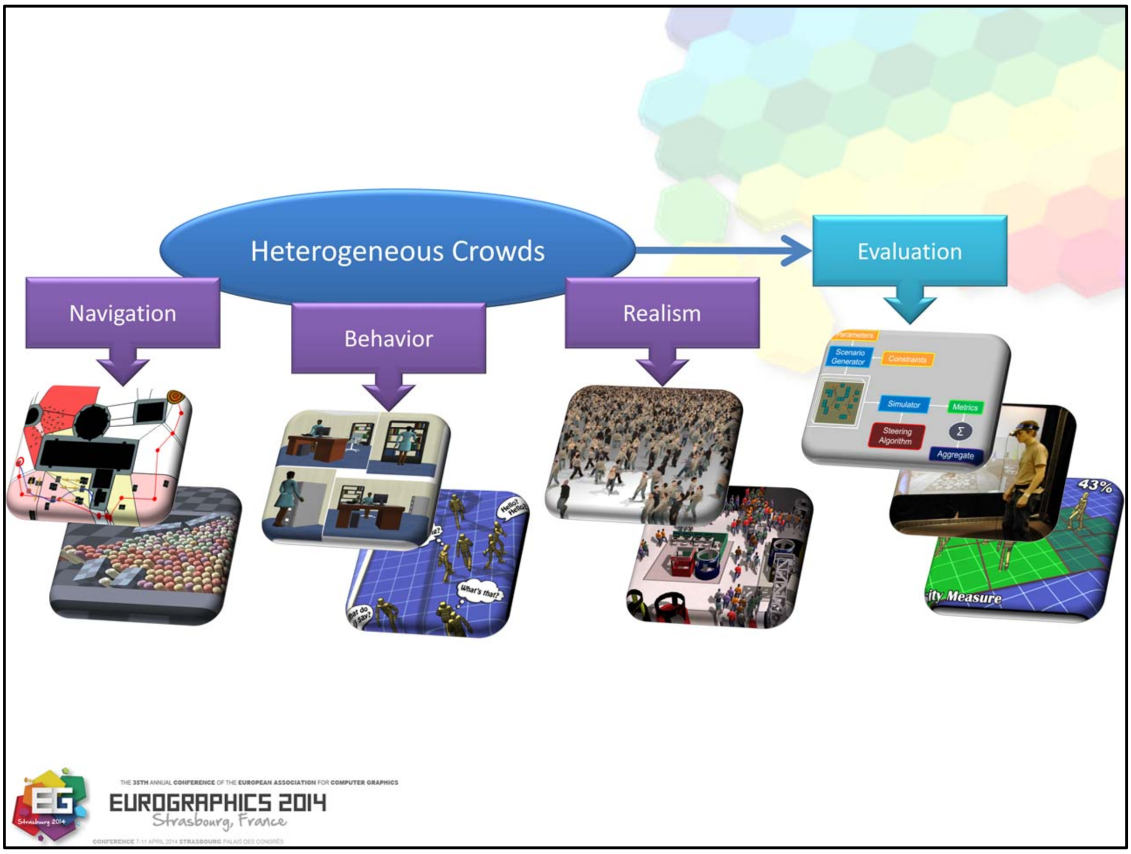
THE 35TH ANNUAL CONFERENCE OF THE EUROPEAN ASSOCIATION FOR COMPUTER GRAPHICS
EUROGRAPHICS 2014
Strasbourg, France
CONFERENCE 7-11 APRIL 2014 STRASBOURG PALAIS DES CONGRÈS

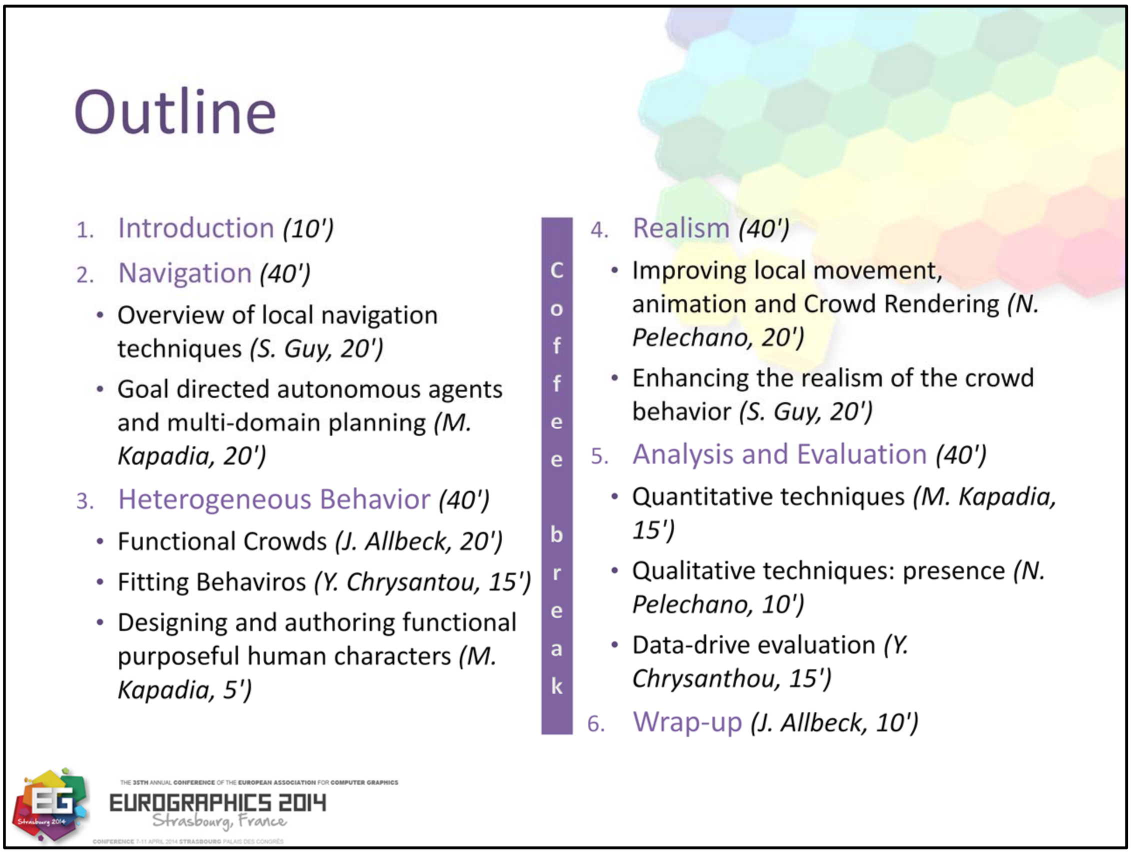
Simulating Heterogeneous Crowds

Tutorial

Mubbasir Kappadia — *Disney Research Zurich*
Nuria Pelechano — *Universitat Politècnica de Catalunya*
Stephen Guy — *University of Minnesota*
Jan Allbeck — *George Mason University*
Yiorgos Chrysanthou — *University of Cyprus*



Constructing frameworks for simulating heterogeneous crowds requires a number of components. Agents in a simulation need to be able to successfully navigate through their environment without collisions. Agent behaviors should appropriately represent the population they are depicting, including plausible interactions with objects and each other and individual behavioral differences from, for example, different personality types. Furthermore, there needs to be mechanisms for feasibly authoring these behaviors either directly from real world data or through computational frameworks. There must also be mechanisms in a crowd simulation system to enable realistic animation of the agents at real-time (or near real-time) rates. Finally, perhaps the most challenging aspect of creating frameworks to simulate crowds is determining how best to evaluate the results.



Over the last decade there has been a large amount of work towards trying to simulate crowds for different applications, such as movies, video games, training, and evacuations.

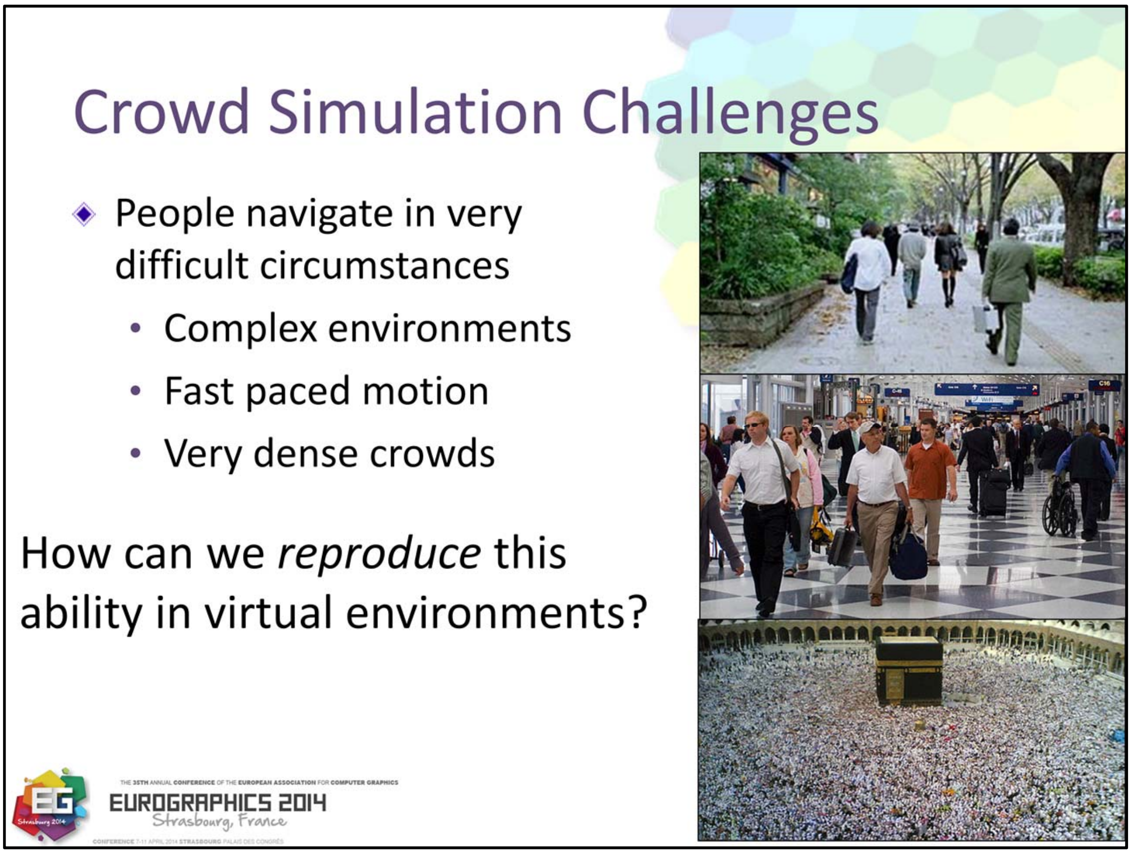
This course focuses on heterogeneous crowd simulation for interactive applications and will describe state of the art methods to simulate large groups of agents exhibiting a variety of behaviors, appearances and animations. We will present different techniques including psychological models and data-driven approaches that attempt to imitate real humans. We also present different systems to speed up both navigation, through multi-domain planners, and rendering, using per-joint impostors on fully animated 3D characters.

Finally we provide quantitative and qualitative techniques to evaluate the quality of the simulated crowds, and include an overview of future research directions in the field.

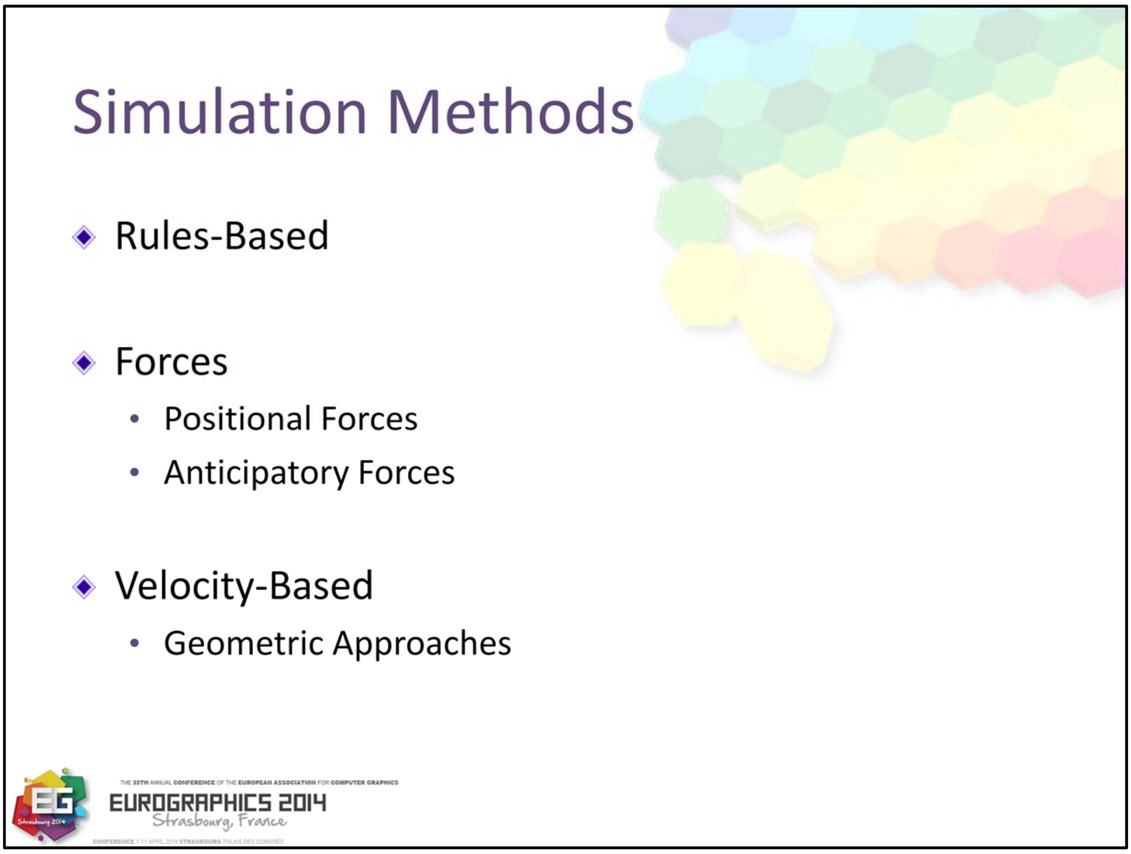
In crowd simulation, navigation refers to the process of moving simulated agents to their goal positions in a natural and collision free manner. This section of the tutorial covers both the basics of computing high-quality local motion (e.g., collision avoidance between agents) and an overview of the process of computing high-level paths and goals for each agent that account for the complex, dynamic nature of many virtual environments.

Speaker: Stephen J. Guy
Assistant Professor, University of Minnesota

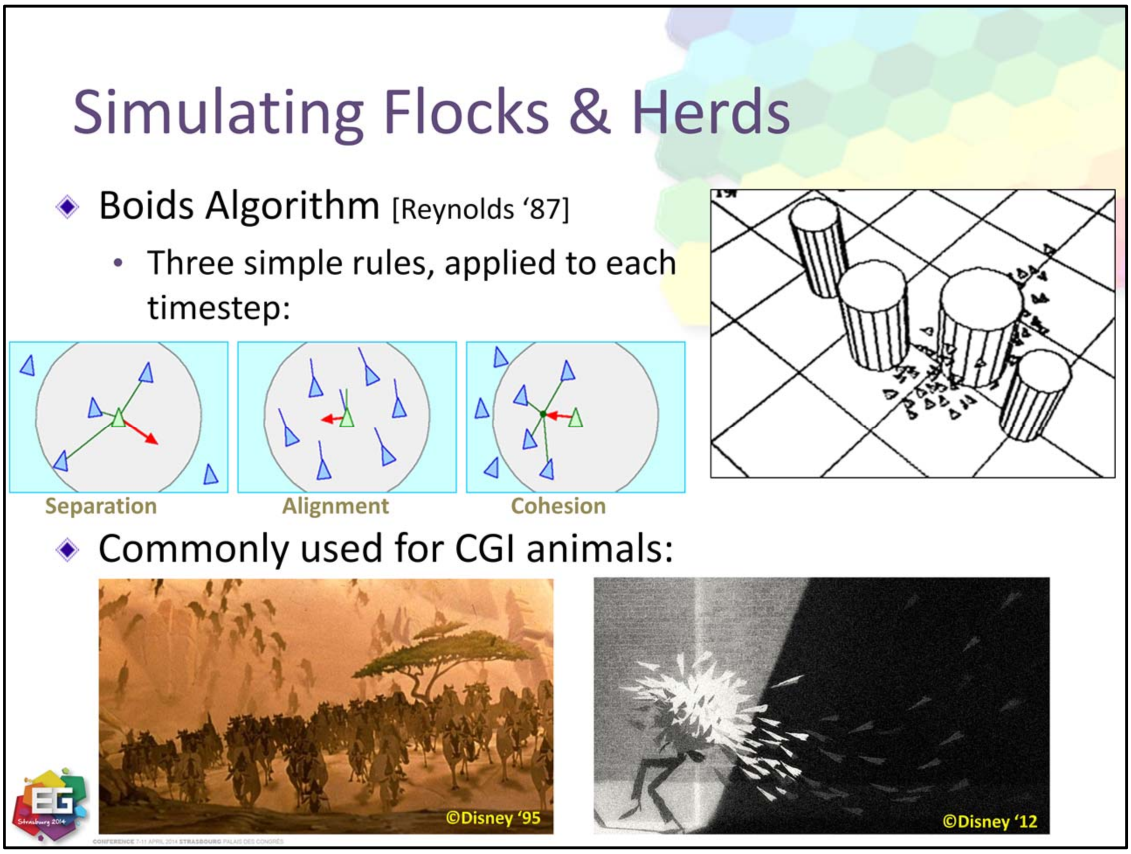
Bio: Stephen J. Guy is an assistant professor in the Department of Computer Science and Engineering at the University of Minnesota. His research focuses on the areas of interactive computer graphics (real-time crowd simulation, path planning, intelligent virtual characters) and multi-robot coordination (collision avoidance, sensor fusion, path planning under uncertainty). Stephen's work on motion planning has been licensed for use in games and virtual environments by Relic Entertainment, EA, and other companies; his work in crowd simulation has been recognized by best paper awards at international conferences.



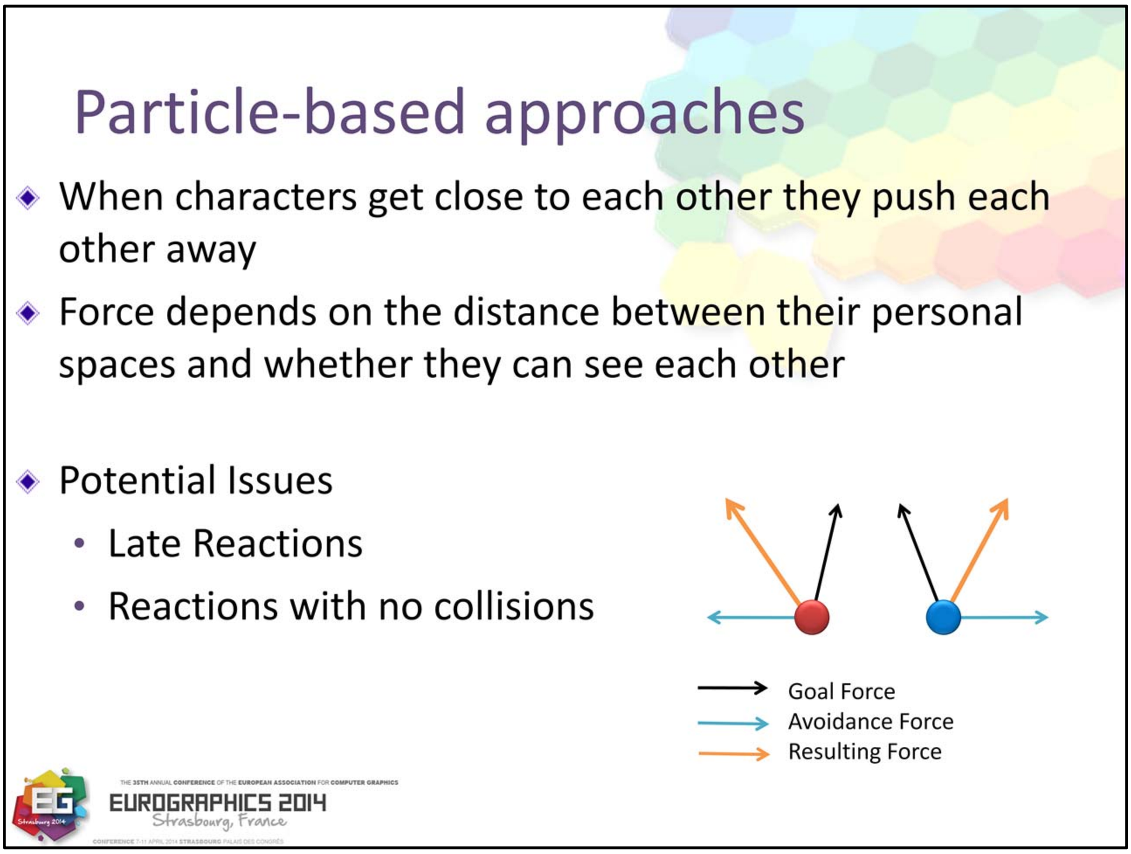
Local path planning, often referred to as collision-avoidance, is the process of computing paths for agents which progress towards their goals while avoiding potential collisions with nearby neighbors. The process is generally accomplished effortlessly by humans in a variety of environments. Even simple tasks such as walking between people on a crowded sidewalk can be difficult to reliably reproduce via automated methods as it involves planning collision free paths between the high-velocity, dynamically moving obstacles that are other people. Even in fast moving settings (such as airports) or very dense environments (religious or sporting gatherings) humans can naturally find clear collision free motion. Our goal is to develop algorithms that reproduce this natural, efficient motion for virtual characters.



A wide variety of crowd simulation methods have been proposed as approaches to solving the local collision avoidance problem. We will discuss three high-level types of simulation techniques: rules-based approaches, force-based approaches, and velocity-based approaches. While examining these different methods, an important aspect to analyze the methods is the role that anticipation plays in computing paths for the agents. As we'll see, both force-based and velocity-based approaches can be used to provide anticipatory motion planning where agents intelligently and efficiently avoid upcoming collisions as they move to their goals.



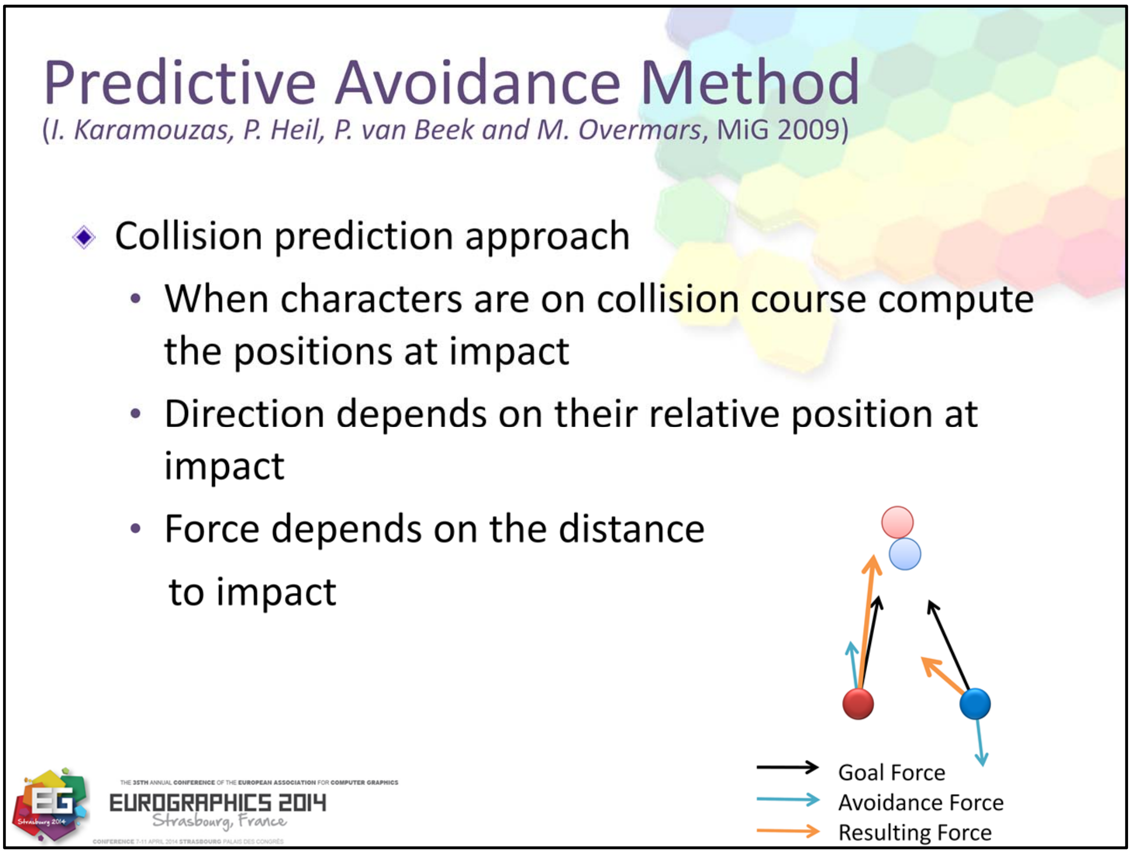
One of the earliest methods proposed for local collision avoidance is the Boids algorithm from Craig Reynolds (1987). Boids are simulated bird-like agents that move according to three simple rules. First: each boid feels a force pushing it away from the location of its closest neighbors; Second: each boid feels a force aligning it's velocity to that of it's neighbors; and Third: each boid feels a force drawing it towards the average position of its local neighbors. The movement of each boid is computed on a per-timestep basis, with the sum of the forces from all three rules determining the next velocity for each agent for each timestep. The emergent motion that results from the collective application of the boids rules to many agents leads to compelling herding and flocking behaviors.



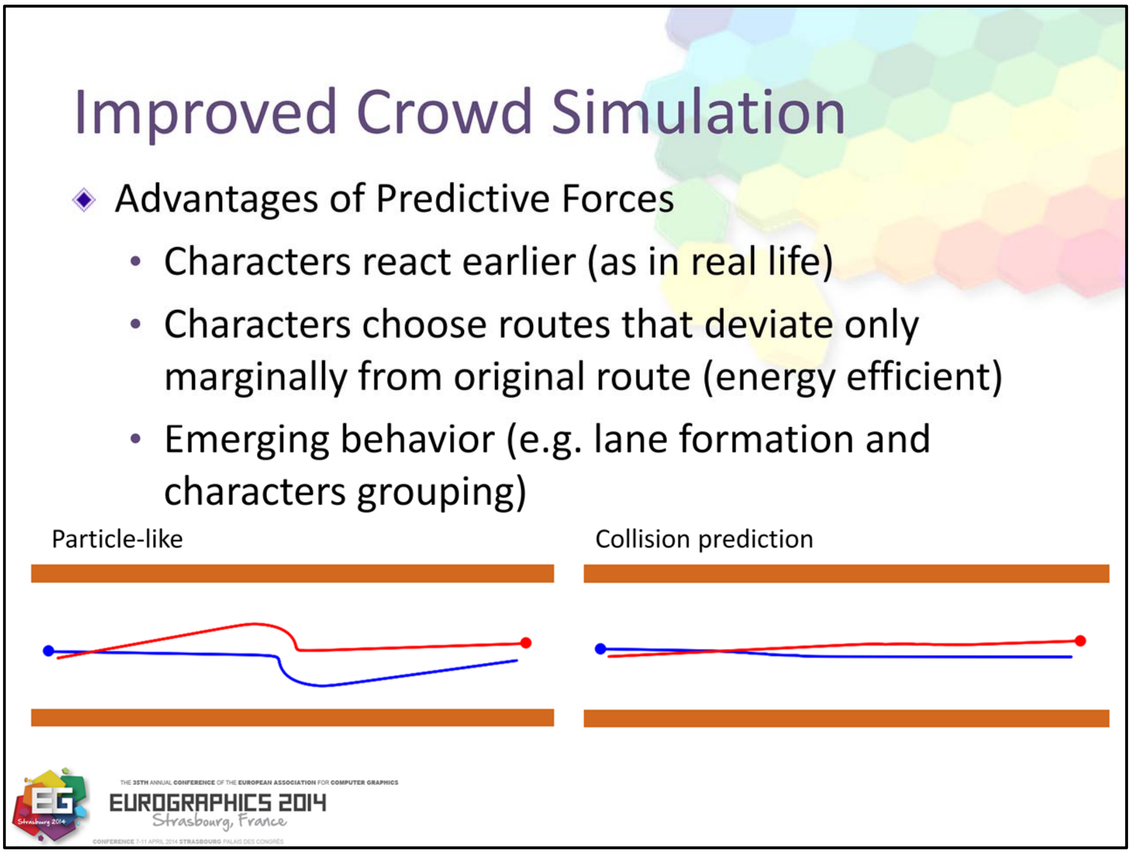
Rather than using boid-like rules, human motion in games and virtual environments is generally simulated using pair-wise interaction forces. In this paradigm, each agent feels a force based on the relative position and velocity of nearby agents. The resulting motion of an agent is computed from the combined forces on an agent from it's nearby neighbors.

In it's simplest implementation, the force on each agent is computed only a function of the current distance to the neighbor – when two agents are close they feel a strong mutually repulsive force pushing them away from each other, when agents are further away they feel little or no interaction forces. The final computed motion of an agent also accounts for a driving force which pulls the agent towards it's goal. When properly balanced, the combination of the driving force and collision avoidance forces can lead to collision-free, goal-oriented motion.

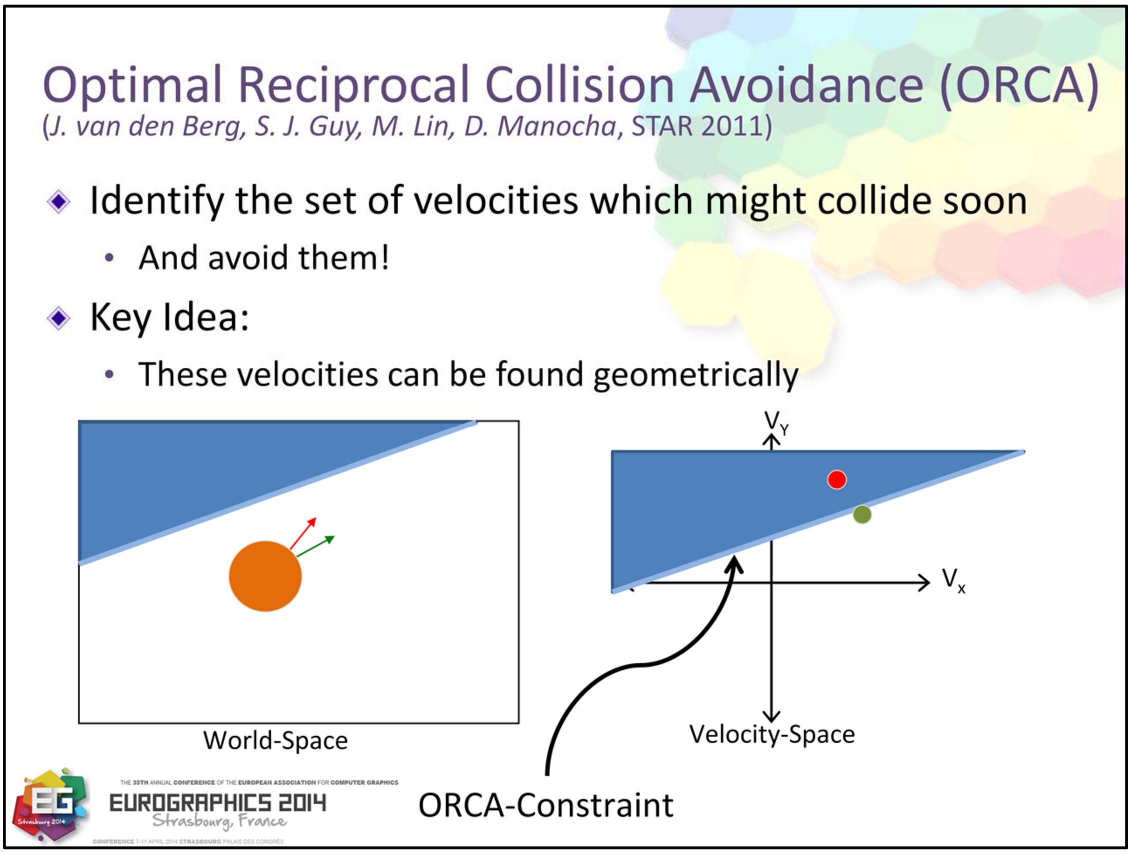
Despite their simplicity, purely positional approaches have important drawbacks which can to lead to unnatural motion. Most importantly, by not accounting for the relative velocity between agents, the resulting behavior is not "anticipatory" with agents reacting very late to upcoming collisions or moving to avoid a nearby agent which is not actually on a collision coarse.



The predictive avoidance method of Karamouzas et al. (2009) provides an example of how to account for the relative velocities of neighboring agents when computing avoidance forces. The proposed method accounts for velocities through two major modifications. Firstly, the direction of the avoidance force between agents is based not on the agent's current position put rather the expected relative positions at the point where the agents would collide. Secondly, the magnitude of the avoiding force is based on the distance to the upcoming collision, with more imminent collisions resulting in larger avoidance forces. The resulting forces result in motion in which agents anticipate likely future states generating smoother motion.

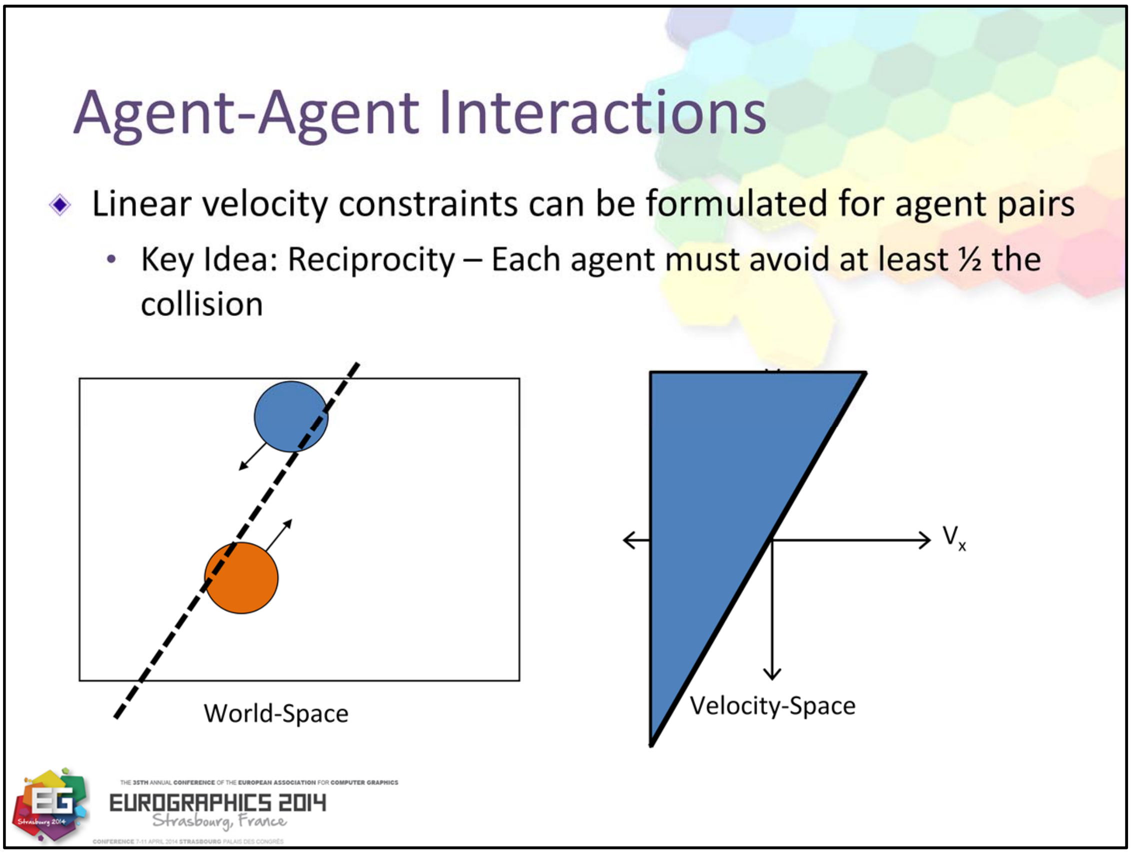


The above scenario clearly shows some of the advantages of predictive avoidance forces as compared to purely position based techniques (labeled "particle-like"). When computing responses with only positions agents react to each other quite late, and are forced to take drastic avoidance maneuvers to avoid collisions. In contrast, when using predictive forces, the agents react much earlier to each other and can therefore take much smaller avoidance actions – the result is significantly more efficient motion for both agents that more closely resembles human behavior.



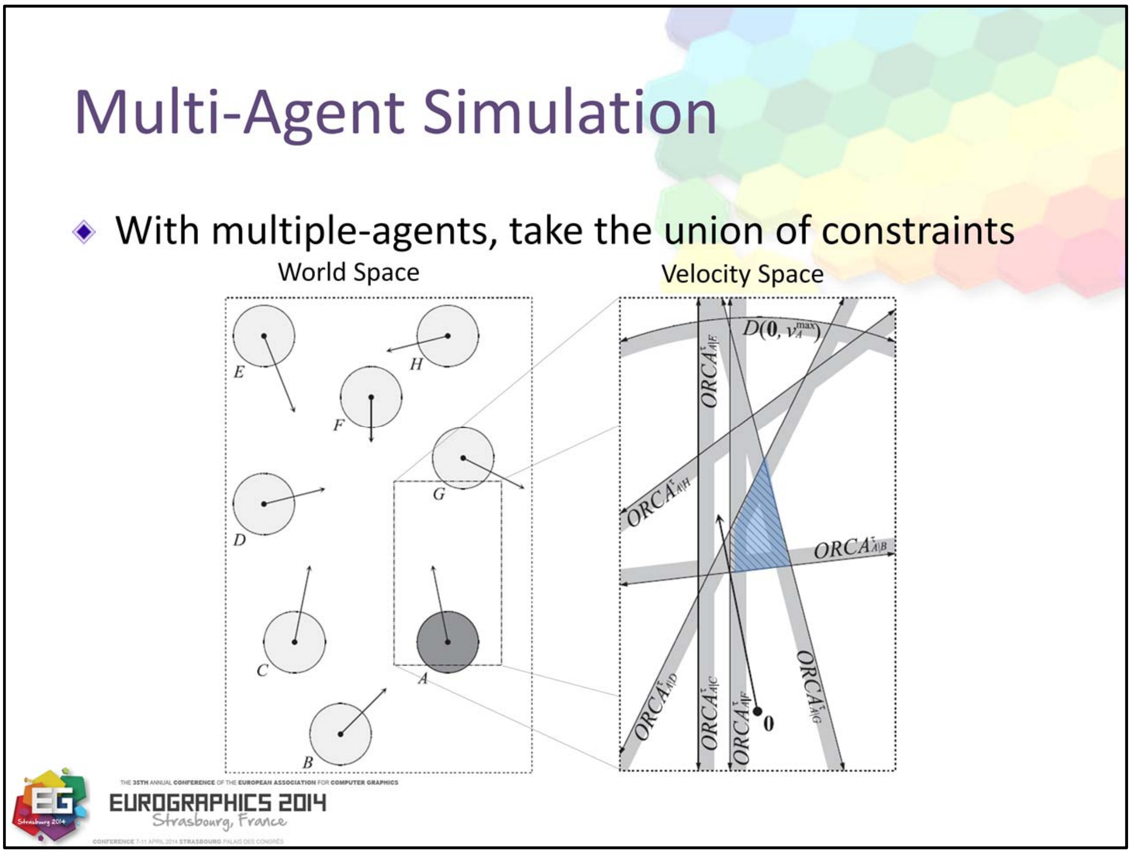
The ORCA collision avoidance technique provides an alternative paradigm which allows agents to compute new velocities without the use of forces. At a high level, ORCA proceeds similarly to the force based method in that each agent's motion will be computed as a function of the position and velocity of neighboring agents. However, agent's choose their velocity by utilizing a concept known as "velocity space". This is a 2D vector space of all possible velocities an agent might take at each timestep. Each nearby agent or obstacle provides a constraint on the velocities in agent may take, in the above example forbid velocities are shown in blue. Importantly, the velocity constraints can be computed geometrically as a function of the relative position and velocity of the neighboring agent or obstacle.

Given these ORCA constraints, we can now compute the motion for the agents. Each timestep, each agent will choose a new velocity (green dot) that is closest to it's goal velocity (red dot) while not being forbidden by any ORCA constraints.

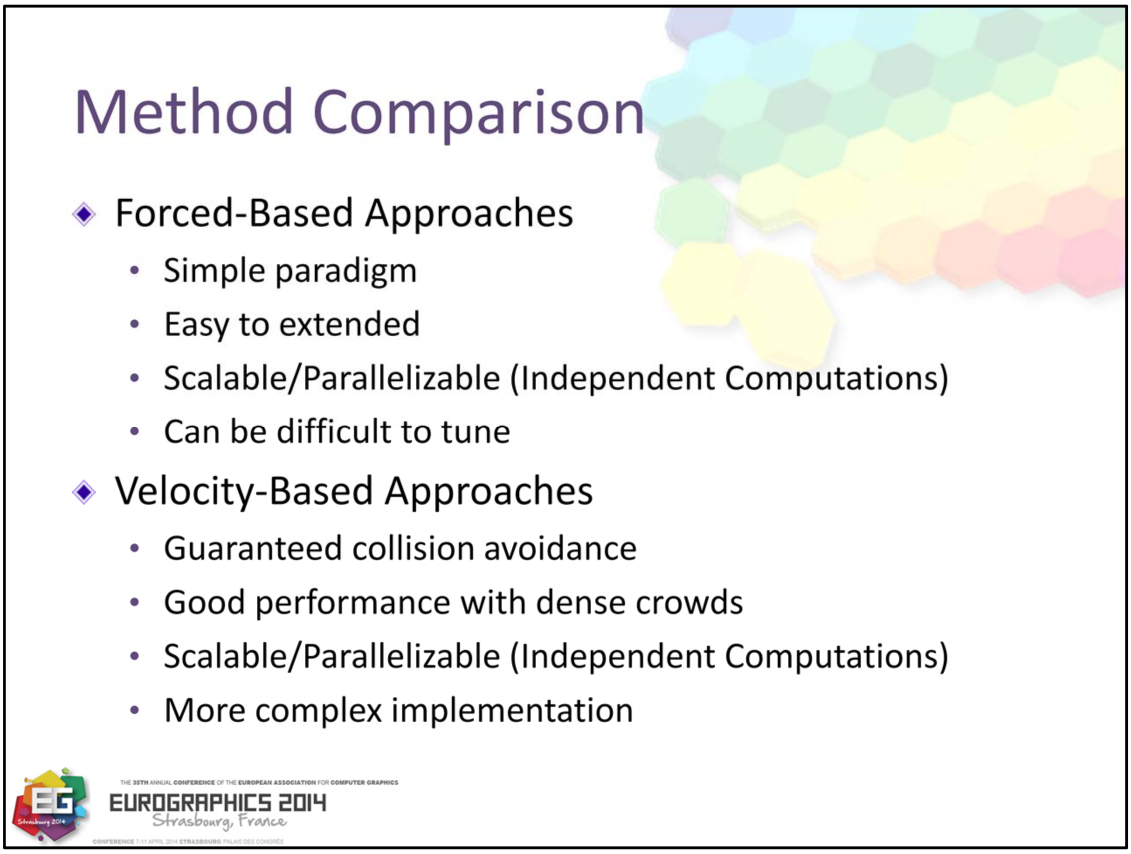


In general, computing an ORCA constraint between two agents is similar to computing the ORCA constraint caused by an obstacle. The one change is that rather than each agent avoiding the entire collision (which would cause redundant effort) each agent avoids only half of the collision. In this way the agents *reciprocally* share the collision avoidance efforts, and between the two avoid the total collision.

Because ORCA reasons directly over velocities, the resulting motion provides anticipatory avoidance behavior. Agents will see any velocities which lead to upcoming collisions as forbidden by an ORCA constraint (by definition) and these velocities well not be chosen. Like the anticipatory force models, these velocity-based reasoning leads to efficient, collision-free motion.



When an agent has multiple neighbors, the ORCA process naturally extends to multiple constraints in an agent's velocity space. As long as an agent chooses a new velocity which satisfies the union of all these constraints (highlighted in blue above) the resulting motion is *guaranteed* to be collision free. The full ORCA procedure then is as follows: each timestep, each agent computes the ORCA constraints on it's new velocities, the union of these constraints provide a region of velocities which is know to be collision free for the near future, the agent then chooses the velocity in this allowed region which is closest to it's preferred, goal velocity.

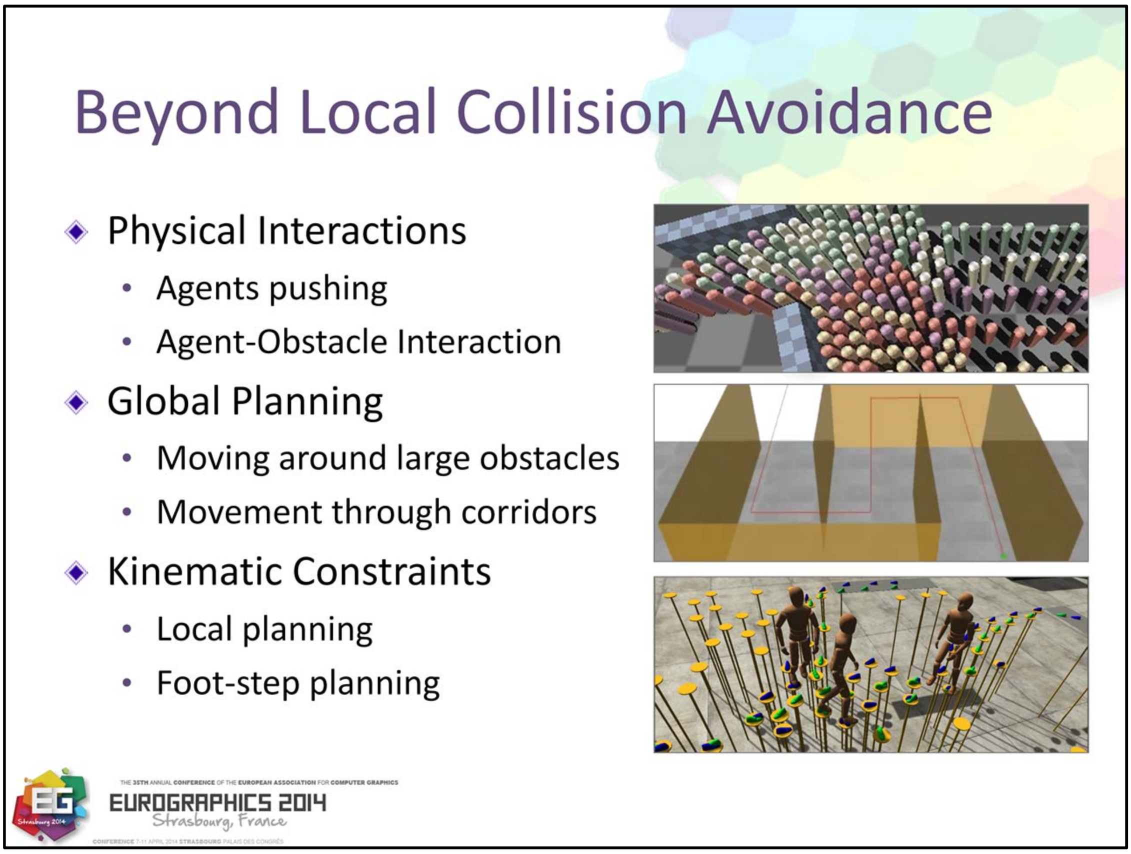


While force-based methods and velocity-based methods both provide means to simulate anticipatory collision avoidance, there are some important differences between the two approaches. Force-based simulation offers a simple, well parallelizable paradigm that is easy to implement and easy to extend by simply adding new forces to an agent. In contrast, because they require correctly computing the geometry of relative collisions (along with union of 2D sets and other computational geometry techniques), velocity based methods can be more difficult to code correctly and generally lead to a more complex code. However, this complexity has a tradeoff as these methods can provide guaranteed collision avoidance between agents. This guaranteed behavior can be important in very dense scenarios where other methods can lead easily to collisions between agents.

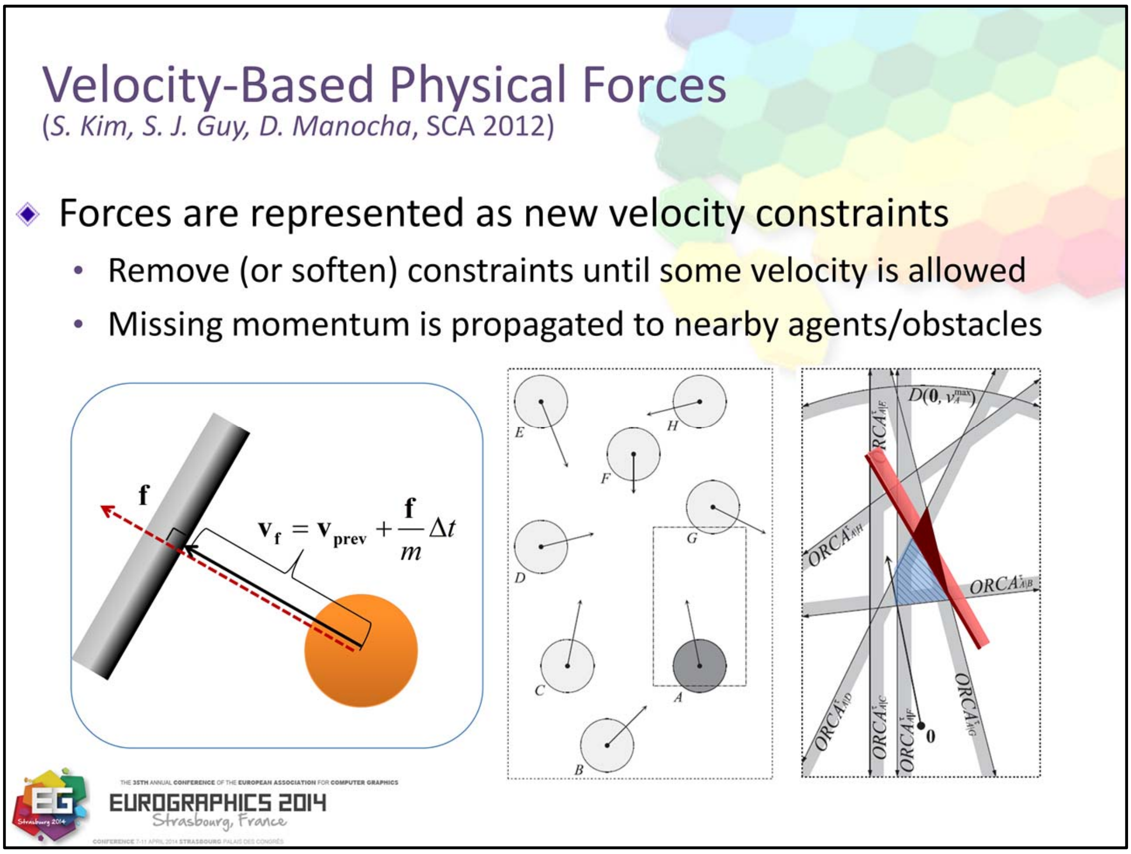
Saleable Dense Simulations (realtime)

- Simulated via ORCA (http://gamma.cs.unc.edu/RVO2)

(Videos)
Here are some example simulations computing using ORCA. The Hajj simulation contains 25,000 agents, but can still run in realtime (~10FPS).



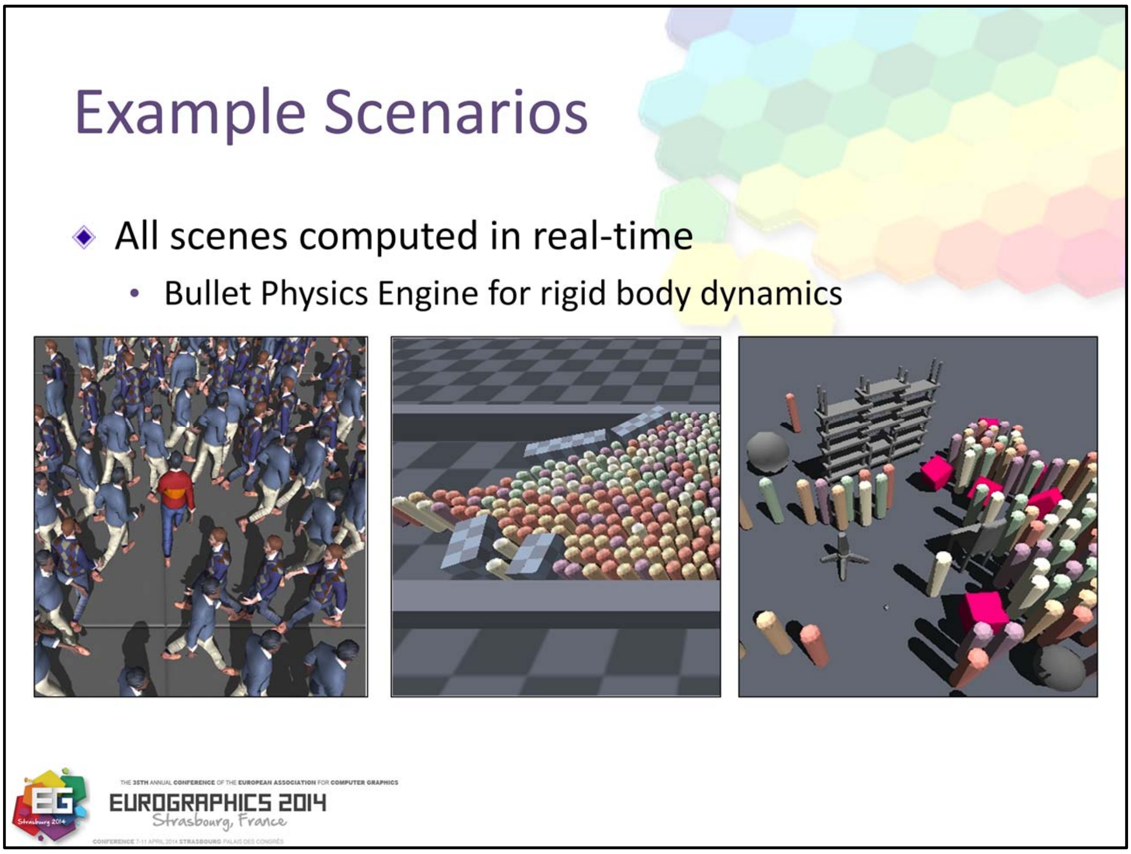
While avoiding collisions with one's neighbors is an important first step, there are many further important aspects to creating natural crowd simulations. For example, we may wish to allow agents to physically interact with each other through pushes and pulls (especially in dense environments). Likewise, when confronted with large obstacle or complex environments it's important to compute some form of global path to keep agents from getting stuck in local minima. We can also look at more detailed representation of humans than as simple disks with constant velocities. Such considerations can help us understand how the compute exact footstep placements to create more natural motion. We will discuss each of these aspects in turn.



Physical interaction between agents can be most naturally accounted for by representing contact between agents with the resulting force. If the agents are being simulated with force-based collision avoidance, this force can be simply added as an additional force affecting the agent's motion (potentially with some appropriate scaling). However, applying forces with a velocity-based simulation is less straight forward.

One approach, suggested by Kim et al. (2012), is to compute the effect the physical forces will have on an agent's velocity by integrating it over the duration of one timestep. This will result in an expected change in velocity as a result of the force. This change in velocity can be directly represented as a new velocity constraint by taking the half-plane of velocities in the direction of the force vector at least f/m*dt m/s away form the current velocity. This new velocity constraint is simply combined with all the previous ORCA constraints when computing a new velocity. The result is a new velocity which both avoids collisions and responds to physical forces.

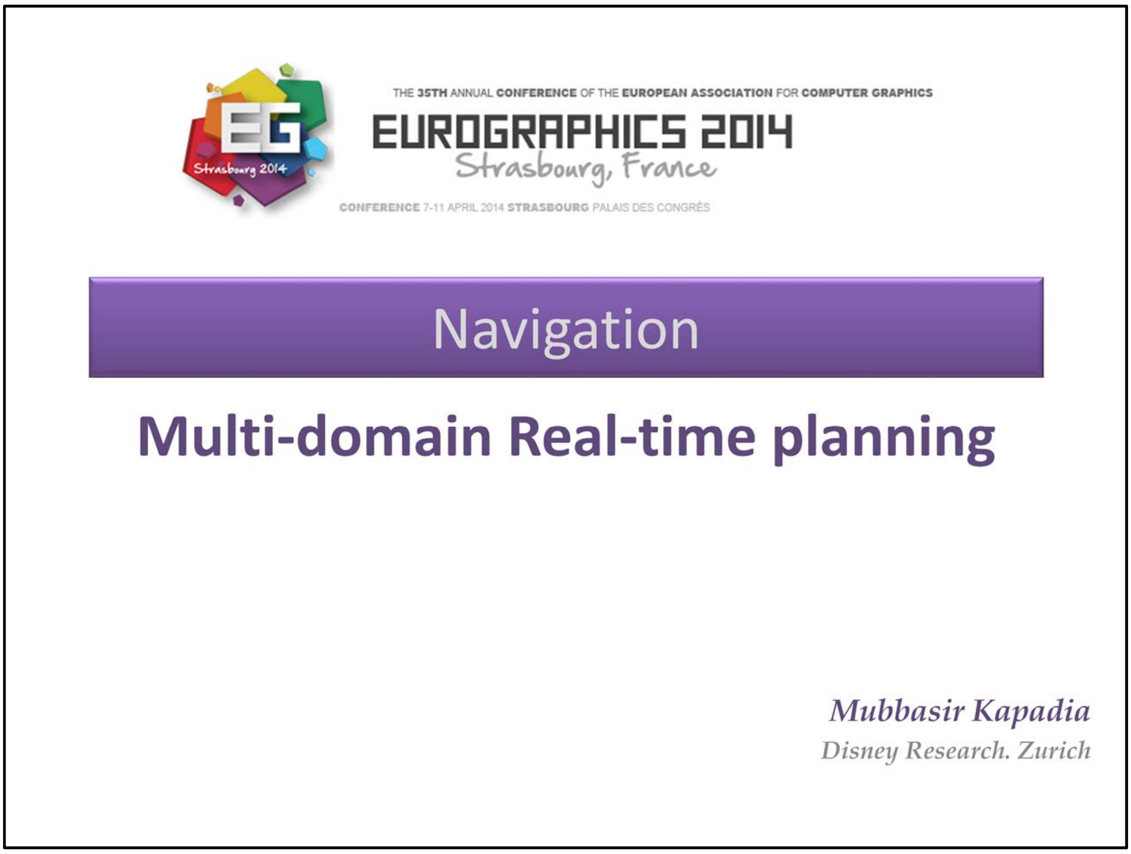
In the case that two many constrains are enacted on an agent (creating a null union of velocities), constrains from far away agents are relaxed or removed until some valid velocity is found. This process eliminates the guarantee of collision free motion and may introduce new collisions into the system. These collisions will then generate new physical forces from the contact between agents, which create new force-based velocity constraints for the next timestep.



Here are three sample scenarios simulated using the above described technique. The Bullet Physics Engine was used in order to compute the motion of the rigid bodies. All three simulations run in real-time.

Acknowledgements – S. J. Guy

- Dinesh Manocha, Ming Lin, Sujeong Kim, Jur van den Berg, Sean Curtis, Sachin Patil, Wenxi Liu, Ioannis Karamouzas
- This work presented here has been supported in part by:
  - ARO Contracts W911NF-04-1-0088,
  - NSF award 0636208, 0904990, 0917040, 1117127, 1000579
  - DARPA/RDECOM Contracts N61339-04-C-0043, WR91CRB-08-C-0137, W911NF-10-1-0506
  - Intel Cooperation, AMD, Minnesota Supercomputing Institute

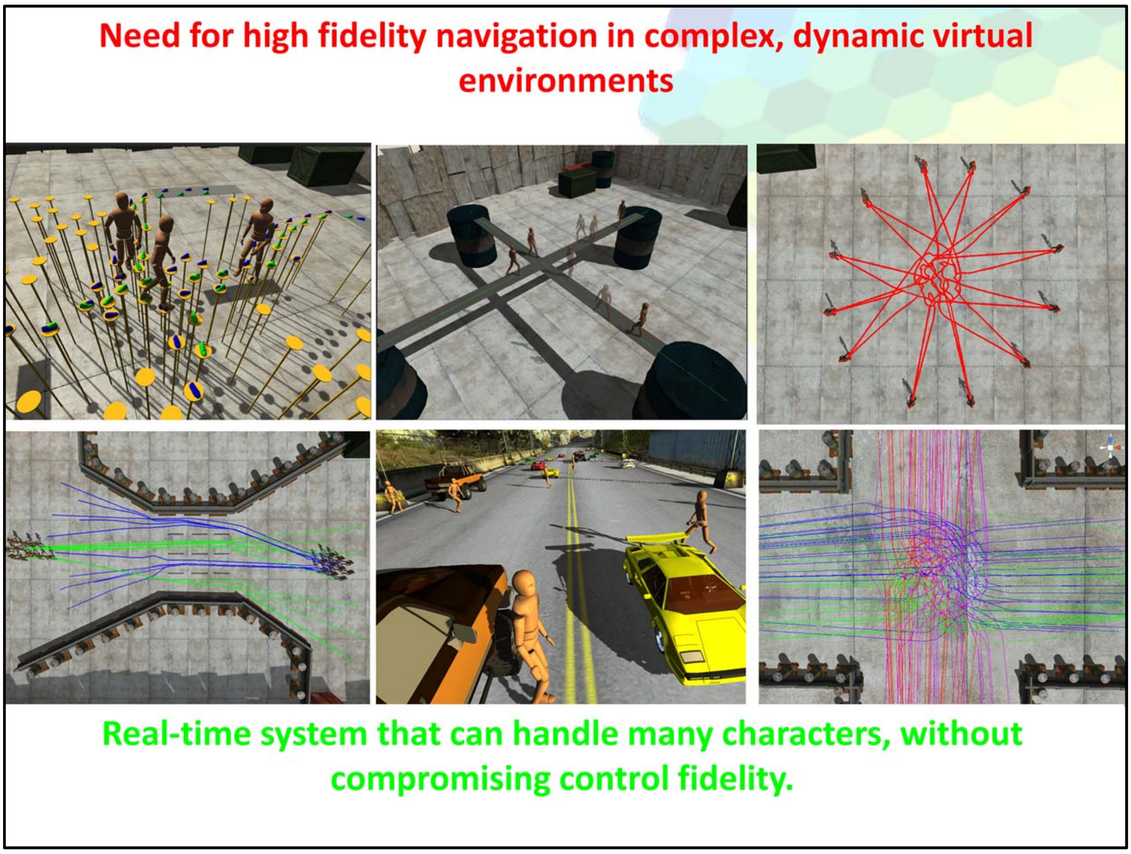


Speaker: Mubbasir Kapadia, Associate Research Scientist, Disney Research Zurich

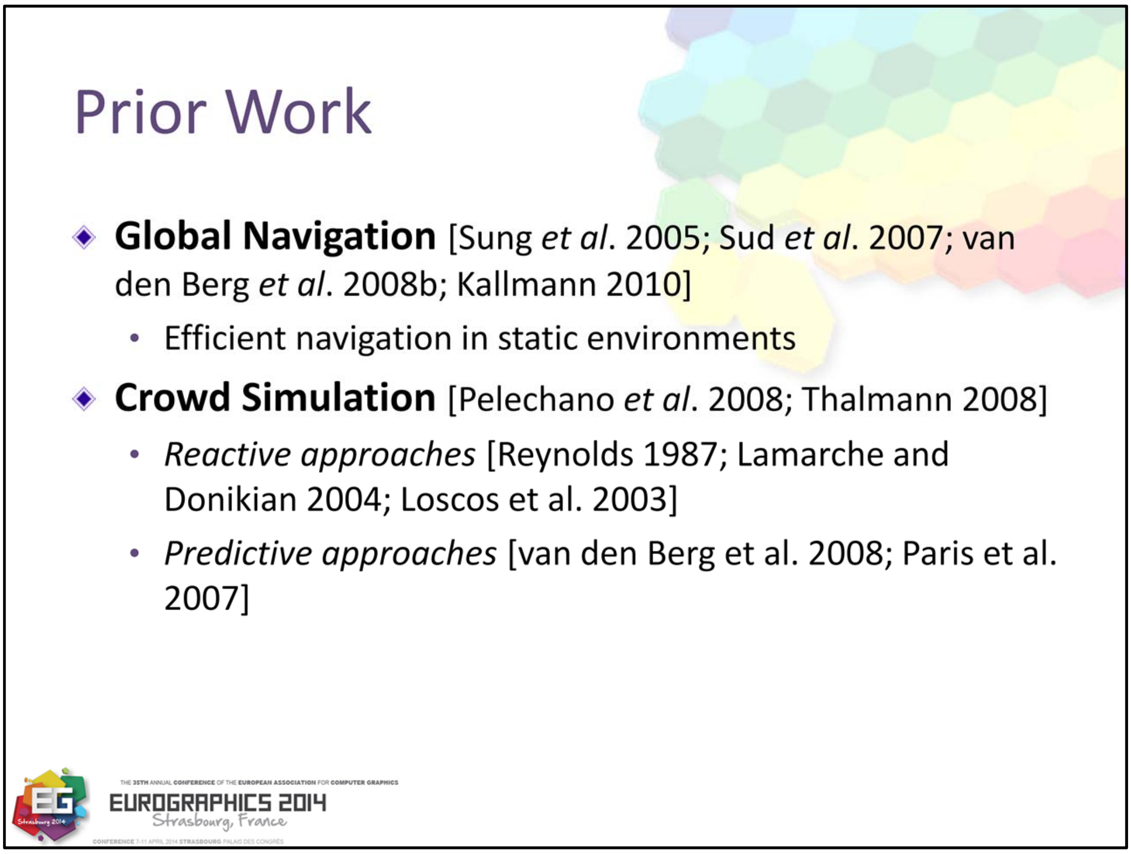
Bio. Mubbasir Kapadia is an Associate Research Scientist at Disney Research Zurich. Previously, he was a postdoctoral researcher and Assistant Director at the Center for Human Modeling and Simulation at University of Pennsylvania. He was the project lead on the United States Army Research Laboratory (ARL) funded project Robotics Collaborative Technology Alliance (RCTA). He received his PhD in Computer Science at University of California, Los Angeles.Kapadia's research aims to develop integrated solutions for full-body character animation, planning based control, behavior authoring, and statistical analysis of autonomous virtual human simulations. The far-reaching goal is to provide functional, purposeful embodied virtual humans, that act and interact in meaningful ways to simulate complex, dynamic, narrative-driven, interactive virtual worlds.

