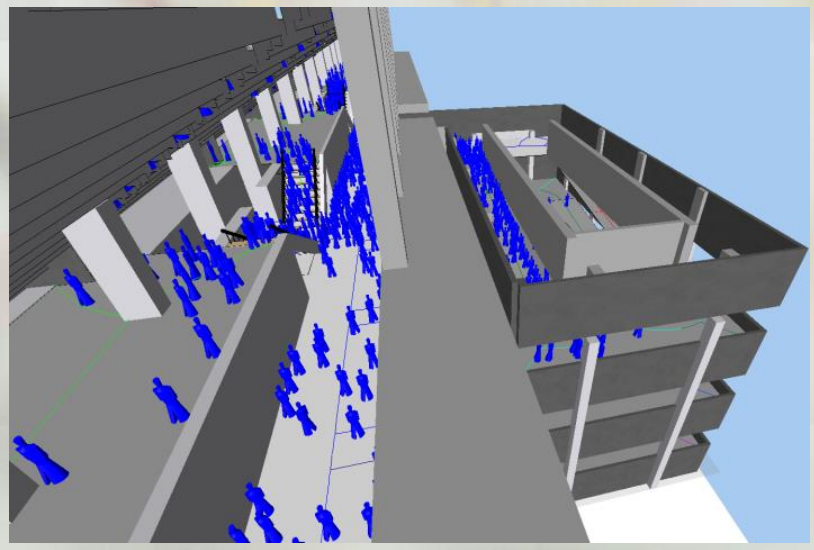
Universiteit Utrecht

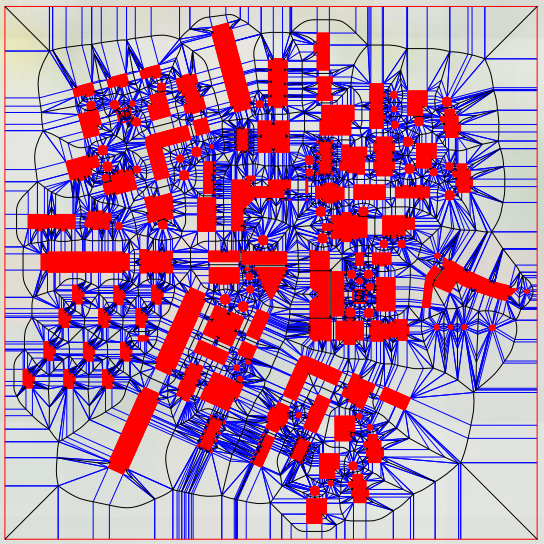
[Faculteit Bètawetenschappen
Informatica]

A computational model of human navigation

Roland Geraerts

PED 2014





Social relevance

- The model is needed to
  - Decide whether crowd pressures do not build up too much during a festival;
  - Find out how to improve crowd flow;
  - Plan escape routes for use during a fire evacuation;
  - Train emergency personnel to deal with evacuation scenarios;
  - Study a range of scenarios during an event;
  - Populate a game environment with realistic characters.

Love Parade in Duisburg, 2010

Rebuilding of
Utrecht train station

Sports stadium

A computational model of human navigation

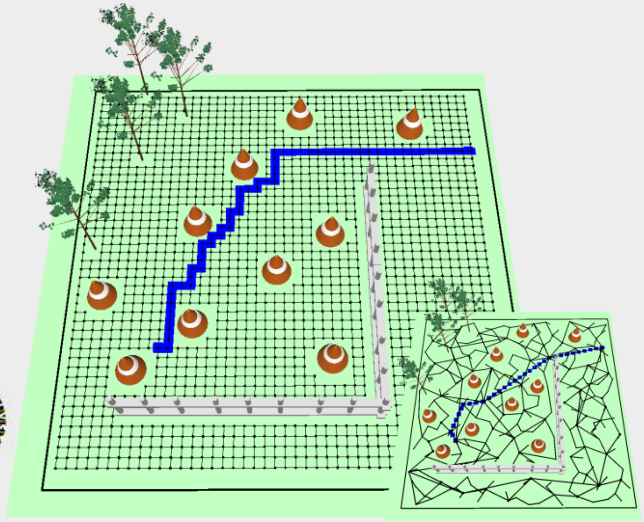
Challenge: Realize a synthesis of *dispersed models* which unifies *realistic,* individual, small group, and collective human movements in *interactive, heterogeneous* environments.

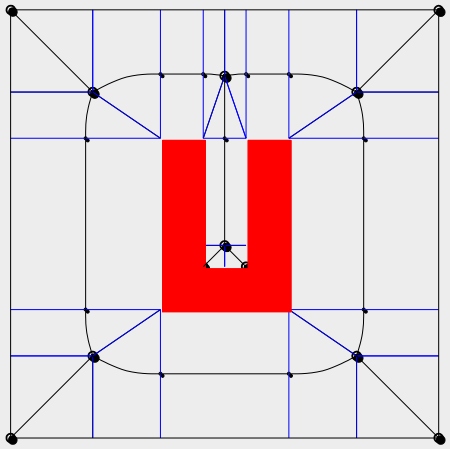
- Dispersed models
  - Agent-based: individuals, but problems with high densities
  - Flow-based: no individuals, but good for high densities
- Realistic movements
  - Comprise collaboration, smooth and energy-efficient movement, collision avoidance, and dealing with unrealistic congestions.
- Interactive environment
  - Geometry can change dynamically, and the crowd has to react.
- Heterogeneous environment
  - People need to take logical, distinct, and realistic paths over heterogeneous terrains in the environment.

Are we there yet?

Surface-based navigation

- We need a *paradigm shift*
  - from graph-based to surface-based navigation
- Graph-based navigation: little support for route deviation
  - Hard to avoid expected collision between humans
  - Hard to support differently sized humans/groups
  - Costly to deal with dynamic changes in the environment
  - Hard to efficiently deal with heterogeneous regions
  - Human navigation is surface-based

