Subspace Gradient Domain Mesh Deformation

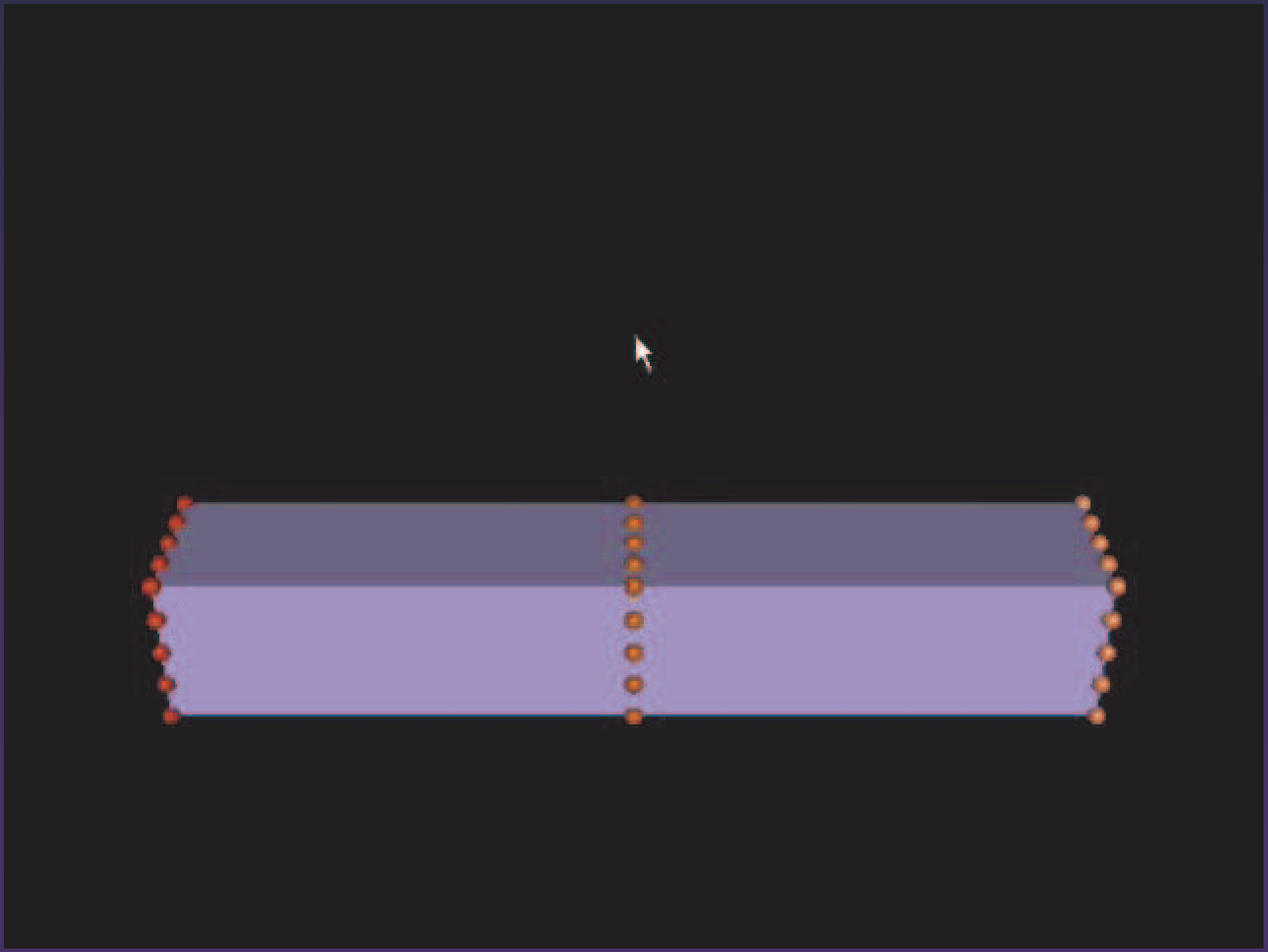
Scott Schaefer

TAMU
Computer
Science

Contributions

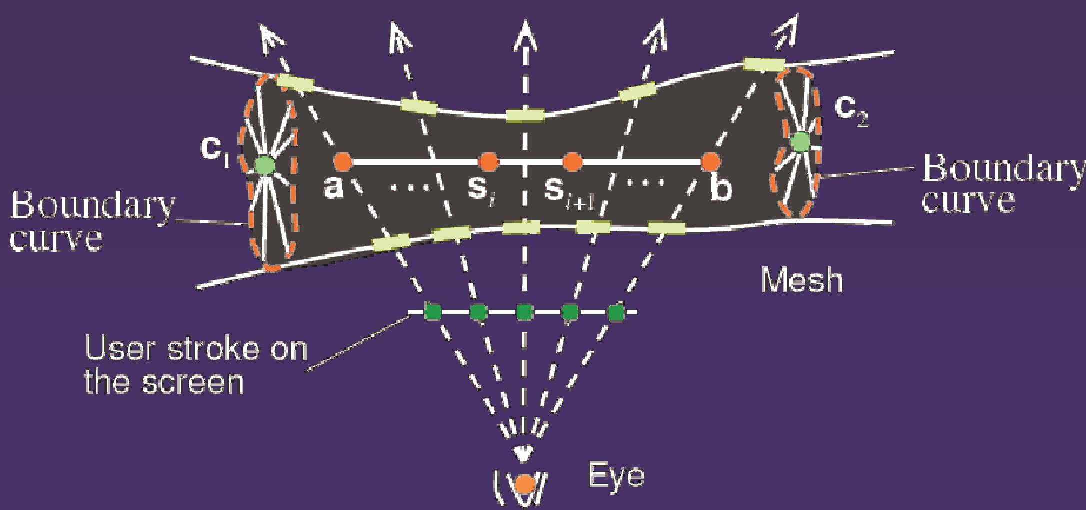
- Framework for constrained deformation
  - Skeletal constraints
  - Volume preservation
  - Projection-based manipulation
  - Detail preservation function
  - Fast non-linear, sub-space solver

Skeletal Constraints



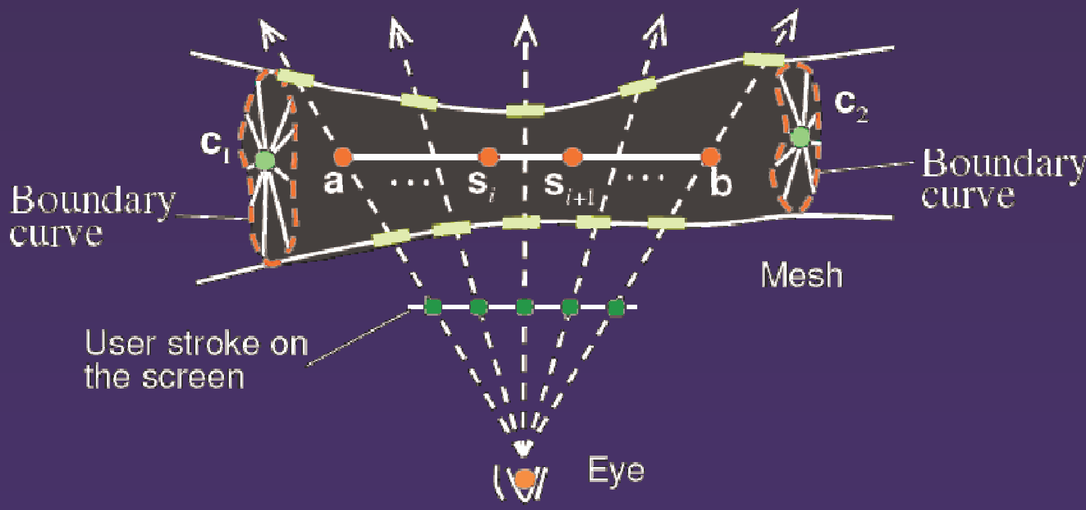
Constructing Bones

- User drags a line in screen space
- For each pixel
  - Find first two intersections with surface
- Fit a least squares line to all midpoints



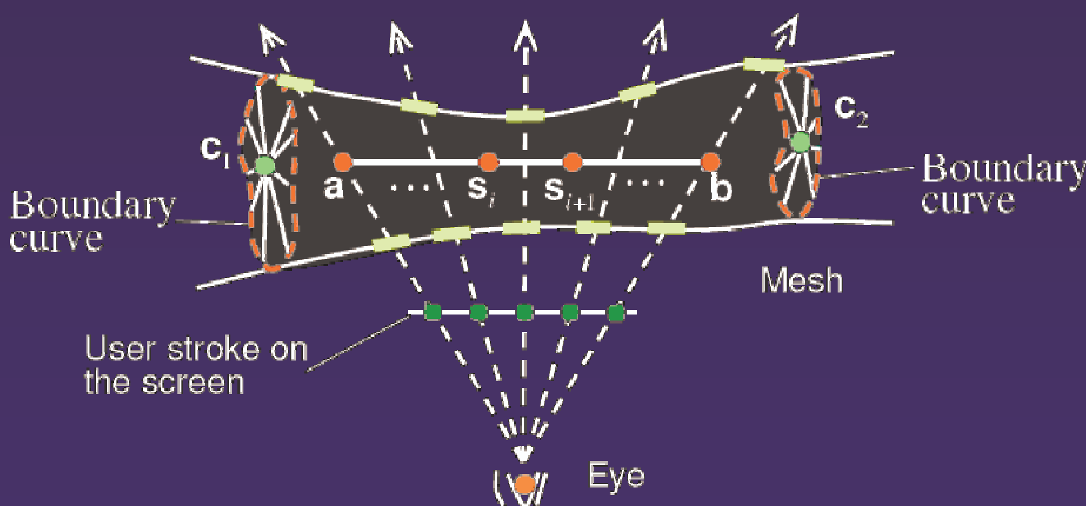
Region of Influence

- Construct supporting planes (normal perpendicular to ab) at end-points
- Flood from intersection triangles outward until all connected