Related Publication(s):
Multi-Domain Planning for Real-Time Applications
Mubbasir Kapadia, Alejandro Beacco, Francisco Garcia, Vivek Reddy, Nuria Pelechano, and Norman I. Badler
ACM SIGGRAPH/EUROGRAPHICS Symposium on Computer Animation, July 2013

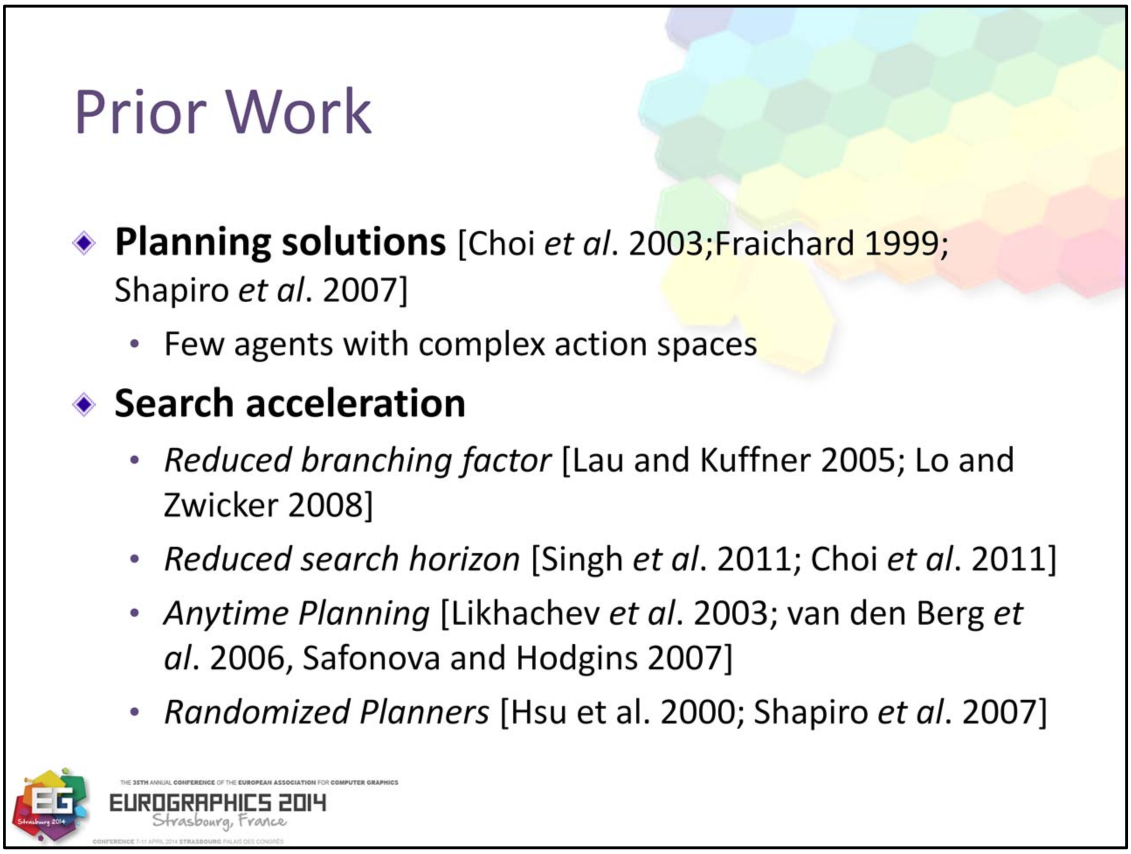
Project webpage: http://people.inf.ethz.ch/kapadiam/projects-mdp.html



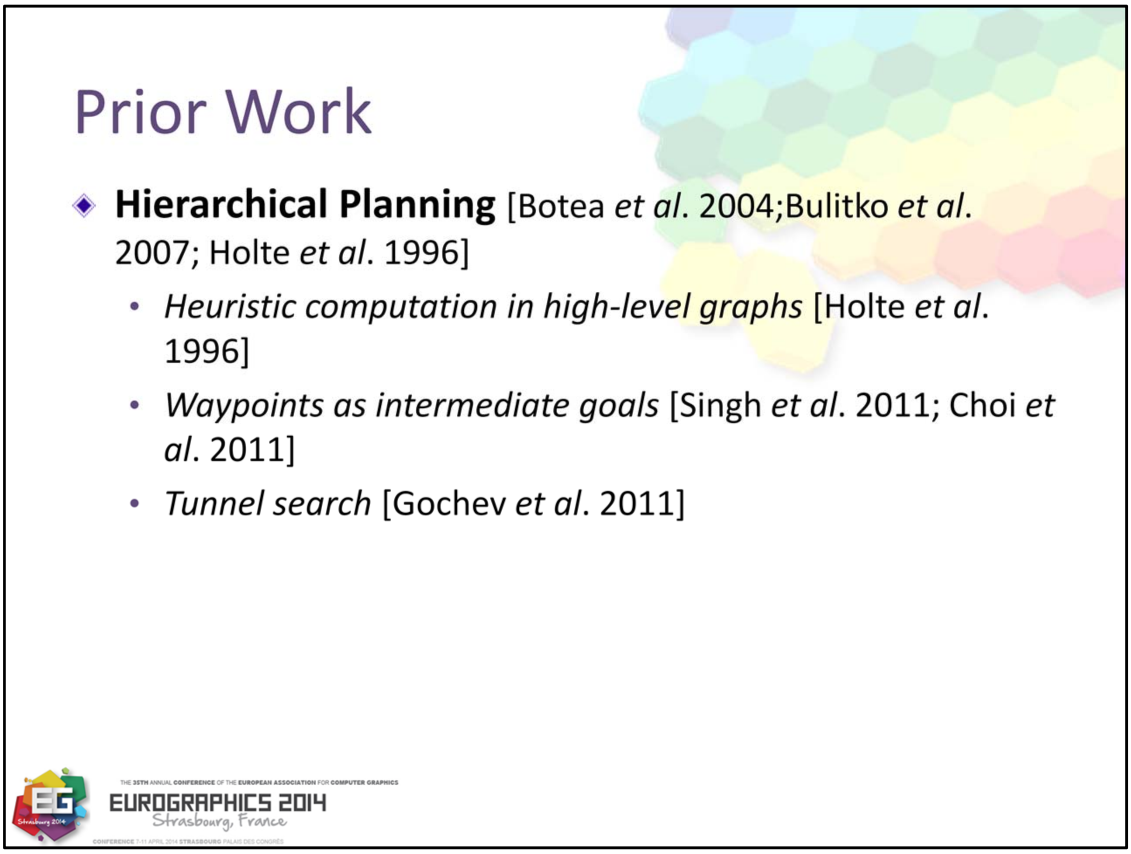
The next generation of interactive applications requires high fidelity navigation of interacting autonomous agents in non-deterministic, dynamic virtual worlds. The environment and agents are constantly affected by unpredictable forces (e.g., human input), making it impossible to accurately extrapolate the future world state to make optimal decisions. These complex domains require robust navigation algorithms that can handle partial and imperfect knowledge,
while still making decisions which satisfy space-time constraints. The far-reaching goal is to develop a real-time navigation system for autonomous agents that can handle many characters at interactive rates without compromising on control fidelity.



Global navigation approaches precompute a roadmap of the global environment which is used for making efficient navigation queries, but generally regard the environment to be static. Crowd approaches compromise on control fidelity in an effort to efficiently simulate a large number of agents in real-time. Reactive approaches avoid collisions with most imminent threats while predictive approaches approximate the trajectories of neighboring agents in choosing collision-free velocities.

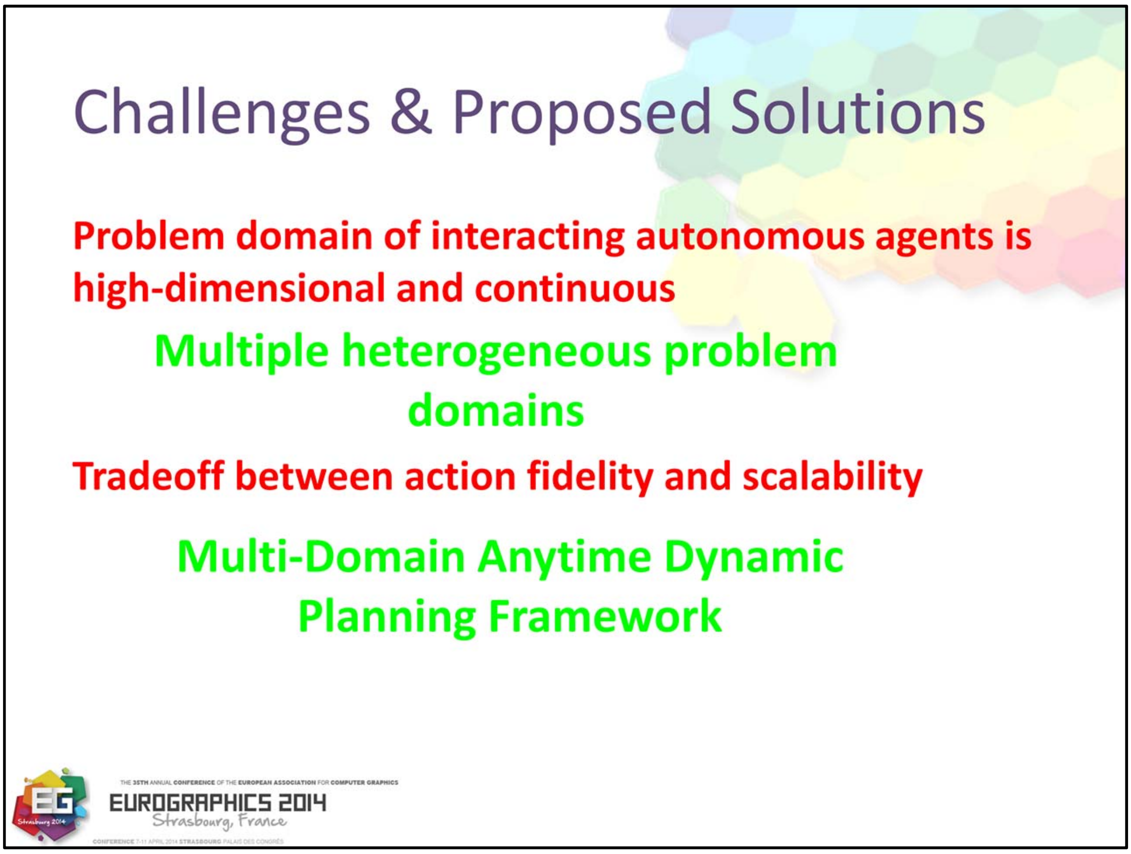


Planning based control of autonomous agents has demonstrated control of single agents with large action spaces. In an effort to scale to a large number of agents, meet real-time constraints, and handle dynamic environments a variety of methods have been proposed. These include reducing domain complexity which effectively reduces the branching factor of the search, or the horizon of the search is limited to a fixed depth. Different planning algorithms sacrifice optimality to satisfy strict time constraints. This can be done by inflating the value of the heuristic to drive the search towards the target, Or use sampling based methods that can efficiently explore very high dimensional continuous state spaces.



Hierarchical planners precompute abstractions in the state space, which can be used to speed up plan efforts. Given a discrete environment representation, neighboring states are first clustered together to precompute abstractions for high-level graphs. Different algorithms are proposed [Kring et al. 2010] which plan paths hierarchically by planning at the top level first, then recursively planning more detailed paths in the lower levels, using different methods [Lacaze 2002; Sturtevant and Geisberger 2010] to communicate information across hierarchies.

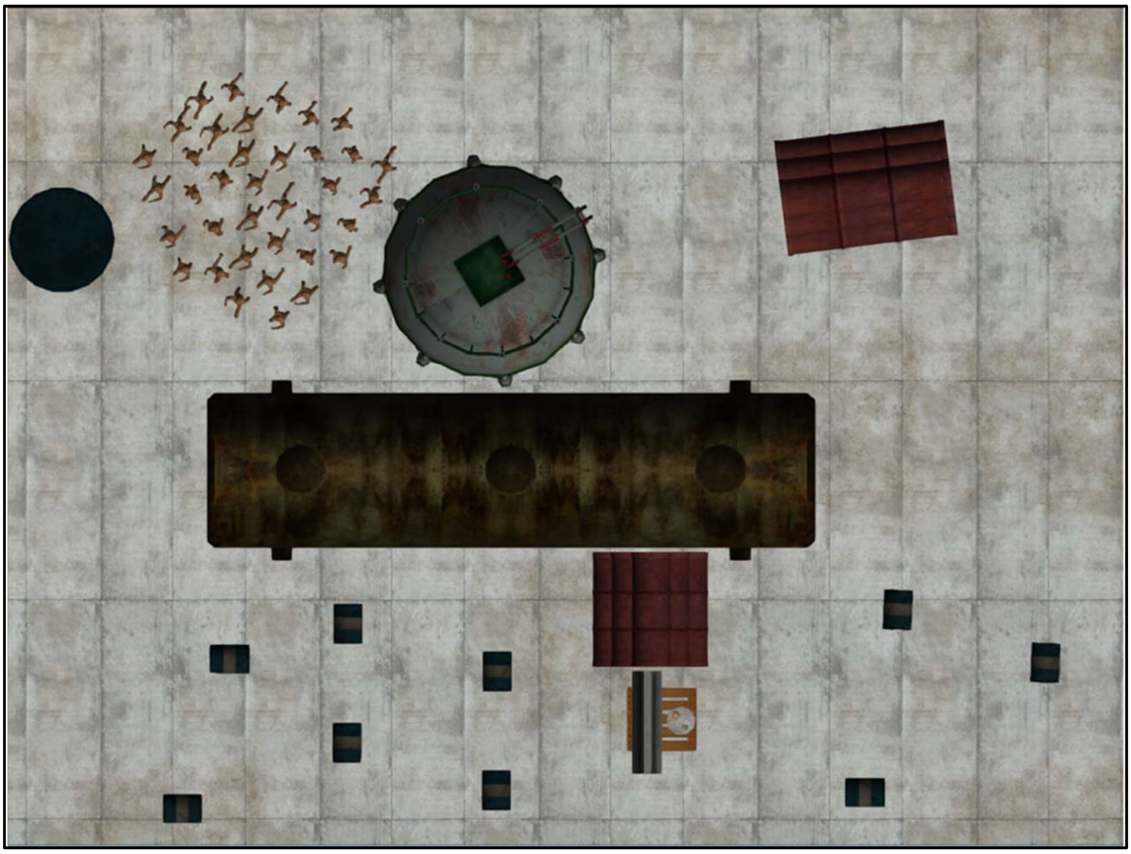
These include using the plans in high-level graphs to compute heuristics for accelerating searches in low-level graphs [Holte et al. 1996], using the waypoints as intermediate goals, or using the high-level path to define a tunnel [Gochev et al. 2011] to focus the search in the low-level graph. The work in [Arikan and Forsyth 2002] demonstrates the use of randomized search in a hierarchy of motion graphs for interactive motion synthesis.



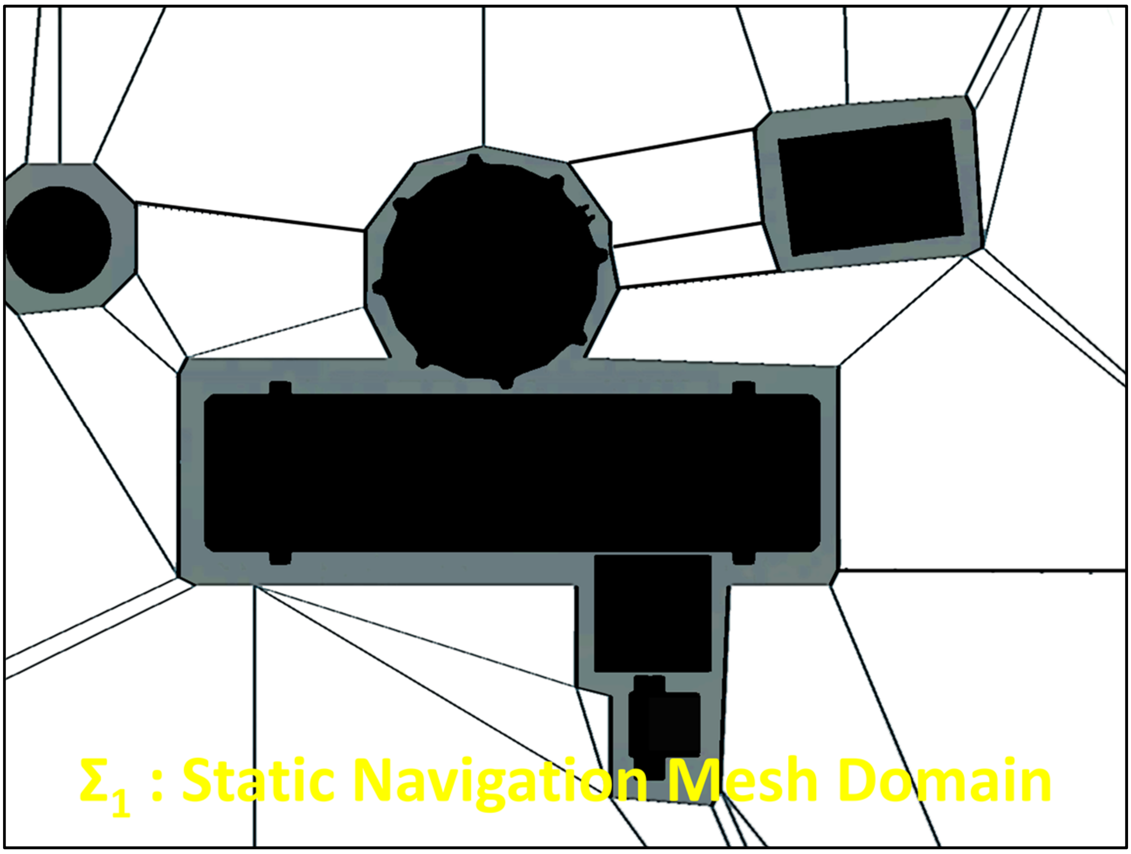
There are two main challenges that need to be addressed: (1) The problem domain of interacting autonomous agents in dynamic environments is extremely high-dimensional and continuous, with infinite ways to interact with objects and other agents. (2) Having a rich action set, and a system that makes intelligent action choices, facilitates robust, intelligent virtual characters, at the expense of interactivity and scalability. Greatly simplifying the problem domain yields interactive virtual worlds with hundreds and thousands of agents that exhibit simple behavior.

We propose a real-time planning framework for multicharacter navigation that uses multiple heterogeneous problem domains of differing complexities for navigation in large, complex, dynamic virtual environments. We define a set of problem domains (spaces of decision-making) which differ in the complexity of their state representations and the fidelity of agent control. These range from a static navigation mesh domain which only accounts for static objects in the environment, to a space-time domain that factors in dynamic obstacles and other agents at much finer resolution.

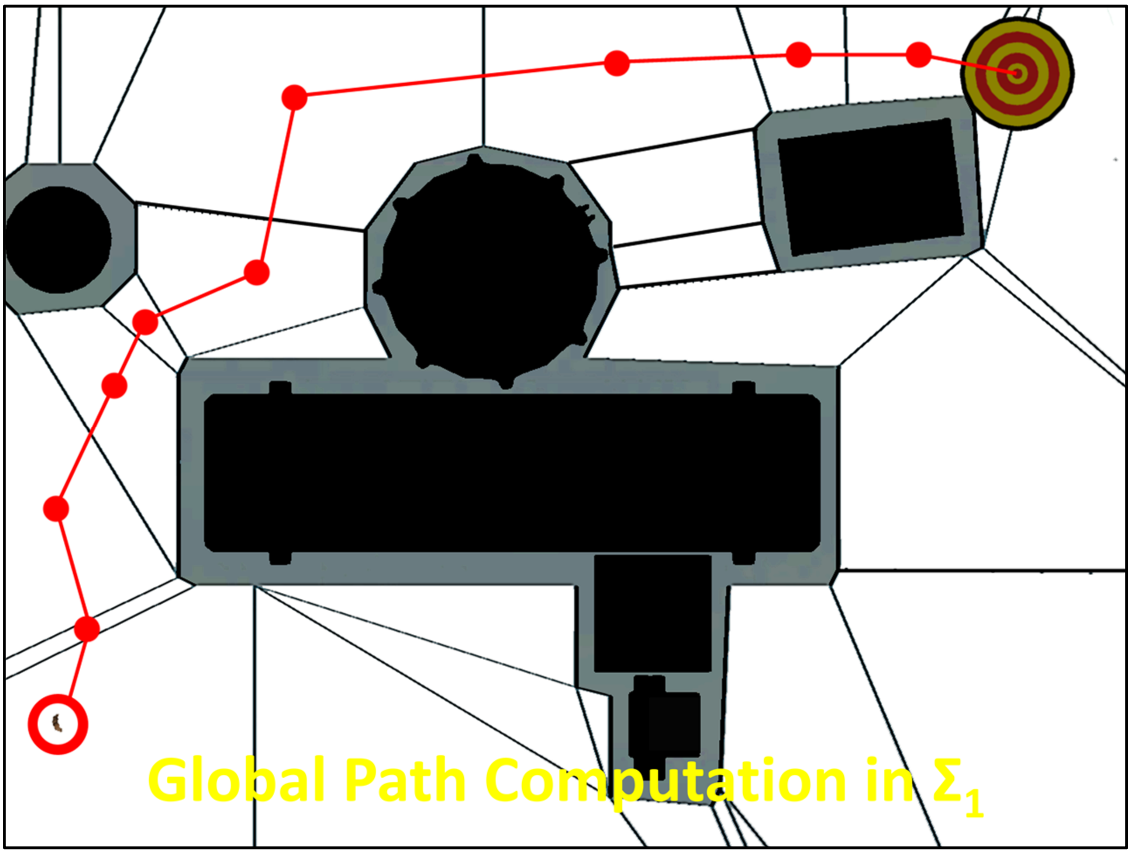
In order to maximize the benefits of each domain without suffering from computational overhead, we propose an anytime dynamic planning framework that can efficiently work across multiple domains, by using plans in one domain to accelerate and focus searches in more complex domains. And we explore different domain relationships including the use of waypoints and tunnels



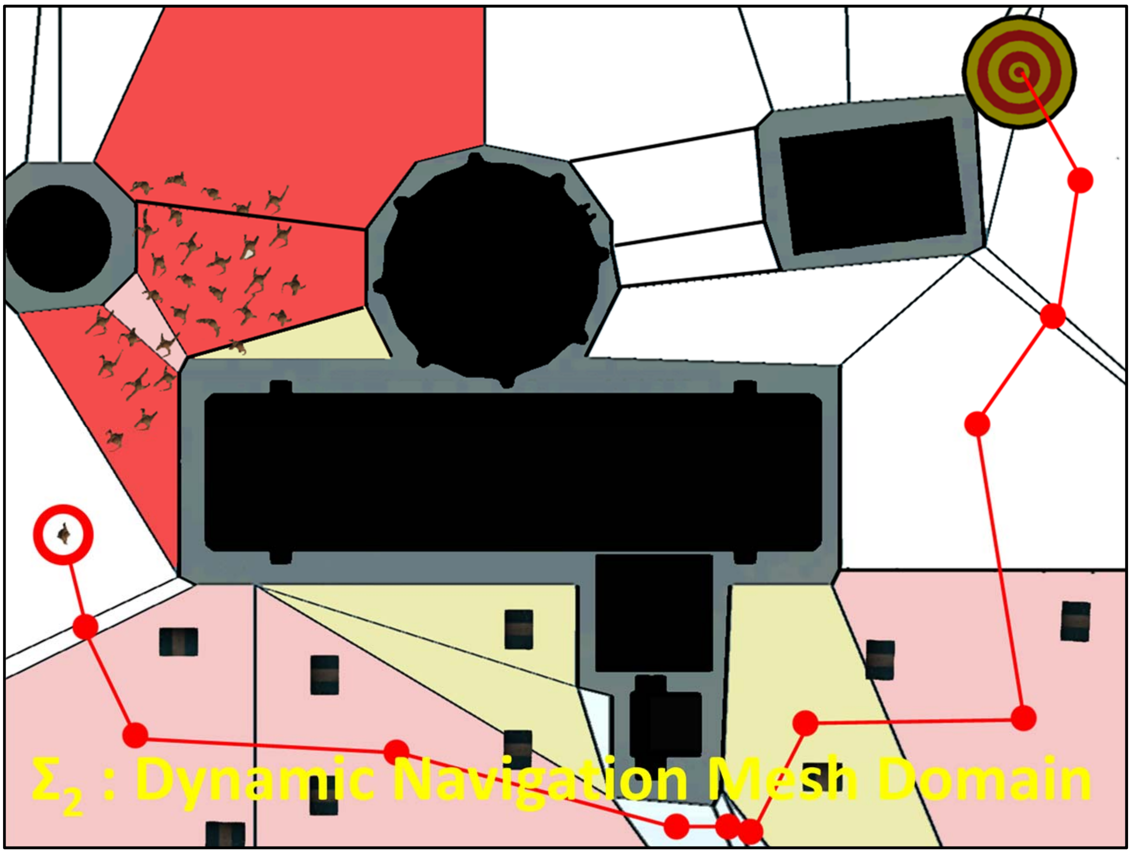
In order to showcase the ability of our framework to efficiently work across heterogeneous domains, we describe 4 domains which provide a nice balance between global static navigation and fine-grained space-time control of agents in dynamic environments.



The static navigation mesh domain uses a triangulated representation of free space and only considers static immovable geometry. The agent is modeled as a point mass, and valid transitions are between connected free spaces, represented as polygons. The cost function is the straight line distance between the center points of two free spaces. Additional connections are also precomputed (or manually annotated) to represent transitions such as jumping with a higher cost definition. The heuristic function is the Euclidean distance between a state and the goal. Searching for an optimal solution in this domain is very efficient and quickly provides a global path for the agent to navigate.

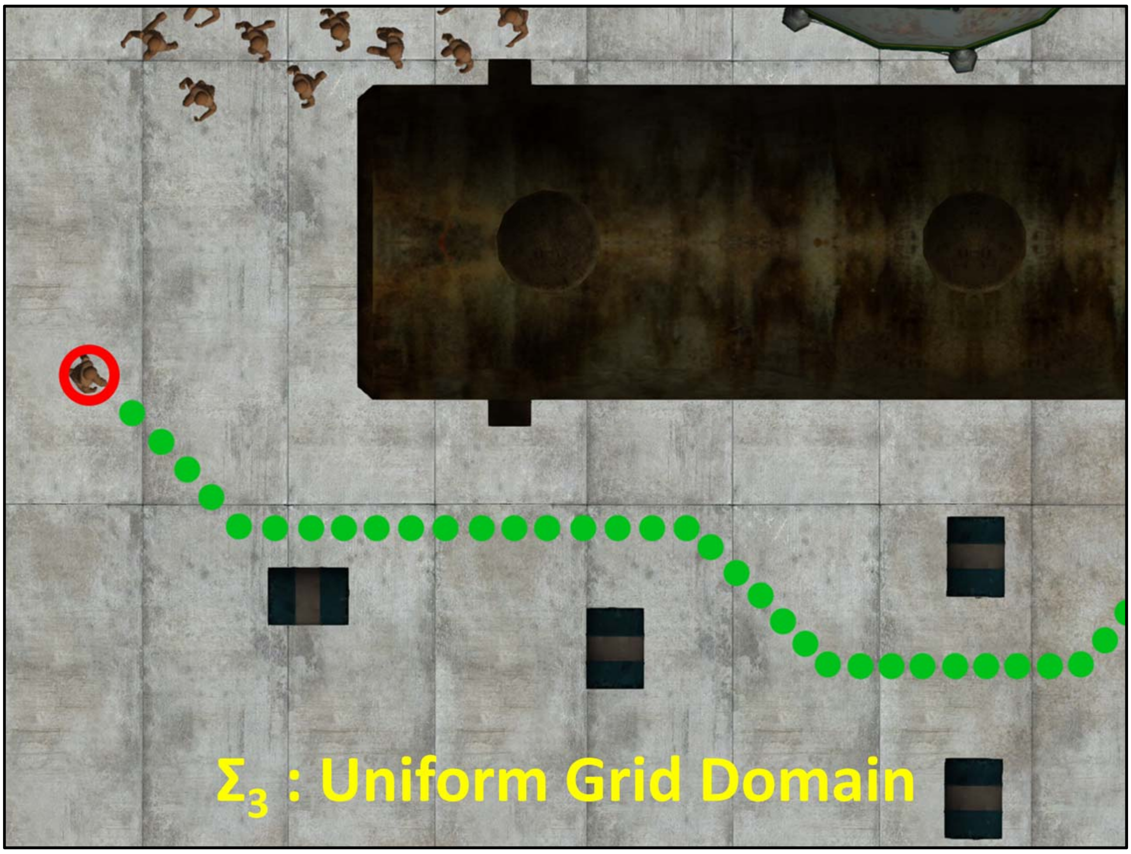


It can be used to compute global static paths very efficient for arbitrarily large environments

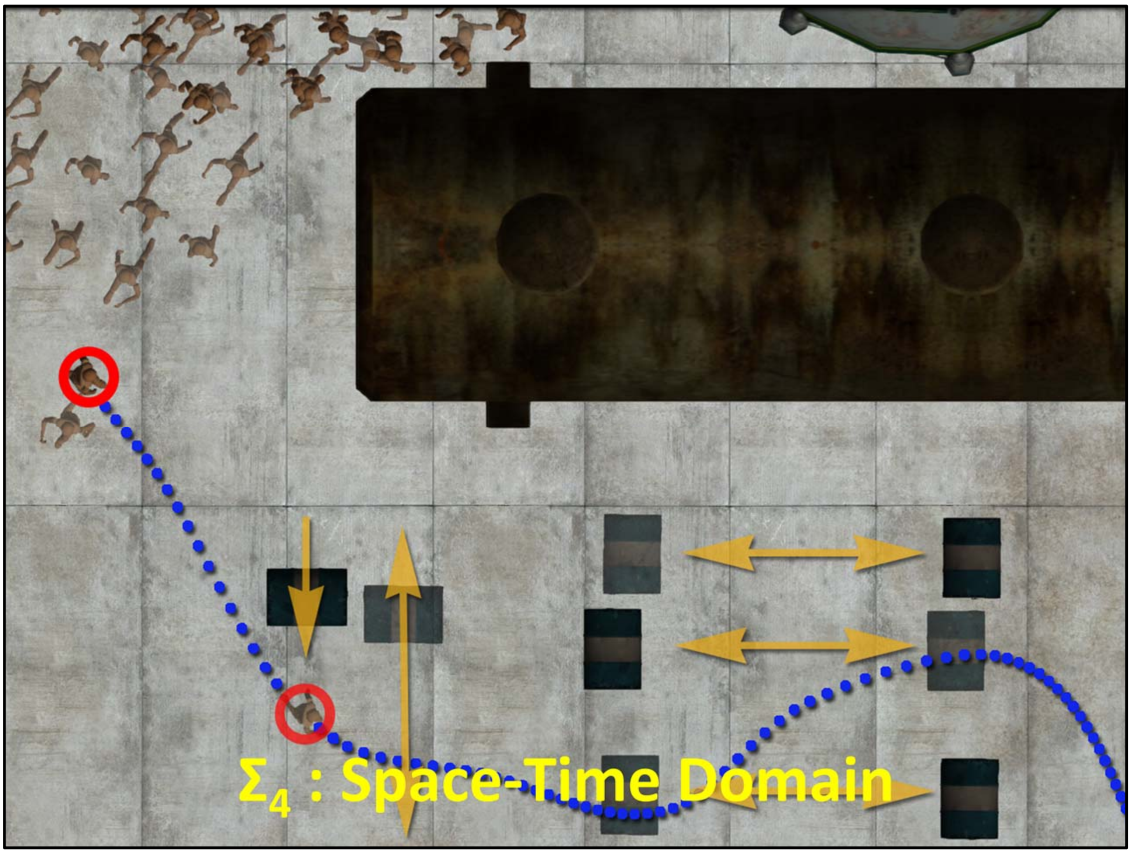


The dynamic navigation mesh also uses triangulations to represent free spaces and coarsely accounts for dynamic properties of the environment by storing a time-varying density field, which contributes to the cost of choosing a triangle for navigation.

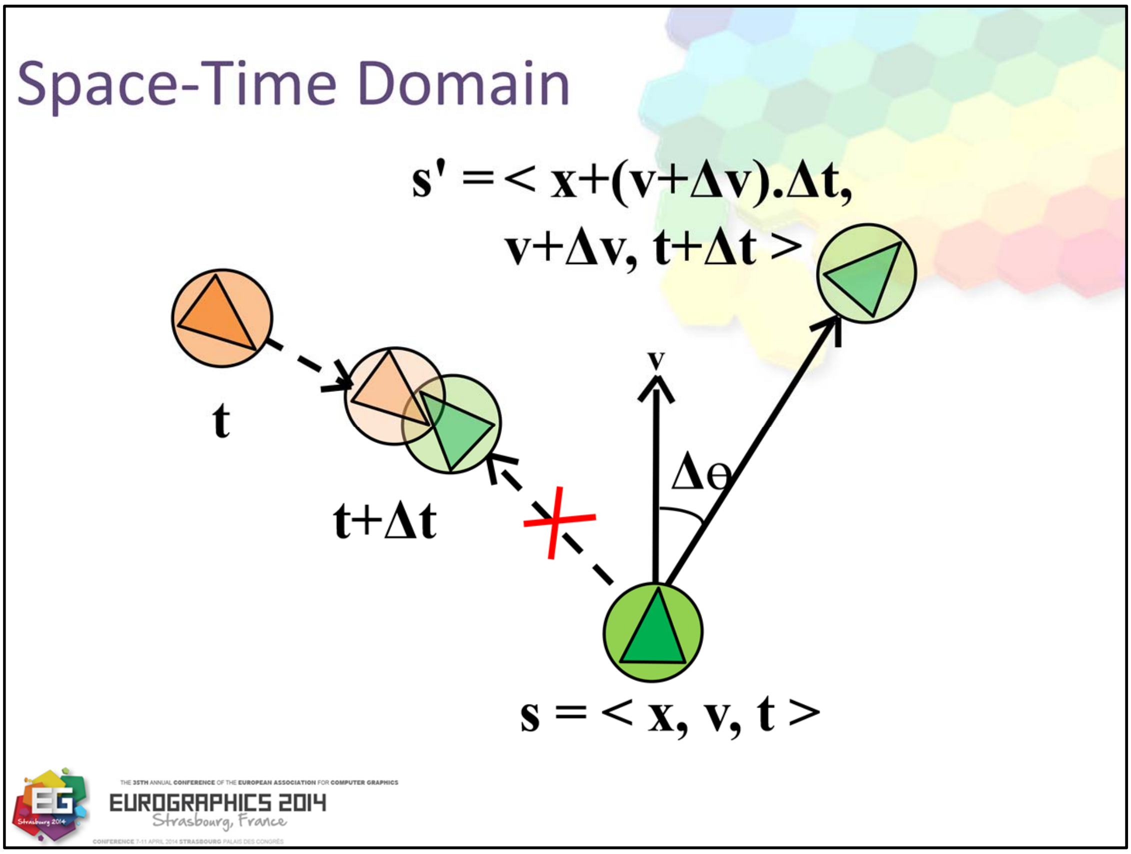
To make a more informed decision at the global planning layer, we define a time-varying density field which stores the density of moveable objects (agents and obstacles) for each polygon in the triangulation at some point of time t. The presence of objects and agents in polygons at future timesteps can be estimated by querying their plans (if available). The space-time positions of deterministic objects can be accurately queried while the future positions of agents can be approximated based on their current computed paths, assuming that they travel with constant speed along the path without deviation. The resolution of the triangulation may be kept finer than 1 to increase the resolution of the dynamic information in this domain. Hence, a set of global waypoints are chosen in this domain which avoids crowded areas or other high cost regions.



The grid domain discretizes the environment into grid cells where a valid transition is considered between adjacent cells that are free. This domain only accounts for the current position of dynamic obstacles and agents, and cannot predict collisions in space-time. An agent is modeled as a point with a radius (orientation and agent speed is not considered in this domain). The cost and heuristic are distance functions that measure the Eucledian distance between grid cells.



The space-time domain accounts for all obstacles (static and dynamic) and other agents. Transition validity queried in space-time by checking to see if moveable obstacles and agents occupy that position at that particular point of time, by using their published paths.



This figure illustrates how transitions are computed in this domain by discretizing your set of permissible velocities which don't produce a collision in space and time. This domain allows agents to compute paths that avoid collisions with dynamic obstacles with space-time precision.

Solving this problem using a coarse triangulation is insufficient as all information is not accounted for. The space-time domain models the necessary complexity but does not scale with environment size and number of agents. So, the question is how can we efficiently work in multiple domains ..

Domain Mapping

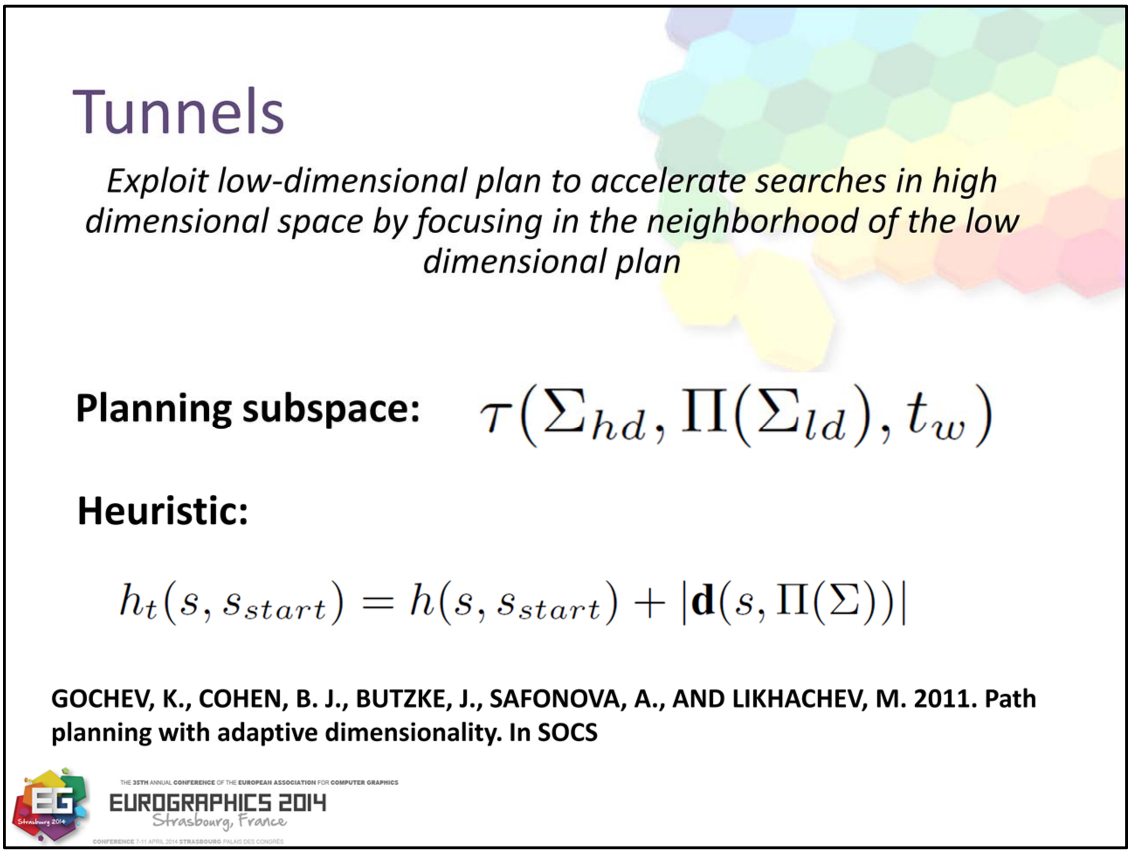
$$\lambda(s, \Sigma, \Sigma') : s \rightarrow \{s' | s' \in \mathbb{S}(\Sigma') \wedge s \equiv s'\}$$

$\lambda(s, \Sigma_1, \Sigma_2)$: polygon -> polygon(s) mapping

$\lambda(s, \Sigma_2, \Sigma_3)$: maps a polygon to multiple grid cells

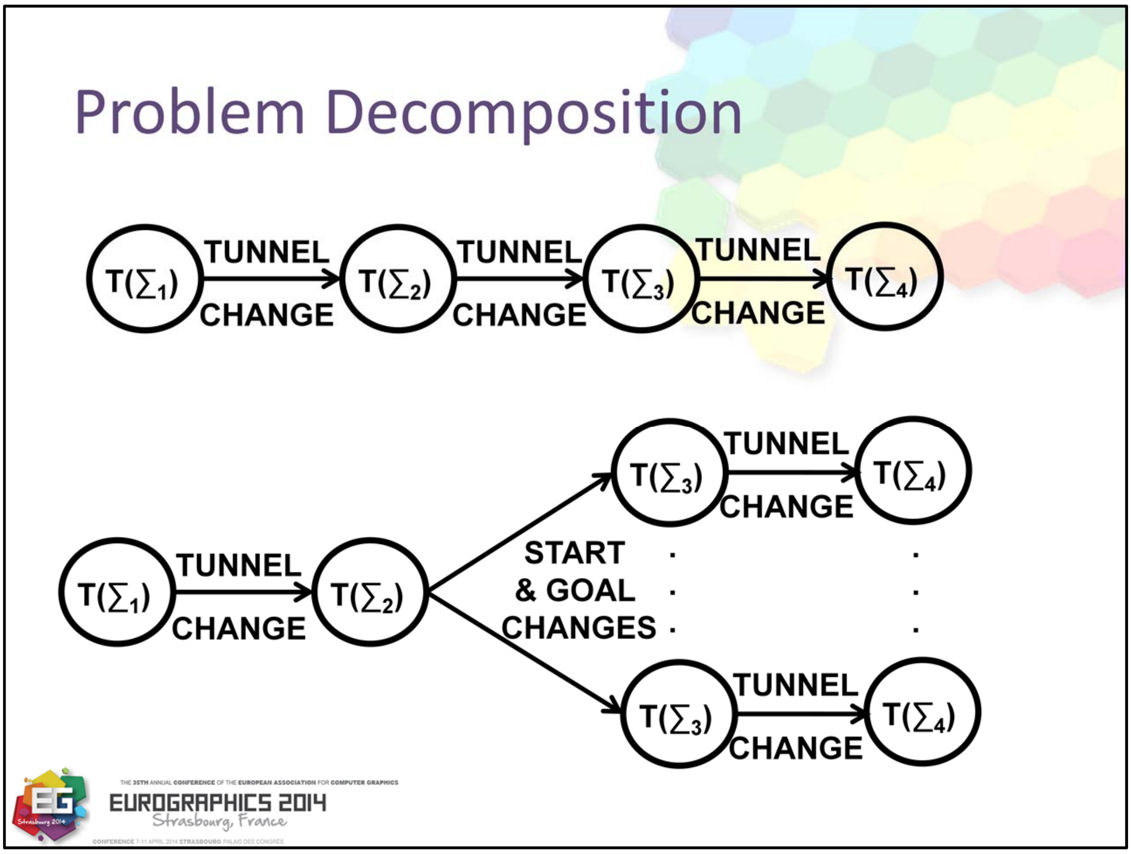
$\lambda(s, \Sigma_3, \Sigma_4) : (\mathbf{x}) \rightarrow \{(\mathbf{x} + W(\Delta\mathbf{x}), t + W(\Delta t))\}$

In order to use information across domains, we define a 1:n mapping function that maps a state in one domain to one or states in another domain. The mapping functions are defined specifically for each domain pair. The static navigation mesh domain maps a polygon to one or more polygons in the dynamic navigation mesh domain such that s0 is spatially contained in s. If the same triangulation is used for both 1 and 2, then there exists a one-to-one mapping between states.



Tunnels are a sub graph in the high dimensional space such that the distance of all states in the tunnel from the low dimensional plan is less than the tunnel width tw. Furthermore, node expansion can be prioritized to states that are closer to the path by modifying the heuristic.

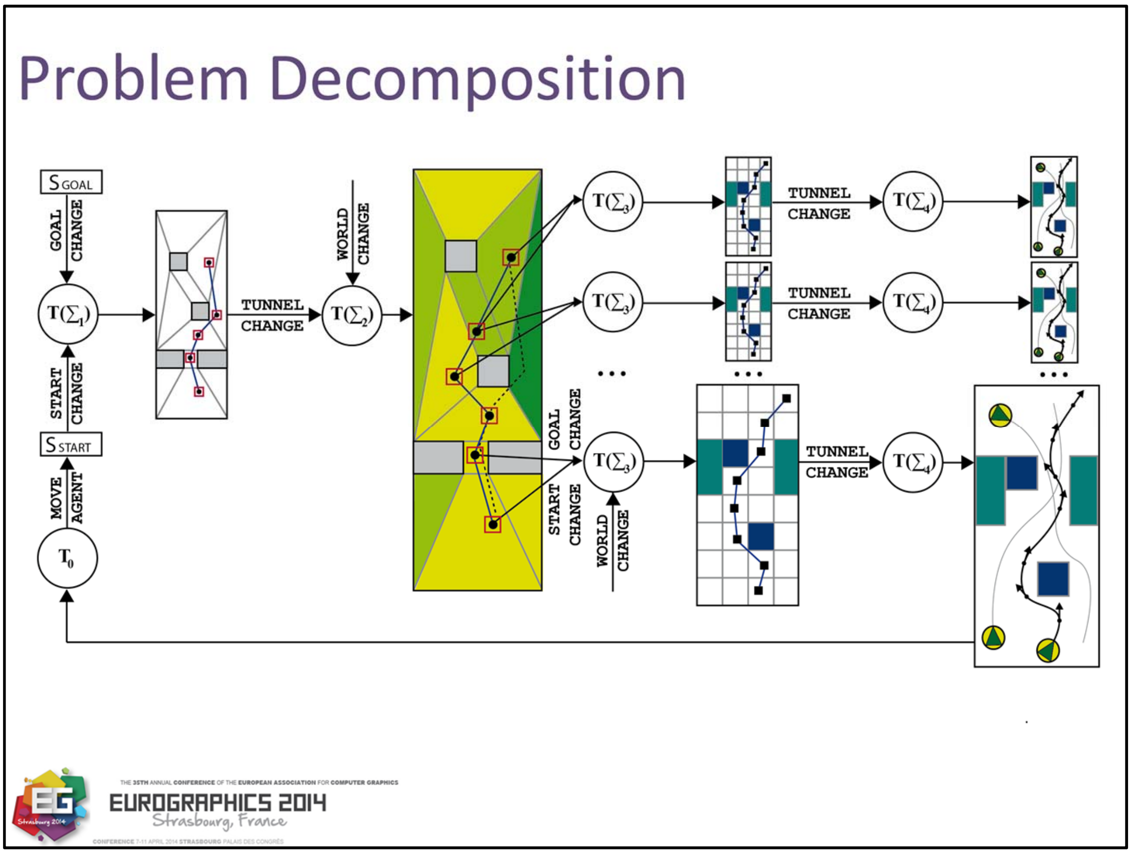
d is the perpendicular distance between s and the line segment connecting the two nearest states. d will return a two-tuple value for spatial distance as well as temporal distance.



Here are two example problem decompositions that use the 4 domains and the two different domain relationships that I described

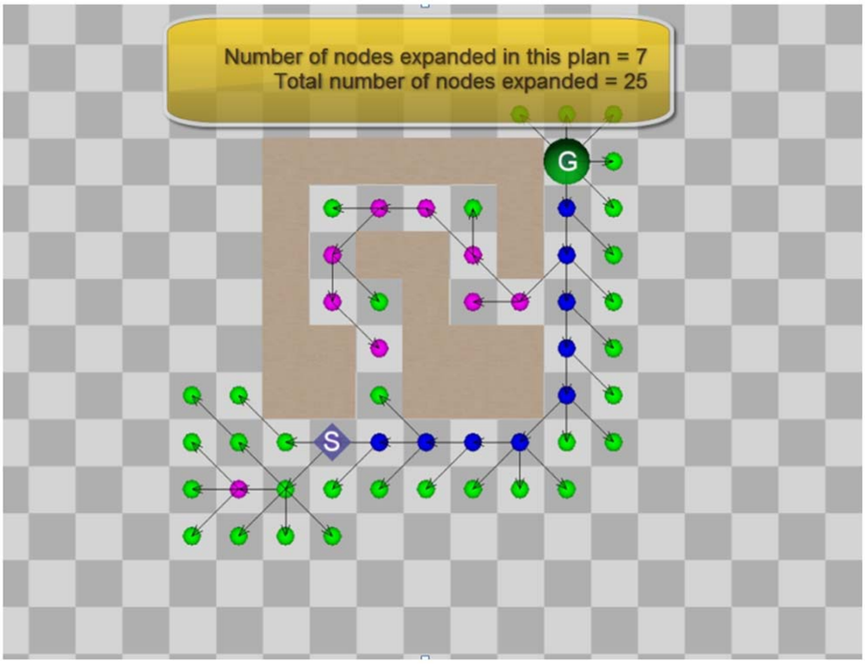
Figure 1 illustrates the use of tunnels to connect each of the 4 domains, ensuring that a complete path from the agents initial position to its global target is computed at all levels.

Figure 2 shows how the dynamic navigation mesh domain and grid domain are connected by using successive waypoints in the nav mesh plan as start and goal for independent planning tasks in the grid domain. This relation allows finer-resolution plans being computed between waypoints in an independent fashion. Limiting grid domain (and the space time domain) to plan between waypoints instead of the global problem instance ensures that the search horizon in these domains is never too large, and that fine-grained space-time trajectories to the initial waypoints are computed quickly. However, completeness and optimality guarantees are relaxed as these domains never compute a single path to the global target.



Here is expanded view of the second decomposition method which shows the original problem instance decomposed into a set of independent planning tasks working across multiple problem domains. To is simply an execution task that simulates the motion of the agent along the closest space-time plan

Planning Task



LIKHACHEV, M., FERGUSON, D. I., GORDON, G. J., STENTZ, A., AND THRUN, S. 2005. Anytime Dynamic A*: An Anytime, Replanning Algorithm. ICAPS.

To solve each individual problem, We use an anytime dynamic planner that combines the properties of anytime and incremental planning. It first quickly generates a suboptimal solution while meeting strict time constraints and iteratively refines the solution if time permits. It can also efficiently repair plans to accommodate world changes as well as start movement which allows it to interleave planning with execution.

Task Priorities

$$p(\mathtt{T}_a) = \begin{cases} 1 \text{ if } \mathtt{T}_a = \mathtt{T}_0 \\ \mu(\mathtt{T}_a, \mathtt{T}_0) \cdot \Omega(\mathtt{T}_a) \text{ else} \end{cases}$$

$$\Omega(\mathtt{T}_a) = \begin{cases} 1 \text{ if } \mathtt{SOLUTION\_INVALID} \\ \epsilon \text{ if plan inflation factor, } \epsilon > 1 \\ \infty \text{ if plan inflation factor, } \epsilon = 1 \end{cases}$$

Now that we have a pool of planning tasks, we use a simple priority scheme that schedules the execution of these tasks. The priority of a task determines the tasks that are picked to be executed at every time step, with tasks having smallest priority chosen for execution. Task T0, which handles agent movement always has a priority of 1.

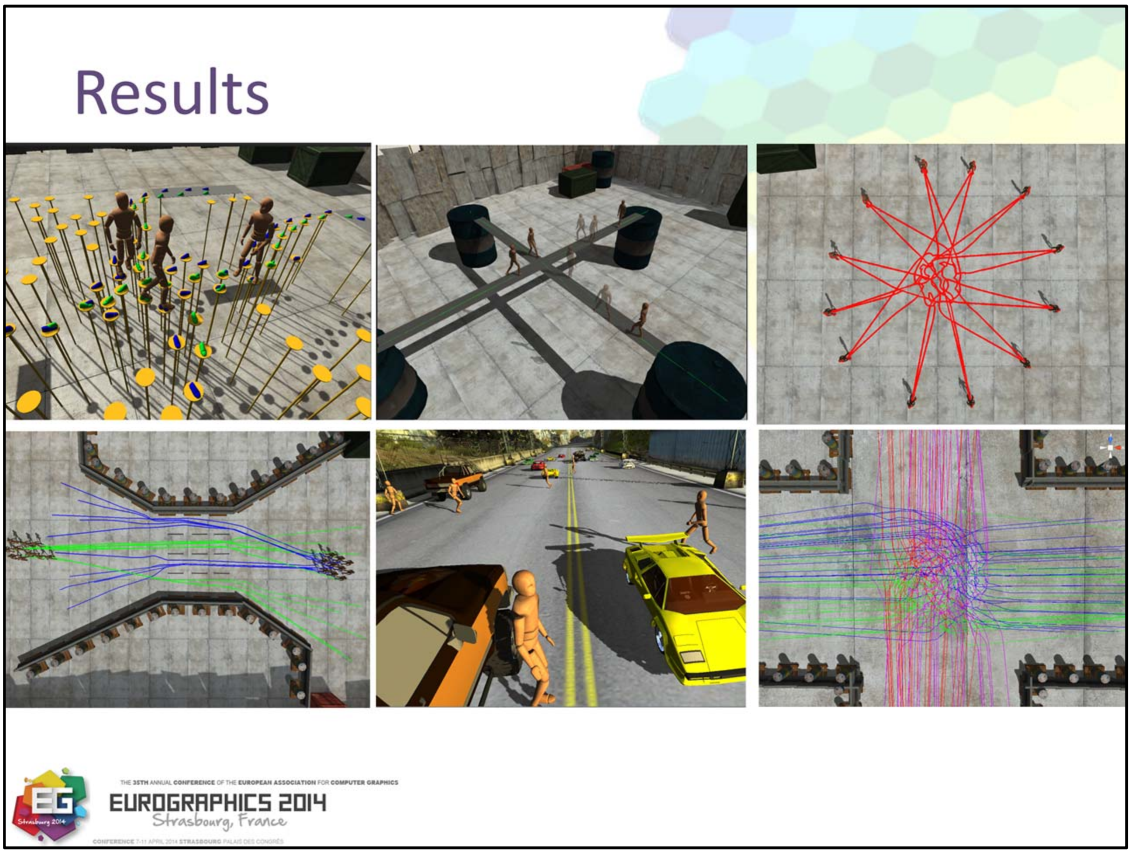
Priority of other tasks is calculated based on the number of edge traversals required to reach T0, and the current state of its computed plan. The agent pops one or more tasks that have highest priority and divides the deliberation time available across tasks, with execution-critical tasks receiving more time. Tasks that have the same priority are ordered based on task dependency. Hence, T0 is always executed at the end of every update after all planning tasks have completed. The overall framework enforces strict time constraints. Given an allocated time to deliberate for each agent (computed based on desired frame rate and number of agents), the time resource is distributed based on task priority. In the remote event that there is no action to execute, the agent remains stationary (no impact on frame-rate) for a few frames (fractions of a second) until a valid plan is computed.

Evaluation of Domain Relationships

Domain	BF	N	T	S	Q
$T(\Sigma_1)$	3.7	43	3	0.17	0.76
$T(\Sigma_2)$	4.6	85	8	0.23	0.57
$T(\Sigma_2, \Pi(\Sigma_1))$	2.1	17	5	0.32	0.65
$T(\Sigma_3)$	7.4	187	18	0.68	0.73
$T(\Sigma_4)$	21.5	10^4	2487	0.34	0.26
$T(\Sigma_4, \Pi(\Sigma_3, \Sigma_2, \Sigma_1))$	5.6	765	136	0.92	0.64
$\sum T_i(\Sigma_4, \Pi(\Sigma_3, \Sigma_2, \Sigma_1))$	5.4	75	8	0.86	0.58

Comparative evaluation of the domains, and the use of multiple domains.
BF = Effective branching factor.
N = Average number of nodes expanded.
T = Average time to compute plan (ms).
S = Success rate of planner to produce collision-free trajectory.
Q = Plan quality.

In an effort to find the optimal domain relationship for our application, We randomly generate 1000 scenarios of size 100 m X 100 m, with random configurations of obstacles (both static and dynamic), start state, and goal state and record the effective branching factor, number of nodes expanded, time to compute a plan, success rate, and quality of the plans obtained. The effective branching factor is the average number of successors that were generated over the course of one search. Plan quality is the ratio of the length of the static optimal path and the path obtained. A plan quality of 1 indicates that the solution obtained was able to minimize distance without any deviations. Similar metrics for analyzing multi-agent simulations have been used in [Kapadia et al. 2011]. Rows 3 and 6 include the added time to compute plans in earlier domains for tunnel search, to provide an absolute basis of comparison. All experiments were performed on a single-threaded 2.80 GHz Intel(R) Core(TM) i7 CPU.



Deadlocks. Multiple oncoming and crossing agents in narrow passageways cooperate with each other with space-time precision to prevent potential deadlocks. Agents observe the presence of dynamic entities at waypoints along their global path and refine their plan if they notice potentially blocked passageways or other high cost situations. Crowd simulators deadlock for these scenarios, while a space-time planner does not scale well for many agents or be able to efficiently replan due to unpredictable world changes

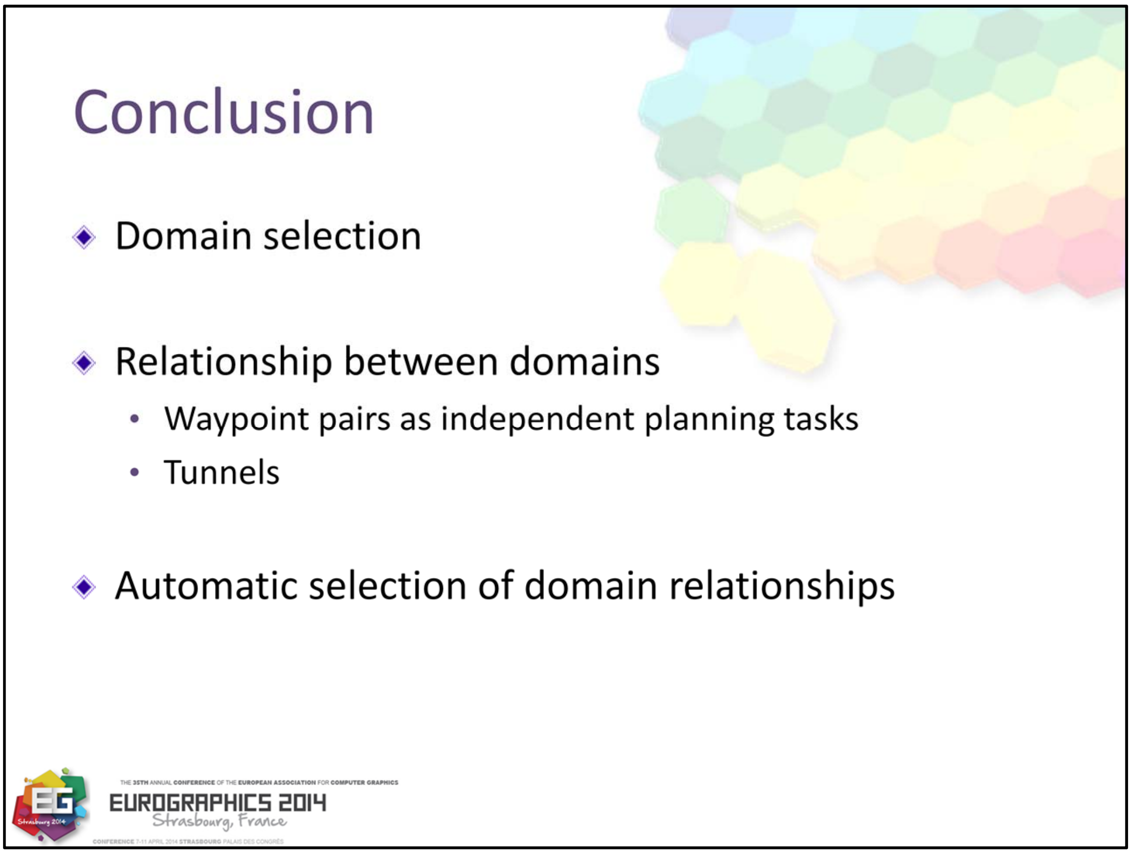
Choke Points. This scenario shows our approach handling agents arriving at a common meeting point at the same time, producing collision-free straight trajectories. Our framework produces considerably smoother trajectories and minimizes deviation by using subtle speed variations to avoid collisions in space-time.

