Interactive Modeling, Simulation and Control of Large-Scale Crowds and Traffic

Ming C. Lin[1], Stephen Guy[1], Rahul Narain[1], Jason Sewall[1], Sachin Patil[1], Jatin Chhugani[2], Abhinav Golas[1], Jur van den Berg[1], Sean Curtis[1], David Wilkie[1], Paul Merrell[1], Changkyu Kim[2], Nadathur Satish[2], Pradeep Dubey[2], and Dinesh Manocha[1]

[1] Department of Computer Science
University of North Carolina
Chapel Hill, NC, U.S.A.
[2] Throughput Computing Lab
Intel Corporation
Santa Clara, CA, U.S.A.
{lin,sjguy,narain,sewall,sachin,golas,
berg,seanc,wilkie,pmerrell,dm}@cs.unc.edu
{jatin.chhugani,changkyu.kim,nadathur.rajagopalan.satish,
pradeep.dubey}@intel.com

Abstract. We survey some of our recent work on interactive modeling, simulation, and control of large-scale crowds and traffic for urban scenes. The driving applications of our work include real-time simulation for computer games, virtual environments, and avatar-based online 3D social networks. We also present some preliminary results and proof-of-concept demonstrations.

Keywords: Velocity Obstacle, Multi-Agent Simulation.

1 Introduction

Aggregates of numerous entities, such as a group of people and fleet of vehicles, form complex systems that exhibit interesting biological, social, cultural, and spatial patterns observed in nature and in society. Modeling of the collective behaviors remains an open research challenge in computer graphics, robotics, architecture, physics, psychology, social sciences, and civil and traffic engineering, as complex systems often exhibit distinct characteristics, such as emergent behaviors, self-organization, and pattern formation, due to multi-scale interactions among individuals and groups of individuals. Despite of decades of observation and studies, collective behaviors are particularly not well understood for groups with *non-uniform spatial distribution* and *heterogeneous behavior characteristics*, such as pedestrian and vehicle traffic in urban scenes, evacuation flows in complex structures, and coupled human-natural systems.

In this paper, we survey our recent works on addressing the problem of modeling, simulating, and directing the large-scale inidvidual agents in complex dynamic environments. Our works focus on (1) understanding the general principles

A. Egges, R. Geraerts, and M. Overmars (Eds.): MIG 2009, LNCS 5884, pp. 94–103, 2009.

that characterize the behavior of macroscopic dynamics for aggregate systems, and (2) identifying simplified, approximate models that capture their movement under varying conditions. This set of algorithms and techniques can potentially provide computational tools for motion planning, designing game plays, and developing simulation engines that are simple and easy to use for authoring games with numerous agents.

We present several complementary approaches for local collision avoidance and global navigation of multiple virtual entities in an interactive environment. These methods can be applied to interactive crowd simulation, motion synthesis, and coordination of multiple autonomous agents in computer games. These include (a) a new local collision avoidence algorithm between multiple agents [GCK+09]; (b) a novel formulation to model aggregate dynamics of dense crowds [NGCL09]; (c) an efficient simulation method of continuum traffic [SWML09]; and (d) an effective approach to direct and control virtual crowds using navigational fields [PvdBC+09]. Each of these methods are targetted at modeling, simulating, and directing the interaction between human crowds or vehicle traffic at various scales and densities. They are complementary techniques that can be integrated to capture aggregate dynamics of a complex urban scene.

2 ClearPath

We present a highly parallel and robust collision avoidance approach, *ClearPath*, for multi-agent simulation [GCK+09]. Our formulation extends and generalizes the concept of *velocity obstacles* (VO) [FS98] for local collision avoidance among dynamic obstacles. We use an efficient velocity-obstacle based formulation that can be combined with any underlying multi-agent simulation. We show that local collision avoidance computations can be reduced to solving a quadratic optimization problem that minimizes the change in underlying velocity of each agent subject to non-collision constraints.

