De mens in mensenmassa’s
Het complexe samenspel van individu en collectief

Roland Geraerts
Universitair docent

Eric de Wilde
CEO

Ron Looy
Projectleider
Societal relevance of simulation

• The number of environments with big crowds are growing
• Questions
  – In how much time can a train station be evacuated?
  – Where and how can potential dangerous situations appear?
  – How can a city accommodate 0.5M people during an event?
  – How can we populate a game world with a believable crowd?

Love Parade, 2010
21 deaths, 510 injuries
Real-time, interactive crowd simulation

UU Crowd Simulation R&D
Unity3D Plugin

Utrecht University
How can you simulate a human crowd interactively?
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 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

From 3D geometry to a navigation mesh
But, this doesn’t this already exist?

- There are standards such as BIM, CityGML,…
  - Not common practice, many geometric errors
- Current solutions make approximations and errors
From 3D geometry to navigation mesh

- Goal: extract the walkable areas exactly
  - Input environment

From 3D geometry to navigation mesh

- Goal: extract the walkable areas \textit{exactly}
  - Remove (annotate) steep polygons
From 3D geometry to navigation mesh

- Goal: extract the walkable areas *exactly*
  - Cut out polygons giving headaches
  - Resolve degeneracies
  - Resolve intersections
From 3D geometry to navigation mesh

• Goal: extract the walkable areas $exactly$
  – Simplify triangulations
From 3D geometry to navigation mesh

• Goal: extract the walkable areas *exactly*
  – Separate into 2D (projectable) layers
From 3D geometry to navigation mesh

• Goal: extract the walkable areas exactly
  – Resolve gaps

From 3D geometry to a navigation mesh that has nice properties and can be queried fast.
From 3D geometry to navigation mesh

What is the best representation for the walkable space of a *multi-layered 3D* environment?

- Compute a 2D navigation mesh per layer
- Stitching the navigation meshes
From 3D geometry to navigation mesh

• Favorable properties
From 3D geometry to navigation mesh

• Large environments are processed within 1 second

From 3D geometry to navigation mesh

- Handles dynamic updates
From navigation mesh to simulation of 1 pedestrian
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
Action planning

• Splits up a task into geometric queries
  – Example: dynamic updates of the crowd

*Standard behavior* pedestrians take the same terminal

*Improved behavior* pedestrians distribute among all terminals

M. Koenis, 2016: Impact of Pedestrians Bringing Along Their Bicycles on Evacuation Times of Subway Stations
Action planning

- Splits up a task into geometric queries
  - Example: dynamic updates of the crowd

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 agent’s freedom
  - Humans don’t do that either

- It should produce
  - An Indicative/Preferred Route
  - A corridor around this route
Computing Indicative Routes

- Example: shortest path with clearance to obstacles

Jaklin et al, 2014: Computing High-Quality Paths in Weighted Regions
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

Following routes

• Basic algorithm
  – An attraction point on the indicative route guides the pedestrian to its goal
  – Obstacles repulse pedestrians when they are too close
From simulation of 1 pedestrian to a crowd
• 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

What is realistic collision-avoidance behavior?
What is realistic collision-avoidance behavior?

Crowd prank in Japan
Adapting the routes: Collision avoidance

- Our model is derived from experiments in the MOCAP lab
Adapting the routes: Collision avoidance

- Our model slightly adjusts the people’s movements

Karamouzas et al, 2009: A Predictive Collision Avoidance Model for Pedestrian Simulation
Adapting the routes: Social groups

- The group members stay close and visible to each other
Adapting the routes: Moving through a dense crowd

- People can make room for a passing individual

Stüvel et al, 2017: Torso crowds
Adapting the routes: Unification of individual and collective movements

• Our stream-based model allows local coordination, based on a agent’s incentive
  – Deviation from the local flow
  – Local density
  – Internal motivation
  – Spent time to reach goal

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

Current developments

- Real-time crowd prediction, analysis and decision support
  - A sensing system computes the pedestrians’ positions
  - This calibrates the simulation in real-time with the real world
  - Makes predictions of the coming minutes
  - May run 24/7
  - Prevents unsafe situations and make the city / station safer
  - Special attention is paid to preserving privacy and complying with ethical requirements set by society
Software
Software package

• Core engine in C++
• Runs on 64bit Windows
  – Linux, MacOS, iOS
• Also available as a plugin for Unity3D
  – https://ucrowds.com/documentation/unity3d/
• To obtain a license, send a request
  – Our startup
  – info@ucrowds.com
Applications
Optimizing crowd flows

Tour de France
Optimizing of outdoor area layout

Utrecht Stationsplein
Evacuation studies (with bicycles)

Metro stations before operation
Conducting what-if scenarios

Rail at transport hub
Tangible interaction

Education and training
Public engagement
Contact

Website: http://www.cs.uu.nl/~roland
E-mail: R.J.Geraerts@uu.nl
LinkedIn: http://www.linkedin.com/in/rolandgeraerts
Portfolio: www.youtube.com/user/drRolandJan
Tel.: +31628804901
Address: Princetonplein 5, Utrecht, room BBG 4.18

UCrowds

Website: www.ucrowds.com
E-mail: contact@ucrowds.com
Twitter: @u_crowds
LinkedIn: https://www.linkedin.com/in/ucrowds/
Portfolio: https://tinyurl.com/u-crowds
Address: Padualaan 8, Utrecht, room W125