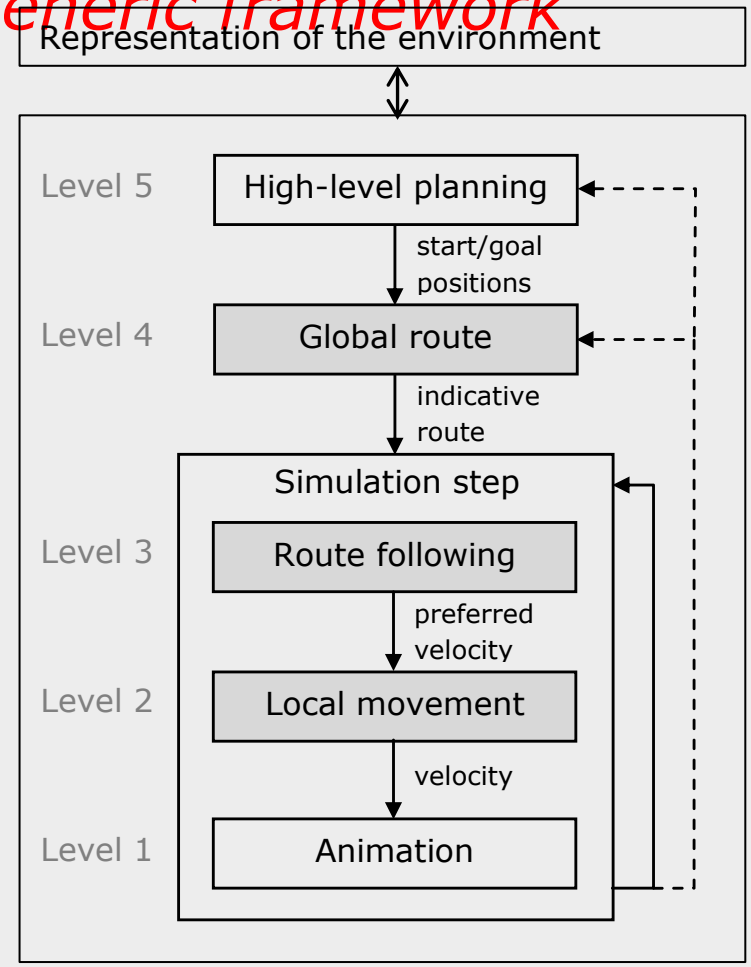




Crowd simulation framework

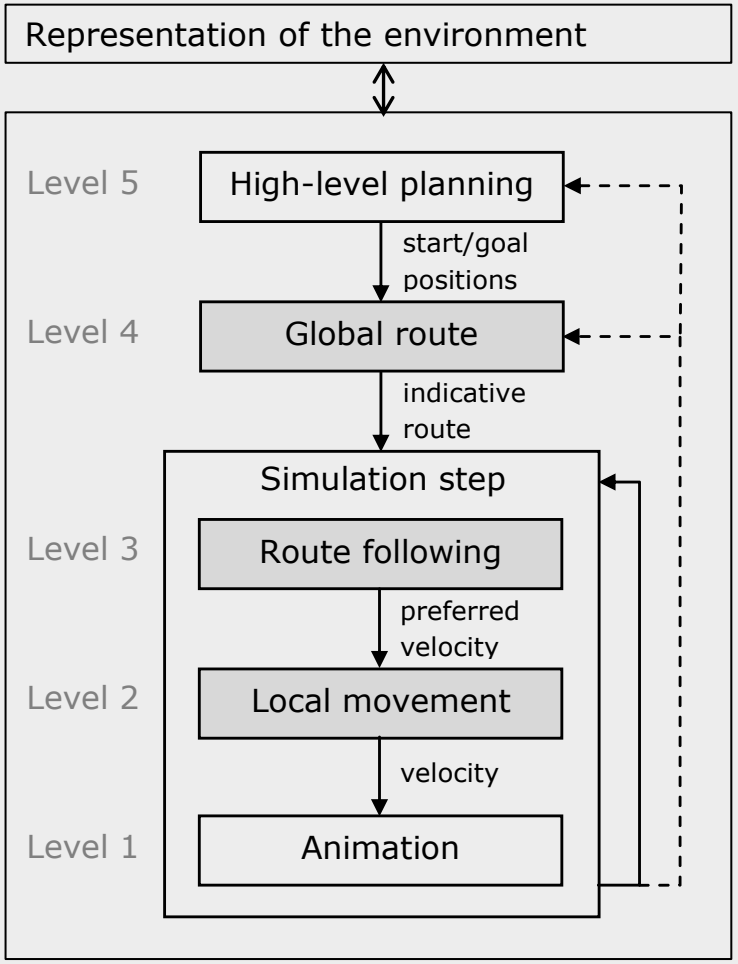
We need a fast and generic framework

- Representation environment
- Level 5
  - Plans actions
- Level 4
  - Creates indicative routes
- Level 3
  - Traverses the routes
  - Yields speed/direction pairs
- Level 2
  - Adapts routes
  - E.g. to avoid collisions
- Level 1
  - Moves the characters



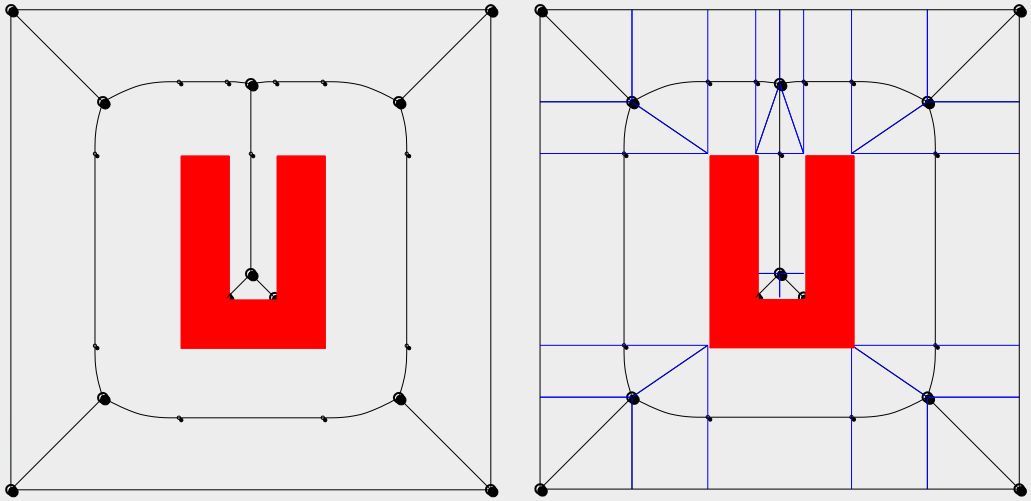
Crowd simulation framework

- Representation environment
- Level 5
  - Plans actions
- Level 4
  - Creates indicative routes
- Level 3
  - Traverses the routes
  - Yields speed/direction pairs
- Level 2
  - Adapts routes
  - E.g. to avoid collisions
- Level 1
  - Moves the characters



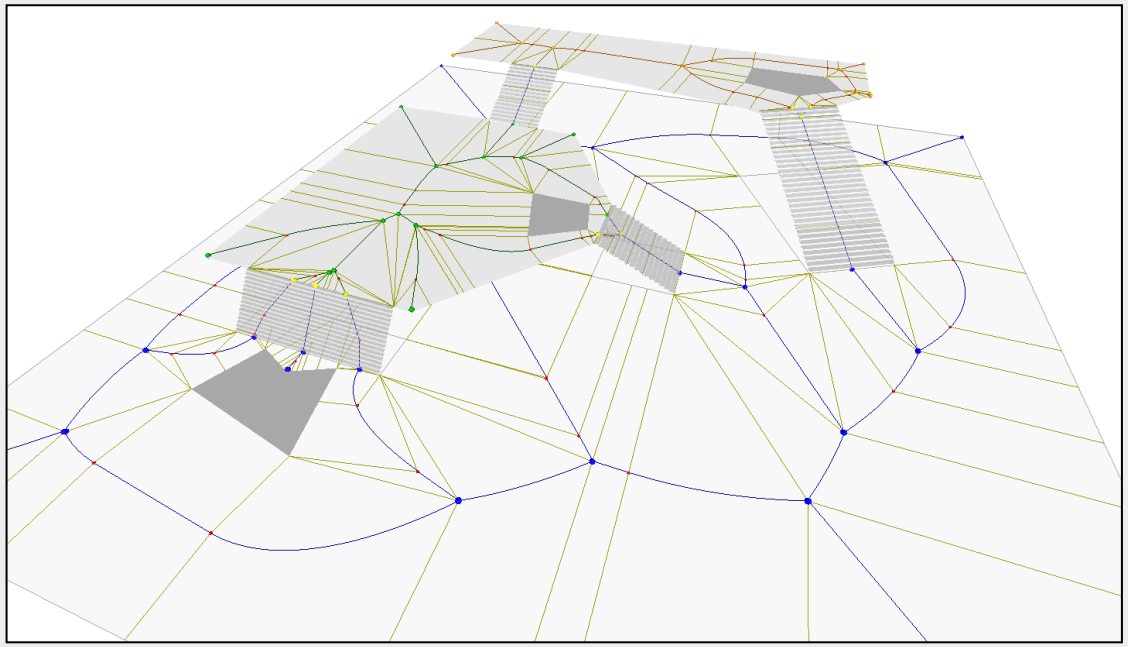
Representation of the traversable environment

- Explicit Corridor Map
  - Navigation mesh
    - Medial axis
    - Closest point annotation



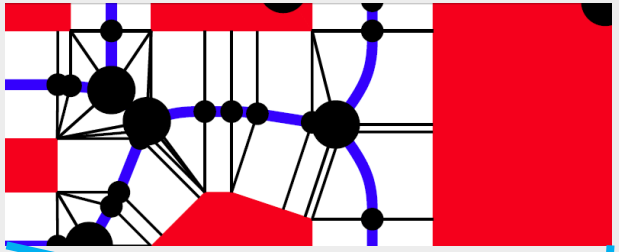
Representation of the traversable environment

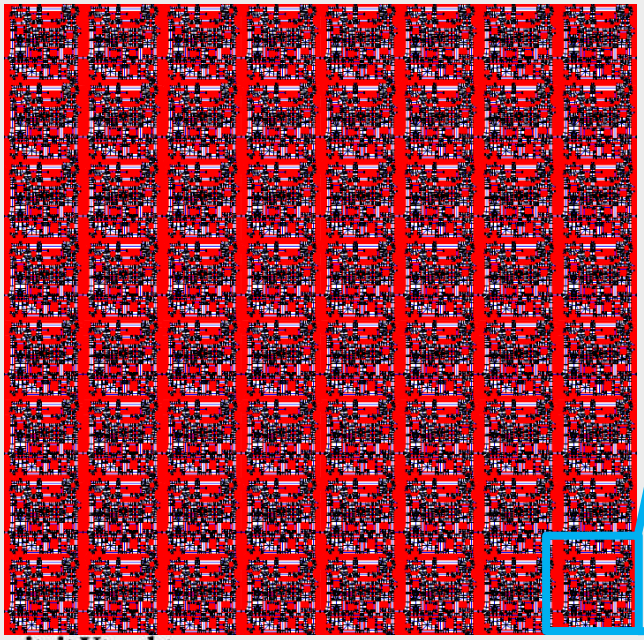
- Explicit Corridor Map
  - Extendible to multi-layered 3D

