Procedural Animation

D.A. Forsyth
Big points

• Two important types of procedural animation
  • slice and dice data, like texture synthesis
  • build (approximate) physical simulation
• Extremely powerful
  • issues
    • how to slice and dice data well
    • what to simulate
Motion graph

- Take measured frames of motion as nodes
  - from motion capture, given us by our friends
- Directed edge from frame to any that could succeed it
  - decide by dynamical similarity criterion
  - see also (Kovar et al 02; Lee et al 02)
- A path is a motion
- Search with constraints
  - root position+orientation
  - length of motion
  - occupy a frame at specified time
  - limb close to a point

Motion Graph:
Nodes = Frames
Edges = Transition
A path = A motion
Search in a motion graph

- Local
  - Kovar et al 02
- With some horizon
  - Lee et al 02; Ikemoto, Arikan+Forsyth 05
- Whole path
  - Arikan+Forsyth 02; Arikan et al 03

Motion Graph:
Nodes = Frames
Edges = Transition
A path = A motion
Local Search methods

• Choose the next edge (Kovar, Gleicher, Pighin 02)
  • ensure that one can’t get stuck locally
  • but can’t guarantee a goal is available on longer scale
Annotation - desirable features

• Composability
  • run and wave;

• Comprehensive but not canonical vocabulary
  • because we don’t know a canonical vocabulary

• Speed and efficiency
  • because we don’t know a canonical vocab.

• Can do this with one classifier per vocabulary item
  • use an SVM applied to joint angles
  • form of on-line learning with human in the loop
  • works startlingly well (in practice 13 bits)
<table>
<thead>
<tr>
<th>Action</th>
<th>1st Frame</th>
<th>2nd Frame</th>
<th>3rd Frame</th>
<th>4th Frame</th>
<th>Motion demand</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walk</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>Run</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jump</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wave</td>
<td>P</td>
<td>P</td>
<td>O</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>Carry</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Synthesis by dynamic programming**

All frames in the database

Arikan+Forsyth+O’Brien 03
Dynamic programming practicalities

- **Scale**
  - Too many frames to synthesize
  - Too many frames in motion graph
- **Obtain good summary path, refine**
  - Form long blocks of motion, cluster
  - DP on stratified sample
    - split blocks on “best” path
    - find similar subblocks
      - DP on this lot
      - etc. to 1-frame blocks
Transplantation

- Motions clearly have a compositional character
  - Why not cut limbs off some motions and attach to others?
    - we get some bad motions
  - build a classifier to tell good from bad
    - avoid foot slide by leaving lower body alone
<table>
<thead>
<tr>
<th>SUBJ</th>
<th>CRS</th>
<th>Section</th>
<th>CRN</th>
<th>Date</th>
<th>Day</th>
<th>Start Time</th>
<th>End Time</th>
<th>Building Room</th>
<th>Exam Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS</td>
<td>419</td>
<td>C3, C4</td>
<td>31366, 39734</td>
<td>5/8/2012</td>
<td>T</td>
<td>8:00 AM</td>
<td>11:00 AM</td>
<td>1SIEBL-1103, 1SIEBL-1105</td>
<td>Extra Space</td>
</tr>
</tbody>
</table>
Physically based animation

- **General idea**
  - take physical models, make assumptions, solve
  - render solution
- **Influential areas**
  - we’ve seen
    - particles,
    - collision+ballistic
  - Others
    - fluids (includes gasses)
Simple example: Accumulating snow

- **Build a fluid simulation**
  - Break volume into cells
  - Compute velocity in each cell (later!)

- **Insert particles**
  - On boundary
  - Proportional to snowfall, n.v
  - Velocity
    - Wind velocity
    - Gravity term

- **Landing cells**
  - Collect a proportion of their snow
  - Return it, if velocity is large
  - Slope snow

Feldman O’Brien 2002
Snow

Feldman O’Brien 2002
Incompressible, inviscid moving fluids

• Examples
  • incompressible fluids
    • water; air at low speeds; honey
  • viscous fluids
    • honey; oil

• Important simplifications
  • compressible, viscous fluids are hard to model
  • compressible flow doesn’t happen at low mach numbers
  • compression is important in explosions, but very hard to model
    • and most undesirable in hollywood style explosions
  • “dry water”
Dry water

- **Variables**
  - $u$: velocity vector
  - $P$: pressure field
  - $f$: force (which could be the result of interactions with particles, etc.)
    - per unit volume
  - $\rho$: density