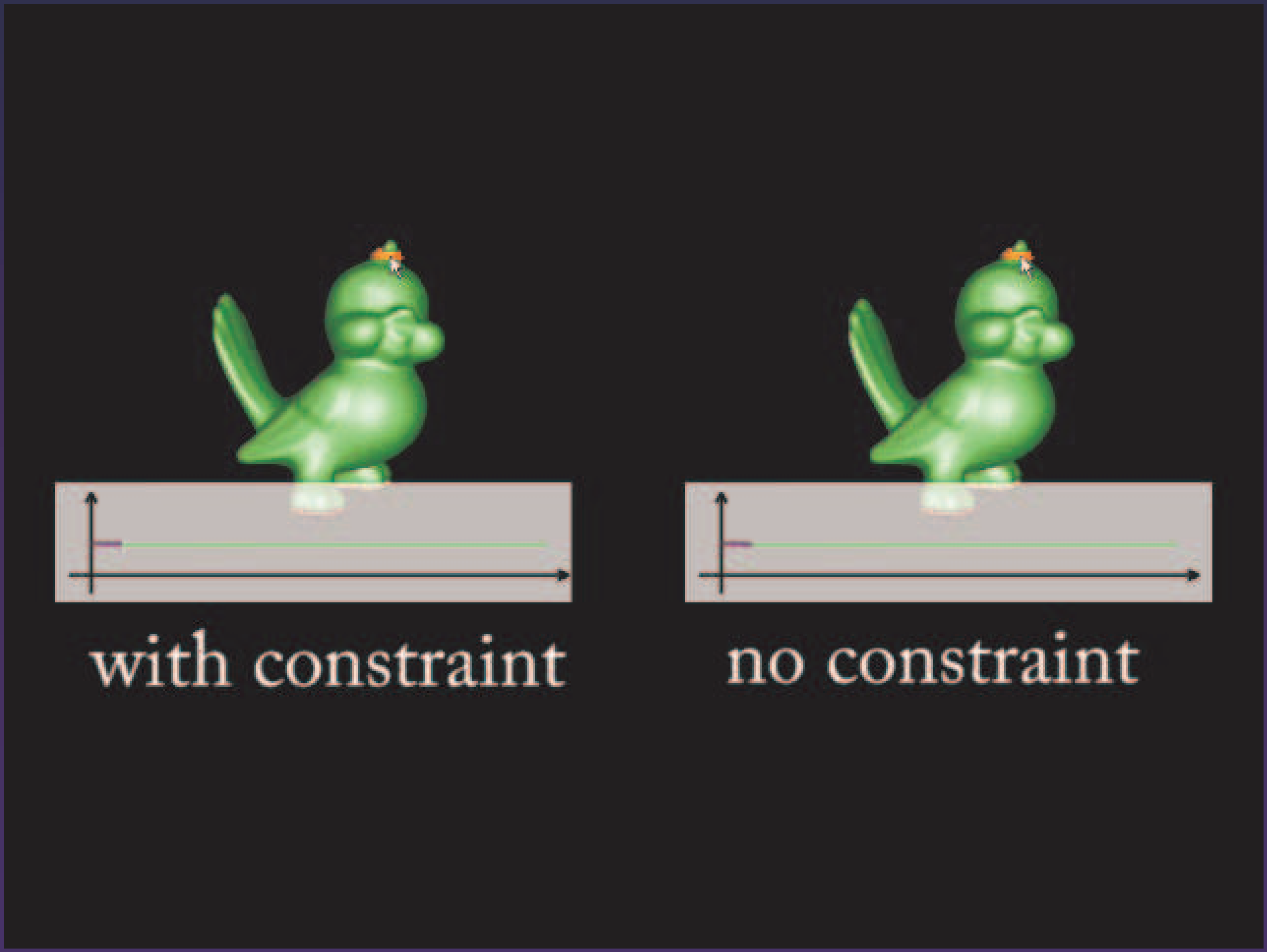


Mathematical Constraint

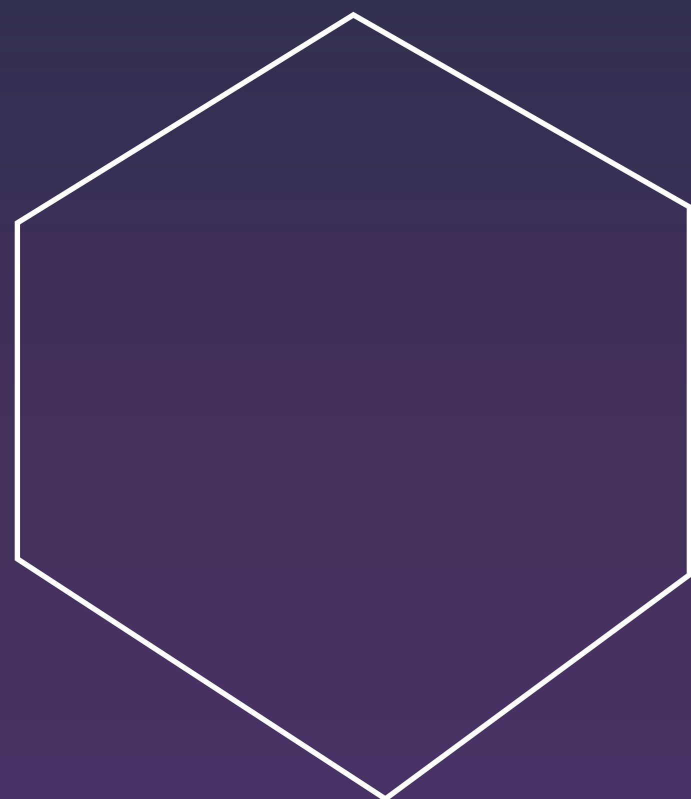
- For each sample point along ab
  - Compute MV coordinates with respect to vertices in region of influence
  - Close mesh by fanning to centroid on ends