Unpredictable Environment Change. Our method efficiently repairs solutions in the presence of unpredictable world events, such as the user-placement of obstacles or other agents, which may invalidate current paths.

Road Crossing. The road crossing scenario demonstrates 40 agents using space-time planning to avoid fast moving vehicles and other crossing agents.

Lane Selection for Bi-directional Traffic. This scenario requires agents to make a navigation decision in choosing one of 4 lanes created by the dividers. Agents distribute themselves among the lanes, while bi-directional traffic chooses different lanes to avoid deadlocks. This scenario requires non-deterministic dynamic information (other agents) to be accounted for while making global navigation decisions. This is different from emergent lane formation in crowd approaches, which bottlenecks at the lanes and cause deadlocks without a more robust navigation technique.

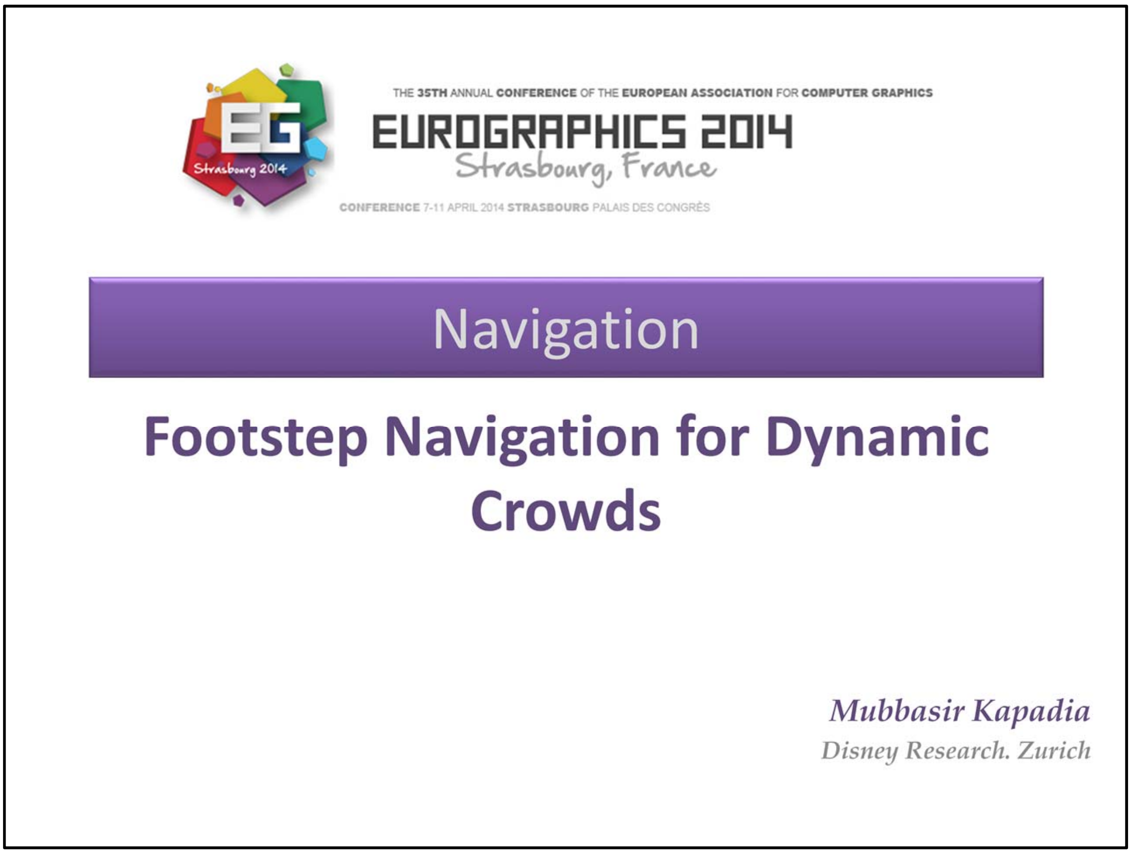
Four-way Crossing We simulate 100 oncoming and crossing agents in a four-way crossing. By virtue of predicting dynamic collisions at the global plan layer and trickling that information down to the finer resolution plans, We see much smoother trajectories produced using our framework. A predictive steering algorithm only accounts for imminent neighboring threats and is unable to avoid mingling with the other groups.



We have introduced a framework for real-time, multi-agent navigation in large-scale, complex, dynamic environments, with space-time precision. We use multiple problem domains to provide a balance between control fidelity and computational complexity by accounting for dynamic aspects of the environment at all stages of decision-making. The original navigation problem is decomposed into a set of smaller problems that are distributed across planning tasks working in these different domains. An anytime dynamic planner is used to efficiently compute and repair plans for each of these tasks, and the use of tunnel based search is particularly useful for working in complex domains.

Choice of Domains. The domains described in this paper represent popular solutions that are used in both academia and industry. These domains provide a nice balance between global navigation and space-time planning, enabling us to showcase the strength of our framework: the ability to use multiple domains of control, and leverage solutions across domains to accelerate computations while still providing a high degree of control fidelity. Additional domains can be easily integrated (e.g., a footstep domain) to meet application-specific needs, or solve more challenging motion planning problems.

Relationship Between Domains. Domains can be connected by using the plan from one domain as a tunnel for the other, or by using successive waypoints along the plan as start and goal pair for multiple planning tasks in a more complex domain. We evaluated both domain relationships based on computational efficiency and coverage. Using waypoints from the navigation mesh domain as start, goal pairs for planning tasks in the grid and space-time domain keeps the search depth within reasonable bounds. The tradeoff is that a space-time plan is never generated at a global level from an agent's start position to its target, thus sacrificing completeness guarantees. This design choice worked well for our experiments where the reduction in success rate of our framework when using this scheme was within reasonable bounds, while providing a considerable performance boost, making it suitable for practical game-like applications. Users may wish to opt for different domain relationships depending on the application.

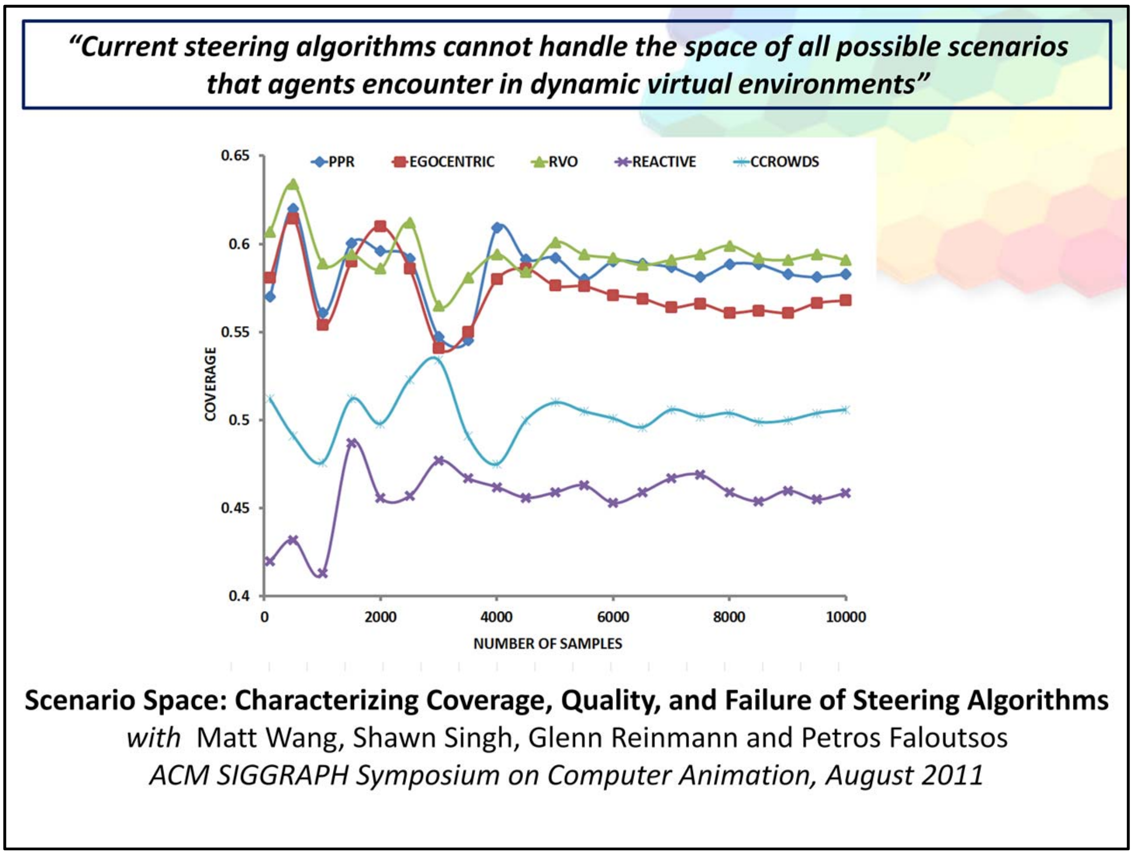


Speaker: Mubbasir Kapadia, Associate Research Scientist, Disney Research Zurich

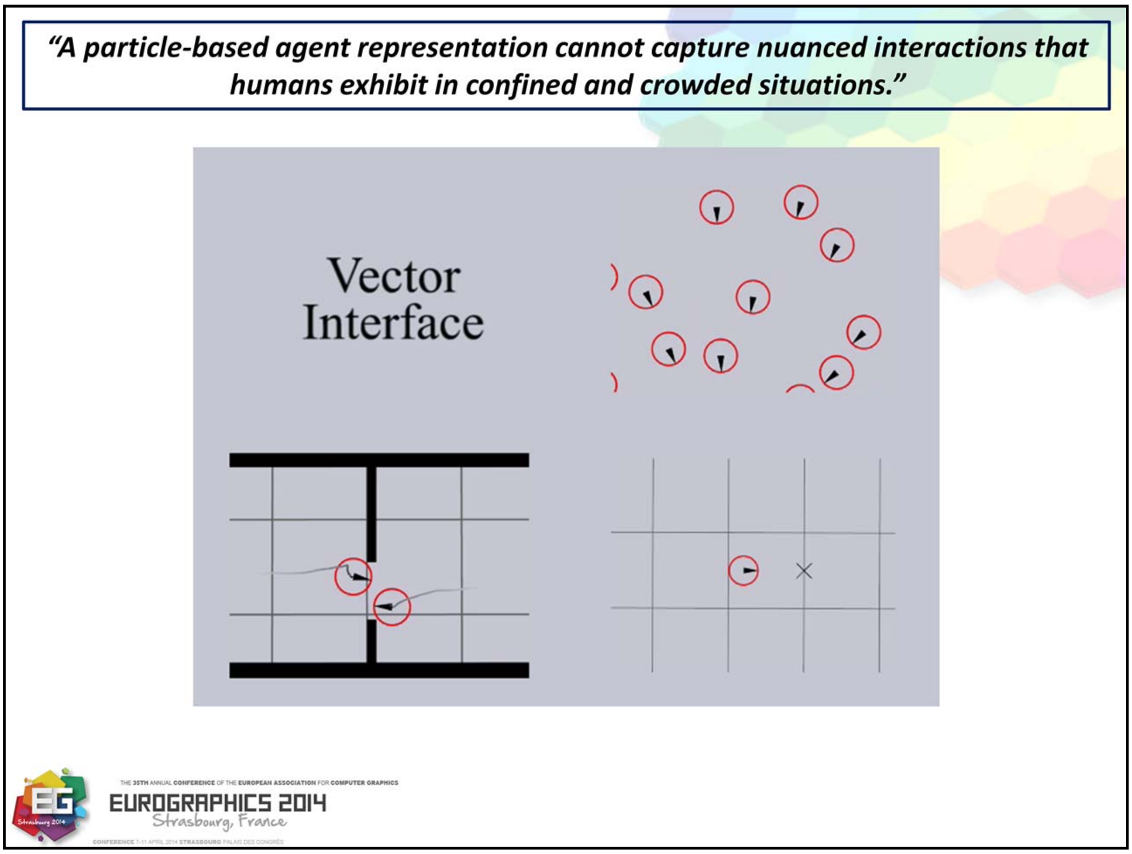
Related Publication(s):

Footstep Navigation for Dynamic Crowds
Shawn Singh, Mubbasir Kapadia, Glenn Reinman and Petros Faloutsos
ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (Poster Proceedings), February 2011

Project webpage: http://people.inf.ethz.ch/kapadiam/projects-footstep.html



A simple particle representation and a vector control interface cannot capture all the capabilities of human locomotion such as side-stepping and careful foot placement. Current state of the art in steering handles only a fraction (60%) of the scenarios that agents encounter in dynamic virtual environments. This graph plots the coverage of different steering algorithms defined as the ratio of the number of scenarios that the algorithm can successfully solve. One of the key observations of this analysis was that a majority of the steering algorithms can solve only about 60% of the represenative scenarios.



The simplification in agent representation and reactive steering policies produces artifacts such as sliding in crowded situations, livelocks at narrow passages, and violation of orientation constraints.

State and Action Space

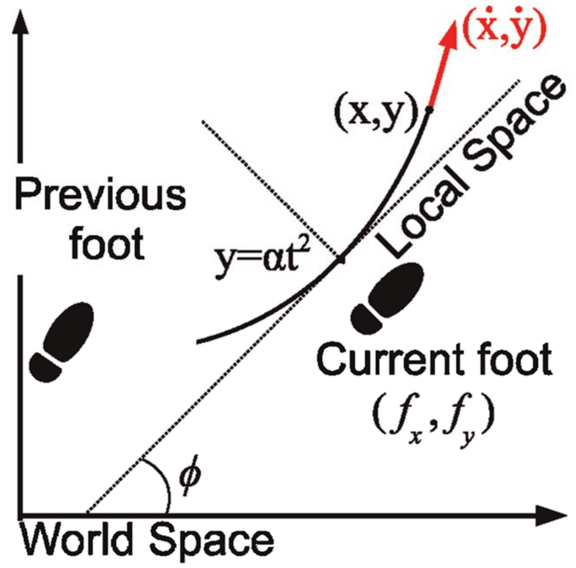
$$\mathbf{S} = \{s|\mathbf{x}, \mathbf{v}, \mathbf{x}_f, \theta_f, I \in \{L, R\}\}$$

$$\mathbf{A} = \{a|\phi, v_{des}, T\}$$

Footstep Domain

Our work generates bio-mechanically plausible footsteps for an agent to navigate in complex dynamic environments providing a tighter coupling with animation producing natural human-like collision avoidance. The state space is extended to include the position and orientation of the support foot. And an indicator of whether it is the left or right foot. Action selection involves 3 parameters: (1) The time period of the flight phase, (2) Desired character speed at the end of the flight phase (3) And orientation of the COM trajectory during the flight phase.

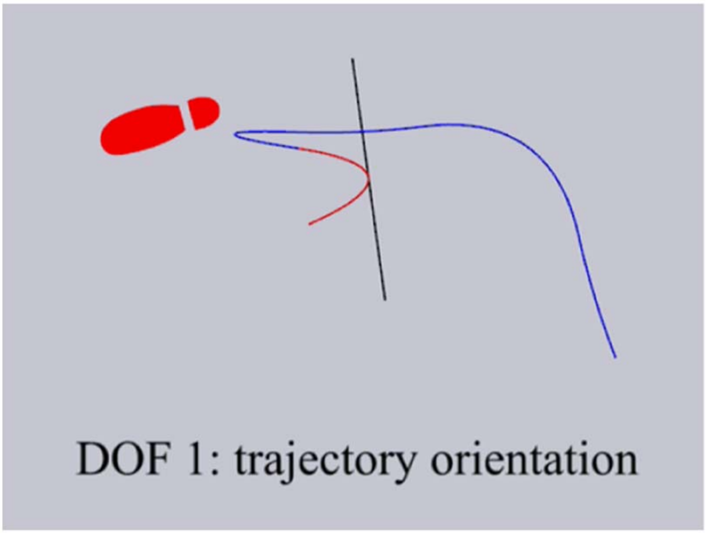
Footstep Action Selection



$$\left(x(t), y(t), \dot{x}(t), \dot{y}(t)\right) = \left(v_{x_0} t, \alpha t^2, v_{x_0}, 2\alpha t\right)$$

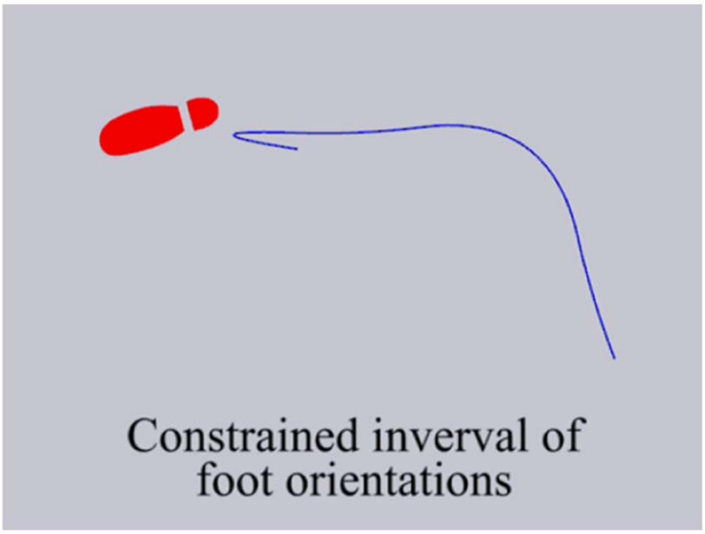
The movement of the character over the duration of a single footstep is defined as a 2D analytical approximation to the dynamics of an inverted spherical pendulum using parabolas. Piecewise parabolic curves are enough to capture the variety of trajectories that a human's center of mass will have: varying curvature, speed, and step sizes.

Footstep Action Selection



By discretizing the trajectory orientation, desired speed, and footstep time period we can define the action space of possible footsteps which can be sampled for footstep selection. Each step motion is defined as a parabola which allows us to analytically determine the COM position which is very useful for collision checks over the course of a footstep action.

Footstep Orientation



$$[f_{\phi\text{next\_inner}}, f_{\phi\text{next\_outer}}] = \left[f_{\phi\text{prev\_outer}}, f_{\phi\text{prev\_inner}} + \frac{\pi}{2}\right] \cap [\phi, \operatorname{atan2}(\dot{y}, \dot{x})]$$

Given the prev step and chosen trajectory we can compute an interval of bio-mechanically valid foot orientations for the current step which is resolved as a simple post process. This in turn constrains the parabola orientation for the next step. This orientation selection instead of expanding the branching factor of the search effectively reduces it.

Cost Formulation

Cost Function

$$c(s, s') = \Delta E_1 + \Delta E_2 + \Delta E_3$$
$$\Delta E_1 = R \cdot T$$
$$\Delta E_2 = \frac{m}{2} \left| (v_{\text{desired}})^2 - (v_0 \cos(2\theta))^2 \right|$$
$$\Delta E_3 = w \cdot \frac{d\mathrm{P}}{dt} \cdot \text{length} = w \cdot m\alpha \cdot \text{length}$$

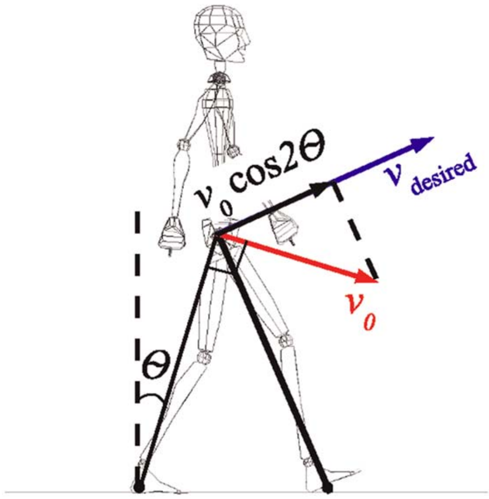
Heuristic Function

$$h(s) = c_{\text{expected}} \times n$$

In order to define an optimality criteria, we define 3 cost measures for energy expenditure over the course of a foostep. E1, a fixed rate of energy the character spends per unit time. The user defines a fixed rate of energy spent per second, denoted as R which is multiplied by the time duration. This cost is proportional to the amount of time it takes to reach the goal, and thus minimizing this cost corresponds to the character trying to minimize the time it spends walking to his goal. We found that good values for R are roughly proportional to the character's mass.
E2 captures only the cost of changing a character's momentum at the beginning of each step. The character's momentum may also change during the trajectory. For relatively straight trajectories, this change in momentum is mostly due to the passive inverted pendulum dynamics that requires no active work. However, for trajectories of high curvature, a character spends additional energy to change his momentum. We model this cost as the work required to change momentum (denoted as P) over the length of the step, weighted by constant w.

Alpha is the acceleration of the COM trajectory. α increases if the curvature of the parabola is larger, and also if the speed of the character along the trajectory is larger. Minimizing this cost corresponds to preferring straight steps when possible, and preferring to go slower (and consequently, taking smaller steps) when changing the direction of momentum significantly. The weight w can be adjusted to change whether it costs more energy to walk around an obstacle or to stop and wait for the obstacle to pass. We found good values of w to be between 0.2 and 0.5, meaning that 20 to 50 per cent of the curvature is due to the character's active effort, and the rest due to the passive inverted pendulum dynamics.

Spherical Inverted Pendulum Model – Sagittal View

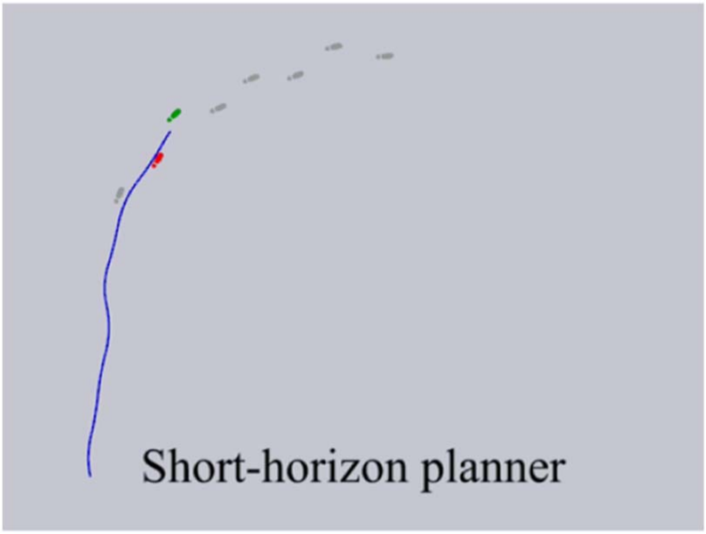


$$\Delta E_2 = \frac{m}{2}\left|(v_{\text{desired}})^2 - (v_0 \cos(2\theta))^2\right|$$

The second cost accounts for the energy required to maintain a desired speed. First, at the beginning of a new step (heelstrike), some of the character's momentum dissipates into the ground. We estimate this as an instantaneous loss of momentum along the pendulum shaft, reducing the character's speed. In order to resume a desired speed, the character actively exerts additional work on his center of mass, computed as shown. The opposite is also true. If a character wants to come to an immediate stop, work is required to remove energy from the system. Minimizing this cost corresponds to finding footsteps that require less effort, and thus tend to look more natural. When taking large strides, cos 2 theta gets smaller, implying greater energy expenditure – resulting in more natural stride lengths.

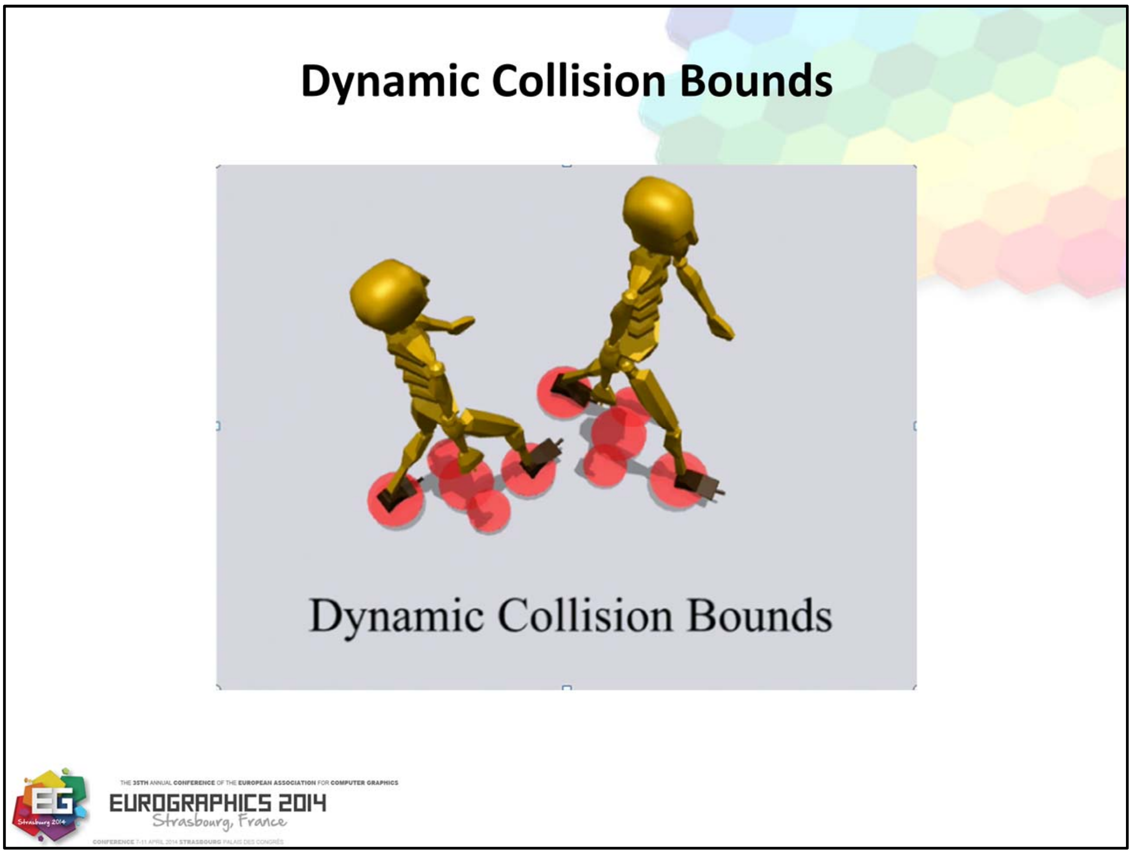
It should be noted that there is much more complexity to real bipedal locomotion than this cost model. For example, the appropriate bending of knees and ankles and the elasticity of human joints can significantly reduce the energy lost per step, reducing the required work for a real human. While the model is not an accurate measurement of energy spent, it is sufficient for comparing the effort of different steps.

Short Horizon Space-Time Planner

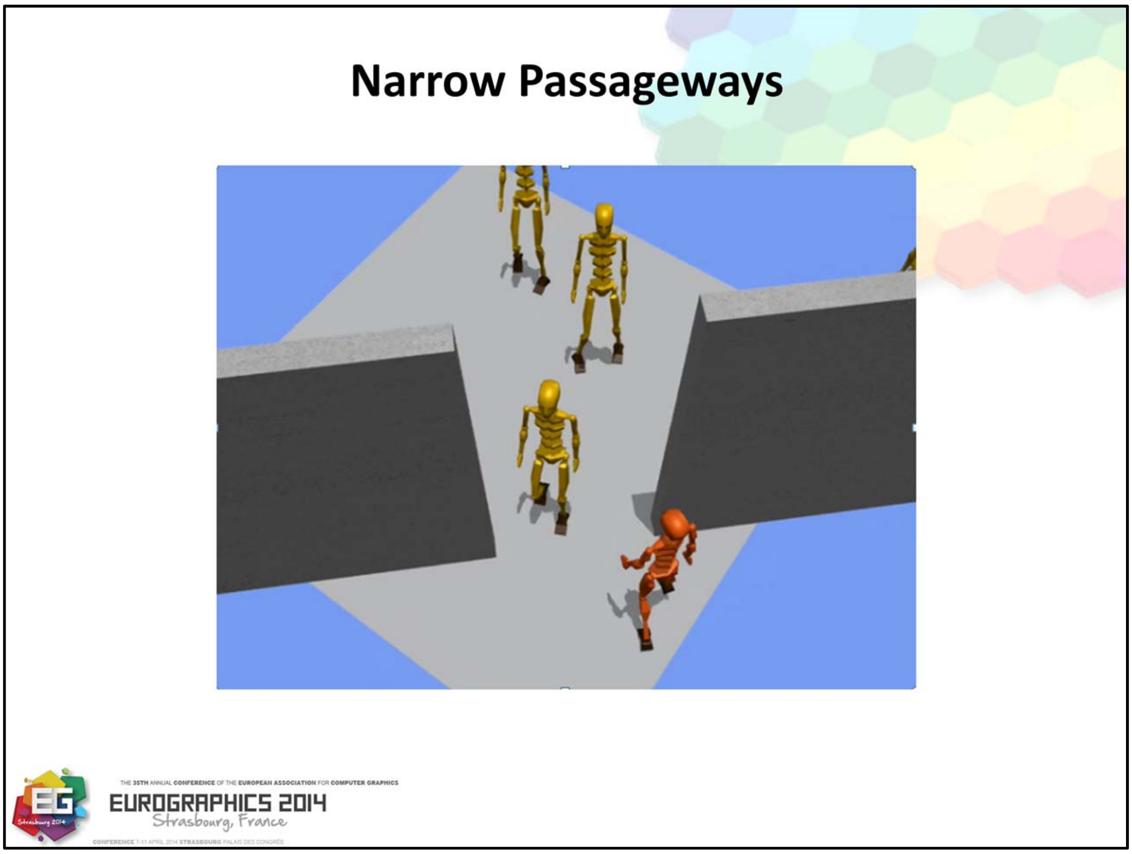


We use a short horizon space time planner to make footstep decisions in a goal-directed fashion while predictively avoiding collisions with other agents with space time precision. We limit the horizon of the planner to a maximum number of nodes to expand before returning a solution. This number is a factor of the desired frame rate of the simulation. For our experiments, we find that the planner returns a solution in most cases. If it reaches the horizon, it returns an incomplete plan corresponding to the most promising node that was reached during the search towards the goal.

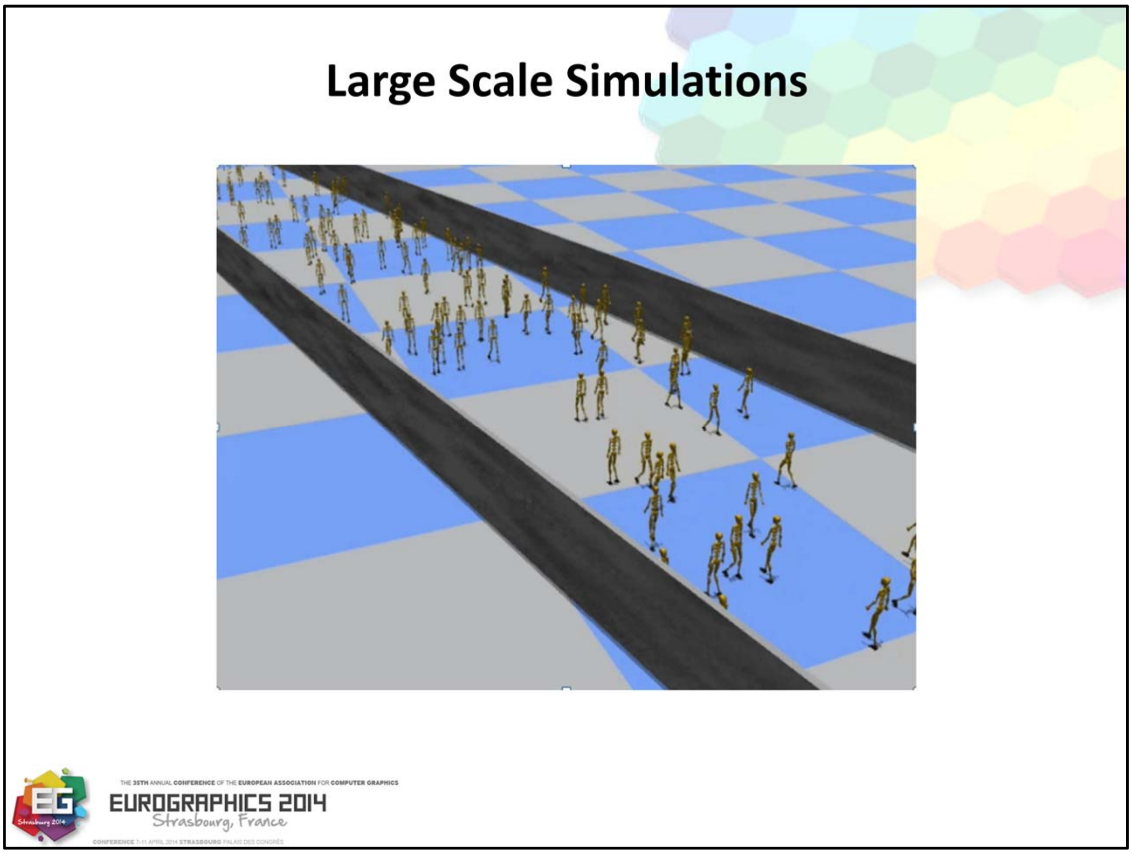
The short horizon approach guarantees that we will have at least some path for the character to use, even in difficult or unsolvable planning problems. In worst case, if no good solution is found, the path will simply be a sequence of ‘stop’ actions. For example, this can occur when a character is stuck dense environment. Eventually when the density clears, the character will continue.



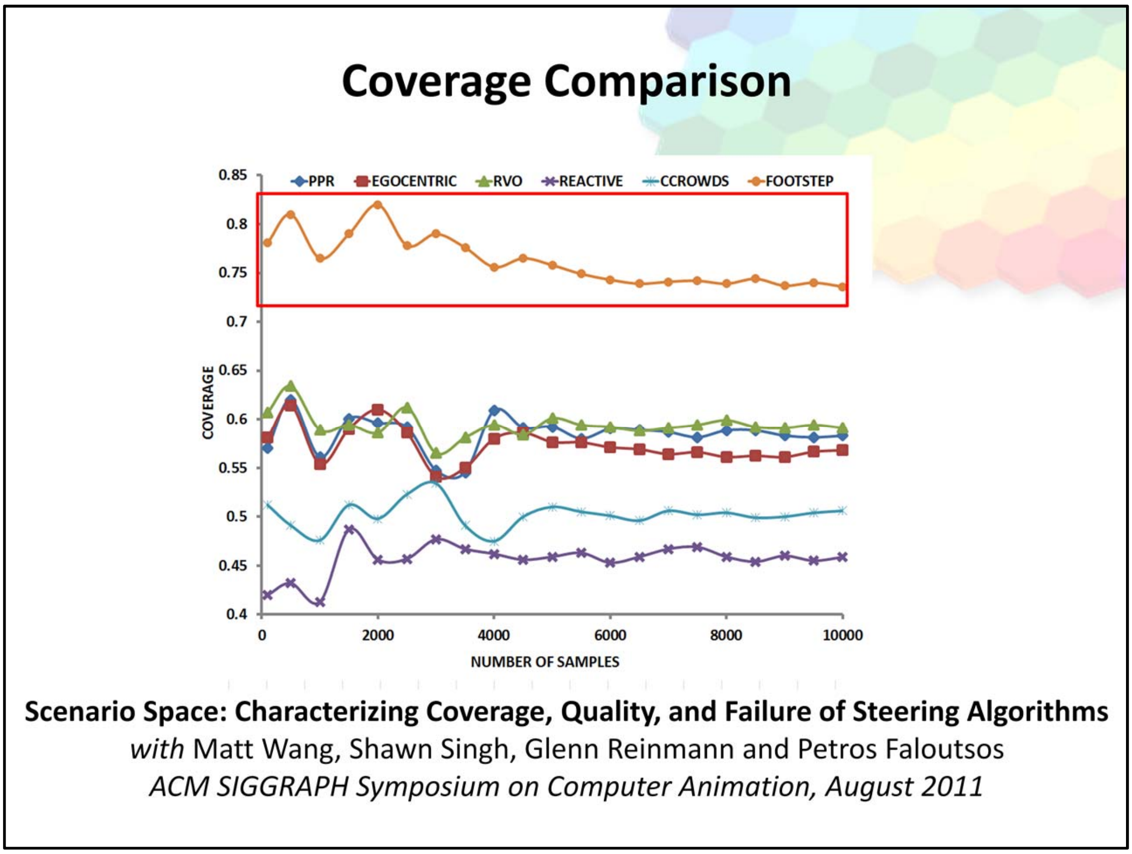
Dynamic collision bounds allow characters to get closer to each other. Characters can pack much closer together in egress simulations without colliding in contrast to a simple collision radius model.



Characters cooperate at doorways with human-like fluidity and precision. This doorway is so narrow that a single character can barely fit through it. Vector-based (particle based) steering algorithms would collide with the wall before eventually squeezing through.



Inspite of the increased domain and control complexity, by limiting the horizon of the space-time planner, Pruning the next possible footsteps based on the current state And amortizing the cost of computing a footstep plan over several frames, our framework can simulate a 1000 characters at 20fps.



The coverage of this approach is considerably greater than previous solutions, demonstrating the benefits of using a biomechanically plausible footstep domain for crowd navigation.

THE 35TH ANNUAL CONFERENCE OF THE EUROPEAN ASSOCIATION FOR COMPUTER GRAPHICS

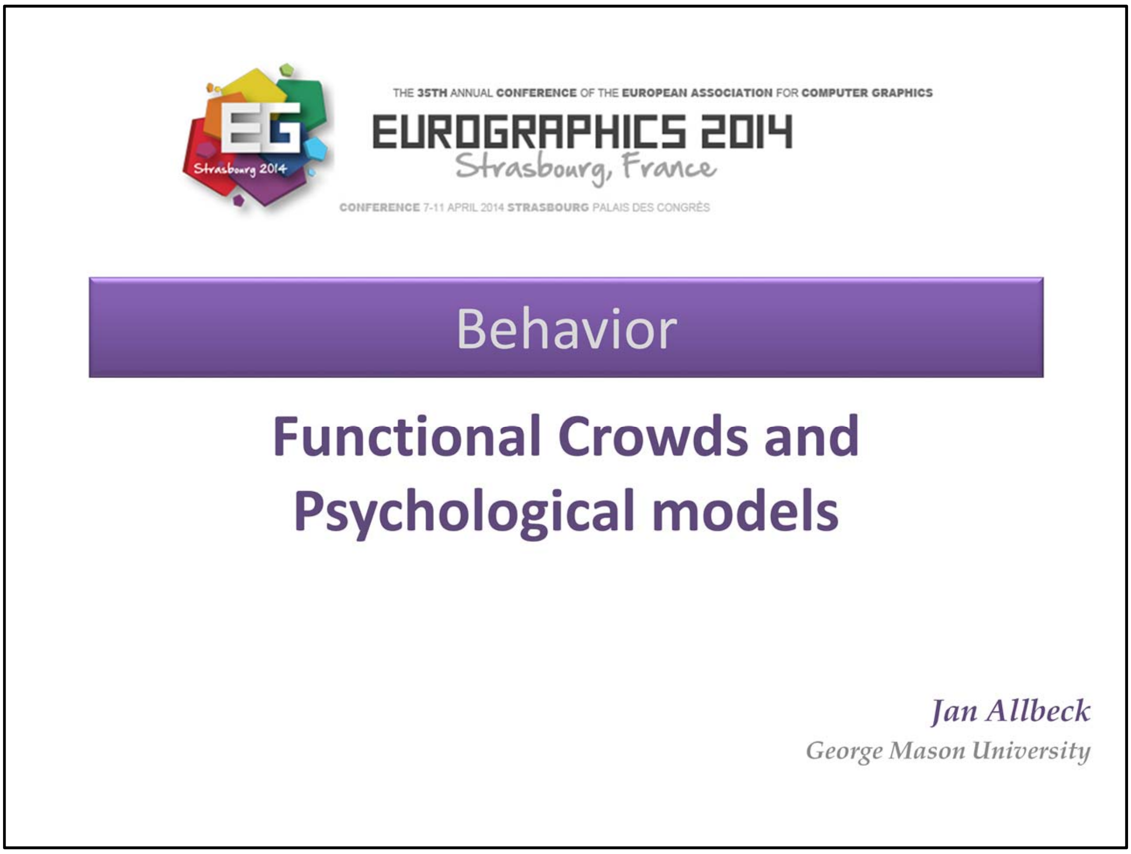
EUROGRAPHICS 2014
Strasbourg, France

CONFERENCE 7-11 APRIL 2014 STRASBOURG PALAIS DES CONGRÈS

Simulating Heterogeneous Crowds

BEHAVIOR

Jan Allbeck
George Mason University
Yiorgos Chrysanthou
University of Cyprus
Mubbasir Kapadia
Disney Research, Zurich



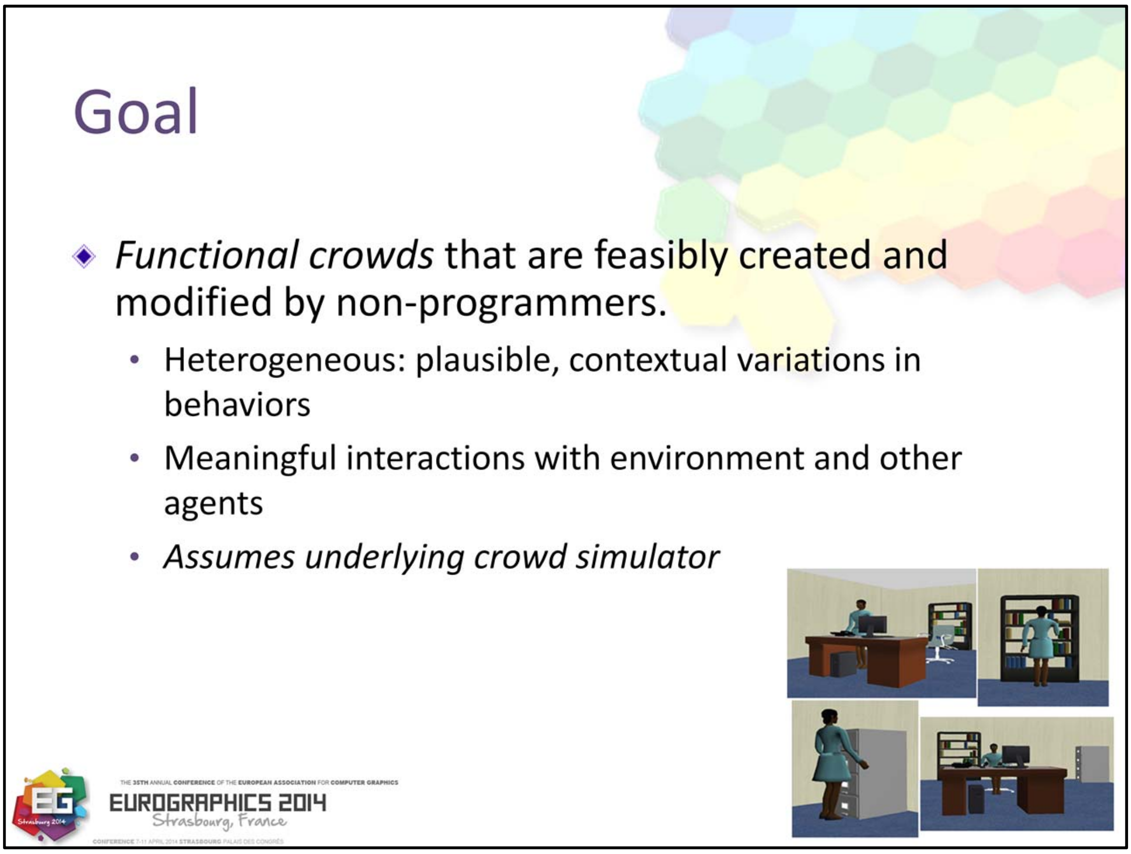
Real human populations are rarely homogeneous. People have different functions in their relationships and in society. As a result, they have different short, medium, and long term goals that are reflected in their behaviors. They also have psychological differences, including psychological needs and personality types. In this section, we explore computational formalisms for some of these factors that lead to more heterogeneous crowd simulations.

POPULATIONS WITH PURPOSE

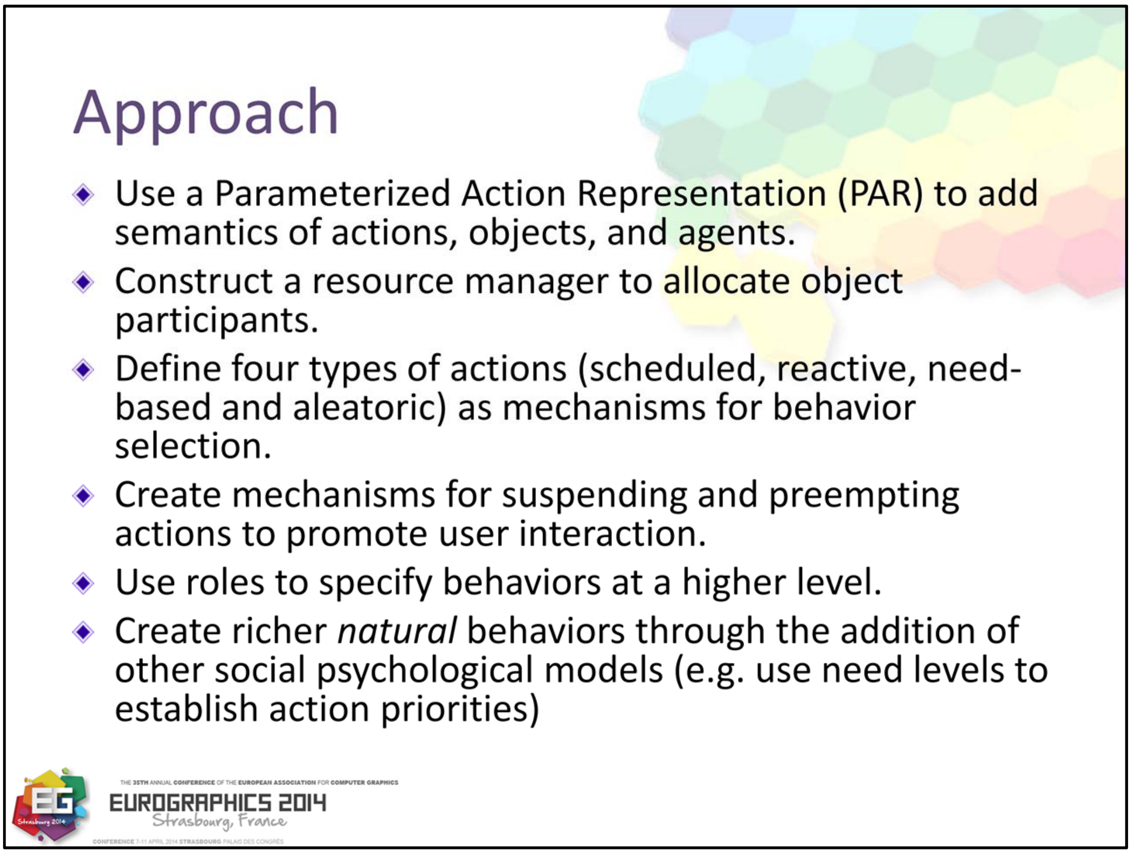
- Allbeck, J.M. CAROSA: A Tool for Authoring NPCs. In Proceedings of Motion in Games. Springer, 2010, pages 182-193.
- Allbeck, J.M. and Badler, N.I. Simulating Human Activities for Synthetic Inputs to Sensor Systems. In Distributed Video Sensor Networks. B. Bhanu, C.V. Ravishankar, A.K. Roy-Chowdhury, H. Aghajan, and D.Terzopoulos (Eds). Springer. 2011, pages 193-206.
- Li, W. and Allbeck, J.M. Populations with Purpose. In Proceedings of Motion in Games. Springer, pages 133-144, 2011.

Populations with Purpose refers to crowd simulations in which agents have a purpose or function in the simulated society.

Additional information including videos can be found here:
http://cs.gmu.edu/~gaia/CrowdSims/popPurpose.html



Creating crowd simulations in which agents perform actions beyond walking from one point to another requires additional data and mechanisms to ensure that the actions are performed in the proper contexts. Ideally scenarios designers and not programmers would be able to create and modify these behaviors. The result would be heterogeneous crowd simulations that depict functional populations in which agents have meaningful interactions with objects in the environment and each other.
This work assumes an underlying crowd simulation framework that enables agents to navigate through their environment avoiding collisions.



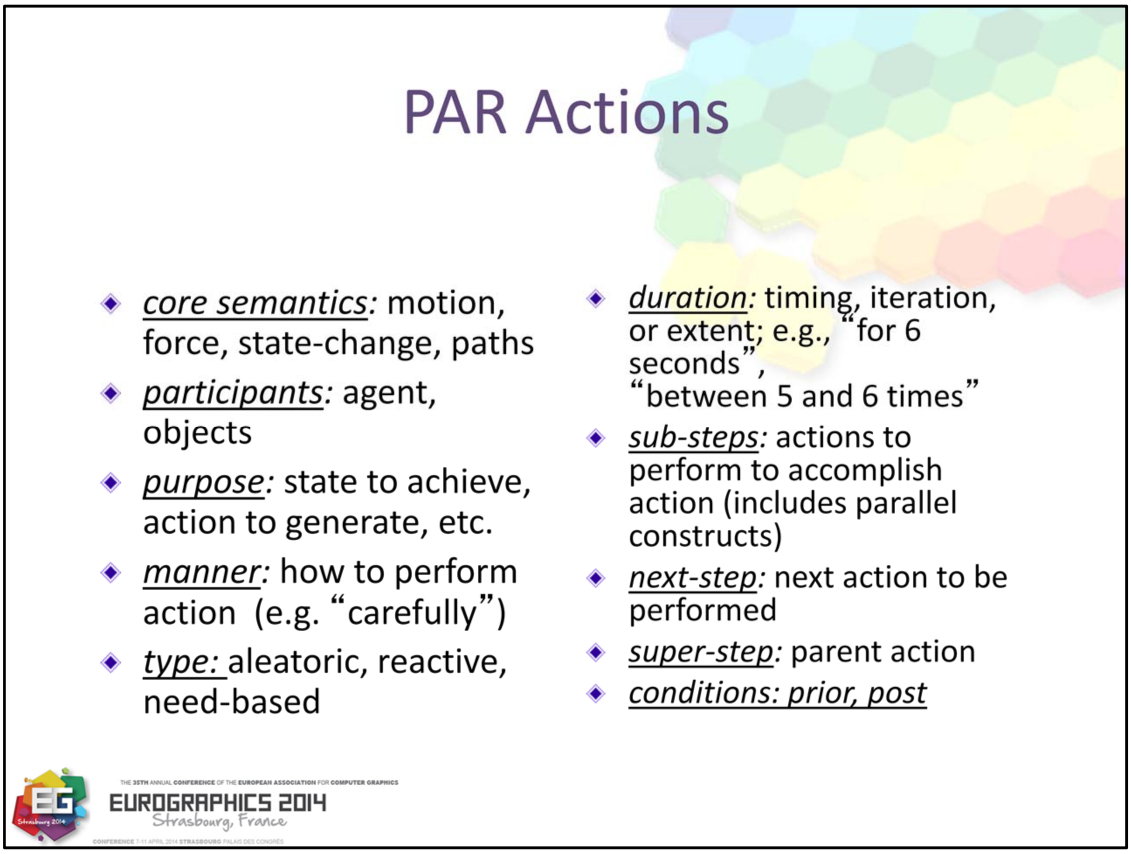
In order to facilitate agent-object interactions without overburdening scenario authors, we use parameterized representations of the semantics of actions and objects. A resource or object manager checks the environment for objects that are needed to successfully perform actions (e.g. *food* for *eat* and *chair* for sit).

At any point in time, a person's behavior can have many possible motivations. Some actions are pre-scheduled either explicitly or by routine. Other behaviors are reactions to environmental stimuli. Certainly, needs motivate actions. When simulating human behaviors, there are times when the specifics of individual behaviors are not important as long as the aggregate of the behaviors seem reasonable. For example, working in an office might be composed of several sub-actions such as typing on a computer, filing papers, answering a phone, etc. Combinations of these aleatoric or stochastic behaviors are distributed over the duration of the parent action.

Given the different possible motivations for an agent's actions (including user intervention), there will be occasions when action performances come into conflict. Mechanisms for suspending and preempting actions allow these conflicts to be resolved while still ensuring actions are performed in the proper contexts.

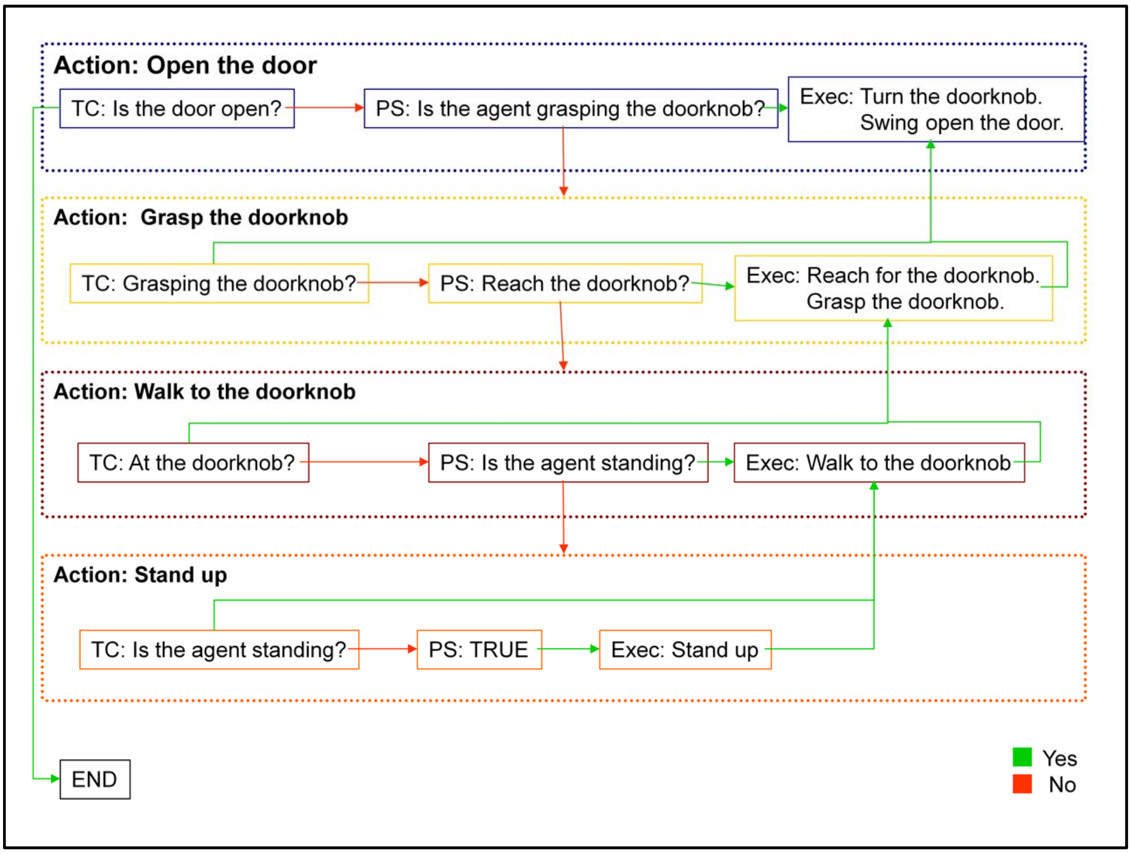
Roles provide an even higher level framework for grouping and describing agent behaviors. These descriptions allow entire populations to be specified more easily.

Social psychology models provide additional richness to the simulations.

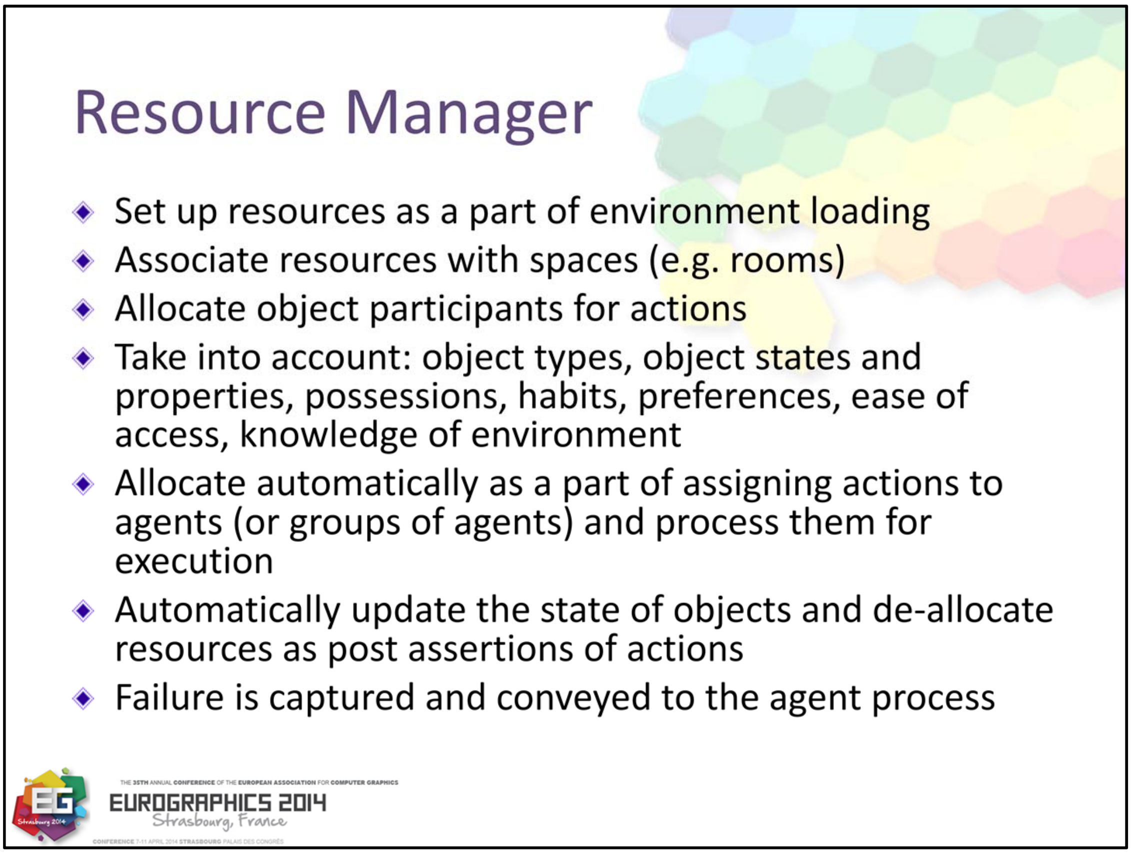


PAR actions and objects have numerous parameters. For more information on PAR and the PAR framework see: http://cs.gmu.edu/~gaia/PAR/

PARs can help capture some “commonsense” semantics. For example, linking actions with object resources that are required for their successful performance. PARs also ensure actions are performed in the proper context and update the world model…

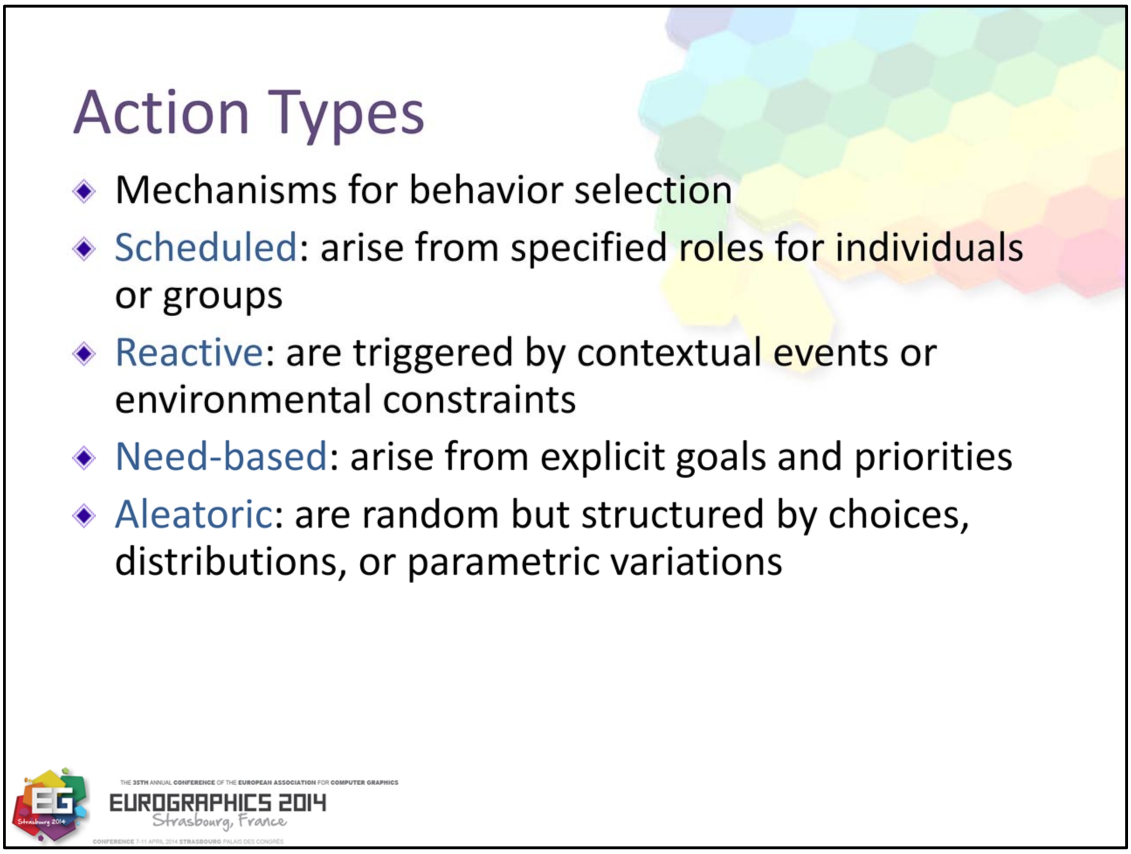


In this example, an agent was instructed to open a door. Prior to performing an "open door" animation, PAR checks various conditions to ensure the animation is played in the right context.
First, the termination condition (TC) for the action is tested. There is no reason to open the door if it is already open. If the door is not open, then the action's preparatory specifications (PS) are checked. These are a list of condition, action pairs. If the condition does not hold, then the paired action can be executed to establish the condition. For example, if the agent is not grasping the doorknob, then it is automatically instructed to grasp the doorknob. This is also a PAR action and is processed in the same way as the original action. This provides a built in backward chaining that only allows actions to be execute (Exec) when appropriate. Depending on the agent's initial state, this "open the door" instruction may automatically result in the agent standing up, walking to the door, reaching for and grasping the doorknob, turning the knob, and swinging the door open. Because the details are handled automatically, a scenario author can concentrate their efforts on higher level, more impactful components of the scenario. For example, agents are very rarely directly instructed to walk. The actions they are instructed to perform involve more meaningful interactions with objects in the world and each other. Walking to appropriate locations is done as an automatic prerequisite of those actions.
Note: this is not a full representation of the conditions and assertions in the execution of a PAR action. In particular, post assertions are executed when a PAR action has been successfully performed and update the world model.

