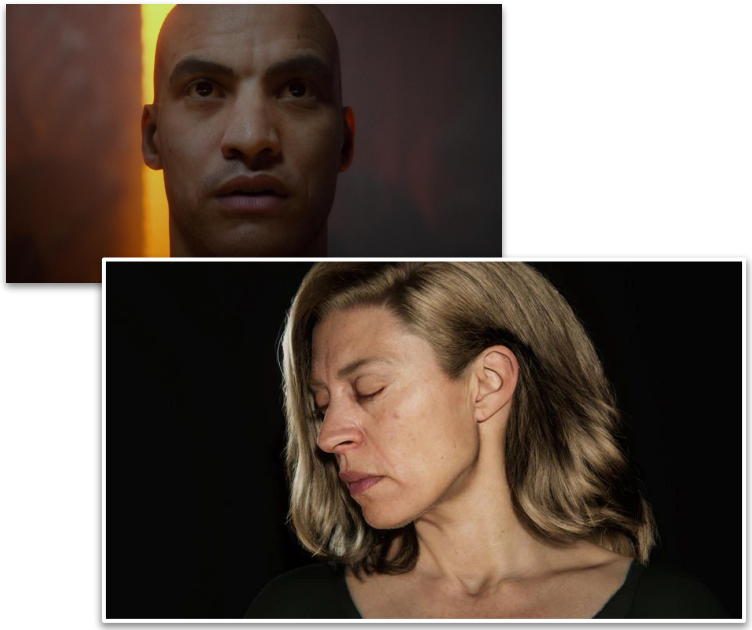
Strand-based Hair System

Lasse Jon Fuglsang Pedersen
@codeverses

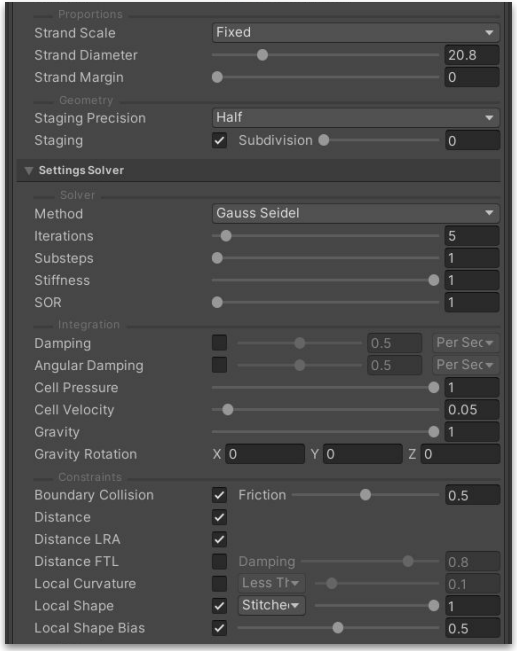
Strand-based Hair System

- A work in progress since ‘The Heretic’
  - We used just stubble back then
- We needed a solution for ‘Enemies’
  - Actress had long flowing hair
  - Wanted workflow for rapid iteration
    - No baking
    - Real-time simulation
- Encouraged by other teams’ results
    - Frostbite video

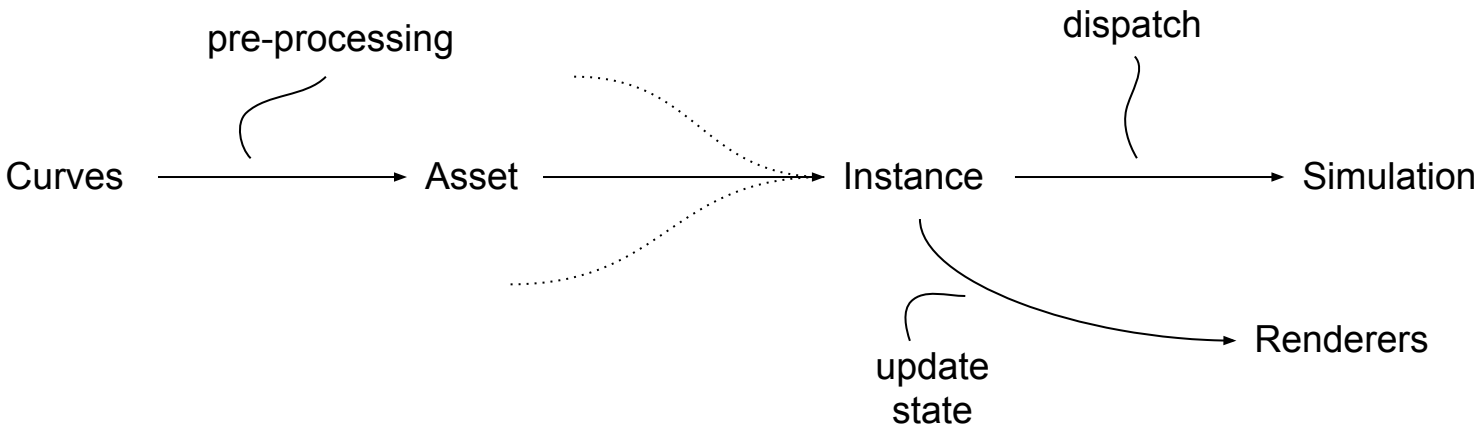


Strand-based Hair System

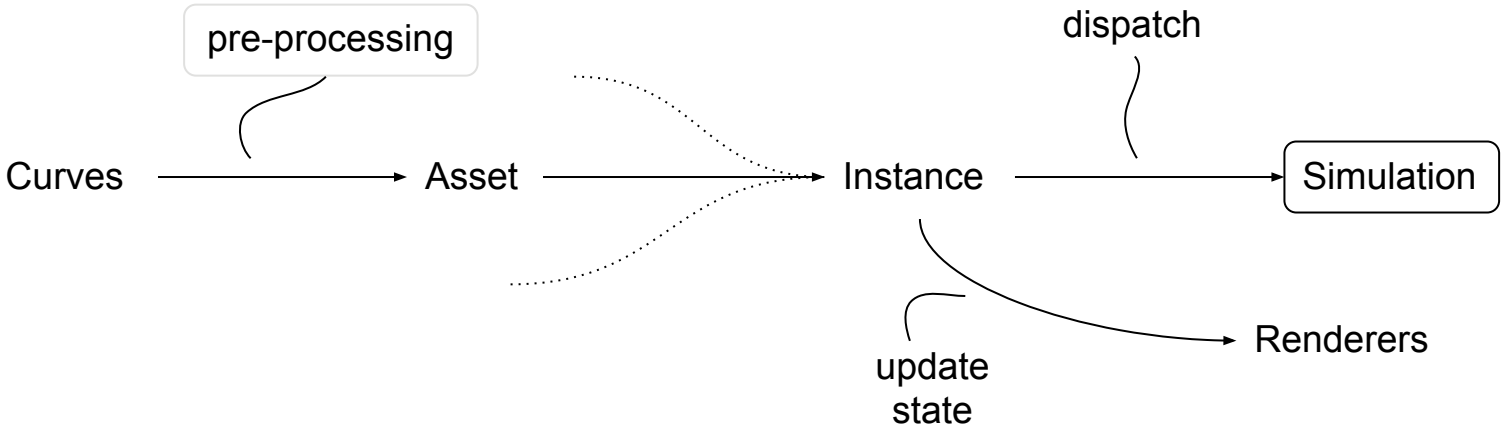
- No prior solution offered by Unity
  - Let's build something nice
  - Something complete and user-facing
- Requirements grew with work on the demo
  - "Long hair to match the actress'"
  - "Adding some curls sure would be nice"
  - "We're going to need to preserve clumps as well"
- Led to configurable system
  - Toggle features based on application needs



