Jerry 7/18

Motivations

- Bottleneck: Collision detection
  - 50%+ of computation is used on collision ALONE
- Step size limits possible movement per detection
  - How do we provide penetration free WITHOUT detection?

Penetration free deformations:

Deform the object to *approximately* match past movement BEFORE full detection.

Problem statement

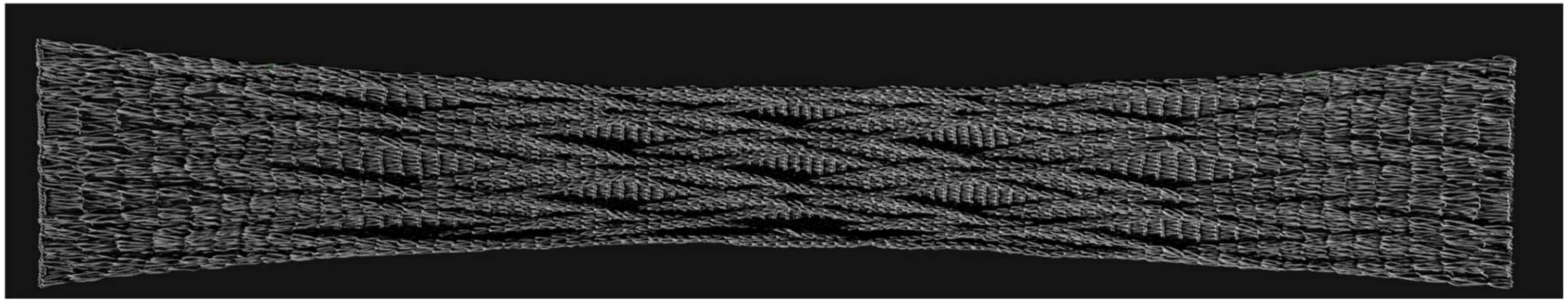
- Problem: Deform current position x_0 to be close to the predicted position $\tilde{x} \approx 2x^t - x^{t-1}$ such that the line from x_0 to x has no penetrations.

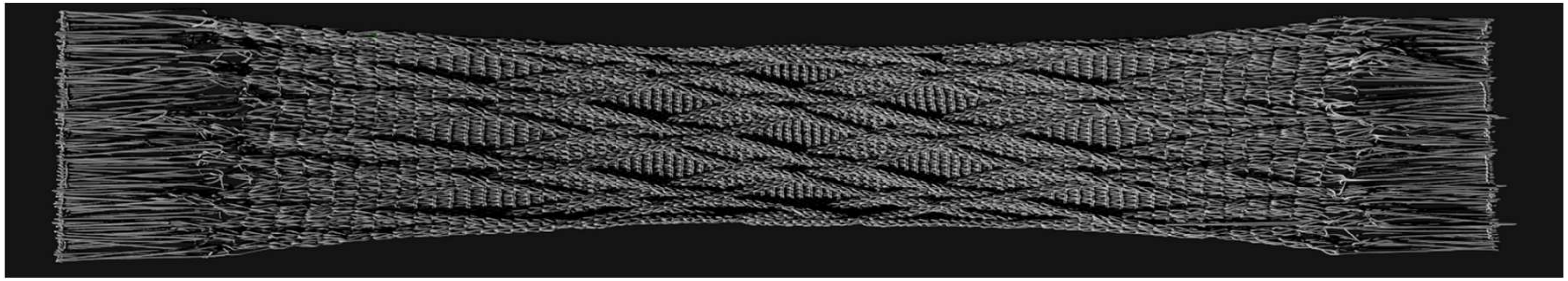
$$\min_x |x - \tilde{x}|^2$$
$$s.t.\ \forall t \in [0, 1] : x_0 + t\,(x - x_0)\ is\ penetration\ free$$

- We want to ...
  - Maximize the likelihood that $\tilde{x}$ is within the trust region of x
    - The closer we are, the less detections we need!
  - Provide non-penetration guarantee
    - Deformation gradient > 0 everywhere
  - Use no expensive neighborhood queries

Example: Affine least squares

- Fit affine transformation to deformation and apply to position before collision query.
    - Affine transformations mean guaranteed penetration free





Affine linear least squares

- For the i'th vertex, the target motion is Δx_i such that the next position is $\mathrm{x_i} + \Delta x_i$
- An affine transformation takes the form

$$f(x) = \begin{bmatrix} a & b & c \\ d & e & f \\ h & j & m \end{bmatrix} x_i + \begin{bmatrix} o \\ p \\ q \end{bmatrix} \approx \Delta x_i$$

We want it approximately match $\Delta x_i\ \forall i$

Ie. $\mathbf{min}\ \sum_i |\boldsymbol{f}(\boldsymbol{x_i}) - \Delta \boldsymbol{x_i}|^2$

