Layered Intelligence for Agent-based Crowd Simulation

Bikramjit Banerjee, Ahmed Abukmail, Landon Kraemer

SIMULATION, Vol. 85, Issue 10, October 2009, 621-633

Presented by Kit Cischke
November 30, 2009

Crowd Behavior Simulation

- Crowd behavior simulation is a very active field of research now.
- The normal approach is to instantiate a "person" in the form of an autonomous agent.
- The difficulty lies in the commonly held paradigm that "simple characters are more efficient... but complex characters create more realistic crowd behaviors".
- This paper attempts to disabuse us of that notion, by showing it is possible to model complex behaviors with extremely simple agents.

Presentation Outline

- Basics of the methodology
- Layered Intelligence
- Path Planning
- Complexity Analysis
- Dynamically Placed Obstacles
- Congestion as a Dynamic Obstacle
- Scalability

Basics

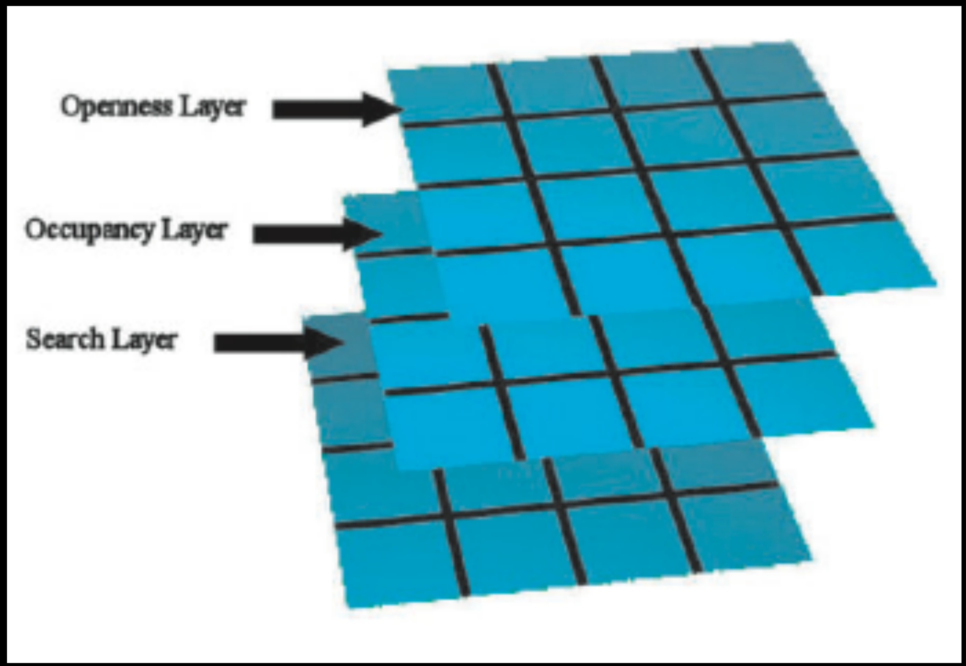
- Distribute the intelligence into the terrain -- called *smart terrain*.
- Studying crowd movement in a 2D world, using a layered AI framework.
- Create a flow field as an Markov Decision Process or semi-Markov DP.
- Combine layers and MDPs to get realistic behavior as agents pursue an assigned goal.
- Extend the layers to get dynamic behavior for new obstacles, threats and congestion.