Overview

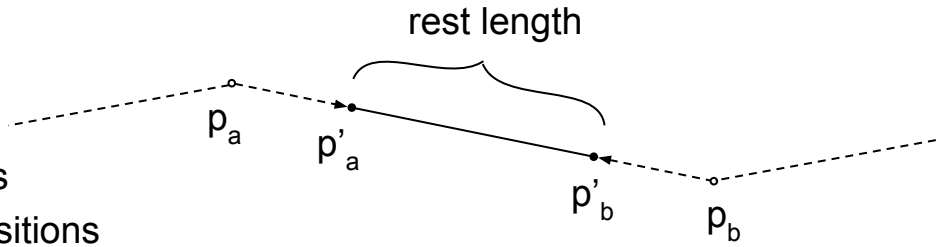


Overview

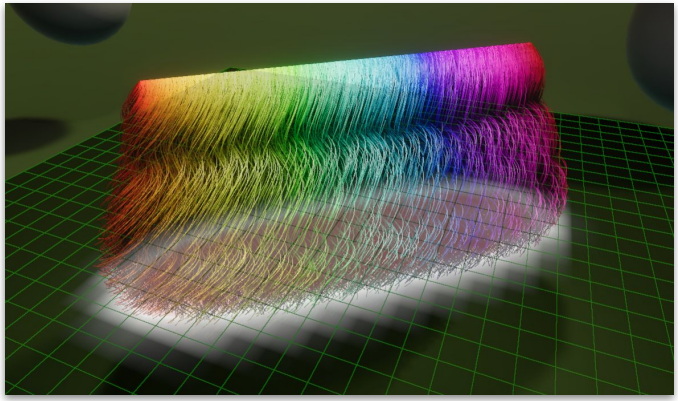


Simulation

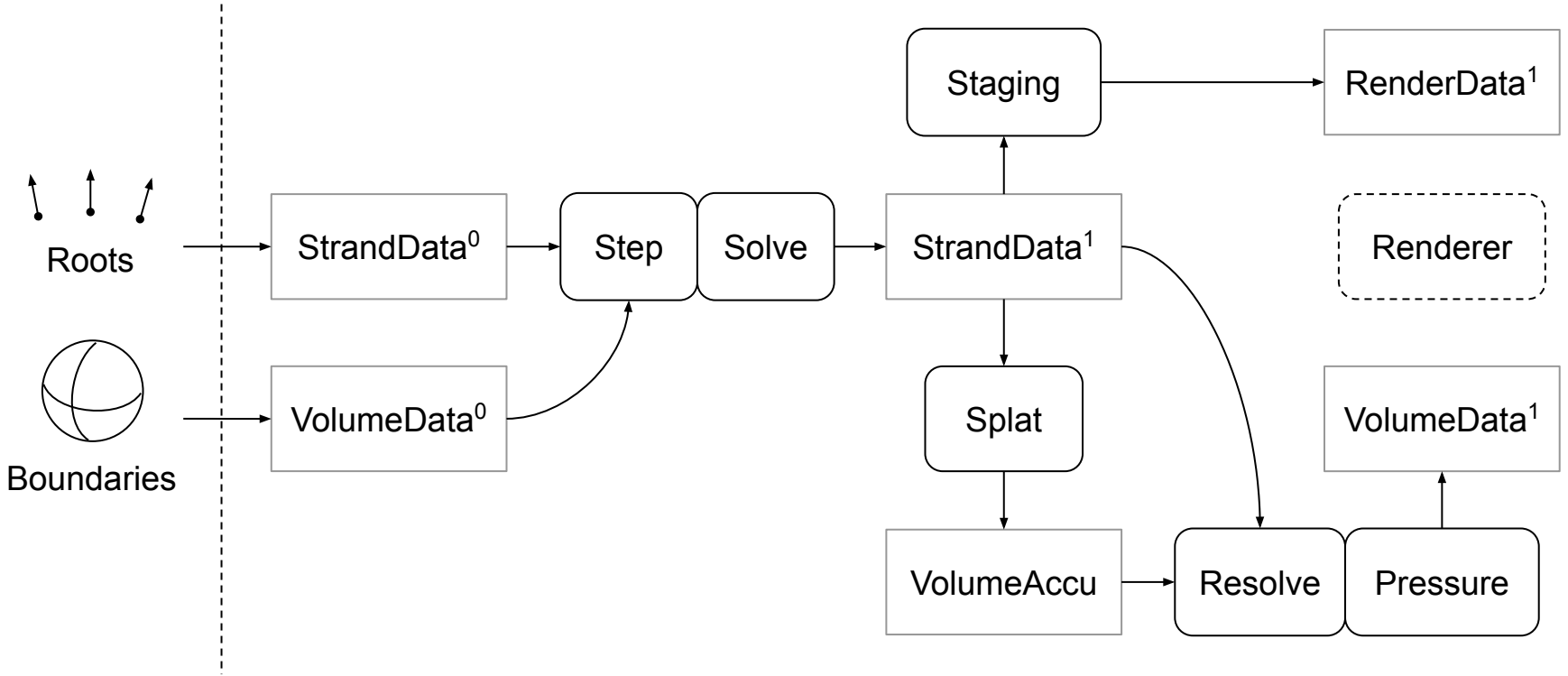
- Iterative strand solver
  - Regular PBD [Müller et al. 2006]
  - Update positions to satisfy constraints
  - Update velocities from changes in positions



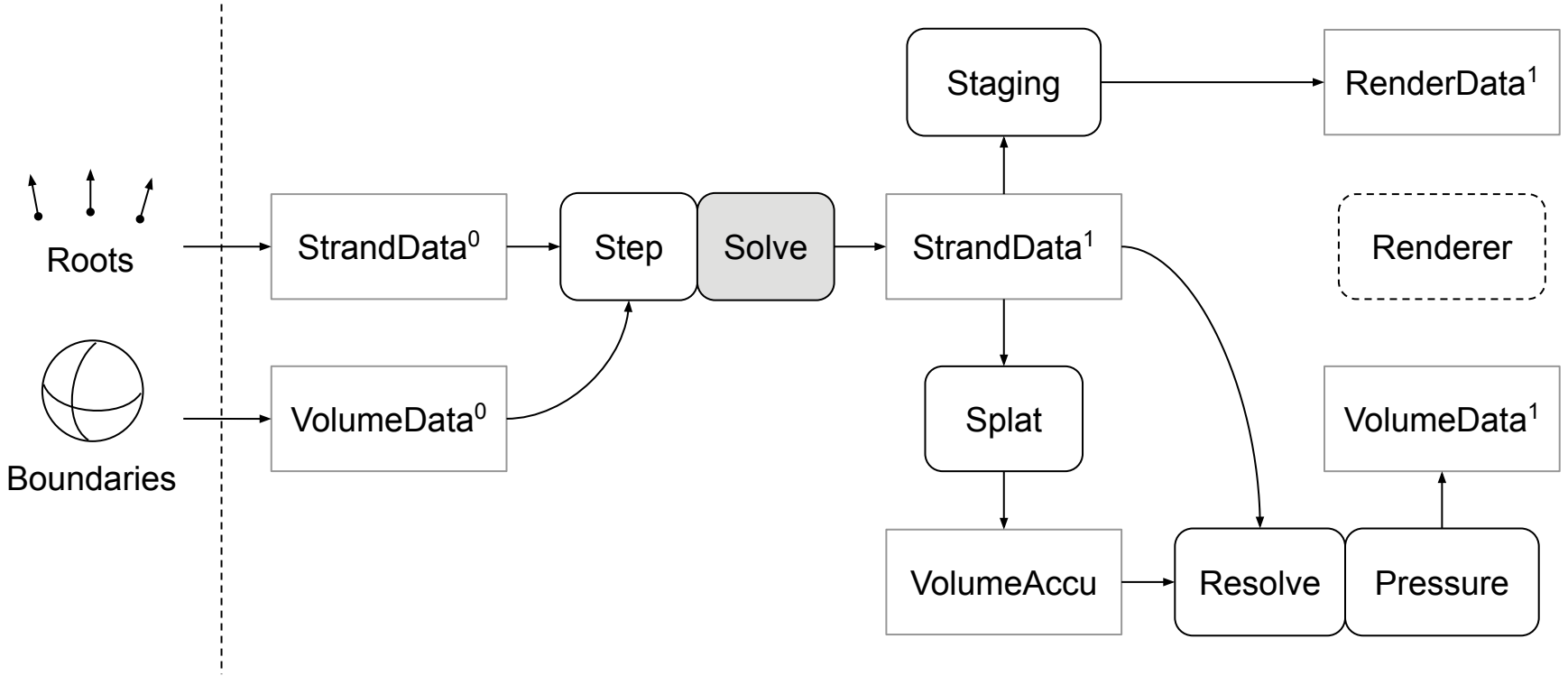
- Volume data
  - Splat particles to grid [Petrovic et al. 2005]
  - Iterative pressure solve [Harris 2004]
    - Poisson equation with divergence on rhs.
    - Modify divergence to include source term
  - Gives us density control and soft hair-hair interaction similar to [McAdams et al. 2009]



