WIREs COGNITIVE SCIENCE

**Article type: Overview**

# Article title: Navigation and Steering for Autonomous Virtual Humans

**First author: Full name and affiliation; plus email address if corresponding author**
Mubbasir Kapadia, University of Pennsylvania.
mubbasir@seas.upen.edu

**Second author: Full name and affiliation; plus email address if corresponding author**
Norman I. Badler, University of Pennsylvania.
badler@seas.upenn.edu

## Abstract

The ever-increasing applicability of interactive virtual worlds in industry and academia has given rise to the need for robust, versatile autonomous virtual humans to inject life into these environments. There are two fundamental problems that must be addressed to produce functional, purposeful autonomous populaces: (1) Navigation: finding a collision-free global path from an agent's start position to its target in large complex environments, and (2) Steering: moving an agent along the path while avoiding static and dynamic threats such as other agents.

In this review, we survey the large body of contributions in steering and navigation for autonomous agents in dynamic virtual worlds. We describe the benefits and limitations of different proposed solutions and identify potential future research directions to meet the needs for the next generation of interactive virtual world applications.

## Introduction

Immersive virtual worlds have quickly come to the forefront in both industry and academia with their applicability being realized in a wide variety of areas from education, security, urban design, cinematic content creation, and interactive entertainment. A key aspect of immersion in virtual environments is the use of autonomous virtual humans to inject life into these worlds. This has led to a strong emphasis on improving the believability and intelligence of these agents. The ultimate far-reaching goal that is still a considerable challenge: to simulate functional, purposeful autonomous virtual humans in complex, dynamic environments for hundreds, even thousands of agents in real-time.

Research in virtual human simulation addresses challenges in one or more of the following areas:

1. *Navigation:* Navigation is the process of finding a collision-free global path from the start position of an agent to its target. Navigation in arbitrarily large, complex environments

requires an agent to be equipped with a mental model that provides some semantically meaningful geometric representation of the world around it.
2. *Steering:* The steering layer interfaces with navigation to move an agent along the planned path, while avoiding both static and dynamic threats such as other agents.
3. *Locomotion:* A locomotion module (e.g., (1)) animates a virtual human to follow the trajectory output by steering, taking into account the locomotion capabilities of the agent.
4. *Scenario Authoring:* An authoring framework (2), (3) provides an appropriate interface for users to specify and automatically generate complex multi-agent interactions which satisfy global narrative constraints.
5. *Visualization:* The animated virtual humans along with the rich virtual environments are visualized at an appropriate level of detail depending upon the needs of the application.

We refer the readers to comprehensive reviews (4), (5), (6) that address the different facets of virtual human research. In this book chapter, we focus on *autonomous agents*: entities that perform -- on their own -- navigation and steering behaviors in complex, dynamic virtual environments for interactive applications. First, we review the different approaches proposed for efficiently computing global paths in large virtual worlds. Second, we present a taxonomy of steering research used to move agents along the computed paths while performing local collision-avoidance. Third, we present an abstraction of the different steering and navigation techniques in an effort to understand the benefits and limitations of each.  Finally, we conclude with a brief discussion on future work.

## NAVIGATION

The efficient computation of free movement paths is an important requirement in the simulation of interactive virtual worlds populated with autonomous virtual humans. The computed paths should take into account different optimality criteria such as path length, time, and energy expended in traversing the path. Efficiency of computation is also of paramount importance since environments may be extremely large and arbitrarily complex. Depending upon application requirements, a variety of path planning approaches have been proposed, some of which are described below.

### Potential Fields

Potential Fields (7), (8), (9)  generate a global field for the entire landscape where the potential gradient is contingent upon the presence of obstacles and distance to goal. The method has some problems, such as local minima where the agents could get stuck and never reach the goal. Dynamic potential fields (10) have been used to integrate global navigation with moving obstacles and people, efficiently solving the motion of large crowds without the need for explicit collision avoidance.

### Uniform Grid Representation of the Environment

Discrete search methods such as A* (11) applied to grid representations (12), (13) are robust and simple to implement, with strict guarantees on optimality and completeness of solution. Hence, they represent a popular and widely used method for path planning in commercial systems such as games. However, the performance and quality of the obtained paths greatly depend on the resolution of the discretization with coarse resolution producing low-quality paths and fine resolution grids proving to be prohibitive for real-time applications. The remainder of this section describes the use of roadmaps, connectivity graphs, and other pre-computed data structures that facilitate efficient pathfinding in arbitrarily complex environments.

**Probabilistic Roadmaps**

Probabilistic Roadmaps (PRMs) (14) compute a simplified representation of free space by sampling configurations at random. PRMs have been used to generate paths for large groups of agents (15) to simulate group behaviors such as homing and flocking.