Rewriting

- We can rewrite

$$f(x_i) - \Delta x_i = \begin{bmatrix} ax_{x,i} + bx_{y,i} + cx_{z,i} + o\ 1 \\ dx_{x,i} + cx_{y,i} + fx_{z,i} + p\ 1 \\ hx_{x,i} + jx_{y,i} + mx_{z,i} + q\ 1 \end{bmatrix} - \Delta x_i$$

$$x_i, \Delta x_i, and\ 1\ are\ all\ known!\ Move\ constants\ to\ matrix\ \mathbf{A_i}\ \ \boldsymbol{s.t}.$$

$$f(x_i) - \Delta x_i = \mathbf{A_i} \begin{bmatrix} a \\ \vdots \\ q \end{bmatrix} - \Delta x_i = \mathbf{A_i} v - \Delta x_i$$

More rewriting

$$\sum_i |f(x_i) - \Delta x_i|^2 = \sum_i |\boldsymbol{A}_i v - \Delta x_i|^2$$

$$= \left|\begin{bmatrix} \mathbf{A}_0 \\ \vdots \\ \boldsymbol{A}_{N-1} \end{bmatrix} v - \Delta x\right|^2 = |\mathbf{A}v - \Delta x|^2$$

$$min \sum_i |f(x_i) - \Delta x_i|^2 = min\ \frac{1}{2}|\mathbf{A}v - \Delta x|^2$$

Linear least squares

- This is just linear least squares!

$$min_v \ \frac{1}{2}|\mathbf{A}v - \Delta x|^2$$

$$\rightarrow v = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\Delta x$$

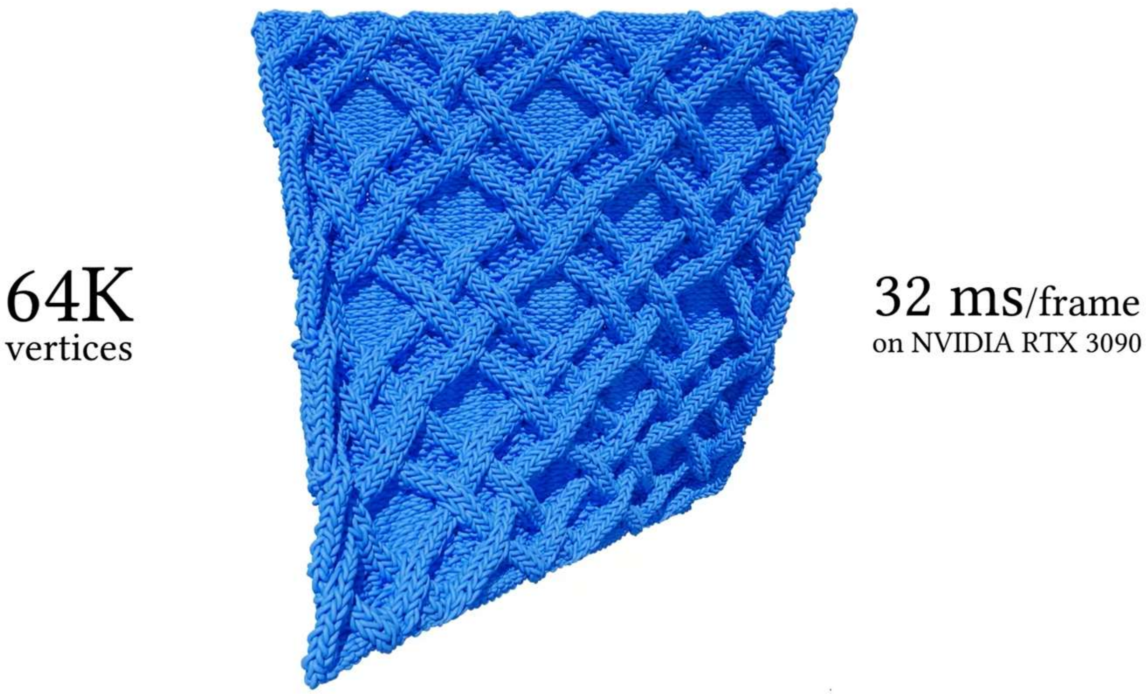
See https://en.wikipedia.org/wiki/Linear_least_squares

for linear least squares math

Affine transformation

- We can a transformation from v
  - Recall $\begin{bmatrix} a & b & c \\ d & e & f \\ h & j & m \end{bmatrix} x_i + \begin{bmatrix} o \\ p \\ q \end{bmatrix}$ and $v = \begin{bmatrix} a \\ \vdots \\ q \end{bmatrix}$
- Project $\begin{bmatrix} a & b & c \\ d & e & f \\ h & j & m \end{bmatrix}$ to be positive definite using SVD and apply to x_i
  - Clamp all eigenvalues to be greater than 0
  - See https://en.wikipedia.org/wiki/Singular_value_decomposition

Test case: Can we do better?



- The meshes Anka has in mind
- I can provide sample frame data for <- this
  - Slightly easier as line segments