Simulation Frame



Simulation Frame

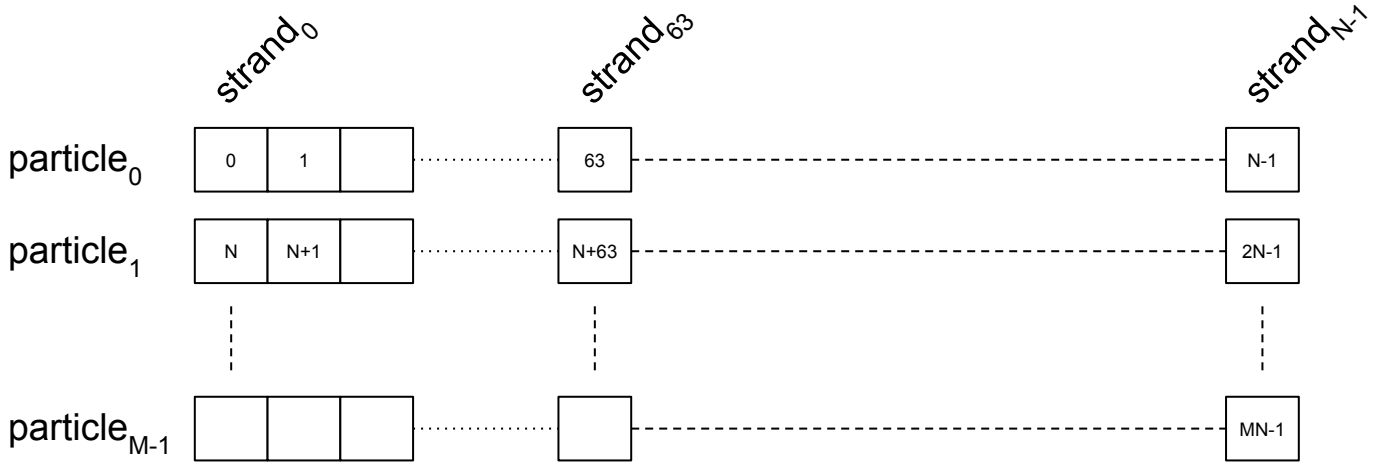


Strand Data

- Strands are pre-processed for the solver to assume certain things
  - Ordered by earliest LOD (that they are primary strands in)
  - Uniform number of particles per strand within group
  - Uniform particle spacing within strand
  - Particle data interleaved

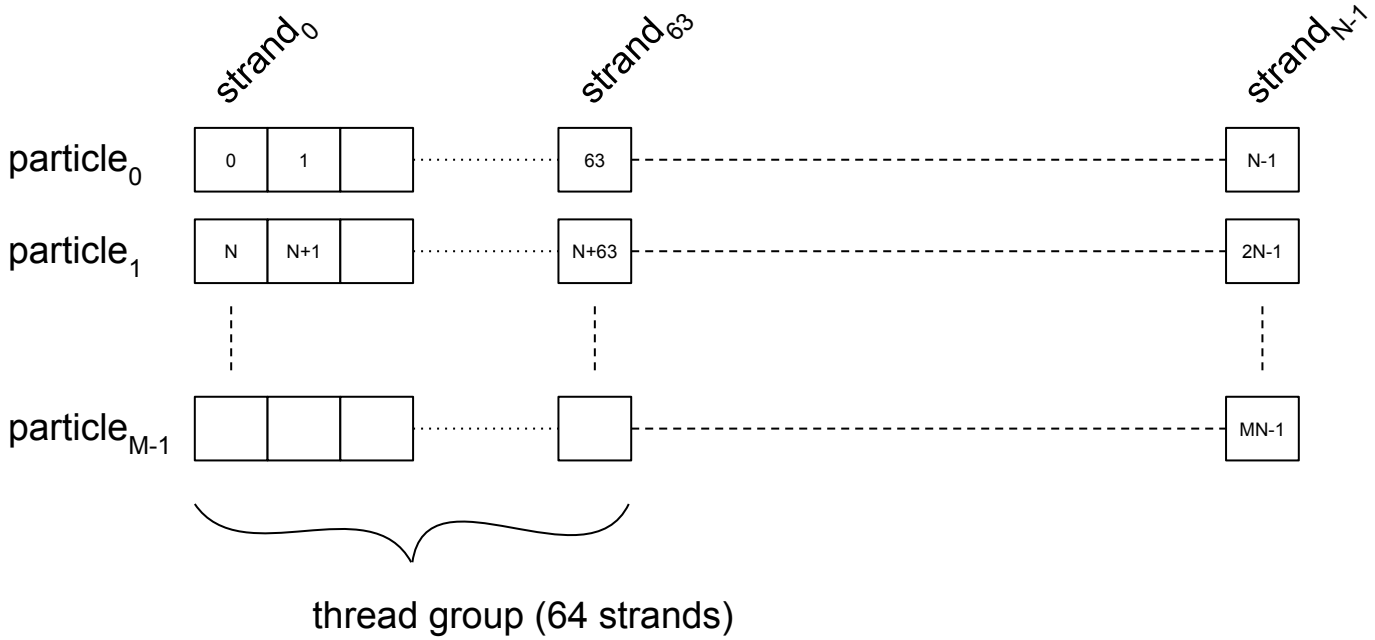
- Particle buffers
  - Position, updated per iteration
  - Velocity, updated per timestep
  - Rest pose root offset, written on init
  - Rest pose frame delta, written on init

Strand Data

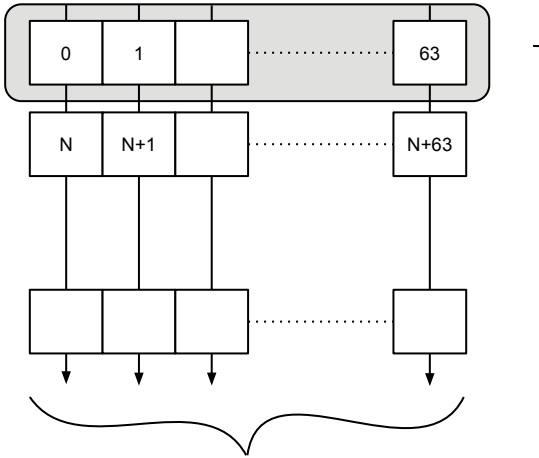


M particles per strand, N strands

Strand Data

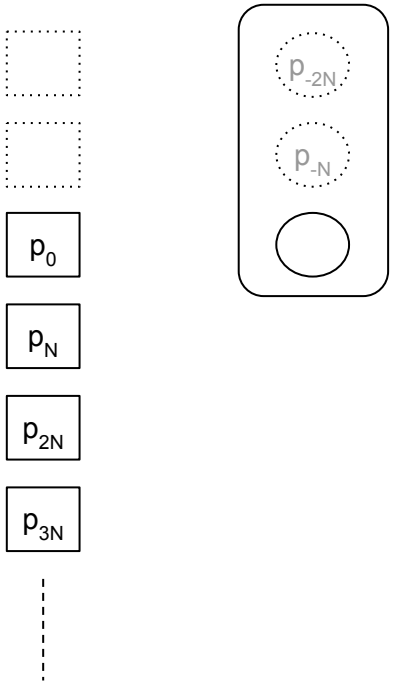


Strand Data



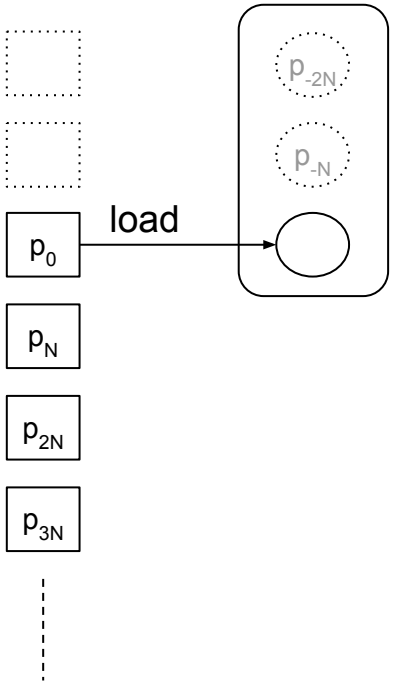
there are only few divergent paths – decent chance they'll stick together