**Cell and Portal Graphs**

Cell and Portal Graphs represent a method of abstracting the geometry of virtual environments. The nodes in the graph map to navigable regions in the environment such as rooms while the portals represent entry points such as doors. The problem of navigation in a CPG is reduced to getting from one node to another using a sequence of nodes and portals (16), (17).

**Topological Representation of the Environment**

Corridor maps (18), (19) use a medial axis based method to represent free space by a graph where edges correspond to collision-free corridors. Each edge of the graph encodes a local path along with the maximum clearance radius which can be used to find paths with arbitrary clearance. Visibility graphs (20), (21) are used for computing shortest paths among polygonal obstacles with a search time of $O(n^2)$ for n segments (22). The work in (23) pre-computes a Voronoi representation of the static environment for path planning. The pre-computation phase has a computational complexity of $O(n^2 \log n)$ but the query phase is very efficiency with $O(n \log n)$ running time. The work in (24) computes a convex cell sub-division of free space while preserving local bottleneck information to represent the topological connectivity of 3D environments. Other approaches including elastic roadmaps (25) and multi-agent navigation graphs (26) have been proposed to efficiently compute near-optimal paths in virtual environments.

Navigation meshes (27) are widely used in the gaming industry to pre-compute an approximation of the free area, represented as a triangulation, which is queried for efficient pathfinding. The work in (28) proposes a method of finding optimal paths of arbitrary clearance directly from the triangulation. The proposed method introduces a new local clearance property which facilitates the efficient computation of path clearance in a triangulated mesh. This allows locally shortest paths of arbitrarily clearance to be computed in $O(n \log n)$ optimal time, and an extended search algorithm is also presented for determining global optimality.

**Space-Time Planning**

Space-time models (29), (30) combine space and time into a single construct by representing space as three dimensions and time as the fourth dimension. Space time planning exploits the inherent advantage of having information in time to predict collisions in the future. These models improve the fidelity of agent behavior at the cost of an additional dimension which greatly increases the search space, incurring a considerable overhead.

Shapiro et al. (31) demonstrate planning-based control in the extremely high dimensional space of a fully articulated virtual human to demonstrate complex object manipulation and dynamic interactions. Randomized sampling-based search reduces the computational overhead of a planner by sacrificing optimality; however, this approach is still offline. The work in (32) demonstrates the use of space-time planning for global agent navigation in deterministically changing virtual environments. Lopez et al. (33) represents moving obstacles as interaction volumes to find temporal paths in dynamically changing environments with unknown evolution. The ability to accurately query the space-time position of dynamic objects mitigates the need for constant re-planning. However, these approaches can only handle a very small number of agents at interactive rates.

### Real-Time Planning in Dynamic Environments

Traditional planners (11) provide optimality and completeness guarantees but are computationally intensive and need to re-plan from scratch when the search space changes. Here, we refer to search techniques in artificial intelligence which focus on real-time path planning in dynamically changing environments.

Anytime planning algorithms (34) try to find the best plan they can within the amount of time available to them. They quickly find a sub-optimal path and then improve the path while time is available. This helps interleave planning and execution. Incremental approaches (35) compute a shortest path from the current position to the goal given the current perceived world state. As the agent moves along the current path, the plan is efficiently updated to incorporate new information and dynamic changes in the environment. Anytime D* (36) is an anytime incremental algorithm with provable bounds on sub-optimality. This algorithm provides anytime solution guarantees and can incrementally repair its solution to accommodate dynamic events.

## STEERING

Steering is the layer of intelligence that interfaces with navigation to move an agent along its planned path by performing a series of successive local searches, taking into consideration locomotion constraints such as turning capabilities and limits on movement velocity, as well as dynamic objects in the environment such as other agents. We broadly classify steering approaches into the following overlapping categories:

### Centralized Approaches

These techniques focus on the system as a whole, modelling the characteristics of the flow of the crowd rather than the local interactions between individual agents. Regression models (37) use statistically established relations to predict crowd flows under specific circumstances such as stairs or corridors. Route choice models (38) describe pedestrian way-finding to maximize the utility of their trip which is described as a function of parameters such as travel time, etc. Queuing approaches (39) use Markov chain models to describe how pedestrians move from one node of the network to another. Henderson models crowds as fluids (40) and uses fluid dynamics to describe crowd flow. The work in (10) uses dynamic potential fields to efficiently simulate crowd flow without explicit collision avoidance. Narain et al. (41) presents a hybrid approach using a dual representation of discrete agents as well as a monolithic continuous system to capture aggregate crowd dynamics.

Such models are of value in computing macroscopic simulations involving thousands of agents (e.g. stadium evacuation scenarios, urban simulations, etc.). However, these approaches cannot simulate complex local agent interactions which are crucial in modelling functional, purposeful autonomous virtual humans for next-generation applications.

