Cloth Simulation on the GPU

Cyril Zeller

NVIDIA Corporation

SIGGRAPH2005

Overview

SIGGRAPH2005

- A method to simulate cloth on any GPU supporting Shader Model 3 (Quadro FX 4500, 4400, 3400, 1400, 540, GeForce 6 and above)
  - Takes advantage of the massive parallel computation horsepower of GPUs
  - Geared toward performance and visual realism, not physical accuracy
  - Suitable to 3D games and virtual reality systems

Outline

SIGGRAPH2005

- Demo
- Algorithm outline
- GPU implementation

Demo

- Available at
  http://download.developer.nvidia.com/developer/SDK/Individual_Samples/samples.html#Cloth

Cloth as a Set of Particles

SIGGRAPH2005

- Each particle is subject to:
  - A force (gravity, wind, drag, etc.)
  - Various constraints:
    - To maintain overall shape (springs)
    - To prevent interpenetration with the environment
- Constraints are resolved by relaxation
- CPU version successfully used in games:

Jakobsen, T. “Advanced character physics”, *GDC 01*

Force

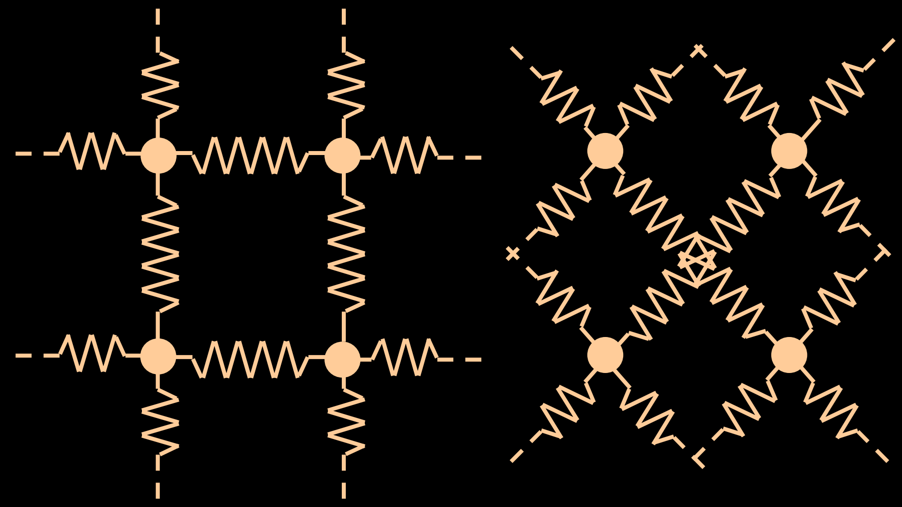
- Verlet integration:

$$P(t + \Delta t) = P(t) + k\,(P(t) - P(t - \Delta t)) + \Delta t^2\, F(t)$$

  - Δt: simulation time step
  - $P(t)$: particle position
  - $F(t)$: force
  - k: damping coefficient
- No force applied to fixed or user-moved particles

Spring Constraints

- Particles are linked by springs:



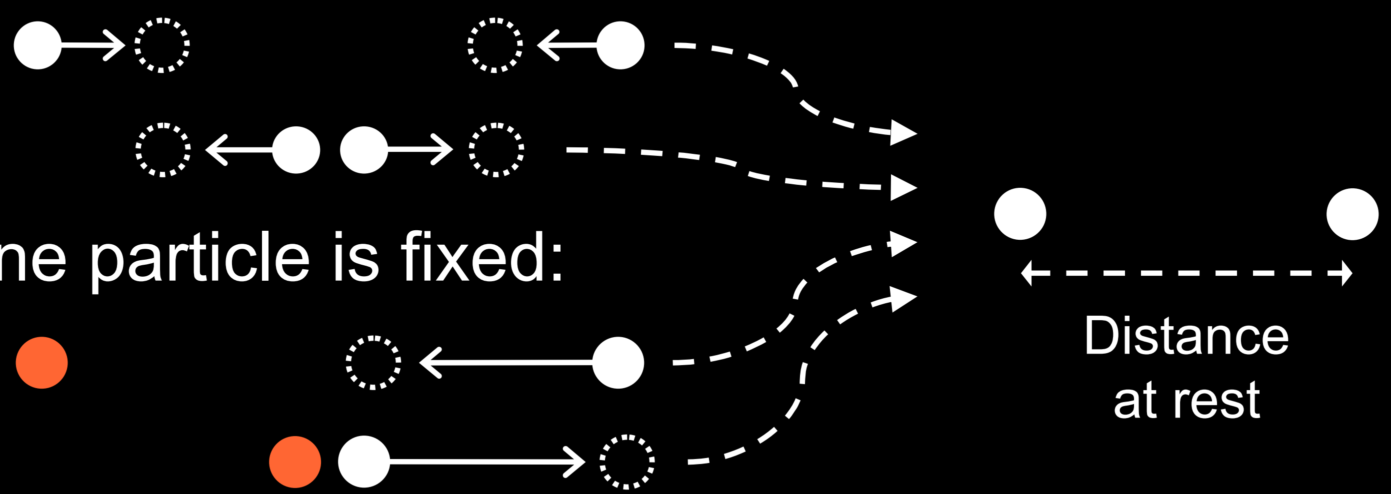
Structural springs Shear springs

- A spring is simulated as a distance constraint between two particles

Distance Constraint

SIGGRAPH2005

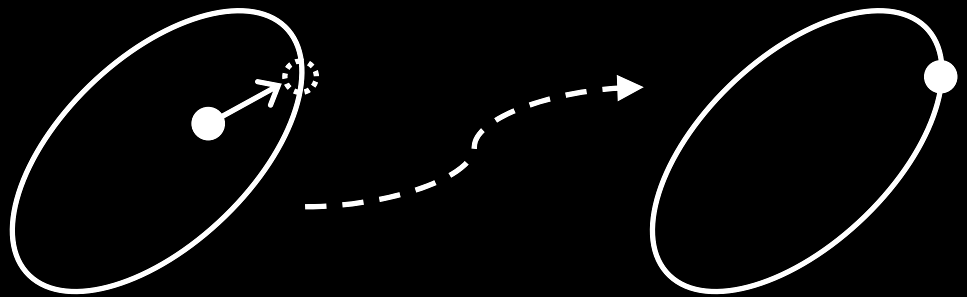
- A distance constraint between two particles is enforced by moving them away or towards each other:
  - If both particles are free:
  - If one particle is fixed:



Collision Constraints

SIGGRAPH2005

- The environment is defined as a set of collision objects (planes, spheres, boxes, ellipsoids)
- A collision constraint between a particle and a collision object is enforced by moving the particle outside the object:



Algorithm Outline

SIGGRAPH2005

- For every simulation time step:
  - For every particle that isn't fixed or user-moved:
    - Apply force
  - For every relaxation step:
    - For every spring constraint:
      - Enforce distance constraint
    - For every particle:
      - For every collision object:
        - If the particle is inside, move it outside

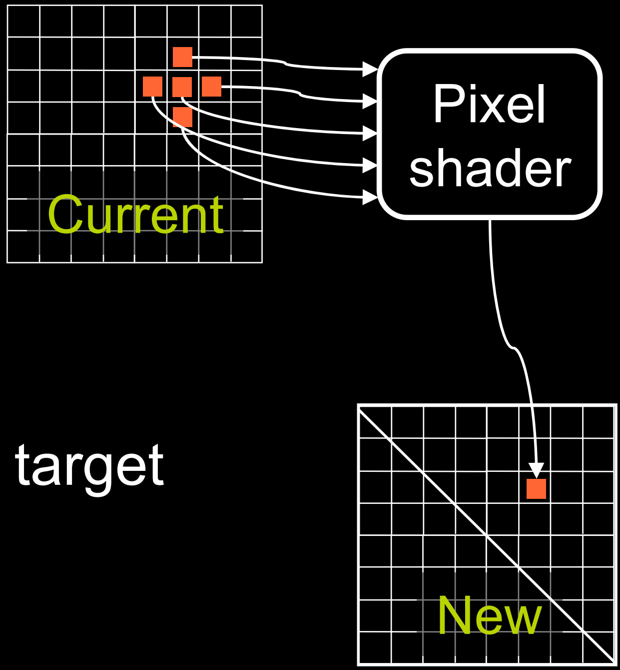
GPU Implementation Overview

SIGGRAPH2005

- The particle positions and normals are stored into floating-point textures
- The CPU never reads back these textures!
- At every frame:
  - GPU simulation: Update the position and normal textures
  - GPU rendering: Render using vertex texture fetch (available on Shader Model 3.0 and above)