We introduce a polynomial-time algorithm for agents to compute collision-free, 2D motion in a distributed manner. We pose the local collision avoidance problem for N agents as a combinatorial optimization problem. We extend the VO formulation by imposing additional constraints that can guarantee collision avoidance for each agent during the discrete interval. The Fast Velocity Obstacle (FVO) is defined using a total of four constraints. The *two* boundary cone constraints of the FVO are same as that of RVO [vdBLM08, vdBPS+08]:

$$\mathrm{FVO}_{L}{}_{B}^{A}(\mathbf{v}) = \phi(\mathbf{v}, (\mathbf{v}_A + \mathbf{v}_B)/2, \mathbf{p}^{\perp}_{AB left}) \geq 0$$
$$\mathrm{FVO}_{R}{}_{B}^{A}(\mathbf{v}) = \phi(\mathbf{v}, (\mathbf{v}_A + \mathbf{v}_B)/2, \mathbf{p}^{\perp}_{AB right}) \geq 0$$

Additionally, we impose *two* more types of constraints:

Type-I Constraint - Finite time interval: We only guarantee collision avoidance for the duration ΔT. We compute a finite subset of the RVO cone that corresponds to the forbidden velocities that could lead to collisions in ΔT. The truncated cone. Due to efficiency reasons, we replace $\gamma_{AB}(\mathbf{v})$ with aconservative linear approximation $\Gamma_{AB}(\mathbf{v})$. This additional constraint is represented as $\text{FVO}_T{}^A_B(\mathbf{v}) = \Gamma_{AB}(\mathbf{v}) = \lambda\left(M - \widehat{\mathbf{p}^{\perp}_{AB}} \times \eta,\ \mathbf{p}^{\perp}_{AB}\right)$, where

$$\eta = \tan\left(\sin^{-1}\frac{\mathbf{r}_A + \mathbf{r}_B}{|\mathbf{p}_{AB}|}\right) \times (|\mathbf{p}_{AB}| - (\mathbf{r}_A + \mathbf{r}_B)), \text{and}$$

$$M = (|\mathbf{p}_{AB}| - (\mathbf{r}_A + \mathbf{r}_B)) \times \widehat{\mathbf{p}_{AB}} + \frac{\mathbf{v}_A + \mathbf{v}_B}{2}$$

Type-II Constraint - Consistent velocity orientation: Without loss of generality, we force each agent to choose its *right* side and impose this constraint as $\text{FVO}_C{}^A_B(\mathbf{v}) = (\phi(\mathbf{v}, \mathbf{p}_A, \mathbf{p}^{\perp}_{AB}) \leq 0)$. This is also a conservative formulation to guarantee collision-free motion.

Any feasible solution to all of constraints, which are separately formulated for each agent, will guarantee collision avoidance. We solve this problem as a quadratic optimization function with non-convex linear constraints for each agent. It can be shown to be NP-Hard [GCK$^+$09] for non-constant dimensions via reduction to quadratic integer programming. It has a polynomial time solution when the dimensionality of the constrains is constant – two in our case. In practice, ClearPath is more than one order of magnitude faster than prior velocity-obstacle based methods.

P-ClearPath: Parallel Collision Avoidance: We further show that ClearPath is amenable to data-parallelism and thread-level parallelism on commodity processors and suggest a parallel extension. The resulting parallel extension, P-ClearPath, exploits the structure of our optimization algorithm and architectural capabilities such as gather/scatter and pack/unpack to provide improved data-parallel scalability. ClearPath operates on a per-agent basis in a distributed manner, finding each agent's nearest neighbors and computes a collison-free velocity w.r.t. those neighbors. There are *two* fundamental ways of exploring Data-Level parallelism (henceforth referred to as DLP).

Intra-Agent: Consider Fig. 1(a). For each agent, we explore DLP within the ClearPath computation. Since the agents operate in 2D, they can perform their X and Y coordinate updates in a SIMD fashion. This approach does not scale to wider SIMD widths.