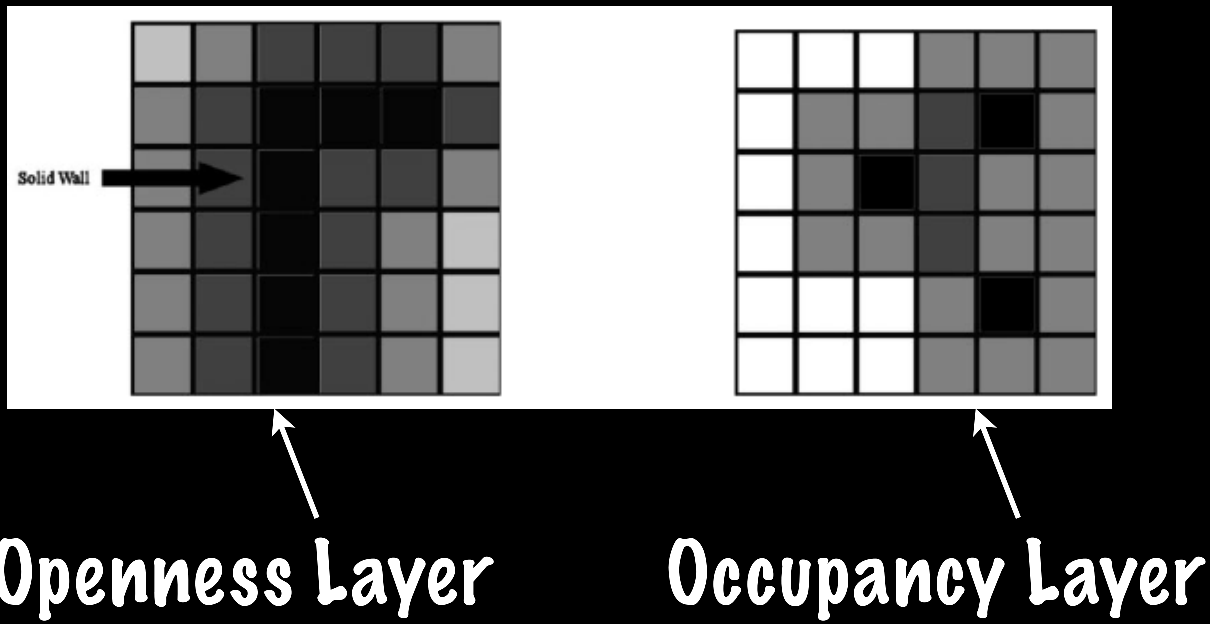
Presentation Outline

- Basics of the methodology
- Layered Intelligence
    - "Put the intelligence in the data, not in the code."
- Path Planning
- Complexity Analysis
- Dynamically Placed Obstacles
- Congestion as a Dynamic Obstacle
- Scalability

Layered Intelligence

- Each cell in the grid is a physical location that holds one person.
- A layer might tell you if a cell is open, if some other agent is in the cell, what your action should be in this cell or something else entirely.
- The values in a layer can be discrete or continuous.
- At each step, the agent simply looks at its neighbors, finds the highest value, and tries to go there.





Complexity

- The complexity of the decision-making process is simply $O(|N|L)$.
  - $|N|$ is the number of neighbor cells, L is the number of layers.
  - That's constant!
- Then, for n agents, figuring out where they all move in the next time interval is $O(n)$.
- This is the principle: *put the intelligence in the data, not in the code.*

Presentation Outline

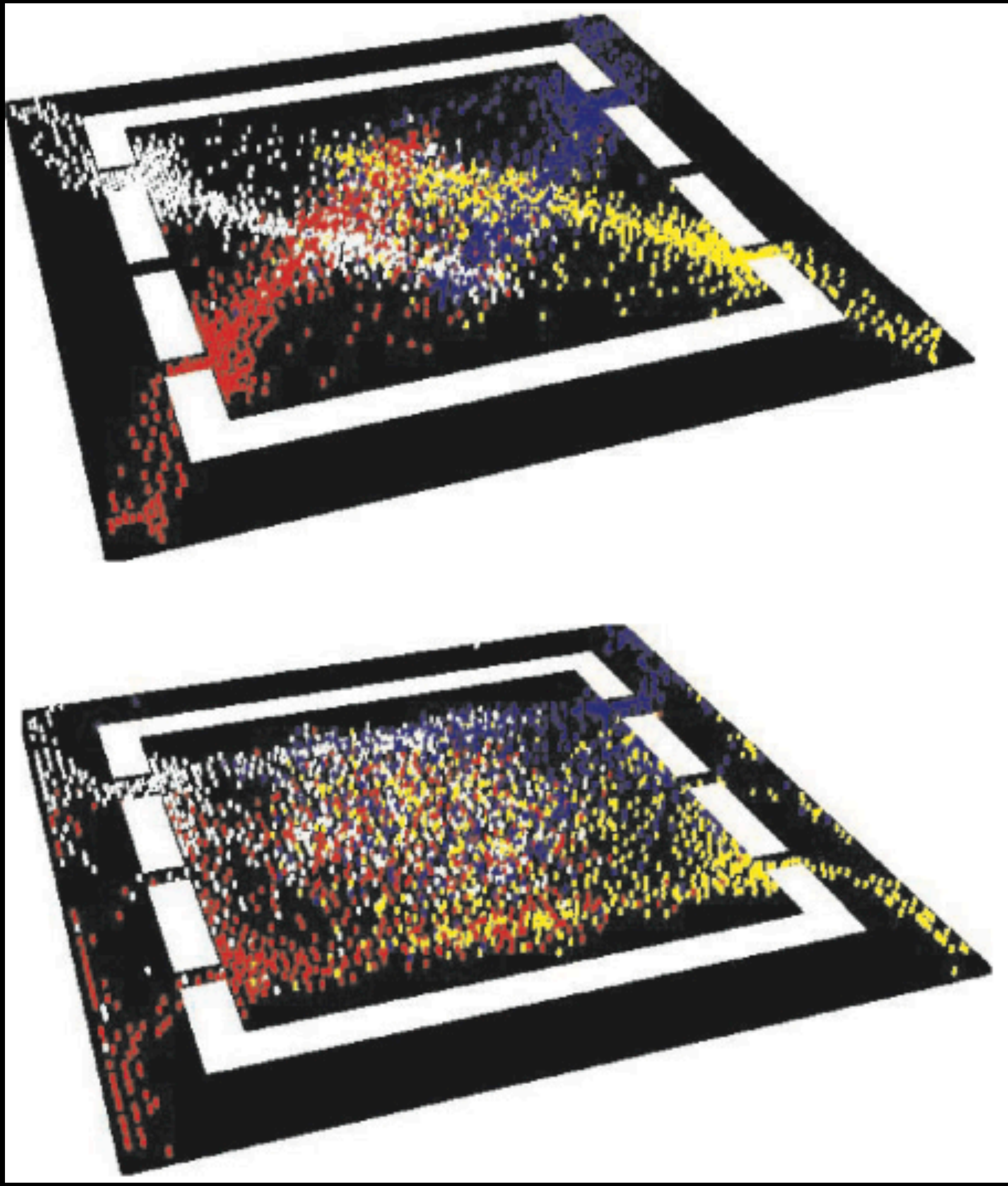
- Basics of the methodology
- Layered Intelligence
- Path Planning
  - Markov and Semi-Markov Decision Problems
- Complexity Analysis
- Dynamically Placed Obstacles
- Congestion as a Dynamic Obstacle
- Scalability