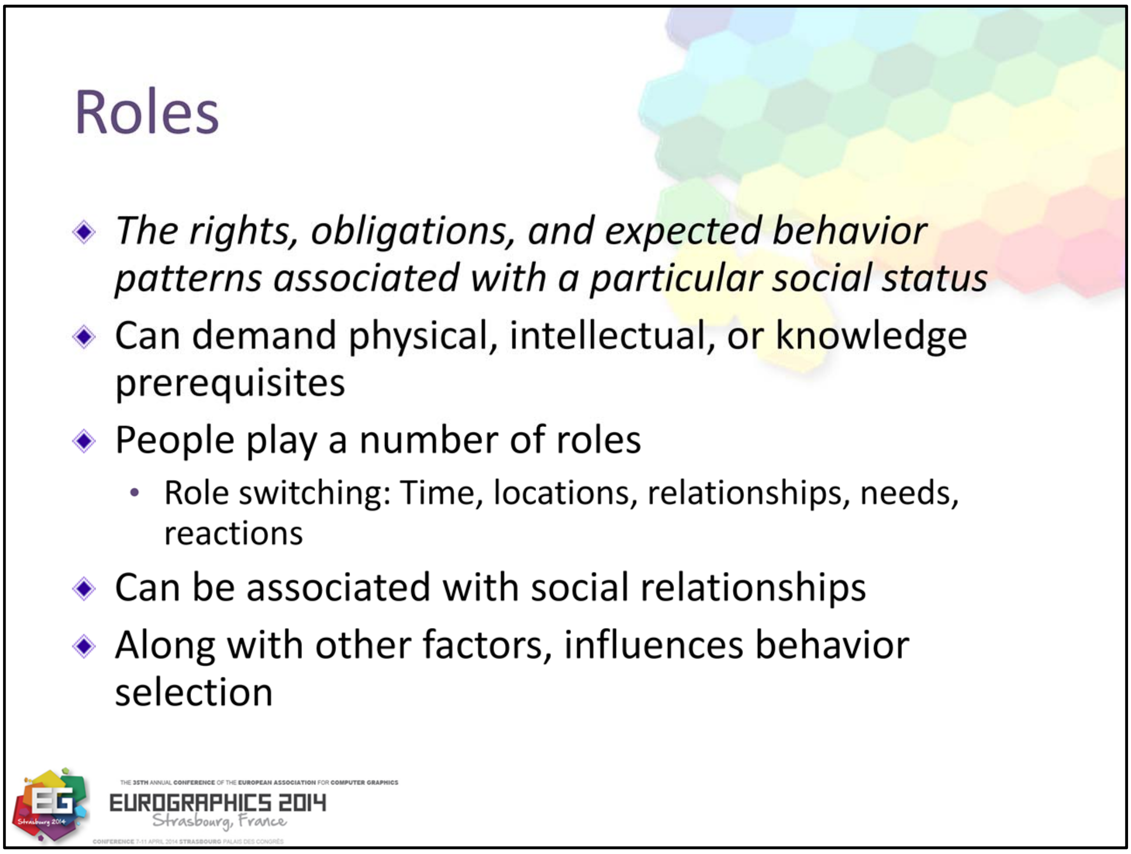


Because one of the goals is for agents to have meaningful interactions with objects in the world, we need mechanisms for finding appropriate objects as participant parameters in PAR actions. This is the responsibility of the resource or object manager. As graphical models in the environment are loaded, the resource manager is automatically informed of the objects and their locations. The resource manager also has access to the current state of the objects during the simulation. When an agent is instructed to perform an action that requires an object participant, the resource manager allocates an appropriate object (if one is available). An agent may be instructed to perform the action with a specific object or with an object in a certain location or it may be left unspecified. The resource manager attempts to allocate an object that matches the constraints provided. If there is no object resource available that fits the specified criteria, then the failure is reported to the agent process, where an alternative action can be selected.

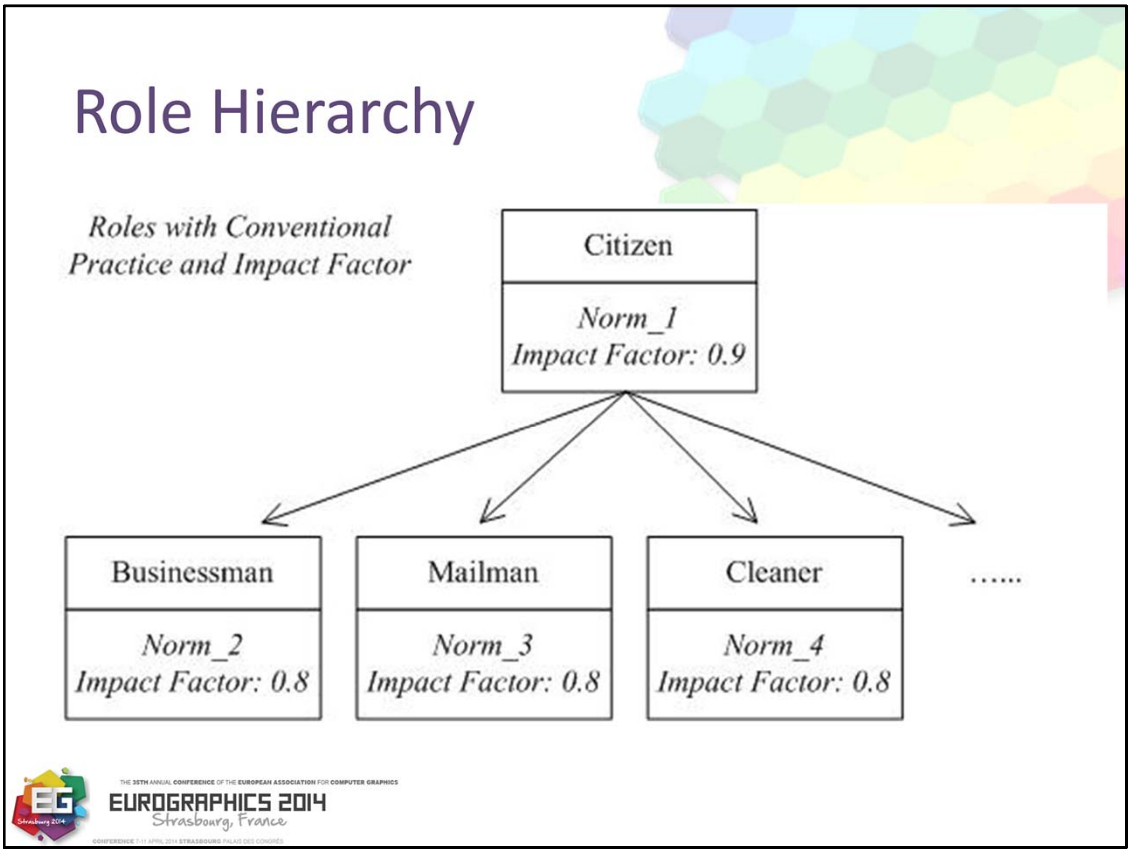
Some other factors that could impact resource allocation:
What objects can the agent see?
What objects can the agent remember?
What objects can the agent assume? (commonsense)
What if perception, memory, or assumptions are wrong?



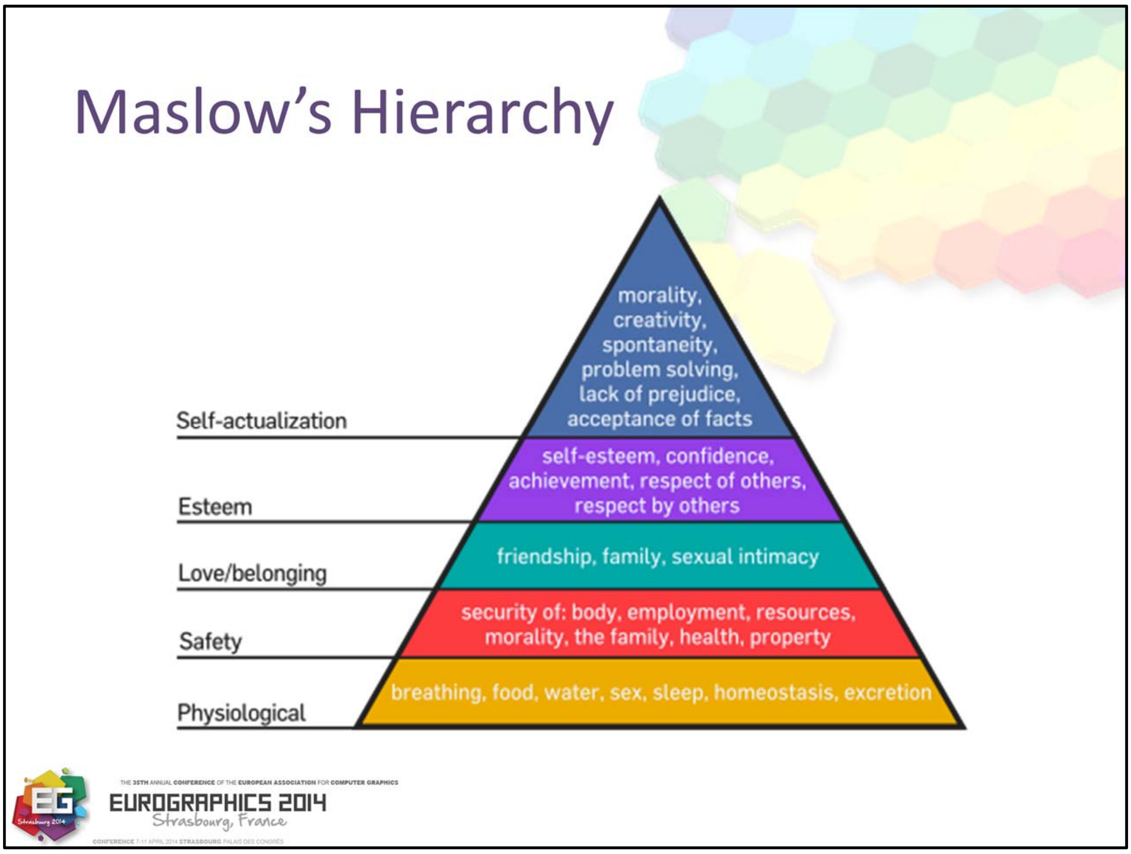
The implemented action types stem from possible motivations for people's behaviors. They also influenced by a desire for easy, straightforward scenario authoring. Scheduled actions are simply PAR actions with start times, durations, and locations. Reactive actions are stimuli (objects, agents, states, etc) paired with actions. Need-based actions have a name, a decay rate, and a list of action, fulfillment level pairs (with an optional object type). A need gets stronger over time according to the decay rate and is met by performing a listed action. An agent can have many needs and will automatically perform an action to fulfill any need that becomes stronger than their desire to perform their current action. Aleatoric actions are specified by a set of sub-actions and their probabilities. When an agent is instructed to perform an aleatoric action, a series of its sub-actions are chosen according to the specified distribution.



Roles are used to group different action types into more comprehensive behaviors. Agents can have multiple roles corresponding to relationships and occupations, for example. Determining when it is appropriate for an agent to switch from one role to another is quite challenging and can be influenced by a number of factors including time, location, relationships, needs, and other stimuli.



Roles can be formed into a hierarchy. Norms for a citizen of a society would apply to everyone regardless of their occupation, for example, but there may be other norms more specific to an occupation or relationship. Citizens of a society are expect to follow the laws of the society. A mailman is expected to wear a uniform and have a bag filled with letters and packages as he moves from house to house.



Social psychological models can be used to help prioritize actions and create more *natural* crowd simulations. Maslow's hierarchy of needs prioritizes physiological needs (at the bottom of the pyramid) over all other needs. It fits in well with the need-based actions and can be linked to personality traits…

Personality Traits

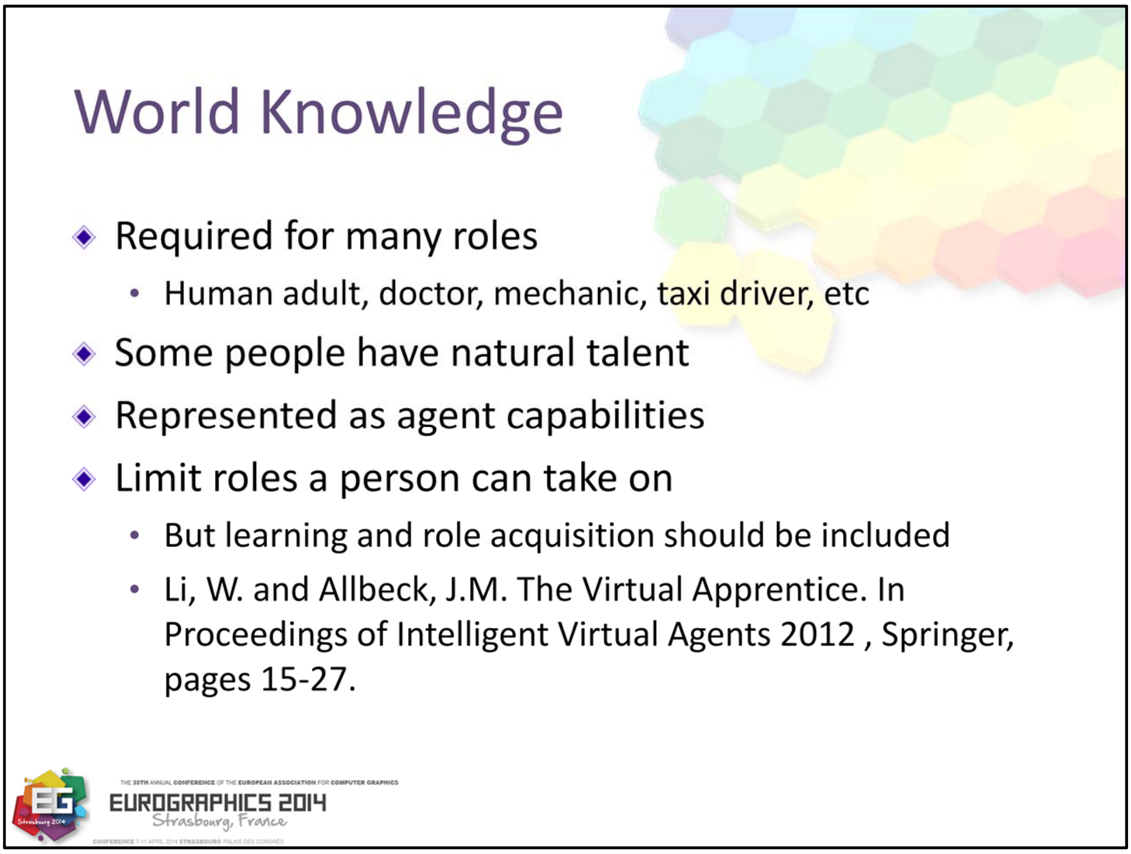
O+	Curious, alert, informed, perceptive
O-	Simple, narrow, ignorant
C+	Persistent, orderly, predictable, dependable, prompt
C-	Messy, careless, rude, changeable
E+	Social, adventurous, active, assertive, dominant, energetic
E-	Distant, unsocial, lethargic, vigorless, shy
A+	Cooperative, tolerant, patient, kind
A-	Bossy, negative, contrary, stubborn, harsh
N+	Oversensitive, fearful, unadventurous, dependent, submissive, unconfident
N-	Calm, independent, confident

The OCEAN personality model is very popular in virtual humans research and will be discussed further in the next section. Some of these traits can be plausibly linked to increased or decreased needs…

Need Examples

Reservoirs	Personality Traits
Problem solving, creativity, lack of prejudice	Open, Agreeable
Achievement, respect for others	Open, Agreeable
Friendship, family	Extraverted
Security of employment, security of family	Conscientious, neurotic
Water, food, excretion, etc	All

Linking need reservoirs with personality traits promotes additional heterogeneity in populations. For example, an agent with a more conscientious, neurotic personality would have an increased need for security of employment and family. Their reservoirs for these needs would deplete faster than other agents, leading them to spend more time performing actions to fulfill them (e.g. actions related to their occupation role).



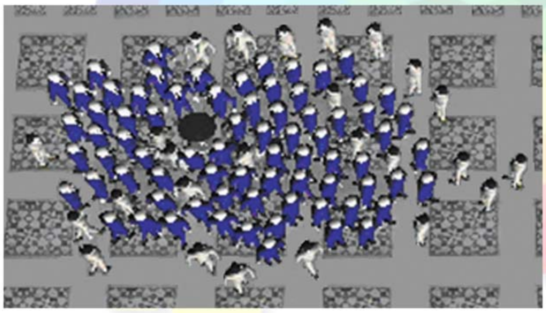
The capabilities of an agent can be enumerated and considered before allowing them to take on a role, but agents can also be taught roles. In fact, they can form their own unique definitions of roles based on their own experiences and observations:

http://cs.gmu.edu/~gaia/IntellAgents/virtualApprentice.html

The demonstration video can be found here:
http://cs.gmu.edu/~gaia/CrowdSims/popPurpose.html

Summary

- *Functional crowds* are heterogeneous populations in which agents have meaningful interactions with their environment.
- Representing the semantics of actions and objects facilitates authoring (control vs. autonomy).
- Simple behavior selection mechanisms can lead to interesting, complex scenes.
- Defining roles, along with personality types and needs, can promote more consistent, compelling long term behaviors

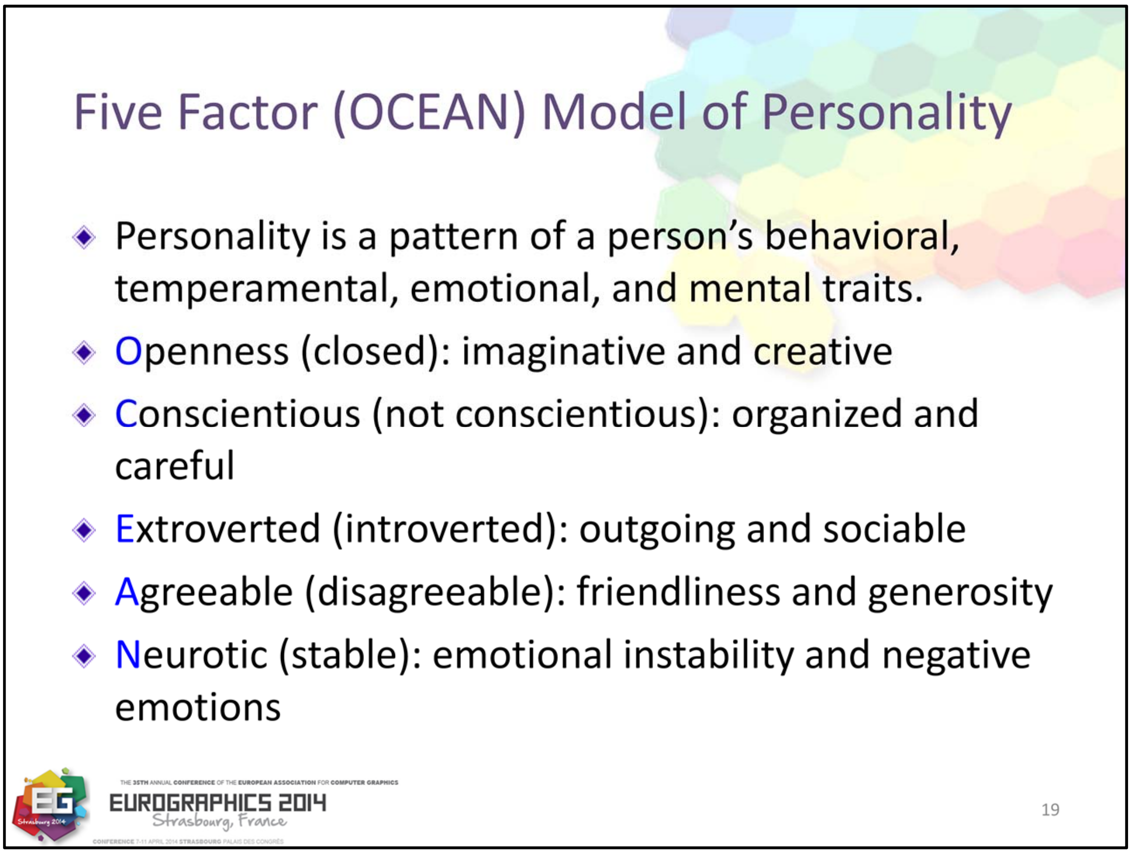


HOW THE OCEAN PERSONALITY MODEL AFFECTS THE PERCEPTION OF CROWDS

- Durupinar, J. Allbeck, N. Pelechano, and N. Badler. Creating Crowd Variation with the OCEAN Personality Model. Proceedings of Autonomous Agents and Multi-Agents Systems 2008. pp 1217-1220.
- Durupinar, F., Pelechano, N., Allbeck, J., Gudukbay, U., and Badler, N. The Impact of the OCEAN Personality Model on the Perception of Crowds. IEEE Computer Graphics and Applications. vol. 31, no. 3, pp. 22-31, May/June, 2011.

Well established psychological models can be used to enhance crowd simulations and create more heterogeneity. In this project, we mapped personality traits to parameters of a crowd simulator to obtain more varied behaviors.

The videos associated with this research can be found here:
http://www.cs.bilkent.edu.tr/~fundad/RESEARCH/ocean.htm



The five factor or OCEAN personality model is well established in psychology, but there are alternative models with varying numbers of factors that could be used. Each of the five factors can have a positive or negative valence. For example, a person can be open minded or close minded or in fact neutral.

Representation

- An agent personality: $\pi = < \Psi_{O,} \, \Psi_{C,} \, \Psi_{E,} \, \Psi_{A,} \, \Psi_{N} >$
- Distribution: $\Psi_i = N\left(\mu_i, {\sigma_i}^2\right), for\ i$ $\in \{O, C, E, A, N\}$ where $\mu \in [0, 1], \sigma \in [-0.1, 0.1]$
- Behavior:
  - $\beta = (\beta_1, \beta_2, \ldots, \beta_n)$
  - $\beta_j = f(\pi), for\ j = 1, \ldots, n$

An agent's personality is represented as a distribution of the five factors. An agent's behavior is then a function of its personality.

Parameter Mapping

Agent behavior	Personality factor	OCEAN
Leadership	Assertive, social, unsocial, calm, fearful	E, N
Trained or untrained	Informed, ignorant	O
Communication	Social, unsocial	E
Panic	Oversensitive, fearful, calm, orderly, predictable	N, C+
Impatience	Rude, assertive, patient, stubborn, tolerant, orderly	E+, C, A
Pushing	Rude, kind, harsh, assertive, shy	A, E
Right preference	Cooperative, predictable, negative, contrary, changeable	A, C
Avoidance or personal space	Social, distant	E
Waiting radius	Tolerant, patient, negative	A
Waiting timer	Kind, patient, negative	A
Exploring environment	Curious, narrow	O
Walking speed	Energetic, lethargic, vigorless	E
Gesturing	Social, unsocial, shy, energetic, lethargic	E

Our underlying crowd simulator had a number of behavior parameters that could be adjusted. These behaviors were mapped to adjectives that could be used to describe the possible resulting behaviors. These adjectives were then also associated with personality factors.
A positive factor takes values in the range [0.5, 1]; a negative factor takes values in the range [0, 0.5). A factor with no sign indicates that both poles apply to that behavior. For instance, E+ for a behavior means that only extroversion is related to that behavior; introversion isn't applicable.

Panic

- Linked to stability and conscientiousness

$$\beta_i^{\text{Panic}} = \omega_{\text{NP}} \Psi_i^{\text{N}} + \omega_{\text{CP}} f(\Psi_i^{\text{C}})$$

$$f(\Psi_i^{\text{C}}) = \begin{cases} -2\Psi_i^{\text{C}} + 2 & \text{if } \Psi_i^{\text{C}} \geq 0, \\ 0 \text{ otherwise} \end{cases}$$

where $\beta_i^{\text{Panic}} \propto \text{N}$, $\beta_i^{\text{Panic}} \propto^{-1} \text{C+}$, and $\beta_i^{\text{Panic}} \in [0, 1]$

THE 35TH ANNUAL CONFERENCE OF THE EUROPEAN ASSOCIATION FOR COMPUTER GRAPHICS
EUROGRAPHICS 2014
Strasbourg, France
CONFERENCE 7-11 APRIL 2014 STRASBOURG PALAIS DES CONGRÈS

Personality factors are incorporated into equations that impact behaviors. Here we see that a panic behavior is linked to stability and conscientiousness.
In emergency situations, agents display panic behavior depending on their stability and conscientiousness. When they panic, their walking speed increases and they don't wait.

Pushing

- Disagreeable and extroverted agents tend to push others

$$P_i(\text{Pushing}) = \omega_{\text{EP}}\Psi_i^{\text{E}} + \omega_{\text{AP}}(1 - \Psi_i^{\text{A}})$$

$$\beta_i^{\text{Pushing}} = \begin{cases} 1 \text{ if } P_i(\text{Pushing}) \geq 0.5 \\ 0 \text{ otherwise} \end{cases}$$

where $P_i(\text{Pushing}) \propto \text{E}$, $P_i(\text{Pushing}) \propto^{-1} \text{A}$, and $\beta_i^{\text{Pushing}} \in \{0,1\}$

A crowd simulator can realistically simulate a person's respect for others. Agents can try to force their way through a crowd by pushing others, exhibit more respectful behavior when desired, make decisions about letting others walk first, and queue when necessary. Disagreeable agents tend to push others more because they're harsh and impolite. Similarly, extroverted agents display pushing behavior because they tend to be assertive.

Waiting Radius

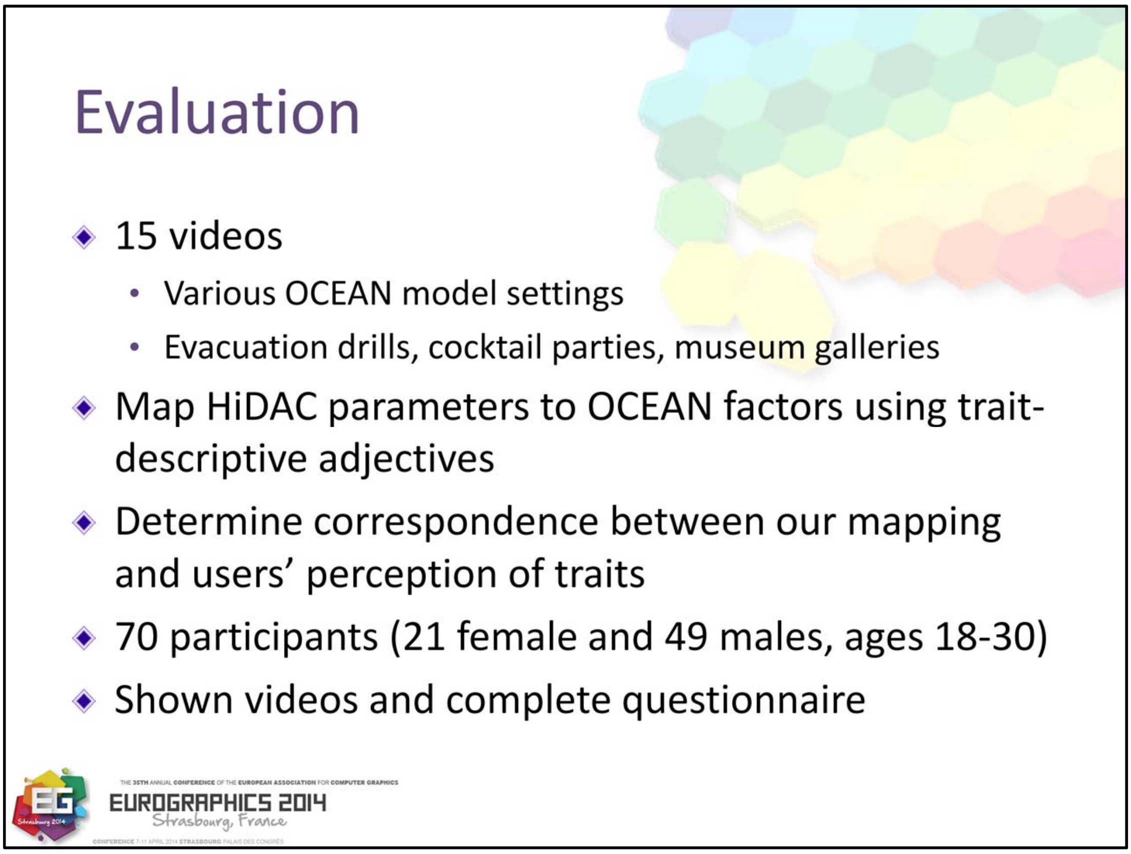
- Influenced by kindness and consideration
- Linked to agreeableness

$$\beta_i^{\mathrm{WaitingRadius}} = \begin{cases} 0.25 & \text{if } \Psi_i{}^{\mathrm{A}} \in \left[\frac{0}{3}\,\frac{1}{3}\right) \\ 0.45 & \text{if } \Psi_i{}^{\mathrm{A}} \in \left[\frac{1}{3}\,\frac{2}{3}\right] \\ 0.65 & \text{if } \Psi_i{}^{\mathrm{A}} \in \left(\frac{2}{3}\,\frac{3}{3}\right] \end{cases}$$

where $\beta_i^{\mathrm{WaitingRadius}} \propto \mathrm{A}$ and
$\beta_i^{\mathrm{WaitingRadius}} \in \{0.25, 0.45, 0.65\}$

THE 35TH ANNUAL CONFERENCE OF THE EUROPEAN ASSOCIATION FOR COMPUTER GRAPHICS
EUROGRAPHICS 2014
Strasbourg, France
CONFERENCE 7-11 APRIL 2014 STRASBOURG PALAIS DES CONGRÈS

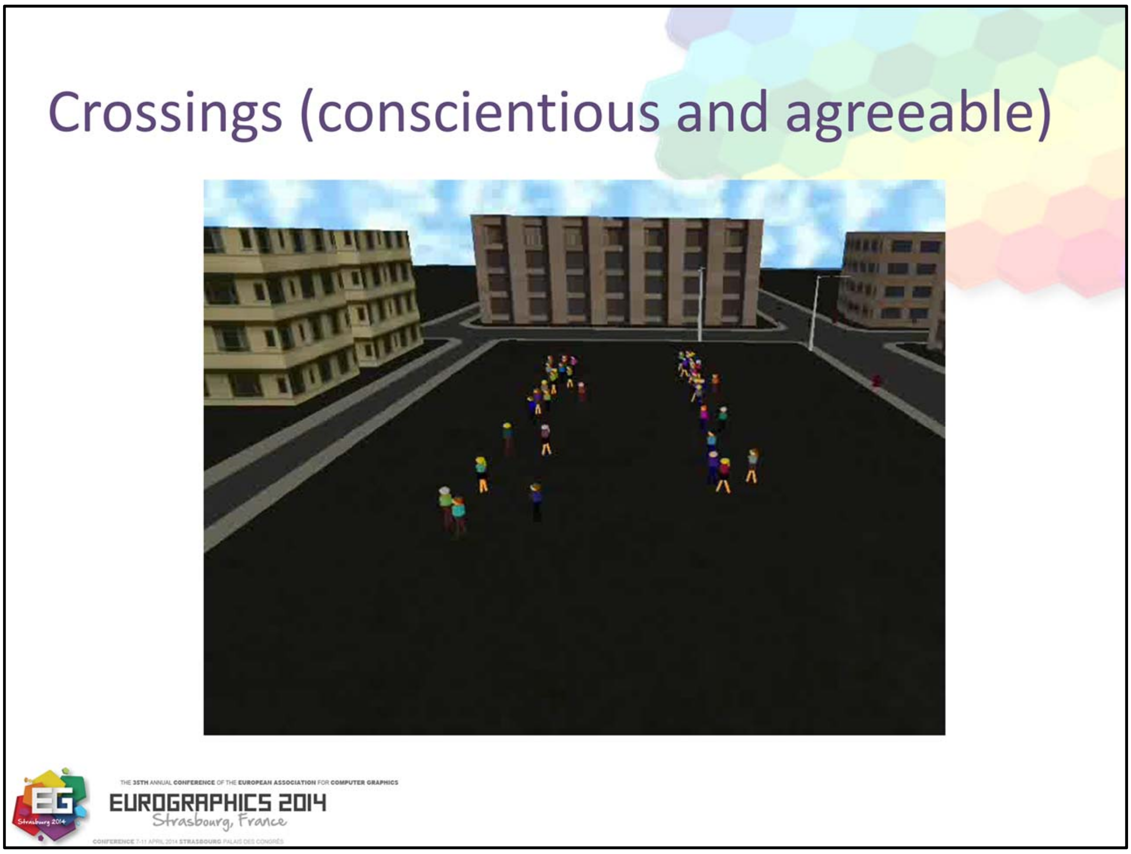
In an organized situation, people tend to wait for available space before moving. We call this space the *waiting radius*; it depends on a person's kindness and consideration—that is, the agreeableness dimension.



To evaluate whether users will correctly perceive our suggested mappings, we conducted user studies. We created several animations to study how modifying subgroups' personality parameters affects global crowd behavior.

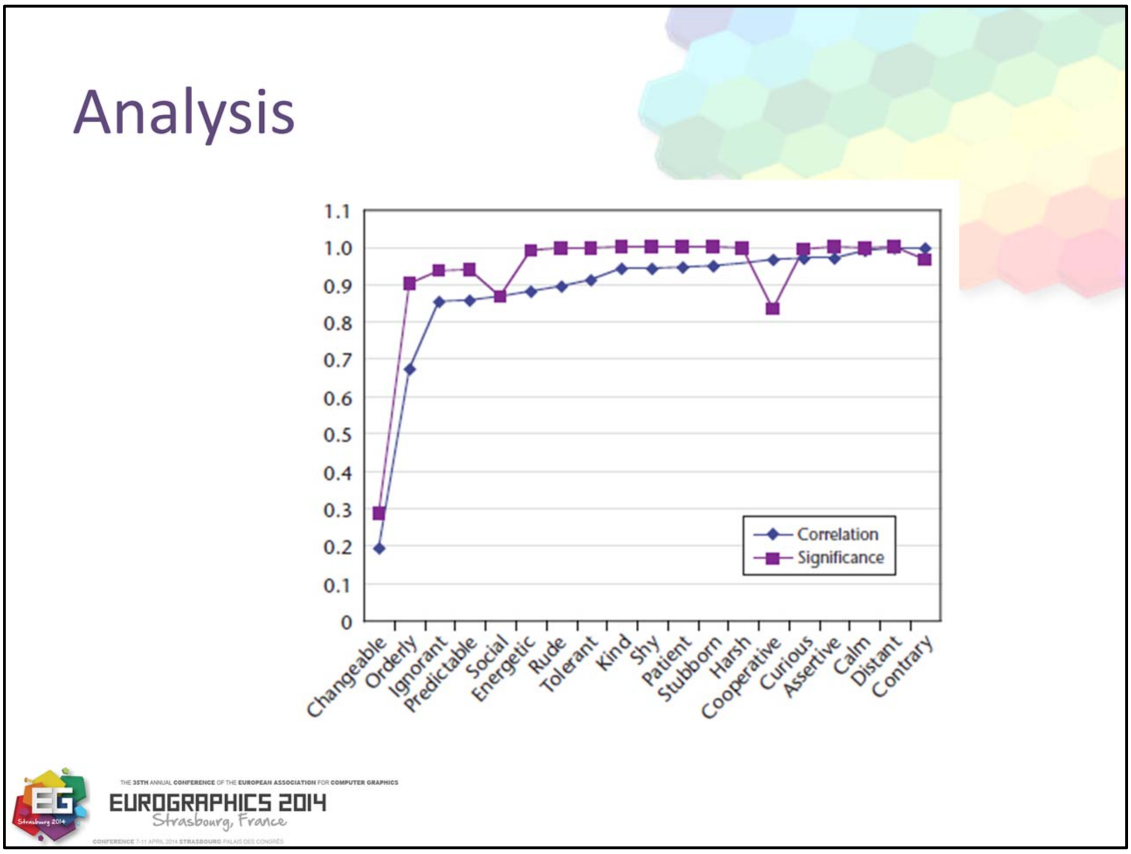
This scenario tested the adjectives *curiosity* and *ignorance*. There were three groups of people, with openness values of 0, 0.5, and 1. We mapped the number of tasks that each agent must perform to openness, with each task requiring looking at a painting. The least open agents (with blue hair) left the museum first, followed by the agents with openness values of 0.5 (with black hair). The most open agents (with red hair) stayed the longest. We asked the participants how they perceived each group.

The videos associated with this research can be found here:
http://www.cs.bilkent.edu.tr/~fundad/RESEARCH/ocean.htm



This video showed people with high agreeableness and conscientiousness values (μ = 0.9 and σ = 0.1 for both traits). Agreeable and conscientious agents always prefer to move towards the right hand side of other agents, whereas others may prefer left or right with equal chance. Agreeable and conscientious crowds have less congestion.

The videos associated with this research can be found here:
http://www.cs.bilkent.edu.tr/~fundad/RESEARCH/ocean.htm



We classified each adjective by its question number, simulation parameter, and subjects' answer.
The correlation coefficients between the parameters and the subjects' answers for the descriptive adjectives (blue), and the significance values for the corresponding correlation coefficients (violet). Significance is low (<0.95) for changeable, orderly, ignorant, predictable, social, and cooperative.

For *changeable* and *orderly*, this is because of low correlation values. For *predictable*, *ignorant*, *social*, and *cooperative*, the correlation coefficients are high, but their significance is low because of the small sample size.

To understand why, consider the setting in which two groups of agents crossed each other. The participants identified the nonconscientious agents as rude but perceived them as persistent in their rudeness. This perception caused the participants to mark lower values for the question about changeability.

Summary

- High correlation between our parameters and the participants' perception of them.
- Low correlation for some adjectives is due to the terms' ambiguity.
- Frees scenario authors from tuning low level parameters
- Decreases number of parameters from 13 to 5
- Focus on agents' character instead of behavioral parameters
- Give observers a sense of knowing the agents

EUROGRAPHICS 2014 Strasbourg, France

Acknowledgements

Collaborators

- Norman I. Badler
- Funda Durupinar
- Uğur Güdükbay
- Weizi Li
- Nuria Pelechano

Sponsors

- Scientific and Technological Research Council of Turkey International PhD Research Fellowship Programme 2214 and projects EEE-AG 104E029 and 105E065
- Spanish Government grant TIN2010-20590-C01-01
- U.S. Army SUBTLE MURI W911NF-07-1-0216

EUROGRAPHICS 2014 Strasbourg, France

THE 35TH ANNUAL CONFERENCE OF THE EUROPEAN ASSOCIATION FOR COMPUTER GRAPHICS

EUROGRAPHICS 2014

Strasbourg, France

CONFERENCE 7-11 APRIL 2014 STRASBOURG PALAIS DES CONGRÈS

Simulating Heterogeneous Crowds

Data Driven Crowd Simulation

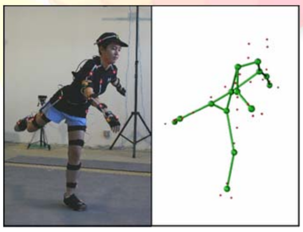
Yiorgos Chrysanthou

(University of Cyprus)

Data Driven Crowds

32

- Data driven algorithms have been used in Computer Animation
  - Motion capture single characters
  - Facial motion capture
  - Even trees…



- In crowds the emphasis is on multi-character interactions
  - Capture subtle actions

EUROGRAPHICS 2014

Strasbourg, France

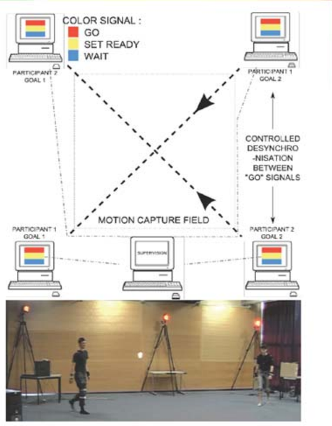
Data Driven Crowds

- Two general categories of approaches
- Analysis of data to extract parameters for a model
  - Social forces
  - Rule based
  - Flocking
- Example based techniques
  - Synthesize motion directly from the input data

PARAMETER EXTRACTION

Collision Avoidance

- Paris et. al [2007], and Pettre at al [2009] proposed a prediction based approach to crowd steering from motion capture data
- Controlled environment
  - Goals are known
  - Easier data acquisition
  - Accuracy
- Data is used to estimate time of collision between entities
- Biomechanics group



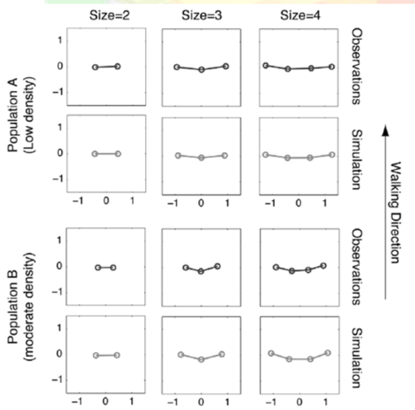
From: Sébastien Paris, Julien Pettré, and Stéphane Donikian. Pedestrian Reactive Navigation for Crowd Simulation: a Predictive Approach. Computer Graphics Forum (Proceedings of Eurographics),26(3): 665-674, September 2007.

Helbing’s Social Forces Model

- Moussaid et. Al [2009, 2010] used data from videos of real crowds to modify Helbing’s Social Force Model
  - New Forces tries to keep social groups together using:
    - Vision of agent
    - Attraction force
    - Repulsion force

$$\frac{d\vec{v}_i}{dt} = \vec{f}_i^{\,0} + \vec{f}_i^{\,wall} + \sum_j \vec{f}_{ij} + \vec{f}_i^{\,group}$$

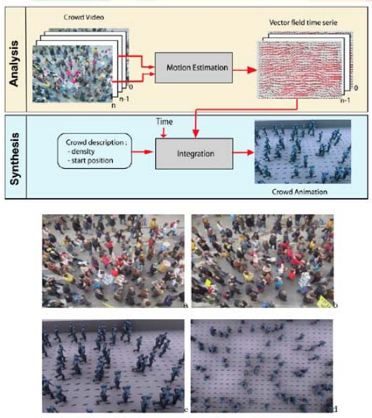
$$\vec{f}_i^{\,group} = \vec{f}_i^{\,vis} + \vec{f}_i^{\,att} + \vec{f}_i^{\,rep}$$



From: Moussaïd M, Perozo N, Garnier S, **Helbing** D, Theraulaz G, The Walking Behaviour of Pedestrian Social Groups and Its Impact on Crowd Dynamics. PLoS ONE 5(4), 2010

Flow Based

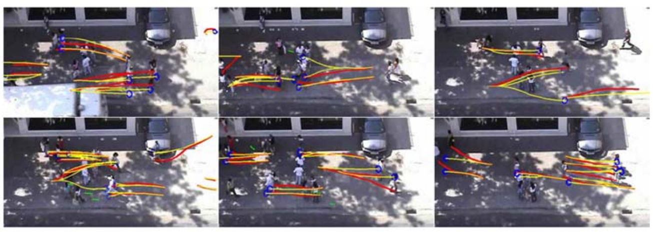
- Macroscopic model
- Assumption: the motions of individuals within the crowd is the expression of a continuous **flow** that drives the crowd motion.
- Dense crowds
- Aims to characterize the crowd behavior as a whole



From: Courty, N. & Corpetti, T.
Crowd motion capture
Comput. Animat. Virtual Worlds, John Wiley and Sons Ltd., **2007**, *18*, 361-370

Computer Vision

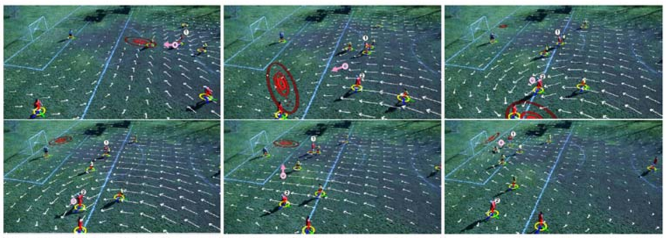
- Pellegrini et. Al [2010] use information from video + a stochastic crowd model to predict future individual motion



From: Pellegrini, S.; Ess, A.; Tanaskovic, M. & Gool, L. V.
Wrong Turn - No Dead End: a Stochastic Pedestrian Motion Model
International Workshop on Socially Intelligent Surveillance and Monitoring (SISM), **2010**

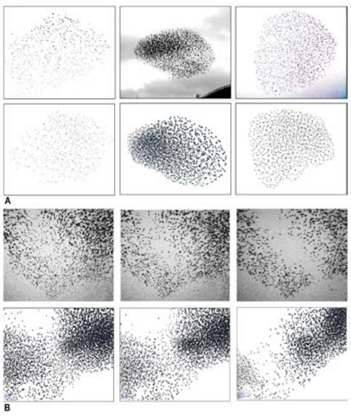
Computer Vision

- Essa et. Al [2010]
- Multi-camera tracking of players in a game
  - Positions
  - Movements
- Generate motion field from all the players in a sports scene to predict future points of interest (i.e. where will the game evolve next)



From: Kim, K.; Grundmann, M.; Shamir, A.; Matthews, I.; Hodgins, J. & Essa, I.
Motion Field to Predict Play Evolution in Dynamic Sport Scenes
IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, **2010**

Flock-like model (Biology)



From: H. Hildenbrandta, C. Carereb,c., C-K. Hemelrijka. Self-organised complex aerial displays of thousands of starlings: a model. arXiv.org, August 2009.

- Hemelrijka et. al [2009], proposed using input from stereoscopic videos of Starling birds to estimate a **statistical** model of their massive and complex flocking behavior:
  - up to 2 million birds in a flock reported!

Starling Flocks (From: Google Images)

User Defined Input

- Oshita and Ogiwara[2009] proposed a sketch based interface for crowd animation
- Users draw example paths → simulation parameters are extracted
  - Parameters: guiding paths, speed, distances, crowd regularity, etc.

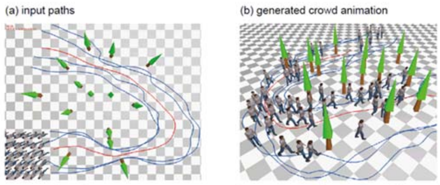
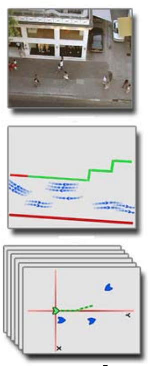
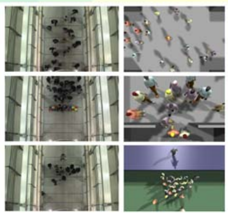


Fig. 1. Overview of our system. The parameters for crowd control are estimated from a few example paths that are given by a user (a). Based on the parameters and an agent model, a crowd animation is generated (b). The user can make a crowd animation interactively.

From: Oshita, Ogiwara, "Sketch-based Interface for Crowd Animation", *9th International Symposium on Smart Graphics 2009*, Lecture Note in Computer Science, Springer, pp. 253-262, Salamanca, Spain, May 2009



Lerner et al., *Crowds by Example*, **Eurographics 2007**



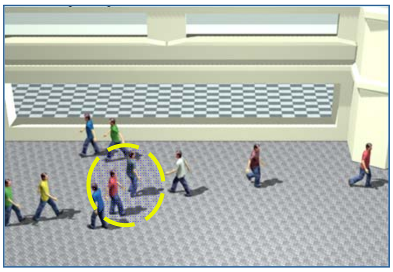
Lee et al., *Group behavior from video*, **In SCA '07**

EXAMPLE BASED SIMULATION

Crowds by Example

Lerner, Chrysanthou and Lischinsci, *Crowds by Example*, **Eurographics 2007**

- If an agent in the simulation can find a person in the video facing a similar situation (state), then it can copy part of its trajectory (action), thereby mimicking

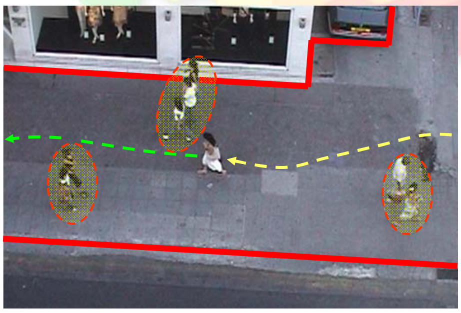


The agent in red and the woman in white both have someone on their left and someone coming towards them

Crowds by Example

Influencing Factors

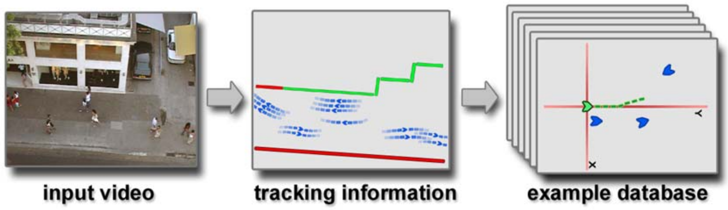
- An *influencing factor* is something that effects a person’s behavior
- This case: people and obstacles
- Not all factors have the same amount of influence on the persons behavior



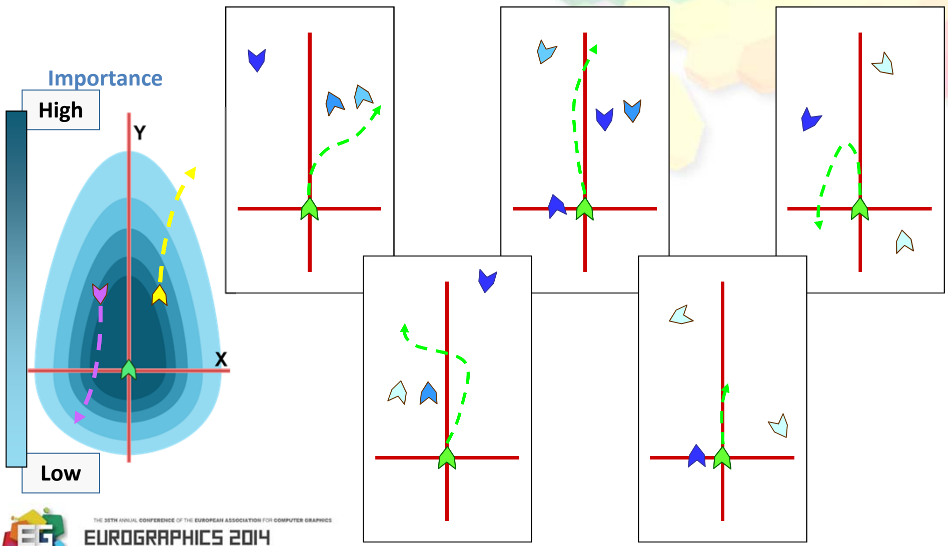
Potentially Influencing Factors:
Personality, Emotions ...
Terrain, Obstacles ...
Surrounding people ...

Crowds by Example
Preprocessing

- During preprocessing:
  - Somehow *track* the videos
  - Define *examples* using the surroundings of each person
  - Build the *database*



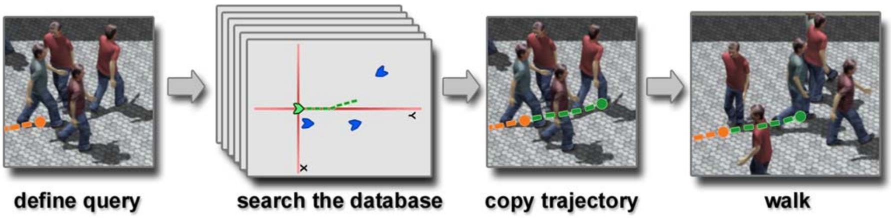
Crowds by Example
The examples



Crowds by Example

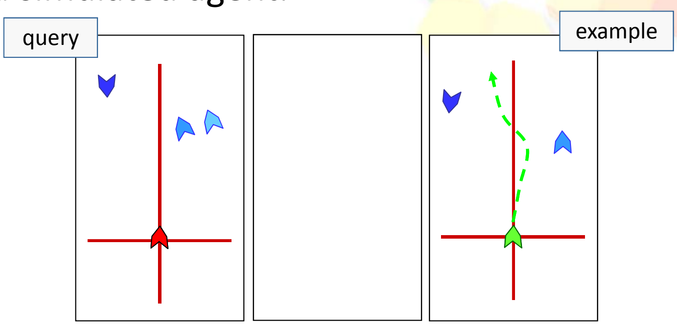
Run-time simulation

- During simulation of an agent:
  - Define a *query*
  - *Search* the database for a matching example
  - *Copy* the trajectory of the matching example
  - *Follow* the trajectory until a new one is needed



Queries

- A *query* is the configuration of influencing factors for a simulated agent.



- To see if an example matches the query, we *align* them and use a *continuous function* to determine their similarity.

Results

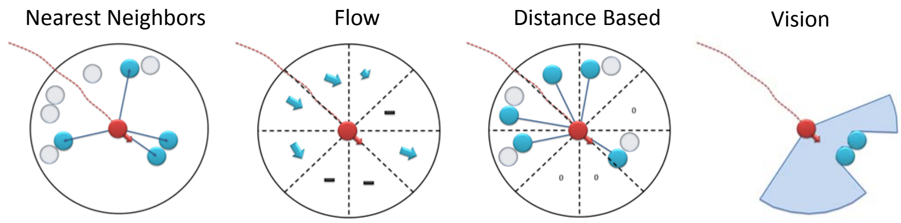
The PAG Crowd: A Graph Based Approach for Efficient Data-Driven Crowd Simulation

- The crowds by example method had several issues
  - Speed
  - High level control
  - Occasional errors
- The current method comes to address these

Charalambous,Chrysanthou, *The PAG Crowd*, **accepted at CGF 2014**

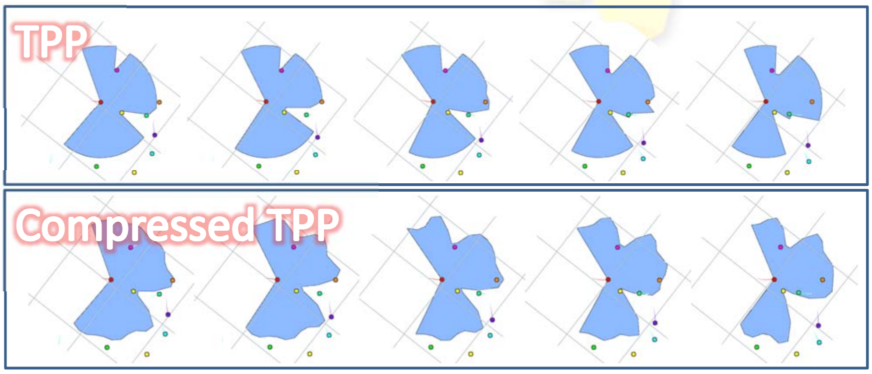
State Representation

- Agent State is encoded using **T**emporal **P**erception **P**atterns (TPP):
  - A **temporal** representation of agent centric **perceptual** information
  - Perception can vary: **vision**, flow, distances, …

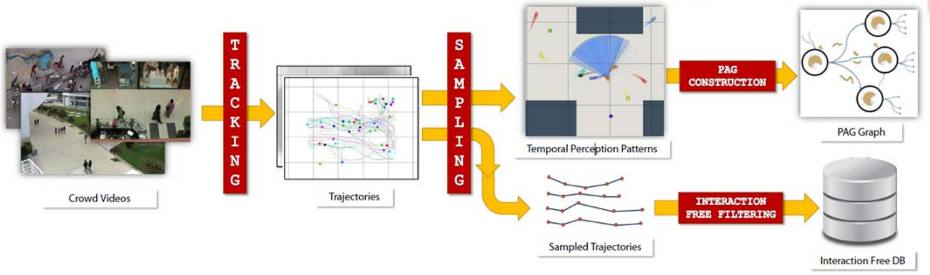


Temporal **P**erception **P**attern (TPP)

- TPP: A window of perceptual samples
- We compress them using a small number of Discrete Cosine Transform (DCT) coefficients



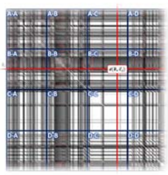
Preprocessing

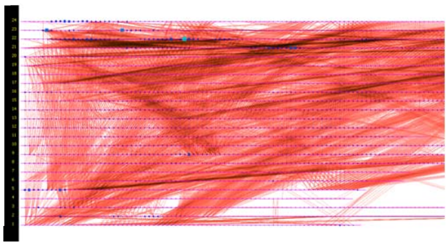


EUROGRAPHICS 2014 Strasbourg, France

PAG Generation

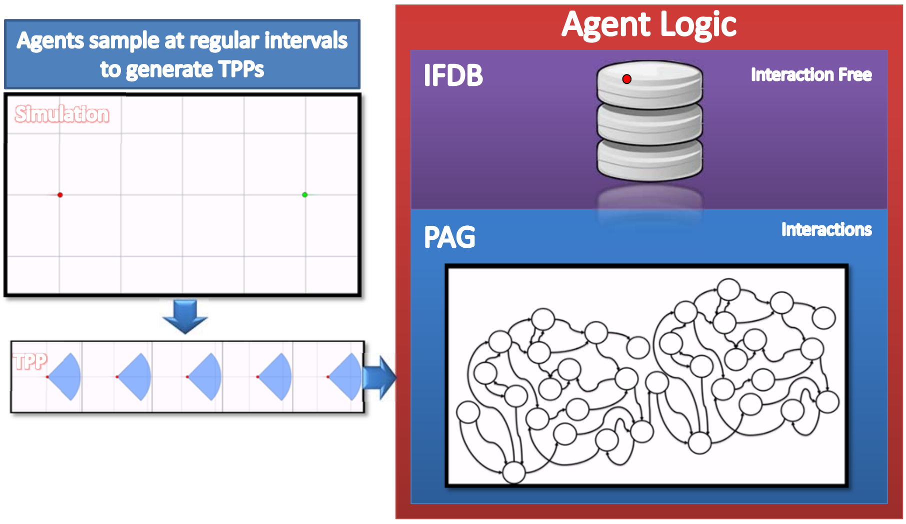
- Find similar states, group them and interconnect them with **actions** that followed them
- Actions: Trajectory Segments



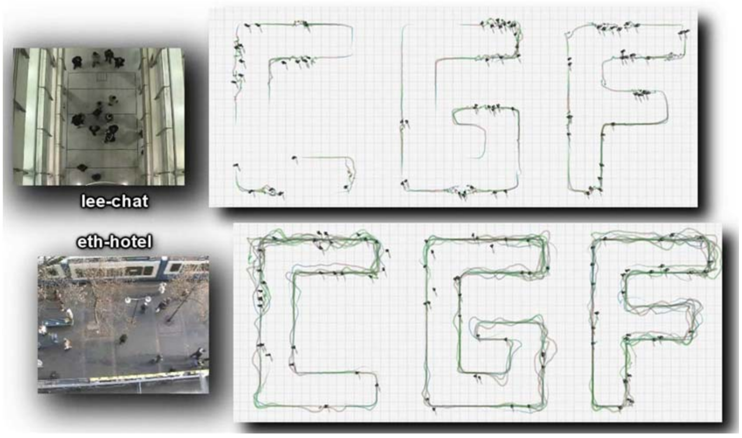


EUROGRAPHICS 2014 Strasbourg, France

Simulation



Some Results



Running many times faster than the basic data-driven method

People Don't Just Walk

57

- Real people:
  - Talk to each other.
  - Look around.
  - Check the time.
- Simulated people:
  - Don't talk to each other.
  - Don't look around.
  - Just walk.

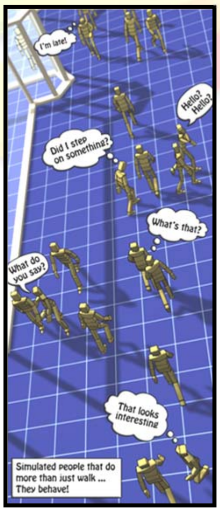
A Regular Crowd on a City Sidewalk

Lerner, Fitusi, Chrysanthou and Cohen-Or, *Fitting Behaviors to Pedestrian Simulations*, SCA 2009

Fitting Secondary Behaviors to Crowds

58

- We fit low-level actions to simulated pedestrian crowds.
- It is a data-driven method that decides for each person:
  - Which action to assign?
  - When to assign it?
  - How long should it last?

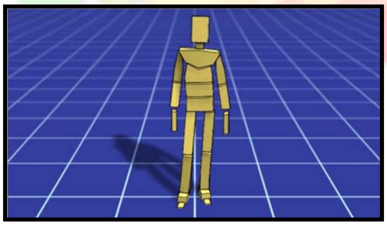


The Actions That We Assign

59

- The assigned actions:
  - Make the people seem more alive.
  - Create the impression of interaction.
  - Enhance the crowd's natural appearance.

  - Not just statistical observations

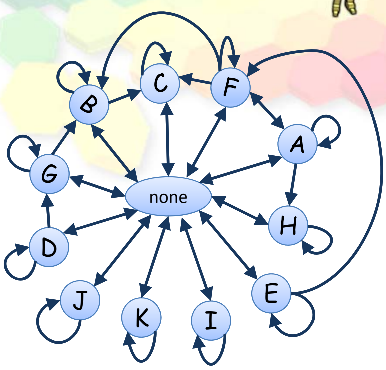


The assigned actions do not affect the trajectories!!!

The Action-Graph

60

- Nodes represent actions.
- Edges represent observed transitions between actions.
- Examples are stored on the edges of the graph.

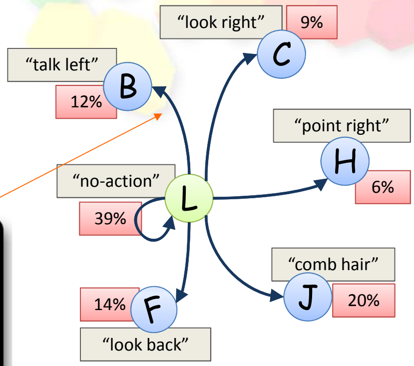


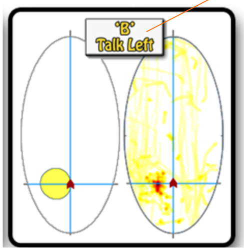
A - Talk (right side)
B - Talk (left side)
C - Look (right)
D - Look (left)
E - Look (down)
F - Look (back)

G - Point (left)
H - Point (right)
I - Talk on cell
J - Comb hair
K - Check time
L - No action

Approximating Probability

1. For each potential action find the examples most similar to the query and sum their similarity values.
2. Normalize the similarity values of the actions.





stimuli and validity-maps

Conclusion

Very nice "natural" looking results **but**:

- How do you extrapolate to non observed conditions?
- How can you easier define the crowd scenario?