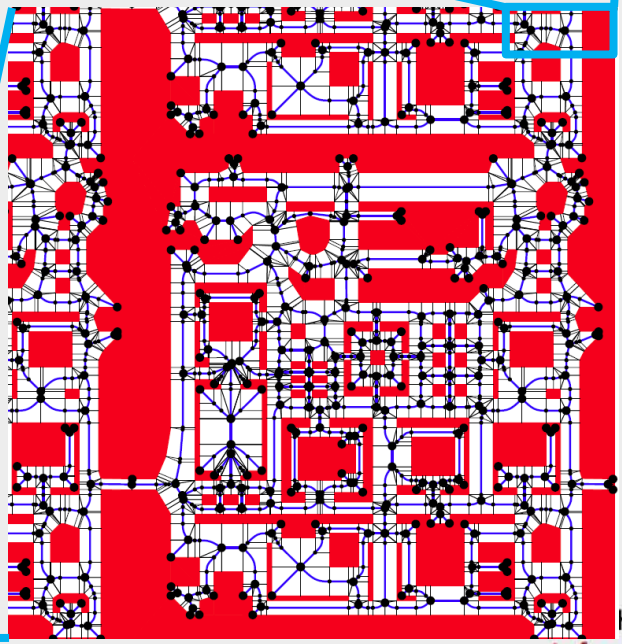


Representation of the traversable environment

- Explicit Corridor Map
  - Supports large environments
  - High precision





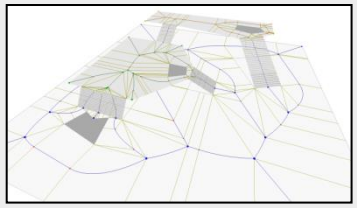


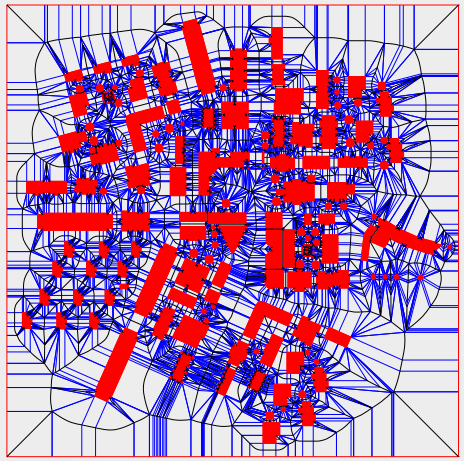
Universiteit Utrecht

happen
Informatica]

Representation of the traversable environment

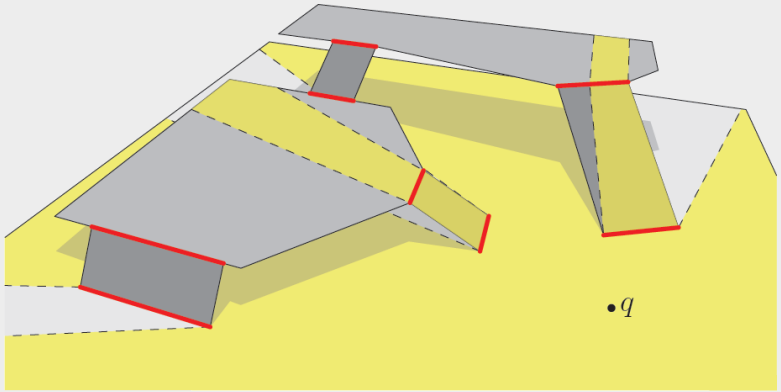
- Explicit Corridor Map
  - Fast to compute
    - 10 ms vs 115 ms

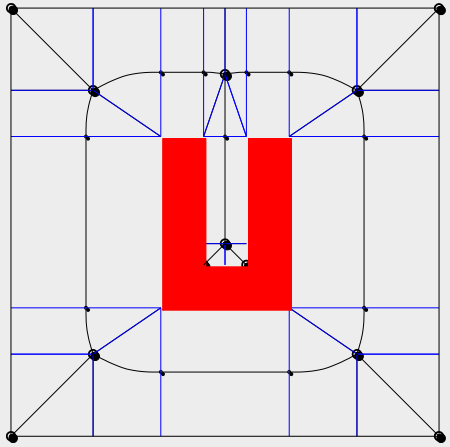




Representation of the traversable environment

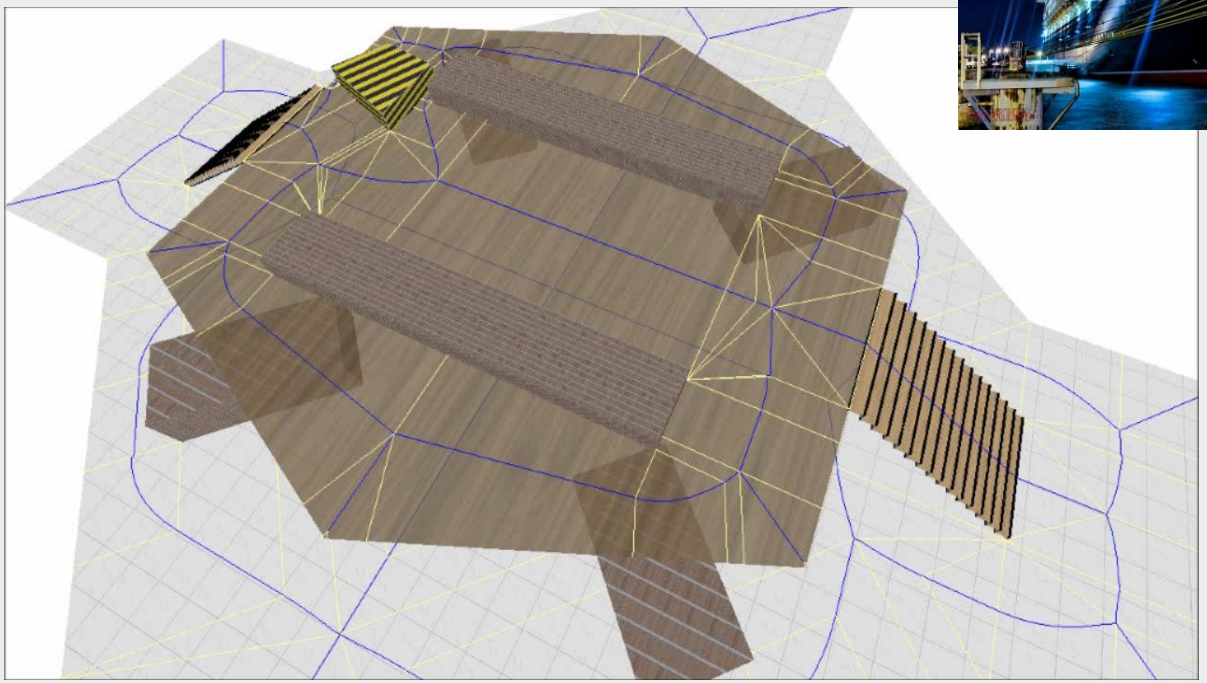
- Explicit Corridor Map
  - Exact representation
    - Captures all homotopically different routes (cycles)
    - Captures 100% of the free space
  - Allows fast extraction of global routes and final paths
  - Nice mathematical properties
    - Fast to compute – O(n log n)
    - Small data structure – O(n)
    - Nearest obstacle computation – O(1)
    - 2D algorithms also work in ML environments