MDP Review

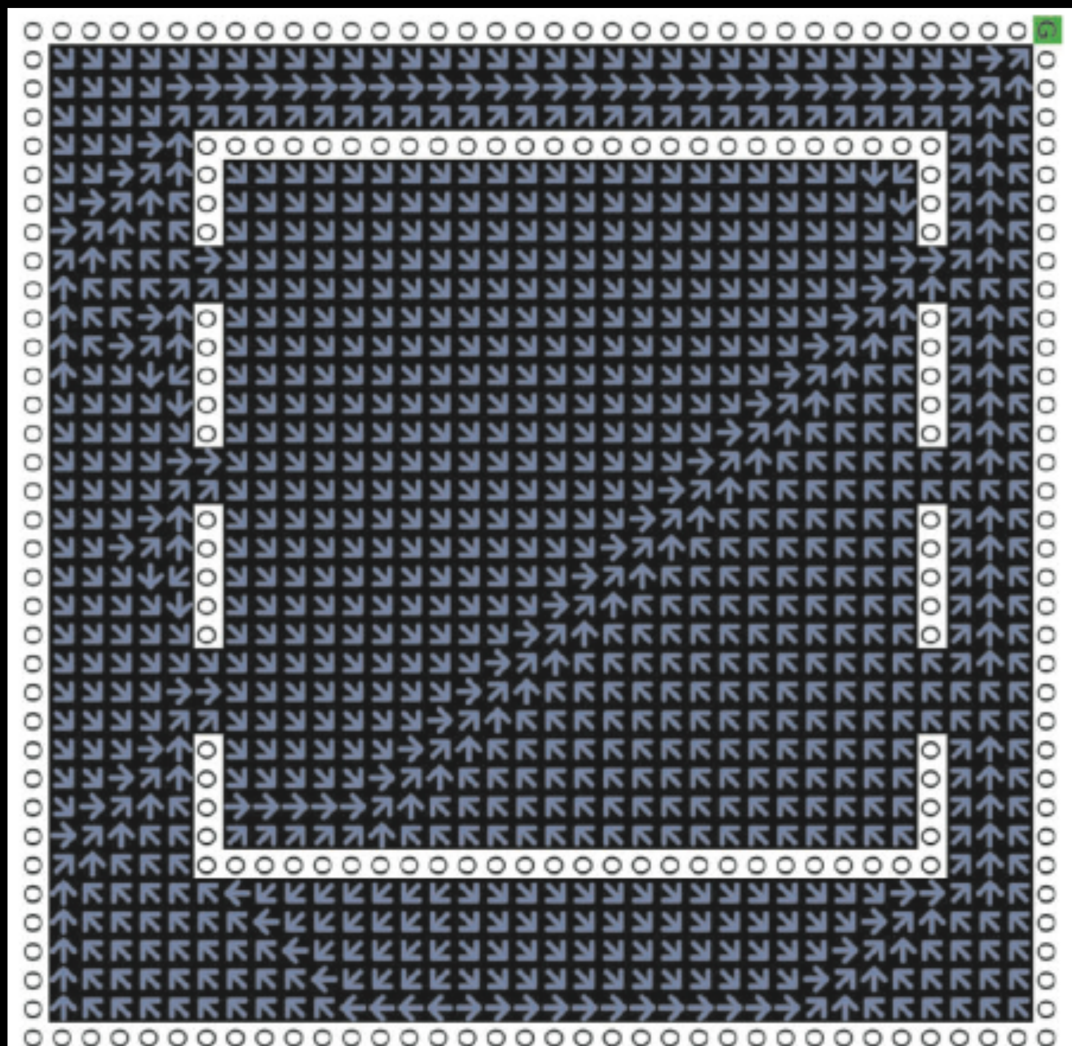
- We covered MDPs in class.
- There are a set of states *S*, rewards for each state *R*, a suite *A* of actions that can be taken at any state and *T* as transition probabilities from any state *s* into another state *s'*.
- We develop a policy π that moves the agent through the state space.
  - The optimal policy is π*.
- Policies are usually developed iteratively, because it otherwise requires solving a non-linear system of equations.
- We can either iterate on policy or value.
  - The authors iterate on value, with a simple reward function.
  - If the cell is a goal, $R(s) = 1$. Otherwise, $R(s) = 0$.