The remainder of this section summarizes the large body of work in agent-based steering which model each agent as an independent being, having its own state and goal, which performs collision avoidance with static obstacles, reacts to dynamic threats in the environment and steers its way to its target. The interactions between agents and the environment result in the emergence of macroscopic crowd behavior such as lane formations and agents cooperating to evacuate through narrow egress points.

**Particle Dynamics**

The simplest approach is to model the agent as a particle and to simulate crowds by using basic particle dynamics (42) (43). This approach works extremely fast, but the heuristic reactions of this technique are not sufficient to create realistic pedestrian steering behaviors.

**Social Force Models**

The social force model (44), (45), (46) simulates hypothetical forces such as repulsion, attraction, friction and dissipation for each agent to simulate pedestrians. Recent extensions (47), (48), (49) apply evolutionary models to determine optimal parameter specifications for the social force model, introduce a self-stopping mechanism to prevent agents from continuously pushing each other and, include an elliptical volume exclusion of pedestrians which shows better agreement with empirical data. The work in (50) demonstrates the influence of social interactions between groups of individuals on crowd dynamics. While these models describe pedestrian behaviors more realistically, they are designed to be as simple and efficient as possible.

**Cellular Automata Models**

Cellular Automata (51), (52) discretize space and time to define a simple, yet elegant mathematical model for a physical simulation. A cellular automaton comprises a uniform lattice where each cell stores its own state. The automaton evolves by each cell changing its state as a function of the state of neighboring cells. The rules describing this evolution govern the behavior of the simulation. Cellular Automata models (53), (54), (55), although fast and simple to implement, do not allow for contact between agents, and can only model local homogeneous interactions between agents. Often the restriction to one agent per cell is too strong to produce realistic crowd behaviors.

**Rule-based Approaches**

Rule-based approaches (43), (42), (24), (56), (57) use various conditions and heuristics to identify the exact situation of an agent.  Once the situation is understood, then the rules compute what steering decision the agent should make. These approaches are limited by the situations that are modelled by the rule developer and cannot generalize to handle the intractable space of all possible steering challenges.

**Data-Driven Approaches**

Research in data-driven steering primarily focuses on generating local-space samples from observations of real people. These observations are used to create databases or serve as training data to learn computational models which are queried to emulate real human behavior. The work in (58) uses manually tracked trajectories from video data which are compiled into a database and queried at runtime. The agent's surroundings are used as the index to retrieve trajectories which are used for simulation. The work in (59) uses a more constrained state space of discretized slices around an agent and focuses on recreating group dynamics. Torrens et al. (60) define a separate state space consisting of a discretized view frustum for environmental navigation. Data-driven solutions are a natural fit to expanding an algorithm to handle new situations, but suffer from two main limitations. Trained models (60), (61) sacrifice specificity to fit a single, monolithic model which generalizes over all the training data. The lossless alternative of database querying (58), (59) leads to an unwieldy amount of data which becomes impractical to store and search.

**Predictive Approaches**

The works of (62), (63) use predictions in the space-time domain to perform steering in environments populated with dynamic threats. The prediction and avoidance of potential collision threats results in more realistic steering behaviors. Reciprocal Velocity Obstacles (64) builds upon the concept of velocity obstacles to propose a new method for local reactive collision avoidance using the assumption that neighboring agents will adopt similar collision-avoidance behaviors. This facilitates the efficient simulation of dense crowds without the need for explicit inter-agent communication. Recent extensions use proxy agents (65) to model agent interactions, allowing an agent to exercise influence greater than its physical properties – facilitating richer and more complex group interactions. ClearPath (66) extends velocity obstacles to formulate collision-free navigation as a quadratic optimization problem which can be parallelized to achieve an order of magnitude performance gain.

The work in (13), (67) uses a variable-resolution egocentric representation to model agent perception and affordance for crowd simulation. This approach provides the benefit of implicit space-time planning without the overhead of modelling time as an extra dimension. Pettre and colleagues (68) use experimental data to derive a predictive model of collision avoidance in virtual walkers. Ondrej et al. (69) demonstrate a vision-based approach which uses a synthetic sense of vision to predict the trajectories of neighboring agents while making steering decisions. Singh et al. (12) presents a hybrid framework that chooses the between reactive, predictive, and planning based steering policies depending on the current situation of the agent.

**Commercial and Open-Source Solutions**

Commercial and open-source software provide complete steering and navigation solutions using variations of the aforementioned techniques. Massive (70) is a popular commercial software that provides an end-to-end but off-line solution for authoring, simulating, and rendering large crowds of virtual humans and is geared towards cinematic content creation. Recast (71) is an open-source software framework that automatically computes navigation meshes for arbitrarily complex 3D virtual worlds and includes Detour, a predictive technique for local-collision avoidance. SpirOps (72) uses a *drive-oriented* approach to artificial intelligence, mimicking the decision-making process of real humans, while allowing users to incrementally create behaviors in their simulations. SteerSuite

(73) is an open-source crowd simulator that provides developers with tools to create their own steering techniques.