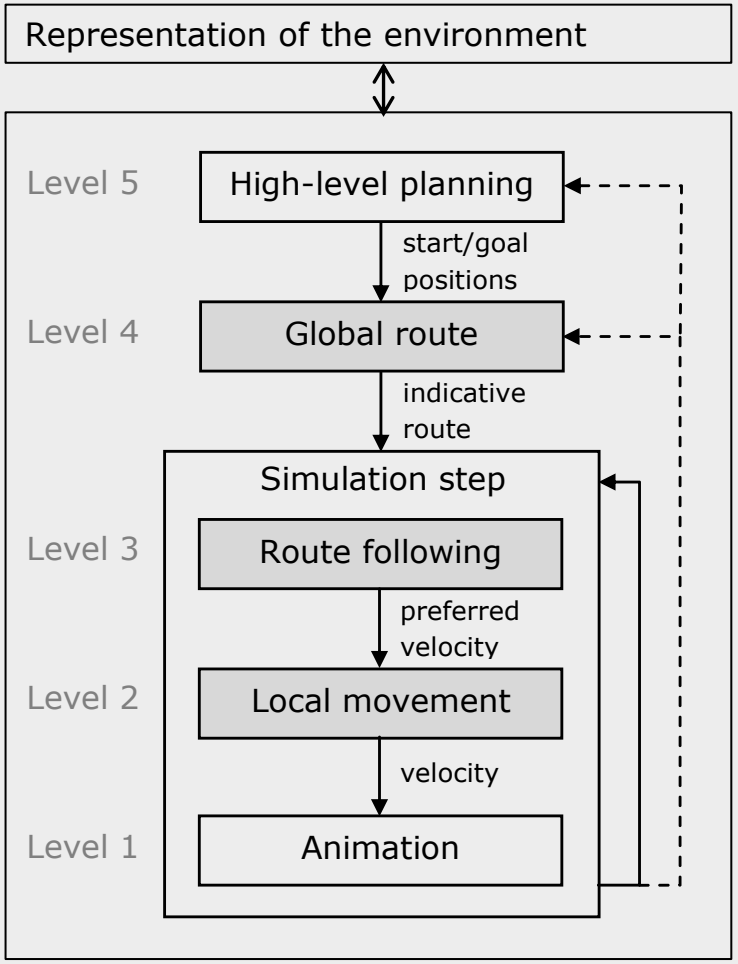
Representation of the traversable environment

- Explicit Corridor Map
  - Handling dynamic changes
    - Update costs < 1 ms



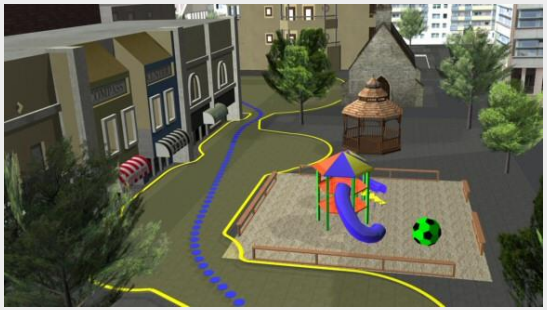
Crowd simulation framework

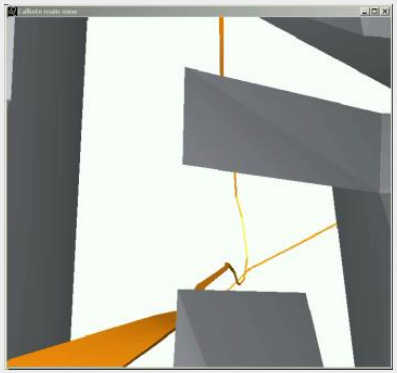
- Representation environment
- Level 5
  - Plans actions
- Level 4
  - Creates indicative routes
- Level 3
  - Traverses the routes
  - Yields speed/direction pairs
- Level 2
  - Adapts routes
  - E.g. to avoid collisions
- Level 1
  - Moves the characters

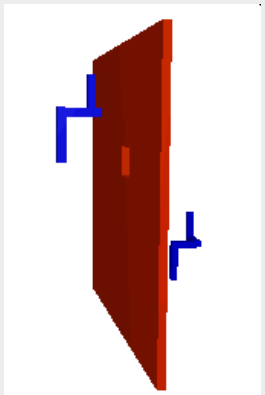


Indicative Routes

- A path planning algorithm should NOT compute a path
  - A one-dimensional path limits the character's freedom
  - Humans don't do that either
- It should produce
  - An Indicative/Preferred Route
    - Guides character to goal