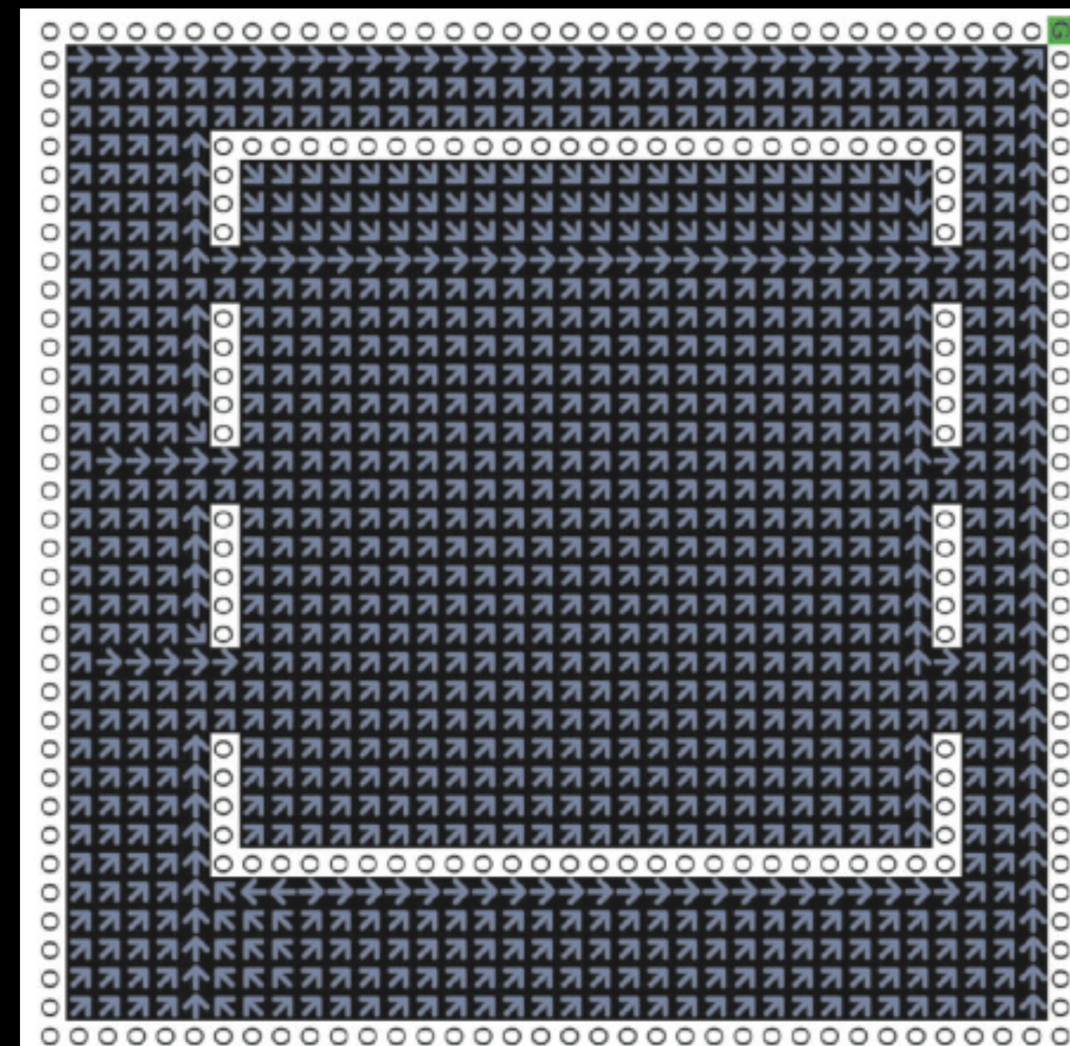
Semi-Markov DP

- We're looking at all 9 possible movements from a given cell. But realistically, the diagonal movements take more time. What if we take into account that you're really moving 1.414 units when you move diagonally?
- We'll put a modifier on the discount in our MDP that is based on the time it takes to complete the action.
  - Now the exponent is $\gamma^{t(a)}$



