Crowd simulation
Animating Crowds

- We have been going through methods to simulate individual characters
- What if we want to simulate the movement of crowds?
  - Pedestrians in the streets
  - Flock of birds, school of fishes
  - People in panic
Why need a crowd simulation?

- Shooting scenes with many people
  - Expensive to hire many extra actors
  - We don’t see the details: use simple computer-based models

- Security reasons:
  - Panic in a buildings
Two streams for crowd simulation

- Agent based
  - Modelling each individual
  - Simulate the movements of many individuals to make an effect of crowds

- Using potential field
  - Compute the potential field and determine the movements of characters based on dynamics

- Most of research are Agent based
Flocking Models – Most basic agent model (Reynolds ’87)

- The agents interact based on simple dynamics
- Good to simulate
  - Flock of birds flying,
  - school of fishes swimming
Forces applied to individuals

- Separation
- Alignment
- Cohesion
- Avoidance
Separation

- Try to avoid running into local flock-mates
  - Works just like potential fields
  - Use a *perception volume* to limit visible flock-mates

Separation:
Fly away away from neighbors that are “too close”
Alignment

- Try to fly in same direction as local flock-mates
  - Gets everyone flying in the same direction

Alignment: steer toward average velocity
Cohesion

- Try to move toward the average position of local flock-mates
  - Spaces everyone out evenly,
  - keep boundary members toward the group

Cohesion: steer toward average position
Avoidance

- Try to avoid obstacles
  - Just like potential fields

Avoidance: steer away from obstacles
Combining Commands

- Consider commands as accelerations
- Give a weight to each desire
  - Ex. High for avoidance, low for cohesion
- Option 1: Apply in order of highest weight, until a max acceleration is reached
  - Ensures that high priority things happen
- Option 2: Take weighted sum and truncate acceleration
  - Makes sure some part of everything happens
Flocking Demo
Simulating dynamical features of escape panic: Helbing ’00

- A particle system model to simulate the crowd under panic
- $f_{ij}$, the force to stay away from other entities,
- $f_{iw}$ the force to stay away from walls
- $v_i$: velocity of entity $i$
- $v_i^o e_i^o$: the desired velocity of entity $i$

\[
m_i \frac{d\vec{v}_i}{dt} = m_i \frac{\vec{v}_i^o(t)\vec{e}_i^o(t) - \vec{v}_i(t)}{\tau_i} + \sum_{j \neq i} f_{ij} + \sum f_{iw}
\]

Figure 1: An image of Helbing’s Model Simulation.
Flocking Evaluation

- **Advantages:**
  - Complex behavior from simple rules
  - Many types of behavior can be expressed with different rules and parameters

- **Disadvantages:**
  - Can be difficult to set parameters to achieve desired result
  - All the problems of potential fields regarding strength of forces
Making the Agents Smarter

- Designing the details of the behaviours
  - What to do when you see a predator
  - What to do when you find somebody you know?
  - What to do when the traffic lights are red?
- Need to add more and more rules…
Artificial Fish
- Terzopoulos et al.
SIGGRAPH ‘94

- Adding further factors such as
  - Hunger
  - Libido
  - Fear
for deciding the behavior of the fish

- Adding sensory perception such as
  - vision
Intention Generator

1. first checks the sensory information for collision
2. If any close predator, either
   - Schooling
   - Escape
3. Otherwise if hungry eat
4. If full mate
Intention Generator

- Decision making for the fish

- Collision detection
  - Danger of collision?
    - Yes → $I^t = avoid$
    - No → Predator detection
      - $F > f_0$?
        - Yes → Generate new intention $I^t$ by checking the mental state and the habit string
        - No → $I^t = escape$
      - $F < f_1$ and likes schooling?
        - Yes → $I^t = school$
        - No → pop the memory
          - empty?
            - Yes → $I^s = eat$ or $mate$
            - No → go to the focuser
                - go to the next layer
Demo Movie
What about humans?

- Can do something similar
- But humans are a bit more high level
  - Not only eating or escaping
  - Socializing
    - Grouping,
    - Talking to somebody you know

Just make a more (ad hoc) complex model
Behavior model

- More complex than a fish
More serious problems for humans

- Getting to the destination:
  - use PRM, A* search

- Interactions with the environment
  - Queuing and buying tickets
    - Embed the motions into the environment

- Collision avoidance
  - A bit more complex than fishes
  - Create a more complex controller
Embedding motions into the environment

- At bus stops, people stop and queue
- At an elevator, people wait for it and ride on it when it comes and travel to different floors
- It is possible to associate such motions with the objects
- Once the character comes across such objects, they launch the associated motion
Embedding motions into the environment: Example