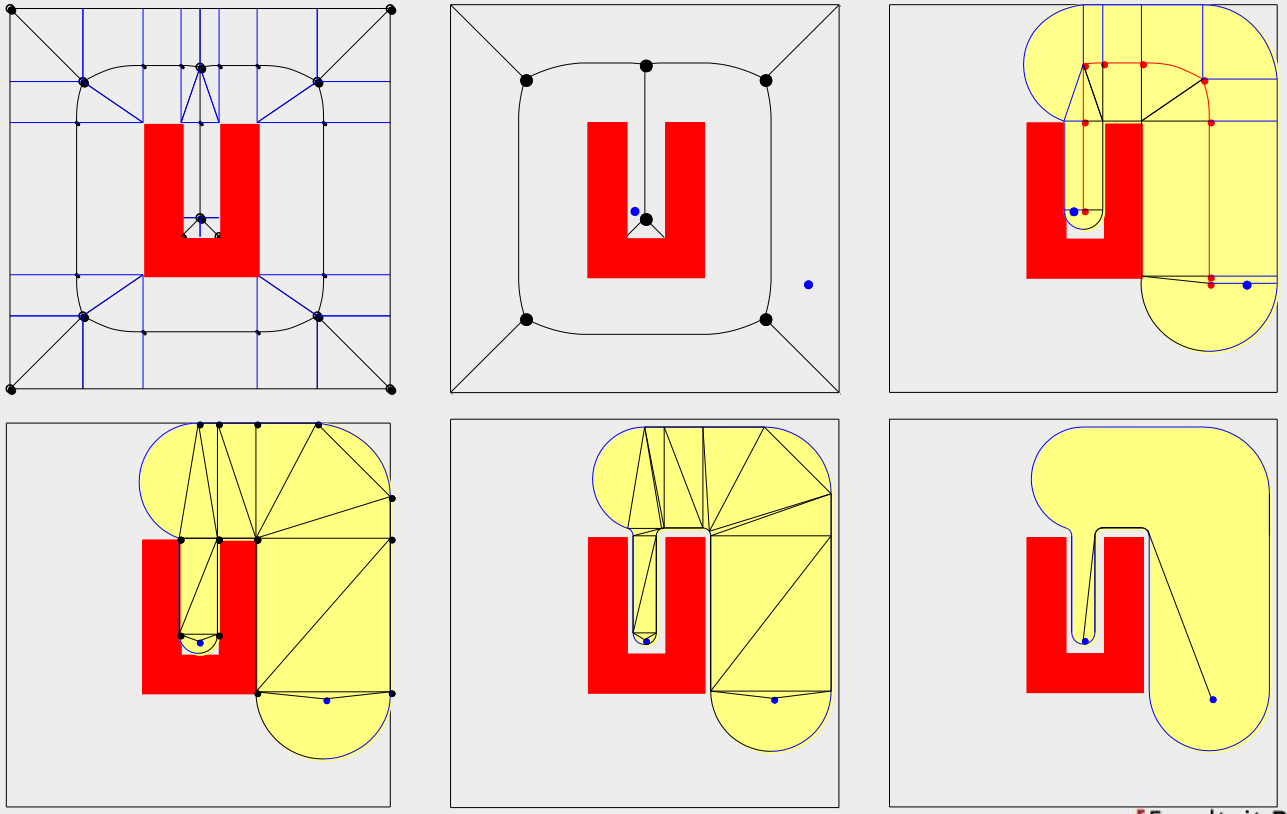


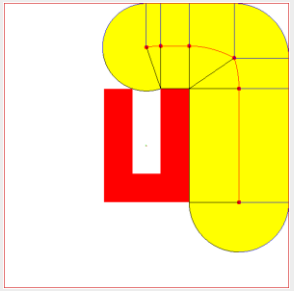




Computing Indicative Routes

- Shortest path with clearance

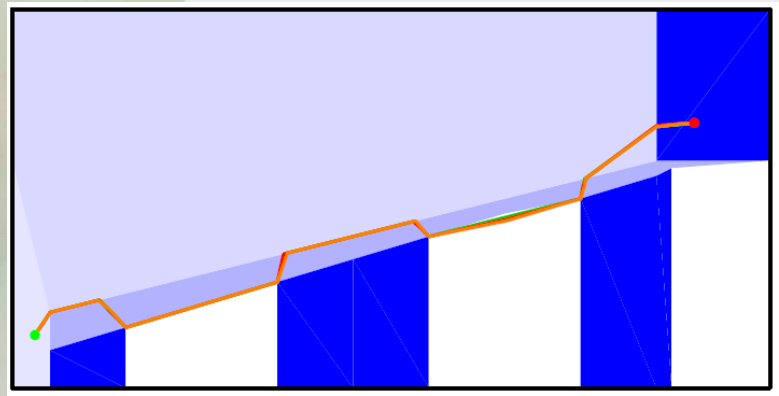




A single navigation mesh supports multiple character widths

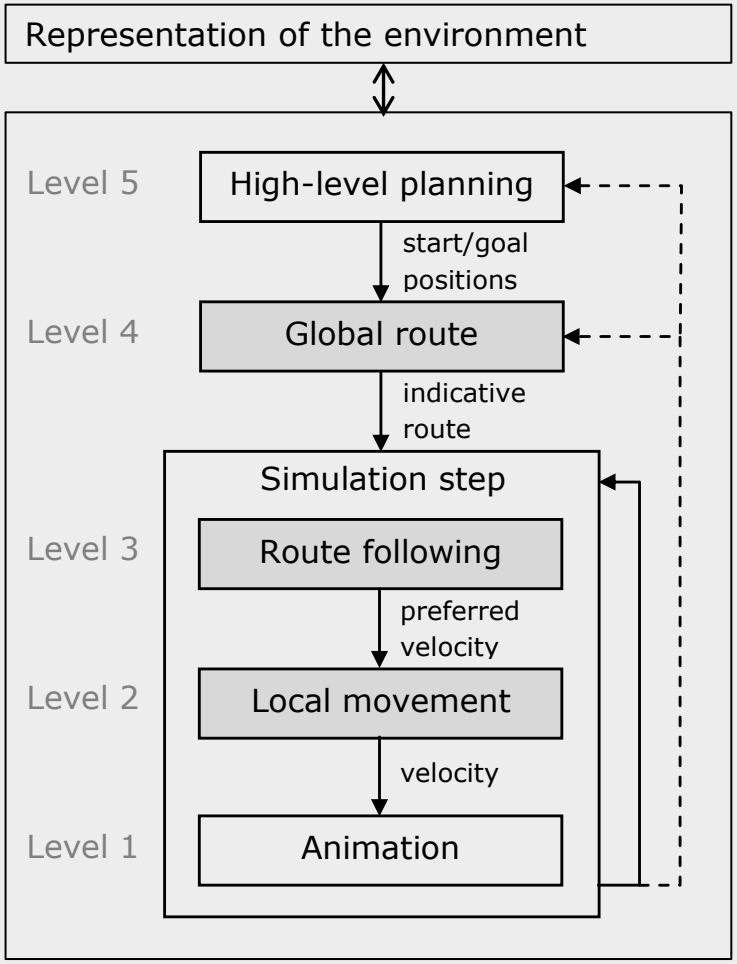
Computing Indicative Routes

- What about weighted regions in the plane?
- Compute the shortest path
  - Unsolvable in the Algebraic Computation Model over the Rational Numbers
  - Approximation algorithms



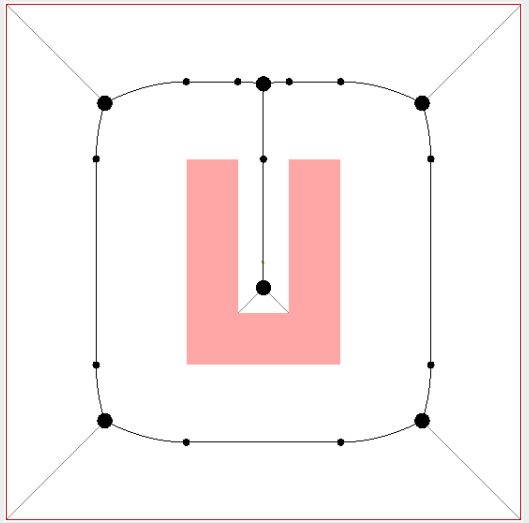
Crowd simulation framework

- Representation environment
- Level 5
  - Plans actions
- Level 4
  - Creates indicative routes
- Level 3
  - Traverses the routes
  - Yields speed/direction pairs
- Level 2
  - Adapts routes
  - E.g. to avoid collisions
- Level 1
  - Moves the characters



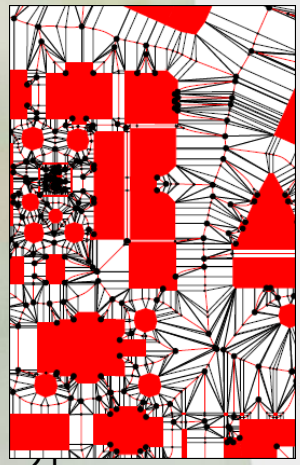
Traversing the Routes: The Indicative Route Method

- "Algorithm"
  - Compute a collision-free indicative route from A to B
  - Compute a corridor containing the route
    - Provides a global route
    - Allows for flexibility
  - Move an attraction point along the indicative route
    - The attraction point attracts the character
    - The boundary of the corridor pushes it away
    - Other characters and local hazards push the character away
  - Integrate the forces (twice) over time

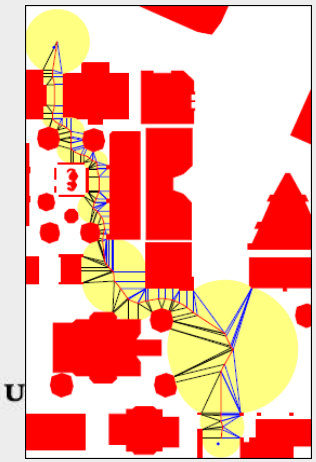


Traversing the Routes: The Indicative Route Method

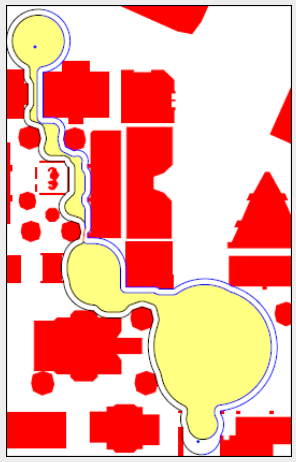
- Example: a short path with clearance
- Follow the path and smooth it
- Results (query time)
  - <1 ms



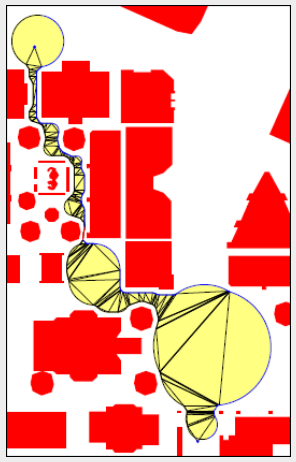
ECM (0.1s)



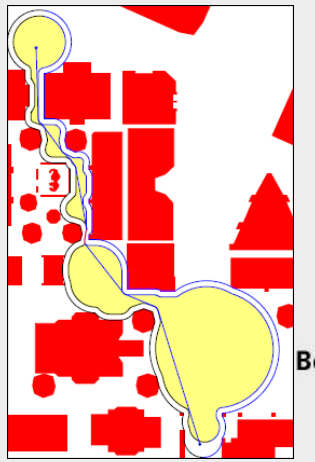
Explicit corridor



Shrunk corridor



Triangulation

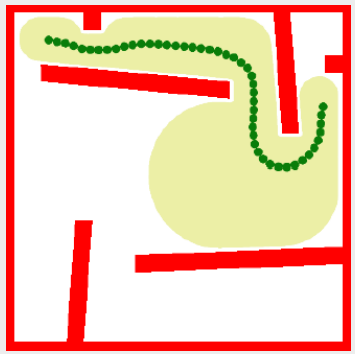


Shortest path

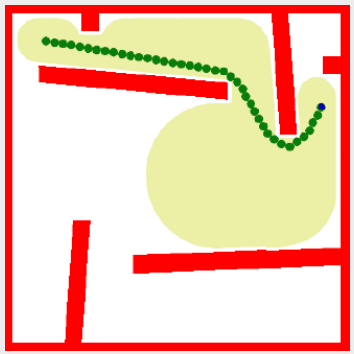
Smooth path

Traversing the Routes: The Indicative Route Method

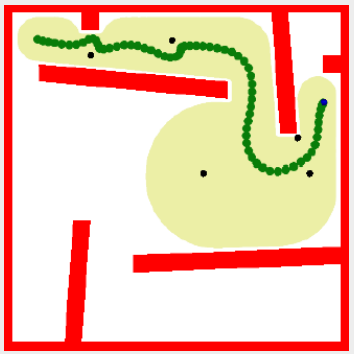
- The Indicative Route Method: examples
  - Adding/changing forces leads to other "behavior"