Constraint Solve

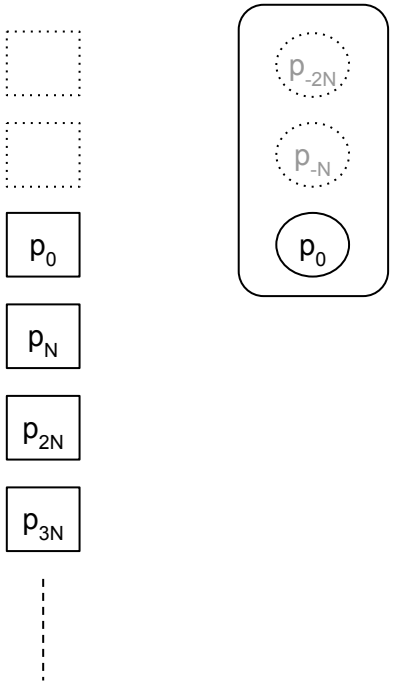


- Constraints are solved within a window
  - System only dealing with strands
  - Relationships are implicit
- Solver window
  - Initialized below the root of the strand
  - Moves down strand

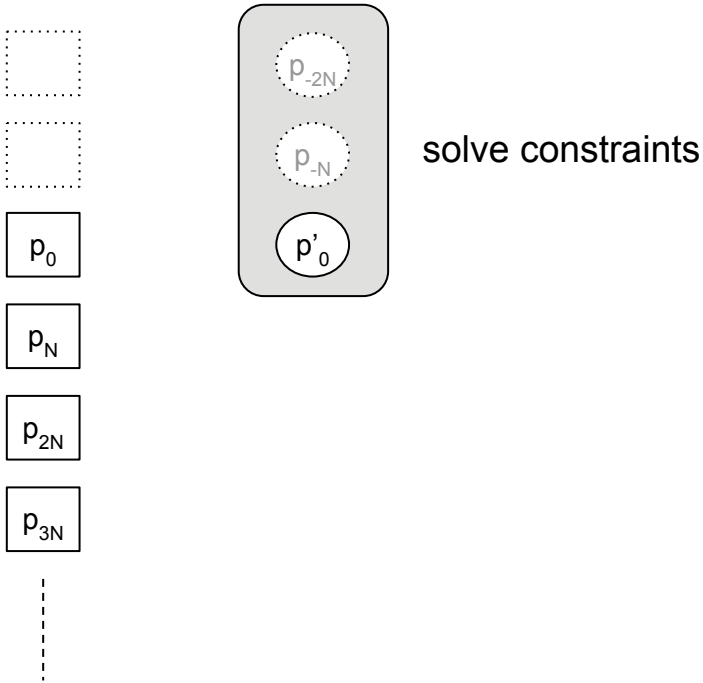
Constraint Solve



Constraint Solve

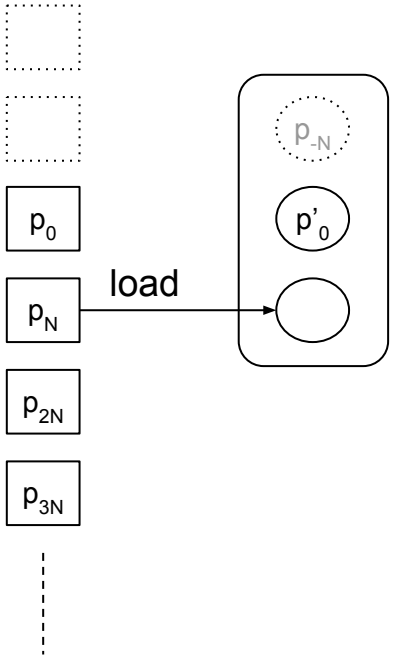


Constraint Solve

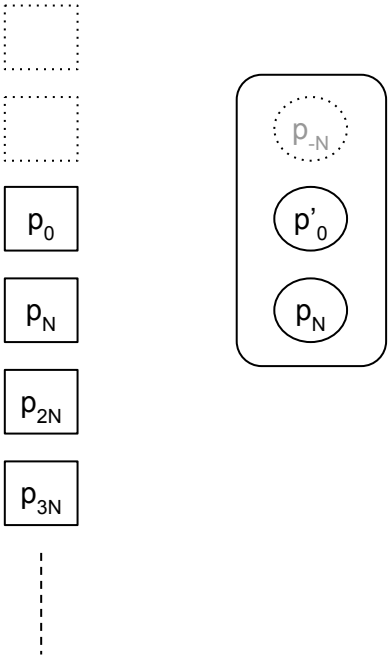


solve constraints

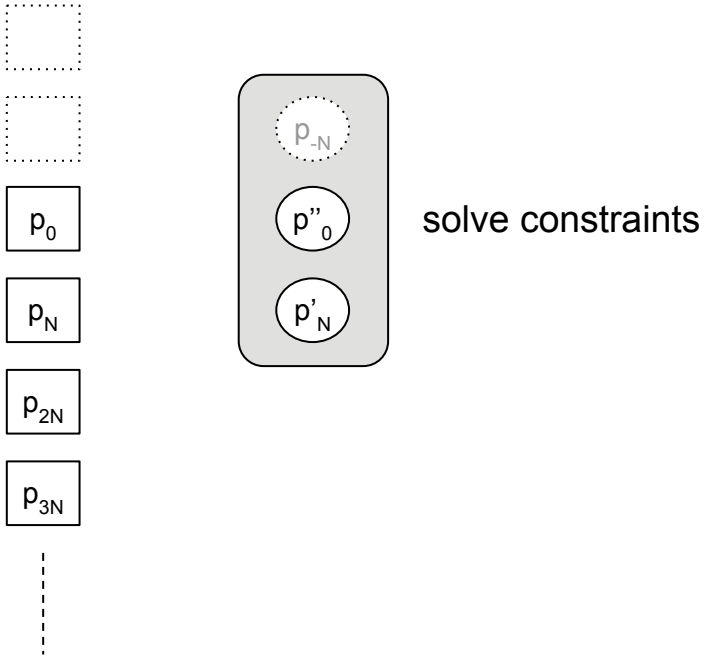
Constraint Solve



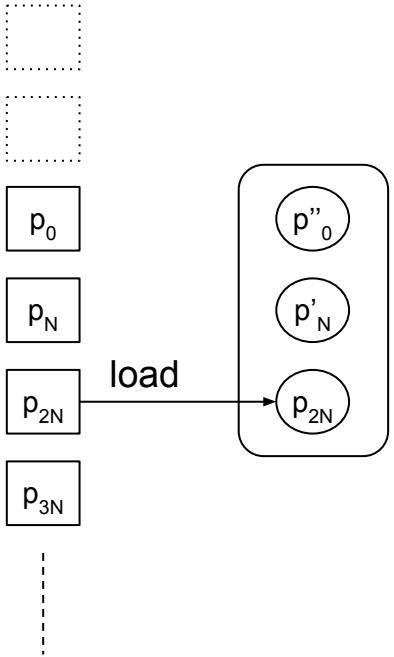
Constraint Solve



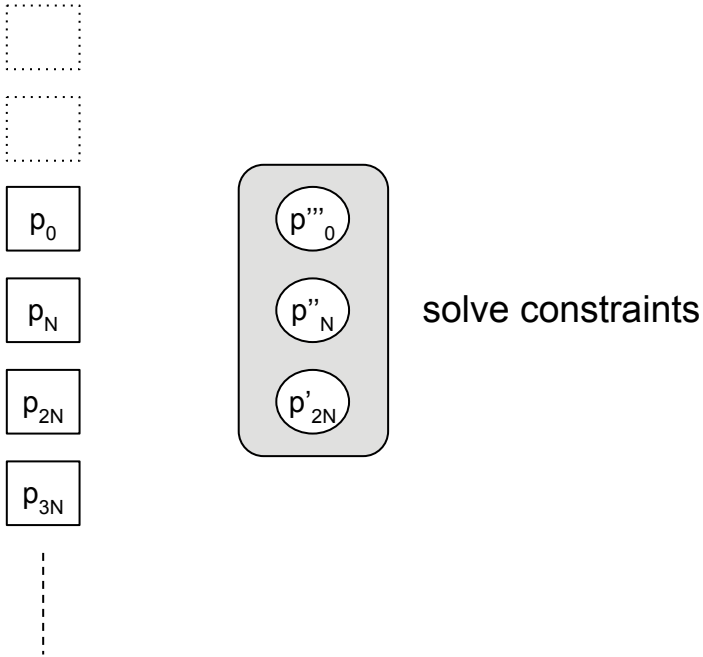
Constraint Solve



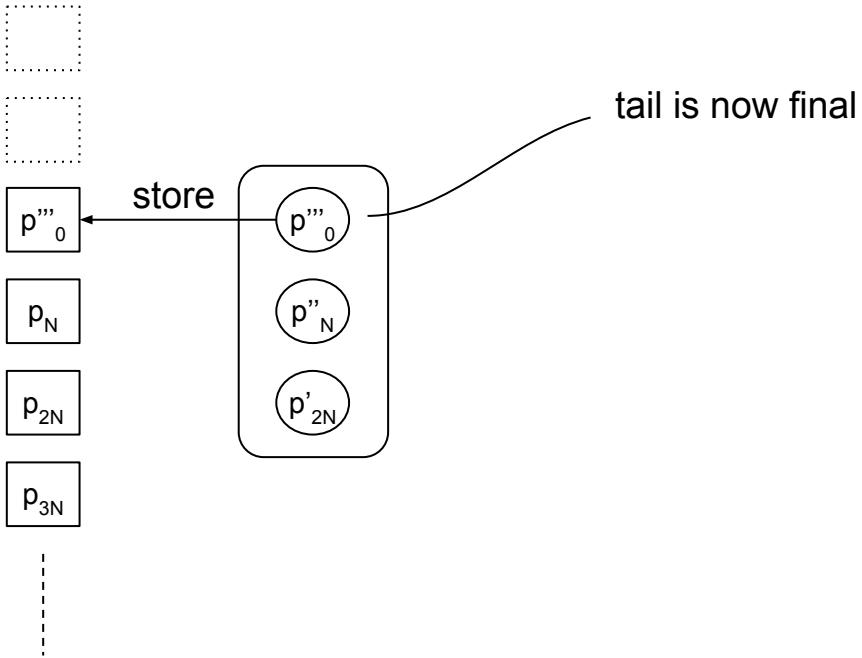
Constraint Solve



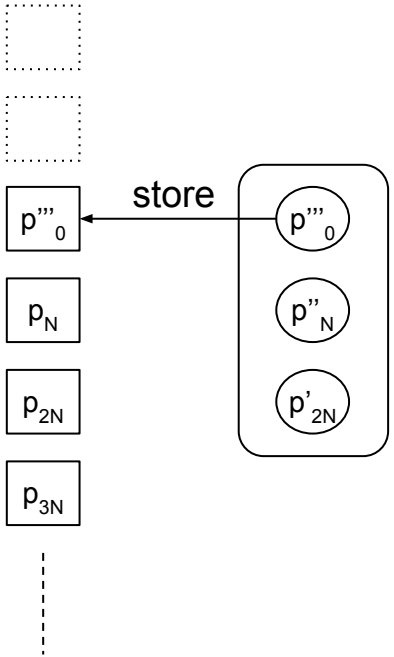
Constraint Solve



Constraint Solve

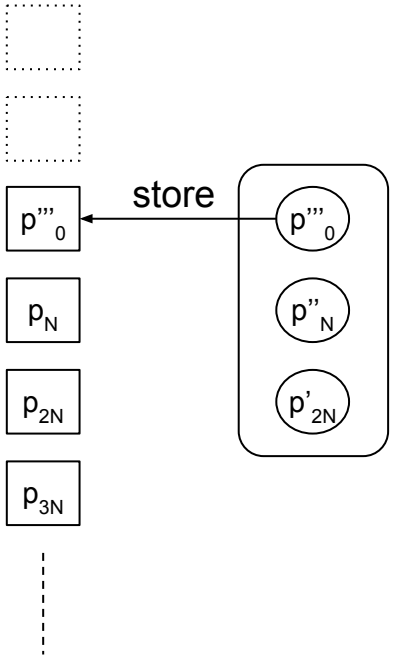


Constraint Solve



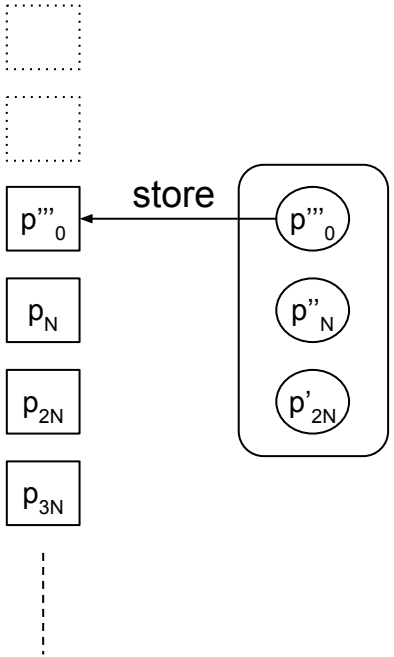
- Window size depends on active set of constraints
