Physically-Based Visual Simulation on Graphics Hardware

Mark J. Harris
Greg Coombe
Thorsten Scheuermann
Anselmo Lastra

The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL
More, Better, Faster

Goal:

- Flexible, real-time visual simulation of diverse dynamic systems
- Further increase visual realism of interactive 3D applications
  - Fluids, gases, fire, etc.
  - Beyond canned texture animations
Visual Simulation?

Goal is visual realism

- Ad hoc simulation methods can provide good visual results with less computation
  - But not necessarily high numerical accuracy
- If the user is convinced, numbers aren’t so important
Coupled Map Lattice

**Mapping:**
- Continuous state $\rightarrow$ lattice nodes

**Coupling:**
- Nodes interact with each other to produce new state according to specified rules
Coupled Map Lattice

CML introduced by Kaneko (1980s)

- Used CML to study spatio-temporal chaos

Others adapted CML to physical simulation:
- Boiling [Yanagita 1992]
- Convection [Yanagita 1993]
- Clouds [Yanagita 1997; Miyazaki 2001]
- Chemical reaction-diffusion [Kapral '93]
- Saltation (sand ripples / dunes) [Nishimori '93]
- And more
CML vs. CA

CML extends cellular automata (CA)

<table>
<thead>
<tr>
<th></th>
<th>CA</th>
<th>CML</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPACE</td>
<td>Discrete</td>
<td>Discrete</td>
</tr>
<tr>
<td>TIME</td>
<td>Discrete</td>
<td>Discrete</td>
</tr>
<tr>
<td>STATE</td>
<td>Discrete</td>
<td>Continuous</td>
</tr>
</tbody>
</table>
CML vs. CA

Continuous state is more useful

- Discrete: physical quantities difficult
  - Must filter over many nodes to get “real” values
- Continuous: physical quantities easy
  - Real physical values at each node
  - Temperature, velocity, concentration, etc.
CML Operations
Graphics Hardware

Why use it?

- Speed: up to 25x speedup in our sims
- GPU perf. grows faster than CPU perf.
- Cheap: GeForce 4 Ti 4200 < $140
- Load balancing in complex applications

Why not use it?

- Low precision computation (not anymore!)
- Difficult to program (not for long!)
Hardware Implementation

CML Operation
- Select Neighbors
- Sample Neighbors
- Combine Samples (Arithmetic)
- Store New State

Iteration

Graphics Pipeline
- Vertex Program (Set Texture Coordinates)
- Texture Shaders
- Texture Blending (Register Combiners)
- Render To Texture

Feedback
CML Operations

Implement operations as building blocks for use in multiple simulations

- Diffusion
- Buoyancy (2 types)
- Latent Heat
- Advection
- Viscosity / Pressure
- Gray-Scott Chemical Reaction
- Boundary Conditions
- User interaction (drawing)
- Transfer function (color gradient)
Results

Implemented multiple simulations

- Boiling (2D and 3D)
- Rayleigh-Bénard Convection (2D)
- Reaction-diffusion (2D and 3D)
Boiling

[Yanagita 1992]

State = Temperature

Three operations:
- Diffusion, buoyancy, latent heat
- 7 passes in 2D, 9 per 3D slice
Rayleigh-Bénard Convection

- Yanagita & Kaneko 1993
- State = temp. (scalar) + velocity (vector)
- Three operations (10 passes):
  - Diffusion, advection, and viscosity / pressure
Reaction-Diffusion

- Gray-Scott reaction-diffusion model [Pearson 1993]
- State = two scalar chemical concentrations
- Simple: just diffusion and reaction ops
- 2 passes in 2D, 3 per 3D slice
Hardware Limitations

Precision, precision, precision!

- 8 or 9 bits is far from enough
- Diffusion is very susceptible to precision problems
  - All of our simulations use it!
- High dynamic range simulations are very susceptible
  - Convection, reaction-diffusion
  - Not boiling – relatively small range of values
Hardware Limitations

3D texturing

- It’s fast ... unless you are doing dynamic texturing
  - Copy to slice of 3D texture is slow
  - We use a stack of 2D slices to represent 3D volume
- Not all 2D texture shaders have 3D analogs
  - Fixed with general purpose fragment programs (NV30, ATI R300)
Future Work

- Explore simulation techniques / issues on graphics hardware
  - Other PDE solution techniques
  - More complex simulations
  - Use of HLSL, next-gen hardware
  - High dynamic range simulations

- Applications:
  - Interactive environments, games
  - Scientific Computation
  - Dynamic painting / modeling applications
  - Dynamic procedural texture synthesis
  - Dynamic procedural model synthesis
  - ...

Conclusion

GPUs are a capable, efficient, and flexible platform for physically-based visual simulation
Acknowledgements

- NVIDIA Developer Relations
- HWW Paper Reviewers

Sponsors:
- NVIDIA Corporation
- US National Institutes of Health
- US Office of Naval Research
- US Department of Energy ASCI program
- US National Science Foundation
For More Information

- [http://www.cs.unc.edu/~harrism/cml](http://www.cs.unc.edu/~harrism/cml)
- Email harrism@cs.unc.edu