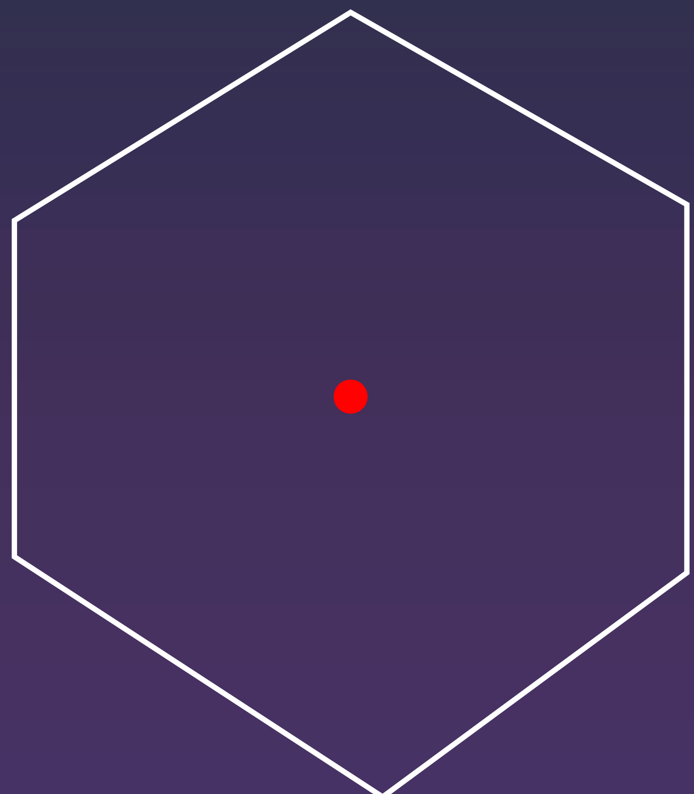
Volume Preservation



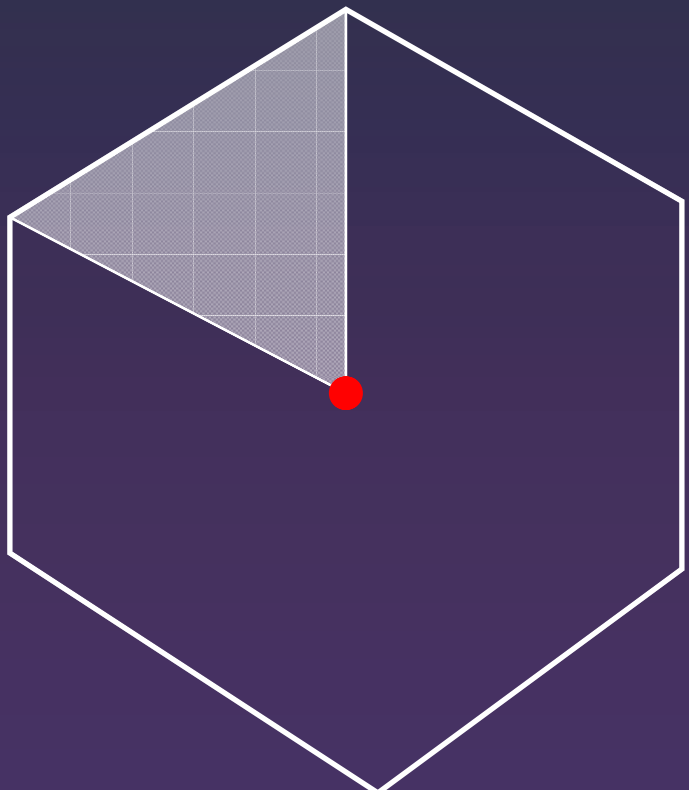
Calculating Area



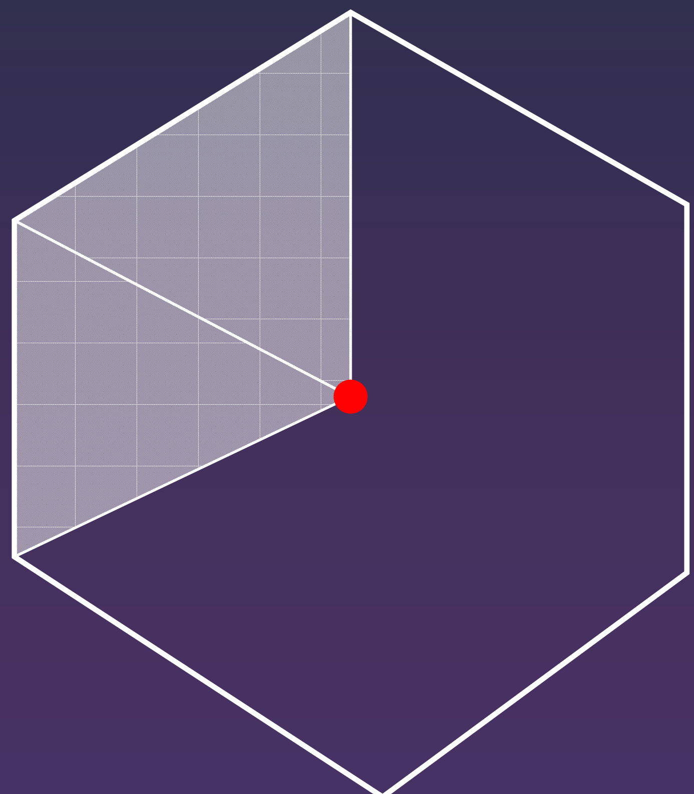
Calculating Area



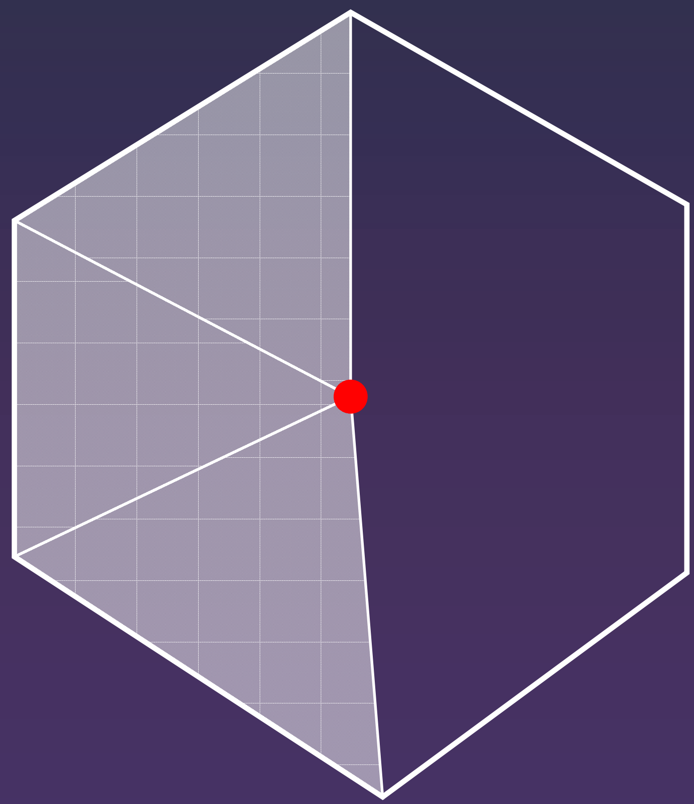
Calculating Area



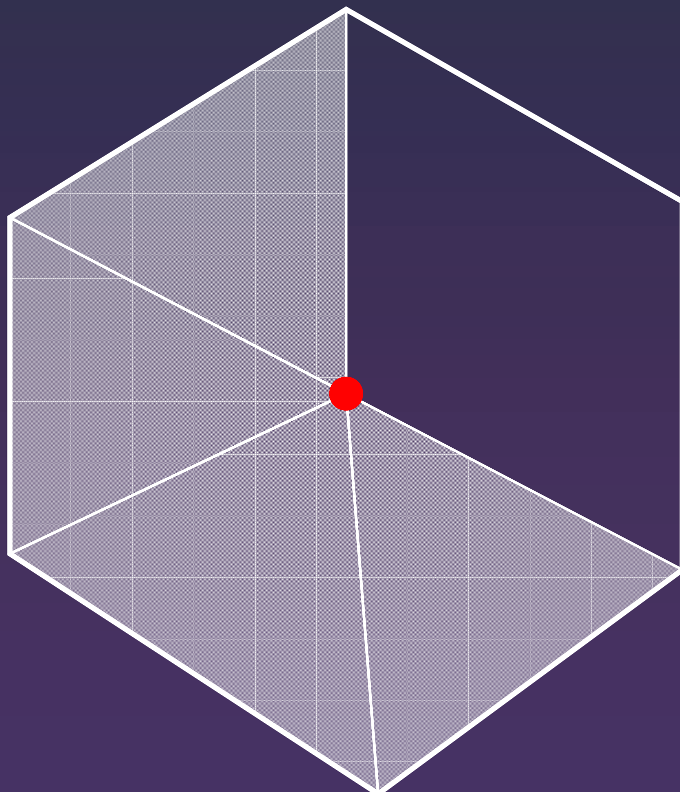
Calculating Area



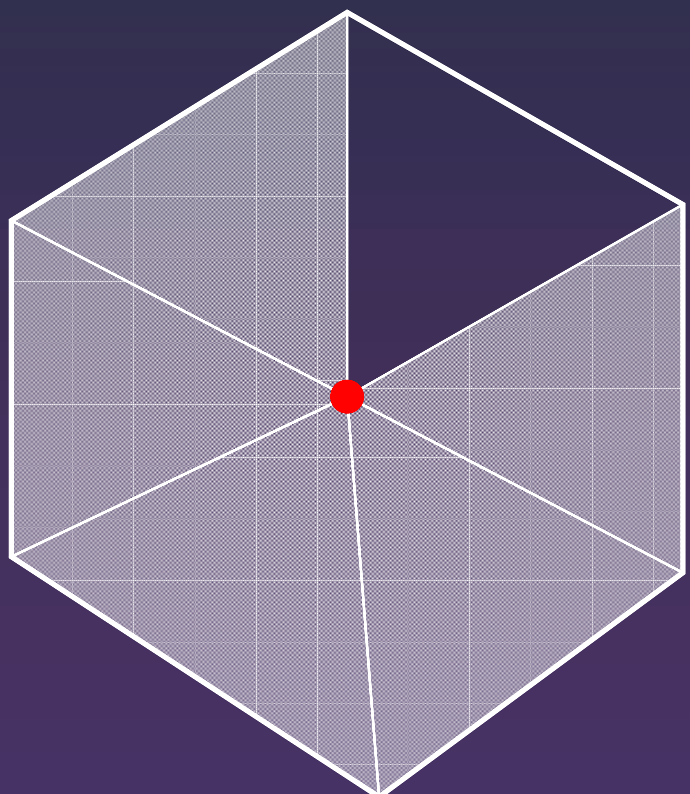
Calculating Area



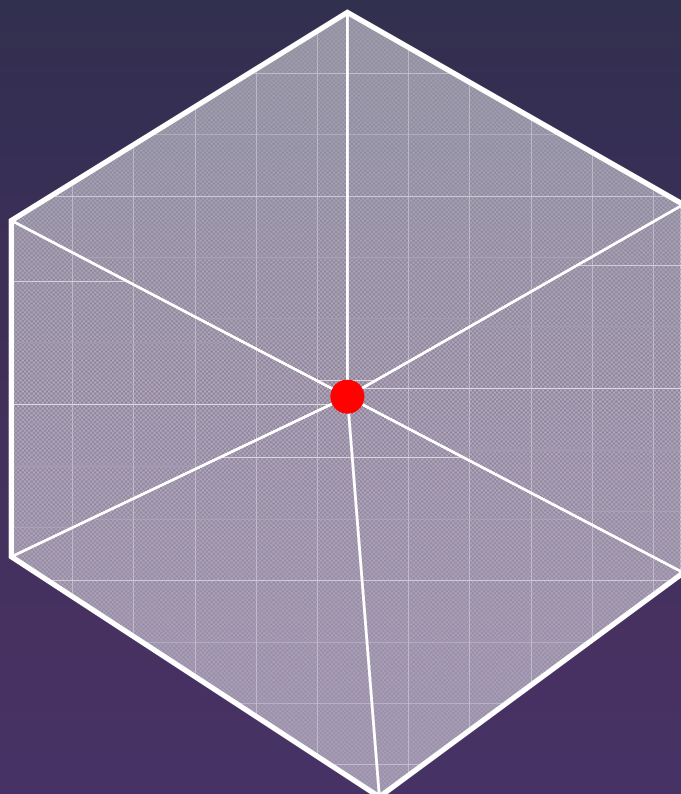
Calculating Area



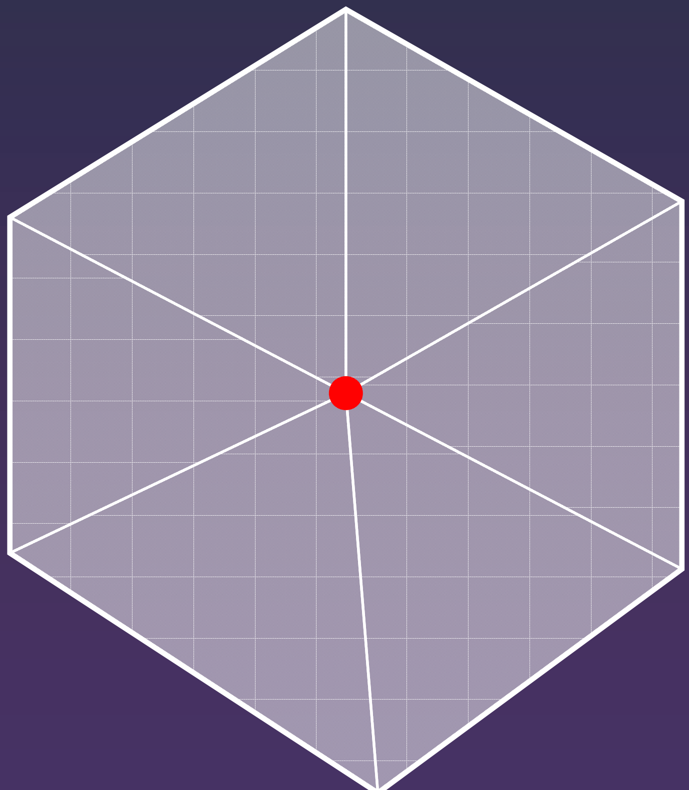
Calculating Area



Calculating Area

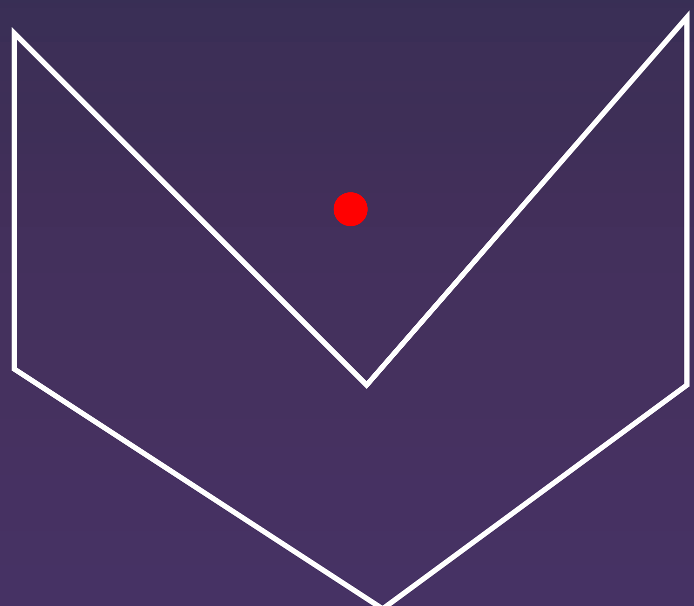


Calculating Area

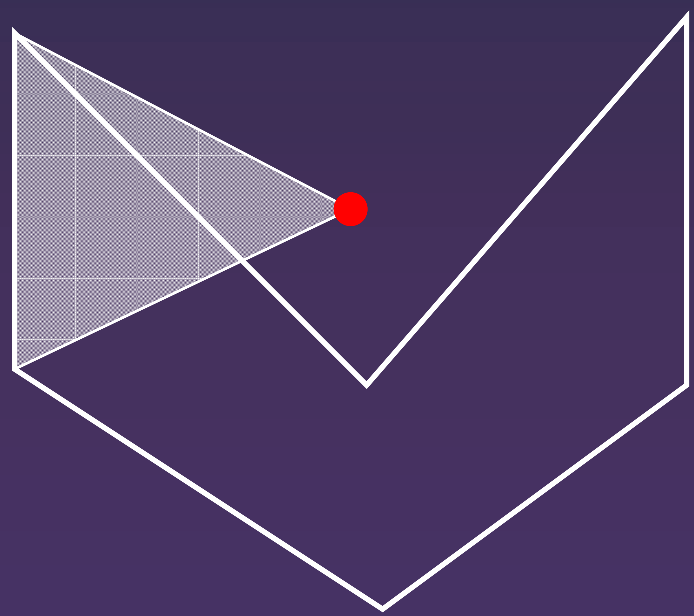


$$\frac{1}{2}\begin{vmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{vmatrix}$$