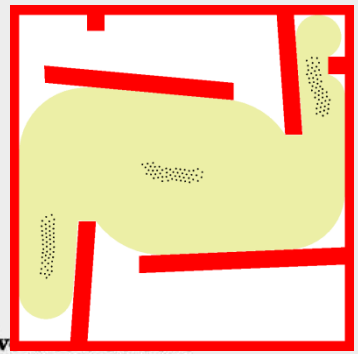
Smooth path



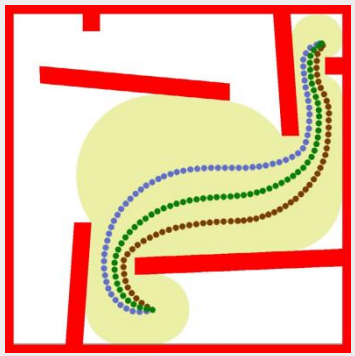
Short path



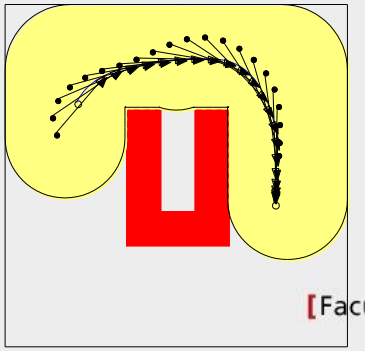
Obstacle avoidance



Coherent groups



Path variation



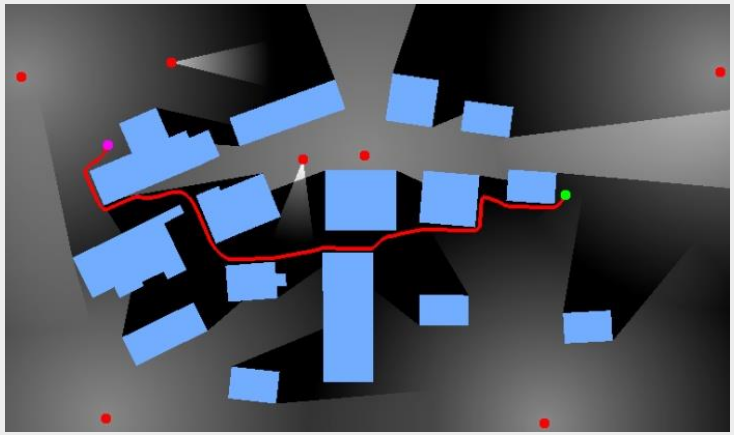
Camera path

Traversing the Routes: The Indicative Route Method

- The Indicative Route Method: examples
  - Adding/changing forces leads to other "behavior"

Stealth-Based Path Planning in Virtual Environments

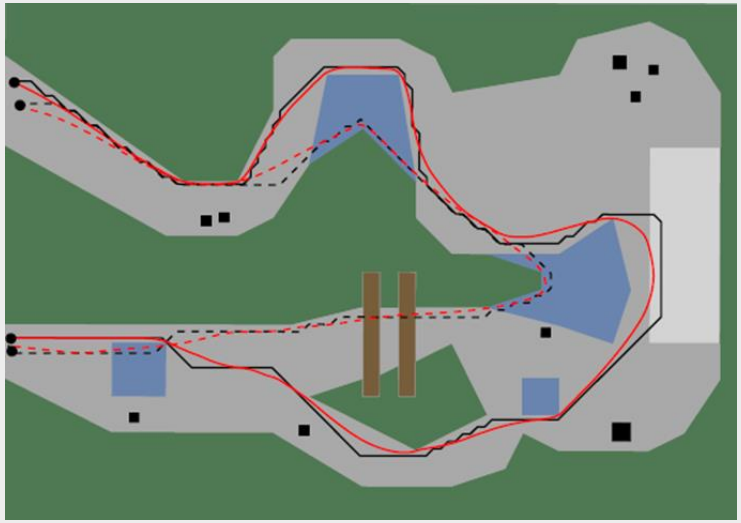
In this movie, a character tries to limit exposure to the 128 moving observers.

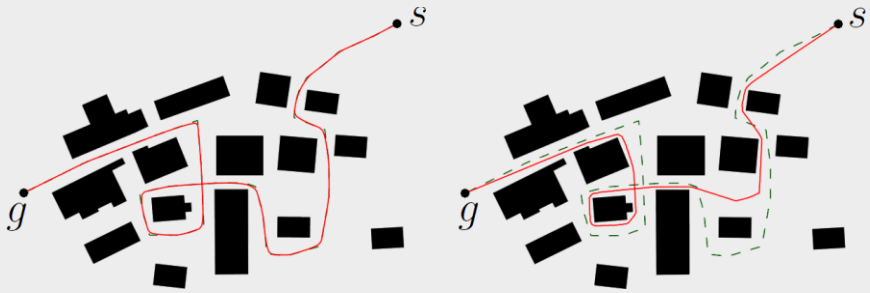


Stealth-based path planning

Traversing the Routes in weighted regions

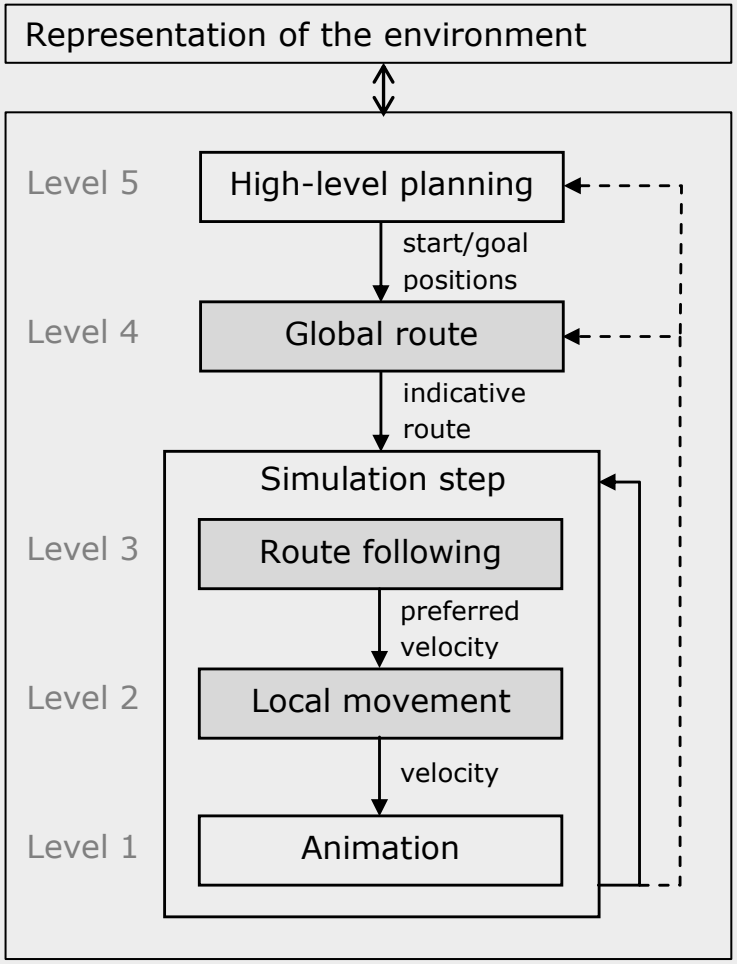
- Modified Indicative Routes And Navigation (MIRAN)
- The MIRAN method supports
  - heterogeneous terrains
  - separate character profiles
  - customized smoothing





Crowd simulation framework

- Representation environment
- Level 5
  - Plans actions
- Level 4
  - Creates indicative routes
- Level 3
  - Traverses the routes
  - Yields speed/direction pairs
- Level 2
  - Adapts routes
  - E.g. to avoid collisions
- Level 1
  - Moves the characters