Crowd Software – tracking and behavior annotation

Data sets – ours and links to others

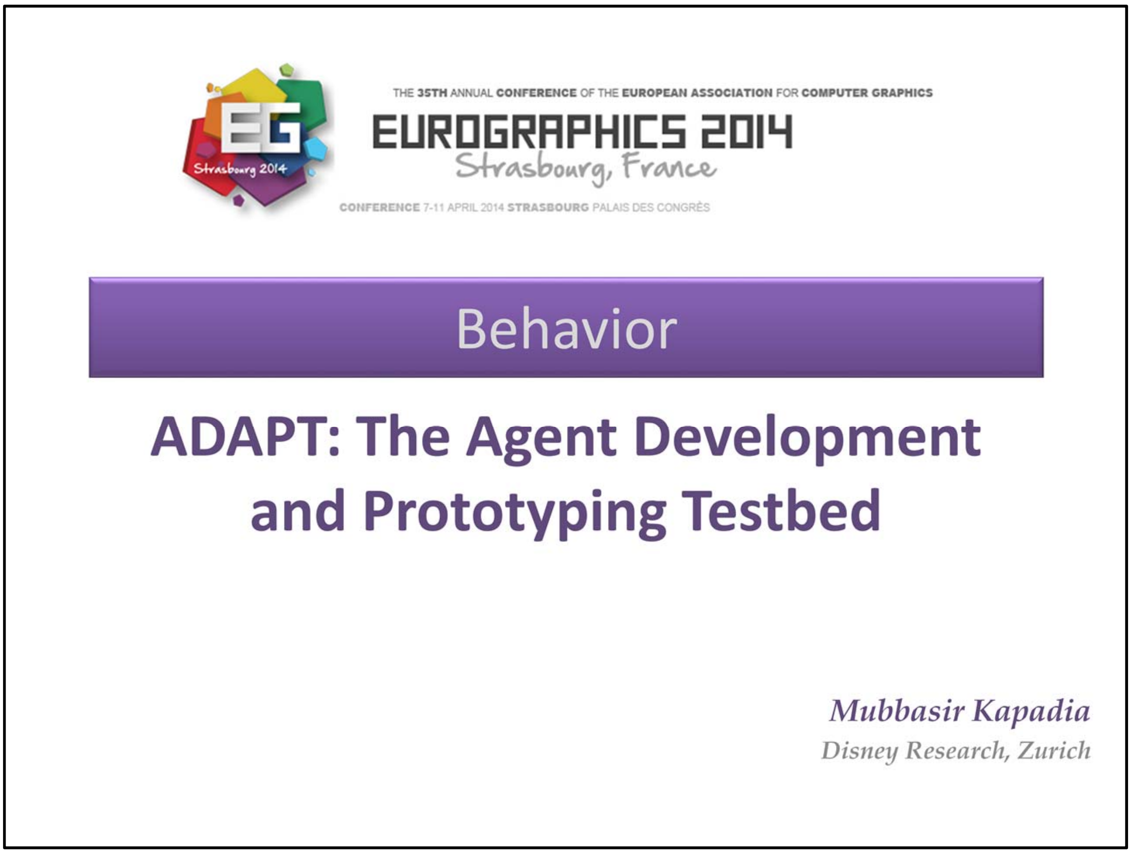
http://graphics.cs.ucy.ac.cy/downloads/crowds-software-and-data

Thank You

- See our web site:
  http://graphics.cs.ucy.ac.cy/downloads/crowds-software-and-data

- Acknowledgments:

Panayiotis Charalambous, University of Cyprus
Nikolas Ladas, University of Cyprus
Andreas Aristides, University of Cyprus
Haris Zacharatos, University of Cyprus
Stathis Stavrakis, University of Cyprus
Alon Lerner, Tel Aviv University
Daniel Cohen-Or, Tel Aviv University
Dani Lischinski, Hebrew University of Jerusalem
Ariel Shamir, IDC Herzliya



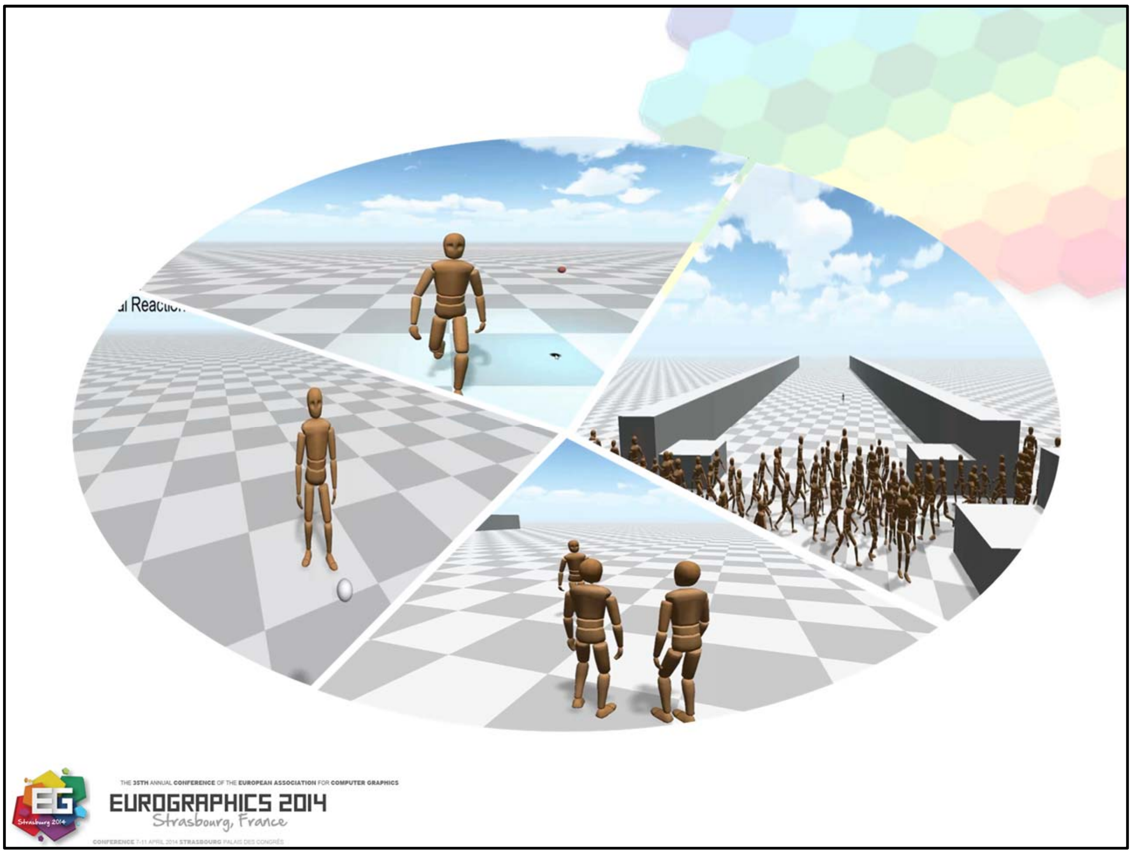
Speaker: Mubbasir Kapadia, Associate Research Scientist, Disney Research Zurich

Bio. Mubbasir Kapadia is an Associate Research Scientist at Disney Research Zurich. Previously, he was a postdoctoral researcher and Assistant Director at the Center for Human Modeling and Simulation at University of Pennsylvania. He was the project lead on the United States Army Research Laboratory (ARL) funded project Robotics Collaborative Technology Alliance (RCTA). He received his PhD in Computer Science at University of California, Los Angeles.Kapadia's research aims to develop integrated solutions for full-body character animation, planning based control, behavior authoring, and statistical analysis of autonomous virtual human simulations. The far-reaching goal is to provide functional, purposeful embodied virtual humans, that act and interact in meaningful ways to simulate complex, dynamic, narrative-driven, interactive virtual worlds.

Related Publication(s):

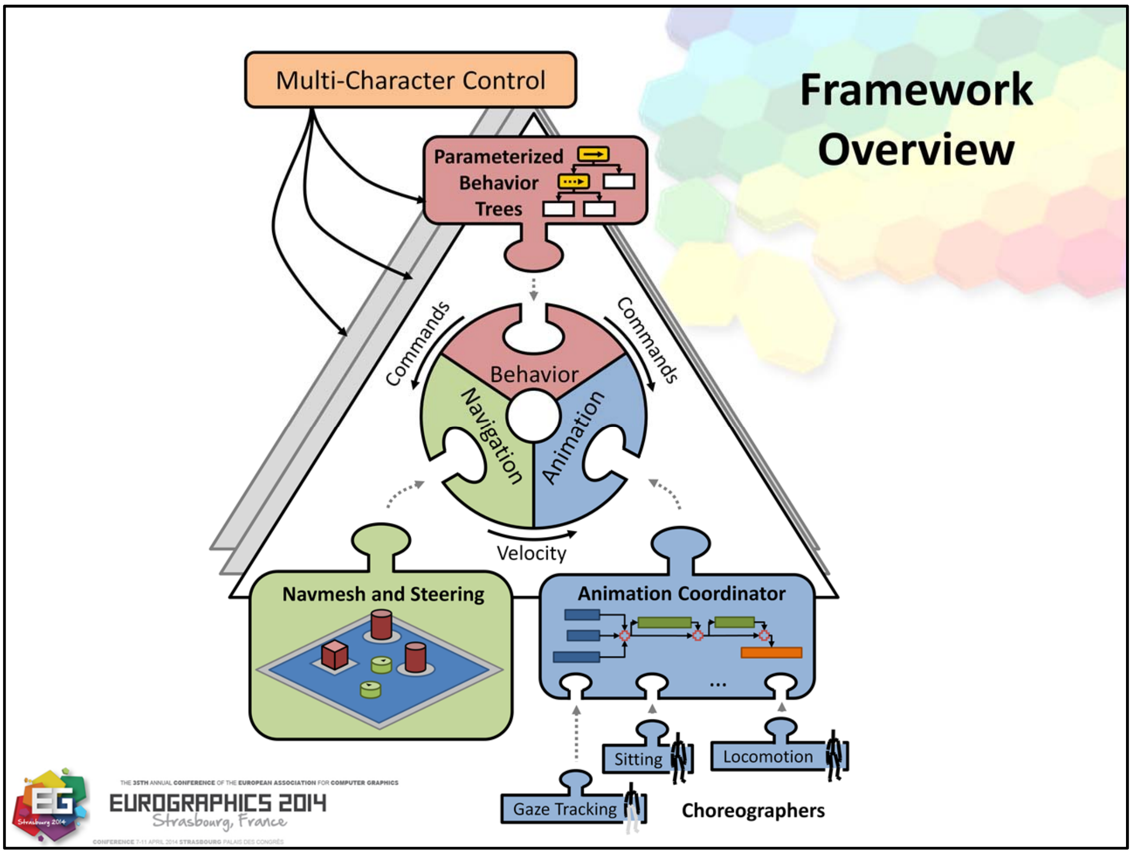
ADAPT: The Agent Development and Prototyping Testbed. Alexander Shoulson, Nathan Marshak, Mubbasir Kapadia, Norman I. Badler . *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, March 2013

ADAPT: The Agent Development and Prototyping Testbed. Alexander Shoulson, Nathan Marshak, Mubbasir Kapadia, Norman I. Badler *IEEE Transactions on Visualization and Computer Graphics (To Appear)*



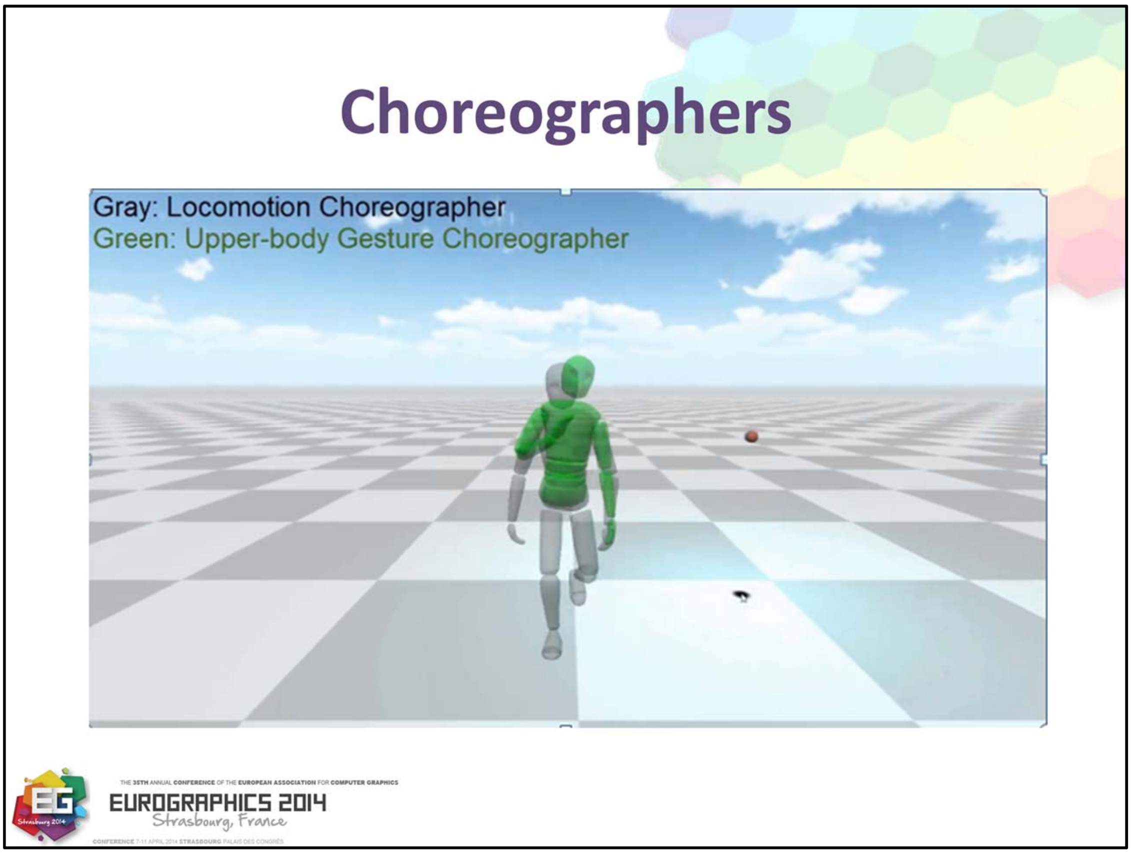
The next generation of interactive virtual worlds demand functional, purposeful, heterogeneous autonomous virtual humans, that exhibit rich, believable interactions with their environment and other agents with the far-reaching goal of complete immersion for the end users. With application in Urban planning, disaster and security simulation, training simulations, and serious games.

Project webpage: http://people.inf.ethz.ch/kapadiam/projects-adapt.html

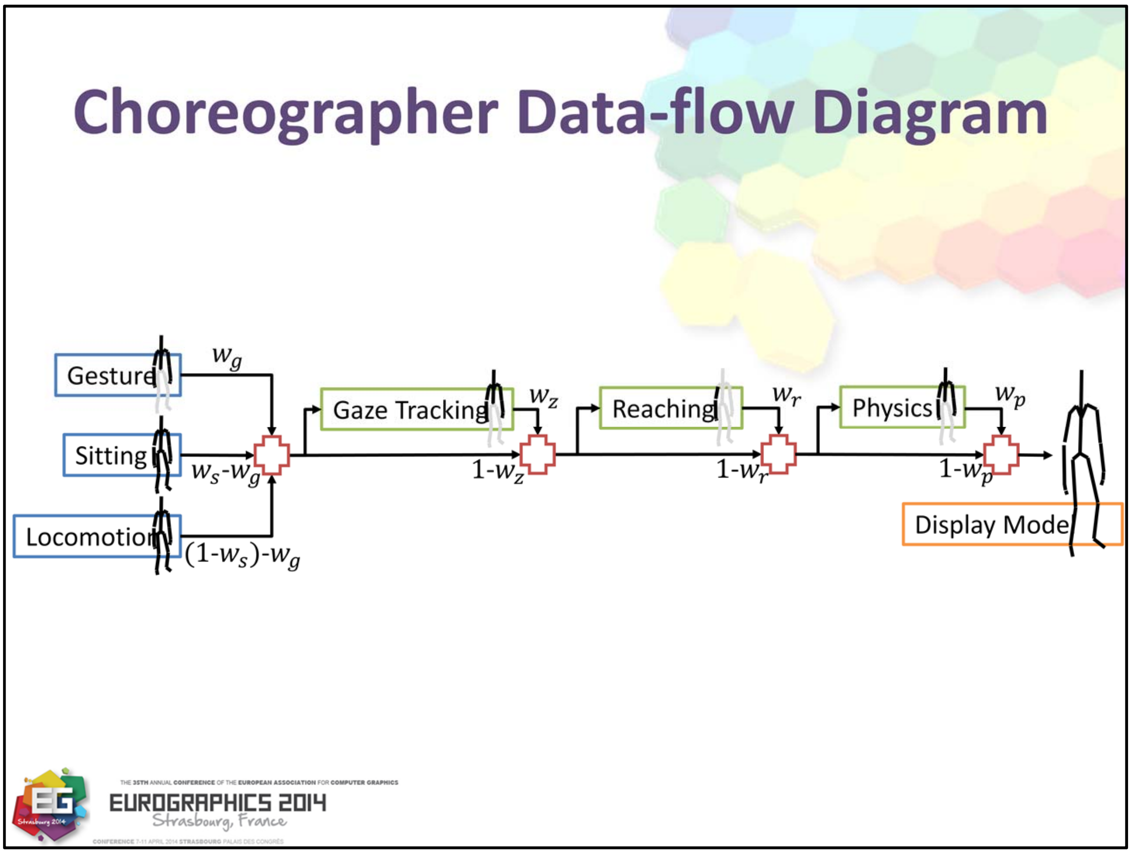


We present ADAPT: an open-source framework for authoring complex interactions between multiple embodied virtual characters. Our framework incorporates Navigation, full-body character animation, And behavior authoring For multi-character interactions using modular interchangeable components to facilitate collaboration and experimenation.

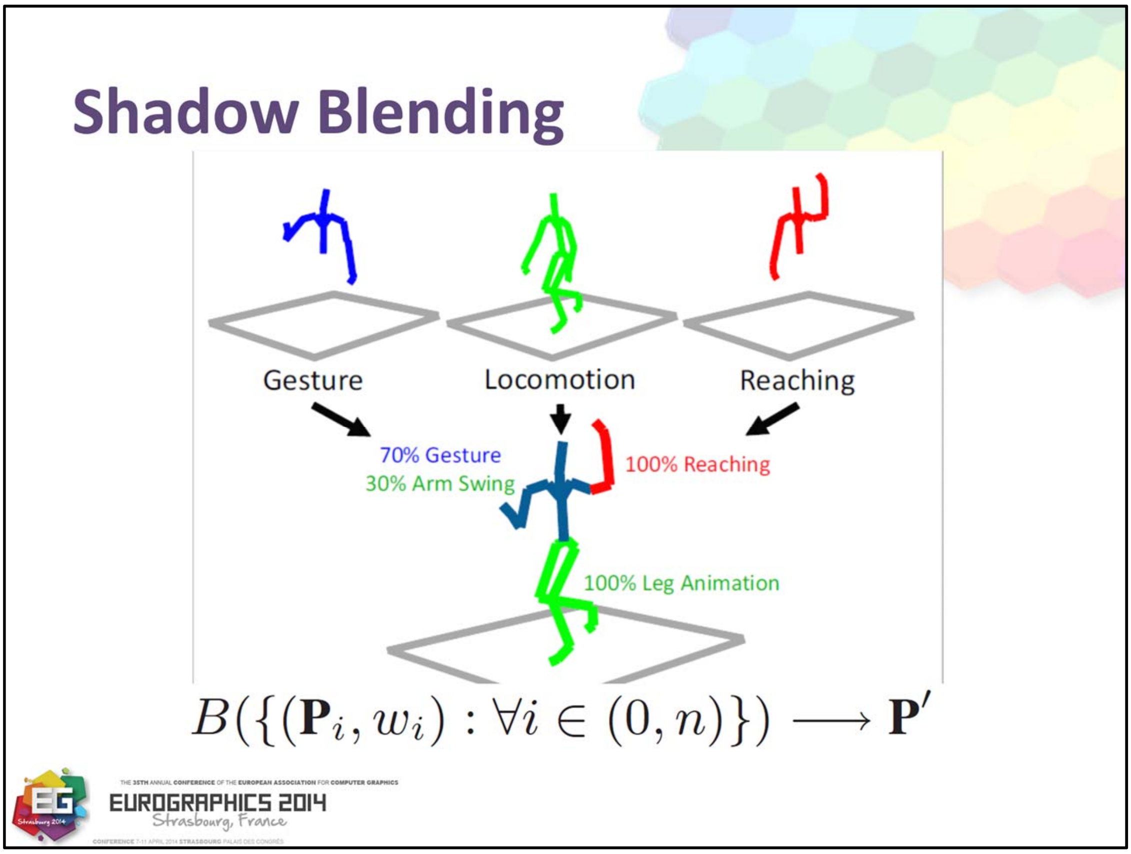
Animation system is divided into discrete controllers called choreographers Which share control of the body without being aware of one another.



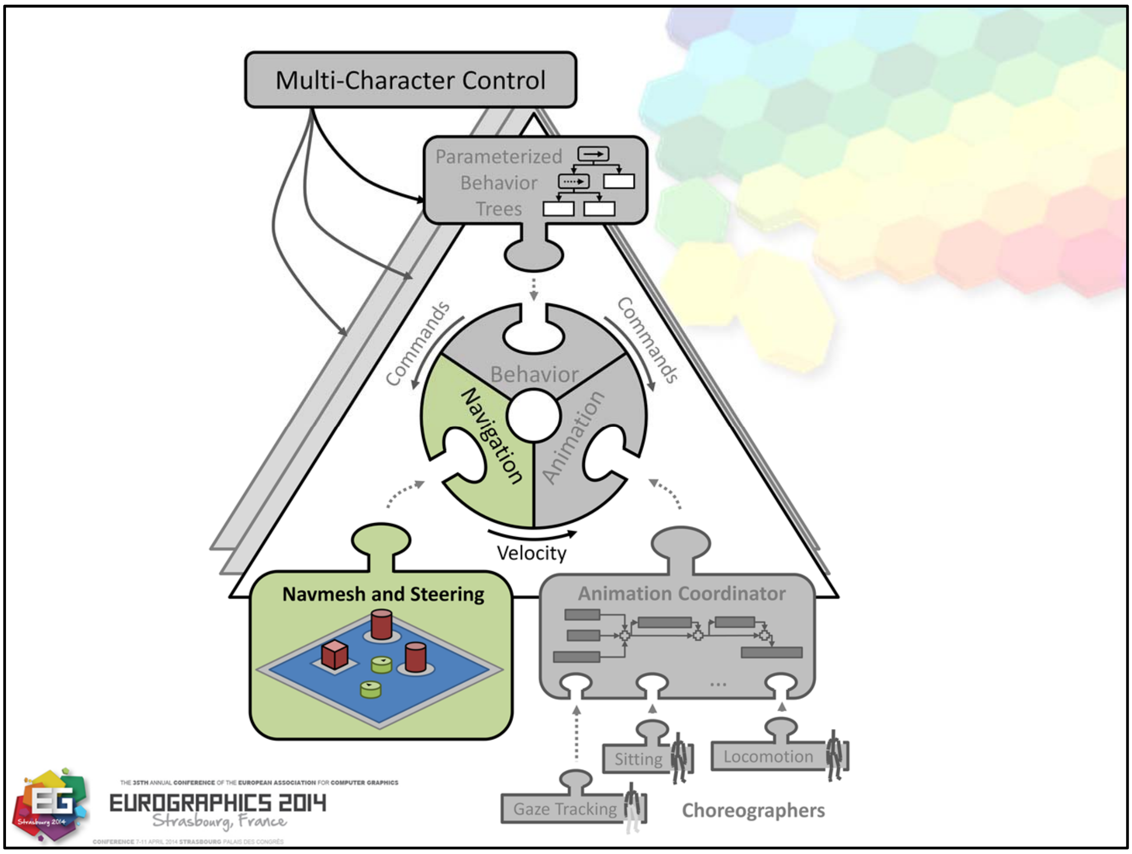
Rather than negotiating over control over parts of the body, Each choregorapher is allocated its own copy of the characters skeleton – called shadows Choreographers manipulate their own shadows without interruption



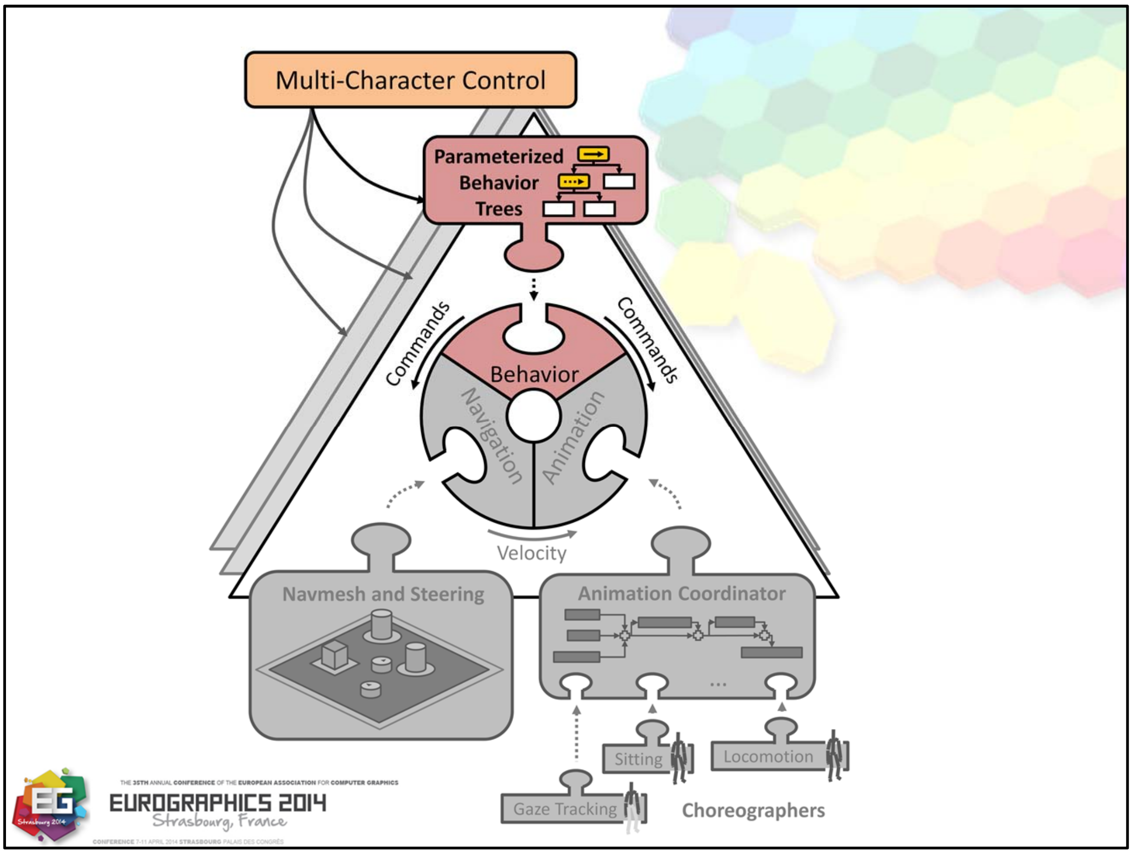
A central coordinator blends the different shadows of each choreographer using a user defined data flow diagram To produce the final character pose



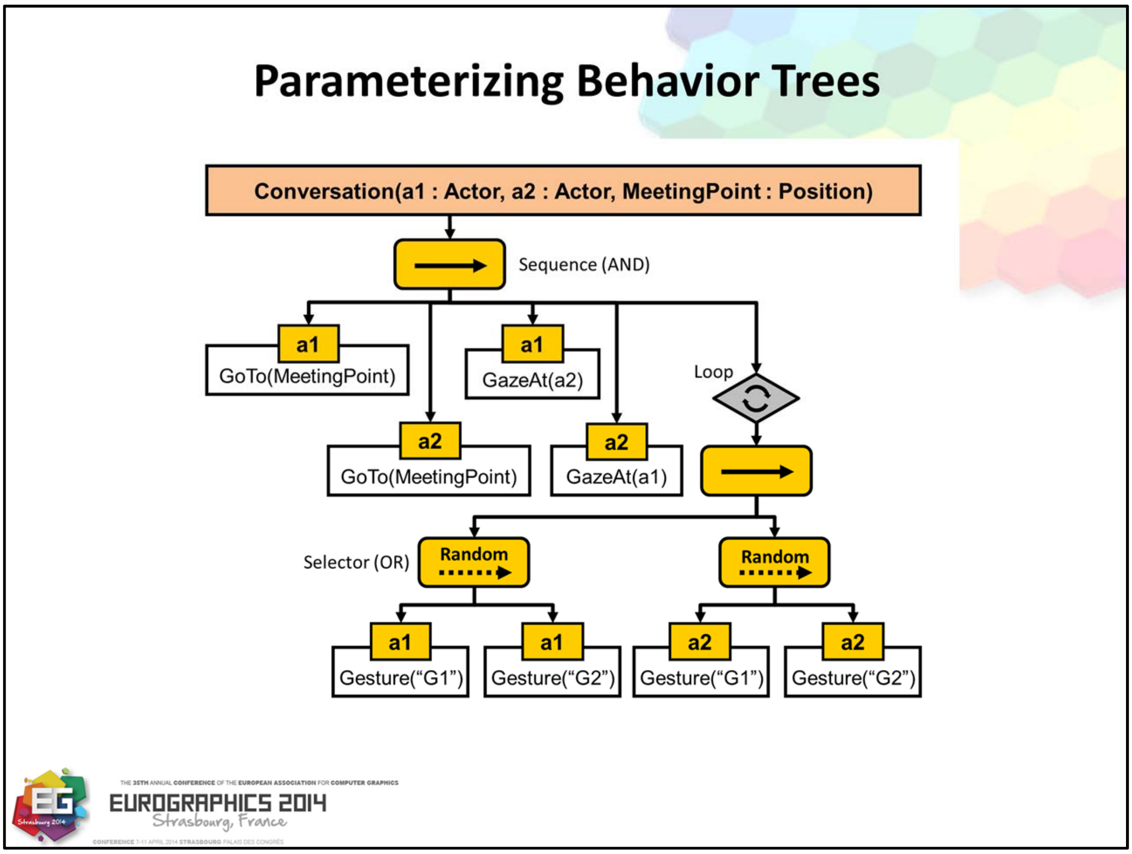
This is illustrated in this example where we have different choreographers such as gestures, locomotion, and reaching that are operating on different subsets of the body. The coordinator takes as input the shadows of these choreographers and produces the final blended result.



ADAPT currently supports RVO, Recast, and SteerSuite for navigation



and uses parameterized behavior trees for multi-character decision making



Events are defined using parameterized behavior trees which take one or more actors as parameters, and is granted exclusive control over those characters' actions for the duration of the event.
A conversation is authored as a simple sequence of commands directing agents to arrive at a meeting point, face one another and take turns playing gesture animations.

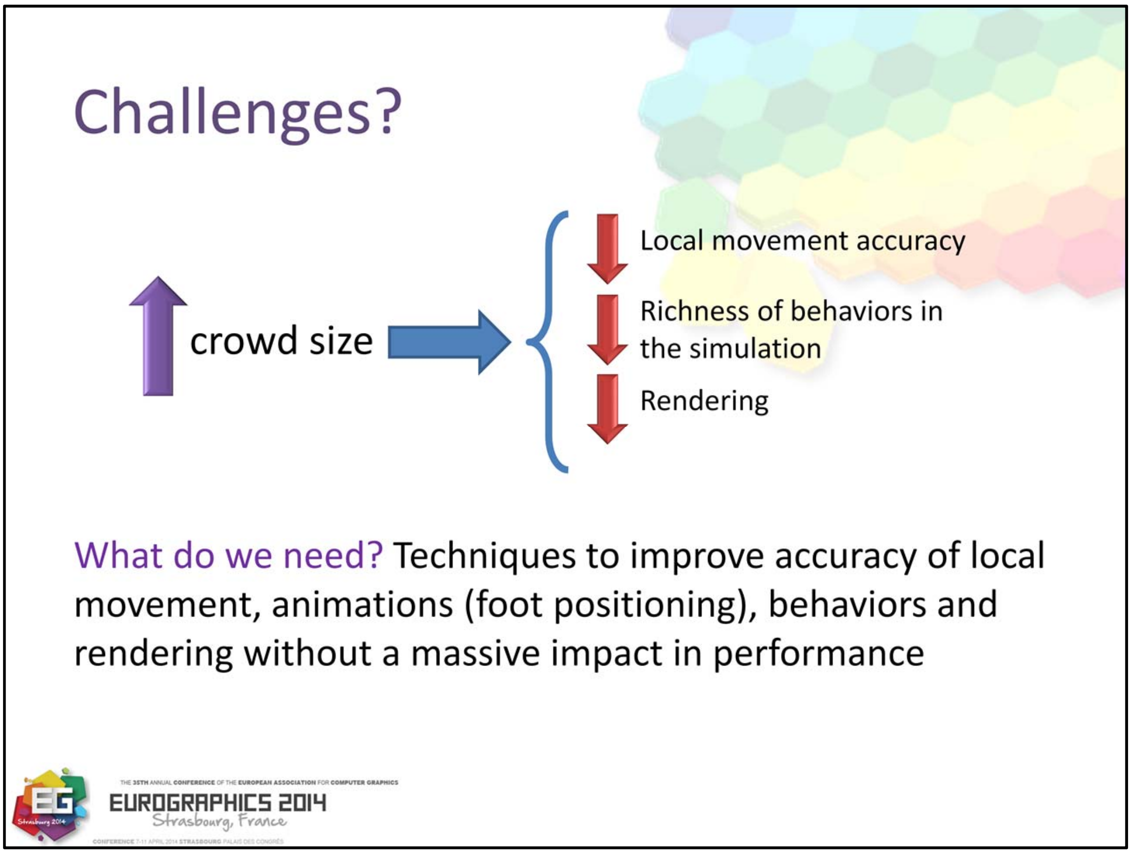
Events once active, can be successfully executed or interrupted by other triggers due to dynamic events, or user input. This produces a rich interactive simulation where virtual characters can be directed with a high degree of fidelity, without sacrificing autonomy or burdening the user with authoring complexity.

Complex narrative sequence can be authored using ADAPTs animation, navigation, and behavior capabilities by defining a library of reusable behaviors which are instantiated based on different triggers.

Once we achieve believable crowds from the point of view of navigation and path planning, we also need to focus our attention on the overall realism. To enhance realism this course focuses on three relevant topics: highly detailed rendering of large crowds at interactive rates, removal of artifacts in local movement and animation, and enhancement of crowd personality.

Speaker: Nuria Pelechano
Assistant Professor, Universitat Politecnica de Catalunya

Nuria Pelechano is an Associate Professor at the Universitat Politecnica de Catalunya, where she is a member of the Virvig-Moving group. Her research interests include animation, simulation and rendering of crowds, real-time 3D graphics, and human avatar interaction in virtual environments.
She has published many papers in international conferences and journals including Symposium on Computer Animation, Computer Graphics Forum, and IEEE Computer
Graphics and Applications. She is also co-author of a book entitled "Virtual Crowds: Methods, Simulation, and Control“ (Morgan and Claypool) with N. Badler and J. Allbeck.

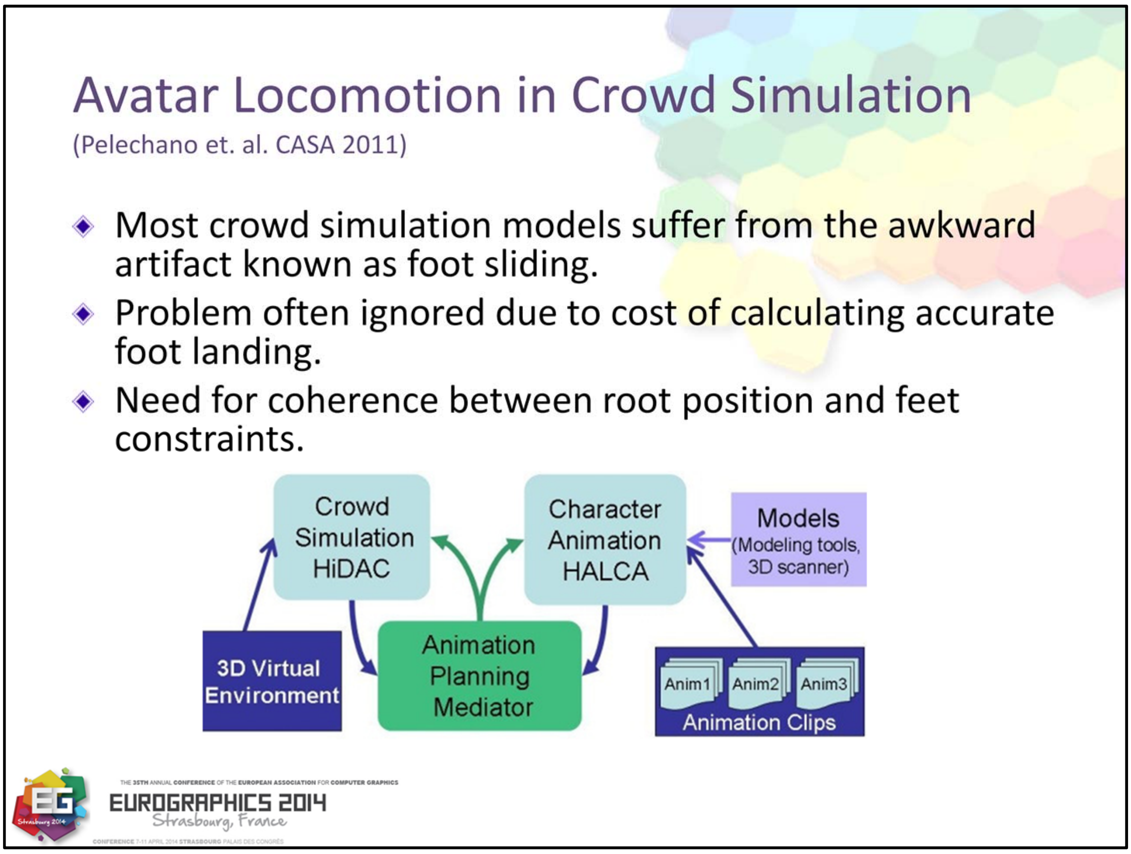


When simulating virtual humans, there is always a trade-off between the number of characters and the realism achieved in terms of simulation, locomotion and rendering.

It is therefore necessary to investigate new techniques that can achieve improvements in realism without having a massive impact over performance.

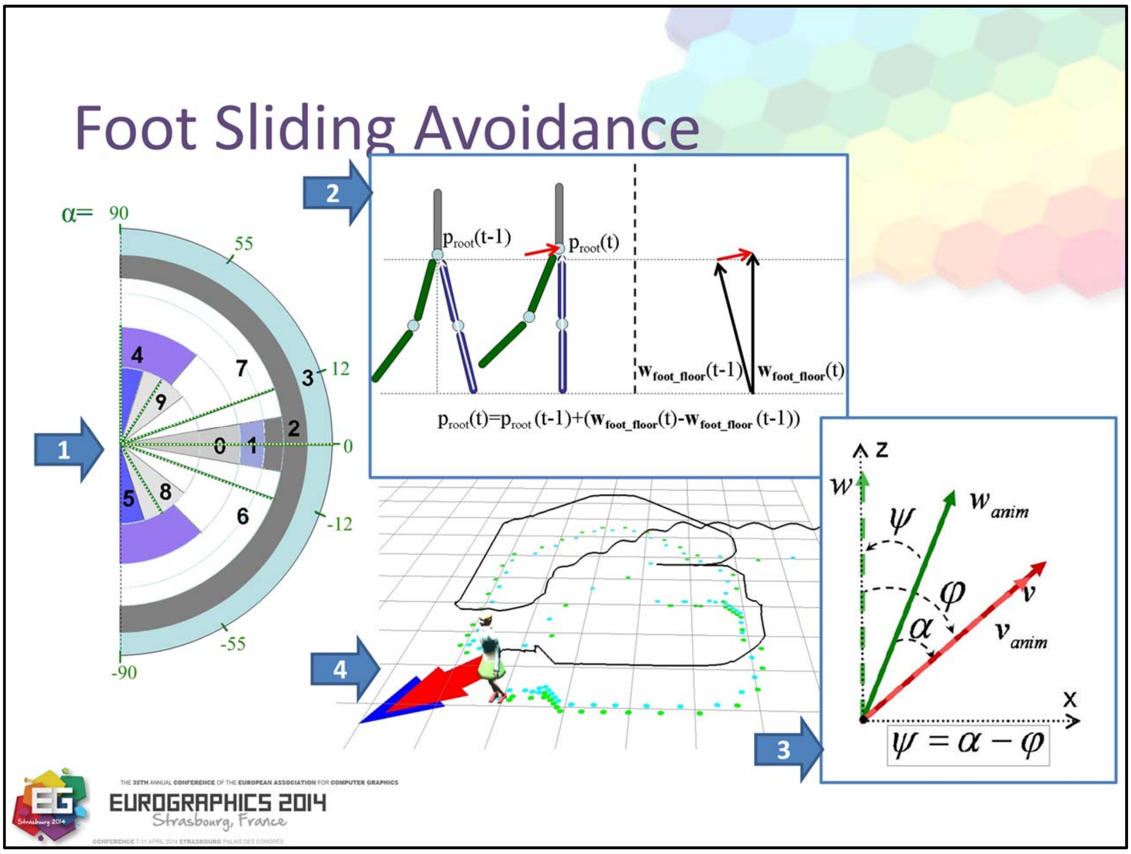
Recommended reading:

Avatar Locomotion in Crowd Simulation. N. Pelechano, B. Spanlang, A. Beacco. (*CASA2011) The International Journal of Virtual Reality (IJVR)* vol. 10, no. 1, pp 13-19, March 2011.



When it comes to character locomotion in crowds, in most situations the characters are moved around the environment following the center of mass trajectories and velocities given by the crowd simulation system. This can result in inconsistencies between the movement of the character and the planting of the feet over the floor. In other words, this situation leads to the foot sliding problem.

In 2011, we presented an Animation Planning Mediator (APM) designed to synthesize animations efficiently for virtual characters in real time crowd simulation. From a set of animation clips, the APM selects the most appropriate and modifies the skeletal configuration of each character to satisfy desired constraints (e.g. eliminating foot-sliding or restricting upper body torsion), while still providing natural looking animations. We use a hardware accelerated character animation library to blend animations increasing the number of possible locomotion types. The APM allows the crowd simulation module to maintain control of path planning, collision avoidance and response. The APM can be integrated with any crowd simulator working in continuous space.

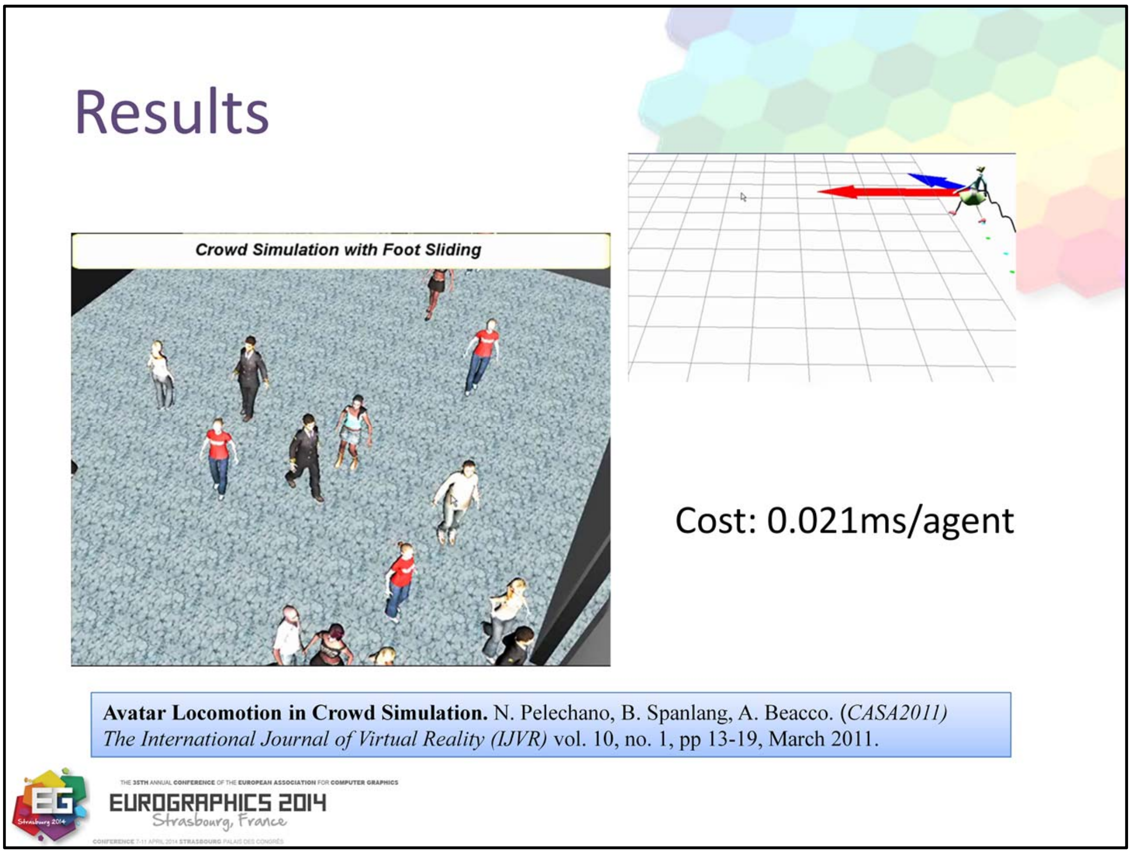


During off-line analysis:
1.- First of all we need to classify the available animations based on velocity and angle between velocity vector and torso orientation. This step is performed off-line, and during simulation we will select and animation based on those two parameters.

During run-time:
2.- Foot displacement is calculated from leg movement.

3.- By aligning the animation velocity vector with the simulation velocity vector, followed by a correction of the torso orientation between current crowd simulation and animation clip being played, we can get our character to move in the desired velocity direction, with the torso facing the desired torso orientation.

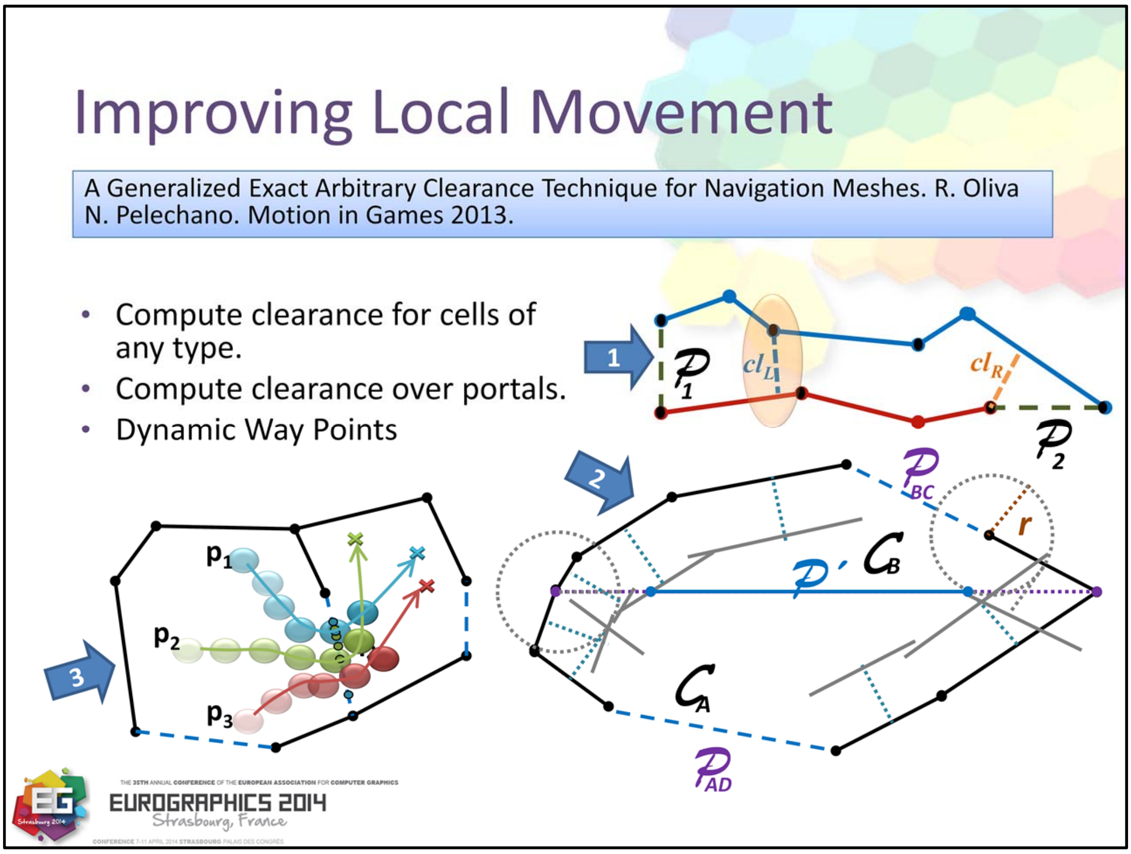
4.- This simple and efficient method not only provides accurate foot positioning, but also adapts to current torso orientation provided by the crowd simulation system.



This method achieves natural animation without foot-sliding for crowds, while being computationally inexpensive.

Even when we only have a small set of animation clips, our technique allows a large and continuous variety of movement. It can be used with any crowd simulation software, since it is the crowd simulation module which drives the movement of the virtual agents and our module limits its work to adjusting the root displacement and skeletal state.

Some videos can be found here:
http://www.lsi.upc.edu/~npelechano/videos/5%20-%20comparisonCrowdCASA.mkv
http://www.lsi.upc.edu/~npelechano/videos/4%20-%20crowdCASA.mkv
http://www.lsi.upc.edu/~npelechano/videos/3%20-%20comparisonOneAgentCASA.mkv
http://www.lsi.upc.edu/~npelechano/videos/2%20-%20oneAgentCASA.mkv

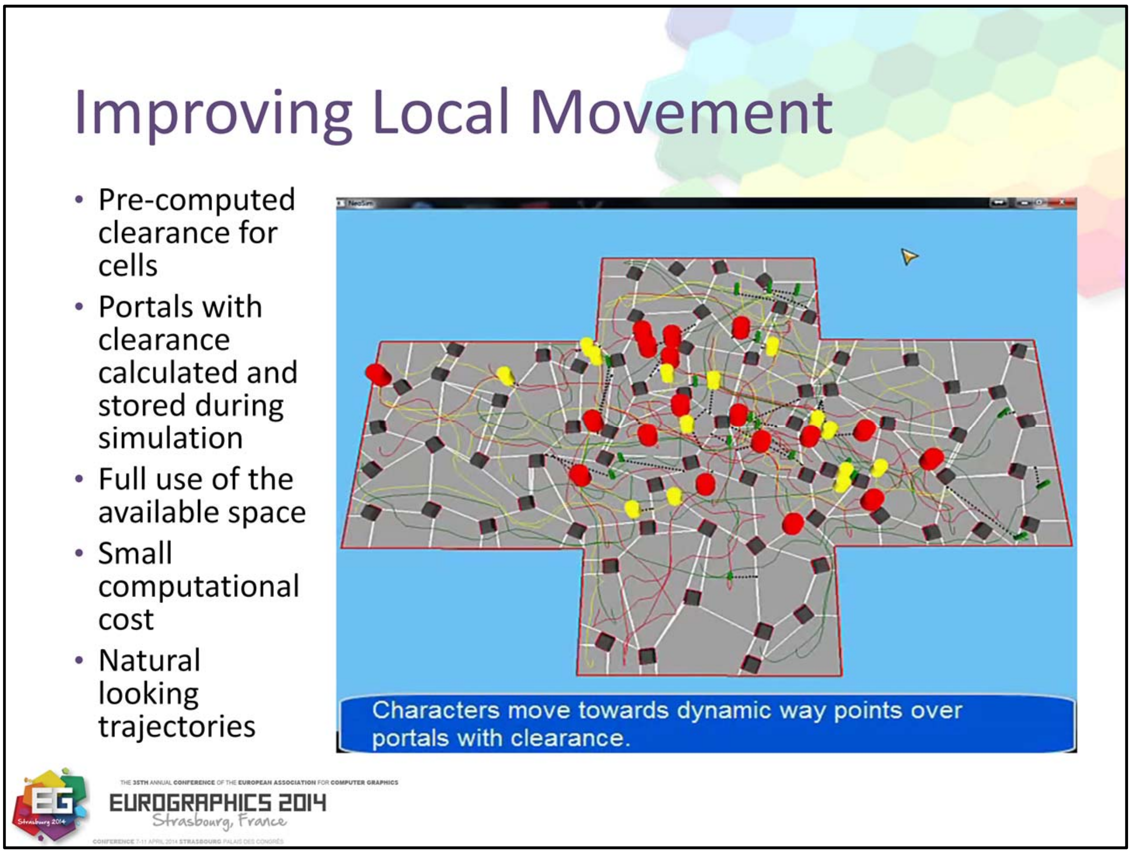


Realism needs to be improved at many levels, if we want to achieve realistic heterogeneous crowds. We have seen how to improve realism in terms of locomotion, now we will briefly see how we can improve the paths followed by characters in virtual environments.

There are two frequent artifacts in crowd simulation, the first one appears when all agents attempt to traverse the navigation mesh sharing the same way point over portals, increasing the probability of collision against other agents and lining up towards portals; the second one is caused by way points being assigned at locations where clearance is not guaranteed which causes the agents to walk too close to the static geometry, slide along walls or even get stuck.

In 2013 we proposed a novel method for:

1) Calculating clearance for paths in cells of any shape.
2) Calculating clearance over portal by displacing the possible obstacles against the portal a distance r (r being the radius of the character). This guarantees that characters will be steered towards points that are free of collision against the static geometry.
3) dynamically calculating way points based on current trajectory, destination, and clearance while using the full length of the portal, thus guaranteeing that agents in a crowd will have different way points assigned. This reduces the number of collisions between agents. Attractors are set mainly at the orthogonal projection of each character over the next portal.



Video can be found in:
http://www.lsi.upc.edu/~npelechano/videos/MIG2013.mp4

Recommended reading:
A Generalized Exact Arbitrary Clearance Technique for Navigation Meshes. R. Oliva N. Pelechano. ACM Siggraph International conference on Motion in Games 2013.

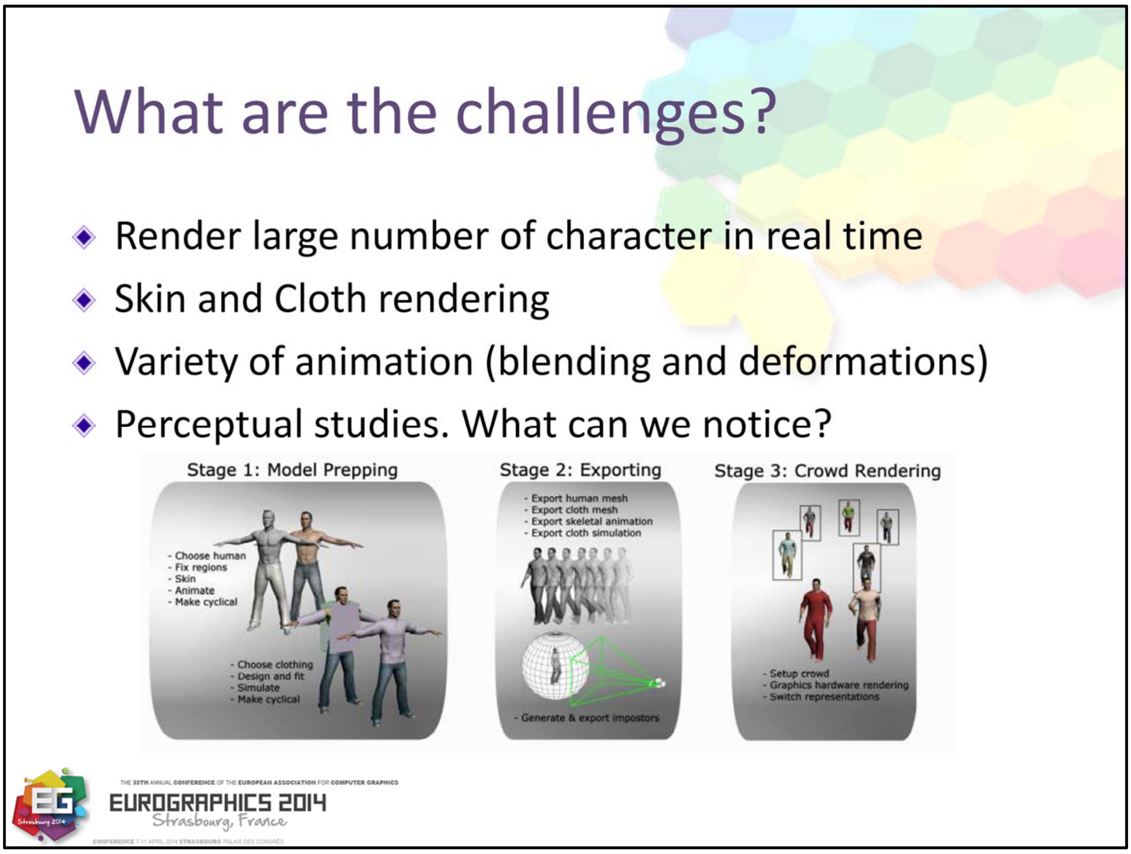
Speaker: Nuria Pelechano
Assistant Professor, Universitat Politecnica de Catalunya

Nuria Pelechano is an Associate Professor at the Universitat Politecnica de Catalunya, where she is a member of the Virvig-Moving group. Her research interests include animation, simulation and rendering of crowds, real-time 3D graphics, and human avatar interaction in virtual environments.
She has published many papers in international conferences and journals including Symposium on Computer Animation, Computer Graphics Forum, and IEEE Computer
Graphics and Applications. She is also co-author of a book entitled "Virtual Crowds: Methods, Simulation, and Control“ (Morgan and Claypool) with N. Badler and J. Allbeck.

Recommended reading:

Efficient rendering of animated characters through optimized per-joint impostors. A. Beacco, C. Andujar, N. Pelechano, B. Spanlang.
Journal of Computer Animation and Virtual Worlds. vol. 23, no.1, pp 33-47. 2012.

A flexible approach for output-sensitive rendering of animated characters. A. Beacco, B. Spanlang, C. Andujar, N. Pelechano.
Computer Graphics Forum. vol. 30, no. 8, pp 2328-2340. December 2011.

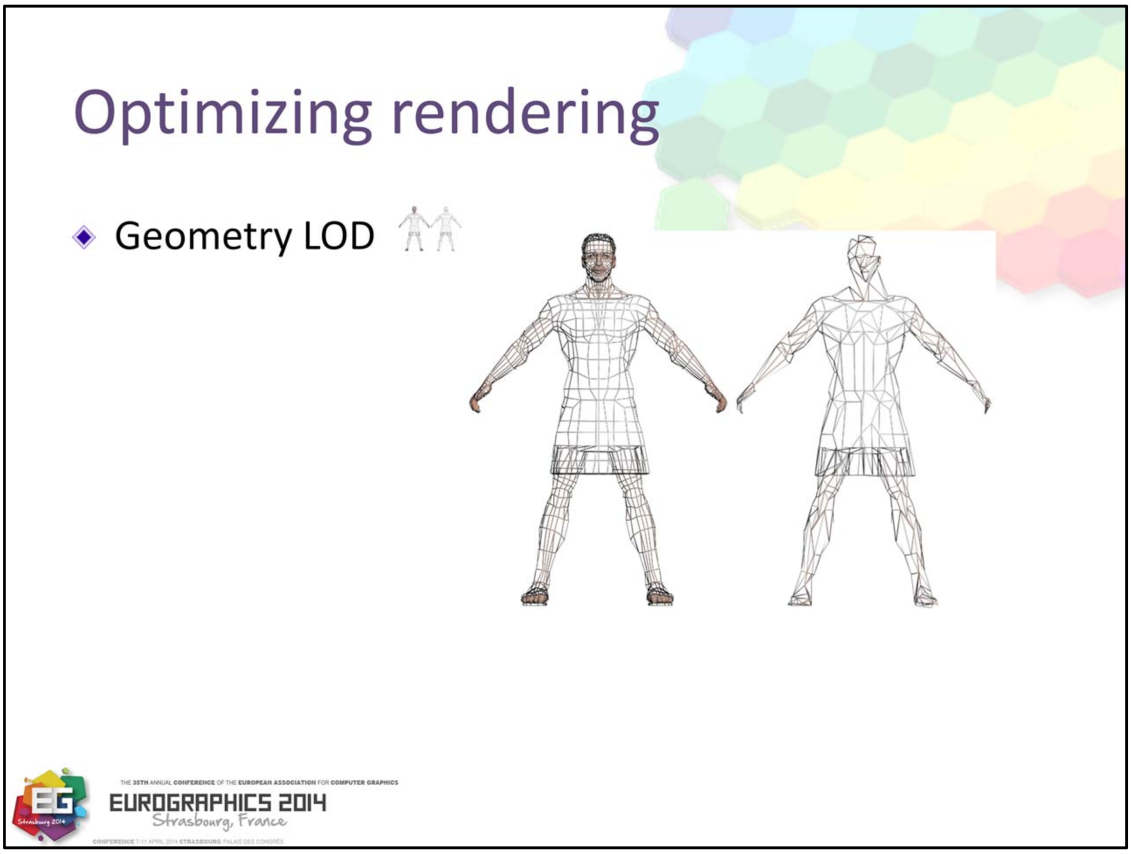


Rendering highly realistic crowd is an important element of achieving realism.

Detailed animated characters are created with high-quality textured polygonal meshes. When rendering a crowd of agents, the high cost of animating and rendering each one of those meshes becomes a problem. Thus rendering has become a major limiting factor in real-time crowd simulation, where we are always eager to have more interacting agents at the same time, with the higher number of possible details.
We will now see an overview and classification the different most relevant approaches in real-time crowd rendering.

When studying the realism of a crowd rendering we need to take into account also the variety of animations being used, and whether we can apply blending between animations to synthesize new ones.

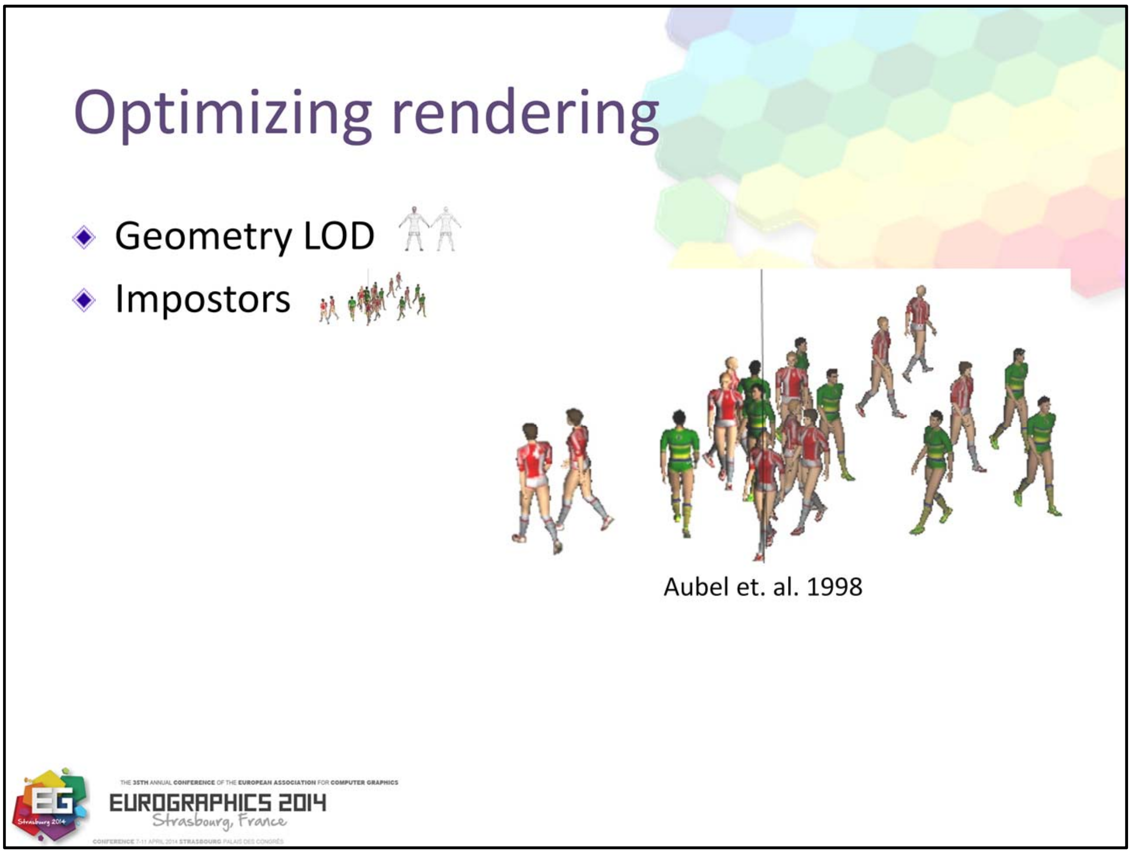
And last but not least, in any new crowd rendering approach it is important to run perceptual studies, both before and after developing a technique.
“Before” is important in order to study what elements of crowds rendering are most noticeable by the human eye, and “after” is important to evaluate if with the new technique developed we can improve performance without dropping realism.