Effect of the SMDP





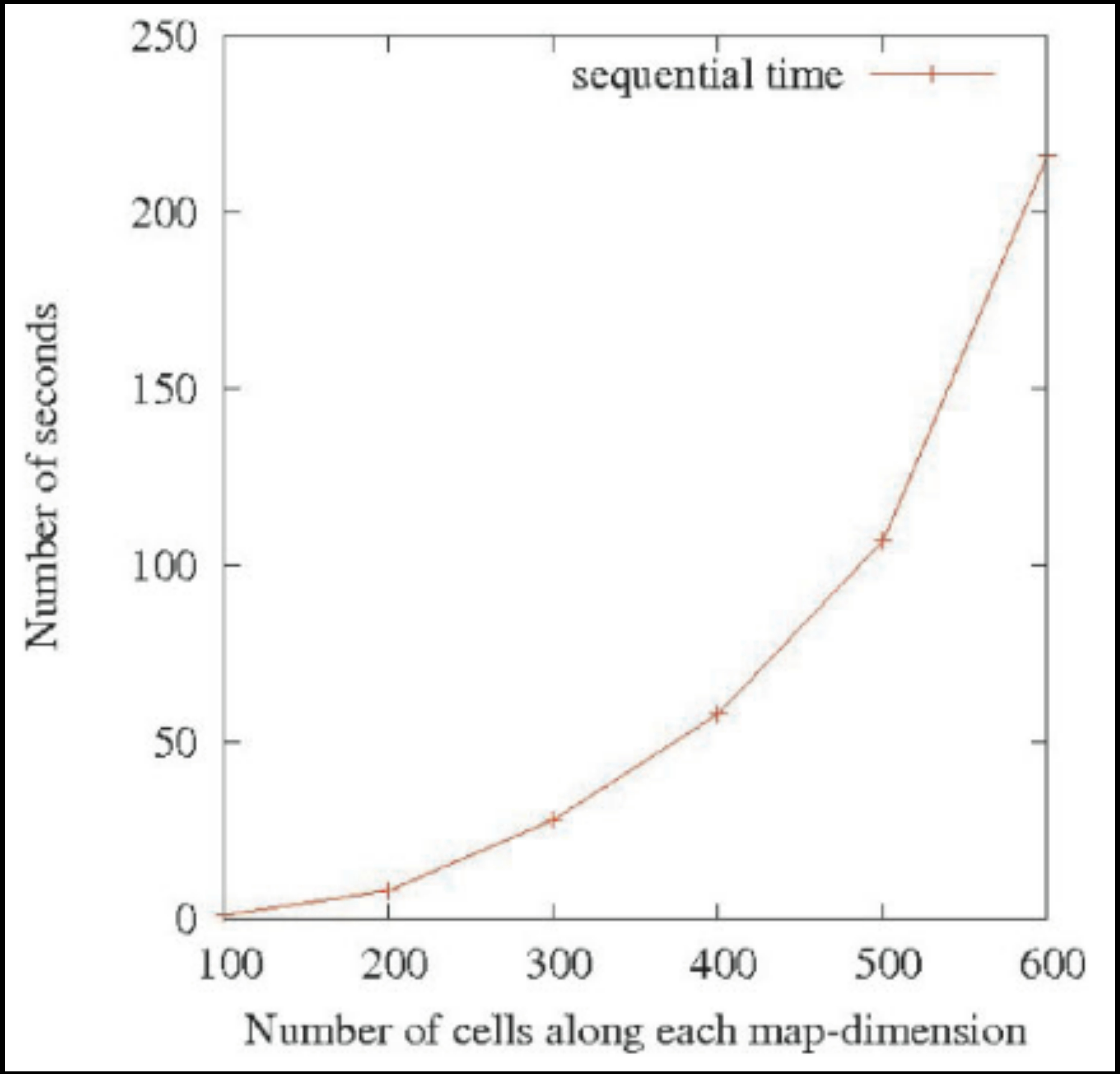
More Effects of the SMDP

Presentation Outline

- Basics of the methodology
- Layered Intelligence
- Path Planning
- Complexity Analysis
  - Is it worth it?
- Dynamically Placed Obstacles
- Congestion as a Dynamic Obstacle
- Scalability

Preprocessing

- The creation of the flow field is very expensive.
- Idea for improvement: generate each goal layer on a separate processor.
- Overall, generating the fields is $O(m^3)$, where m is the size of the world.
  - This makes it competitive with Floyd-Warshall for most maps.
- Full results on the next slide.



A 1200x1200 map is approximately a football stadium.

A map 1/4 that size takes 3.5 minutes to process!

Runtime of the Preprocessor

Presentation Outline

- Basics of the methodology
- Layered Intelligence
