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:
  - Need to evaluate the safety of buildings
Three streams for crowd simulation

- Agent based
  - Modelling each individual
  - Simulate the movements of many individuals to make an effect of crowds
- Patch-based methods
  - Create scenes by building blocks
- Using potential / vector fields
  - Compute the potential field and determine the movements of characters based on dynamics
- Most methods 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
  - Use a *perception volume* to limit visible flock-mates

\[
F_s = \sum_{i \in S} \left( \frac{x - x_i}{\|x - x_i\|^2} \right)
\]

- \( S \) : surrounding entities

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

\[ F_A = \sum_{i \in S} (\dot{x}_i - \dot{x}) \]

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

\[ \mathbf{F}_C = \sum_{i \in S} \frac{\mathbf{x}_i}{N} - \mathbf{x} \]
Avoidance

- Try to avoid obstacles
  - The closer the character is, the stronger the force

\[ \mathbf{F}_o = \frac{\mathbf{N}}{D(O, x)} \]

\( \mathbf{N} \): the normal vector of the obstacle at the point that is closest to \( x \)

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

http://www.youtube.com/watch?v=4kPSDW7gQHA
Simulating dynamical features of escape panic: Helbing ‘00

- A particle system model to simulate the crowd under panic
- People rush towards the exit
- $f_{ij}$, the force to stay away from other entities,
- $f_{iw}$, the force to stay away from walls
- $v_i$: velocity of entity $i$
- $v_i^o \dot{e}_i^o$: the desired velocity of entity $i$

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

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

http://www.youtube.com/watch?v=mUKYtOXg5Zw
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
  - Problems 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 check the sensory information for collision
2. If any close predator, either
   - Schooling
   - Escape
Otherwise if hungry eat
- If full mate

http://www.youtube.com/watch?v=RHt_8ZYQVZw
Intention Generator

- Decision making for the fish
What about humans?

- Can do something similar
- But humans are a bit more high level
  - Not only eating or escaping
  - Have destinations
  - Contexts
  - Socializing
    - Grouping,
    - Talking to somebody you know
- Just make a more (ad hoc) complex model
Behavior model: Autonomous Pedestrians

- More complex than a fish

```
Update internal state values

Is fulfilling any desire \( D \)?

Y

Need a ticket?

\( g = \) buy a ticket

N

Y

Pick the most urgent desire \( U \)

\( U = \) attracted

\( g = \) go and watch

\( U = \) thirsty

\( g = \) buy a drink

\( U = \) tired

\( g = \) take a rest

\( U = \) hurried

\( g = \) go to the platform

Put \( g \) on the stack

Decrease the value of \( D \)

Y

Is \( D \) fulfilled?

N

Remove current goal from stack

N

Pass control to behavior module

http://www.youtube.com/watch?v=cqG7ADSvQ5o
```
More serious problems for humans: context

- Interactions with the environment
  - Queuing and buying tickets
    - Embed the motions into the environment
    - patch-based approaches
  - Interactions with others
- Collision avoidance
  - A bit more complex than fishes
  - Create a more complex controller
- Getting to the destination:
  - use path-finding algorithms such as A* search (shortest route to the destination)
Interactions with the Environment

- **Examples**
  - 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
  - Long streets - The characters should walk straight 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

- **IDEA:** associate such motions with the objects
  - Once the character comes across such objects, they launch the associated motion
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
Interactions with others

- Flocks just walking do not look so realistic
- Interactions between the flocks make the animation more lively and realistic
- Associate an input stimuli and an action
  - Find these using videos
- At runtime, compute the probability of performing each action according to the stimuli

http://www.youtube.com/watch?v=FA3y3fu5m2k
Patch-based approaches

- Pre-compute the patches (building blocks) which includes the characters and the environment
- Concatenate them to generate large scale scenes during runtime
- Motion Patches
- Crowd patches
Motion Patches

- Building Blocks for Virtual Environments
  - Embed the motions into the environment
  - The patches are spatially aligned
Crowd Patches

- A patch-based approach to generate scenes of crowded environment
- Crowds avoiding each other in the patches
- The timing and location the characters entering the patches are fixed so that the people can keep on entering / exiting
- The characters can be switched to make more variations
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

(B) Safety in turning  (C) Temporary crowd  (D1) Cross collision  (D2) Head-on coll’n  (E) Front safe area  (F) Verify direction
Collision Avoidance: Local vs. Global

- Combining various ad hoc rules do not work so well (local model)
- Need a good controller that
  - Considers the future
  - Finds an optimal path for avoidance
to simulate this effect well (global model)
Continuum Crowds

- Solving the path-planning and collision avoidance at the same time
  1. Compute the potential field at every time step
     - Based on the other avatars and obstacles
     - The goal location
  2. The character’s movement determined based on the potential field
  3. Update the potential field
Continuum Crowds: procedure

- Discretize the space into grids
- Decide the start / goal of the characters
Continuum Crowds: procedure 2

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

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

- Path Length
- Time
- Discomfort

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

- Produced by other obstacles / characters

\[ \alpha \int_P 1 \, ds + \beta \int_P 1 \, dt + \gamma \int_P g \, dt \]

- Path Length
- Time
- Discomfort
Continuum Crowds: procedure 2

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

- **Path Length**
- **Time**
- **Discomfort**

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

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

It makes sense that for any person, the optimal strategy is to move opposite the gradient of, $\Phi$ as this will decrease cost of the path most rapidly.

$$||\nabla \phi(x)|| = C,$$
Fast Marching Method

A method to solve efficiently the problem of front propagation

Such as computing the distance fields, potential fields, isosurfaces

Similar to Djikstra’s algorithm
Fast Marching Method

Initialize Step

Alive Points: Let $A$ be the set of all grid points $\{i_A, j_A\}$ that represents the initial curve;

Narrow Band: Let NarrowBand be the set of all grid points neighbors of $A$.

Far Away Points: Let FarAway be the set of all others grid points $\{i,j\}$. Set $T_{i,j} = ¥$ for all points in FarAway.
Marching Forwards

1. Begin Loop: Let \((i_{\text{min}}, j_{\text{min}})\) be the point in NarrowBand with the smallest value for \(T\).

2. Add the point \((i_{\text{min}}, j_{\text{min}})\) to \(A\); remove it from NarrowBand.

3. Tag as neighbors any points \((i_{\text{min}}-1, j_{\text{min}}), (i_{\text{min}}+1, j_{\text{min}}), (i_{\text{min}}, j_{\text{min}}-1), (i_{\text{min}}, j_{\text{min}}+1)\) that are either in NarrowBand or FarAway. If the neighbor is in FarAway, remove it from that list and add it to the set NarrowBand.

4. Recompute the values of \(T\) at all neighbors

Return to top of Loop.
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.
Can simulate the dynamics of crowds

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

http://www.youtube.com/watch?v=lqIuVhDFSp8
Reference

- Crowd Patches: Populating Large-Scale Virtual Environments for Real-Time Applications. 
  Barbara Yersin, Jonathan Maïm, Julien Pettré and Daniel Thalmann. 
- Kang Hoon Lee, Myung Geol Choi and Jehee Lee, Motion Patches: Building Blocks for Virtual Environments Annotated with Motion Data, accepted to ACM SIGGRAPH 2006.
- Adrien Treuille, Seth Cooper, Zoran Popovic, Continuum Crowds, SIGGRAPH 2005