## AN ABSTRACTION OF STEERING AND NAVIGATION TECHNIQUES

The previous sections summarize a large variety of steering and navigation techniques that have been proposed to simulate autonomous agents. Navigation techniques perform global searches to produce a coarse trajectory from an agent's start position to its target location, accounting for static geometry, and rely on offline pre-computations, efficient representations of *free space,* and bounds on sub-optimality to meet real-time constraints. Steering approaches perform successive local searches to smoothly follow the global path, while accounting for the most imminent dynamic threats. The simulation of multiple interacting agents in large-scale crowds requires the explicit or implicit modeling of navigation (global pathfinding) and steering (local collision-avoidance). Regardless of the underlying techniques used in these approaches, most steering and navigation frameworks fall under the umbrella of the following abstraction:

1. **Environment Representation.** The environment representation in all steering approaches is either explicitly or implicitly modelled as a field. The field may be sparse, storing only information in relevant spatial regions, or dense, storing information at all points within the spatial region. The field maybe local to the agent or global, spanning the entire area of the environment. The data contained in the field represents a modeling of the environment (static and dynamic obstacles, other agents, etc.) and is queried by agents to acquire perceptual information. Finally, the environment may be computed on-demand or by lazy evaluation to represent the current configuration of the virtual world.
2. **Decision-Making.** The decision making process in these approaches can be generalized to be a search. Some approaches may model different levels of search ranging from a long term, one-time search to the goal to a per time-step search that optimizes the velocity that steers the agent to the goal. This search process can be customized for the particular steering technique used or could be modeled as a generalized best-first search technique.
3. **Locomotion.** Finally, the agent is usually modeled as a disc with locomotion constraints governing its maximum speed of movement, rate of turning etc. The steering decision is one of the following: 1) a velocity vector, 2) target direction and speed, or 3) a force which moves the disc along the path. A pedestrian model is then superimposed on top of the disc and the mapping from a point mass to an animated human is done as a post-processing step.

Most field-based techniques (7), (8), (9) focus their complexity on the environment representation so that decision-making is usually a simple spatial query. The resolution of the field greatly impacts the simulation results with finer resolution providing high-fidelity control at a greater computational overhead.

Grid-based representations don't capture the topological connectivity of navigable regions in the environment and represent all regions of the environment at uniform resolution.  Approaches such as visibility graphs (22), corridor maps (19) , and navigation meshes (27) rely on offline pre-computation to better represent free space, facilitating efficient navigation queries. The computational complexity of the search process can be greatly accelerated by relaxing the constraints on optimality, usually at a negligible compromise to path quality (28).

Adding time into the search domain (32), (33) produces global planning solutions that account for dynamic obstacles such as other moving agents, at a considerable computational overhead. The exploration of anytime and dynamic search techniques (34), (35), (36), widely used in robotics is a promising avenue of exploration for integrating global navigation and local collision-avoidance in a single search framework.

Approaches such as regression models (38), route-choice models (39), and dynamic potential fields (10) use a unified representation of the virtual world, and centralized decision-making to simulate large crowds, but are unable to model local agent interactions and produce homogeneous behavior.

Particle-dynamic approaches (43), (42) and social force models (44) have simplified grid-based environment representations and decision-making is governed using a simple set of rules that result in emergent behavior. The ad-hoc combination of atomic rules which are used to describe widely different steering challenges can lead to unexpected or erroneous behavior, and these approaches generally rely on the robustness of their locomotion interface (forces) and collision-avoidance technique to produce a plausible simulation. Artificial life techniques (74) model the environment as a hierarchical collection of discretized fields, which support fast perceptual queries for global as well as local navigation. The decision-making performs a search to compute a long-term plan for navigation and utilizes heuristics for reactive decisions to avoid dynamic threats.

Data-driven approaches (59), (58), (60), (61) rely on real-world data to automatically identify the most optimal steering decision in different situations, and can potentially scale well to handle new situations. However, the availability of training data that covers a representative set of steering challenges remains a fundamental challenge.

The approaches described above produce reactive behaviors where agents react to the presence of imminent threats, with an effective search horizon of one. Predictive approaches (63), (64) forecast the outcome of the simulation in the future to choose the next steering decision by assuming that agents travel along a straight path determined by their current velocity. The work in (13) uses dense, local fields with variable resolution to model the environment. The field is computed at each time step and the decision-making process is a recursive breadth-first search to plan a trajectory of high affordance from current position to goal. Planning-based approaches (75) account for the non-linear trajectories of neighboring agents in order to make an optimal steering decision, enabling the solution of complex agent interactions such as potential deadlocks, at an exponential increase in computational cost. The search depth is usually limited to a fixed horizon to meet real-time constraints. Hybrid approaches (12) combine reactive rules, linear prediction, and space-time planning to produce robust steering agents that can react to simple scenarios such as oncoming or crossing threats, predictively solve more complex situations involving multiple agents, and use space-time planning to solve challenging deadlock situations.