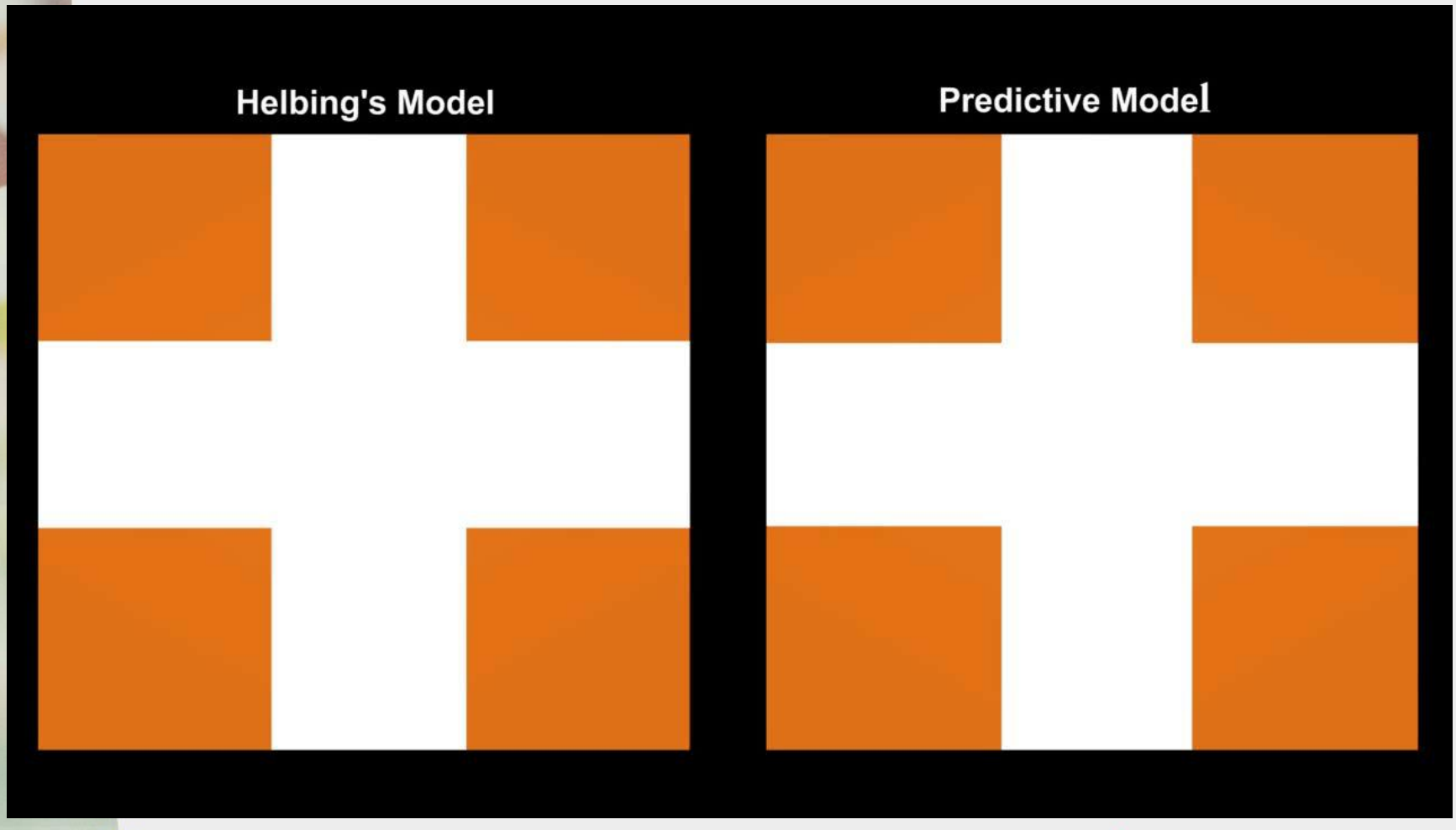


Adapt Routes: Collision avoidance

- What is realistic collision avoidance?

Smack the pony s01x02

Adapt Routes: Improved collision-avoidance model



Helbing and Molnár, 1998: Social Force Model for Pedestrian Dynamics
Karamouzas et al, 2009: A Predictive Collision Avoidance Model for Pedestrian Simulation

Adapt Routes:
Collision avoidance: predictive model

Karamouzas et al, 2009: A Predictive Collision Avoidance Model for Pedestrian Simulation

Adapt Routes: Collisions avoidance: small groups

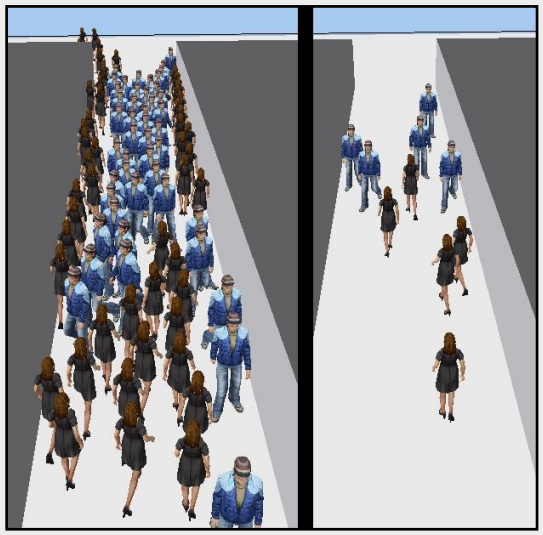
- Also allow speed changes
- Deal with small groups

Karamouzas et al, 2010: Simulating Human Collision Avoidance Using a Velocity-Based Approach
Karamouzas et al, 2010: Simulating the Local Behaviour of Small Pedestrian Groups

Unification of individual and collective movements

- Our stream-based model allows local coordination, based on a character's *incentive*
  - Deviation from the local flow
  - Local density
  - Internal motivation
  - Spent time to reach goal

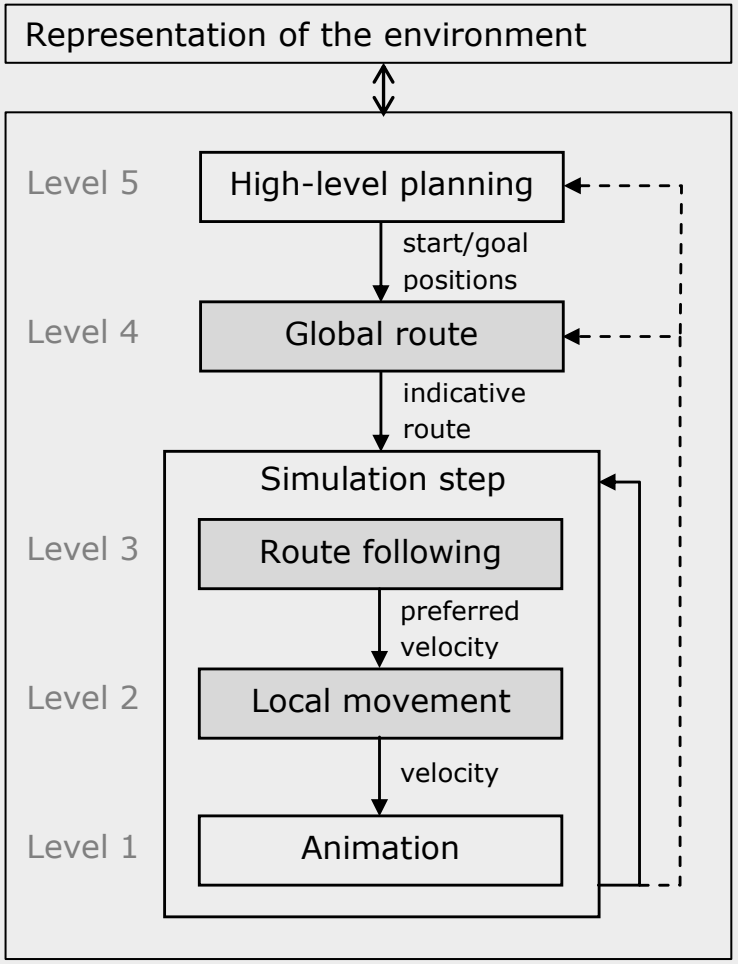
Only collision-avoidance



Hi-density streams *Low-density streams*

Crowd simulation framework

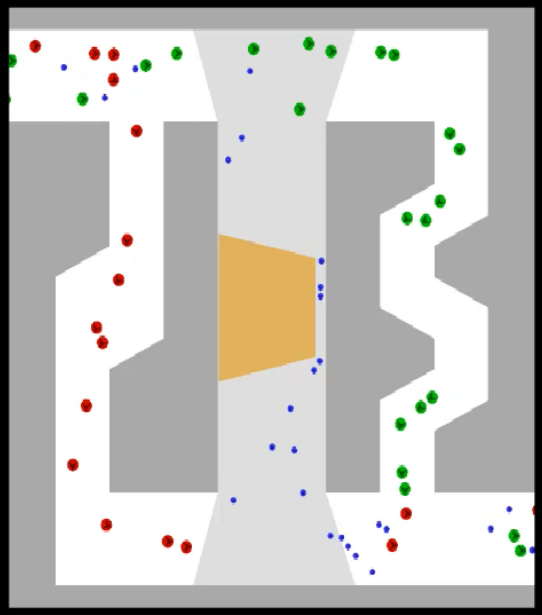
- Representation environment
- Level 5
  - Plans actions
- Level 4
  - Creates indicative routes
- Level 3
  - Traverses the routes
  - Yields speed/direction pairs
- Level 2
  - Adapts routes
  - E.g. to avoid collisions
- Level 1
  - Moves the characters



Crowd simulation

- Dynamic updates of the crowd

We implement our algorithms in the **Explicit Corridor Map** framework.
This framework can now model various **dynamic** crowd behaviors.