Inter-Agent: Operate on multiple agents at a time, with each agent occupying a slot in the SIMD computation. This approach is scalable to larger SIMD widths, but needs to handle the following two issues:

1. Non-contiguous data access: In order to operate on multiple agents, ClearPath requires *gathering* their obstacle data structure into a contiguous location in memory. After computing the collision-free velocity, the results need to be *scattered* back to their respective non-contiguous locations. Such data accesses become a performance bottleneck without efficient *gather/scatter* operations.

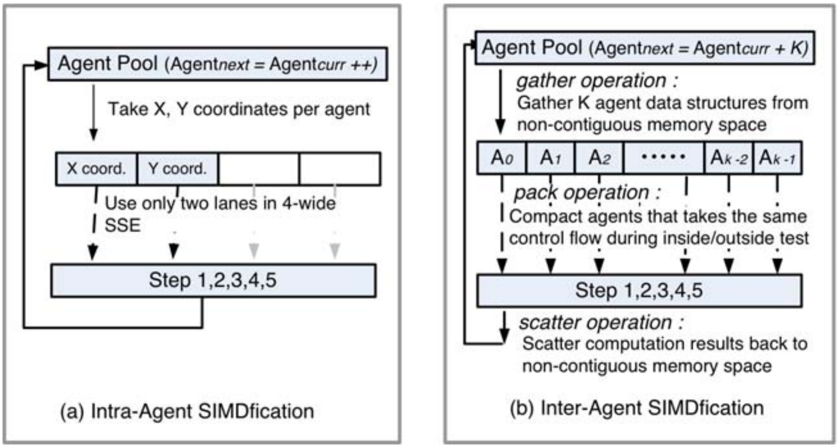


Fig. 1. Data-Parallel Computations: (a) Intra-Agent SIMDfication for SSE; (b) Inter-Agent SIMDfication for wide SIMD

2. Incoherent branching: Multiple agents within a SIMD register may take divergent paths. This degrades SIMD performance, and is a big performance limiter during intersection computations and inside/outside tests. One or more of the agents may terminate early, while the remaining ones may still be performing comparisons.

Current SSE architectures on commodity CPUs do not have efficient instructions to resolve the above two problems. Hence, we used the intra-object SIMDfication approach and obtain only moderate speedups. P-ClearPath adopts the Inter-agent approach, and performs computation on $\mathcal{K}$ agents together. Fig. 1(b) shows a detailed mapping of the various steps in ClearPath algorithm. For collision-free velocity computation, each agent A_i is given as input its neighboring velocity obstacles (truncated cones) and the desired velocity. For more detail, see [GCK+09].

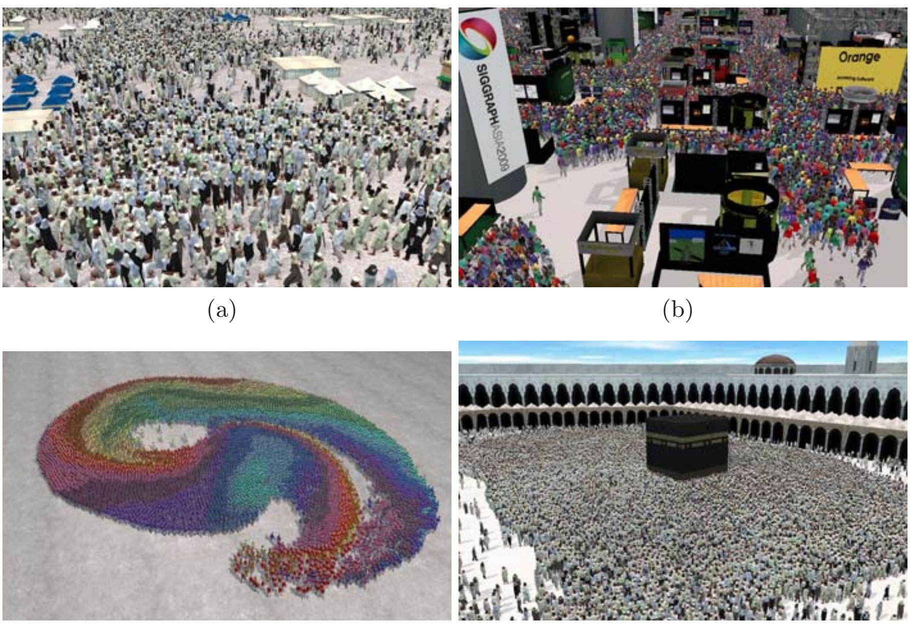
Results: We evaluate its performance in various scenarios on different platforms like current multi-core CPUs and the upcoming many-core processor codenamed Larrabee. In practice, P-ClearPath demonstrates 8-15X speedup on a conventional quad-core processor over prior VO-based algorithms on similar platforms. When executed on a Larrabee simulator with 32-64 cores, P-ClearPath achieves additional speedup of up to 15X, resulting in up to 100-200X speedup over prior VO-based approaches.