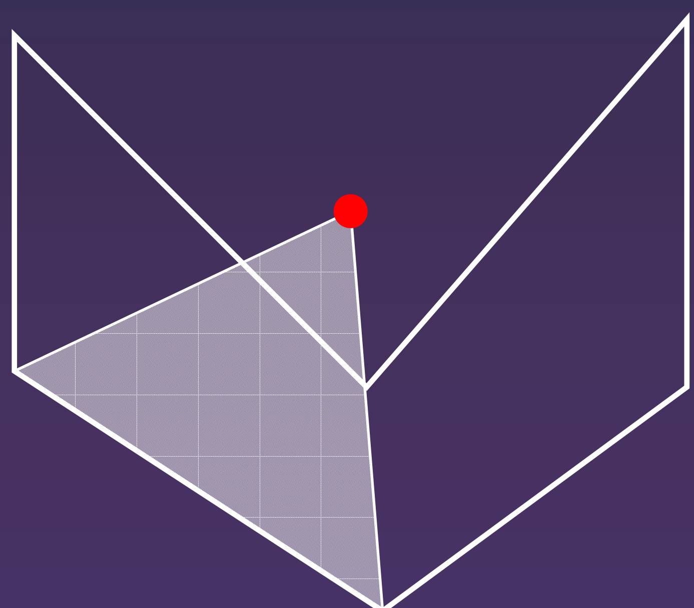
Calculating Area



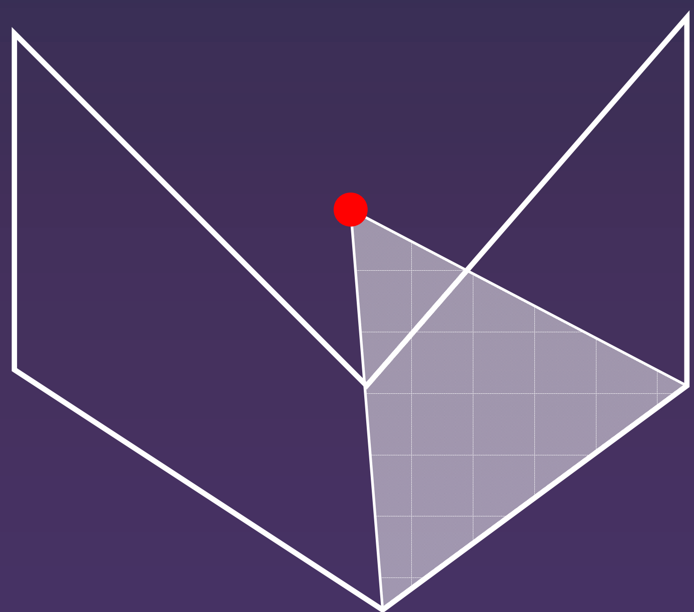
Calculating Area



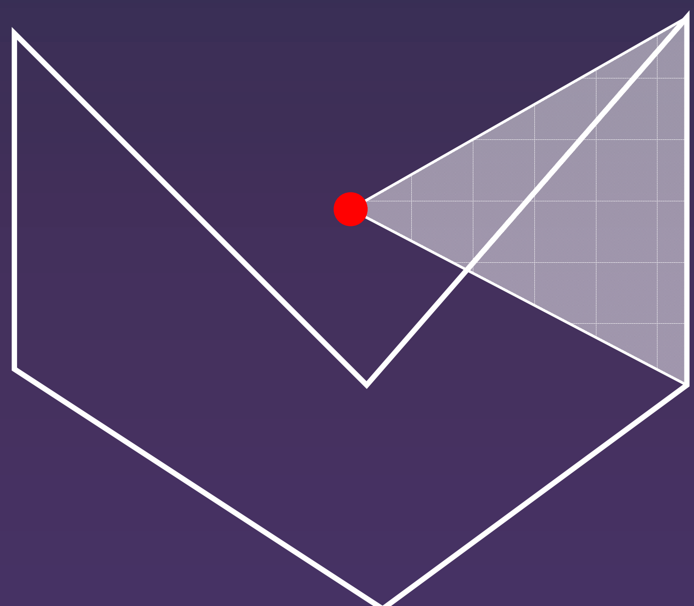
Calculating Area



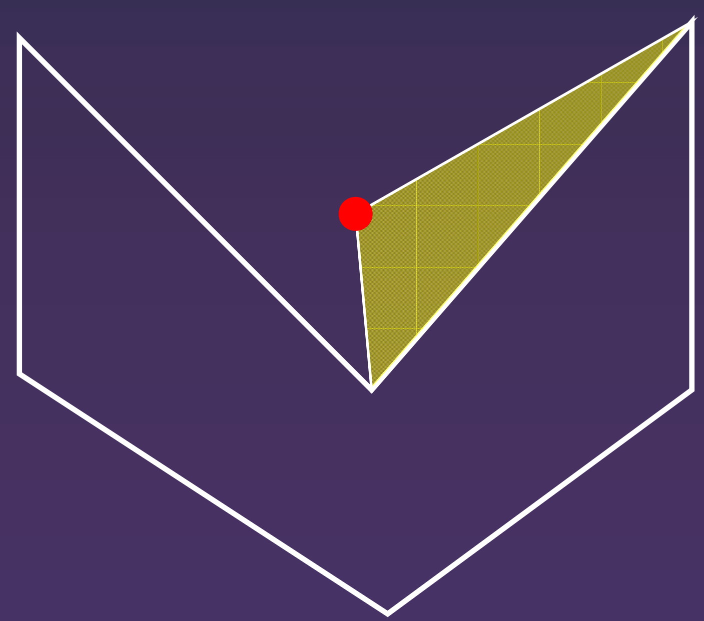
Calculating Area



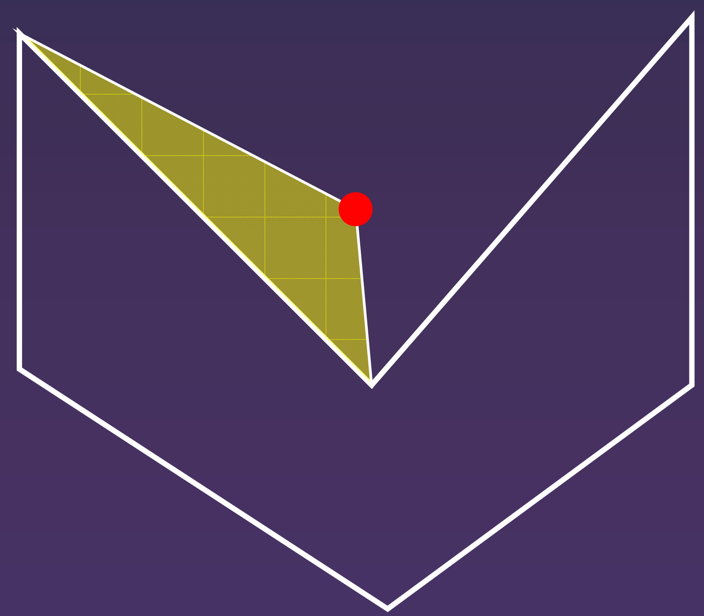
Calculating Area



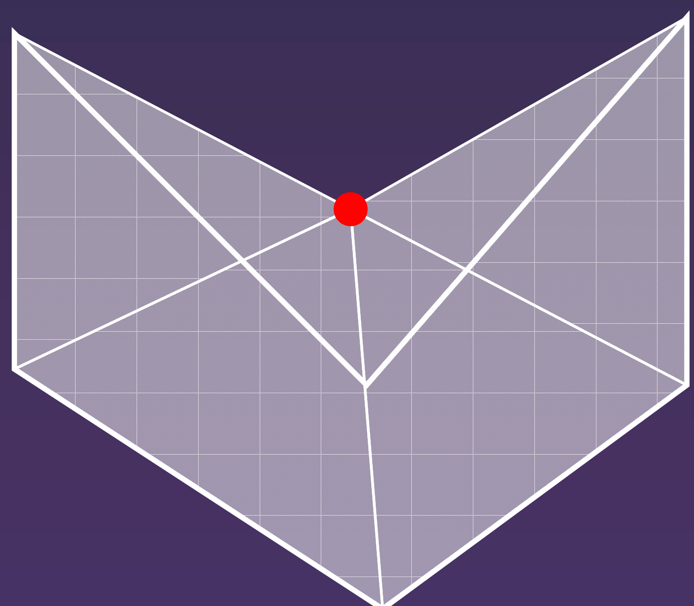
Calculating Area



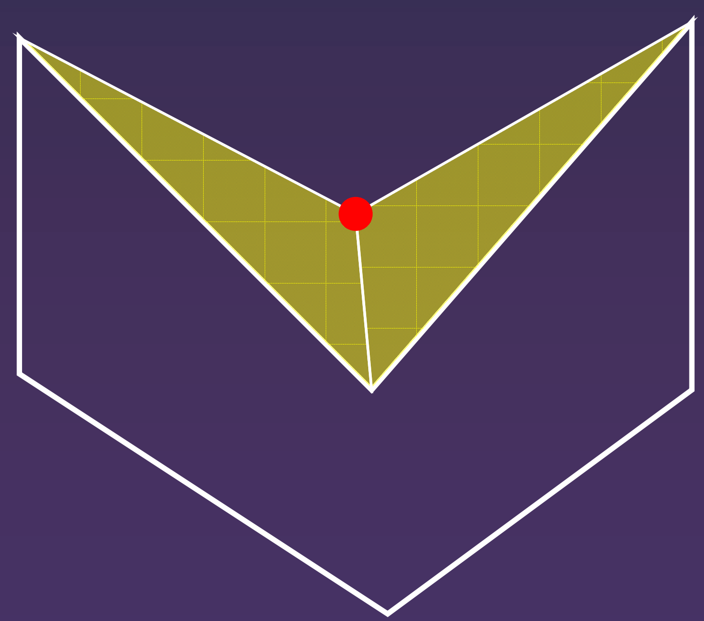
Calculating Area



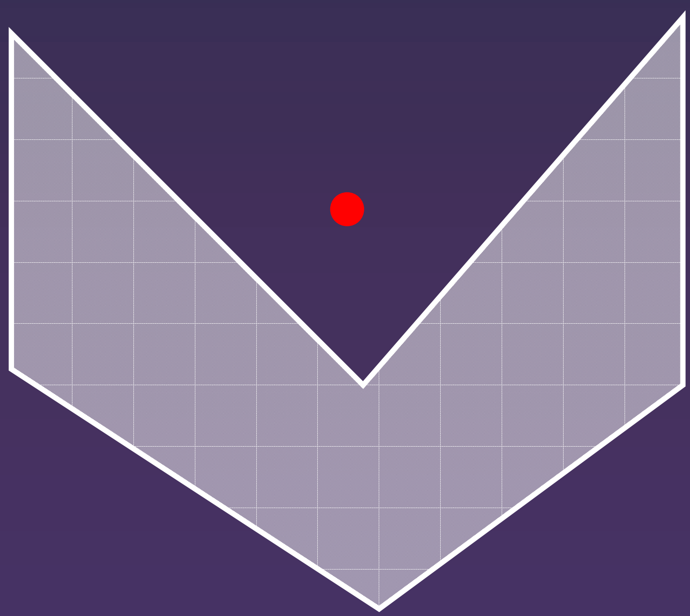
Calculating Area



Calculating Area

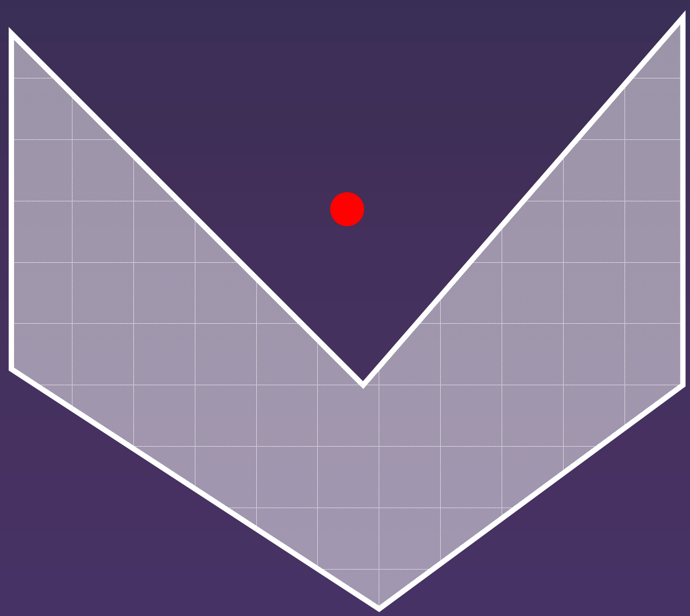


Calculating Area



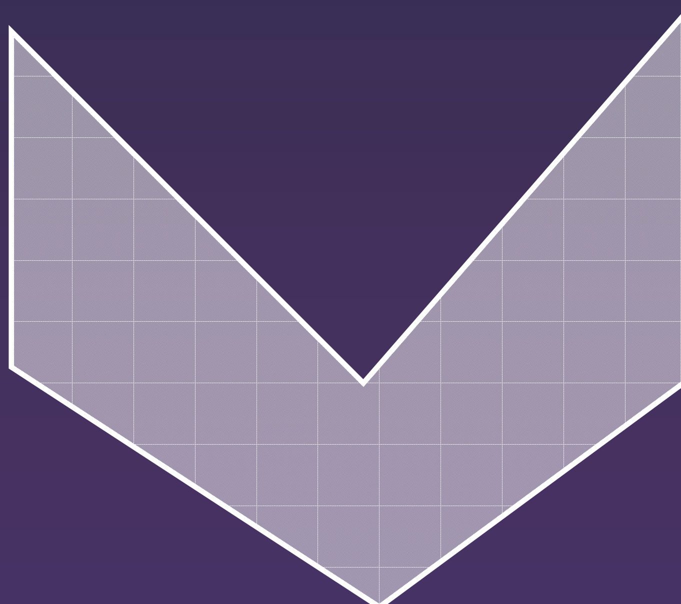
Calculating Area