- Path Planning
- Complexity Analysis
- Dynamically Placed Obstacles
  - What happens when the roof caves in?
- Congestion as a Dynamic Obstacle
- Scalability

Dynamic Obstacles

- How can this framework handle agents that die, rubble from a collapsed ceiling or other obstacles that change the situation while the simulation is running?
  - With a layer specifically for dynamic obstacles!
- The dynamic obstacle layer creates a "trough" of values around the obstacle so that agents will avoid the obstacle and its immediate surroundings.

Describing the Obstacle

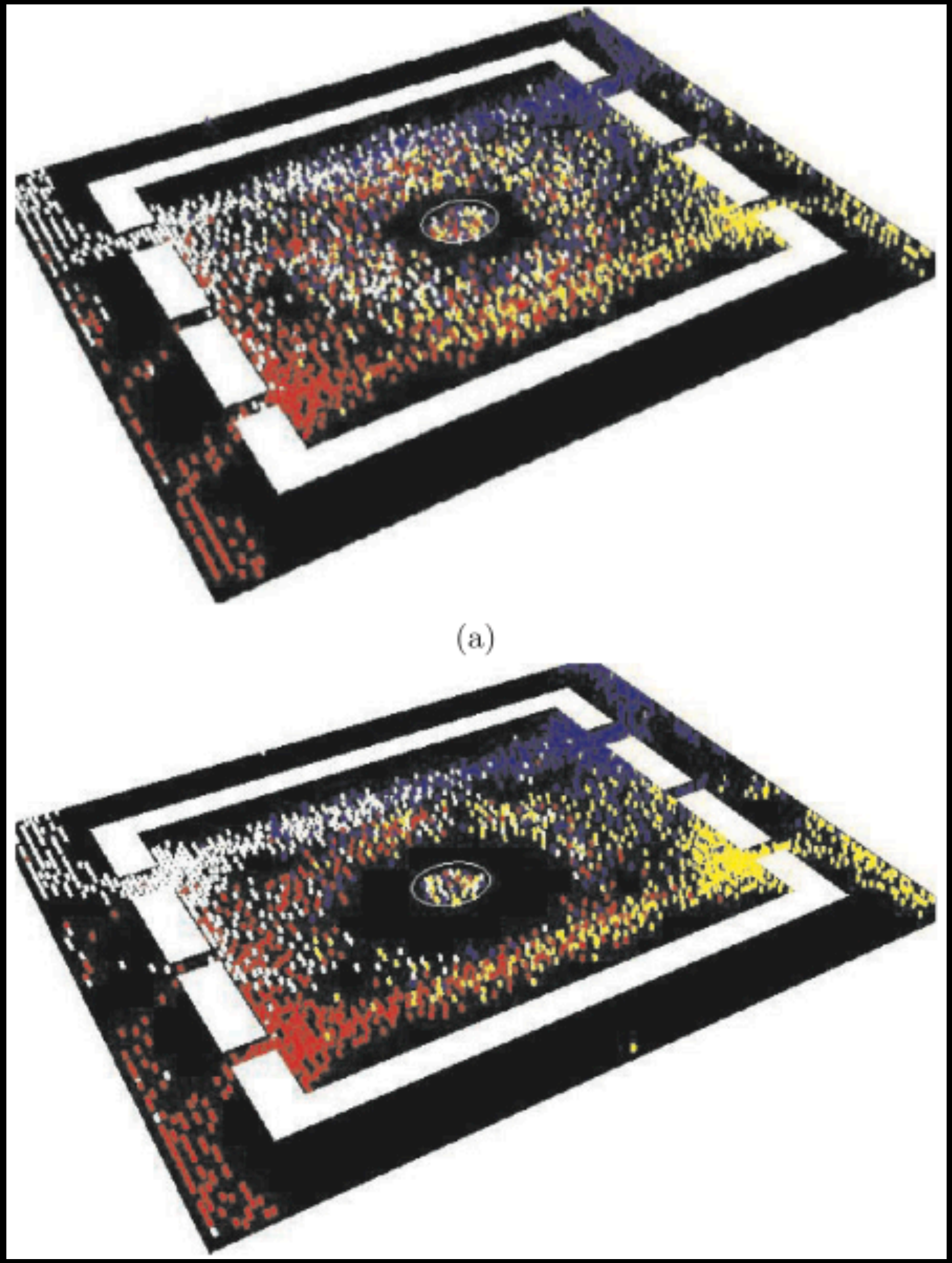
- The obstacle is characterized with 5 parameters:
  - (c_x, c_y), the center coordinates of the obstacle
  - r_i, the inner radius
  - r_o, the outer radius
  - a, the avoidance intensity
- The value of a cell is calculated as:

$$layer(p,q) = \begin{cases} 0 & \text{if} \quad d \le r_\mathrm{i} \\ \left(\dfrac{d - r_\mathrm{i}}{r_\mathrm{o} - r_\mathrm{i}}\right)^a & \text{if} \quad r_\mathrm{i} < d < r_\mathrm{o} \\ 1 & \text{if} \quad d \ge r_\mathrm{o} \end{cases}$$

Dynamic Obstacle Example

The two snapshots at right were taken 2 seconds apart. All the agents in the middle of the "fire" are dead.

It's not just the area that's on fire that is avoided, but also an area of effect.



Limitations

- Agents can be affected by obstacles on the other side of a wall.
  - Solved by "Influence clipping" by performing a Bresenham line rasterization test.
- Immobilized agents -- ostensibly dead -- come back to life after the effects of the dynamic obstacle expire.
  - Solved by really killing the agents and marking them "dead".

Presentation Outline

- Basics of the methodology
- Layered Intelligence
- Path Planning
- Complexity Analysis
- Dynamically Placed Obstacles
- Congestion as a Dynamic Obstacle
  - 500 people heading out the same door.
- Scalability

Congestion

- A major limitation of most crowd simulations is congestion avoidance behavior.
- There are three steps the authors take to address this:
  - Bottleneck identification
  - Congestion identification
  - Congestion avoidance.
- At the heart, congestion is modeled as a dynamic obstacle.

Bottleneck Identification

- Since we have plans for each agent (in the form of flow fields), we can identify bottlenecks where the number of states that transition into some state *s* is greater than the number of states that transition into some other state *s'*.
  - *s'* is the state that s transitions into.
- We might find that multiple bottleneck points are very close to each other, so we might move the effective bottleneck to a central point.
- The effect of congestion is not clipped by the Bresenham method.