Overall, for simple game-like scenarios with a few hundred agents, P-ClearPath takes about 2.5 milliseconds on a single Larrabee core, while a complex simulation with few hundreds of thousands of heterogeneous agents takes only 35 milliseconds on the simulated 64-core Larrabee processor. To the best of our knowledge, P-ClearPath is the first scalable collision avoidance algorithm for multi-agent simulations with a few hundred thousand agents at interactive rates.

3 Hybrid Crowds

Dense crowds exhibit a low interpersonal distance and a corresponding loss of individual freedom of motion. This observation suggests that the behavior of such crowds may be modeled efficiently on a coarser level, treating its motion as the flow of a single aggregate system. Based on such an abstraction, we develop a novel inter-agent avoidance model which decouples the computational cost of local planning from the number of agents, allowing very large-scale crowds consisting of hundreds of thousands of agents to be simulated scalably at interactive rates.

Our method combines a Lagrangian representation of individuals with a coarser Eulerian crowd model, thus capturing both the discrete motion of each agent and the macroscopic flow of the crowd. In dense crowds, the finite spatial extent occupied by humans becomes a significant factor. This effect introduces new challenges, as the flow varies from freely compressible when the density is low to incompressible when the agents are close together. This characteristic is shared by many other dynamical systems consisting of numerous objects of finite size, including granular materials, hair, and dense traffic. We propose a new mathematical formulation to model the dynamics of such aggregate systems in a principled way, which is detailed in [NGCL09].



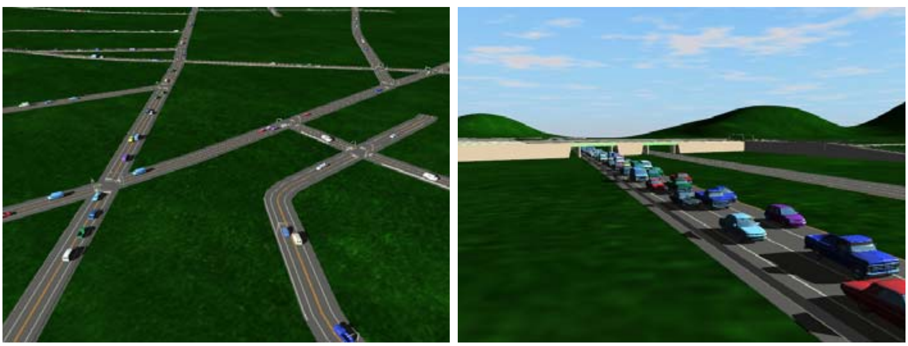
(a) (b) (c) (d)

Fig. 2. Some examples of large, dense crowds simulated with our technique. (a) 100,000 pilgrims moving through a campsite. (b) 80,000 people on a trade show floor. (c) 10,000 agents attempt to cross over to the opposite points of a circle and meet in the middle, before forming a vortex to resolve the deadlock. (d) 25,000 pilgrims with heterogeneous goals in a mosque.

Results: Fig. 2 shows some of our results for dense crowds of up to 100,000 agents closely packed in complex scenes simulated at interactive rates with our techniques. We measured the performance of our algorithm on an Intel Core i7-965 machine at 3.2 GHz with 6 GB of RAM. Even with very large numbers of agents, we can achieve close to interactive performance. Our method supports general scenarios with independent, heterogeneous agents; the number of unique goals has no effect on the performance.