$$\frac{1}{2}\begin{vmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{vmatrix}$$



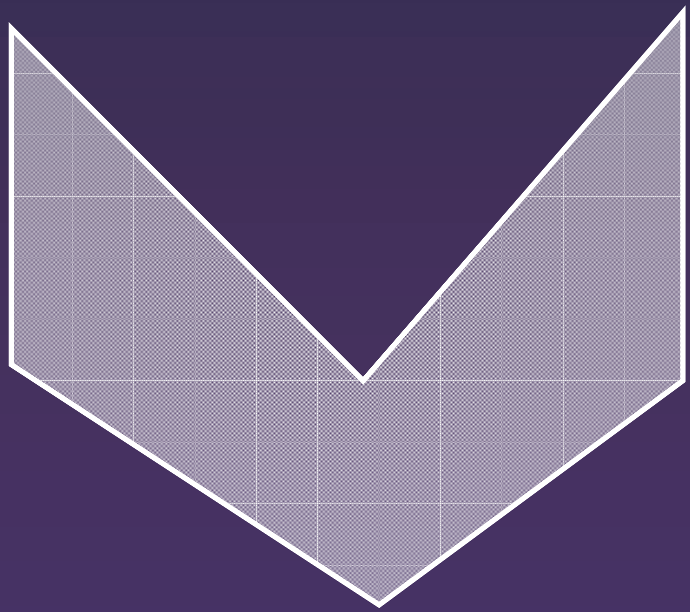
Calculating Area

$$\frac{1}{2}\begin{vmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ 0 & 0 & 1 \end{vmatrix}$$



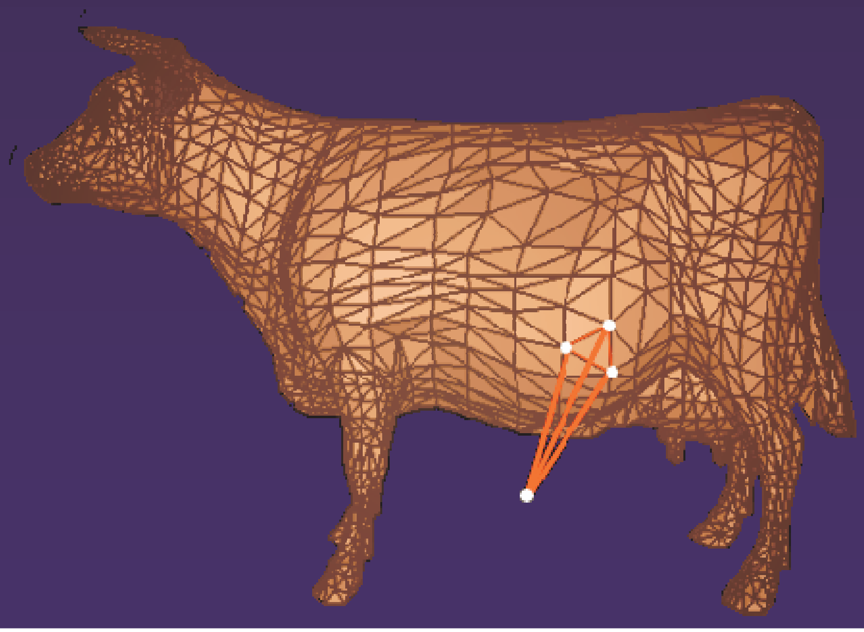
Calculating Area

$$area = \sum_i \frac{1}{2} \begin{vmatrix} x_i & y_i \\ x_{i+1} & y_{i+1} \end{vmatrix}$$



Calculating Volume

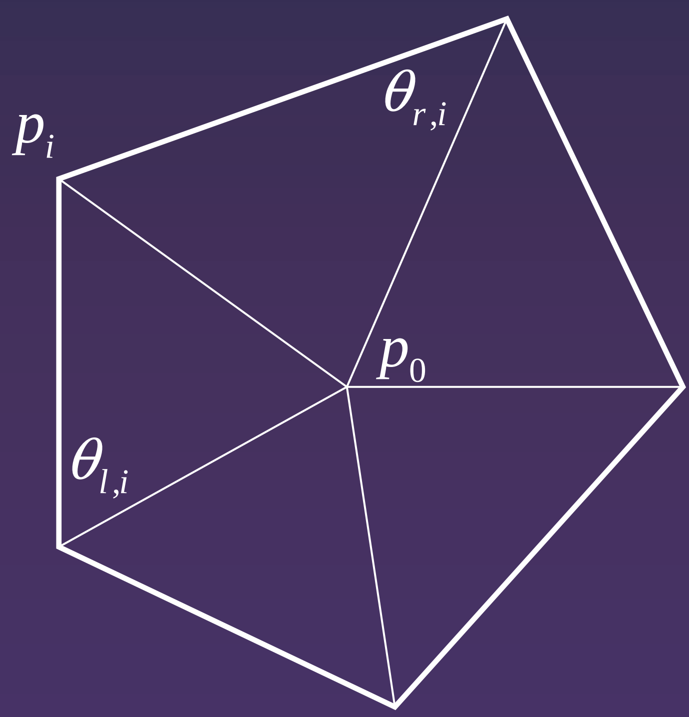
$$volume = \sum_i \frac{1}{6} \begin{vmatrix} x_{i,1} & y_{i,1} & z_{i,1} \\ x_{i,2} & y_{i,2} & z_{i,2} \\ x_{i,3} & y_{i,3} & z_{i,3} \end{vmatrix}$$