A well known solution to the problem of accelerating crowd rendering involves applying level-of-detail (LOD) for the characters depending on their distance to the camera. LOD usually consists on decreasing the complexity of the 3D object or avatar representation as it moves away from the camera (i.e. using meshes to represent the skin and cloth with less number of vertices). Since it is further away, the viewer should not notice any difference.

The main problem appears when we animate the characters, because even with the best rigging techniques, artifact will appear due to inconsistent deformation.

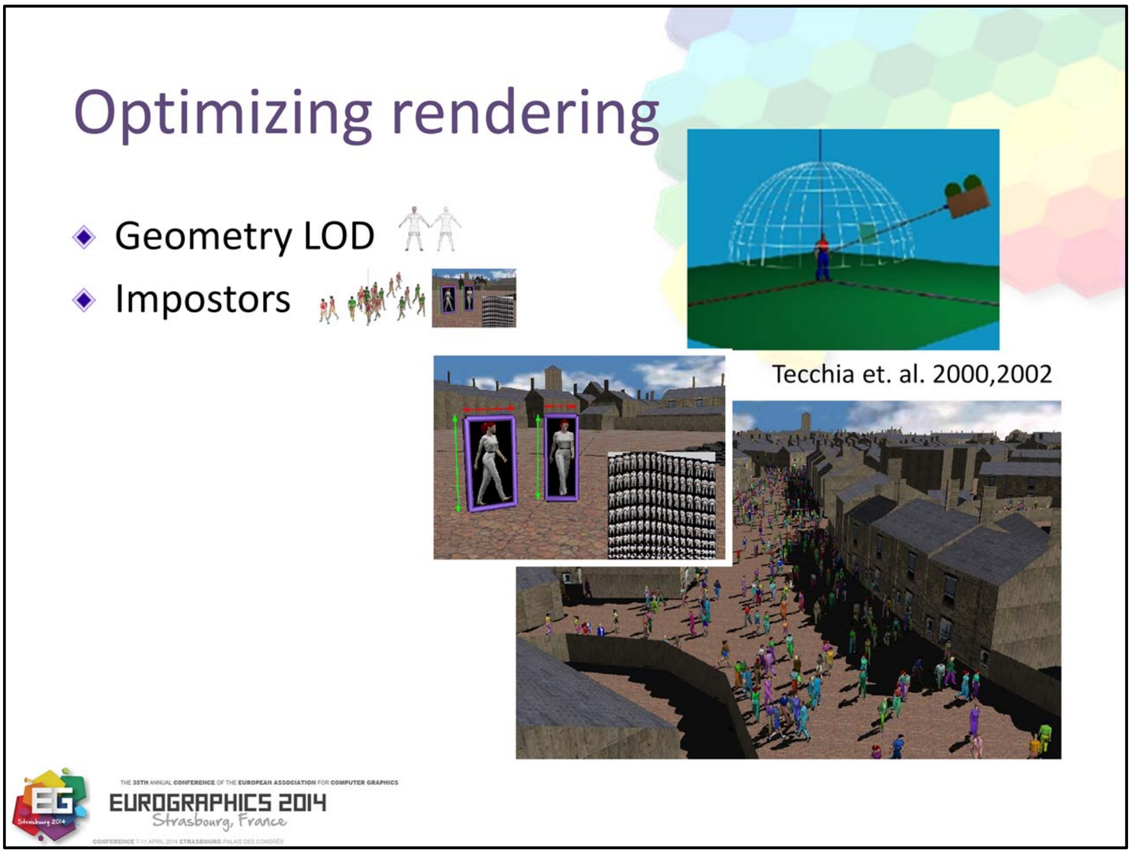
Approaches using LOD in the geometry and can either animate the vertices on the fly, or store deformed meshes that then can be used in cycles to animate the characters.



Then impostors were used as "a simple textured plane that rotates to continuously face the viewer." A snapshot of the virtual human is textured onto the plane. They could either be created at run-time or pre-computed as we will see in the next slide.

In the case of run-time impostors, as the humanoid moves or the camera moves, the mapped texture might need to be refreshed. To take updated snapshots they set-up an off-screen buffer to receive it and they place their multiresolution virtual human in front of the camera in the right posture. Then they render it and copy into texture memory, ready to be mapped onto the billboard.

The problem with impostors created at run-time is that it only works for crowds of individuals with similar orientation and animation, since it relies on reusing the current dynamically generated image over several frames in order to be efficient.



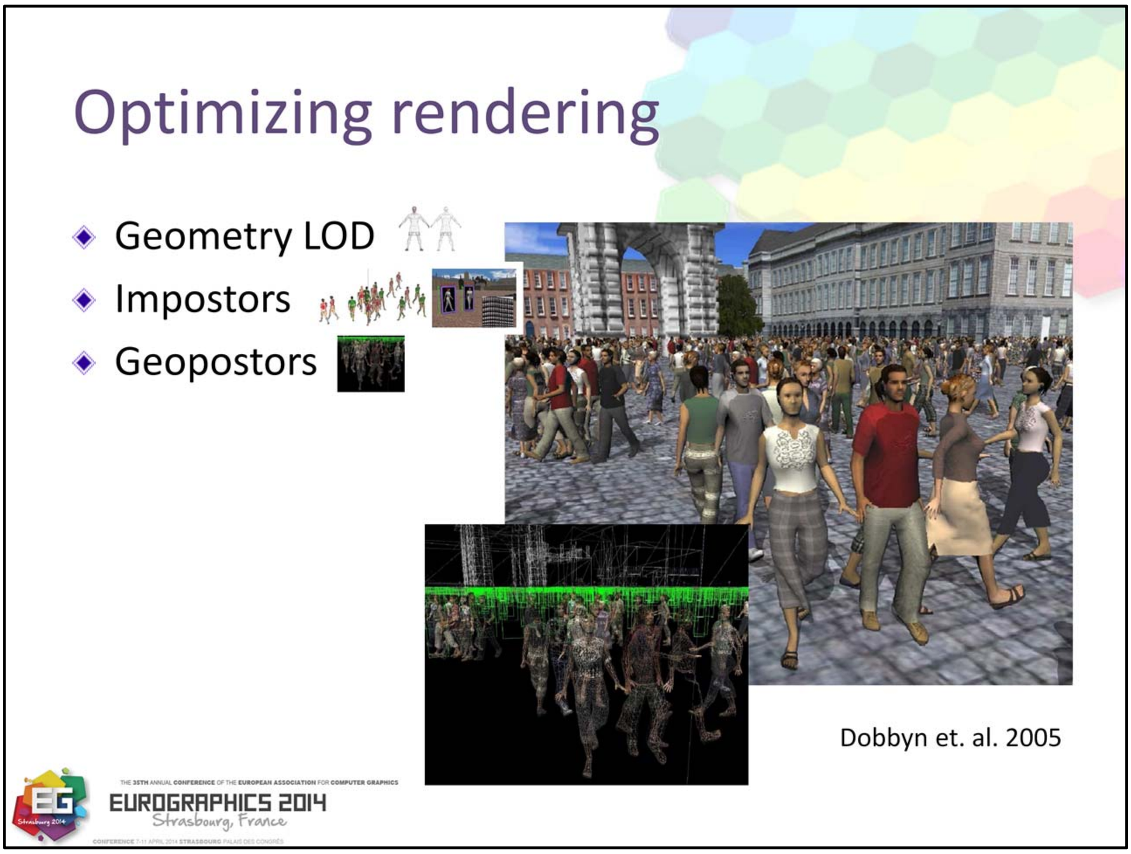
The other possibility with impostors is to have them pre-computed and stored. Tecchia and Chrysantou proposed this method in 2000. Pre-generated impostors where created by rendering each character from several viewpoints and for every animation frame of a simple animation cycle. The images were stored in a single texture atlas, and each crowd agent was rendered as a single polygon with suitable texture coordinates according to the view angle and frame (Sampled hemisphere, 32 positions, 8 elevations).

With this technique they were able to render a virtual city with thousands of walking humans on a standard PC by using pre-generated impostors to represent their humans.

Main disadvantages:

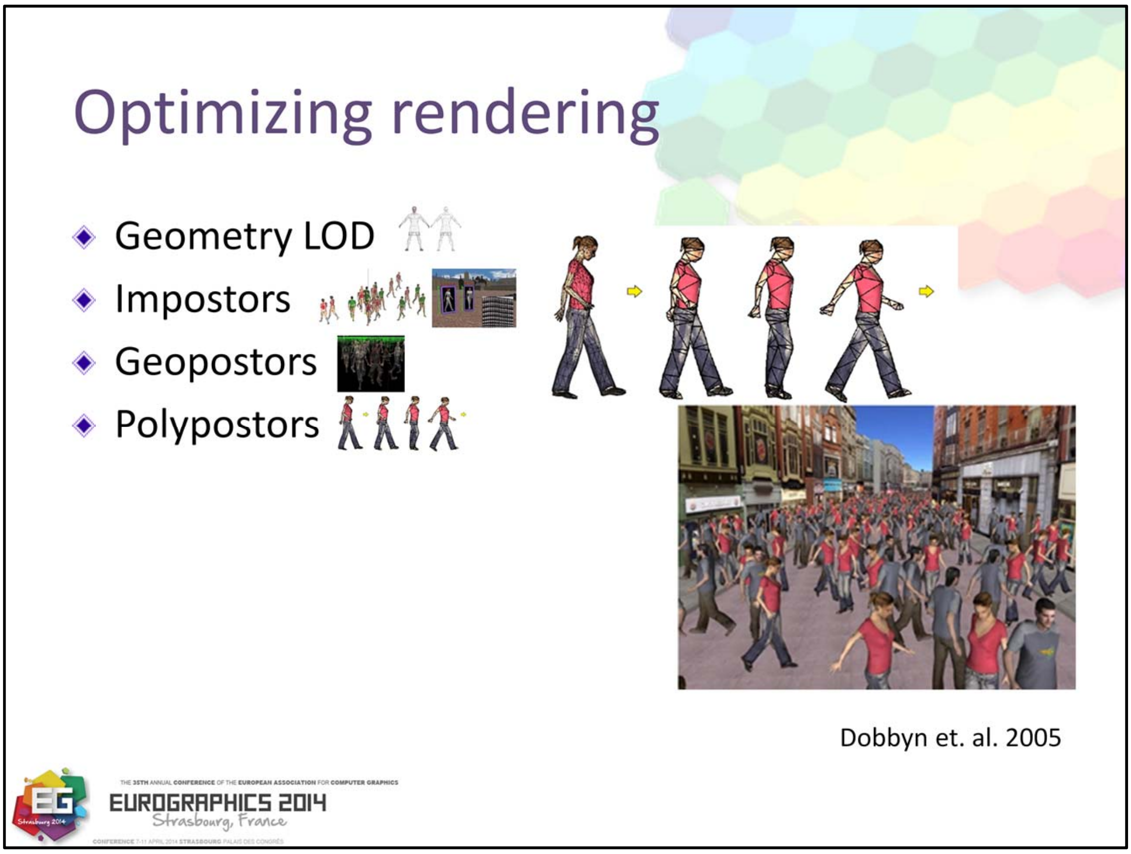
Impostors look pixelated when viewed up close.

Excessive texture requirements



Dobbyn et al. [2005] presented the first hybrid crowd system to solve the problem of degraded quality of impostors at close distances.

Impostor system on top of a full, geometry-based human animation system, and switching between the two representations with minimal popping artifacts based on a pixel to texel ratio.

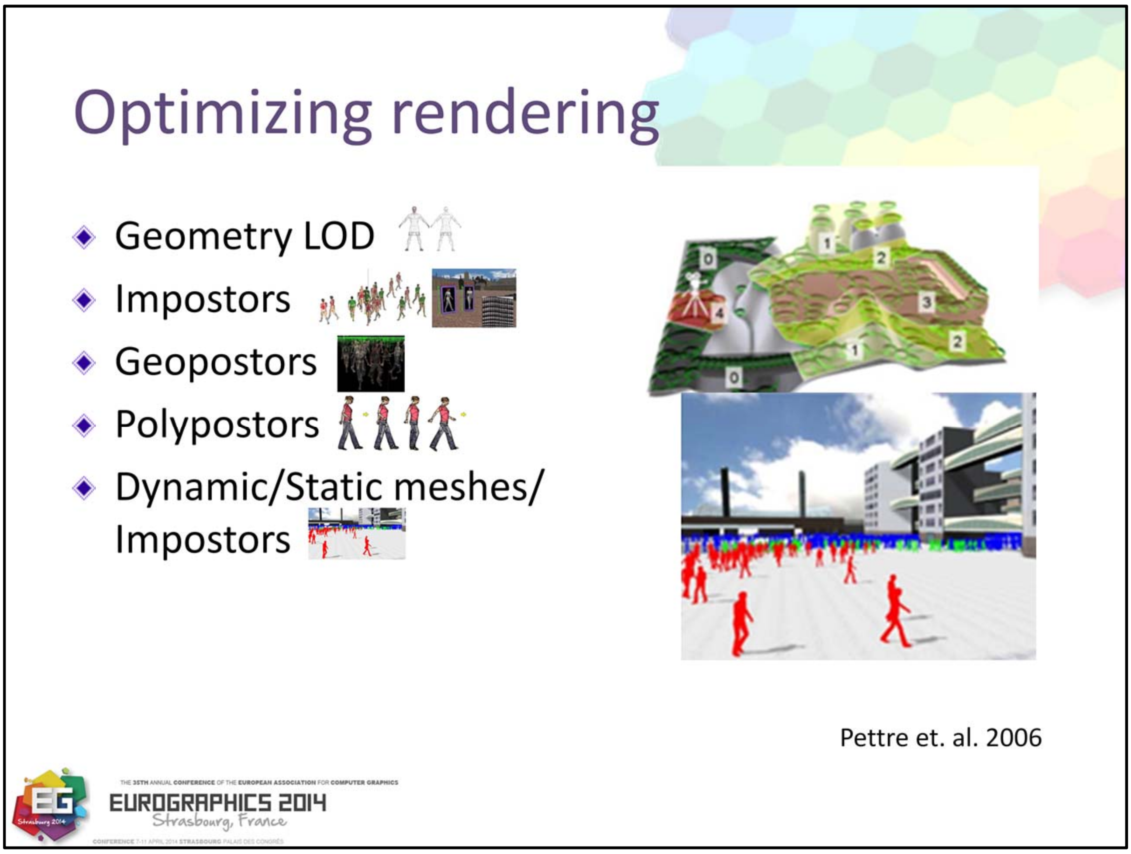


Problems with Impostors:
- Consume a lot of texture memory
- No in-betweening is possible
- Variety of animations very limited

Polypostors: “Automatically” generated 2D polygonal characters

Advantages: similar high level of rendering efficiency and visual fidelity, with considerably lower memory requirements (up to a factor of 30)

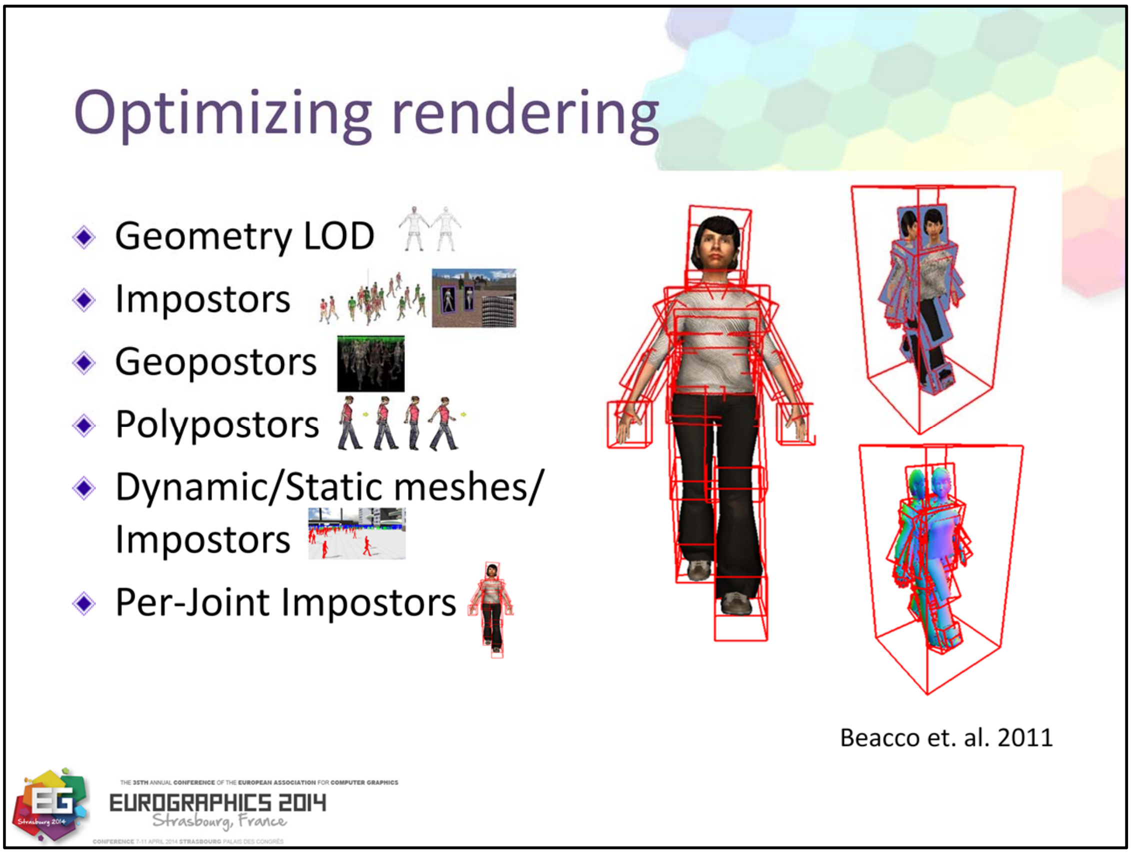
Disadvantages: not fully automatic, and suffers from many visual artifacts (gaps and overlaps)



Pettre et al. [2006] 3 LOD, combining the strengths of the animation quality of dynamic meshes with the performance of static meshes and impostors.
Static meshes: Vertex deformations precomputed with respect to a set of animations.

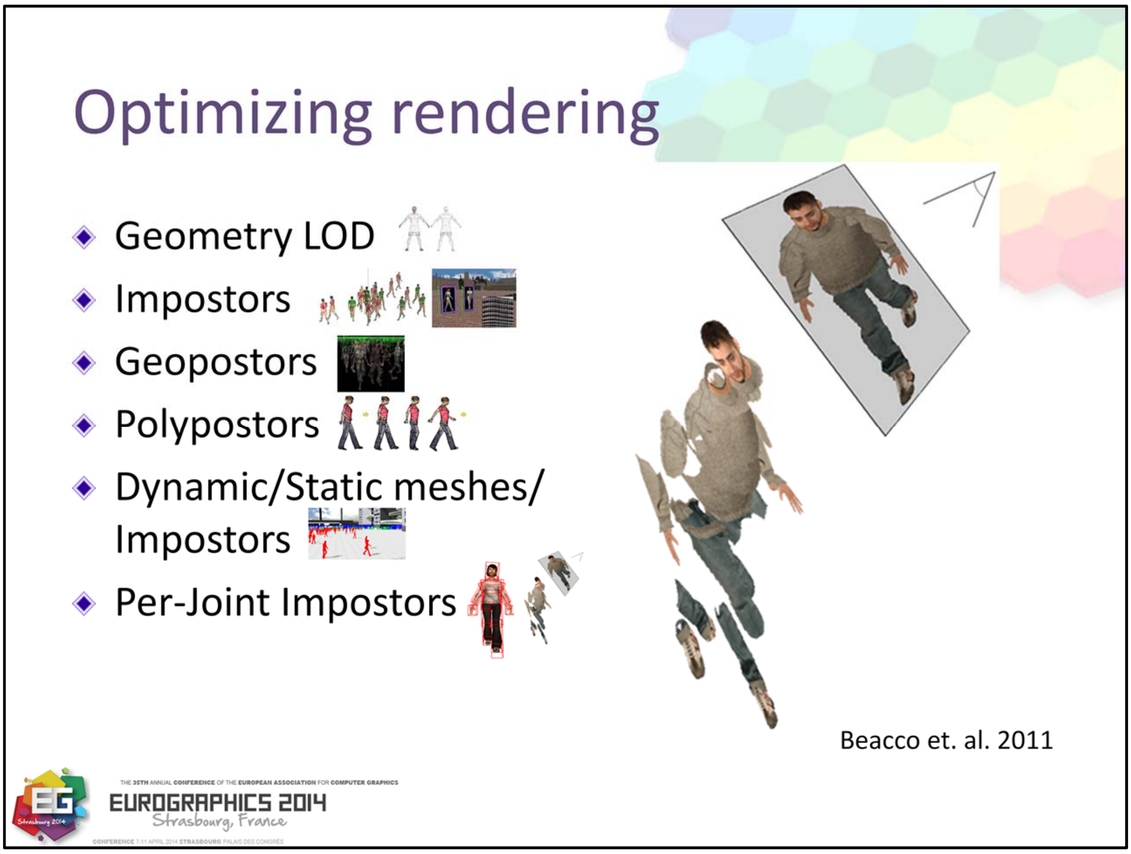
Advantage: Reduce the cost of animation updates as well as mesh deformations.

They manage to render up to 35,000 pedestrians in real-time

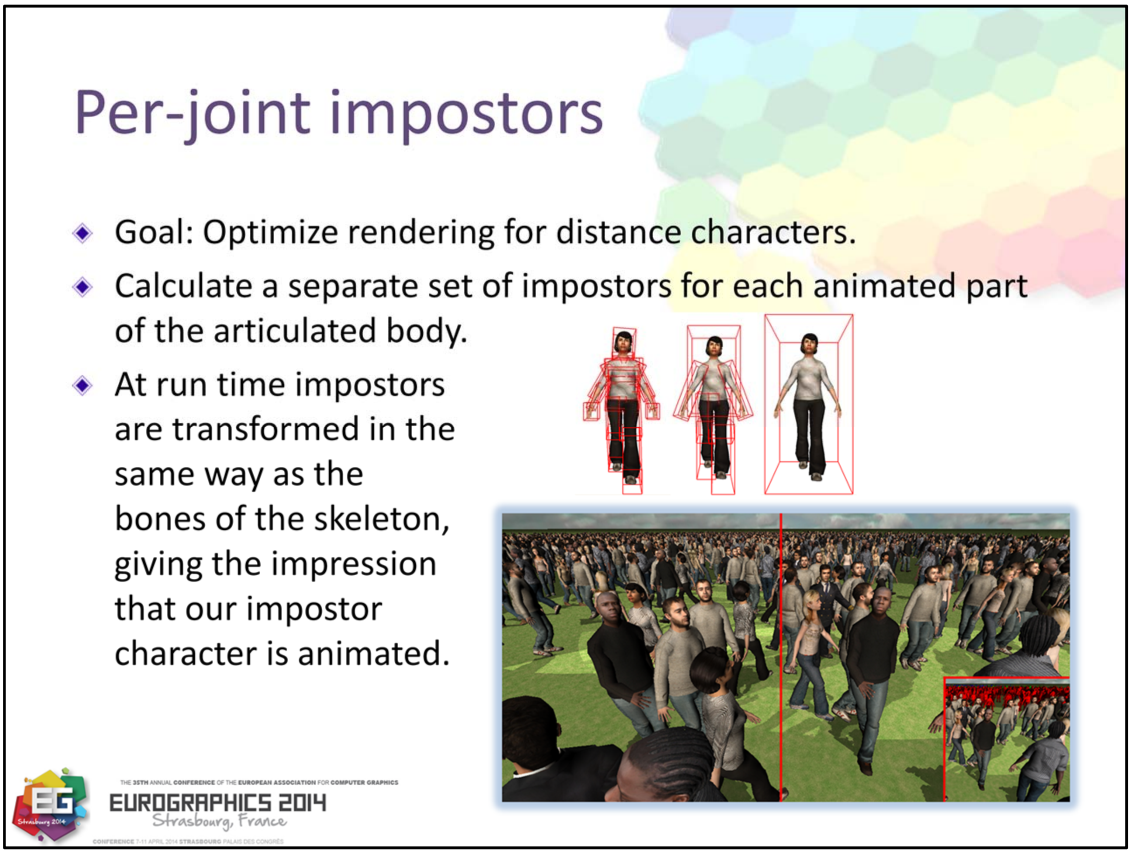


view-dependent per-joint rather than per character impostors. Characters are animated by applying the joint rotations directly to the impostors, instead of choosing a single impostor for the whole character from a set of predefined poses. This representation supports any arbitrary pose and thus the agent behavior is not constrained to a small collection of predefined clips.

We can use relief impostors where the geometry of each joint is captured with 6 relief impostors (one for each side of the cube)

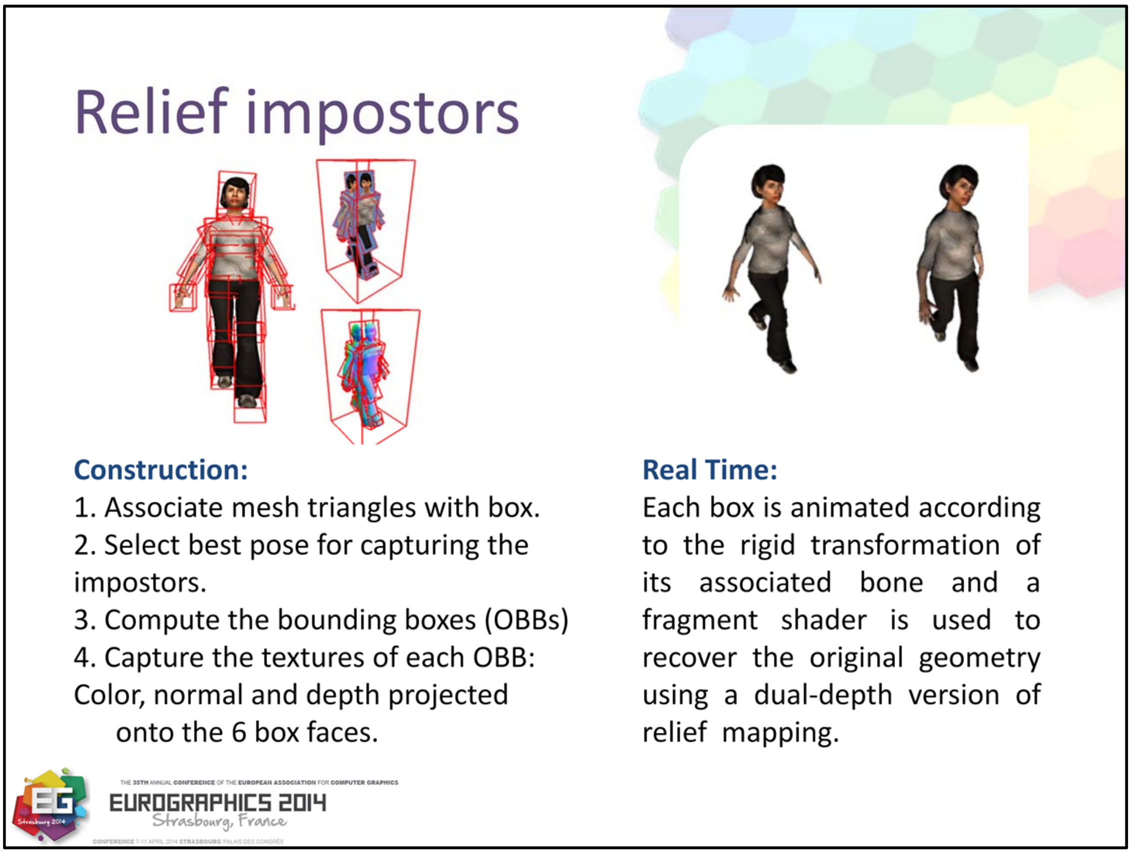


Or we can use traditional flat impostors. In this case, instead of using six orthogonal relief maps for each joint, which requires multiple dependent texture accesses per fragment, flat impostors are created by sampling each joint from multiple view directions.



The goal of this work is to optimize rendering for distance characters by having image-based performance, small GPU footprint and animation-independence, and low memory requirements.

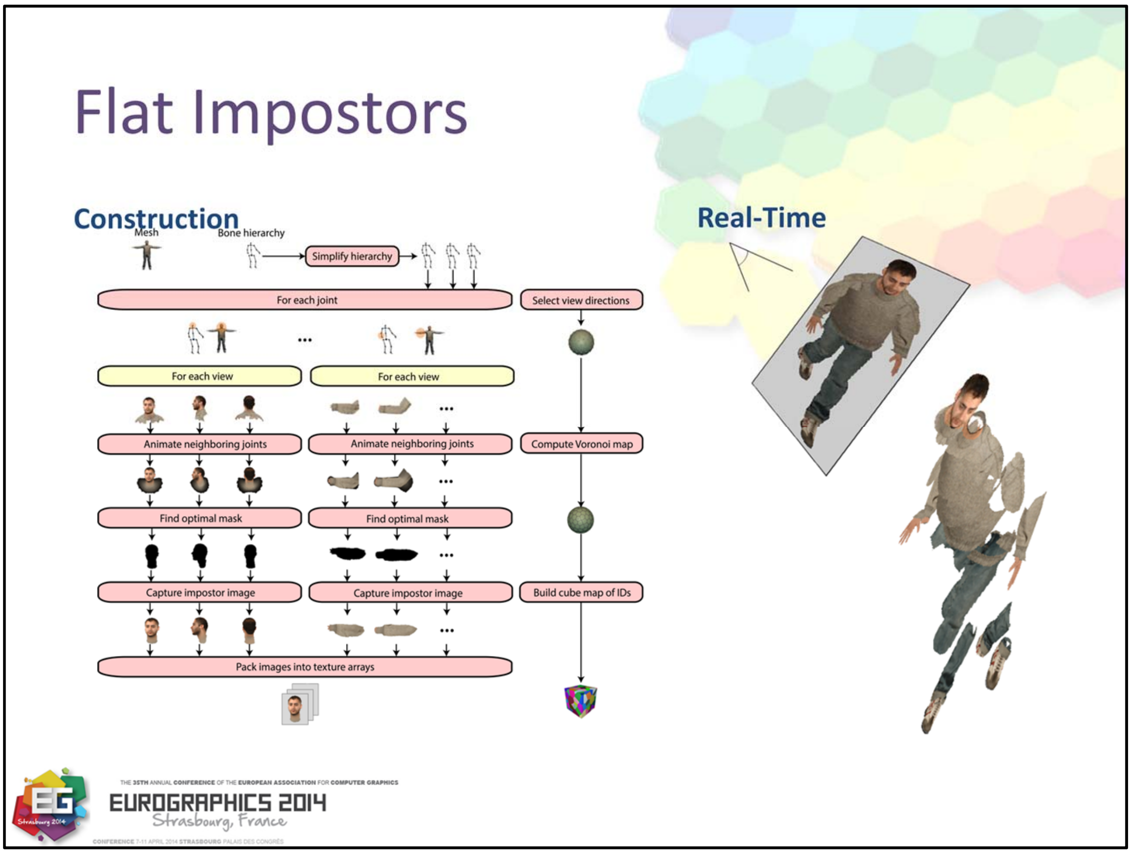
Characters are animated by applying the joint rotations directly to the impostors, instead of choosing a single impostor for the whole character from a set of predefined poses. This representation supports any arbitrary pose and thus the agent behavior is not constrained to a small collection of predefined clips.



Each character is encoded through a small collection of textured boxes storing color and depth values. At runtime, each box is animated according to the rigid transformation of its associated bone and a fragment shader is used to recover the original geometry using a dual-depth version of relief mapping. This compact representation is able to recover high-frequency surface details and reproduces view motion parallax effectively. It drastically reduces both the number of primitives being drawn and the number of bones influencing each primitive, at the expense of a very slight per-fragment overhead. Beyond a certain distance threshold, this compact representation is much faster to render than traditional level-of-detail triangle meshes.

User study results carried out demonstrated that replacing polygonal geometry by per-joint relief impostors produces negligible visual artifacts

Videos can be found in:
http://www.lsi.upc.edu/~npelechano/videos/MeshVsImpostors.mkv
http://www.lsi.upc.edu/~npelechano/videos/comparisonVideo.mkv



Instead of using six orthogonal relief maps for each joint, which requires multiple dependent texture accesses per fragment,
we use flat impostors created by sampling each joint from multiple view directions.

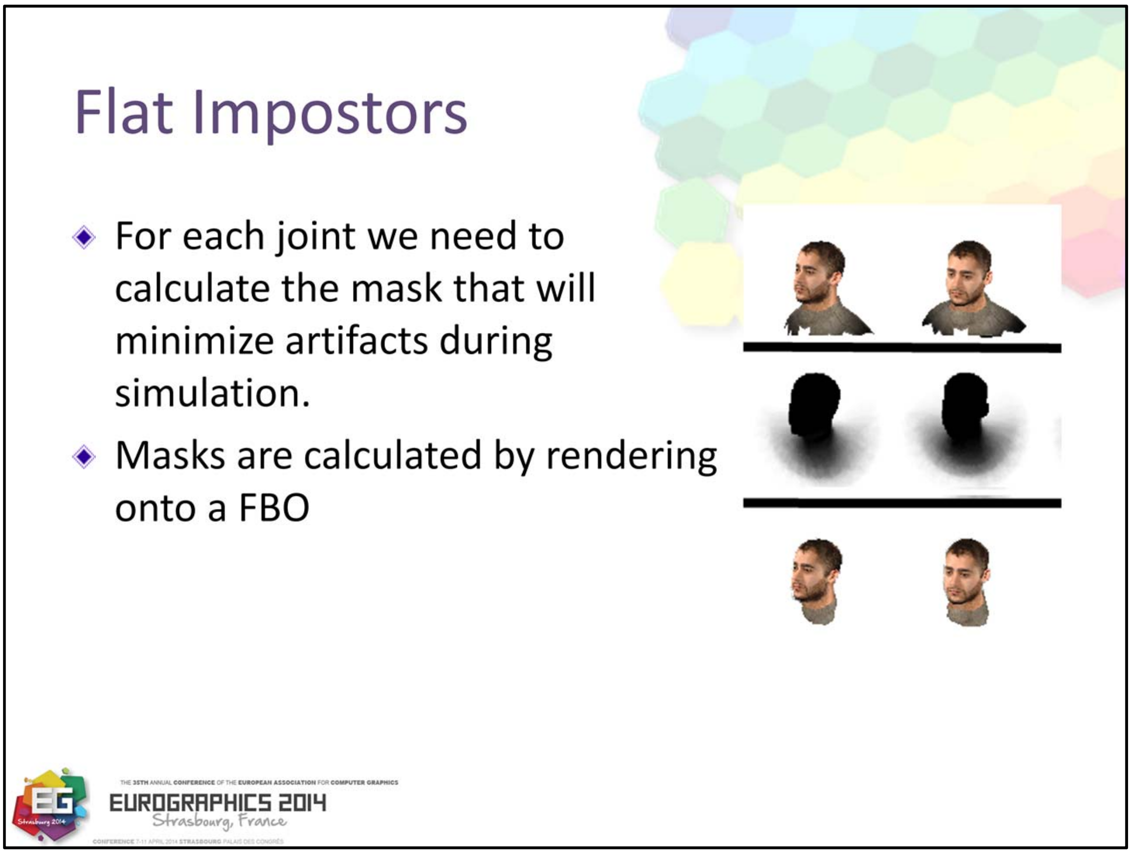
During pre-process we need to:

1. Compute a spherical Voronoi map for the desired number of samples
2. Build a cube map by projecting the Voronoi cells onto the cube faces.
3. Compute mask and apply to textures (masks consider how the geometry is affected by neighboring bones).

During real time rendering:

1. Animate character and cube maps.
2. Retrieve the fragment color with a single texture lookup. This process is one order of magnitude faster than relief mapping.

Videos can be found in:
http://www.lsi.upc.edu/~npelechano/videos/PerJointImpostors.mkv

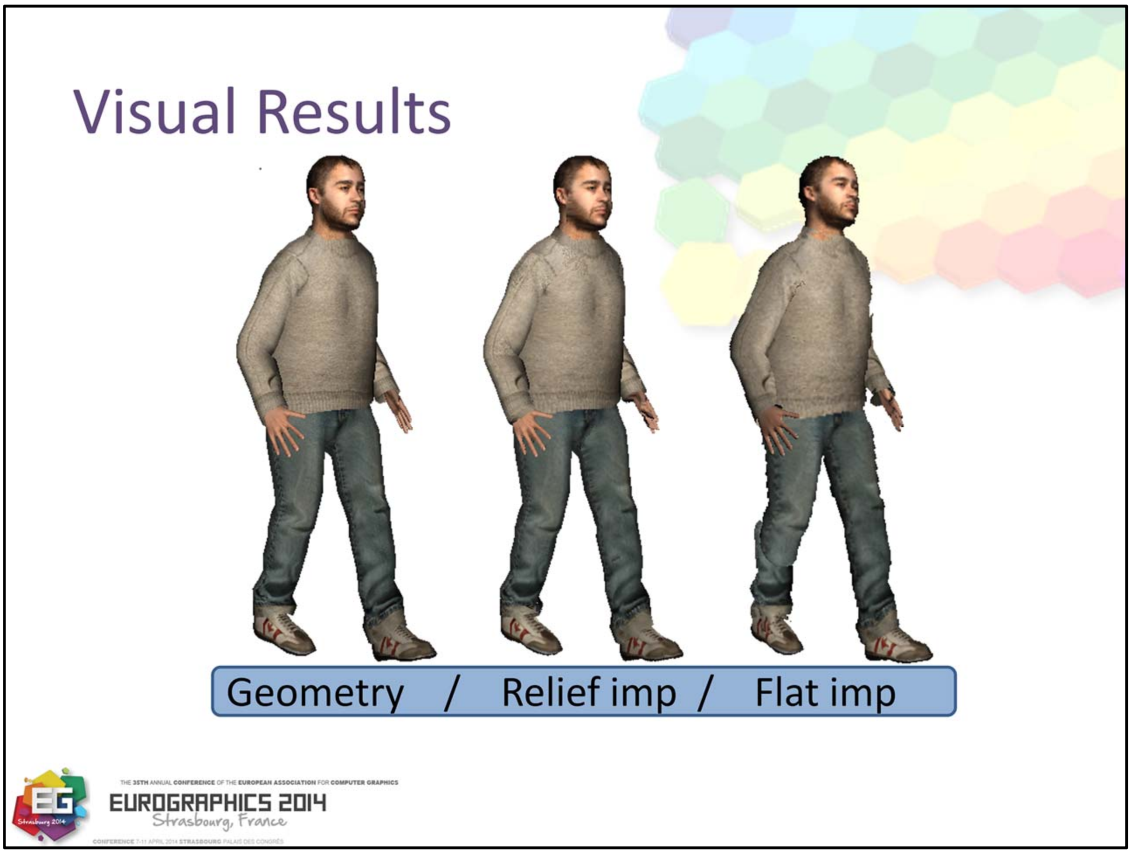


Since our impostors are intended to be valid for any pose, a key issue is to properly define which part of the geometry influenced by each joint must be represented as opaque pixels in the corresponding impostor (mask).

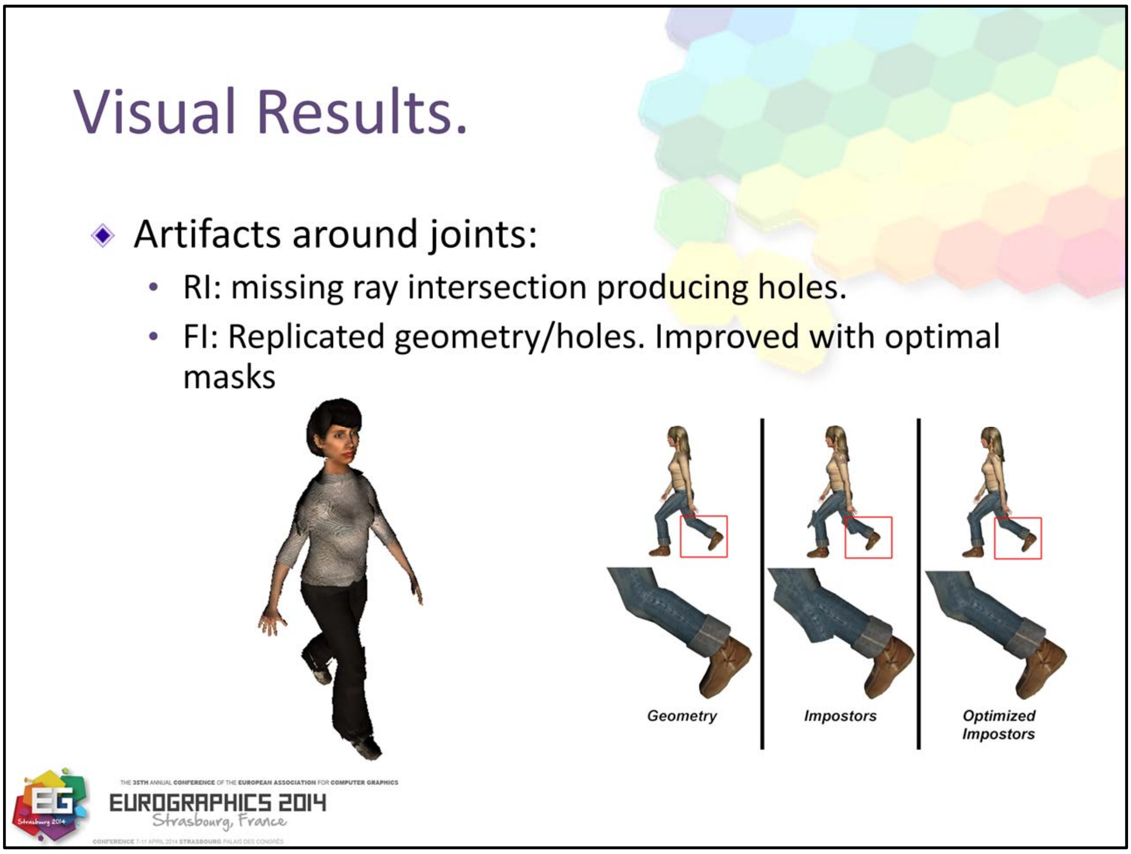
We provide an efficient algorithm for computing optimized masks which considers how the geometry of each bone is affected by the transformation of neighboring joints.

Mask are calculated by rendering onto a frame buffer object which plays the role of the accumulation buffer, with a set of feasible positions for the corresponding joint. For each position, the resulting image is generated by applying linear blend skinning to the set of triangles influenced by the joint. After applying a threshold over the values stored in the FBO, we obtain the masks . Notice that masks exhibit rounded boundaries near joints due to the blending effect of neighboring joint rotations on the deformed geometry. These shapes are somewhat similar to the rounded joints used in traditional art mannequins and articulated figures.

This approach clearly outperforms competing animation-independent approaches for crowd rendering, being over 5 times faster than per-joint relief impostors .

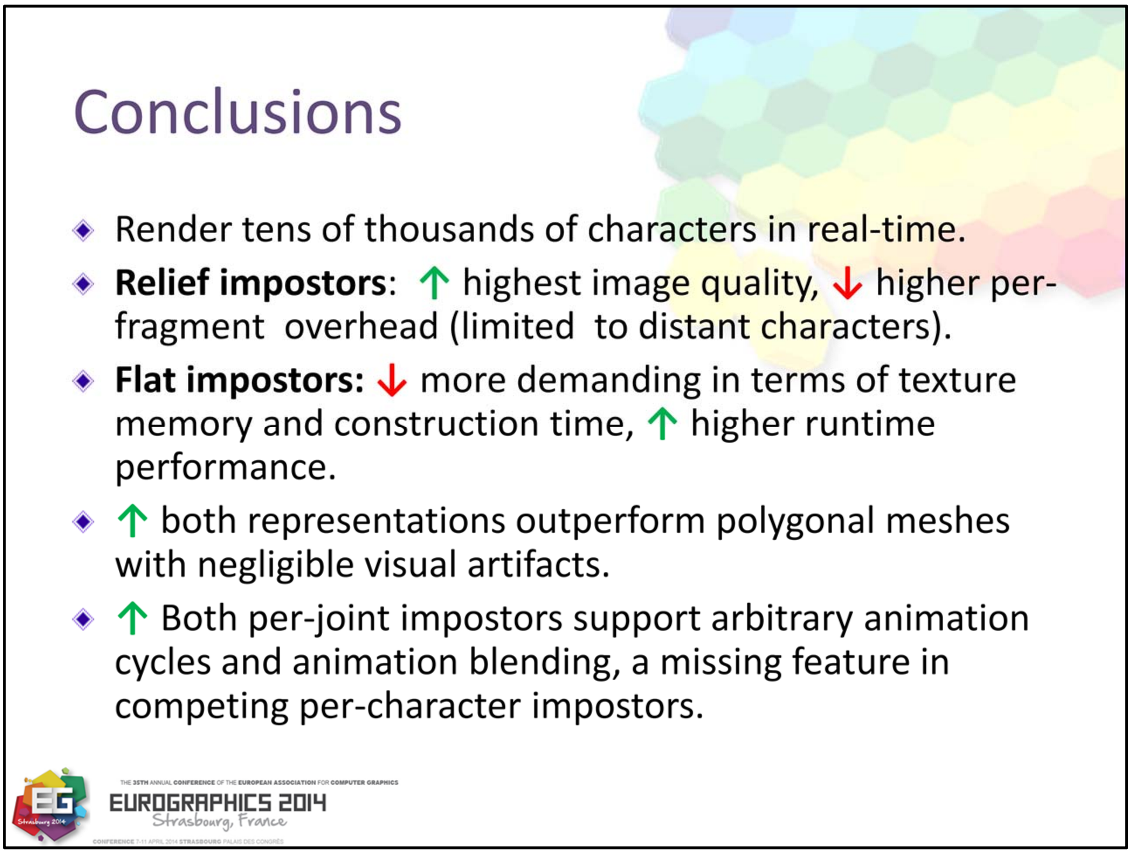


Here we can see a close up of the same character rendered with geometry, relief impostors and flat impostors. As we can see even for short distances we achieve highly realistic rendering. Artifacts are most noticeable around joints, since we apply rigid transformations.



In the case of relief impostors, most artifacts come from missing ray intersections which result in holes.

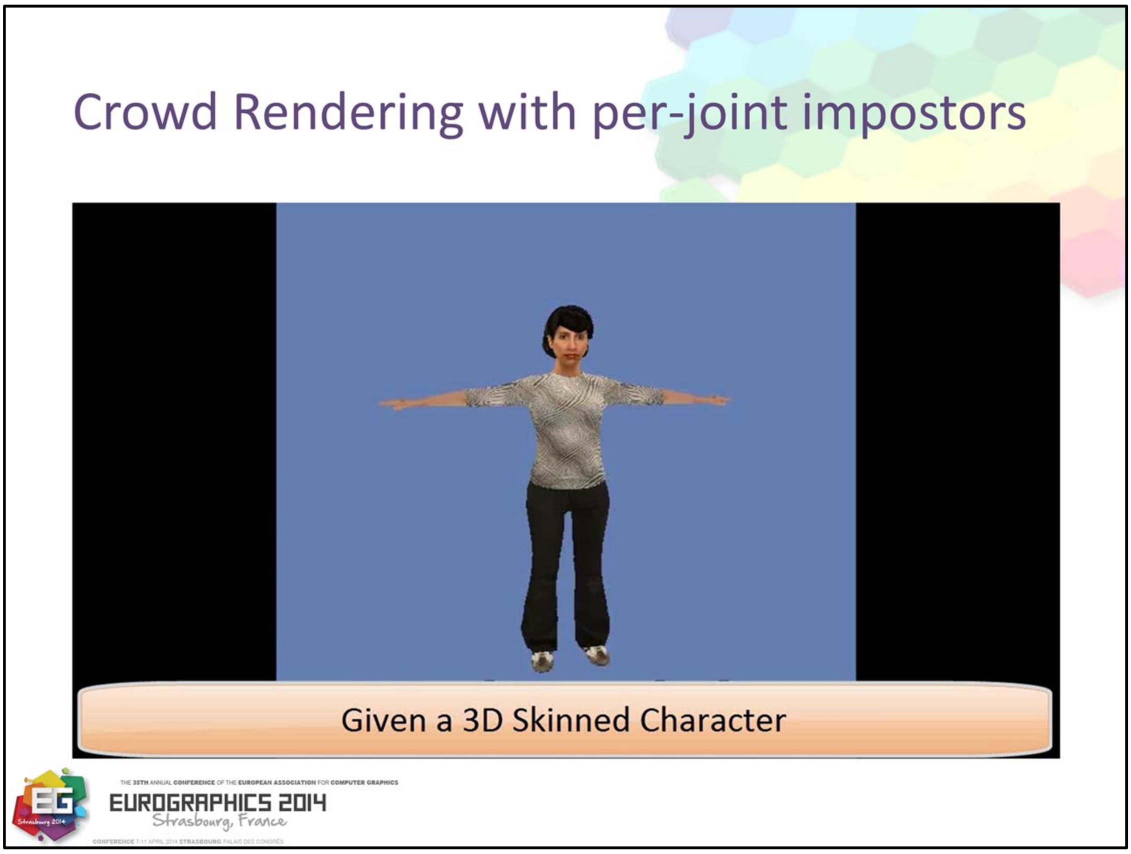
In the case of flat impostors, artifacts depend on the geometry rendered to create each of impostors per joint and view direction. Notice that with our optimized solution (calculating the masks as indicated earlier in the tutorial) we drastically reduce those artifacts.



Per-joint impostors allows us to render tens of thousands of characters in real-time. Encoding per-joint geometry and appearance
with relief maps provides the highest image quality at the expense of a higher per-fragment overhead, which in practice limits their applicability to distant characters.

View dependent flat impostors are more demanding in terms of texture memory and construction time, but provide the highest runtime performance even for close-up characters. With properly chosen switch distances, both representations outperform polygonal meshes with negligible visual artifacts.

Regardless of the particular encoding, per-joint impostors support arbitrary animation cycles and animation blending, a missing feature in competing per-character impostors.



Videos can be found in:
http://www.lsi.upc.edu/~npelechano/videos/fastforwardVideo.wmv

Acknowledgements

Collaborators

- Alejandro Beacco
- Ramon Oliva
- Carlos Andujar
- Bernhard Spanlang

Sponsors

- Spanish Government grant TIN2010-20590-C01-01
- TRAVERSE ERC advanced grant 227985

THE 35TH ANNUAL CONFERENCE OF THE EUROPEAN ASSOCIATION FOR COMPUTER GRAPHICS
EUROGRAPHICS 2014
Strasbourg, France
CONFERENCE 7-11 APRIL 2014 STRASBOURG PALAIS DES CONGRÈS

THE 35TH ANNUAL CONFERENCE OF THE EUROPEAN ASSOCIATION FOR COMPUTER GRAPHICS

EUROGRAPHICS 2014

Strasbourg, France

CONFERENCE 7-11 APRIL 2014 STRASBOURG PALAIS DES CONGRÈS

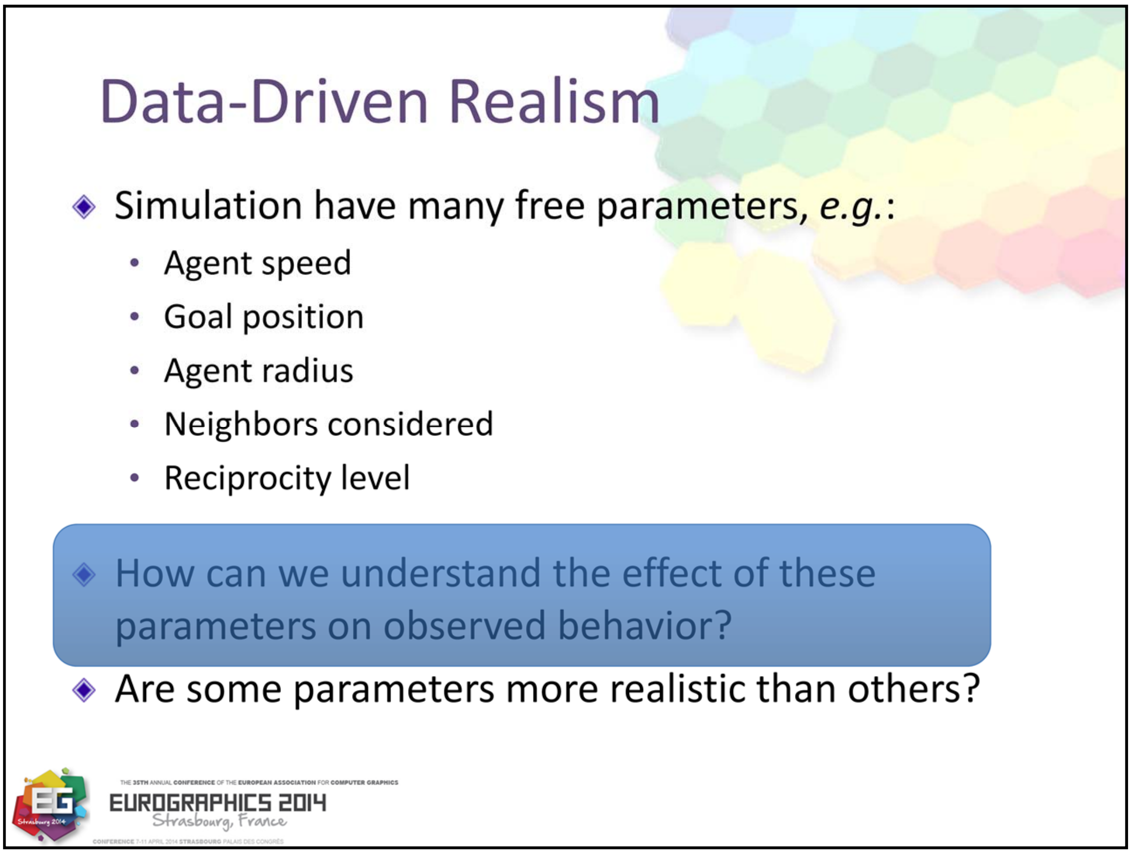
Realism

Statistical Crowd Methods

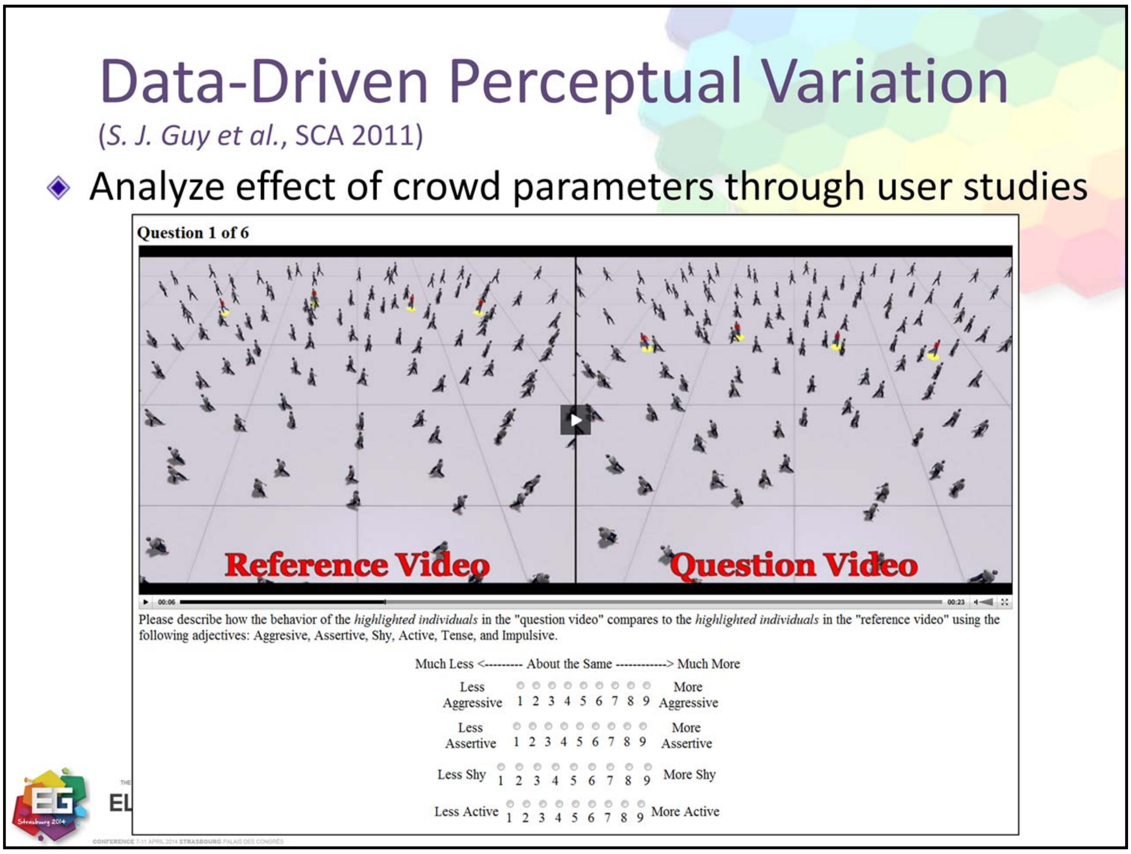
For Enhancing and Evaluating Crowd Realism

Stephen J. Guy

University of Minnesota

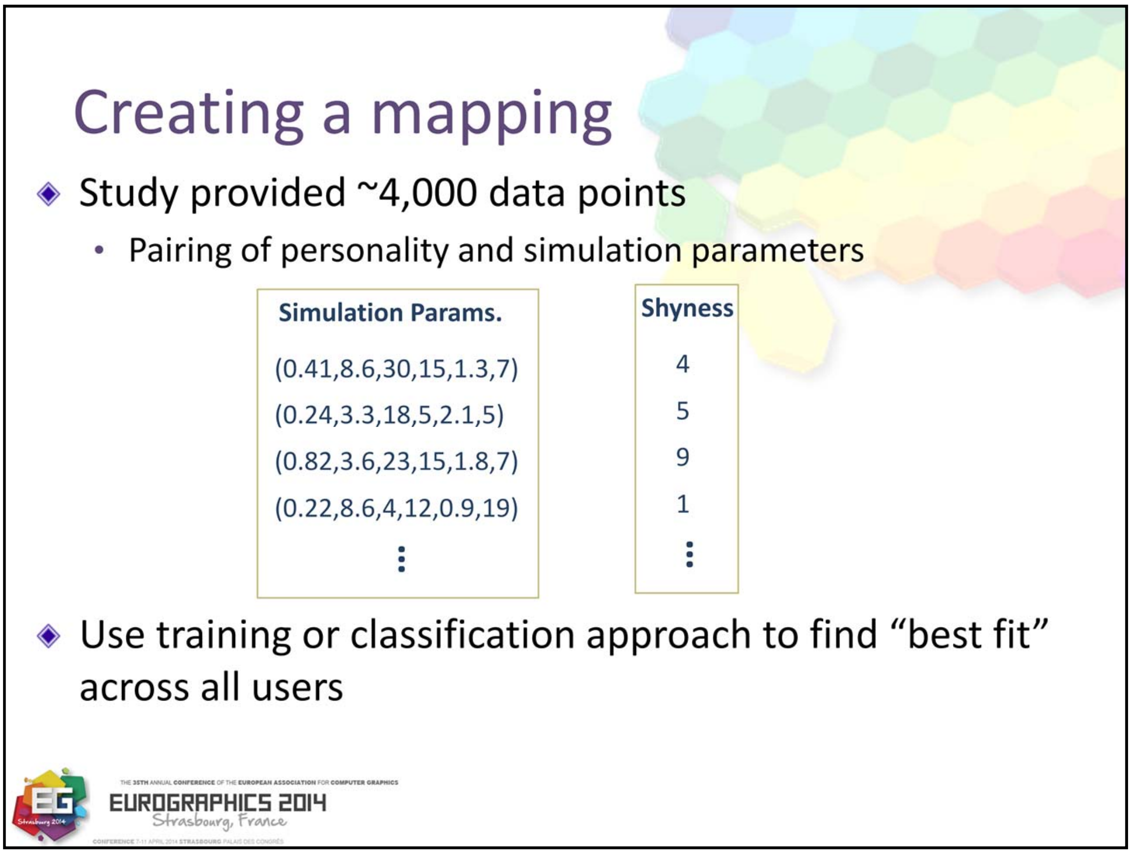


Free parameters in a simulation have a large impact in determining how realistic the resulting motion will be. Generating the desired crowd behaviors often has as much to do with simulation parameters as it does with choosing the correct simulation method. Statistical analysis techniques can help us better understand the effects changing these parameters have on a simulation, and can help us choose parameters which lead to the most realistic results.



Paper and videos available at: http://gamma.cs.unc.edu/personality/

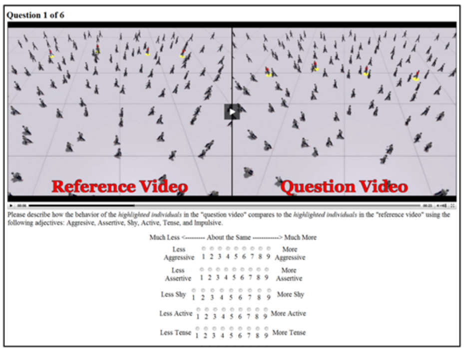
In Guy et al. 2011, we used a comparison-based user study to gather data on the perceptual impact of varying different parameters. Users were shown a video with randomly chosen simulation parameters side-by-side with a video showing baseline crowd behaviors. The users were asked to rate the randomly generated video in terms of the common psychological descriptors of shy, tense, aggressive, active, impulsive and assertive.



The result of the study was a large set of data points, sampling the 6D->6D mapping between the six various simulation parameters used and the six psychological terms studied. We then fit a linear model across the data from all users in the study.