4 Continuum Traffic

There exists a vast amount of literature on modeling and simulation of traffic flows, with existing traffic simulation techniques generally focusing on either microscopic [CSS00] or macroscopic behaviors ([AR00, Zha02]). However, little attention has been paid to the possibility of extending macroscopic models to produce detailed 3D animations and visualization of traffic flows.



(a) (b)

Fig. 3. Images from our traffic simulator: (a) A bird's-eye view of a busy intersection; (b) a traffic jam occurring on a highway

In this section, we present an overview of a method for efficient simulations of large-scale, real-world networks of traffic using macroscopic (continuum-level) dynamics that uses microscopic, individual vehicle information to display each vehicle [SWML09]. We model the movement of many vehicles with a single computational cell — while individual vehicle information facilitates visual representation and allows for per-vehicle information be incorporated into the large-scale simulation.

Our technique produces detailed, interactive simulation of traffic flows on a wide variety of road types, including urban streets, multi-lane highways, and winding rural roads. Our approach extends a continuum, per-lane flow model by introducing a novel model of lane changes that uses our discrete vehicle representation. We introduce new techniques for describing the behavior of vehicles at intersections. This simulation technique is able to effectively utilize the processing power of many-core shared memory architectures for scalable simulation.

We also validate the results of our traffic simulations with real-world observed traffic flows.

Results: We have tested our technique on a number of synthetic road networks, a digital representation of a modest-sized city, and on a high-volume traffic on a short segment of freeway. Our city data set features sparse rural roads, short dense arrangements punctuated with stoplights, and a highway running through it. This data set was automatically generated from publicly-available GIS data from the US Census Bureau's TIGER database, which is widely available and has excellent coverage of the roads in the United States. Fig. 3 shows vehicles from our simulation of the six-lane stretch of I-80 freeway in Emeryville, California and the results of our simulation.

Our approach leverages an inexpensive continuum solve with a lightweight, particle-like visual representation of each vehicle to produce realistic motion synthesis of vehicle motion on large networks. Rather than be input-sensitive to the number of vehicles in the network, the compute expense of our technique is directly related to the number of computational cells that must be update each step. The small city data data set shown in Fig. 3 covers an area over 23 square kilometers and contains over 180 km with of lanes. The scenarios in the accompanying video have as many as 2500 vehicles in the network at a given time, and these simulations can be peformed on a single core of an Intel Core2 process at frame rates in excess of 120 video frames/second.

5 Directing Crwods Using Navigation Fields

Most existing agent-based systems assume that each agent is an independent decision making entity. Some of the methods also focus on group-level behaviors and complex rules for decision making. The problem with these approaches is that interactions of an agent with other agents or with the environment are often performed at a local level and can sometimes result in undesirable macroscopic behaviors. Due to the complex inter-agent interactions and multi-agent collision avoidance, it is often difficult to generatedesired crowd movements or motion patterns that follow the local rules.

In this work, we address the problem of directing the flow of agents in a simulation and interactively control a simulation at runtime [PvdBC$^+$09]. Our approach is mainly designed for goal-directed multi-agent systems, where each agent has knowledge of theenvironment and a desired goal position at each step of the simulation. The goal position for each agent can be computed from a higher-level objective and can also dynamically change during the simulation.

Our approach uses discretized *guidance fields* to direct the agents. Based on these inputs, we compute a unified, goal-directed, smooth *navigation field* that avoids collisions with the obstacles in the environment. The guidance fields can be edited by a user to interactively control the trajectories of the agents in an ongoing simulation, while guaranteeing that their individual objectives are attained. The microscopic behaviors, such as local collision avoidance, personal