  - Constraints in window affect the same particle a fixed number of times down the strand
  - E.g. particle-particle distance constraint will affect each particle twice
- Compile-time variants for window size + feature flags for individual constraints
- Simpler setup => less we try to carry

Constraint Solve

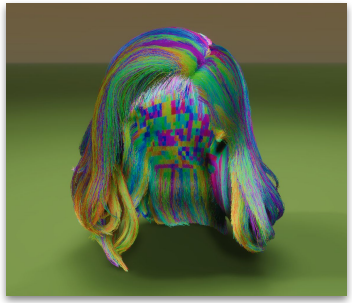


- Supported constraints
  - Boundary w/ friction [Macklin et al. 2014]
  - Distance particle-particle [Müller et al. 2006]
  - Distance particle-root [Kim et al. 2012]
  - Distance FTL [Müller et al. 2012]
  - Local bend limiter [Kelager et al. 2010]
  - Local shape [Kugelstadt and Schömer 2016]
  - Global rotation (local shape with bias)
- Order of evaluation is fixed

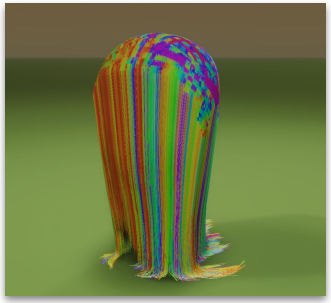
Local Shape



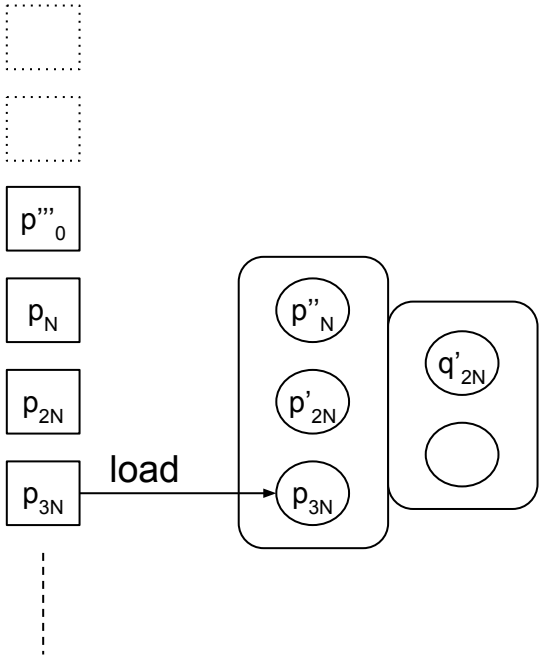
- What about orientation?
  - Fundamental to local shape
  - Resisting to load/store more data per particle
- … local shape is rather important ☺



vs.