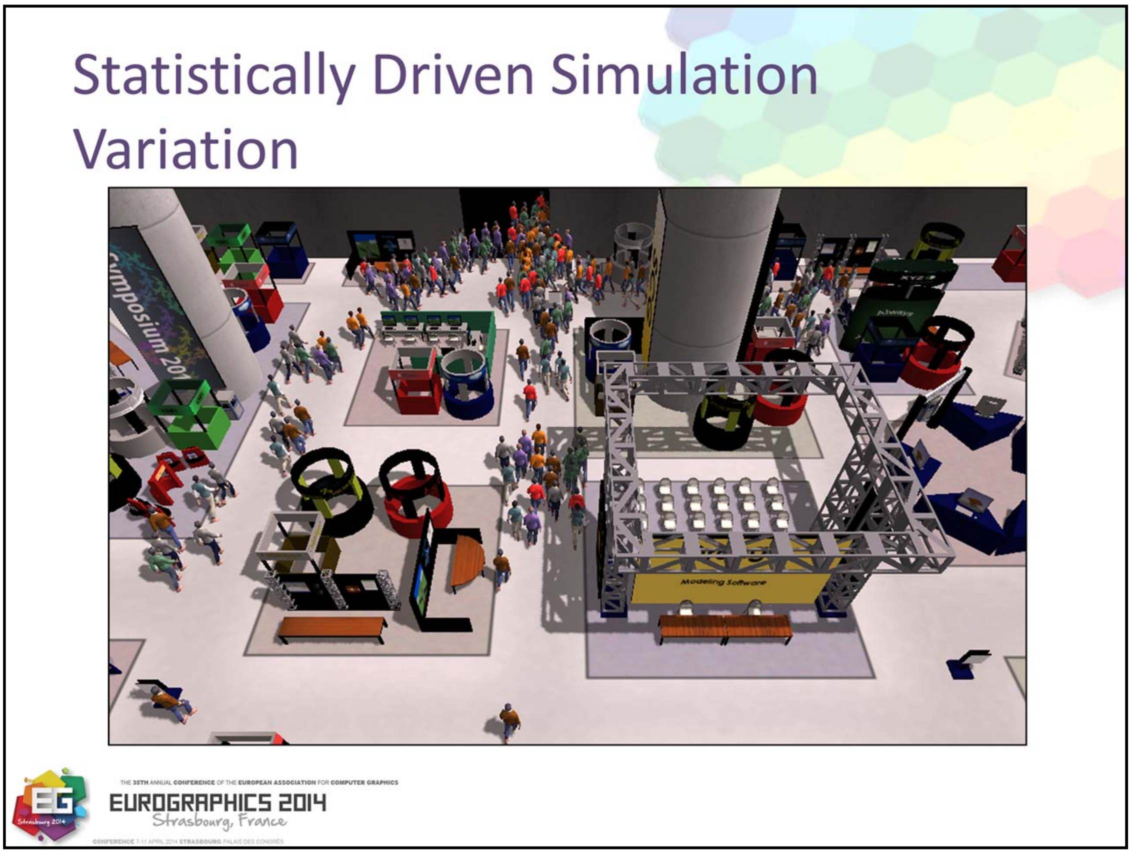
Study Results

- Changing parameters changes observed behavior
  - Analysis of user studies can help understand connection
  - Example Learned correlations:
    - Fewer neighbors + higher speed → aggressive
    - Larger radius → shy
  - The complete (6x6) model is provide in the paper

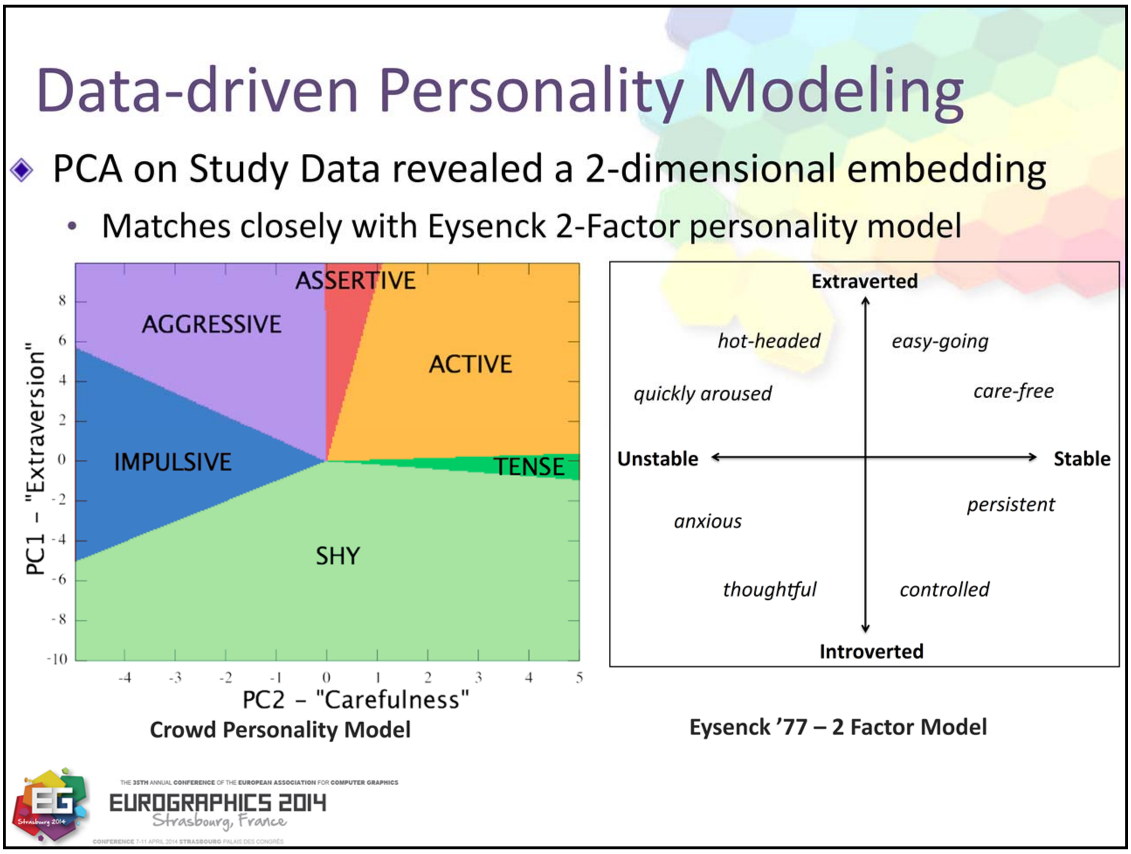


EUROGRAPHICS 2014
Strasbourg, France

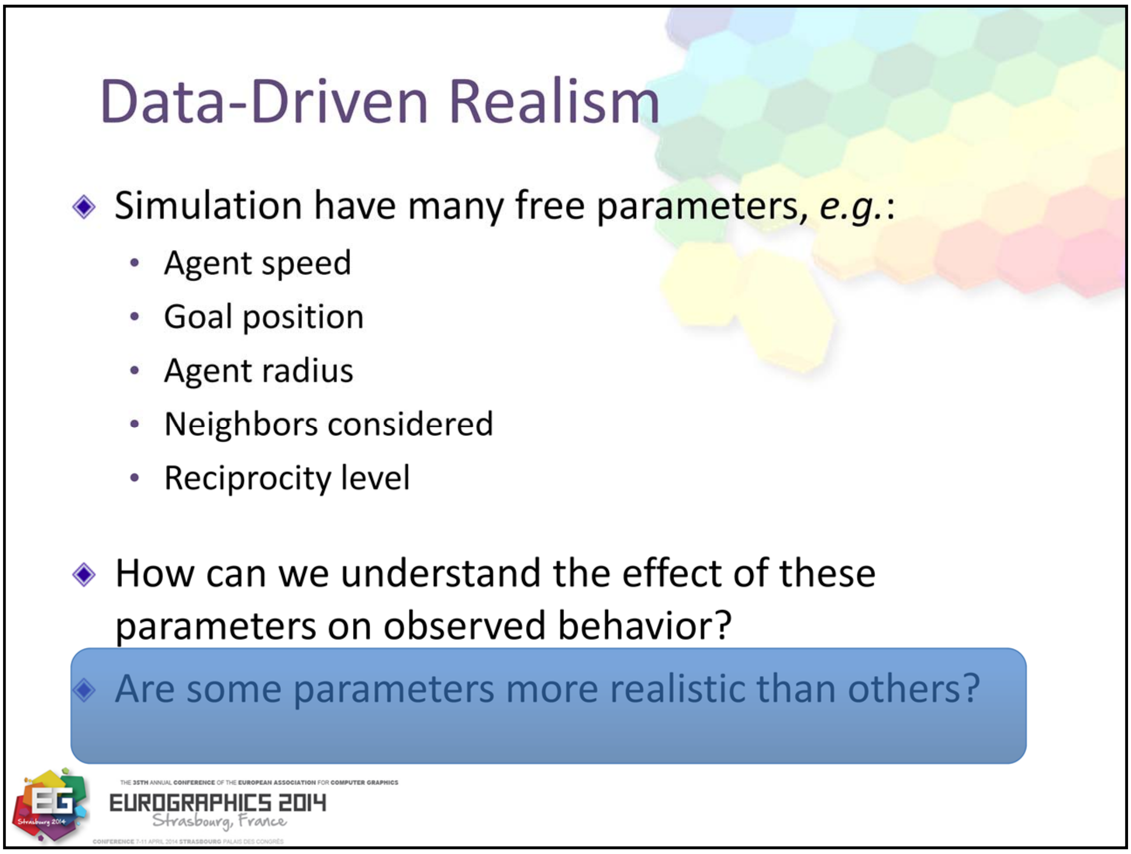
Several high-level trends in the data are captured by the model. For example, the size of an agent's simulation radius (i.e., personal space) is positively correlated with how shy the agents appear.



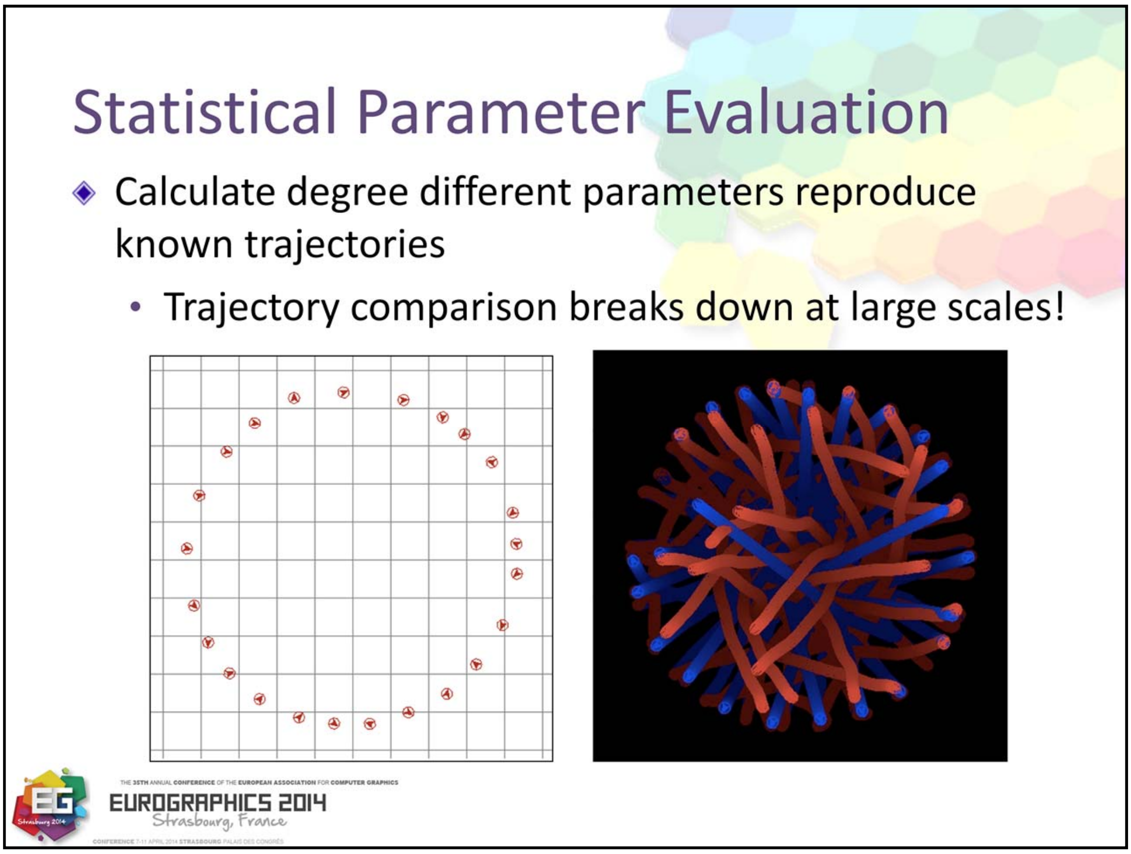
This video shows a simulation of agents exiting a large room while displaying various personalities. Overall, our experiments showed users have a nearly 90% success rate when choosing which of two videos mapped to two randomly chosen adjectives. The success rate rose to 98% when similar adjectives were grouped together (Aggressive/Impulsive, Assertive/Active, Tense/Shy).



While the learning model was six dimensions, techniques such as PCA can be used to find good low-dimensional projections of the model. Here we see a 2D version of the model (which accounts for ~90% of the variation of the full 6D model), with the most dominate personality trait for each region of the model highlighted. Interestingly, this 2D model matches fairly well with established personality models from the psychology community.

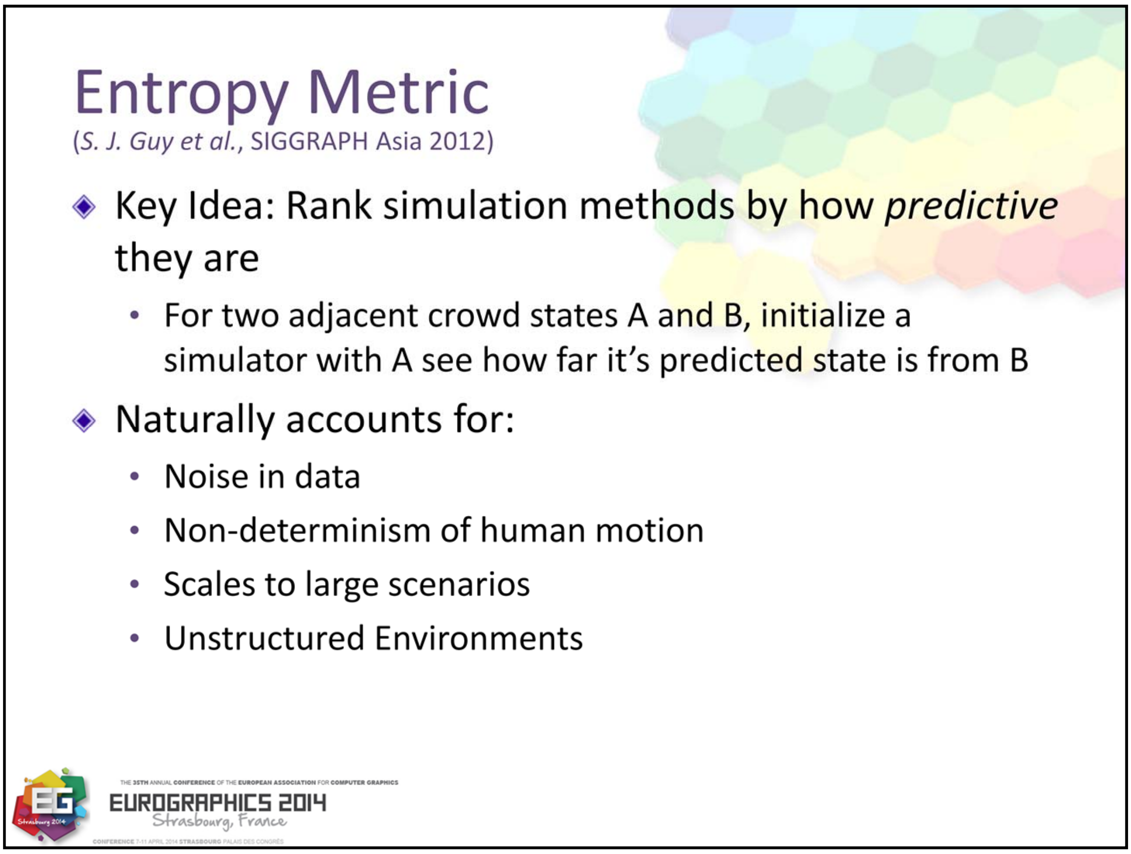


Statistical analysis approaches can also help determine the most realistic parameters to use in a variety of scenarios.

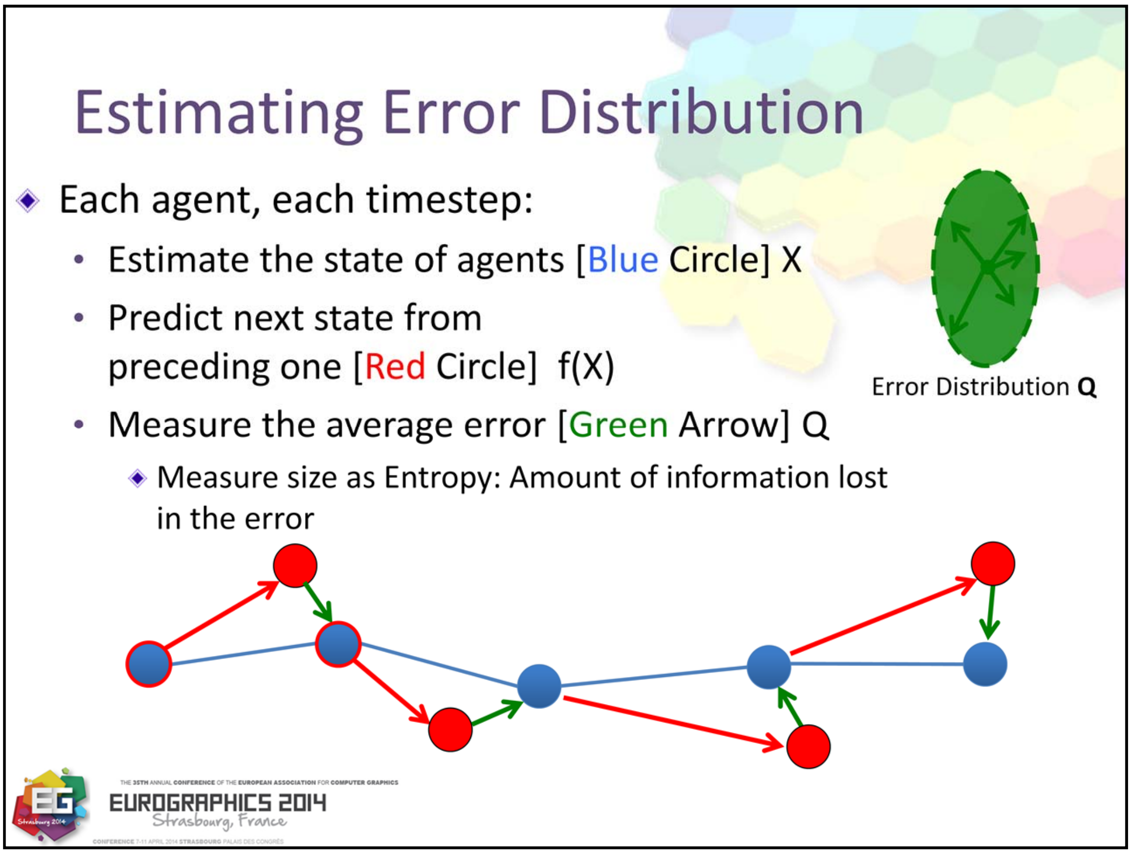


Paper and Videos available at: http://gamma.cs.unc.edu/Entropy/

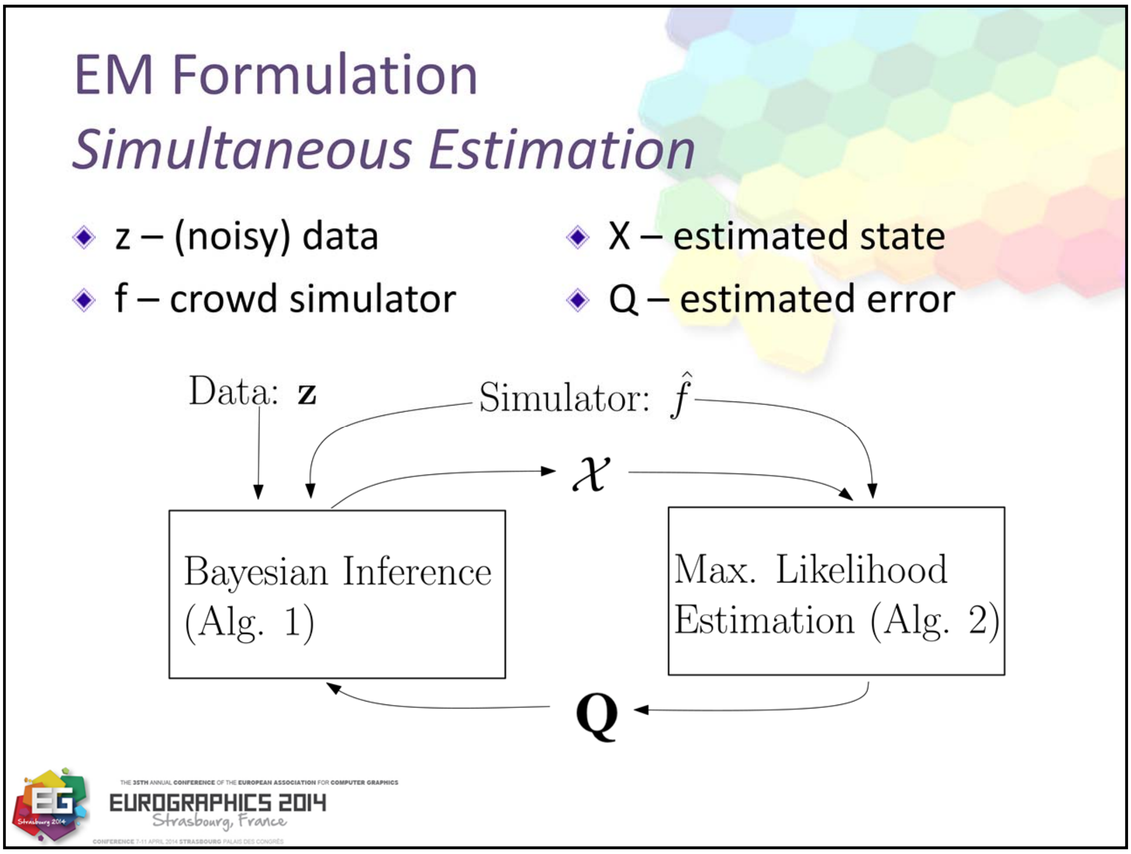
At a high level, we'd like to test different simulation parameters to see which ones do the best job of producing paths that match real-world data. The main issue preventing such as approach is that direct comparison between trajectories is uninformative in complex scenarios with dozens of interacting agents.



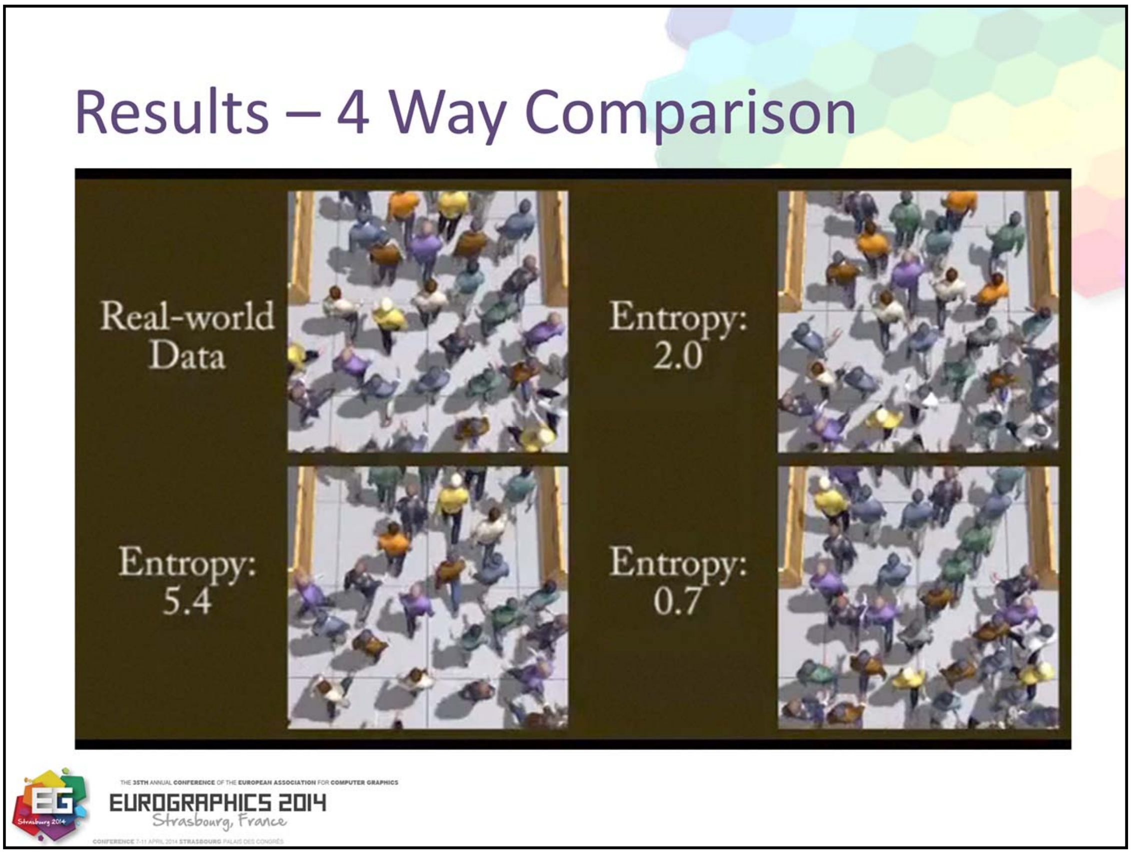
Guy et al. 2012 presents a robust method for comparing simulations to data known as the Entropy Metric. Rather than comparing the final results of a simulation versus the data, the Entropy metric works on a per-timestep basis, ranking different simulation methods by how well they capture the decisions made by the individuals in the data on a timestep-by-stepstep basis. By resetting the simulation each timestep, we can remove issues caused by small errors early in the simulation becoming large errors later on.



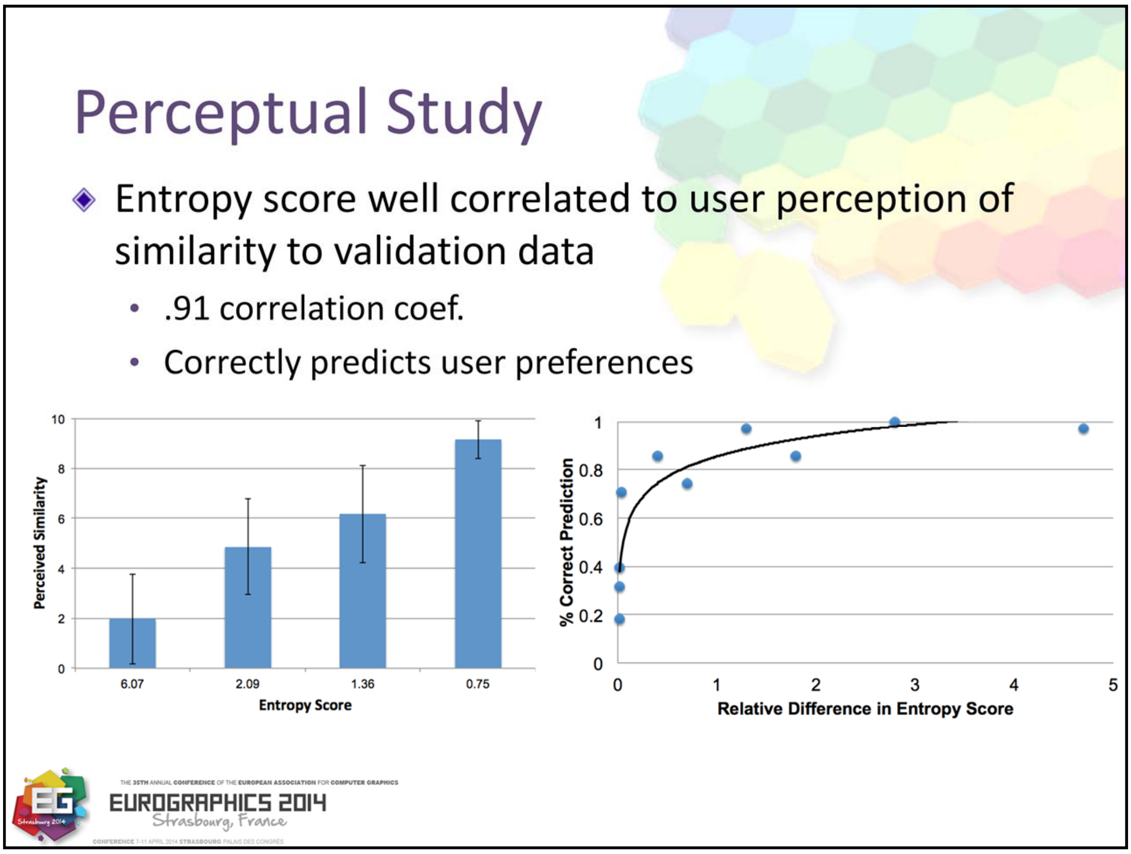
At a high level, there are two tasks that must be accomplished for each timestep in order to compute the entropy metric. First, we must estimate the true state of each agent at every timestep (blue dots). These estimates must include the entire state information needed to compute the next step in the simulation (i.e., position, velocity, etc.). Once these estimates for the agent’s states are made, the second step is using a simulator to predict the next step given the agent states at the previous step. The size of the distribution of the differences between predicted and actual states (as measured by it’s Entropy) forms the metric, with smaller error distributions corresponding to more accurate simulations.



Bayesian inference techniques such as Extended Kalman Filters or Ensemble Kalman Smoothers, can be used to produce hiqh-quality estimates of the true state of the agents. Unfortunately, using these techniques, requires already knowing the error of the crowd simulation method – even though this is the unknown quantity we sought to compute in the first place! To resolve this issue, we use an iterative approach, in the style of the EM algorithm, where initial values are guessed at, and estimates for the two distributions are computed iteratively until the estimates converge.



The entropy metric can capture qualitative similarities between resulting simulations and real-world data, even when the quantitative positions produced are very different.



Subsequent user studies suggest that simulation methods with a lower entropy score were more likely to be picked by users as “matching the real-world behavior”. In fact, the larger the difference in entropy scores between two simulators, the more accurate the method was at predicting a user’s preference.

THE 35TH ANNUAL CONFERENCE OF THE EUROPEAN ASSOCIATION FOR COMPUTER GRAPHICS

EUROGRAPHICS 2014

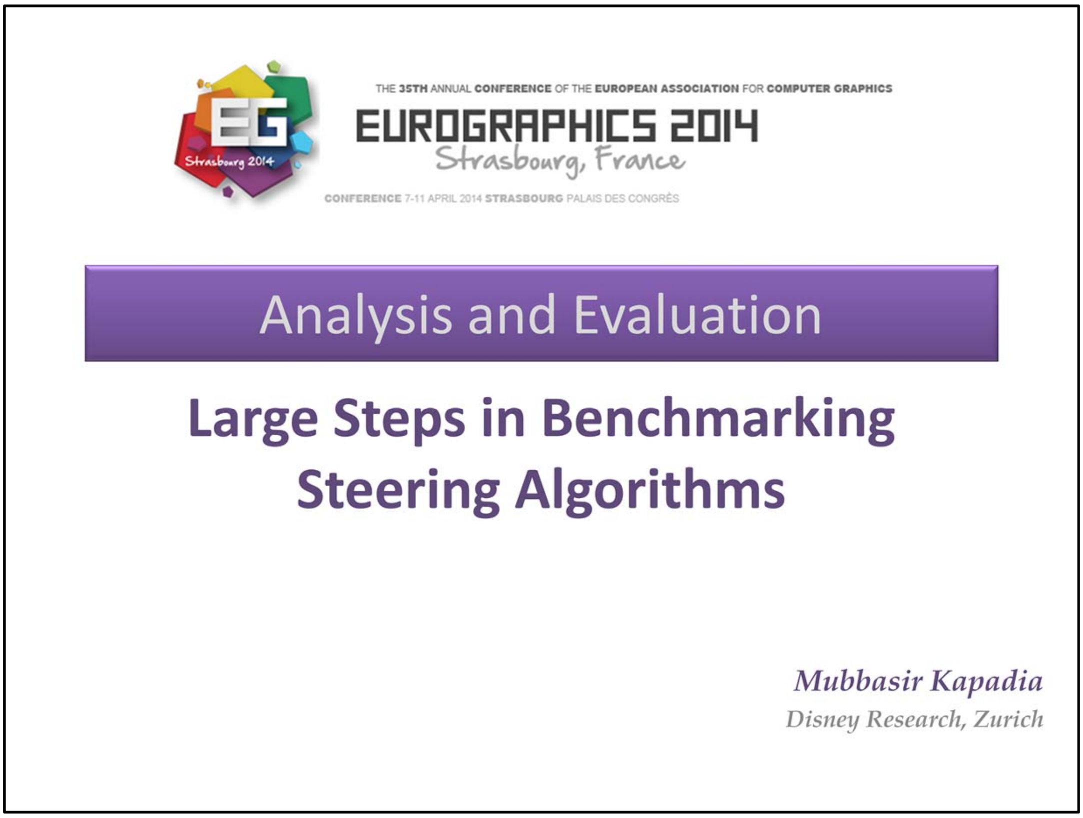
Strasbourg, France

CONFERENCE 7-11 APRIL 2014 STRASBOURG PALAIS DES CONGRÈS

Simulating Heterogeneous Crowds

ANALYSIS AND EVALUATION

Mubbasir Kapadia
Disney Research, Zurich
Nuria Pelechano
Universitat Politècnica de Catalunya
Yiorgos Chrysanthou
University of Cyprus

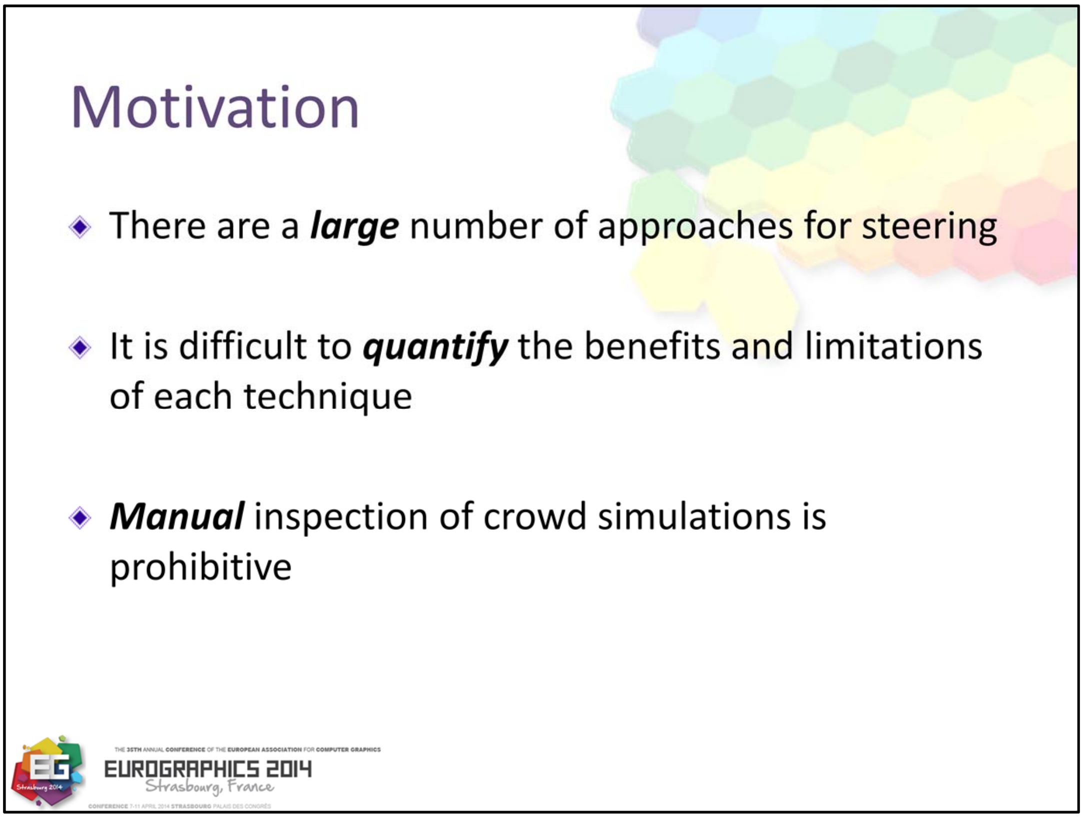


Speaker: Mubbasir Kapadia, Associate Research Scientist, Disney Research Zurich

Bio: Mubbasir Kapadia is an Associate Research Scientist at Disney Research Zurich. Previously, he was a postdoctoral researcher and Assistant Director at the Center for Human Modeling and Simulation at University of Pennsylvania. He was the project lead on the United States Army Research Laboratory (ARL) funded project Robotics Collaborative Technology Alliance (RCTA). He received his PhD in Computer Science at University of California, Los Angeles.Kapadia's research aims to develop integrated solutions for full-body character animation, planning based control, behavior authoring, and statistical analysis of autonomous virtual human simulations. The far-reaching goal is to provide functional, purposeful embodied virtual humans, that act and interact in meaningful ways to simulate complex, dynamic, narrative-driven, interactive virtual worlds.

Related Publication(s):
Scenario Space: Characterizing Coverage, Quality, and Failure of Steering Algorithms. Mubbasir Kapadia, Matthew Wang, Shawn Singh, Glenn Reinman and Petros Faloutsos. *ACM SIGGRAPH Symposium on Computer Animation*, August 2011

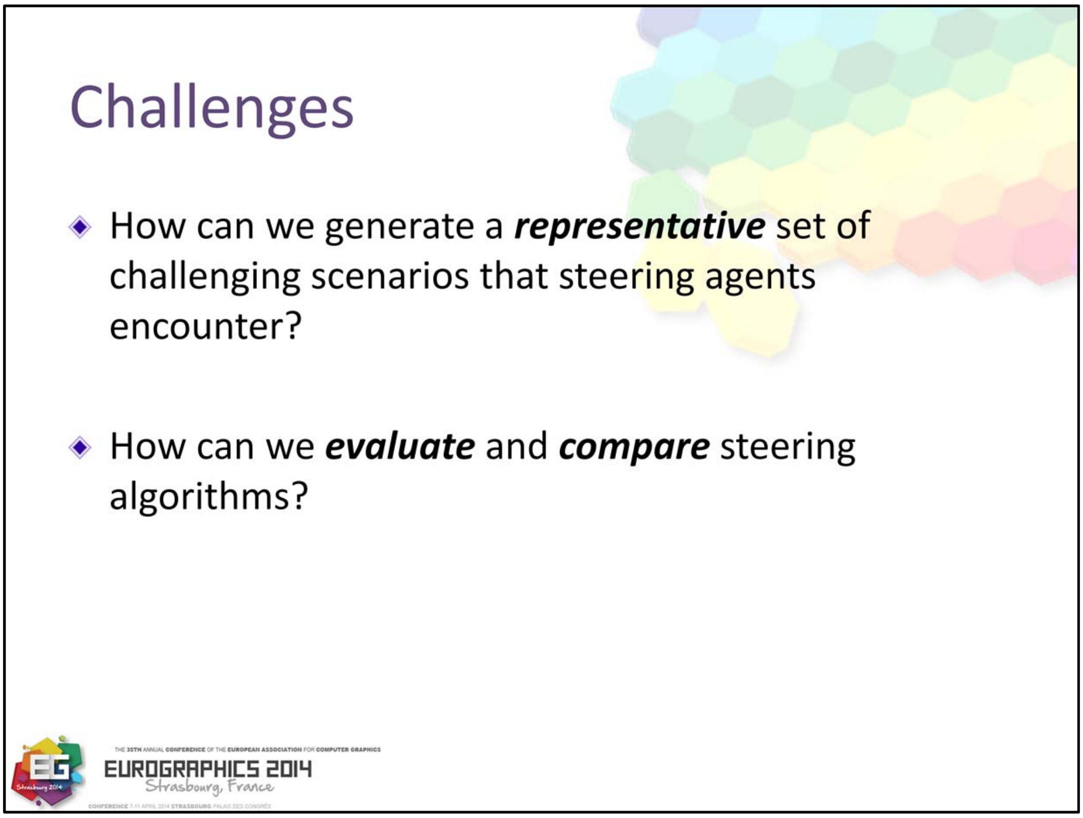
Improved Benchmarking for Steering Algorithms. Mubbasir Kapadia, Matthew Wang, Glenn Reinman and Petros Faloutsos . *4th International Conference on Motion in Games*, Novemeber 2011



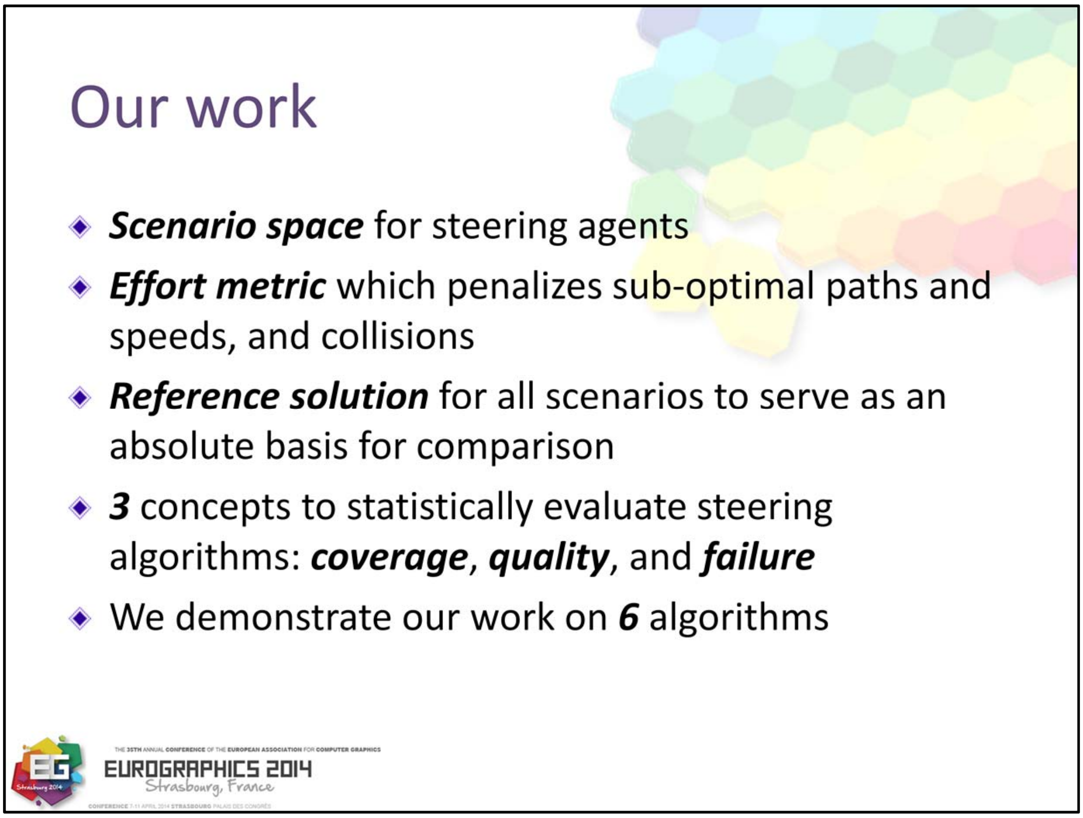
There exist a wide variety of approaches for simulating steering agents. However, it is extremely difficult to evaluate the performance of different steering approaches and to be able to compare the benefits and limitations of each.

There is no AUTOMATED method for comprehensively evaluating steering algorithms .

Project webpage: http://people.inf.ethz.ch/kapadiam/projects-benchmarking.html



There are two fundamental challenges we face in the automated analysis of steering algorithms. First, we need the ability to provide a representative set of challenging scenarios that agents encounter in dynamic worlds. Second, we need a measure of performance of an algorithm in this space of scenarios



Our work makes the following contributions:
We define the space of all possible scenarios that steering agents encounter in dynamic worlds and provide a method for sampling these scenarios in a tractable fashion

We introduce an effort metric which takes into account time, path length, and collisions in one single quantity

We use an offline space-time planner to provide a reference solution for all scenarios to serve as an absolute basis of comparison

We define coverage, quality, and failure for steering algorithms

Finally, we demonstrate the effectiveness of our method by evaluating 6 state of the art published steering approaches

SteerBench Analysis

Benchmark	Ego	PPR	RVO	CC	Footstep	React
Simple-1	251.6	266.3	254.8	263.2	235.7	254.8
Crossing	261.3	261.4	259.6	271.2	220.0	259.4
Oncoming	269.2	270.0	265.4	275.1	244.5	268.4
Cut-across	538.5	571(.6)	545.8	881(2.6)	477.3	551(.3)
Surprise	424.6(1)	484.9(2)	408(.5)	566.1(2)	470(1)	403.9
Overtake	290.8	288.2	306.4	277.4	266.4	281.2
Confusion	268.6	267.1	269.2	716(1.5)	234.5	265.6
Frogger	241.8	240.3	241.6	443(1.6)	216.5	241.4
Wall-Squeeze	Fail	Fail	434.2(2)	567.1(2)	287.8	Fail
Doorway	Fail	Fail	Fail	Fail	Fail	Fail

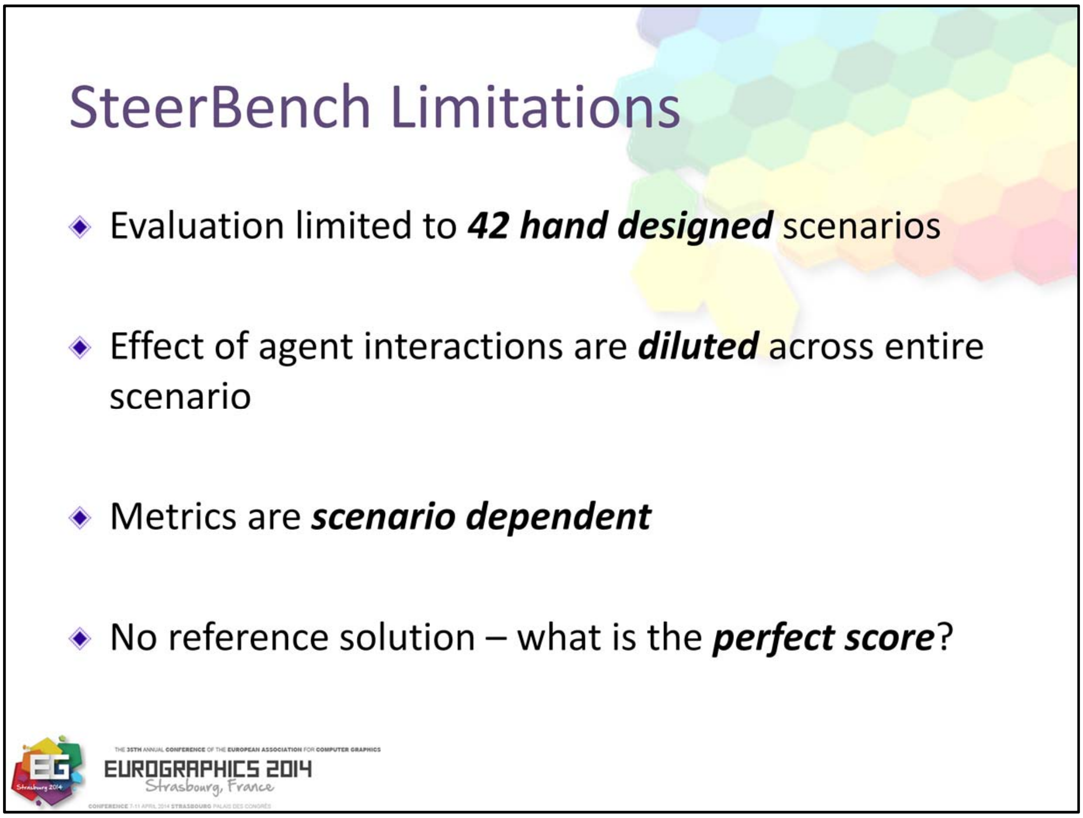
SteerBench: A Benchmark Suite for Evaluating Steering Behaviors. Shawn Singh, Mubbasir Kapadia, Glenn Reinman and Petros Faloutsos Computer Animation and Virtual Worlds, 2009

EUROGRAPHICS 2014 Strasbourg, France

We evaluate the 6 steering algorithms using SteerBench. All algortihms including Reactive can solve at least 36/42 scenarios provided by SteerBench. The reference space-time planner can solve all 42 test cases. The values of the scores range from 100 to 500 depending on the length and difficulty of the scenario.

The Surprise scenarios challenge agents to suddenly react to crossing threats in narrow corridors. However, the effect of the interesting interaction between agents on the overall score is diluted due to the length of the scenario.

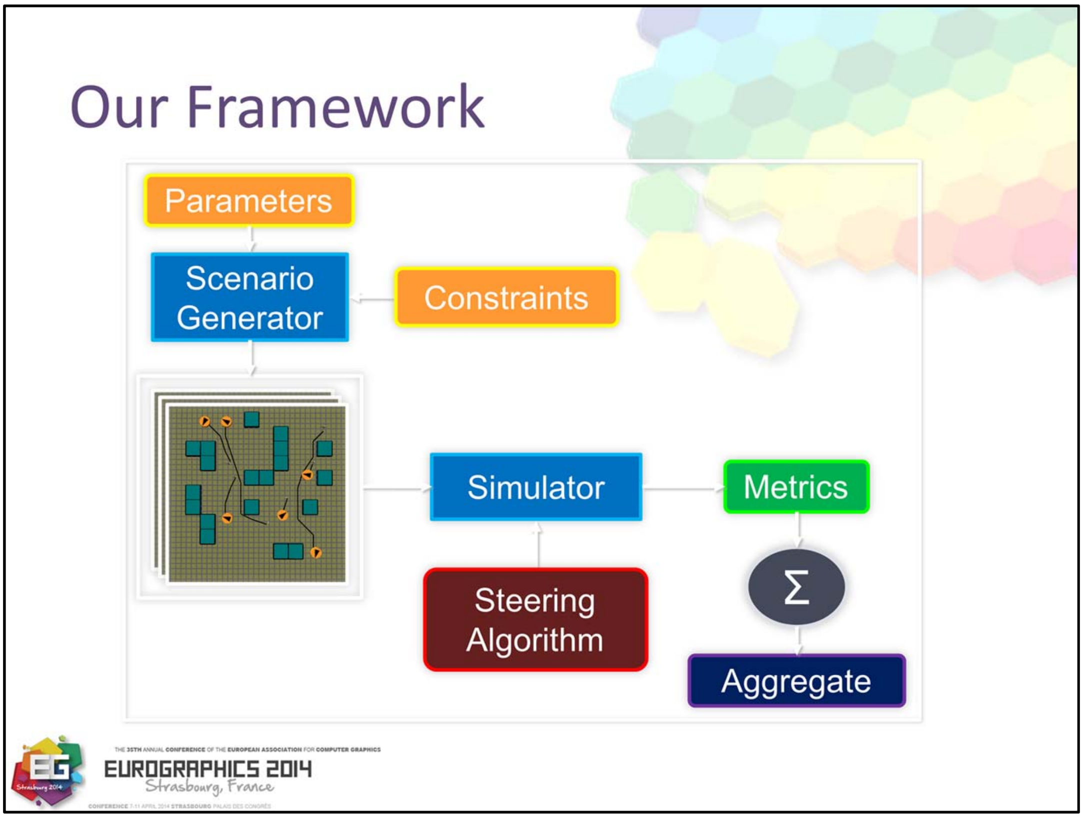
The Overtake scenarios were designed to test the ability of an agent to pass an agent from behind while in a narrow passage. We observe that the scores for all algorithms are approximately the same. However, However, visual inspection of the simulation shows that PPR and Reactive did not demonstrate an overtaking behavior.



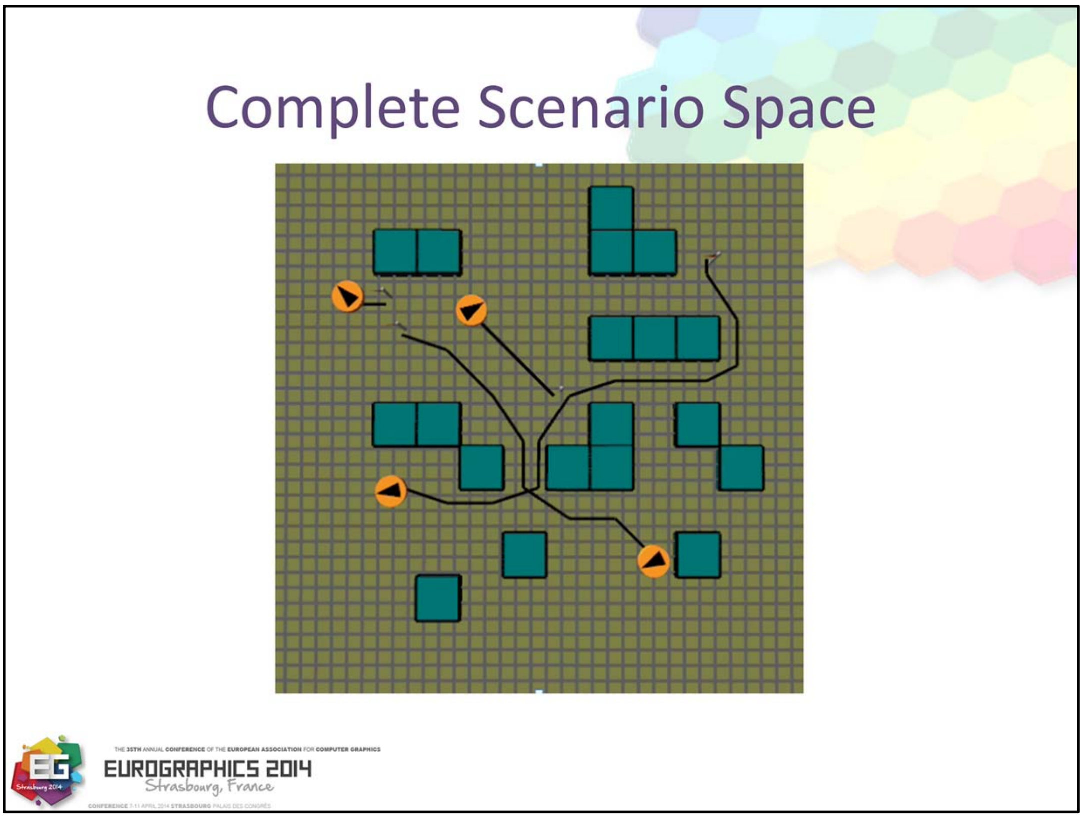
SteerBench only provides 42 hand designed scenarios for evaluation
The metrics depend on the length of the scenarios and cannot be compared across scenarios

The scores are only intended to serve as a basis for comparison between two algorithms and have no meaning on an absolute scale.

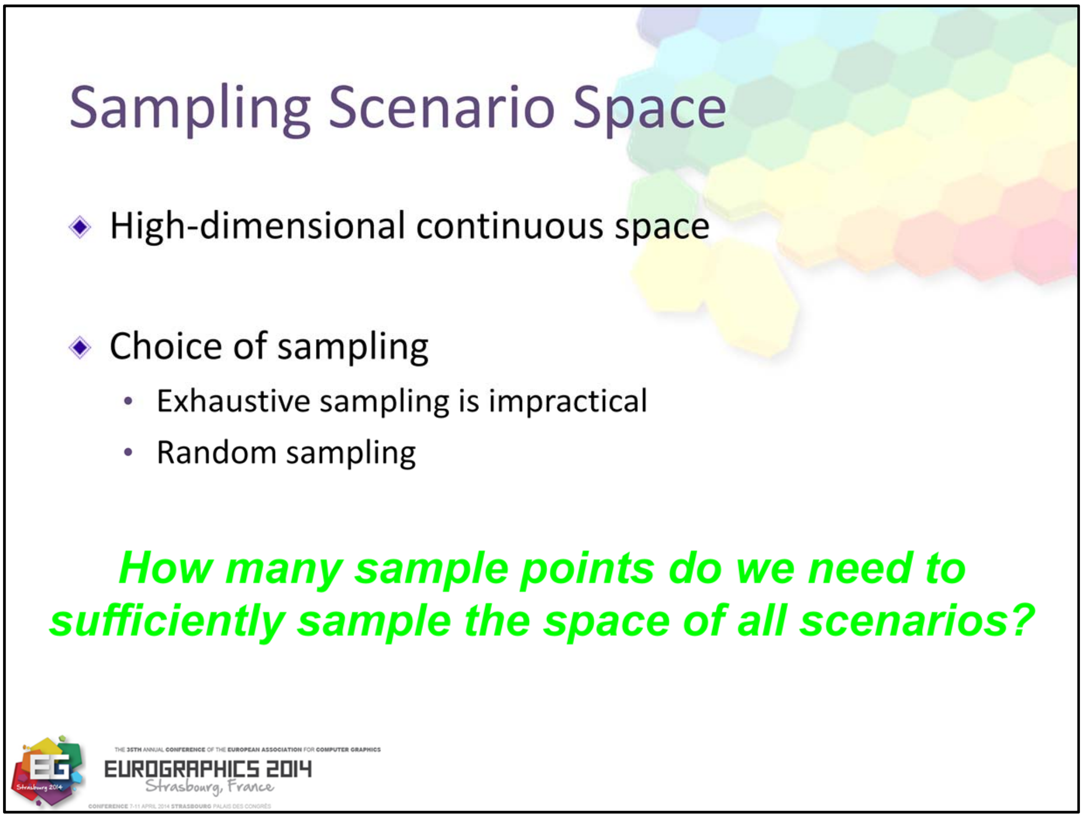
Also: SteerBench does not provide any notion of what a perfect score is for a particular scenario



In our framework, a user can specify a set of parameters to define the space of all scenarios. Constraints can also be defined to generate only meaningful scenarios and also define subspaces in the scenario space. We then sample scenarios in this space and evaluate a steering algorithm by computing a set of normalized metrics for each scenario by aggregating the metrics over the set of all scenarios, we can quantify meaningful quantities such as average quality and coverage of a steering algorithm as well as the failure set of a steering algorithm.

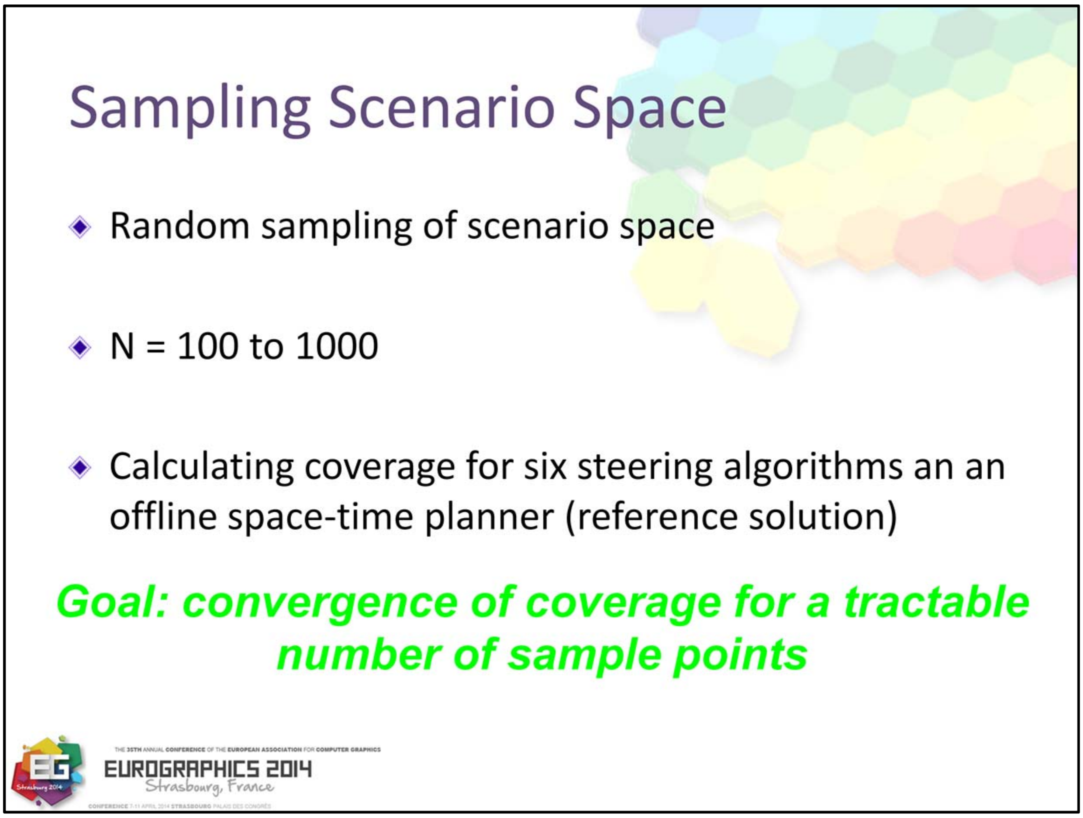


We generate scenarios in the complete scenario space by randomly positioning agents and obstacles. Scenarios are guaranteed to be collision free and have a solution



The complete scenario space is an extremely high-dimensional and continuous space which makes exhaustive sampling impractical
Hence, we choose to randomly sample scenarios within this space

The number of sample points needed is the minimum number of scenarios for which we see convergence of coverage of the steering algorithms

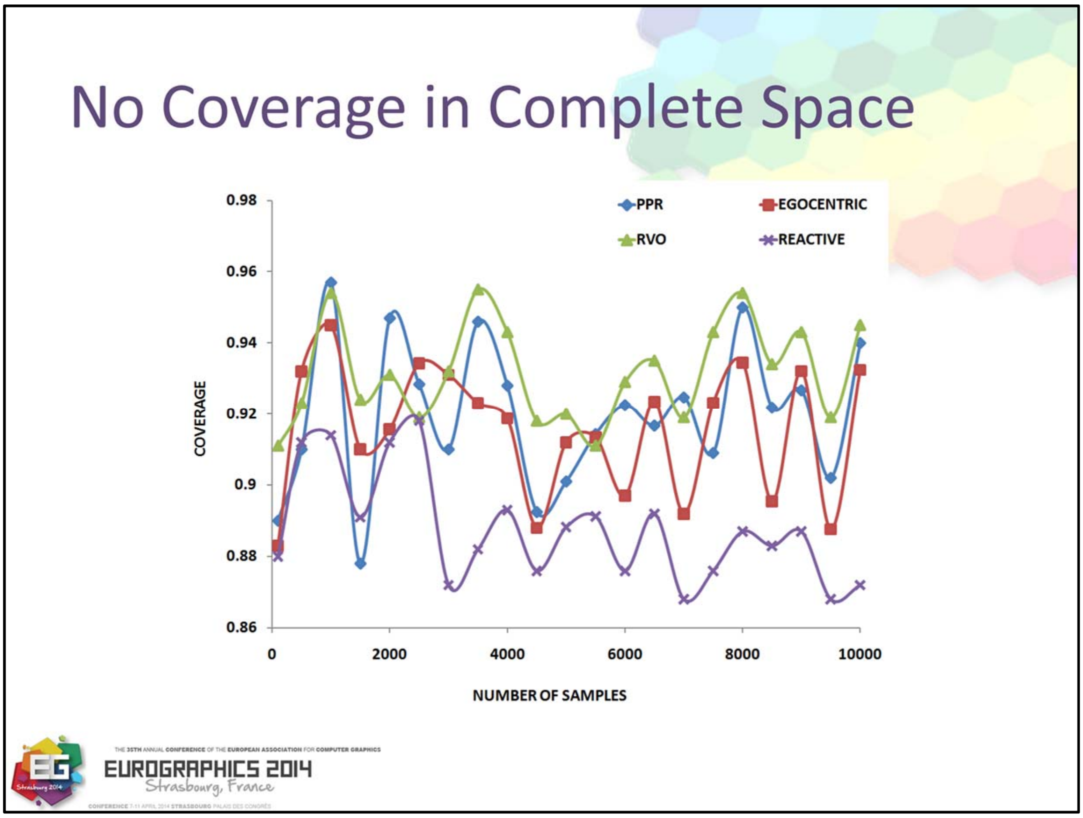


We perform a set of sampling experiments where we vary the number of samples from 100 to 10000.