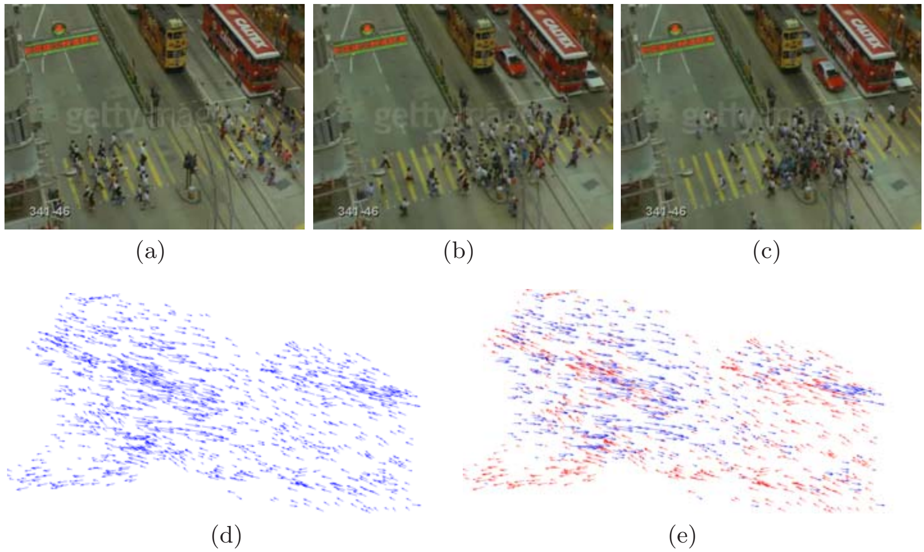


Fig. 4. Motion patterns detected in a video of a crosswalk in Hong Kong. (a,b,c) Frames from the original video; (d) Motion flow field; (e) Detected motion patterns.

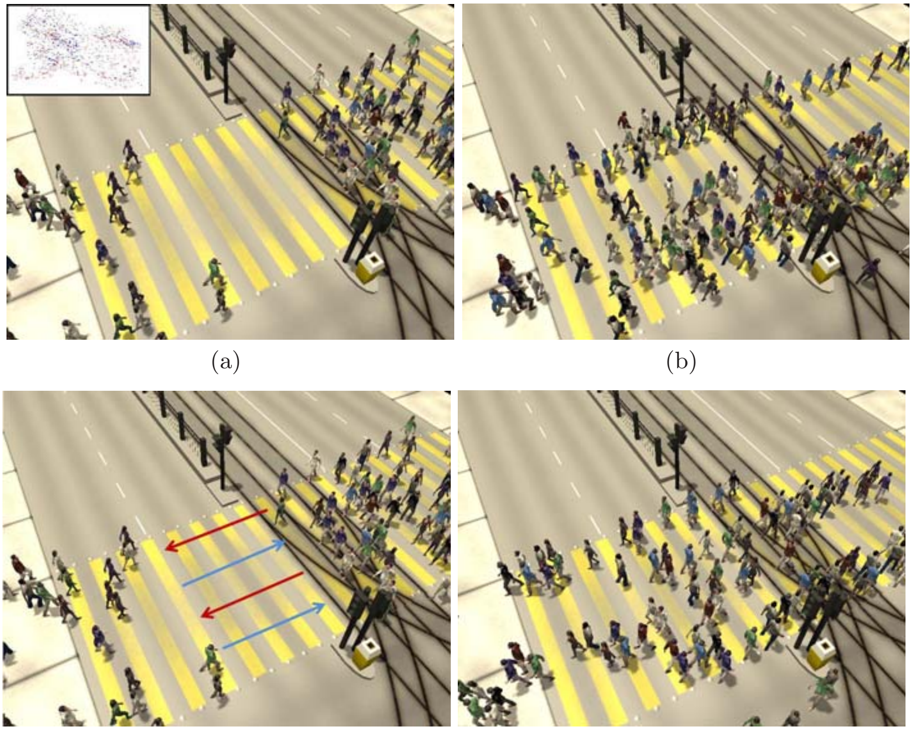
space and communication between individual agents, are governed by the underlying agent-based simulation algorithm.

Results: This approach is general and applicable to a variety of existing agent-based methods. We illustrate the usefulness of our approach in the context of several simulation scenarios shown in Fig. 5. The user edits the simulation by specifying guidance fields, that are either drawn by the user or extracted from a video sequence (Fig. 4). The overall approach can be useful from both artistic and data-driven perspectives, as it allows the user to interactively model some macroscopic phenomena and group dynamics.