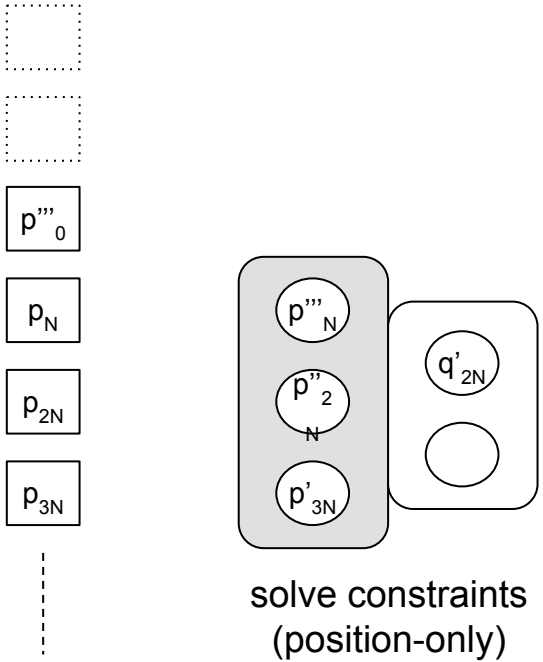


Local Shape



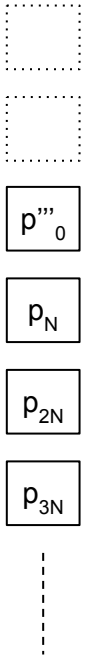
- What about orientation?
  - Fundamental to local shape
  - Resisting to load/store more data per particle
- Build frame as we move down the strand
  - Added window always has size 2

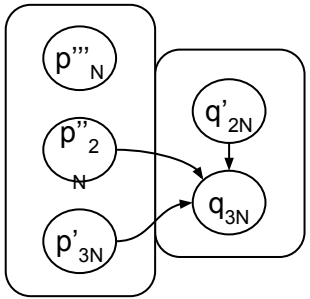
Local Shape



- What about orientation?
  - Fundamental to local shape
  - Resisting to load/store more data per particle
- Build frame as we move down the strand
  - Added window always has size 2
  - Position-only constraints handled first

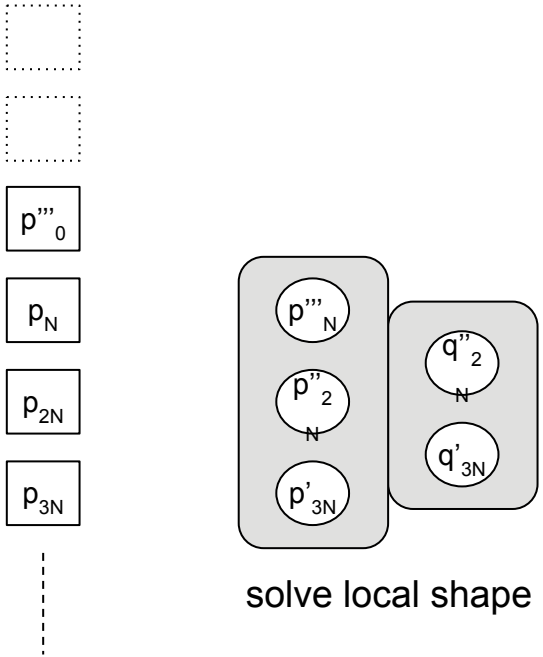
Local Shape





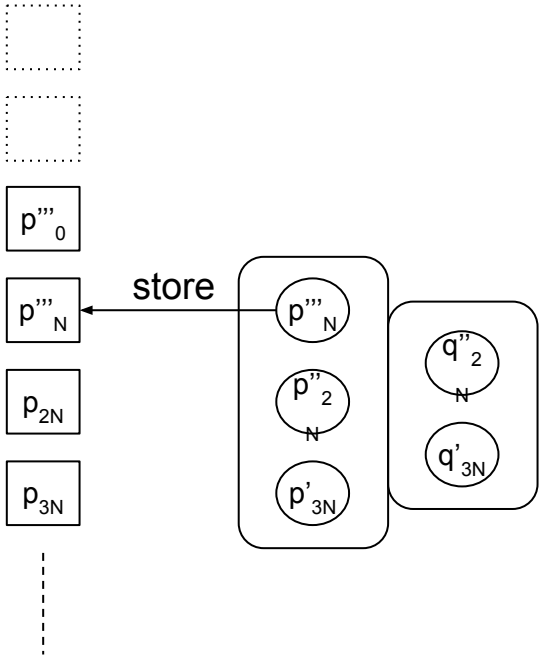
- What about orientation?
  - Fundamental to local shape
  - Resisting to load/store more data per particle
- Build frame as we move down the strand
  - Added window always has size 2
  - Position-only constraints handled first
  - Then
    - $dq = qfromto(qmul(q'_{2N}, e^3), p'_{3N} - p''_{2N})$
    - $q_{3N} = qmul(dq, q'_{2N})$
    - (e^3 = forward)

Local Shape



- What about orientation?
  - Fundamental to local shape
  - Resisting to load/store more data per particle
- Build frame as we move down the strand
  - Added window always has size 2
  - Position-only constraints handled first
  - Then
    - $dq = qfromto(qmul(q'_{2N}, e^3), p'_{3N} - p''_{2N})$
    - $q_{3N} = qmul(dq, q'_{2N})$

Local Shape



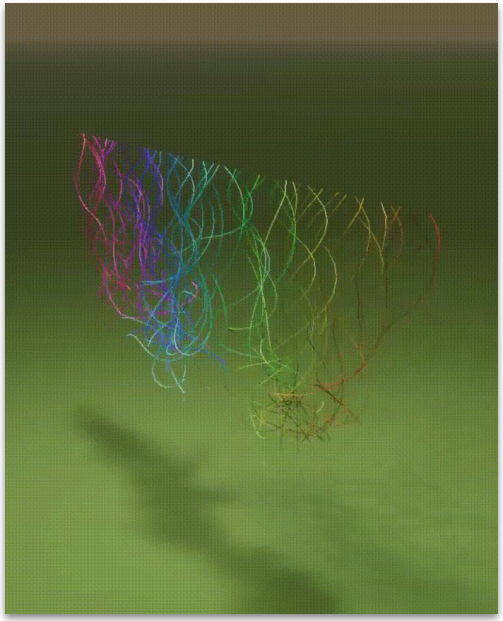
- What about orientation?
  - Fundamental to local shape
  - Resisting to load/store more data per particle
- Build frame as we move down the strand
  - Added window always has size 2
  - Position-only constraints handled first
  - Then
    - $dq = qfromto(qmul(q'_{2N}, e^3), p'_{3N} - p''_{2N})$
    - $q_{3N} = qmul(dq, q'_{2N})$
  - Still storing just position

Local Shape w/ inferred material frame

- Some immediate downsides
  - Always working with rotation minimizing frame
  - Twist not preserved between iterations/steps
  - Can’t twist strands to make them coil
  - Can’t render e.g. textured tubes that visibly twist

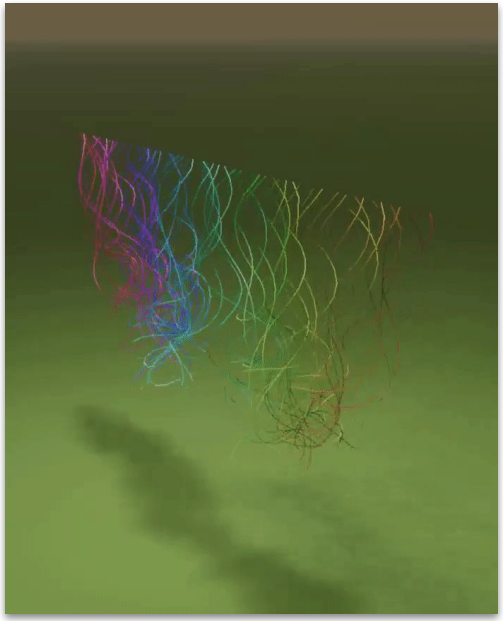
Local Shape w/ inferred material frame

- Some immediate downsides
  - Always working with rotation minimizing frame
  - Twist not preserved between iterations/steps
  - Can’t twist strands to make them coil
  - Can’t render e.g. textured tubes that visibly twist
- More problematic: Stability 😱
  - [Kugelstadt et al. 2016] suggest not solving in sequential order => would require storing (q, w)