GPU Simulation: Updating the Position Texture

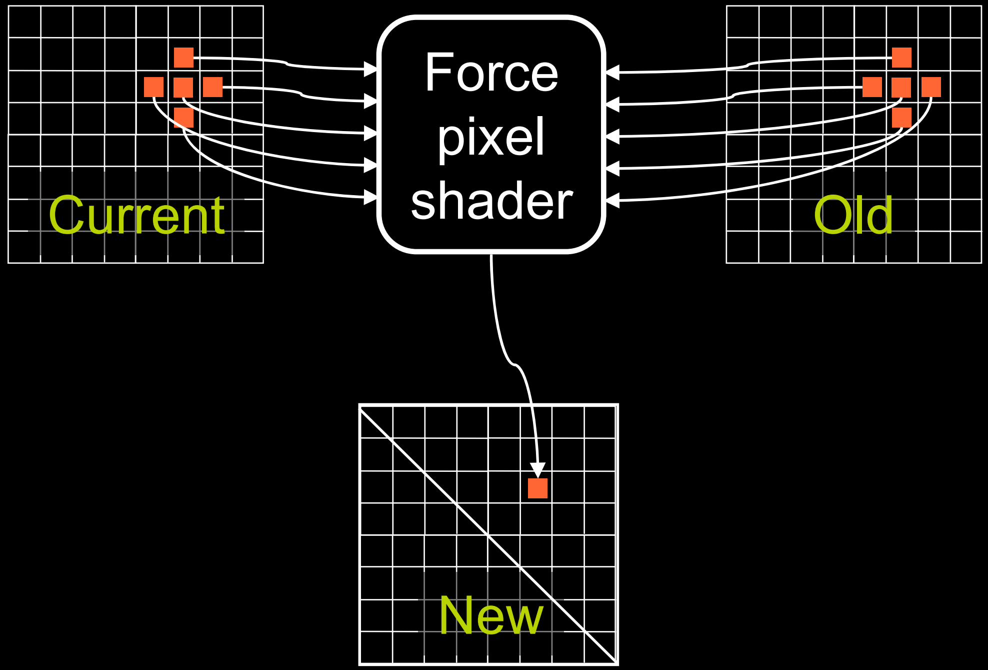
- Two textures are required: Current and New
- Positions get updated through a series of draw calls
- Each draw call is of the form:
  - Set the appropriate pixel shader
  - Set New as the render target
  - Draw a quad covering the entire render target
  - Swap Current and New



GPU Simulation: Force

SIGGRAPH2005

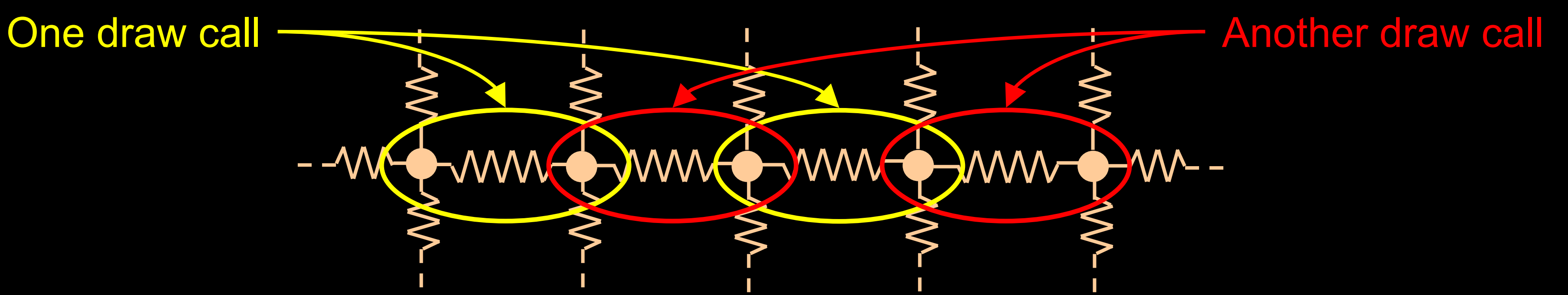
- One draw call
- Verlet integration requires three textures:



GPU Simulation: Spring Constraints

SIGGRAPH2005

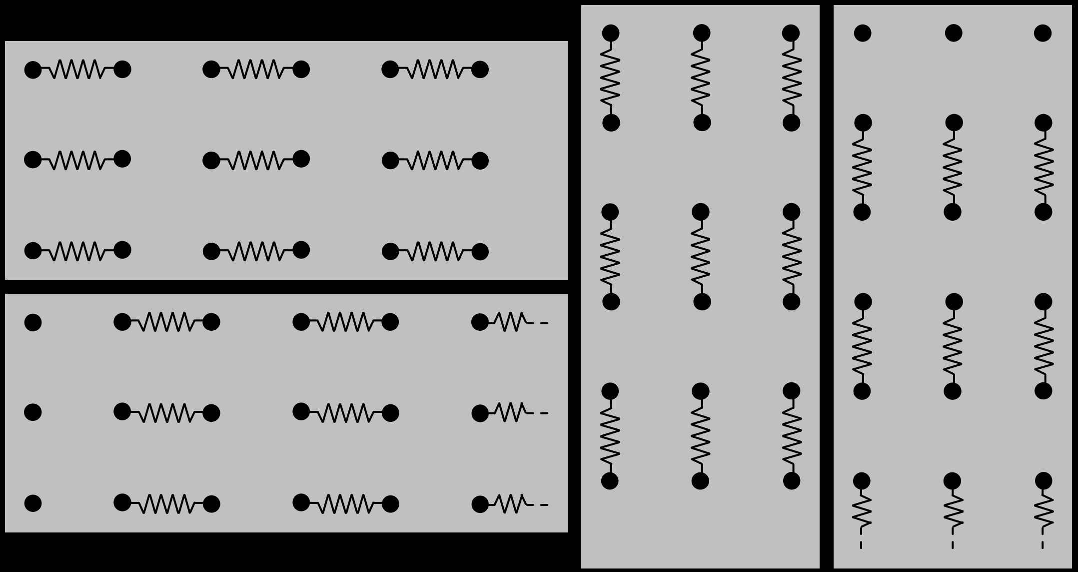
- Interdependent constraints must be enforced sequentially for the relaxation to converge
- So constraints are divided into 8 groups of independent constraints
- One draw call per group



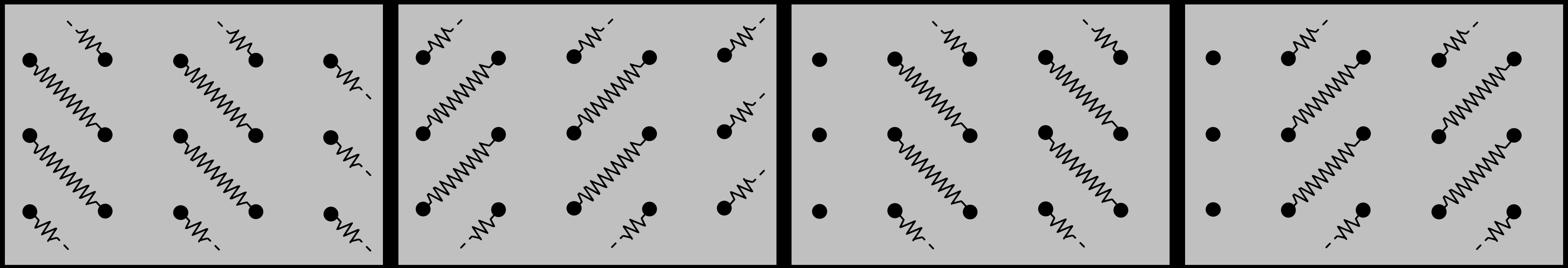
GPU Simulation: Spring Constraints

SIGGRAPH2005

- 4 draw calls for the structural springs:



- 4 draw calls for the shear springs:



GPU Simulation: Collision Constraints

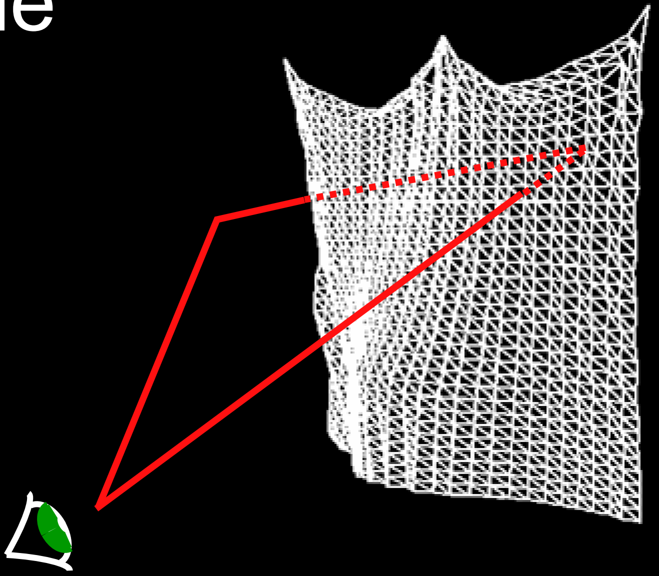
SIGGRAPH2005

- One draw call
- The parameters of each collision object (center, dimension) are stored into 1D-textures:
  - One texture per geometric type
  - Textures are necessary for looping through the collision objects since Shader Model 3.0 does not support indexing of constant registers

GPU Simulation: Cloth Cutting

SIGGRAPH2005

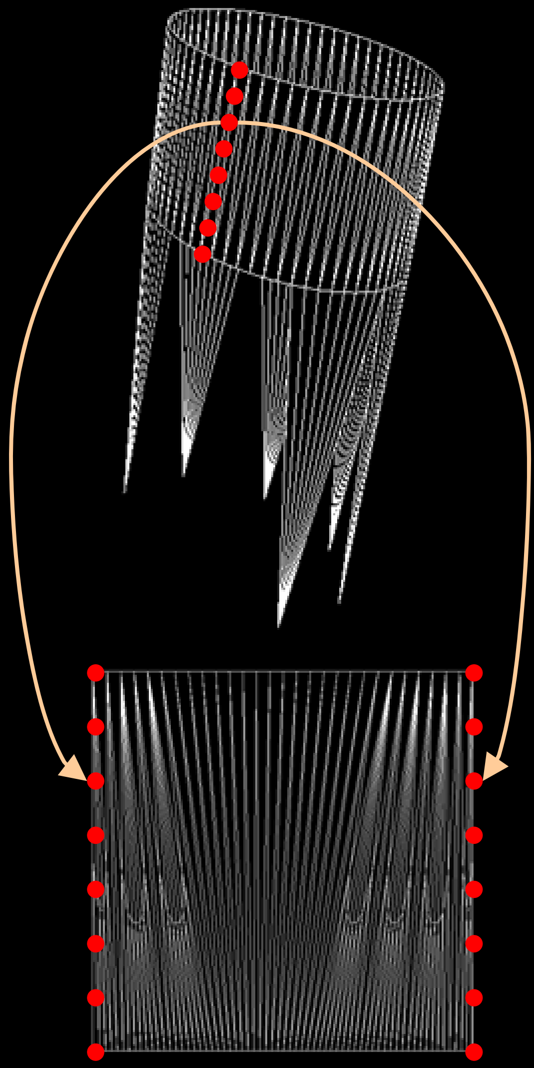
- In cut mode each mouse motion defines a cutter triangle
- A pixel shader intersects the cutter with each cloth triangle
- The result is read back to the CPU and cut triangles and springs are removed



Cloth with a Generic Shape

SIGGRAPH2005

- Non-rectangular mesh or mesh with holes:
  - Create a geometry image from the mesh [Gu et al. "Geometry Images" Siggraph 02]
  - Handle split vertices by:
    - Enforcing spring constraints attached to each duplicate
    - Averaging the resulting positions before enforcing the collision constraints



Performance

SIGGRAPH2005

- 400 frames per second
- On a GeForce 6800 Ultra
- With:
  - 100 x 100 particles
  - One relaxation step
  - Structural and shear springs

Future Work

SIGGRAPH2005

- Better collision detection
- Self-collision detection
- Simulation level of details