A majority of crowd approaches models agents as simple particles with velocity or force-based control in an effort to simulate large crowds at interactive rates. However, this simplification produces artifacts such as oscillations or discrepancies in the mapping of the particle trajectories to animated virtual characters. The work in (75) introduces a bio-mechanically based footstep locomotion model for steering agents to provide a tighter coupling with character animation.

## Conclusion

Virtual human research has seen a dramatic rise in recent years with growing applicability in a large number of areas ranging from interactive entertainment, to education and urban simulations. Navigation and steering are two fundamental problems that must be addressed in an effort to simulate functional, purposeful but autonomous populaces that inhabit large, complex virtual environments.

Current approaches first compute a global path that only accounts for static geometry. The steering layer is then responsible for following the path while factoring in dynamic world events. The next generation of interactive applications require high-fidelity navigation in non-deterministic dynamic virtual worlds. The environments and agents may be constantly changing due to unpredictable events (e.g., other agents, dynamic obstacles, and human input), potentially invalidating computed navigation paths. Hence, future navigation systems must be able to efficiently repair any global solution by accounting for the current dynamic world state.

Different steering approaches model different aspects of human steering and can each only handle a fraction of the entire space of challenging scenarios (76) that agents encounter in dynamic environments. In an effort to better understand the benefits and limitations of different approaches, and to identify scenarios that cannot be solved by any steering approach, we must develop tools (77), (78), (79) that facilitate the objective evaluation of steering and navigation techniques.

The majority of steering algorithms output a force or velocity vector for the locomotion system to follow. This simplistic interface does not model the constraints and capabilities of human-like movement. Future approaches must investigate the use of bio-mechanically plausible models (75) that provide a tighter coupling between steering and human locomotion.

The union of the subsets of scenarios that different steering algorithms can handle is far greater than what each algorithm can individually solve. A promising avenue of future work is the exploration of hybrid approaches that can choose the most appropriate steering and navigation policy based on an understanding of the agent's current situation.

## References


1. *Interactive motion generation from examples.* **Arikan, Okan and Forsyth, D. A.** 2002, ACM SIGGRAPH , pp. 483-490.

2. *A Behavior Authoring Framework for Multi-Actor Simulations.* **Kapadia, Mubbasir and Singh, Shawn and Reinman, Glenn and Faloutsos, Petros.** 2011, IEEE Computer Graphics and Applications (CGA), Special Issue.

3. *Event-Centric Control for Background Agents.* **Badler, Alexander Shoulson and Norman I.** 2011. Fourth International Conference on Interactive Digital Storytelling, ICIDS 2011 .

4. **Pelechano, Nuria and Allbeck , Jan M. and Badler, Norman I.** *Virtual Crowds: Methods, Simulation, and Control.* s.l. : Morgan & Claypool Publishers, 2008.

5. **Thalmann, Daniel.** *Crowd Simulation.* s.l. : Wiley Encyclopedia of Computer Science and Engineering, 2008.

6. **Kapadia, Mubbasir.** *Simulating Autonomous Virtual Humans.* s.l. : LAP Lambert Academic Publishing, 2012.

7. *Global path planning using artificial potential fields.* **Warren, C.W.** 1989, Robotics and Automation, Vol. 1, pp. 316-321.

8. *Multiple robot path coordination using artificial potential fields.* **Warren, C.W.** 1990, Robotics and Automation, Vol. 1, pp. 500-505.

9. *Potential Field Navigation of High Speed Unmanned Ground Vehicles on Uneven Terrain.* **Shimoda, S. and Kuroda, Y. and Iagnemma, K.** 2005, Robotics and Automation, pp. 2828-2833.

10. *Continuum crowds.* **Treuille, Adrien and Cooper,Seth and Popovic, Zoran.** 2006, ACM Transactions of Graphics, Vol. 25, pp. 1160-1168.

11. *A Formal Basis for the Heuristic Determination of Minimum Cost Paths.* **Hart, P.E. and Nilsson, N.J. and Raphael, B.** 1968, IEEE Transactions on Systems Science and Cybernetics, Vol. 4, pp. 100-107.

12. *A modular framework for adaptive agent-based steering.* **Singh, Shawn and Kapadia, Mubbasir and Hewlett, Billy and Reinman, Glenn and Faloutsos, Petros.** 2011. Symposium on Interactive 3D Graphics and Games.

13. *Egocentric affordance fields in pedestrian steering.* **Kapadia, Mubbasir and Singh, Shawn and Hewlett, William and Faloutsos, Petros.** 2009. Symposium on Interactive 3D graphics and games.