- Long streets
  - The characters should walk straightly along the street
- desk and a seat: walk and sit on the seat and face the desk
- TV in the living room: turn on the TV and sit on the sofa to watch TV
- Just let the character conduct the associated action
Embedding motions into the environment:

**Advantages of the approach**

- **Efficient data handling**
  - Each agent holds the data only needed at that moment

- **The control is decentralized**
  - The system is scalable, and large crowds can be simulated
Collision avoidance

- Humans avoid others in streets in a complex way
- Sometimes wait, sometimes, move aside while walking
- Need to either
  - Model them based on a complex collision avoidance engine
  - Find a good mathematical model
Collision Avoidance: Local vs. Global

- Methods based on combining various adhoc rules do not work well (local model)
- Need a good controller that
  - Considers the future
  - Find an optimal path for avoidance
to simulate this effect well (global model)
Global Collision Avoidance

- Fluid Dynamics
  - Not an agent based method
  - Dynamically compute the potential field

- Reinforcement Learning
  - Find the optimal path to avoid the person in the front
Continuum Crowds

- Using fluid dynamics to model the flow of the characters
- Solving the path-planning and collision avoidance at the same time

3. Compute the potential field at every time step
   - Based on the other avatars and obstacles
   - The goal location

4. The character’s movement determined based on the potential field

5. Update the potential field
Continuum Crowds: procedure

- Discretize the space into grids
- Decide the start / goal of the characters
Continuum Crowds: Cost function

\[ \alpha \int_P 1 ds + \beta \int_P \frac{1}{f} ds + \gamma \int_P \frac{g}{f} ds \]

- Path Length
- Time
- Discomfort

\[ \int_P C ds, \quad \text{where} \quad C \equiv \frac{\alpha f + \beta + \gamma g}{f} \]

Characters picks a path that minimizes the above cost function
Continuum Crowds: procedure 2

- The path that minimizes the cost function is going to be picked

\[ \alpha \int_P ds + \beta \int_P \frac{1}{f} ds + \gamma \int_P \frac{g}{f} ds \]

\[ \int_P C ds, \quad \text{where} \quad C \equiv \frac{\alpha f + \beta + \gamma g}{f} \]
Discomfort Field

- Produced by other obstacles / characters

\[ \alpha \int_{P} 1 ds + \beta \int_{P} 1 dt + \gamma \int_{P} g dt \]

Path Length \hspace{1cm} Time \hspace{1cm} Discomfort
Continuum Crowds: procedure 2

\[ \alpha \int_P 1ds + \beta \int_P \frac{1}{f} ds + \gamma \int_P \frac{g}{f} ds \]

- Path Length
- Time
- Discomfort

\[ \int_P Cds, \quad \text{where} \quad C \equiv \frac{\alpha f + \beta + \gamma g}{f} \]

- Potential field \( \phi : \mathbb{R}^2 \rightarrow \mathbb{R} \)
- Cost to the goal at all the grid points
Potential Field

\[ \| \nabla \phi (x) \| = C, \]

\[ \dot{x} = -f(x, \theta) \frac{\nabla \phi (x)}{\| \nabla \phi (x) \|} \]

- All optimal paths follow the direction of \( \nabla \phi (x) \)
- Just compute \( \nabla \phi (x) \) at all the grid points
- If the group of people share the same speed, goal, and discomfort, we can use the same potential field for all these characters
  - Very efficient if there are little number of groups
Procedure 3: Computing the Potential field from gradients

\[ \| \nabla \phi(x) \| = C, \]

\[ \alpha \int_P 1ds + \beta \int_P \frac{1}{f} ds + \gamma \int_P \frac{g}{f} ds \]

\[ \text{Path Length, Time, Discomfort} \]

\[ \int_P C ds, \quad \text{where} \quad C \equiv \frac{\alpha f + \beta + \gamma g}{f} \]

1. Compute from goal where \( C = 0 \) and \( \phi(x) = 0 \)
2. Compute \( \nabla \phi(x) \) at the neighbour of points where \( \phi(x) \) is known
3. Compute \( \phi(x) \) at the neighbours
4. Repeat until \( \phi(x) \) is computed at all the points
Can simulate the dynamics of crowds

- A global approach (optimal towards the goal)
- Can simulate phenomena observed in real humans
Summary

- Agent based approach and Potential field based approach
- Agent-based approach: Flocking
- Need to add many knowledge to make the flocks smart
- Human environment
  - Embedding actions in the environment
  - Require complex collision avoidance
- Collision avoidance
  - Local and global solutions
- Fluid based approach can give global solution to the collision avoidance problem
Reference


- Adrien Treuille, Seth Cooper, Zoran Popovic, Continuum Crowds, SIGGRAPH 2005