Congestion Identification

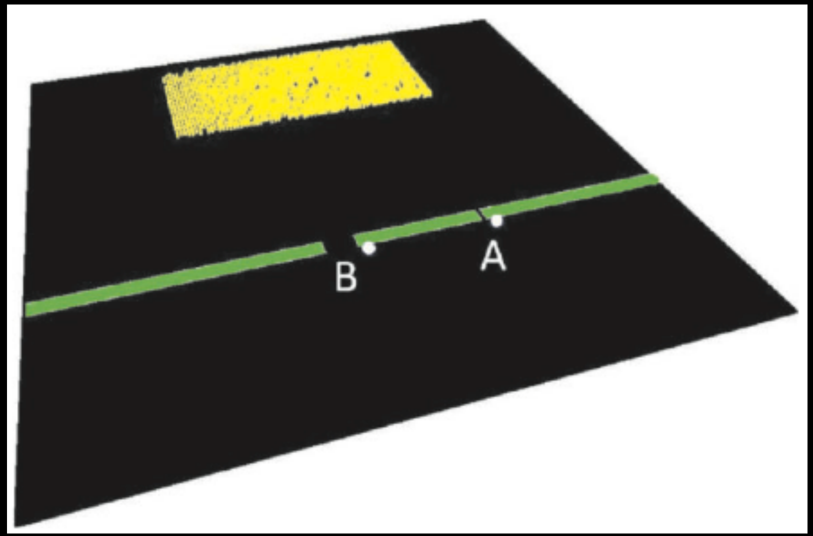
- Once the list of bottlenecks is established, they are checked every few frames of the simulation.
- If the number of agents in the area is greater than some threshold, we declare the area "congested", create a new dynamic obstacle and that initiates congestion avoidance behavior in the agents.
- Complexity of this is $O(|B|k^2)$, where $|B|$ is the number of bottlenecks and k is the size of the influence area of the congestion.
- If congestion falls below another threshold, the congestion is removed from the dynamic obstacle layer.

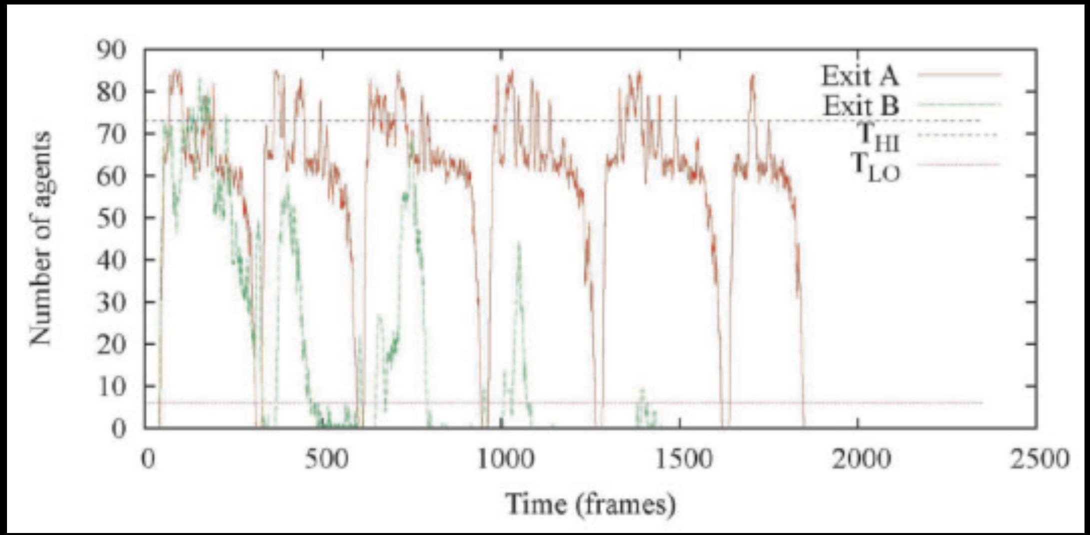
Congestion Avoidance

- A dynamic obstacle is instantiated with r_i equal to the congestion area, r_0 as a multiple of r_i, and *a* is around 2 to achieve "non-linear decay of the congestion's influence".
- Agents involved in this dynamic obstacle are not marked dead or immobile.
- An example of congestion avoidance is on the next slide.

Avoidance Example

- All the agents start out at the top.
- They initially prefer exit A, but it gets congested easily.
- The agents then start utilizing exit B.
- The congestion effects overlap, which accounts for some of the aberrations in the graph.





Presentation Outline

- Basics of the methodology
- Layered Intelligence
- Path Planning
- Complexity Analysis
- Dynamically Placed Obstacles
- Congestion as a Dynamic Obstacle
- Scalability
  - How many agents can we effectively model?

Some Numbers

- On an HP Pavilion 1.6 Ghz laptop with 2 GB RAM, we can effectively visualize 10,000 to 20,000 agents at 30-60 fps.
- Of course, the visualization is little colored dots, but it gets the point across.
  - And it is an order of magnitude greater than competing work.
- The authors point out the numbers would be even higher with more powerful hardware.

Summary

- Demonstrated layered intelligence for a crowd simulation with several navigational behaviors including congestion avoidance and other dynamic obstacles.
- It is also highly scalable.
- Future work includes family groups and leader-follower behaviors.