14. *Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces.* **Kavraki, Lydia E. and Svestka, Petr and Latombe, Jean-claude and Overmars, Mark H.** 1996, EEE Transactions on Robotics and Automation, Vol. 12, pp. 566-580.

15. *Roadmap-Based Flocking for Complex Environments.* **Bayazit, O. Burchan.** 2004. Proc. 10th Pacific Conference on Computer Graphics and Applications (PG'02).

16. **Pettre, J and Laumond, J P and Thalmann, D.** A navigation graph for real-time crowd animation on multilayered and uneven terrain. *First International Workshop on Crowd Simulation.* 2005, pp. 81-90.

17. *Controlling individual agents in high-density crowd simulation.* **Pelechano, Nuria and Allbeck, Jan M. and Badler, Norman I.** 2007. SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation.

18. *The corridoor map method: a general framework for real-time high-quality path planning.* **Geraerts R. and Overmars, Mark H.** 2007, Computer Animation and Virtual Worlds, pp. 107-119.

19. *Planning short paths with clearance using explicit corridors .* **R., Geraerts.** 2010. IEEE International Conference on Robotics and Automation.

20. *An algorithm for planning collision-free paths among polyhedral obstacles.* **A, Lozano-Perez T. and Wesley M.** 1979, Communications of ACM, pp. 560-570.

21. *Computational Geometry: Algorithms and Applications.* **De Berg M., Cheong O., van Kreveld M.** 2008, Springer.

22. *New methods for computing visibility graphs.* **E., Overmars M.H and Welze.** 1988. Proceedings of Fourth Annual Symposium on Computational Geometry.

23. *Fast computation of generalized voronoi diagrams using graphics hardware.* **Hoff K.E., Culver T., Keyser J., Lin M., Manocha D.** 2000. Proceedings of 16th Annual Symposium on Computational Geometry.

24. *Crowd of virtual humans: a new approach for real time navigation in complex and structured environments.* **Lamarche, F. and Donikian, S.** 2004, Computer Graphics Forum, Vol. 10, pp. 509-518.

25. *Real-time navigation of independent agents using adaptive roadmaps.* **Sud, Avneesh and Gayle, Russell and Andersen,Erik and Guy, Stephen and Lin, Ming and Manocha, Dinesh.** 2007. VRST '07: Proceedings of the 2007 ACM symposium on Virtual reality software and technology.

26. *Real-time path planning in dynamic environments using mult-agent navigation graphs.* **Sud A., Anderson E., Curtis. S., Lin M. C., Manocha D.** 2008, IEEE Transactions on Visualization and Computer Graphics , pp. 526-538.

27. **Snook, Greg.** *Simplified 3D Movement and Pathfinding using Navigation Meshes.* s.l. : Game Programming Gems, 2000.

28. *Shortest paths with arbitrary clearance from navigation meshes.* **Kallmann, Marcelo.** 2010. Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation.

29. *Planning collision-free trajectories in time-varying environments: a two-level hierarchy.* **Kant, Kamal and Zucker, Steven W.** 1988, The Visual Computer, Vol. 3, pp. 304-313.

30. *Iterated forecast and planning algorithm to steer and drive a mobile robot in the presence of multiple moving objects.* **Tsubouchi, T. and Kuramochi, S. and Arimoto, S.** 1995. IROS '95: Proceedings of the International Conference on Intelligent Robots and Systems.

31. *Interactive motion correction and object manipulation.* **Shapiro, Ari and Kallmann, Marcelo and Faloutsos, Petros.** 2007. 3D '07: Proceedings of the 2007 symposium on Interactive 3D graphics and games.

32. *Space-time Planning with parameterized locomotion controllers.* **Levine S., Lee Y., Koltun V., Popovic Z.** 2011, ACM Transaction of Graphics.

33. **Lopez, Thomas and Lamarche, Fabrice and Li, Tsai-Yen.** *Space-time planning in changing environments : using dynamic objects for accessibility.* s.l. : Computer Animation and Virtual Worlds, Special Issue, 2012.

34. **Likhachev, Maxim and Gordon, Geoff and Thrun, Sebastian.** *ARA*: Anytime A* with Provable Bounds on Sub-Optimality.* s.l. : Advances in Neural Information Processing Systems, MIT Press, 2004.

35. **Koenig, Sven and Likhachev, Maxim.** *D* Lite.* s.l. : Eighteenth National Conference on Artificial Intelligence (AAAI), 2002.

36. *Anytime search in dynamic graphs.* **Likhachev, Maxim and Ferguson, Dave and Gordon, Geoff and Stentz, Anthony and Thrun, Sebastian.** 2008, Artificial Intelligence, Vol. 172, pp. 1613-1643.

