for Immersive Environments (VHCIE), 2017 IEEE*, pp. 1–5. IEEE, 2017.

[34] S. Stuvel, N. Magnenat-Thalmann, D. Thalmann, A. F. van der Stappen, and A. Egges. Torso crowds. *IEEE Transactions on Visualization and Computer Graphics*, 2016.

[35] A. Treuille, S. Cooper, and Z. Popović. Continuum crowds. In *Proc. of ACM SIGGRAPH*, pp. 1160–1168, 2006.

[36] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha. Reciprocal n-body collision avoidance. In *Inter. Symp. on Robotics Research*, pp. 3–19, 2011.

[37] C. L. Vaughan, B. L. Davis, and J. C. O'connor. *Dynamics of human gait*. Human Kinetics Publishers Champaign, Illinois, 1992.

[38] A. M. Von Der Pütten, N. C. Krämer, and J. Gratch. 1 who´ s there? can a virtual agent really elicit social presence? 2009.

[39] H. Welbergen, B. Basten, A. Egges, Z. Ruttkay, and M. Overmars. Real time character animation: A trade-off between naturalness and control. *Computer Graphics Forum*, 29(8), 2010.

[40] D. A. Winter. Human balance and posture control during standing and walking. *Gait & posture*, 3(4):193–214, 1995.

[41] J. Won, K. Lee, C. O'Sullivan, J. K. Hodgins, and J. Lee. Generating and ranking diverse multi-character interactions. *ACM Transactions on Graphics (TOG)*, 33(6):219, 2014.