Modeling and Simulation of Digital Humans:
From Individuals to Crowds

Ming C. Lin
Department of Computer Science
University of Maryland at College Park

http://www.cs.umd.edu/~lin
http://gamma.umd.edu

Digital Humans

ViCrowd [Musse & Thalmann 01; EPFL]
ABS [Tecchia et al. 01; UCL]
Virtual Iraq [ICT/USC 06]
Disaster Response [ICT/USC]
Applications

- (a) Games
- (b) Crowds
- (c) Movie
- (d) Robot
- (e) CAD/Human factor
- (f) ergonomics

Motivation

- Dynamic simulation of deformable solids
- Highly detailed surface geometry
- Large contact area: objects bounce, roll, slide,…

[Galoppo et al; Eurographics 2007]
Motivation: Layered Models

- Detailed, small-scale deformations
- Global (skeletal) deformations
- Dynamic interplay between skeletal motion and surface deformation during contact

Modeling Soft Articulated Bodies in Contact Using Dynamic Deformation Textures

Global deformations ↔ Detailed deformations

[Galoppo et al; SCA 2006]
Motivation

- Human-like characters are widely used in computer animation and virtual environments
- Synthesize natural-looking human motion
  - Key-frame animation methods are tedious and need considerable input from the animators
  - Current automated tools are open limited to simple or open environments
Crowd Simulation

- Simulating movement of a large number of agents to replicate collective behavior

Agents in Games

Interactive simulation of virtual agents

Assassin’s Creed
Second Life
Spore
Planning & Architectural design

- Stampedes at Hajj
- Improvements to the Jamarat Bridge

Large, Dense Crowds

- Commonplace occurrences
- Greater safety risks
  - Crowd panic

(Top) Iraq war protest: Broadway
(Bottom) Presidential swearing in
©MSNBC
Challenges

• Realistic human locomotion behaviors
• Simulating very large, dense crowd
• Control and direct crowd flows
• Modeling the interaction with and due to traffic flows & vehicles

Outline: Crowd Simulations

• “Principle of Least Efforts” Navigation
• Modeling of Dense Crowd
• Control and Direction Crowds
Outline: Crowd Simulations

• “Principle of Least Efforts” Navigation
• Modeling of Dense Crowd
• Control and Direction Crowds

Principle of Least Efforts

• Our hypothesis: Effort = Biomechanical Energy
• Imperially measured as function of speed [Whittle, 2007]
  \[ E = m \int (e_s + e_w |v|^2) dt \]
  \( e_s \) - energy when still
  \( e_w \) - energy at speed
  which we seek to minimize
Algorithm Overview

1. Determine goal position
2. Find permissible (non-colliding) velocities (PV)
3. Choose velocity $\in PV$ with minimum energy

Optimal Reciprocal Collision Avoidance (ORCA) \([\text{ISRR2009}]\)

- A new algorithm for collision avoidance
- A linear programming based formulation
- Scalable approach to collision avoidance
  - From two agents to thousands of agents
- Extends Velocity Obstacle concepts
  - Decentralized planning
  - Decisions are made \textit{independently}, with no communication nor assumptions of the motion
  - Sufficient conditions for avoiding collisions
Problem Overview

• Inputs:
  • Independent Agents
  • Current Velocity of all
  • Own Desired Velocity ($V^{\text{pref}}$)

• Outputs:
  • New n-way collision-free velocity ($V^{\text{out}}$)

• Description – Each Agent:
  • Determines permitted (collision free) velocities
  • Chooses velocity closest to $V^{\text{pref}}$ which is permitted

Velocity Space & Forbidden Regions

• Forbidden Regions
  • Potentially Colliding Velocities
  • An “obstacle” in velocity space

• VO: Velocity Obstacle [Fiorini & Shillier 98]
  • Assumes other agent is unresponsive
  • Appropriate for static & unresponsive obstacles

• RVO: Reciprocal VO [Berg et al., 08]
  • Assumes other agent is mutually cooperating
Time Horizon

- Ignore collisions more than $\tau$ seconds away
- Diagram of $\tau$ adjusted $VO - VO^T_{A|B}$

ORCA

- $u$ – Vector which escapes $VO^T_{A|B}$
  - Each robot is responsible for $\frac{1}{2}u$
- $\text{ORCA}^T_{A|B}$
  - The set of Velocities allow to $A$
  - Sufficient condition for collision avoidance if $B$ chooses from $\text{ORCA}^T_{A|B}$
Multi-Robot Navigation