37. **Milazzo, J. and Rouphail, N. and Hummer, J. and Allen, D.** The Effect of Pedestrians on the Capacity of Signalized Intersections. *Transportation Research Record.* 1998, pp. 37-46.

38. *Pedestrian Travel Behavior Modeling.* **Hoogendoorn, Serge P.** 2003. In 10th International Conference on Travel Behavior Research, Lucerne.

39. *Modeling and Simulation of Pedestrian Traffic Flow.* **Lovas, G.C.** 1994. Transportation Research Record.

40. *The statistics of crowd fluids.* **Henderson, L. F.** 1971, Nature, pp. 381-383.

41. **Narain, Rahul and Golas, Abhinav and Curtis, Sean and Lin, Ming.** *Aggregate Dynamics for Dense Crowd Simulation.* s.l. : ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia), 2009.

42. *Steering Behaviors for Autonomous Characters.* **Reynolds, Craig.** 1999. Game Developers Conference.

43. *Flocks, herds and schools: A distributed behavioral model.* **Reynolds, Craig W.** 1987, SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques, pp. 25-34.

44. *Self-Organized Pedestrian Crowd Dynamics: Experiments, Simulations, and Design Solutions.* **Helbing, Dirk and Buzna, Lubos and Johansson, Anders and Werner, Torsten.** 2005, Transportation Science}, Vol. 1, pp. 1-24.

45. *Modeling Individual Behaviors in Crowd Simulation.* **Braun, Adriana and Musse, Soraia R. and Oliveira, Luiz P. L. de and Bodmann, Bardo E. J.** 2003, CASA '03: Proceedings of the 16th International Conference on Computer Animation and Social Agents, p. 143.

46. *Group Behaviors for Systems with Significant Dynamics.* **Brogan, David C. and Hodgins, Jessica K.** 1997, Autonomous Robotics, Vol. 4, pp. 137-153.

47. *Specification of the Social Force Pedestrian Model by Evolutionary Adjustment to Video Tracking Data.* **Johansson, A and Helbing, D and Shukla, P.** 2007, Advances in Complex Systems, pp. 271-288.

48. *Centrifugal force model for pedestrian dynamics.* **Yu, W. J. and Chen, R. and Dong, L. Y. and Dai, S. Q.** 2005, Phys. Rev. E.

49. *Generalized centrifugal-force model for pedestrian dynamics.* **Chraibi, Mohcine and Seyfried, Armin and Schadschneider, Andreas.** 2010, Phys. Rev. E.

50. **Moussaid, Mehdi and Perozo, Niriaska and Garnier, Simon and Helbing, Dirk and Theraulaz, Guy.** *The Walking Behaviour of Pedestrian Social Groups and Its Impact on Crowd Dynamics.* s.l. : PLoS ONE, Public Library of Science, 2010.

51. *A Multi-Agent Cellular Automata System for Visualising Simulated Pedestrian Activity.* **Dijkstra, J. and Timmermans, H. J. P. and Jessurun, A. J.** 2000. Theoretical and Practical Issues on Cellular Automata - Proceedings on the 4th International Conference on Cellular Automata for research and Industry.

52. *Conflicts and Friction in Pedestrian Dynamics.* **Schadschneider, Andreas.** 2008. ACRI '08: Proceedings of the 8th international conference on Cellular Automata for Reseach and Industry.

53. *Agent Behaviour Simulator (ABS): A Platform for Urban Behaviour Development.* **Tecchia, Franco and Loscos, Céline and Conroy, Ruth and Chrysanthou, Yiorgos.** 2001. GTEC.

54. *Flow tiles.* **Chenney, Stephen.** 2004. SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation.

55. *Behavioral Intelligence for Geospatial Agents in Urban Environments.* **Torrens, Dr. Paul M.** 2007. IAT '07: Proceedings of the 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology.

56. *Intuitive Crowd Behaviour in Dense Urban Environments using Local Laws.* **Loscos, Celine and Marchal, David and Meyer, Alexandre.** 2003. Proceedings of the Theory and Practice of Computer Graphics 2003.

57. *Fragment shaders for agent animation using finite state machines.* **Rudomin, Isaac and Millan, Erik and Hernandez, Benjamin.** 2005, Simulation Modelling Practice and Theory, Vol. 13, pp. 741-751.

58. *Crowds by Example.* **Lerner, Alon and Chrysanthou, Yiorgos and Lischinski, Dani.** 2007, Computer Graphics Forum, Vol. 26, pp. 655-664.

59. *Group behavior from video: a data-driven approach to crowd simulation.* **Lee, Kang Hoon and Choi, Myung Geol and Hong, Qyoun and Lee, Jehee.** 2007. SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation.