For each experiment, we calculate coverage of the 4 steering algorithms for the number of scenarios that are sampled.

Coverage is defined as the ratio of the scenarios that could be successfully solved without collisions within a maximum amount of time

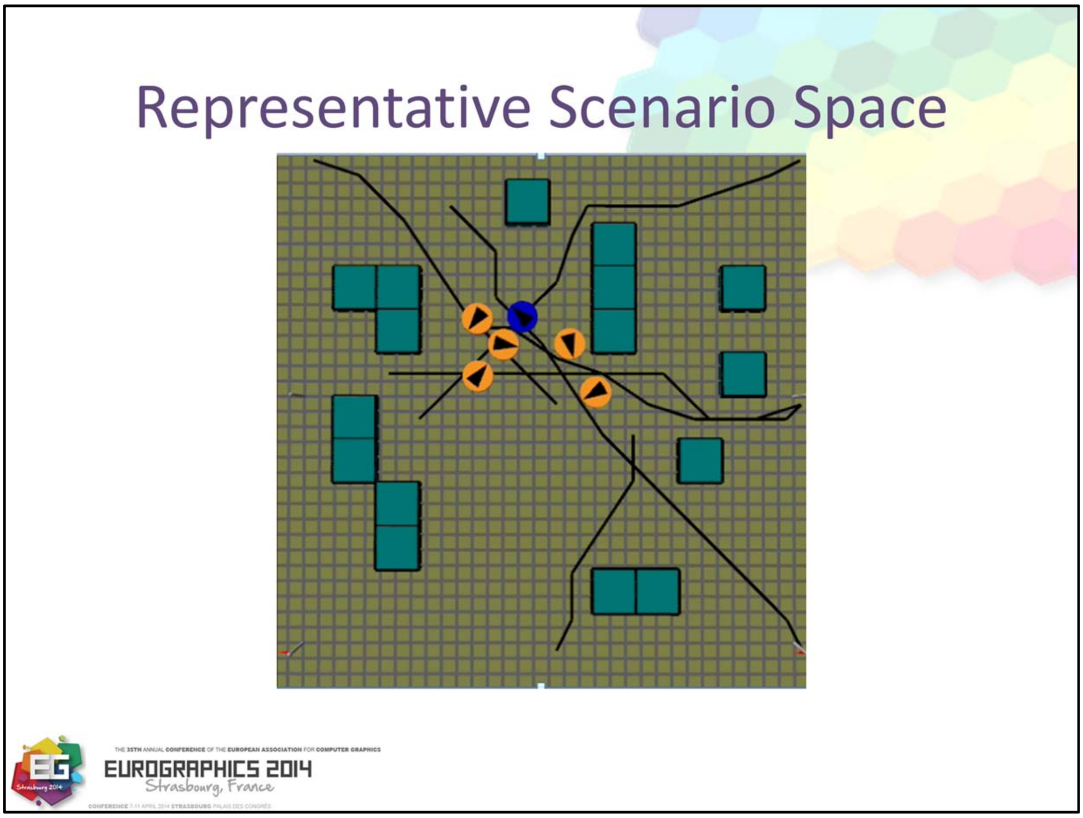
The goal of this experiment is to observe convergence of coverage for a tractable number of sample points.



We don’t see convergence of coverage for as many as 10000 sample points. Coverage of the space time planner is nearly 1

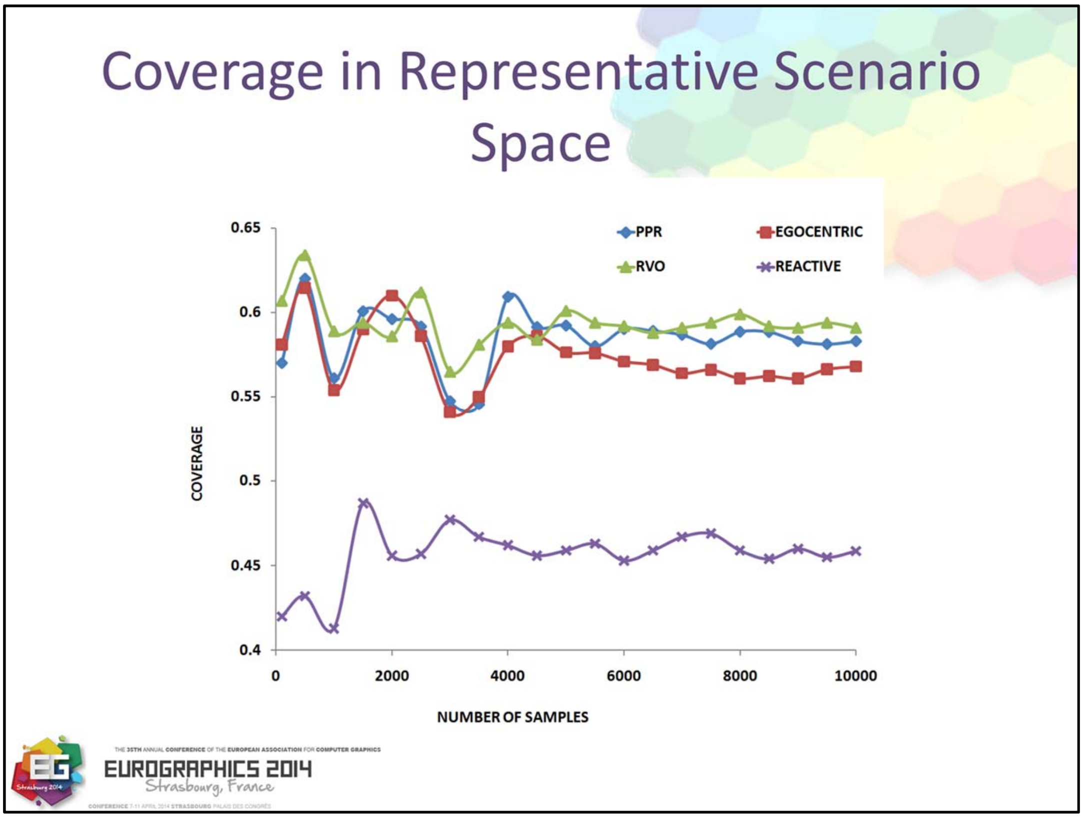
Also coverage of all 4 algorithms, even the simple reactive approach is quite high -- greater that 0.9

This indicates that many of the scenarios generated were trivial and could be easily solved which skew the measure of coverage and affect its convergence



To offset these limitations, We define scenarios in the representative space with respect to an agent which is always positioned at the center of the environment. Agent goals are always at the boundary.

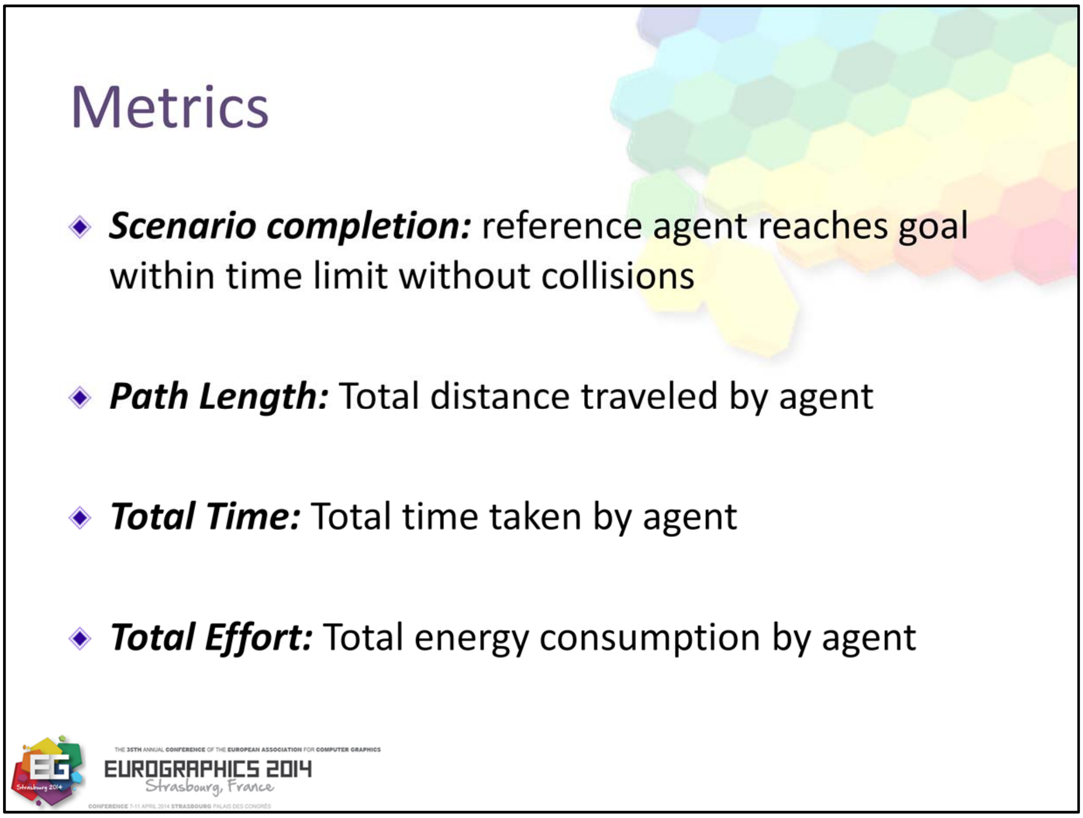
Other agents are more likely to be positioned closer to the reference agent and we ensure that the agents in the scenario interact with one another by constraining their static optimal paths to intersect in space-time



We observe convergance between N=5000 to 10000 sample points in the representative space

Also, coverage is much lower where the 3 reference algorithms can only solve 60% of the scenarios that were generated

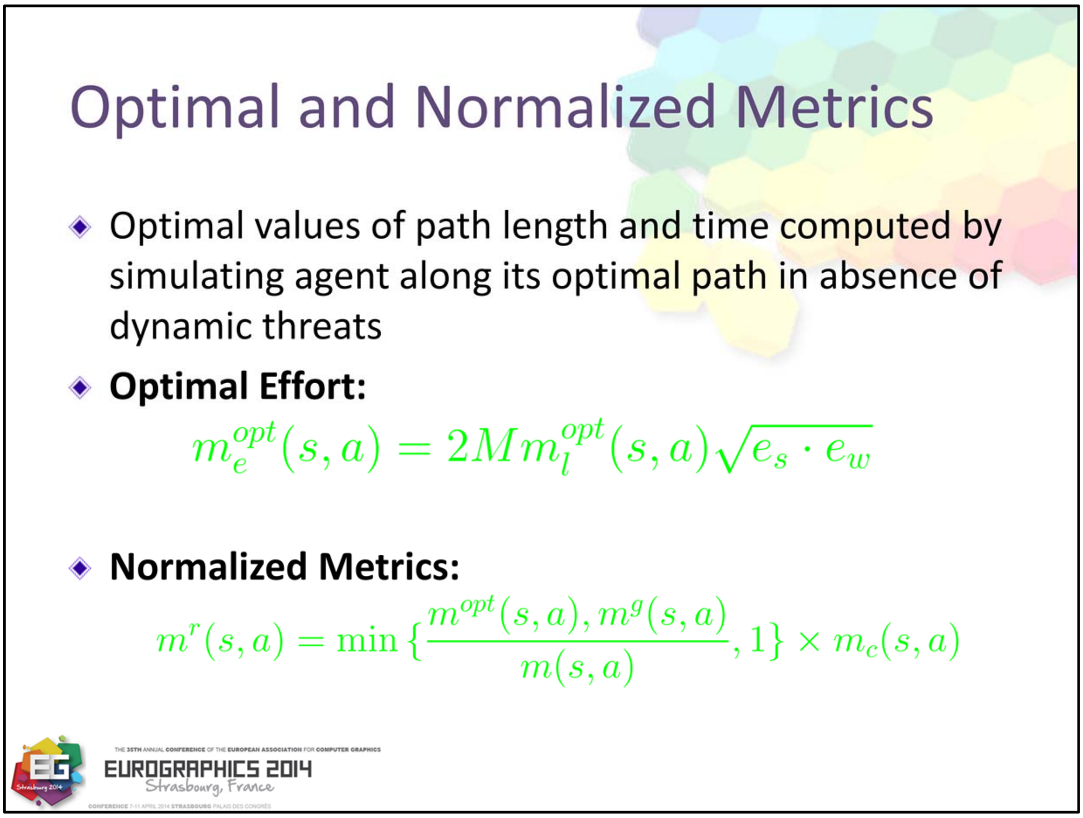
Also, there is a much larger difference in coverage between the baseline reactive approach and the other algorithms



Scenario completion which indicates the success of the reference agent in reaching the goal without collisions

Path length and total time measure the total distance traveled and the total time taken by the reference agent in reaching the goal

In addition, we introduce an effort metric which penalizes deviation from desired speed, the optimal path, as well as collisions



We compute the optimal values of path length and time by simulating the agent along its optimal path in the absence of dynamic threats

The optimal effort is the energy consumed while traveling along the optimal path at the desired walking speed

In order to make these metrics scenario independent, we normalize them as a ratio from their optimal or reference values.

A metric value is valid only if the scenario is successfully completed.

Scenario Clustering

- **Scenario Set:** subset of scenarios that satisfy metric range

$$S^{\mathrm{a}}_{\mathrm{m}}(T_1, T_2) = \{s | s \in R \wedge T_1 \leq m(s,a) < T_2\}$$

- **Success Set:** subset of scenarios for which metric is greater than threshold

$$S^{\mathrm{a}}_{\mathrm{m}}(T_{max}, 1) = \{s | s \in R \wedge T_{max} \leq m(s,a) < 1\}$$

- **Failure Set:** subset of scenarios for which metric is below minimum threshold

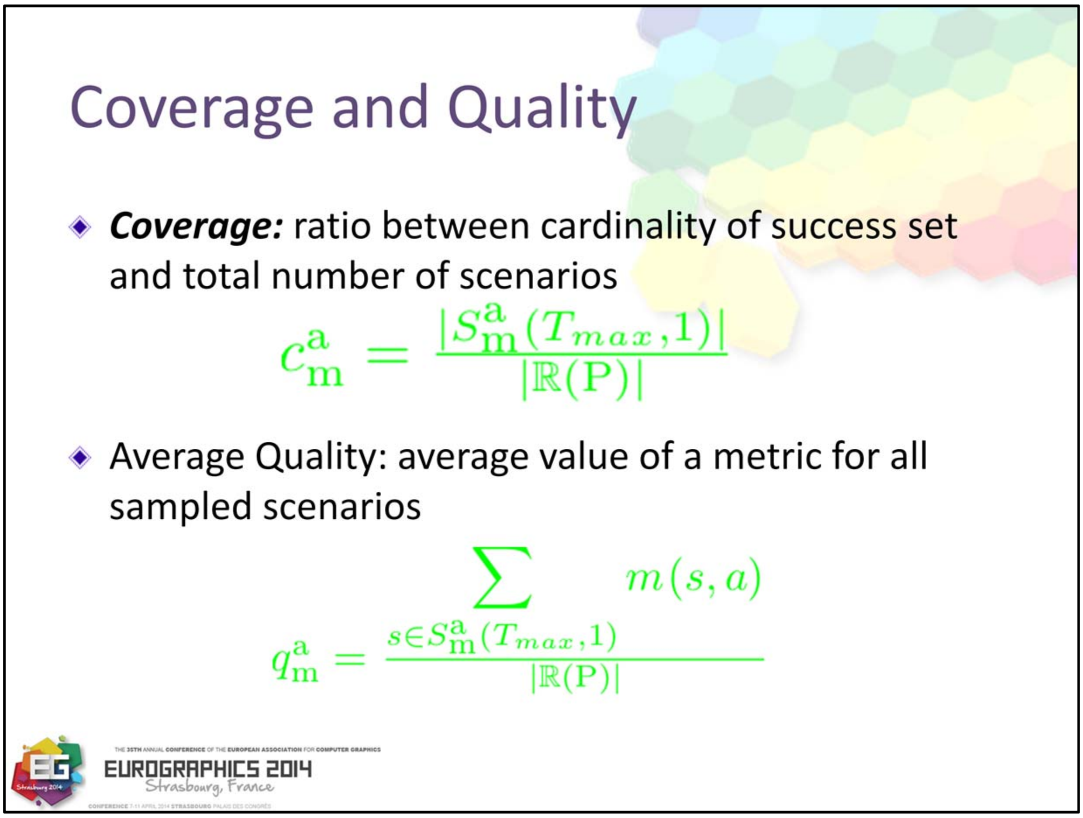
$$S^{\mathrm{a}}_{\mathrm{m}}(0, T_{min}) = \{s | s \in R \wedge 0 \leq m(s,a) < T_{min}\}$$

We can cluster scenarios together based on the measure of a metric for that scenario

For instance, the success set of an algorithm is the subset of scenarios for which the metric was greater than a threshold

Similarly, the failure set is the subset of all scenarios for which the metric was below a threshold

The failure set is a very useful tool for developers as it automatically generates scenarios which are particularly challenging for a steering algorithm



The coverage of a steering algorithm is computed as the ratio of the cardinality of the success set and the total number of scenarios

Similarly, the average quality can be computed as the average value of a metric for all sampled scenarios

An ideal algorithm would have a coverage and quality of 1

Average Quality

Algorithm	Path Length	Time	Effort
PPR	0.86	0.78	0.74
Egocentric	0.82	0.73	0.71
RVO	0.84	0.79	0.76
Ccrowds	0.78	0.69	0.67
Footstep	0.88	0.77	0.74
Reactive	0.67	0.63	0.59

The following table gives the average quality measures of the 6 algorithms for 10000 sampled scenarios in the representative scenario space with respect to path length, time and effort.

We observe that the average quality of the algorithms for path length is approximately 0.84. This implies that the algorithms generally produce solutions with path lengths that are 84% of the reference values.
In contrast, the average quality for time is 0.75 which is considerably lower. This is because steering algorithms generally prefer to slow down instead of deviating from their planned paths.

The quality measure for effort captures length, time, as well as collision penalties and provides a single numeric value to quantify the performance of a steering algorithm in the representative space of scenarios.

Coverage

Algorithm	Success	Path Length	Time	Effort
PPR	0.58	0.91	0.88	0.83
Egocentric	0.57	0.86	0.75	0.79
RVO	0.59	0.92	0.96	0.86
Ccrowds	0.51	0.76	0.75	0.68
Footstep	0.74	0.94	0.96	0.90
Reactive	0.46	0.27	0.26	0.63

EUROGRAPHICS 2014 Strasbourg, France

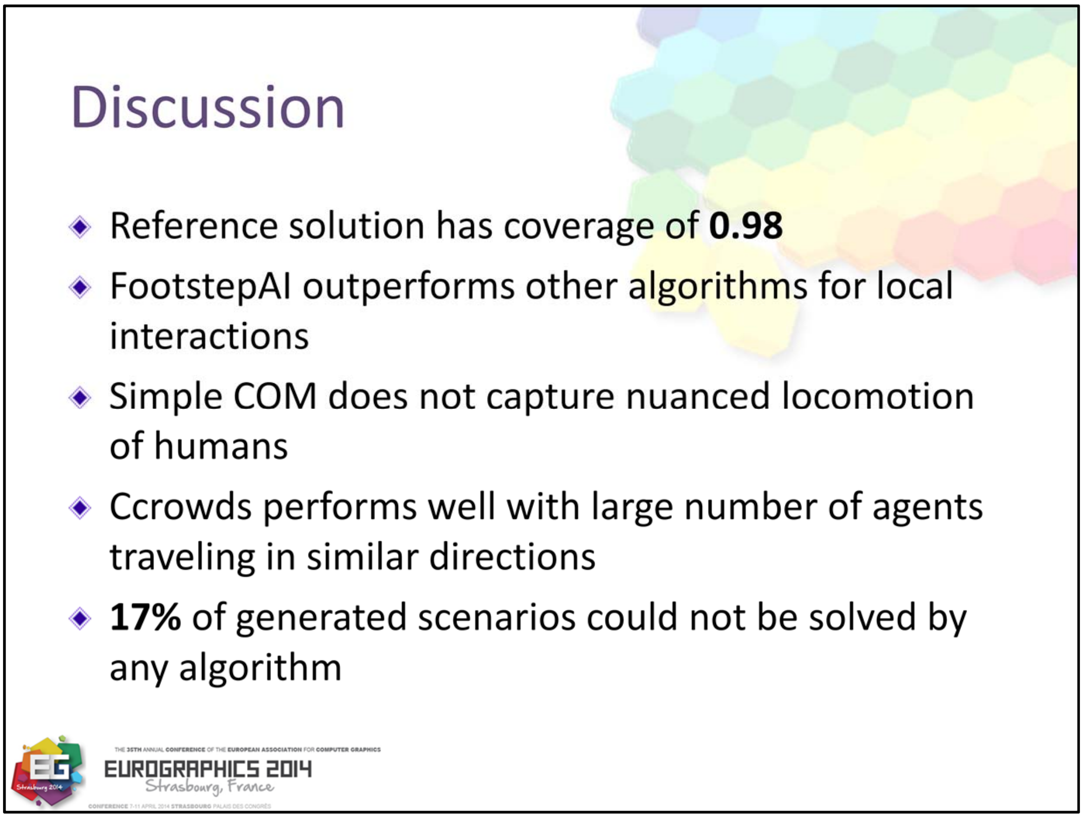
This table gives the coverage of all 6 algorithms with respect to success, path length, time, and effort

Note that success is a boolean value but length and time are not.
For this reason, we need to choose thresholds which determine whether an algorithm successfully solved a scenario with respect to length and time
We choose the threshold as the mean of the average quality of the 5 published algorithms (reactive is not considered here)

So, we can now interpret coverage as the ratio of scenarios for which an algorithm produced a solution which was greater than the average quality measure of the state of the art

PPR, Egocentric, and RVO successfully solve nearly 60% of all scenarios while Footstep has a success rate of 74%

The performance of Reactive is reflected in its measure of coverage. We observe that Reactive can only solve 45% of the scenarios, and that only 25% of its solutions are above the average quality measure.



Our reference space time planning solution has a coverage of 0.98 in the representative scenario space.

Perfect coverage is not achieved because agents don’t cooperate while planning but instead plan in order.

A global space time planner should have a coverage of 1, but would be intractable in practice. A major reason for the low coverage is the simple COM control in these approaches which can cannot capture the nuanced locomotion of real humans

Continuum crowds has low coverage for local inter-agent interactions with independent goals.

17% of the generated scenarios could not be solved by any of the 3 approaches.
these scenarios include sequences of sharp turns where agents encounter dynamic threats in tight corners
Combinations of oncoming and crossing threats also cause complex interactions between agents which results in collisions
Another potential cause for failure are potential deadlock situations where agents must backtrack to allow other agents to go through.

Acknowledgements

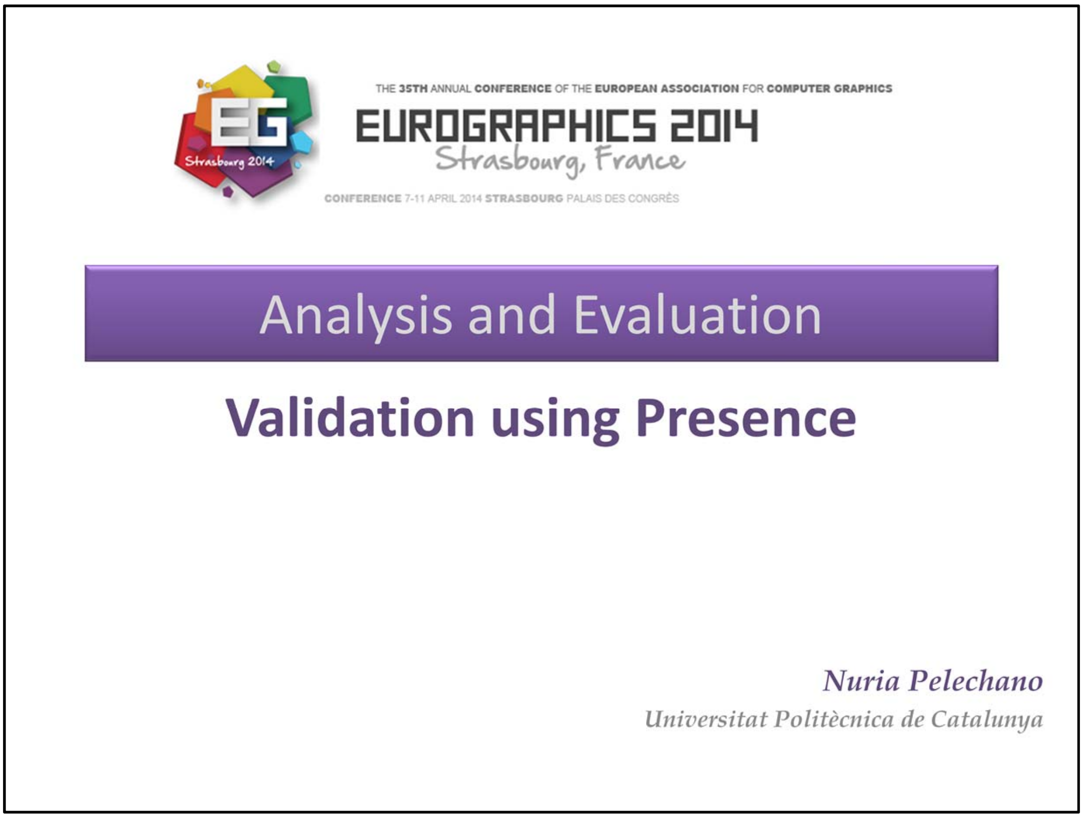
- Norman Badler, Petros Faloutsos, Glenn Reinman, Demetri Terzopoulos , Ari Shapiro, Nuria Pelechano, Ladislav Kavan, Mitchell Lum, Nuria Pelechano, Shawn Singh, Alexander Shoulson, Pengfei Huang, Cory Boatright, Funda Durupinar, Alejandro Beacco Porres , Aline Normoyle, Tiantian Liu, Tianyu Huang

Part of the research reported in this presentation was supported through:

- Intel Visual Computing Grant: *Using a many-core environment for high performance large scale virtual character simulation.*
- Intel Equipment Grant.

Part of the research reported in this presentation was performed in connection with Contract Number W911NF-10-2-0016 with the U.S. Army Research Laboratory. The views and conclusions contained in this document are those of the authors and should not be interpreted as presenting the official policies or position, either expressed or implied, of the U.S. Army Research Laboratory, or the U.S. Government unless so designated by other authorized documents. Citation of manufacturers or trade names does not constitute an official endorsement or approval of the use thereof. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation heron.

THE 35TH ANNUAL CONFERENCE OF THE EUROPEAN ASSOCIATION FOR COMPUTER GRAPHICS
EUROGRAPHICS 2014
Strasbourg, France
CONFERENCE 7-11 APRIL 2014 STRASBOURG PALAIS DES CONGRÈS



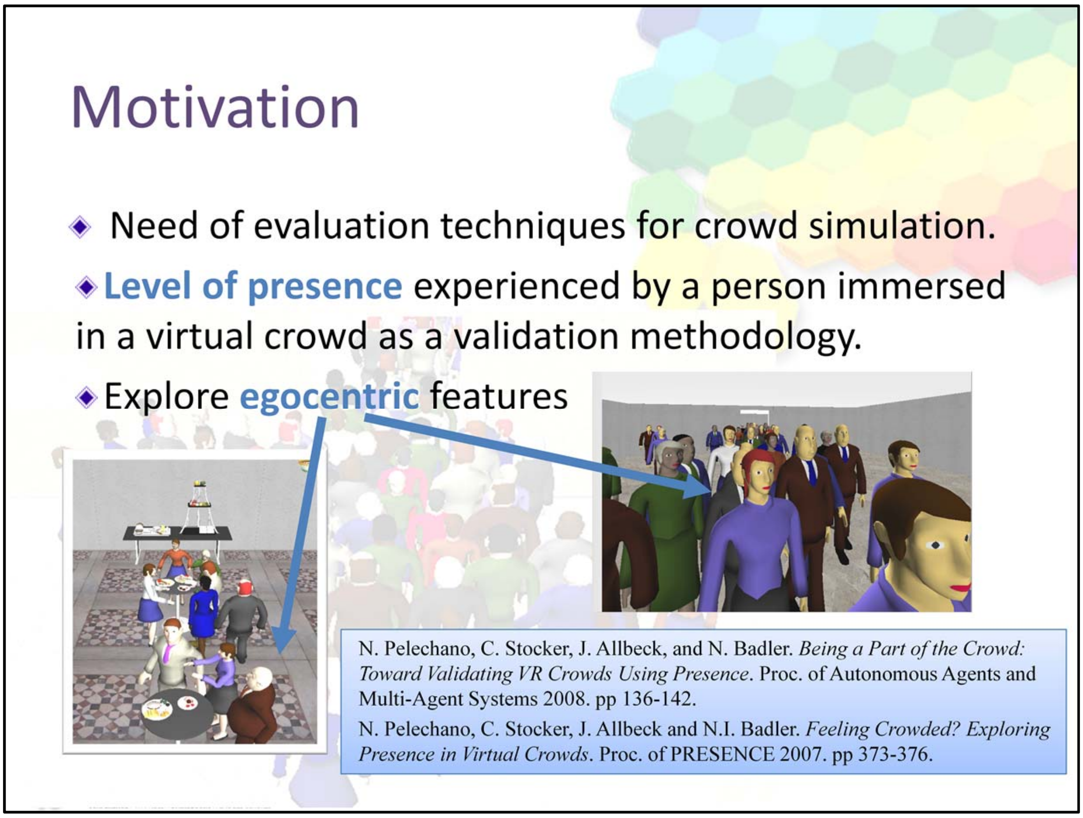
Speaker: Nuria Pelechano
Assistant Professor, Universitat Politecnica de Catalunya

Nuria Pelechano is an Associate Professor at the Universitat Politecnica de Catalunya, where she is a member of the Virvig-Moving group. Her research interests include animation, simulation and rendering of crowds, real-time 3D graphics, and human avatar interaction in virtual environments.
She has published many papers in international conferences and journals including Symposium on Computer Animation, Computer Graphics Forum, and IEEE Computer
Graphics and Applications. She is also co-author of a book entitled "Virtual Crowds: Methods, Simulation, and Control“ (Morgan and Claypool) with N. Badler and J. Allbeck.

Recommended reading:

N. Pelechano, C. Stocker, J. Allbeck, and N. Badler. *Being a Part of the Crowd: Toward Validating VR Crowds Using Presence*. Proc. of Autonomous Agents and Multi-Agent Systems 2008. pp 136-142.

N. Pelechano, C. Stocker, J. Allbeck and N.I. Badler. *Feeling Crowded? Exploring Presence in Virtual Crowds*. Proc. of PRESENCE 2007. pp 373-376.

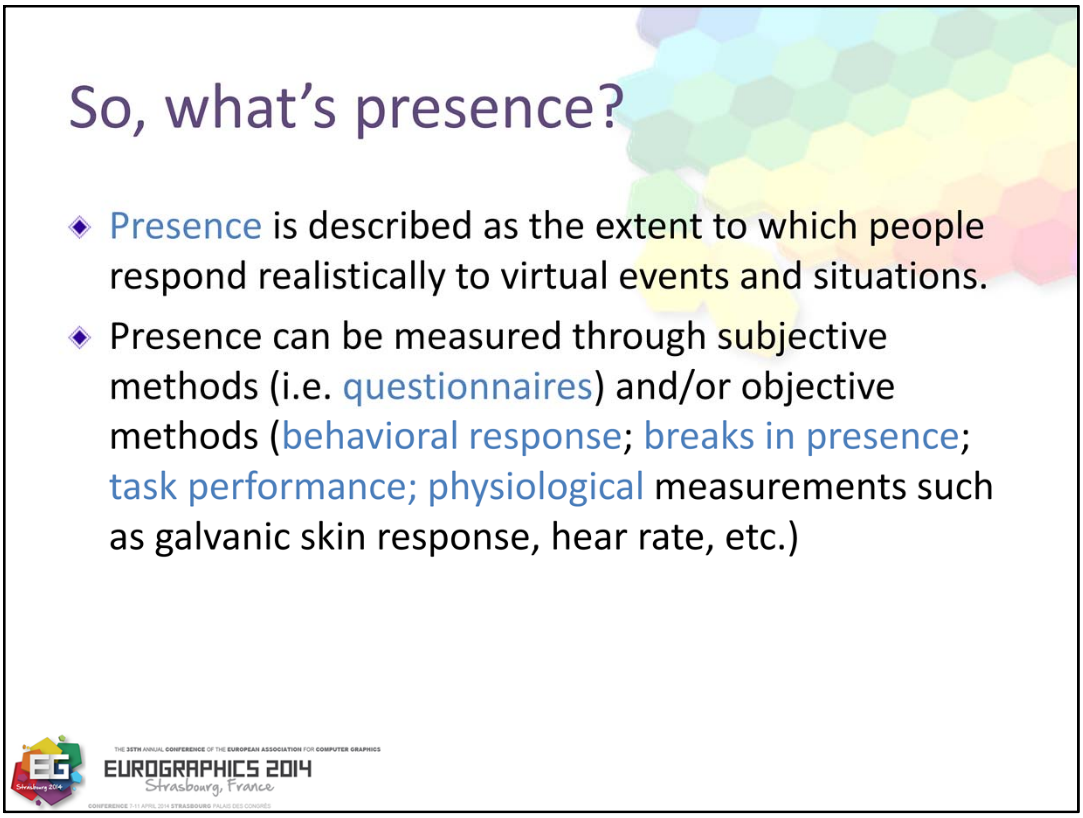


Crowd simulation models are currently lacking a commonly accepted validation method. In our work, we proposed to study levels of presence achieved by a human in a virtual environment (VE) as a metric for virtual crowd behavior. Using experimental evidence from the presence literature and the results of a pilot experiment, we can make decisions regarding the egocentric features that a crowd simulation model should have in order to achieve high levels of presence and thus be used as a testbed for validation of simulated crowd behavior.

Controlled experiments in VE are invaluable to gather information that is difficult to gather in real life such as:

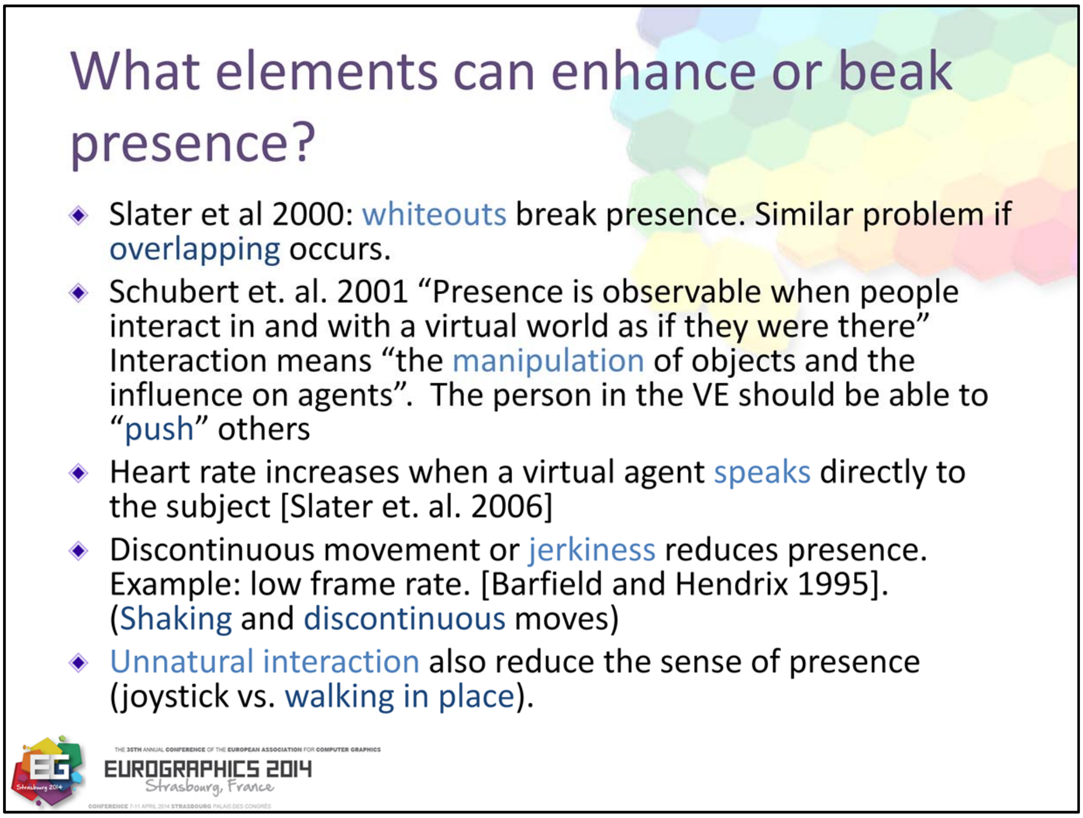
- Exit routes during a fire?
- How many people would follow leaders?
- How much people trust what other’s communicate?

Virtual reality allows us to replicate experiments, modify variables and study their direct impact on human behavior. So if we can set up an environment that highly simulate reality in a way that gets human participants to behave as they would do in real life, we have the perfect safe scenario to study human performance and behavior.



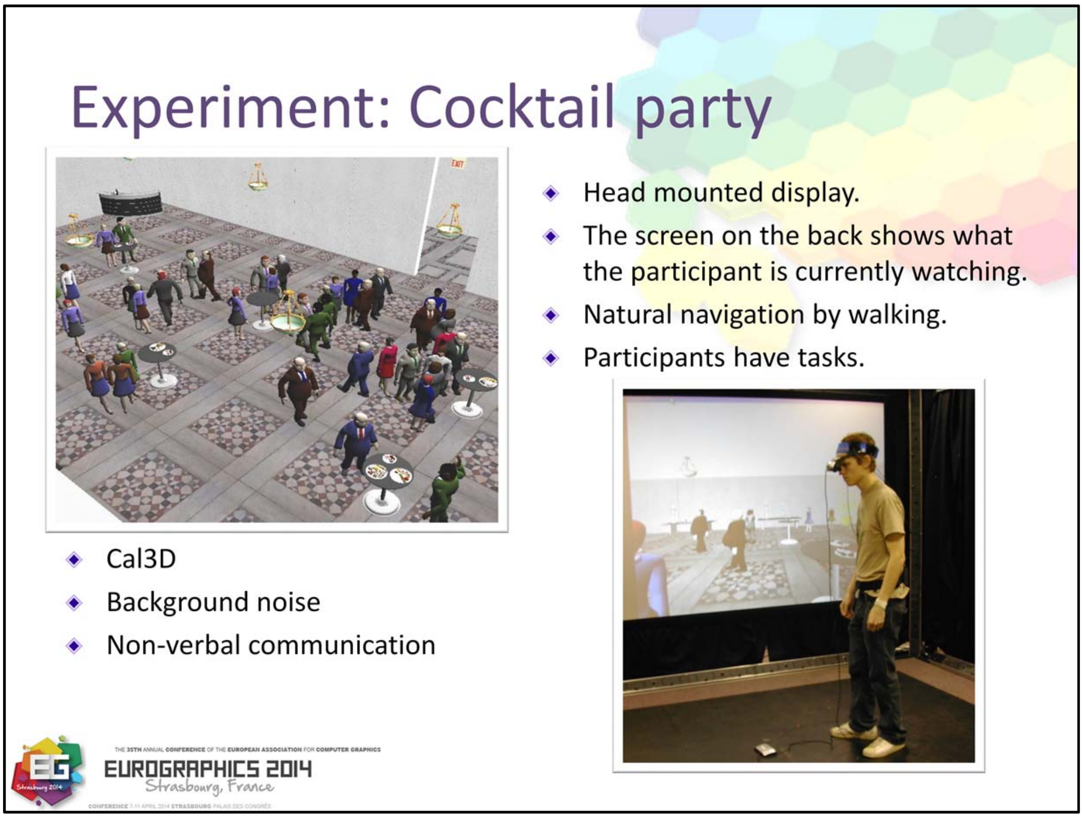
Presence is described as the extent to which people respond realistically to virtual events and situations. Responding realistically implies realism at many levels, ranging from physiological through behavioral, emotional and cognitive behaviors.

Classic presence work relied on questionnaires to measure presence, but since questionnaires depend entirely on a user's subjective view of their experience, researchers found it necessary to develop other supplementary methods. Those methods include behavioral measurements (social and postural responses, etc.), physiological measurements (galvanic skin response, heart rate, etc.), task performance measurements (completion times and error rates, etc.), and counting breaks in presence.



We closely studied the presence literature to determine what elements needed to be taken into account when designing the experiment, in order to achieve high levels of presence. Publications in this field have proven the following:

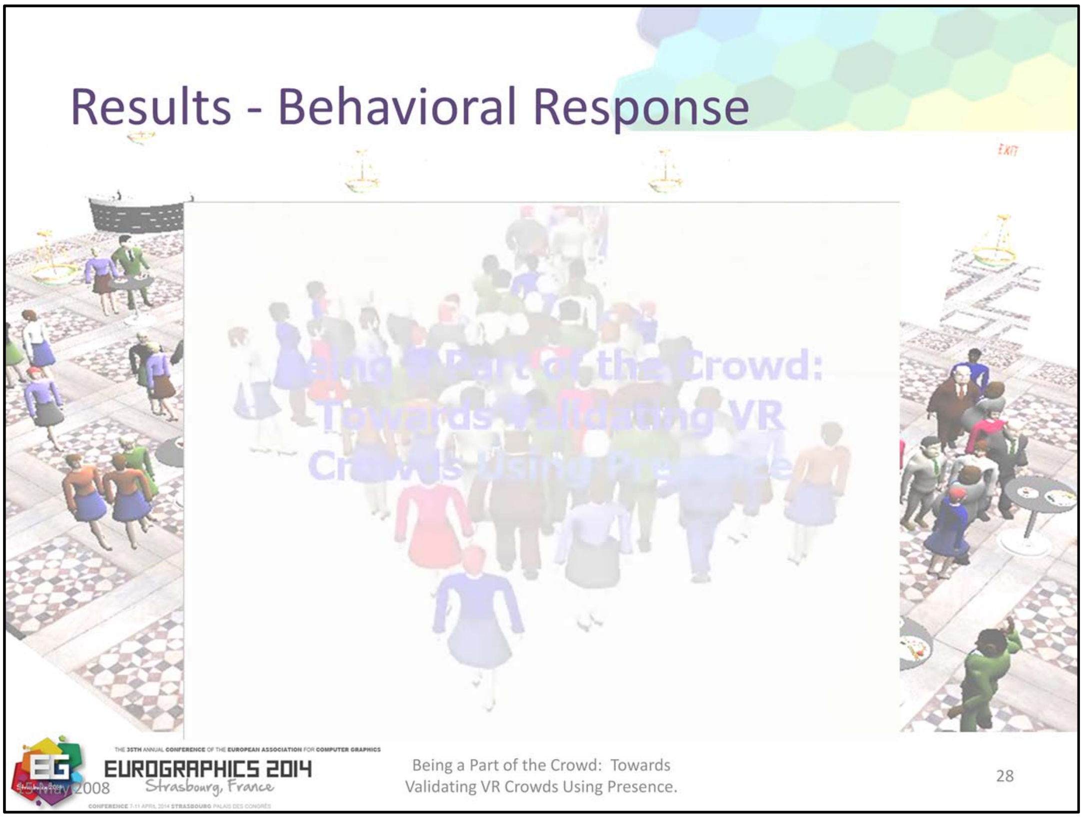
- Breaks in presence have been used to count the transitions from the virtual to the real world. These transitions can be triggered by occurrences such as bumping into a wall in an immersive environment, tripping over cables, and whiteouts.
- Being able to physically manipulate objects and communicate with virtual humans in a VE increases a sense of presence.
- Some studies show that the heart rate of a participant increases when a virtual agent speaks directly to him.
- discontinuous movement or jerkiness reduces presence. Jerkiness can be observed when, for example, the VE
- suffers from low frame rate. Therefore we can expect that crowd models suffering from agents shaking continuously or appearing to move between large discrete positions will likewise diminish the participant's sense of presence.
- Unnatural interactions with the VE, such as using a joystick to maneuver, can reduce the sense of presence when compared to techniques that resemble real life navigation such as "walking in place" .



For the experiment we created a virtual scenario simulating a cocktail party. At the party were virtual party-goers who walked around “mingling” with others through non-verbal communication and gestures. After a specified time, a bell rang and the virtual agents calmly exited the party. The virtual agents had several animations assigned including different walking styles that could be blended smoothly, and a set of idle and gesturing animation clips that could be used when agents stop walking or gather around a table.

Participant’s tasks:
- Training phase: wander around while the crowd was still, locating three objects.
- Once the crowd starts moving, the subjects will have to count the number of people with red hair, and leave when an alarm goes off.
- Each subject will experience 2 crowd models (We had a total of 4 crowd models)

Our Goal: to study whether participants would react to a virtual crowd as they would do in reality.
Results came from questionnaires and from the authors studying the behavioral response from the videos.

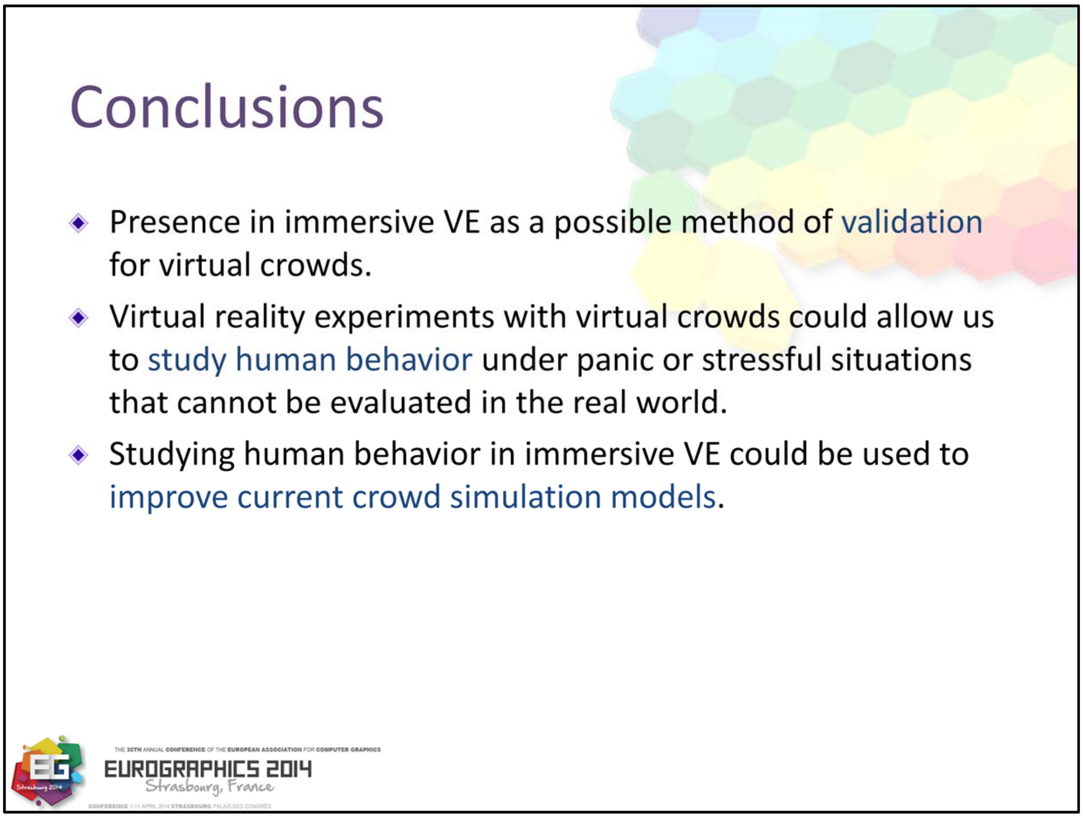


This video can be found in:
http://www.lsi.upc.edu/~npelechano/videos/crowds&presence_divX.avi Observations from their behavioral response:

-Moving backwards after bumping into a virtual character.
-Stepping sideways to avoid a virtual agent walking into them
-Turning their head to watch an agent walk around them.
-One of the participants even waved back in response to a virtual agent's wave.

From the comments that we gathered. It’s worth mentioning a few:
“The sense of crowd movement was most compelling during the evacuation.” “I felt bad whenever I bumped into someone.” “The second time, everyone immediately started leaving and it made me really want to leave as well.”

These examples show that some people do think about the interaction with virtual agents in a similar way as when they interact with real people.



The goal of this pilot experiment was to examine whether participants interacting with a virtual crowd experience would react to the virtual crowd as they would do in a similar real situation.
From our experiments we have been able to observe that some participants exhibit some behaviors consistent with the notion that they were responding to the crowd realistically.

Using egocentric features based on established presence enhancing experiences, we can conclude that interacting with the other agents in a crowd (by our virtual representation being pushed physically and by communicating with them) and being able to materially affect the movements of other members of the crowd (by pushing on them and having them avoid collisions with the self) will likely enhance a subject's sense of presence. Arranging for the virtual crowd to push back (physically) on the subject is clearly more difficult, and we may be able to explore a haptic solution using vibrotactile elements.

Virtual reality experiments with virtual crowds are necessary to study human behavior under panic or stressful situations that cannot be evaluated in the real world (i.e., building evacuation due to fire). In order to carry out those experiments it is necessary to use a crowd simulation model in which a real person is seamlessly immersed and experiences a high sense of presence when interacting with such a crowd.

Acknowledgements

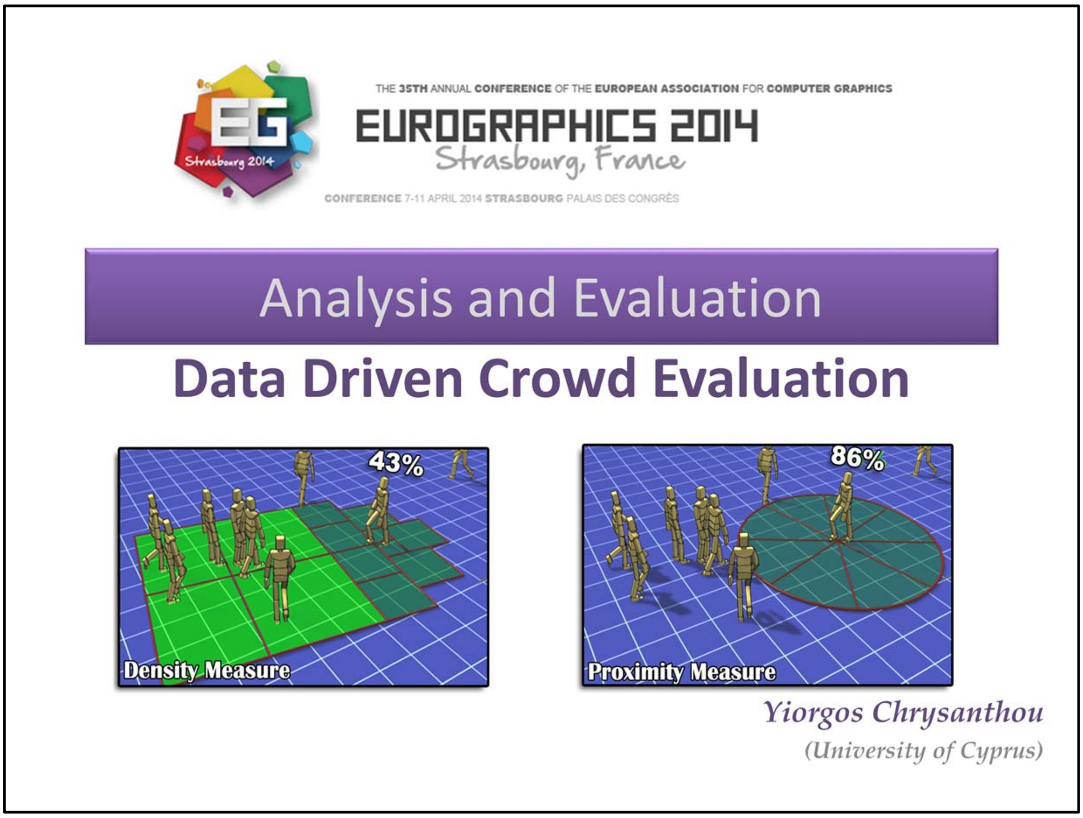
Collaborators

- Mel Slater
- Norman I. Badler
- Jan Allbeck
- Catherine Stocker

Sponsors

- Army MURI W911NF-07-1-0216
- Spanish Ministry of Science and Education (grant TIN2007-67982-C02-01)

EUROGRAPHICS 2014
Strasbourg, France



Speaker: Yiorgos L. Chrysanthou, University of Cyprus

Yiorgos L. Chrysanthou is an Associate Professor at the Computer Science Department of the University of Cyprus, where he is heading the Graphics and Hypermedia lab. He received his PhD from Queen Mary and Westfield College (1996) and worked for several years as a research fellow and a lecturer at University College London (up to 2001). Yiorgos has published over 65 papers in journals and international conferences on Computer Graphics and Virtual Reality and is a co-author of the book "Computer Graphics and Virtual Environments: From Realism to Real-Time", (Addison-Wesley 2001 + China Machine Press 2004). He has served as Program Chair for international conferences (VAST 2004, ACM VRST 2005 and ACM VRST 2006, ECMS 2008, MIG 2010) and has been the local or overall coordinator of many research projects funded through various sources. His research interests are in the general area of 3D Computer Graphics, including real-time rendering, reconstruction of urban environments and virtual humans, with applications to serious games and cultural heritage. Most of his recent work has focused specifically on data driven simulation and evaluation of character behavior.

Why is it Needed?

- Crowds are important in many applications
- Evaluation can help debug and develop better methods, compare methods
- Recent approaches
  - Vision methods
    - whole trajectories, judged individually, only in captured scene
  - CG - User experiments
    - Ennis, Peters, O'Sullivan '08 & Pelechano et. al '08
  - CG - Measuring time varying metrics for steering
    - Singh, Faloutsos et al. '08 & Kapadia, Faloutsos et al '09, Guy et al. '12, Karamouzas, Overmars '12

EUROGRAPHICS 2014 Strasbourg, France

Why is it Hard?

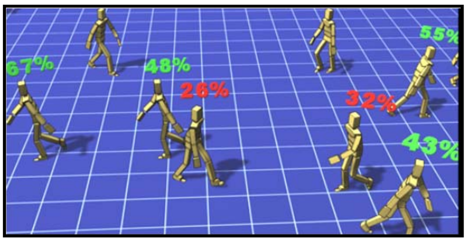
- Efficiency not always the driving force in behavior
- Behavior depends heavily on the context
- Defining what is "natural" is almost as hard as the simulation it self

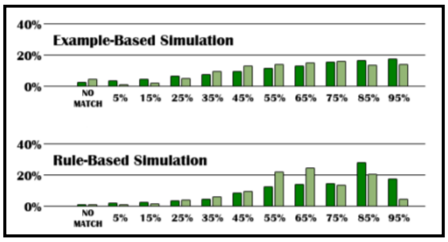
Lerner, Chrysanthou, Ariel Shamir, Cohen-Or, Context-Dependent Crowd Evaluation, **PG2010**

Data-Driven Evaluation of Crowds

34

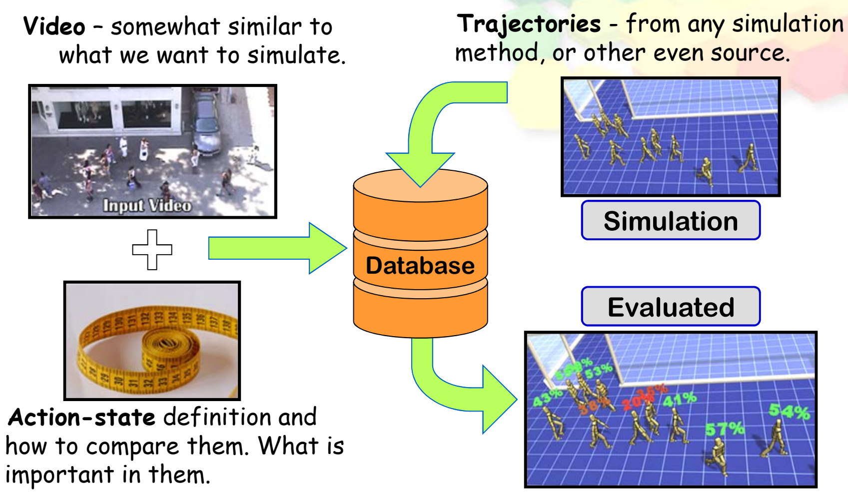
- Compare to how real-people respond under similar conditions
  - Create instances of {state-action} pairs
- It evaluates both:
  - individual behaviors
  - the simulation as a whole





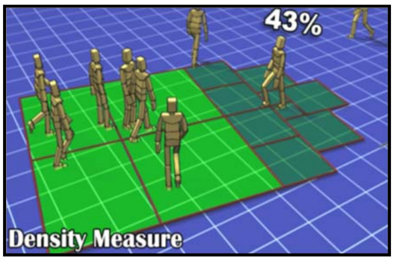
Overview

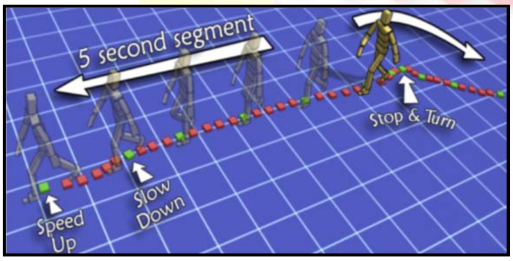
35



Video

36



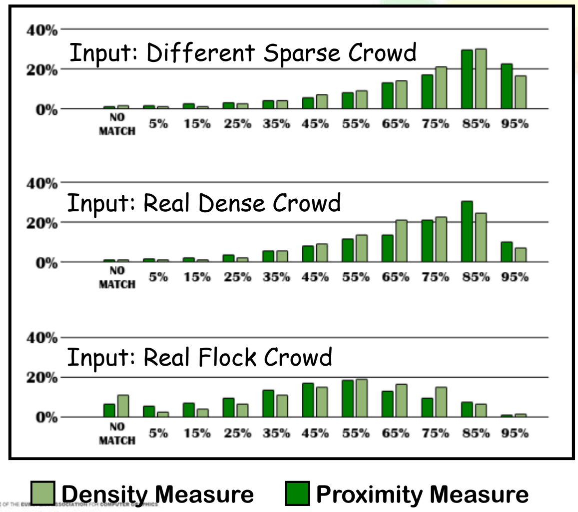


1:50 & 3:12

EUROGRAPHICS 2014
Strasbourg, France

Evaluating a Real Sparse Crowd

37



EUROGRAPHICS 2014
Strasbourg, France

THE 35TH ANNUAL CONFERENCE OF THE EUROPEAN ASSOCIATION FOR COMPUTER GRAPHICS
EUROGRAPHICS 2014
Strasbourg, France
CONFERENCE 7-11 APRIL 2014 STRASBOURG PALAIS DES CONGRÈS

Simulating Heterogeneous Crowds

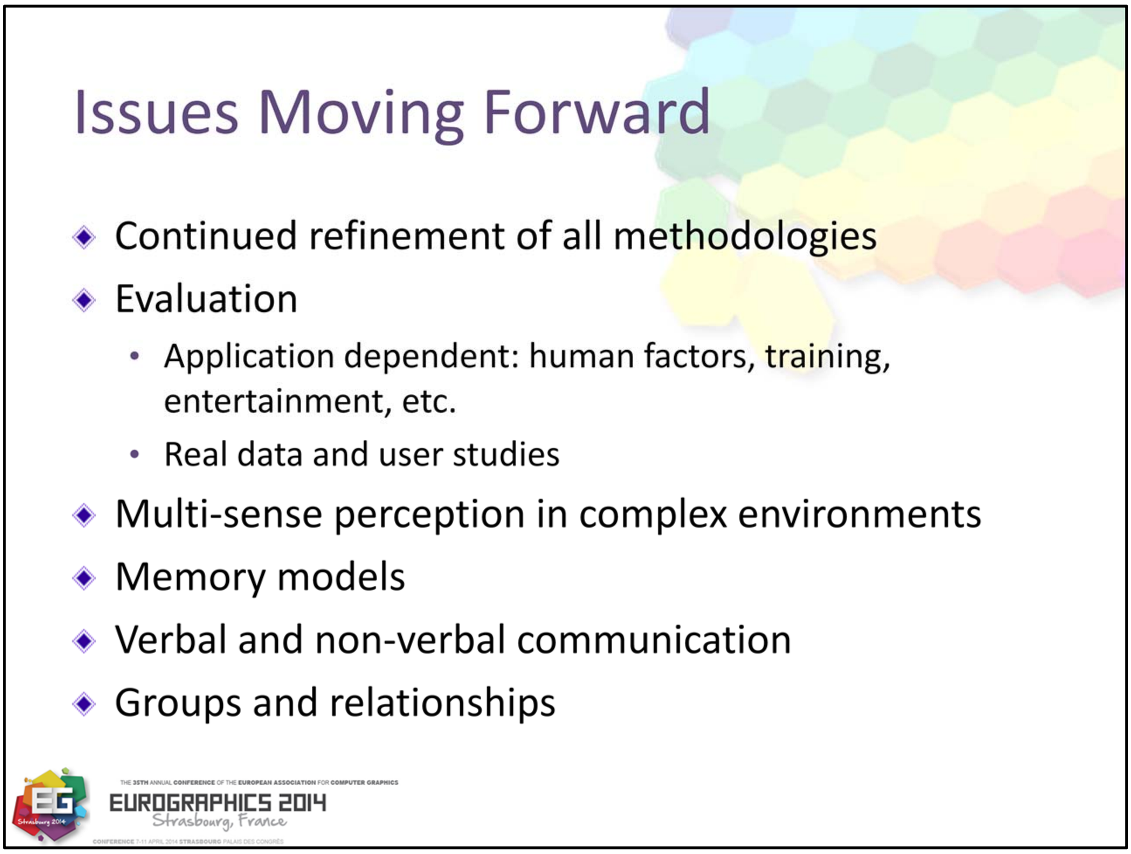
Wrap-up

Jan Allbeck
George Mason University

Summary

- Navigation
- Heterogeneous Behavior
- Realism
- Analysis and Evaluation

Creating crowd simulation frameworks can be quite challenging. In order to obtain good results, many components need to constructed and tuned. We need to ensure that agents can maneuver through large, complex environments. Their behaviors need to be appropriately varied and consistent with environmental context. To facilitate real-time simulation of large crowds, we must implement techniques that are computationally light, but that do not overly degrade realism. Finally, there needs to be methods for evaluating crowd simulations and determine what techniques are most appropriate for various applications.



Crowd simulation research continues to evolve. All of the research areas discussed in this tutorial continue to progress toward more robust, computationally feasible frameworks. Evaluating simulation techniques and determining a basis for comparison remains challenging. Critical features may be application dependent. Is absolute correspondence to real humans most important or is it more important to allow animators to modify the simulations to achieve their creative vision? Moreover, how can we best determine whether or not our simulation frameworks achieve the features we set up to achieve?

What additional human capabilities might enhance crowd simulations? Multi-sense perception would enable agents to sense and ultimately react to visual, auditory, olfactory, and haptic cues. Memory models could be used to create additional variations in agent behaviors by allowing agents' histories to influence their decision-making. If agents were able to express themselves and effectively communicate with one another, it might dramatically alter crowd simulations. In addition to communicating information about paths, resources, and hazards, we can imagine agents observing emotional states (i.e. emotional contagion) and reputations. Naturally, the formation of groups and relationships would alter how agents navigate the environment and impact many aspects of the animation and decision-making components.