- Choose a velocity inside ALL pair-wise ORCAs
- Efficient O(n) implementation w/ Linear Programming

Estimating Least Effort

- Evaluate potential paths based on least effort
- Approximate total effort in a greedy piecewise fashion
- Compute optimum in velocity space using linear programming
Summary

- Powerful and simple (easy to implement) navigation method for multi-agent simulations
- Allows for easy integration with global planning, kinodynamic constraints, visibility constraints, etc.
- \textit{Scalable with number of agents and number of cores used}
- Application to Behavior Modeling & Crowd Simulation
Outline: Crowd Simulations

- “Principle of Least Efforts” navigation
- Modeling of Dense Crowd
- Control and Direction Crowds

Dense Crowds

(Top): Obama campaign rally © The Telegraph
(Bottom): Subway Station, Beijing © ABC
(Top): Al-Masjid Al-Haram, Mecca © SacredSites
(Bottom): Carnivale, Milan © Dan, Picasa
Challenges

- Human behavior is complex
  - Avoid other people
- Complex emergent behavior

Challenges

- Parameters
  - Scene complexity
  - Crowd distribution
- Density dependent behaviors

http://bobscrafts.com/bobstuff/maze.htm

Shibuya crossing © NextStop
Large, Dense Crowds

- Per-agent Local navigation expensive
  - Large number of possible collisions
  - Continuous collisions
- Becomes computational bottleneck
- Infeasible to simulate large crowds

Density-dependent Behavior

- Low density
  - Similar to gases
- Medium density
  - Fluid flow
- High density
  - Granular flow

[Sud et al. 2007]
**Intuition**

- Crowd behaves as an aggregate at medium-high densities
- Reduced individual freedom of movement

**Key Ideas**

- Model crowd as hybrid of
  - Discrete agents
  - Continuum based crowd
- Collision avoidance ↔ Minimum separation
Key Ideas (2)

- Minimum separation
- Density must be below a maximum
- Inequality constraint on density
- Unilateral Incompressibility Constraint (UIC)

[Narain et al; SIGGRAPH Asia 2009]

System Overview

Discrete agents

- Preferred velocities
- Global planner
- Splatting

Continuous flow

- Actual velocities
- Interpolation
- Collision Avoidance (UIC)
- Compressive flow
- Unilaterally incompressible flow
System Overview

- System
  - Agents $i$
    - Mass ($m_i$)
    - Velocity ($v_i$)
  - Crowd continuum
    - Density ($\rho$)
    - Velocity ($v$)
- UIC: $\rho < \rho_{\text{max}}$

Building Crowd Continuum

- Accumulate
  - Agent velocity
  - Agent mass
- Bilinear interpolation weights
Collision Avoidance (UIC)

• How do we model UIC?

• If density is high
  • Apply some force to prevent agents from coming closer

• Else, leave cell as it is

• Solve this constraint as LCP

Collision Avoidance (UIC)

• What should this force be?
  • Isotropic

• Any analogues?
  • Pressure (fluids)

• Pressure force only acts when density is high
Collision Avoidance (UIC)

• Pressure modifies velocities
• What is the pressure force, and thus the new velocity field?
• Modified velocity should make maximum possible progress to goal
  - Maximize $\int \rho v_{\text{original}} \cdot v_{\text{modified}}$

Collision Avoidance (UIC)

• Move with maximum speed possible in the direction of modified velocity

$$v^{n+1} = v_{\text{max}} \frac{v^n - \nabla p}{\|v^n - \nabla p\|}$$

- $\rho < \rho_{\text{max}} \Rightarrow p = 0$
- $p > 0 \Rightarrow \nabla \cdot v = 0$

Steady state
Conservation of
Modified velocity mass

• This is a non-linear formulation
• Approximate iterative solution?
Collision Avoidance (UIC)

- Split into 2 parts
  - Pressure solve
  \[ psolve(v^n) = v^n - \nabla p \]
  - Velocity renormalization
  \[ \text{renorm}(\hat{v}) = v_{\text{max}} \frac{\hat{v}}{\|\hat{v}\|} \]

- Iteratively solve using these 2 primitives
  \[ v^{n+1} = psolve(\text{renorm}(psolve(v^n))) \]

- In practice we use:

- How to do the pressure solve?
  \[ v^n = v_{\text{original}} \]

Collision avoidance (UIC)

- Discretize conservation of mass equation with
  - Density
  - Modified velocity

- Linear Complimentarity problem
- Efficient solvers exist
Collision Avoidance

- Getting collision-free velocities for each agent
  - Interpolate between agent velocity and grid velocity
- There may be some collisions still
  - Push apart intersecting agents
  - Sufficient: only 0.12% agents approach closer than minimum separation

Advantages

- Gross collision avoidance independent of number of agents
- Makes large dense crowds feasible
System Demonstration

Outline: Crowd Simulations

- “Principle of Least Efforts” navigation
- Modeling of Dense Crowd
- Control and Direction Crowds
Approach

- Our method allows the user to ‘direct’ the flow of agents in an ongoing simulation

- Salient features:
  - Interactive scheme to direct virtual crowds
  - Novel formulation for compositing arbitrary user input into navigation fields
  - Importing data from real video

[Patil et al; TVCG 2010]

Framework
Specifying User Input

- Stroke based interface

Importing Guidance Fields From Video

Flow field extraction

Clustering

Individual Guidance Fields
Framework

User Input:
Guidance Fields

User-specified strokes
Flow fields from video
Procedural generation

Composition module

Environment description

Navigation Field

Multi-agent system

Navigation Fields

Key Features
- Goal-directed
- Encodes paths of least effort (minimum cost)
- Singularity-free (except for minima at goal positions)
Navigation Fields

- Store gradient at each grid-cell

Analysis

- Computes discrete approximation of following static Hamilton-Bellman-Jacobi PDE:

\[ \max_{a \in S}{\left\{ (-\nabla T(X) \cdot a) s(X, a) \right\}} = 1 \]

- Necessary and sufficient condition for cost-optimal paths under an *anisotropic* speed function
### Performance

- Complexity: $O(m \cdot n \log (m \cdot n))$

*All times measured on single Intel Xeon 2.66 GHz processor*

<table>
<thead>
<tr>
<th>Scene</th>
<th>#Agents</th>
<th>Grid Dimension (m x n)</th>
<th>NF compute time* (ms)</th>
<th>Local Collision Avoidance</th>
<th>Average sim time* (ms/frame)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 Blocks</td>
<td>100</td>
<td>100 x 100</td>
<td>5.0</td>
<td>RVO</td>
<td>2.0</td>
</tr>
<tr>
<td>Crossing</td>
<td>640</td>
<td>100 x 100</td>
<td>5.0</td>
<td>Helbing</td>
<td>1.1</td>
</tr>
<tr>
<td>Crosswalk</td>
<td>145</td>
<td>225 x 100</td>
<td>13.0</td>
<td>Helbing</td>
<td>0.4</td>
</tr>
<tr>
<td>Subway</td>
<td>435</td>
<td>200 x 200</td>
<td>22.0</td>
<td>RVO</td>
<td>5.5</td>
</tr>
</tbody>
</table>

### Overall System

- For every group of agents sharing a common goal
  - Compute Navigation Field
  - Blend inputs into Guidance field
- Simulation (continuous cycle) for each agent
  - Set preferred velocity from Navigation Field
  - Update position and orientation of agent
  - Local collision avoidance
  - Update visualization
System Demonstration

- **Directing Crowds**

- **Animating Crowds in Blender**

Reconstructing Traffic

- **Virtualized Traffic**
  [van den Berg et al. VR2009; TVCG2010]

- **Continuum Traffic Simulation**
  [Sewall et al. Eurographics 2010; CGF2010]
Summary

- Modeling & simulation of digital humans present many new computational challenges
- New techniques for motion synthesis for virtual humans from individuals to crowds
  - Layer Representation for accelerated collision detection & Dynamics
  - Motion Planning with high degrees of freedom and constraints
  - Multi-Agent Planning and Collision Avoidance
- Map well to new Moore’s Law

Future Research Challenges

- Investigate issues associated with adaptive algorithms that use hierarchical structures (e.g. multigrids, pedestrian level of detail, etc.)
- Hybrid models (physics+AI, data-driven+physics, etc.)
- Incorporation of behaviors and variety
- Integration of locomotion and foot-step planning
- Integrated crowd & traffic simulations in virtual cities
- Parallel algorithms for solving real-world problems (e.g. emergency response, entertainment, shopping, e-commerce, travel)