Projection-Based Manipulation

Detail Preservation

- Non-linear Laplacian coordinates

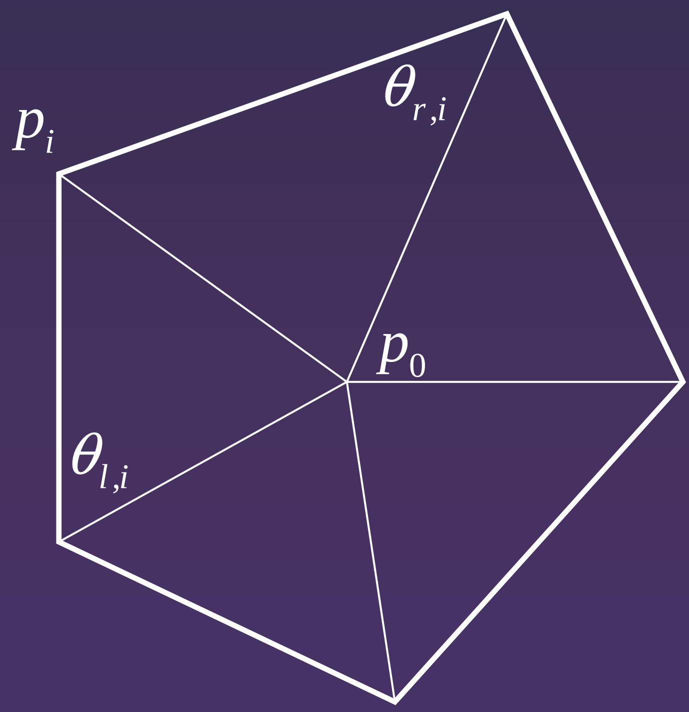


$$-\bar{k}n \propto \sum_i \tfrac{1}{2}(\cot(\theta_{r,i}) + \cot(\theta_{l,i}))(p_i - p_0)$$

Detail Preservation

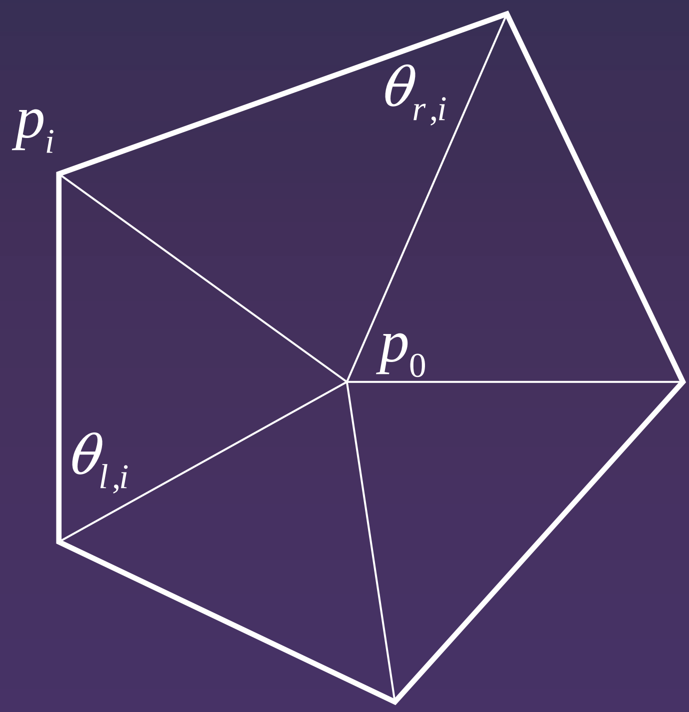
- Non-linear Laplacian coordinates

$$-\alpha\bar{k}n=\sum_i \tfrac{1}{2}(\cot(\theta_{r,i})+\cot(\theta_{l,i}))(p_i-p_0)$$



Detail Preservation

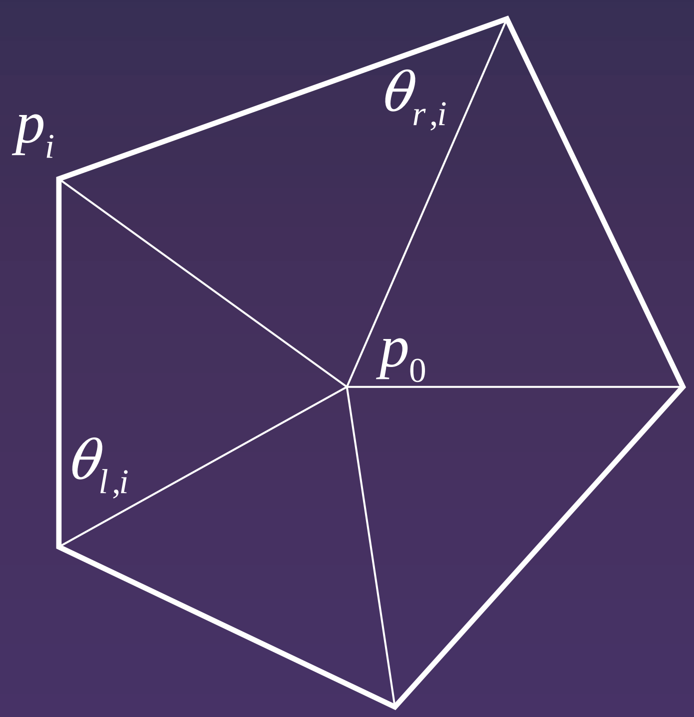
- Non-linear Laplacian coordinates



$$p_0 = \frac{\sum_i \frac{1}{2}(\cot(\theta_{r,i}) + \cot(\theta_{l,i}))p_i + \alpha \bar{k} n}{\sum_i \frac{1}{2}(\cot(\theta_{r,i}) + \cot(\theta_{l,i}))}$$

Detail Preservation

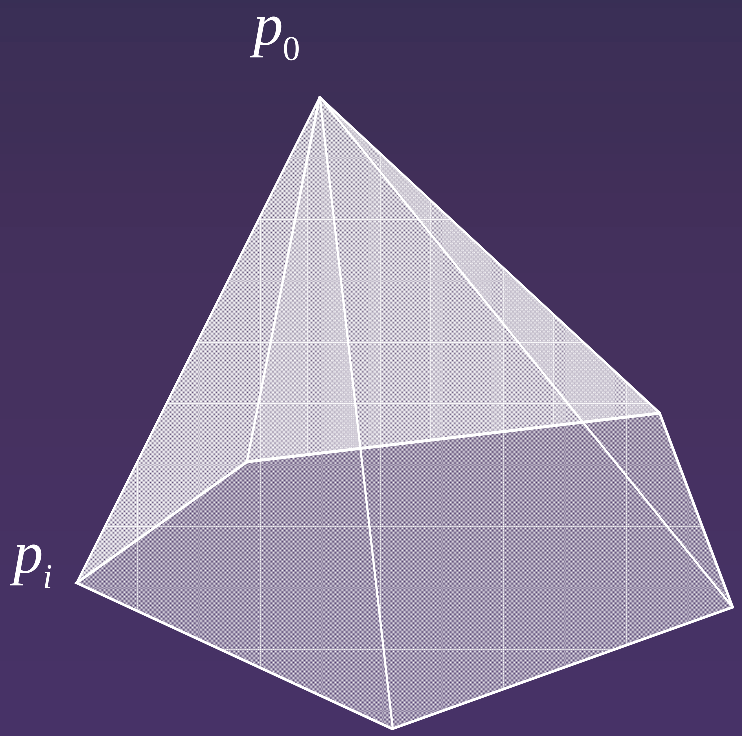
- Non-linear Laplacian coordinates



$$p_0 = \frac{\sum_i (\cot(\theta_{r,i}) + \cot(\theta_{l,i})) p_i}{\sum_i (\cot(\theta_{r,i}) + \cot(\theta_{l,i}))} + v$$

Detail Preservation

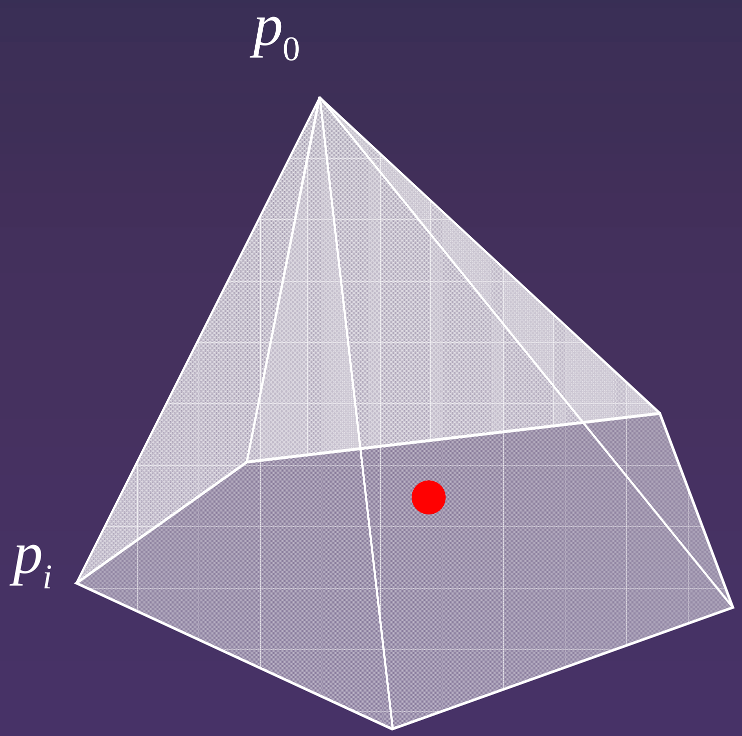
- Non-linear Laplacian coordinates



$$p_0 = \frac{\sum_i (\cot(\theta_{r,i}) + \cot(\theta_{l,i}))p_i}{\sum_i (\cot(\theta_{r,i}) + \cot(\theta_{l,i}))} + v$$

Detail Preservation

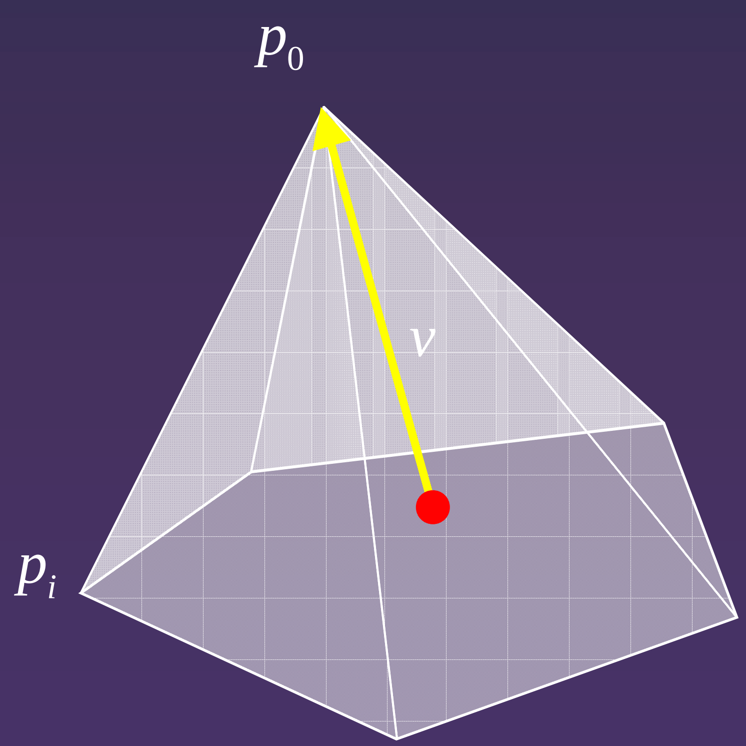
- Non-linear Laplacian coordinates



$$p_0 = \frac{\sum_i (\cot(\theta_{r,i}) + \cot(\theta_{l,i})) p_i}{\sum_i (\cot(\theta_{r,i}) + \cot(\theta_{l,i}))} + v$$