- **Dynamics**
  - Density $\times$ Acceleration = Force/unit volume

$$\rho \frac{Du}{dt} = f - \nabla P$$
Dry Water

\[ \rho \frac{Du}{dt} = f - \nabla P \]

Substitute

\[ \frac{Du}{dt} = \frac{\partial u}{\partial t} + u \nabla u \]

Rearrange, to get

\[ \frac{\partial u}{\partial t} = -u \nabla u - \frac{\nabla P}{\rho} + \frac{f}{\rho} \]
Dry water

- Euler equations
  - Mass is conserved
  - Change of momentum is due to
    - change of pressure
    - external forces

\[ \nabla u = 0 \]

\[ \frac{\partial u}{\partial t} = -u \nabla u - \frac{\nabla P}{\rho} + \frac{f}{\rho} \]
Solving dry water

- Set up a grid
  - values of $u, P$ at grid vertices

- Get intermediate velocity field
  - by taking a small time step, ignoring pressure effects
  - we will choose a pressure field to correct this to be an incompressible flow

\[
\frac{u^* - u}{\delta t} = - (u \cdot \nabla) u + f
\]

\[
u = u^* - \delta t \nabla P
\]

- Correct the intermediate velocity field

\[
\nabla^2 P = \frac{1}{\delta t} \nabla \cdot u^*
\]
Example: Suspended particle explosion

- There is hot gas, moving under forces generated by:
  - burning
  - momentum
  - changes in pressure
  - etc.

- In the gas, there are particles that:
  - move
  - heat and cool
  - radiate

- Render by rendering the particles:
  - different colors for different temperatures
  - soot particles are black
  - from 1e6 to 4e6 particles

Feldman, O’Brien, Arikan, 03
Modified dry water

- For an explosion, we must have some fluid expansion
  - at points of detonation
  - we do not want to allow the fluid to expand everywhere,
    - or couple this to the fluid’s dynamics
  - pressure waves

- So the pressure update step changes

\[ \nabla u = \phi \]

\[ \nabla^2 P = \frac{1}{\delta t} (\nabla \cdot \mathbf{u}^* - \phi) \]
Heat

- The fluid has heat
  - which is lost by radiation, etc. and gained from particles
- so do particles
  - generate heat by burning
    - which drives the temperature of the particle
    - which drives the transfer of heat into the fluid
Temperature field model

- Fluid temperature
- temperature grid

\[
\frac{DT}{dt} = -c_r \left( \frac{T - T_a}{T_{\text{max}} - T_a} \right) + c_k \nabla^2 T + \frac{1}{\rho c_v} \frac{\partial H}{\partial t}
\]

- Heat lost by radiation
- Heat diffusion (in model, \( c_k \) is set large)
- Heat gained from hot particles moving around
Particles in the fluid

- **Move**
  \[
  \frac{d^2 x}{dt^2} = \frac{f}{m}
  \]

- **Heat**
  \[
  \frac{dY}{dt} = \frac{1}{c_m} \frac{\partial H_p}{\partial t}
  \]
Particle fluid interactions

- **Drag on particle**
  - force in opposite direction applied to fluid
  - low mass - no drag

- **Thermal exchange**
  - heat transfer to a particle from fluid
  - transfer goes both ways
  - $T$ - fluid temperature field

\[
\mathbf{f} = \alpha_d r^2 (\mathbf{u} - \frac{d\mathbf{x}}{dt}) \parallel (\mathbf{u} - \frac{d\mathbf{x}}{dt})
\]

\[
\frac{\partial H_p}{\partial t} = \alpha_h r^2 (T - Y)
\]

\[
\frac{\partial H}{\partial t} = \alpha_h r^2 (Y - T)
\]
Particle behaviour

• Particles burn
  • Simplified combustion
    • combustion is independent of oxygen
    • independent of temperature
    • products do not depend on temperature

• Model
  • Particle ignites when its temperature exceeds a fixed threshold
  • fixed amount of fuel
  • burn at a fixed rate (burn rate)
  • dies when its mass is zero

\[
\frac{dm}{dt} = z
\]

• Products
  • Heat
  • Gas
Products of combustion

- Heat
  \[ \frac{\partial H_p}{\partial t} = b_h z \]
  Add this term to \( \frac{dH_p}{dt} \)

- Gas

- Soot
  - this builds up to a threshold - then a soot particle is released.

\[ \Delta \phi = \frac{1}{V} b_g z \]

\[ \frac{ds}{dt} = b_s z \]