6 Discussion

In this paper, we have presented a brief survey of algorithms for real-time collision avoidance, modeling, simulation, and control of multiple virtual agents, including both humans and vehicles, in dynamic scenes. We also demonstrated their applications on interactive crowd simulation for games and virtual environments.

Among these methods, ClearPath is designed to take advantages of upcoming many-core architectures to achieve real-time collision avoidance of hundreds of thousands of discrete agents at interactive rates. In contrast, our hybrid representation for large-scale crowds are better suited for highly dense scenarios, capable of simulating upto millions of indvidiual agents in nearly real time by exploiting their constrained movement. But, its implementation on many-core architecture is yet to be explored.



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

Fig. 5. Crosswalk Simulation: (a) Video-based motion patterns from Figure 4(e) used as guidance fields; (b) Agent motion generated by the navigation fields. (c) Sketch-based guidance fields to specify lane formation in the simulation; (d) Lane formation generated by goal-directed navigation fields.

Our simulation technique for modeling continuum traffic is a high-level, macroscopic method for simulating a wide variety of traffic condition. We would like to further develop the idea of a coupled continuum-discrete traffic simulation to take advantage of the unique advantages each has to offer; continuum models are fast and can handle large areas inexpensively, while discrete models are capable of describing more individualistic behavior. Another natural extension would be an integration of traffic and crowd simulations to model a urban scene in all aspects.

Our current approach to direct and control crowds using navigation fields is general and versatile. It can use any low-level collision avoidance algorithm. The system allows the user to specify the navigation fields by either sketching paths directly in the scene via an intuitive authoring interface or by importing motion flow fields extracted automatically from crowd video footage. This technique is complementary to other methods and most suitable for interactive editing and testing during the conceptual game design.

Acknowledgments. The research presented here is supported in part by Army Research Office, Intel, National Science Foundation, and DARPA/RDECOM.

References

[AR00] Aw, A., Rascle, M.: Resurrection of second order models of traffic flow. SIAM Journal of Applied Mathmatics (60), 916–938 (2000)

[CSS00] Chowdhury, D., Santen, L., Schadschneider, A.: Statistical Physics of Vehicular Traffic and Some Related Systems. Physics Reports 329, 199 (2000)

[FS98] Fiorini, P., Shiller, Z.: Motion planning in dynamic environments using velocity obstacles. International Journal on Robotics Research 17(7), 760–772 (1998)

[GCK+09] Guy, S., Chhugani, J., Kim, C., Satish, N., Lin, M.C., Manocha, D., Dubey, P.: Clearpath: Highly parallel collision avoidance for multi-agent simulation. In: Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation (2009)

[NGCL09] Narain, R., Golas, A., Curtis, S., Lin, M.C.: Aggregate dynamics for dense crowd simulation. ACM Transactions on Graphics (Proc. of ACM SIGGRAPH Asia) (2009)

[PvdBC+09] Patil, S., van den Berg, J., Curtis, S., Lin, M.C., Manocha, D.: Directing crowd simulations using navigation fields. Technical report, Department of Computer Science, University of North Carolina (May 2009)

[SWML09] Sewall, J., Welkie, D., Merrell, P., Lin, M.C.: Continuum traffic simulation. Technical report, Department of Computer Science, University of North Carolina (May 2009)

[vdBLM08] van den Berg, J., Lin, M.C., Manocha, D.: Reciprocal velocity obstacles for realtime multi-agent navigation. In: Proc. of IEEE Conference on Robotics and Automation, pp. 1928–1935 (2008)

[vdBPS+08] van den Berg, J., Patil, S., Seawall, J., Manocha, D., Lin, M.C.: Interactive navigation of individual agents in crowded environments. Proc. of ACM Symposium on Interactive 3D Graphics and Games, 139–147 (2008)

[Zha02] Zhang, H.M.: A non-equilibrium traffic model deviod of gas-like behavior. Transportation Research B (36), 275–290 (2002)