Detail Preservation

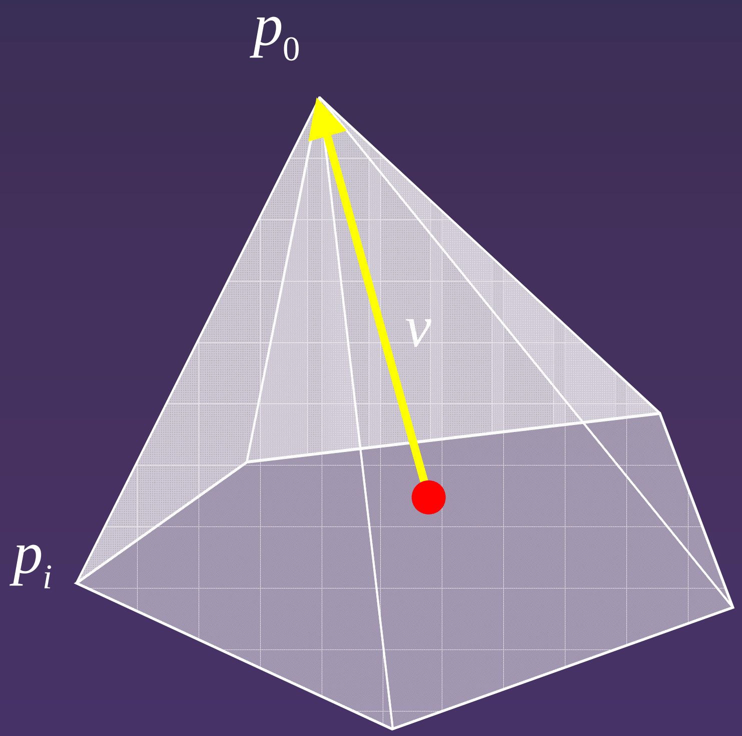
- Non-linear Laplacian coordinates



$$p_0 = \frac{\sum_i (\cot(\theta_{r,i}) + \cot(\theta_{l,i}))p_i}{\sum_i (\cot(\theta_{r,i}) + \cot(\theta_{l,i}))} + v$$

Detail Preservation

- Non-linear Laplacian coordinates

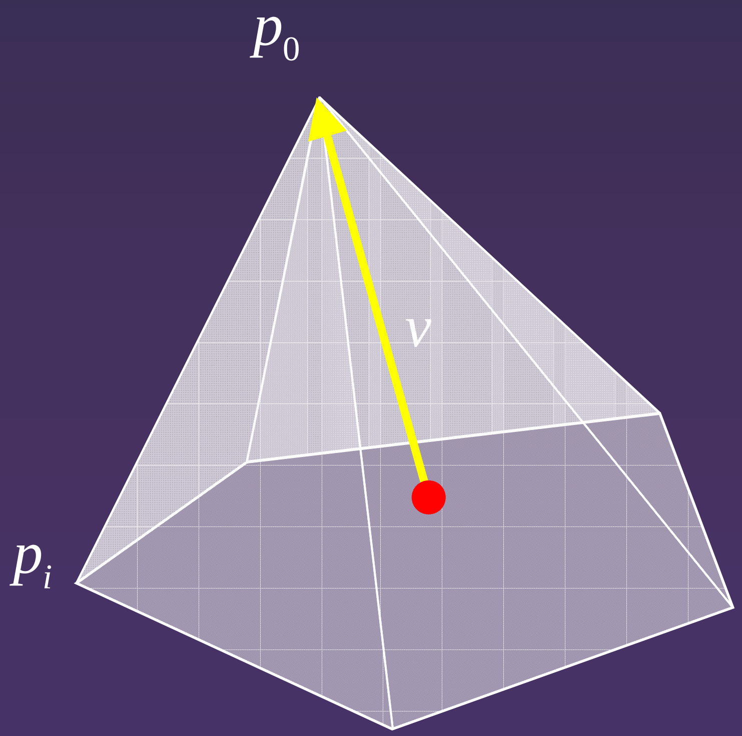


$$\sum_i \alpha_i (p_i - p_0) \times (p_{i+1} - p_0) = v$$

Find α_i through pseudoinverse

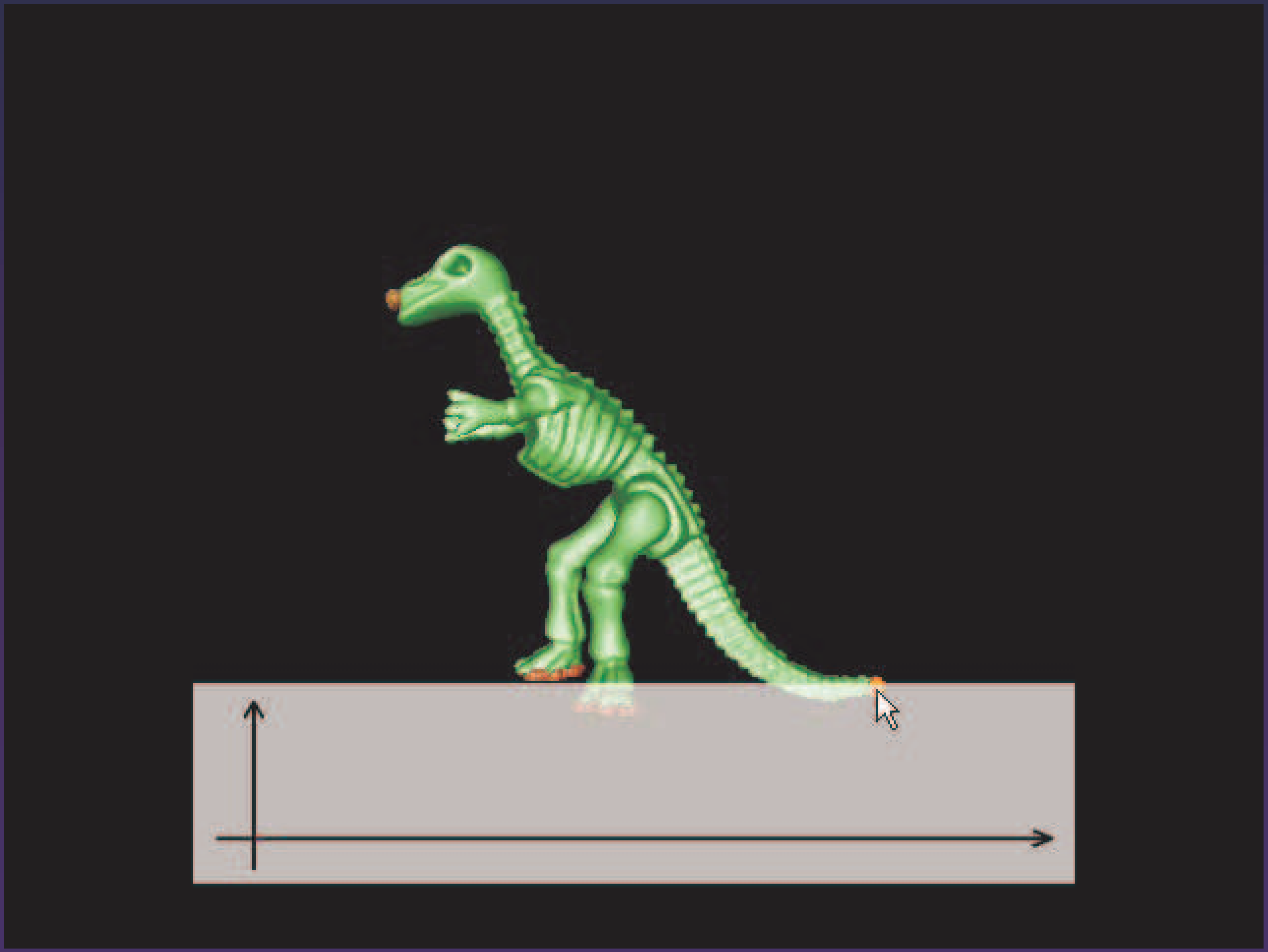
Detail Preservation

- Non-linear Laplacian coordinates



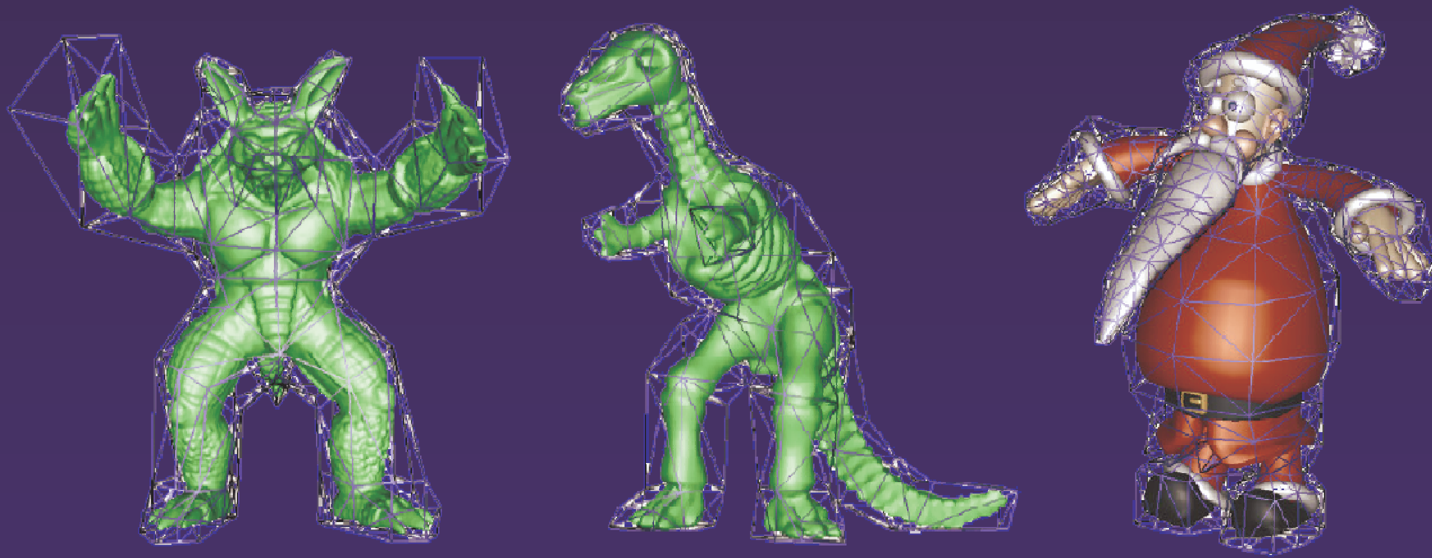
$$\hat{p}_0 = \sum_i \beta_i \hat{p}_i + \frac{\sum_i \alpha_i (\hat{p}_i - \hat{p}_0) \times (\hat{p}_{i+1} - \hat{p}_0)}{\left| \sum_i \alpha (\hat{p}_i - \hat{p}_0) \times (\hat{p}_{i+1} - \hat{p}_0) \right|} |v|$$

