Realtime Hair simulation & rendering
Problem

• Human head has ~150k hair strands
• Physics simulation
• Rendering
Problem

• Scaling problem
• Reduce number of objects
• Reduce computation time
• Exploit problem characteristics
  • Approximations
  • Simulate only a subset of hair strands
Simulation

• Simulate forces acting on hair
• Simulate hair-object interactions (collision)
• Simulate hair-hair interactions (collision)
Simulation Modeling

• Each hair strand is a set of linked particles
• Forces acts on particles respecting some constraints
  • A strand is inextensible (can’t change its total length)
  • Hair can not penetrate objects
  • Hair must maintain its volume
Simulation Modeling

- Particles have equal masses
- Root particle is static (infinity mass)
- Forces acting on particles
  - Gravity
  - Wind
- Shape matching
Simulation Modeling

- Particles have equal mass
- Root particle is static (infinity mass)
- Forces acting on particles
  - Gravity
  - Wind
- Shape matching
Simulation Modeling

- Particles have equal mass
- Root particle is static (infinity mass)
- Forces acting on particles
  - Gravity
  - Wind
- Shape matching
Simulation Modeling

- Particles have equal mass
- Root particle is static (infinity mass)
- Forces acting on particles
  - Gravity
  - Wind
- Shape matching
Position Based Simulation

- Update particle position
- Solve constraints
- Derive new velocities
Position Based Simulation

- Update particle position
- Solve constraints
- Derive new velocities
Position Based Simulation

• Update particle position
• Solve constraints
• Derive new velocities
Position Based Simulation

• Update particle position
• Solve constraints
• Derive new velocities
Position Based Simulation

- Update particle position
- Solve constraints
- Derive new velocities
Position Based Simulation

• Strand is a series of mass-spring systems
• Apply newton’s law for each particle
  • $v'_{k+1} = v_k + a \Delta t$
  • $x'_{k+1} = x_k + v'_{k+1} \Delta t$
• Move particles to a valid position
  • $x_{k+1} = \text{SolveConstraints}(x'_{k+1})$
• Derive new velocity
  • $v_{k+1} = (x_{k+1} - x_k) / \Delta t$
Position Based Simulation

• Constraint solver
• Ideal: find a valid configuration which solves all constraints at once
• Reality: Solve each constraint sequentially iteratively
Constraints

- Distance constraints
  - Correct particle distances
  - Must maintain linear and angular momentum

- Constraint Projection

- Follow The Leader (FTL)
Constraints

- Distance constraints
  - Correct particle distances
  - Must maintain linear and angular momentum

- Constraint Projection
- Follow The Leader (FTL)
Constraints

• Distance constraints
  • Correct particle distances
  • Must maintain linear and angular momentum

• Constraint Projection

• Follow The Leader (FTL)
Constraints

- Collision constraints
  - Avoid penetration
  - Move particles to valid position
Constraints

- Collision constraints
  - Avoid penetration
  - Move particles to valid position
Constraints

• Collision constraints
  • Avoid penetration
  • Move particles to valid position
Hair-Hair interaction

- Simulate global behavior
  - Energy diffusion
  - Hair volume
- Brute force $O(n^2)$
  - Collision detection of each strand against all strands
- Volumetric approach
  - Treat particles as fluid
  - Diffuse energy to neighboring particles
  - Apply repulsion to maintain volume
Hair-Hair interaction

• Volumetric approach

• Construct a 3D grid
  • Calculate particle density on grid nodes
  • Average velocities of particles on grid nodes

\[ \nu = (1 - \alpha)\nu_{particle} + \alpha\nu_{grid} + \nu_{repulsion} \]
Hair-Hair interaction

• Volumetric approach
• Construct a 3D grid
  • Calculate particle density on grid nodes
  • Average velocities of particles on grid nodes

\[ v = (1 - \alpha) v_{particle} + \alpha v_{grid} + v_{repulsion} \]
**Hair-Hair interaction**

- Volumetric approach
- Construct a 3D grid
  - Calculate particle density on grid nodes
  - Average velocities of particles on grid nodes

\[ v = (1 - \alpha)v_{\text{particle}} + \alpha v_{\text{grid}} + v_{\text{repulsion}} \]
Hair-Hair interaction

- Volumetric approach
- Construct a 3D grid
  - Calculate particle density on grid nodes
  - Average velocities of particles on grid nodes

\[ \mathbf{v} = (1 - \alpha) \mathbf{v}_{\text{particle}} + \alpha \mathbf{v}_{\text{grid}} + \mathbf{v}_{\text{repulsion}} \]
Hair-Hair interaction

Velocity smoothing: off vs on
Hair-Hair interaction

Grid filtering ($\alpha = 0.9$): off vs on
Rendering

• Geometry
• Appearance
Rendering

• Geometry
  • Coarse geometry
  • Strands rendered as texture
Rendering

- Geometry
  - Coarse geometry
  - Strands rendered as texture
Rendering

• Geometry (refined)
  • Spline interpolate strand particles
    • Generate polygon strip
    • Generate follow strands
Rendering

- Geometry (refined)
  - Spline interpolate strand particles
    - Generate polygon strip
    - Generate follow strands
Rendering

• Geometry (refined)
  • Spline interpolate strand particles
    • Generate polygon strip
    • Generate follow strands
Rendering

- Geometry (refined)
  - Spline interpolate strand particles
  - Generate polygon strip
  - Generate follow strands
Rendering

• Geometry (refined)
  • Spline interpolate strand particles
  • Generate polygon strip
  • Generate follow strands
Rendering

• Geometry (refined)
  • Spline interpolate strand particles
  • Generate polygon strip
• Generate follow strands
  • Barycentric interpolation
Rendering

• Geometry (refined)
  • Spline interpolate strand particles
  • Generate polygon strip
  • Generate follow strands
Rendering

• Appearance
  • Shading
  • Self-Shadowing
  • Transparency
Rendering

• Shading
  • Kajiya-Kay model
  • Marschner model
  • Photograph
Rendering

• Self-Shadowing
  • Opacity maps
  • Voxel based
Rendering

• Self-Shadowing
  • Opacity maps
    • Voxel based
Rendering

- Self-Shadowing
  - Opacity maps
  - Voxel based
Rendering

• Transparency
  • Blending operation does not commute
  • Must render back to front
Rendering

• Order independent Transparency
  • Store multiples fragments (color, alpha and depth) per pixel
  • Sort and blend
Rendering

- Stochastic Transparency
  - Use multiple samples per pixel
  - Randomly paint samples according to fragment alpha
  - Guarantee correct color on average
Rendering

• Commutative blending op
  • Approximate blending
  • Low error only on specific cases
Rendering summary

- Complex light interaction
  - Anisotropic specular
  - Refraction
  - Reflection
  - Self-Shadowing
  - Transparency
References

• MESIT, Jarawan. Modeling And Simulation Of Soft Bodies. 2010.
• LE MUZIC, Mathieu. Real-Time Hair Simulation and Rendering with OpenCL and OpenGL. 2012.