60. *Building Agent-Based Walking Models by Machine-Learning on Diverse Databases of Space-Time Trajectory Samples.* **Torrens, Paul and Li, Xun and Griffin, William A.** 2011, Transactions in GIS, Vol. 15, pp. 67-94.

61. *Reactive Pedestrian Path Following from Examples.* **Metoyer, Ronald A. and Hodgins, Jessica K.** 2003, Proceedings of the 16th International Conference on Computer Animation and Social Agents (CASA 2003), pp. 149-156.

62. **Feurtey, F.** *Simulating the Collision Avoidance Behavior of Pedestrians.* s.l. : The University of Tokyo, School of Engineering, 2000.

63. *Pedestrian Reactive Navigation for Crowd Simulation: a Predictive Approach.* **Paris, Sebastien and Pettre, Julien and Donikian, Stephane.** 2007, EUROGRAPHICS, Vol. 26, pp. 665-674.

64. *Reciprocal Velocity Obstacles for Real-time Multi-Agent Navigation.* **van den Berg J., Manocha D.** 2008. IEEE International Conference on Robotics and Automation.

65. *Composite Agents.* **Yeh, Hengchin and Curtis, Sean and Patil, Sachin and van den Berg, Jur and Manocha, Dinesh and Lin, Ming.** s.l. : ACM SIGGRAPH Symposium of Computer Animation, 2008.

66. **Guy, Stephen J. and Chhugani, Jatin and Kim, Changkyu and Satish, Nadathur and Lin, Ming and Manocha, Dinesh and Dubey, Pradeep.** *ClearPath: Highly Parallel Collision Avoidance for Multi-Agent Simulation.* s.l. : Eurographics/ACM SIGGRAPH Symposium on Computer Animation, 2009.

67. *Parallelized Egocentric Fields for Autonomous Navigation.* **Kapadia, Mubbasir and Singh, Shawn and Hewlett, William and Reinman, Glenn and Faloutsos, Petros.** 2012, The Visual Computer, International Journal of Computer Graphics.

68. **Pettre, Julien and Ondrej, Jan and Olivier, Anne-Helene and Cretual, Armel and Donikian, Stephane.** *Experiment-based Modeling, Simulation and Validation of Interactions between Virtual Walkers.* s.l. : Eurographics/ACM SIGGRAPH Symposium on Computer Animation, 2009.

69. *A synthetic-vision based steering approach for crowd simulation.* **Ondrej, Jan and Pettre, Julien and Olivier, Anne-Helene and Donikian, Stephane.** 2010, ACM SIGGRAPH 2010 papers.

70. **Regelous, Stephen.** Massive. [Online] 2002. http://www.massivesoftware.com/.

71. **Mononen, Mikko.** Recast Navigation. [Online] 2009. http://code.google.com/p/recastnavigation/.

72. **Axel Buendia, Jerome Hoibian.** SpirOps: Scientific Research Labs in Artificial Intelligence. [Online] 2002. http://www.spirops.com/.

73. *An Open Framework for Developing, Evaluating, and Sharing Steering Algorithms.* **Singh, Shawn and Kapadia, Mubbasir and Faloutsos, Petros and Reinman, Glenn.** 2009. Proceedings of the 2nd International Workshop on Motion in Games.

74. *Autonomous pedestrians.* **Shao, Wei and Terzopoulos, Demetri.** 2005, ACM SIGGRAPH/Eurographics symposium on Computer animation, pp. 19-28.

75. *Footstep navigation for dynamic crowds.* **Singh, Shawn and Kapadia, Mubbasir and Reinman, Glenn and Faloutsos, Petros.** 2011, Computer Animation and Virtual Worlds, Vol. 22, pp. 151-158.

76. *Scenario Space: Characterizing Coverage, Quality, and Failure of Steering Algorithms.* **Kapadia, Mubbasir and Wang, Matthew and Singh, Shawn and Reinmann, Glenn and Faloutsos, Petros.** 2011. ACM SIGGRAPH Symposium on Computer Animation.

77. *SteerBench: A Benchmark Suite for Evaluating Steering Behaviors.* **Singh, Shawn and Kapadia, Mubbasir and Reinmann, Glenn and Faloutsos, Petros.** 2009, Computer Animation and Virtual Worlds.

78. *Improved Benchmarking for Steering Algorithms.* **Kapadia, Mubbasir and Wang, Matthew and Reinman, Glenn and Faloutsos, Petros.** 2011. International Conference on Motion in Games.

79. *Towards a quantitative approach for comparing crowds.* **Musse, Soraia R. and Cassol, Vinicius J. and Jung, Cláudio R.** 2012, Computer Animation and Virtual Worlds, Vol. 23, pp. 49-57.

| Article ID | Article title |
|---|---|
| COGSCI-019 | Cognitive Processing via the Interaction of Many Agents |
| COGSCI-034 | Natural Behaviors, Human/Primate Models |
| | |