Non-linear Minimization

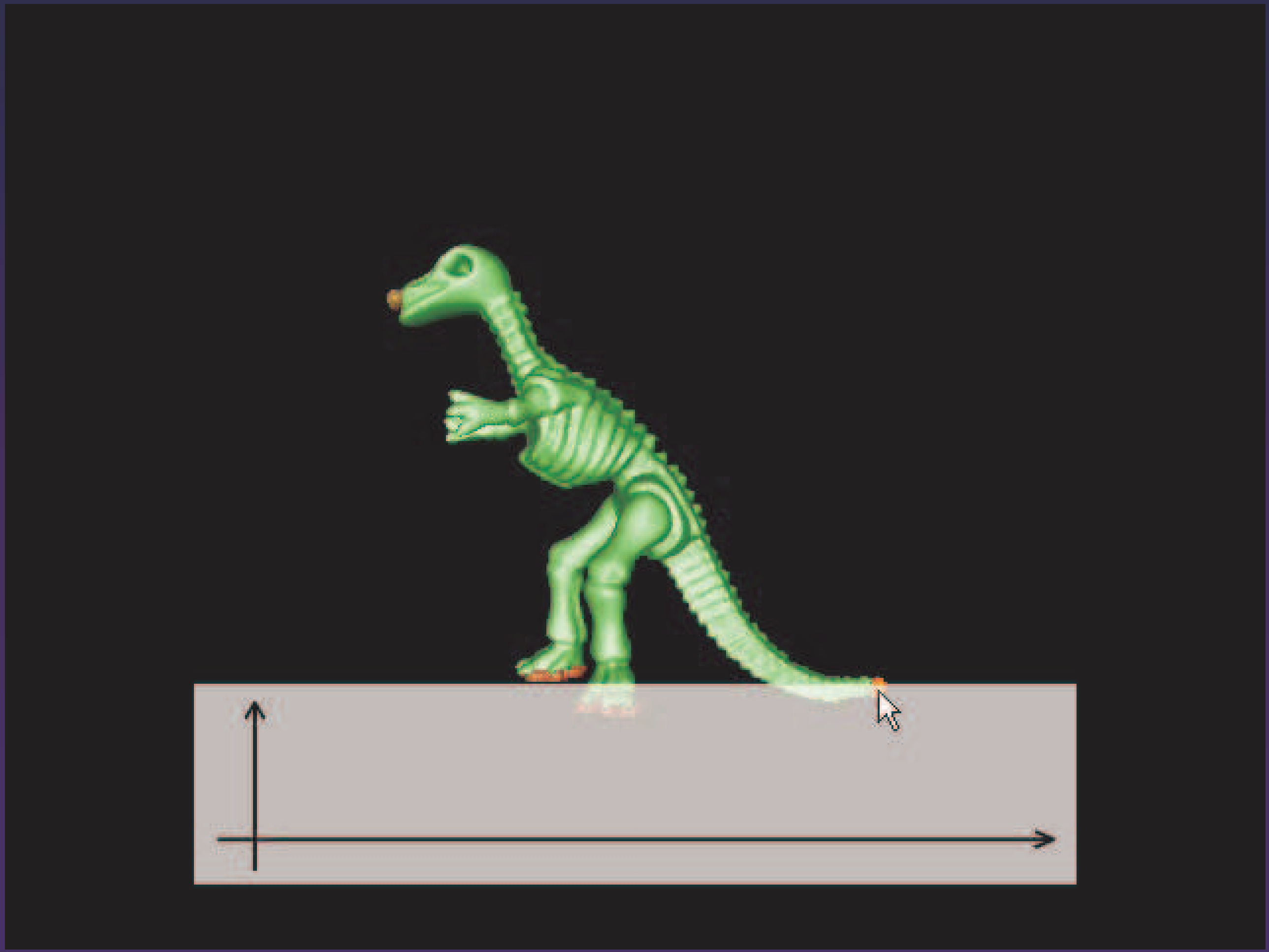


Subspace Solver

- Construct a low-res approximation of mesh
- Express constraints in terms of MV coordinates

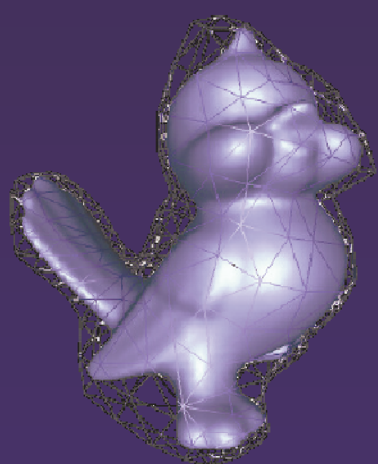


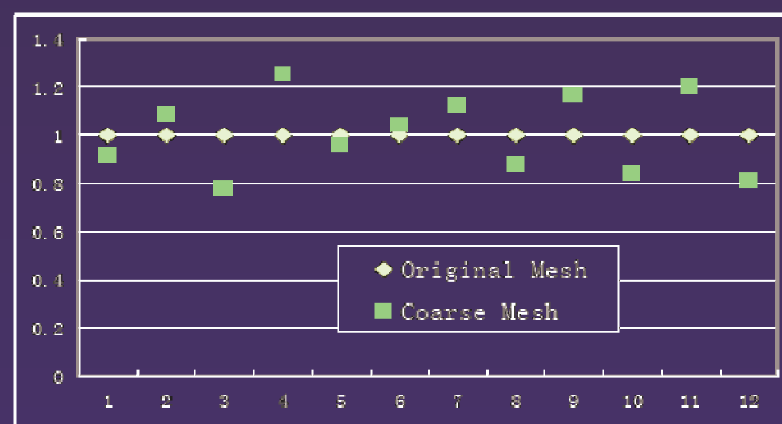
Subspace Solver



Subspace Solver

- Constraints are on the high-res mesh… NOT the low-res mesh
- A variant of a multi-grid solver
- Speeds up convergence and helps stability





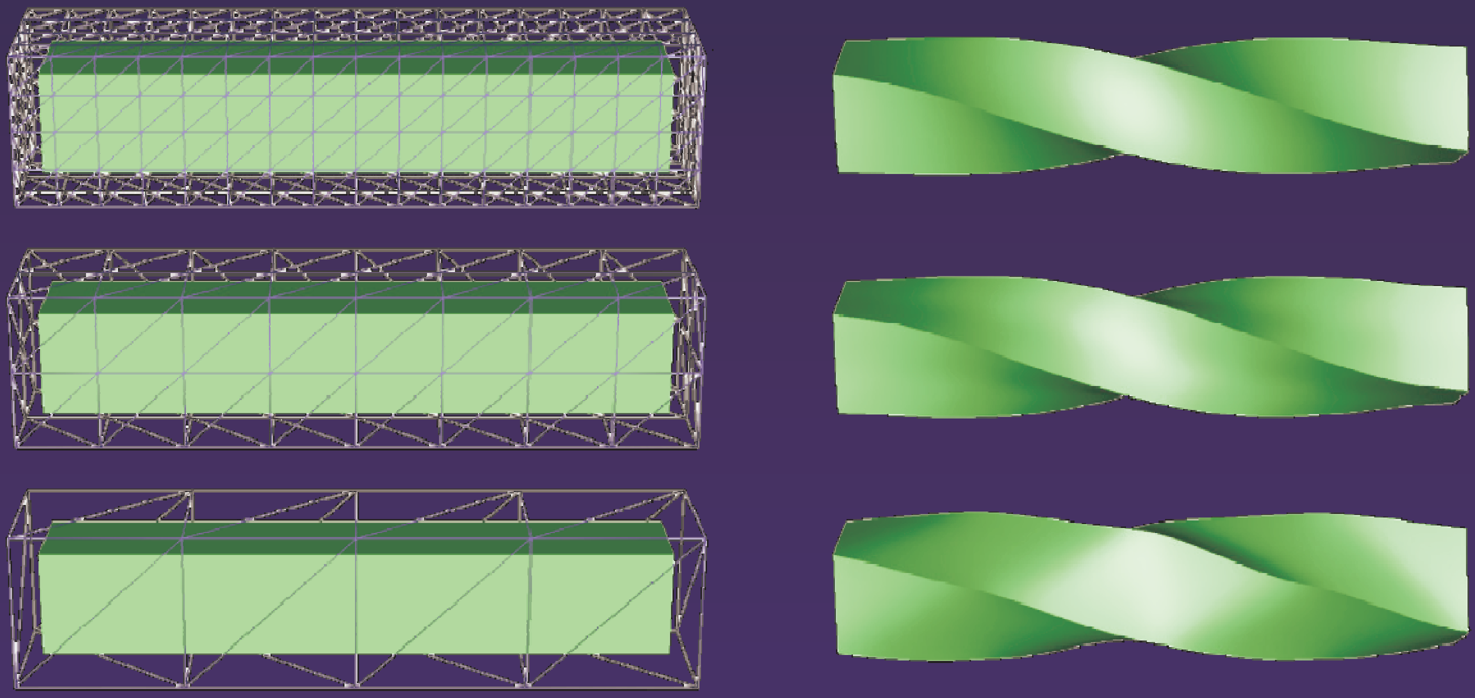
Subspace Solver

- Constraints are on the high-res mesh… NOT the low-res mesh
- A variant of a multi-grid solver
- Speeds up convergence and helps stability

model	# vertices (original mesh)	# vertices (coarse mesh)	full space	subspace
Armadillo	30,002	220	2.8	9.1
Horse	14,285	427	6.9	8.2
Tweety	10,240	286	12	23.8
Dinosaur	10,002	159	9.5	34.5
DNA	19,184	194	NA*	16.7
Santa	25,777	448	NA*	5.3

Subspace Solver

- Resolution of low-res mesh affects quality of deformation



Results

Conclusions

- Most useful deformation constraints involve non-linear functions
- MV coordinates accelerate solving
- Don't be afraid of non-linear optimization!!!