Local Shape w/ inferred material frame

- Some immediate downsides
  - Always working with rotation minimizing frame
  - Twist not preserved between iterations/steps
  - Can’t twist strands to make them coil
  - Can’t render e.g. textured tubes that visibly twist
- More problematic: Stability
  - [Kugelstadt et al. 2016] suggest not solving in sequential order => would require storing (q, w)
  - Bring out the workarounds ☺
    - Angular damping, bias towards reference, composition

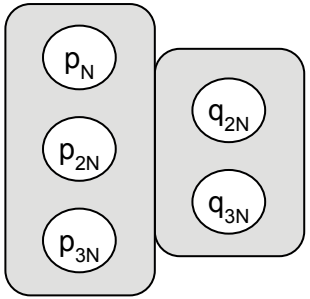


Local Shape w/ inferred material frame

- Angular damping
  - Per-segment during velocity update
    - w = angular(r, v)
    - w_{damp} = decay(w)
    - dv = linear(r, w_{damp} - w)
- Bias towards reference
  - Internally, local shape is a composition of two constraints defined by [Kugelstadt et al. 2016]
    - bend-twist(q_a,q_b,w_a,w_b, ...)
    - stretch-shear(p_a,p_b,q, ...)
  - w_a = 0.5 makes q_a more resistant to change

Local Shape w/ inferred material frame

- Composition of
  - bend-twist(q_a,q_b, ...)
  - stretch-shear(p_a,p_b,q, ...)

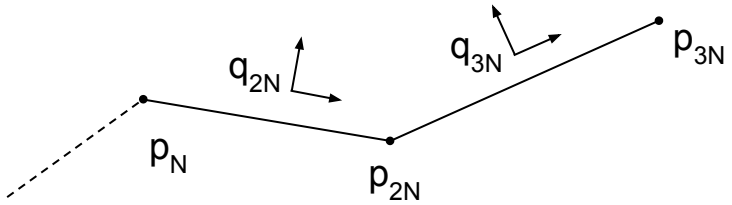


solve local shape

- We have two modes
  - “Forward”
  - “Stitched"

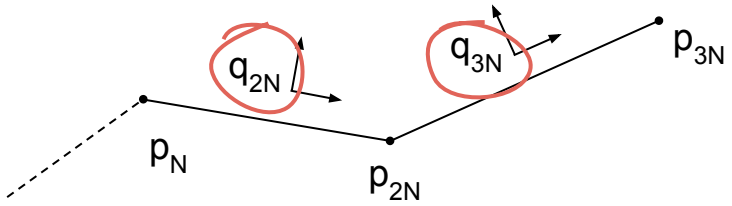
Local Shape w/ inferred material frame

- “Forward”
  - bend-twist(q_{2N},q_{3N}, ...)
  - stretch-shear(p_{2N},p_{3N},q_{3N}, ...)



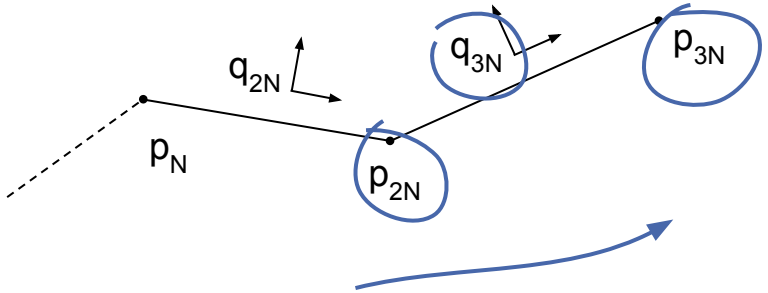
Local Shape w/ inferred material frame

- “Forward”
  - **bend-twist(q_{2N},q_{3N}, ...)**
  - stretch-shear(p_{2N},p_{3N},q_{3N}, ...)



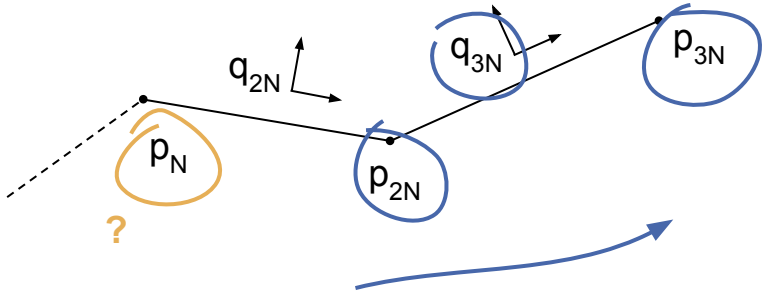
Local Shape w/ inferred material frame

- “Forward”
  - bend-twist(q_{2N},q_{3N}, ...)
  - **stretch-shear(p_{2N},p_{3N},q_{3N}, ...)**



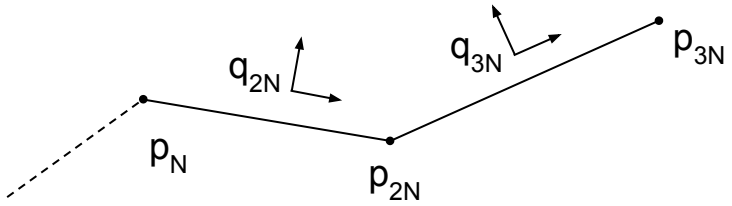
Local Shape w/ inferred material frame

- “Forward”
  - bend-twist(q_{2N},q_{3N}, ...)
  - **stretch-shear(p_{2N},p_{3N},q_{3N}, ...)**



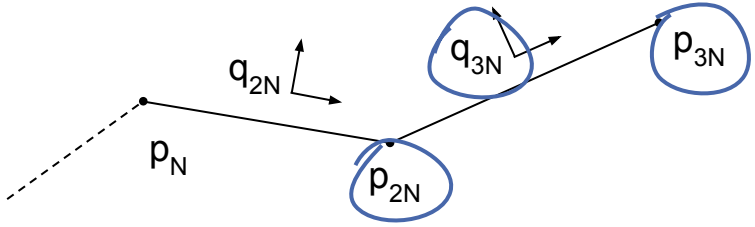
Local Shape w/ inferred material frame

- “Stitched”
  - stretch-shear(p_{2N},p_{3N},q_{3N}, ...)
  - stretch-shear(p_N,p_{2N},q_{2N}, ...)
  - bend-twist(q_{2N},q_{3N}, ...)



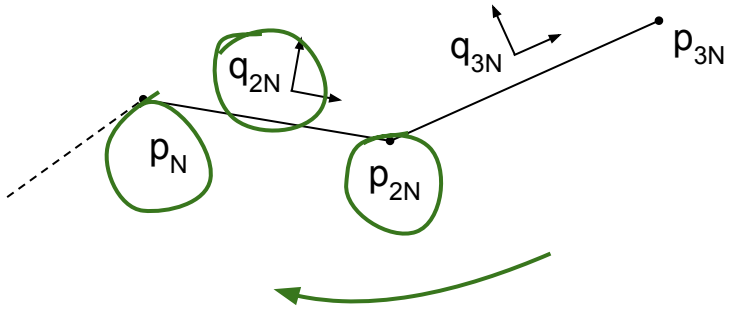
Local Shape w/ inferred material frame

- “Stitched”
  - **stretch-shear(p_{2N},p_{3N},q_{3N}, ...)**
  - stretch-shear(p_N,p_{2N},q_{2N}, ...)
  - bend-twist(q_{2N},q_{3N}, ...)