In this example, all characters follow the shortest path (the middle corridor).

We will now insert an obstacle in the middle.

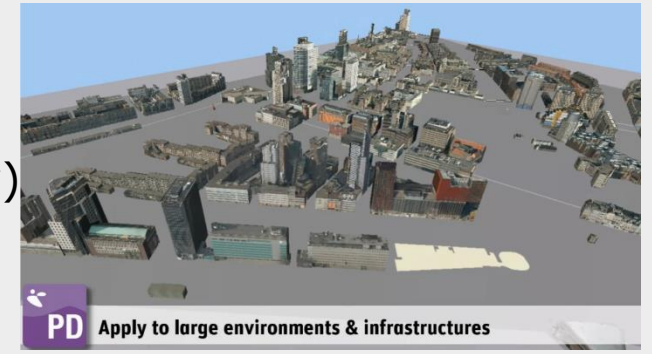
Blue characters are **small enough** to move along the obstacle.

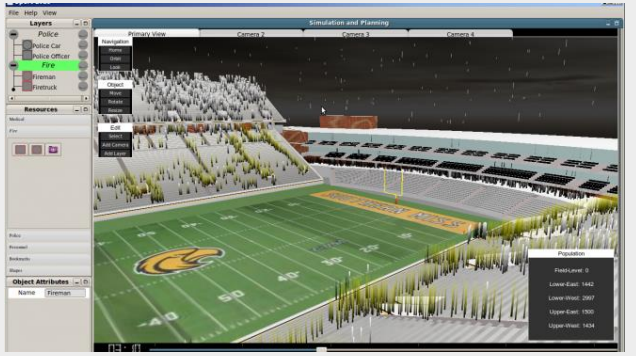
Red characters have **perfect knowledge**.
They know that the left route is the shortest.

Green characters try to go through the middle.
They re-plan when they **see** the obstacle.
By that time, the right route is the shortest.

Crowd simulation software package

- Properties
  - Most research has been integrated
  - Efficient
  - Easily extendible
  - Clear interface and documentation
- Licensed to companies
  - Pedestrian Dynamics
  - SportEvac (NCS4, Homeland Security)
    - Training, education, scenarios, …
  - Queensday, stadiums, Efteling,…
- Availability
  - Free for researchers
  - Can be licensed to companies
- Demo





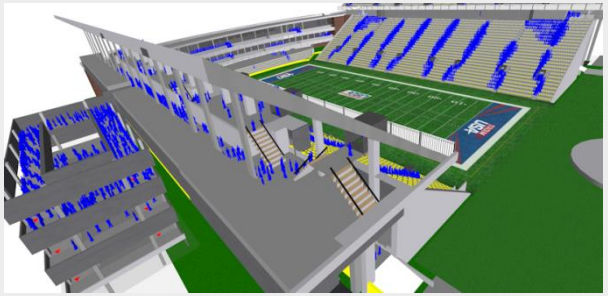
Universiteit Utrecht

Messages

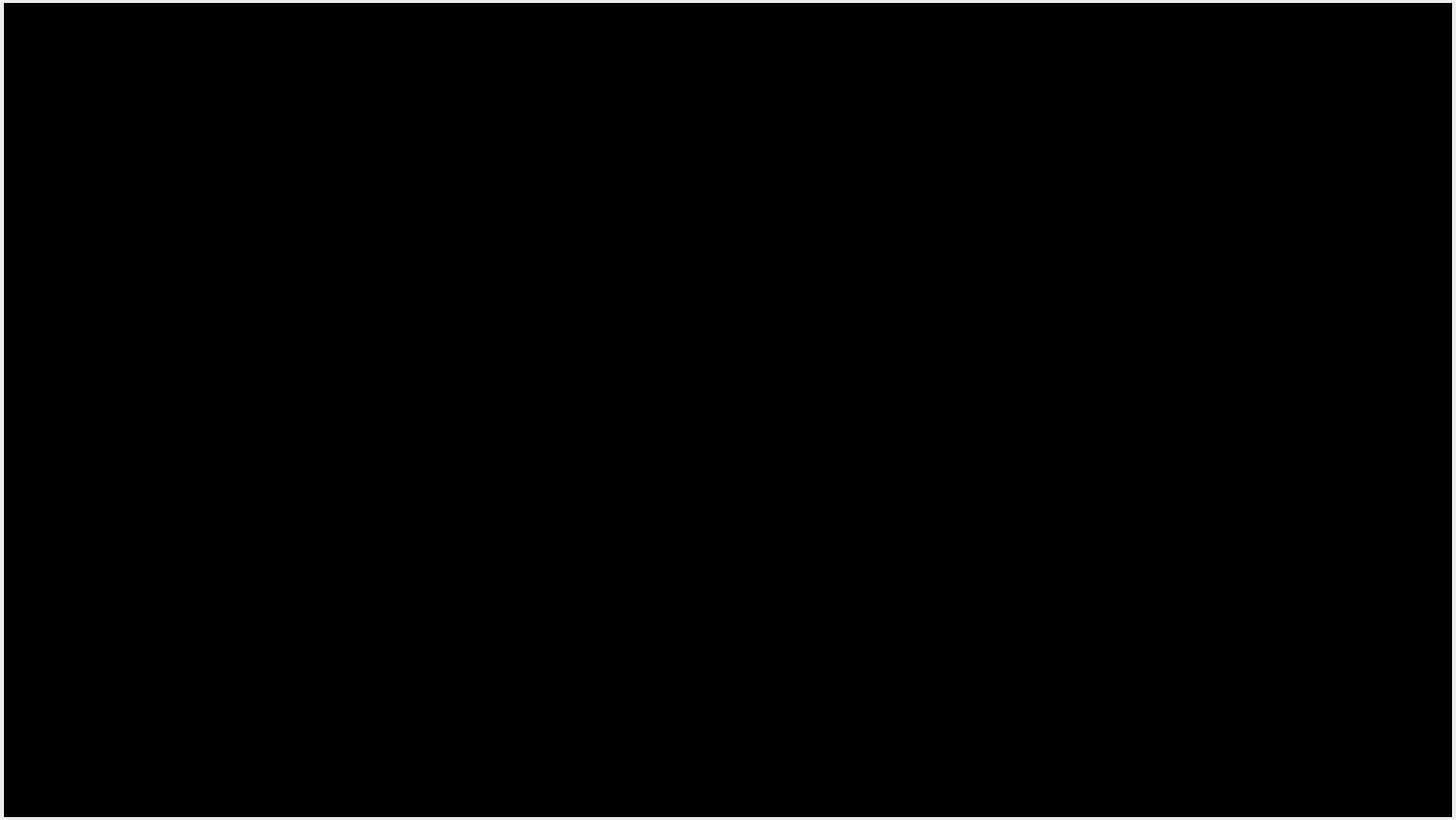
- For efficiently and flexibly simulating crowds, we need
  - a generic and efficient representation of the navigable areas;
  - a framework of (at least) 5 complexity levels.
- Methods must be compatible with surface-based navigation at all levels (paradigm shift!)
  - so a graph-based approach is not going to be sufficient
- A path planning algorithm should not compute a path
- Our simulation software is freely available for researchers

List of contributors

- Staff
  - *Roland Geraerts*
  - Marjan van den Akker
  - Han Hoogeveen
  - Mark Overmars
  - Frank van der Stappen
- Companies
  - GreenDino
  - InControl
  - NCS4
- PhD students
  - Arthur van Goethem
  - Ioannis Karamouzas
  - Wouter van Toll
  - Arne Hillebrand
  - Norman Jaklin
- MSc students
  - Corien Prins
  - Eric Schrager
  - David Weterings



So what *is* realistic collision avoidance?



Rush Hour; by Fernando Livschitz

Contact

- Roland Geraerts
- The Netherlands, Utrecht, Princetonplein 5, BBL 4.18
- www.cs.uu.nl/~roland
- R.J.Geraerts@uu.nl