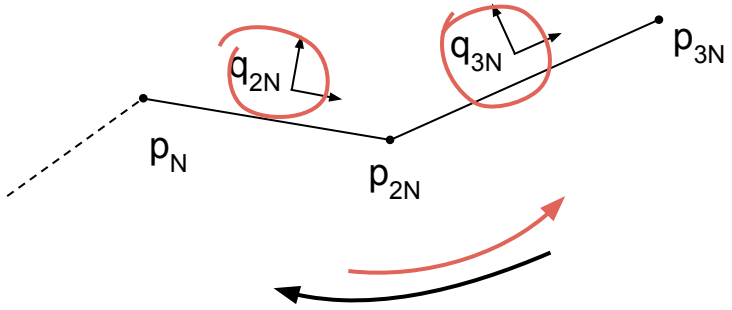
Local Shape w/ inferred material frame

- “Stitched”
  - stretch-shear(p_{2N},p_{3N},q_{3N}, ...)
  - **stretch-shear(p_N,p_{2N},q_{2N}, ...)**
  - bend-twist(q_{2N},q_{3N}, ...)



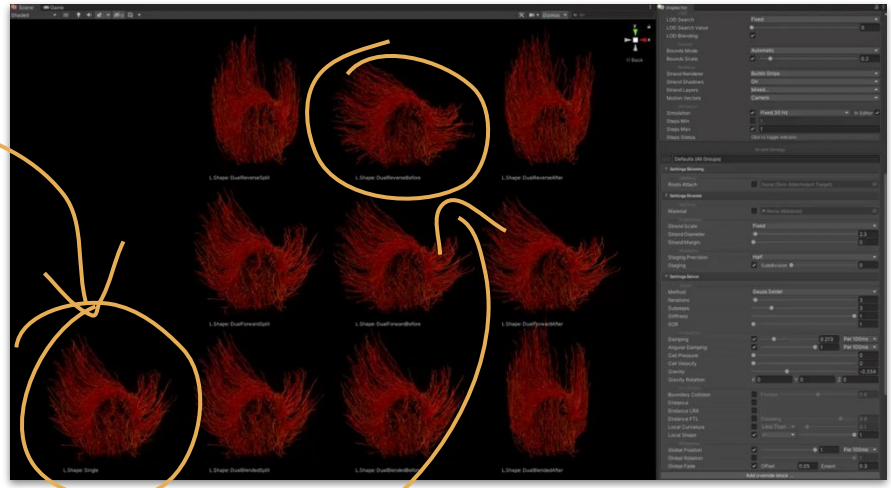
Local Shape w/ inferred material frame

- “Stitched”
  - stretch-shear(p_{2N},p_{3N},q_{3N}, ...)
  - stretch-shear(p_N,p_{2N},q_{2N}, ...)
  - **bend-twist(q_{2N},q_{3N}, ...)**



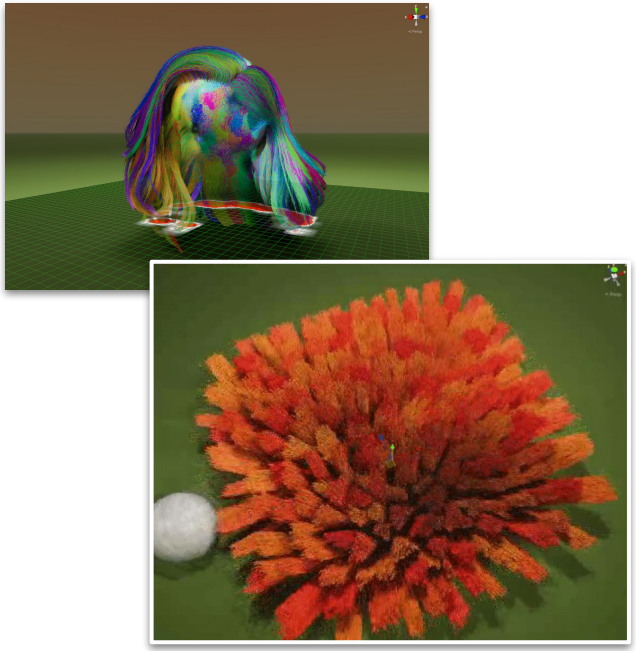
Local Shape w/ inferred material frame

- “Forward”
  - First / immediate approach
  - Carries information only forward
  - Thought maybe we can do better
- “Stitched”
  - After parallel implementations
  - Carries information back and forth within window (hence the name)
  - Converges faster



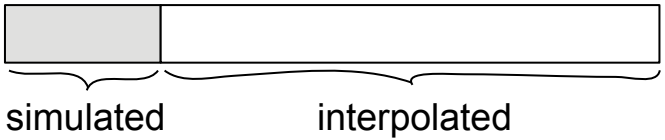
Clustering

- LOD data from just curves
  - Cluster roots, full strands, three-point simplified
  - Centroids => primary strands of single LOD
- Artistic use
  - Initially we were just simulating all the strands
    - No physical model of strands clumping together
    - Clumps easily diffusing != artistic vision
  - Simulating reduced set helps preserve clumps
    - Also cheaper
- Scalability
  - Primary strands carry volume of cluster



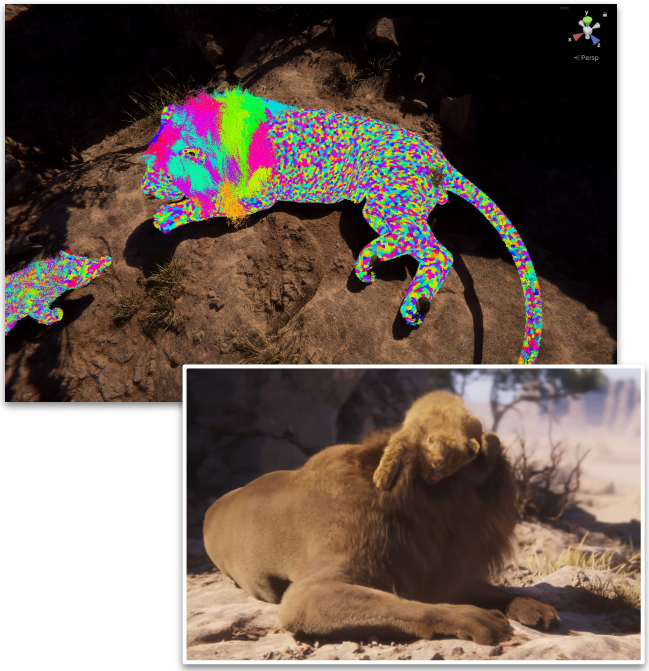
Clustering

- Ordering



- Simulation fully supports the data
  - Volume and collisions intact for lower LODs
  - Enables higher density grooms without completely breaking the budget

- See the ‘Lion’ demo
  - ~2M strands
  - Smart use of clustering
  - ~4ms sim. on PS5 (w/ volume)



References

- [Bender et al. 2015] Position-Based Simulation Methods in Computer Graphics
- [Bridson and Müller-Fischer 2007] Fluid Simulation SIGGRAPH 2007 Course Notes
- [Gibou et al. 2002] A Second Order Accurate Symmetric Discretization of the Poisson Equation on Irregular Domains
- [Harris 2004] Fast Fluid Dynamics Simulation on the GPU
- [Kelager et al. 2010] A Triangle Bending Constraint Model for Position-Based Dynamics
- [Kim et al. 2012] Long Range Attachments - A Method to Simulate Inextensible Clothing in Computer Games
- [Kugelstadt and Schömer 2016] Position and Orientation Based Cosserat Rods
- [Losasso et al. 2008] Two-Way Coupled SPH and Particle Level Set Fluid Simulation
- [Macklin et al. 2014] Unified particle physics for real-time application
- [Macklin et al. 2019] Small Steps in Physics Simulation
- [McAdams et al. 2009] Detail Preserving Continuum Simulation of Straight Hair
- [Müller et al. 2006] Position Based Dynamics
- [Müller et al. 2012] Fast Simulation of Inextensible Hair and Fur
- [Petrovic et al. 2005] Volumetric Methods for Simulation and Rendering of Hair
- [Zhu and Bridson 2005] Animating Sand as a Fluid