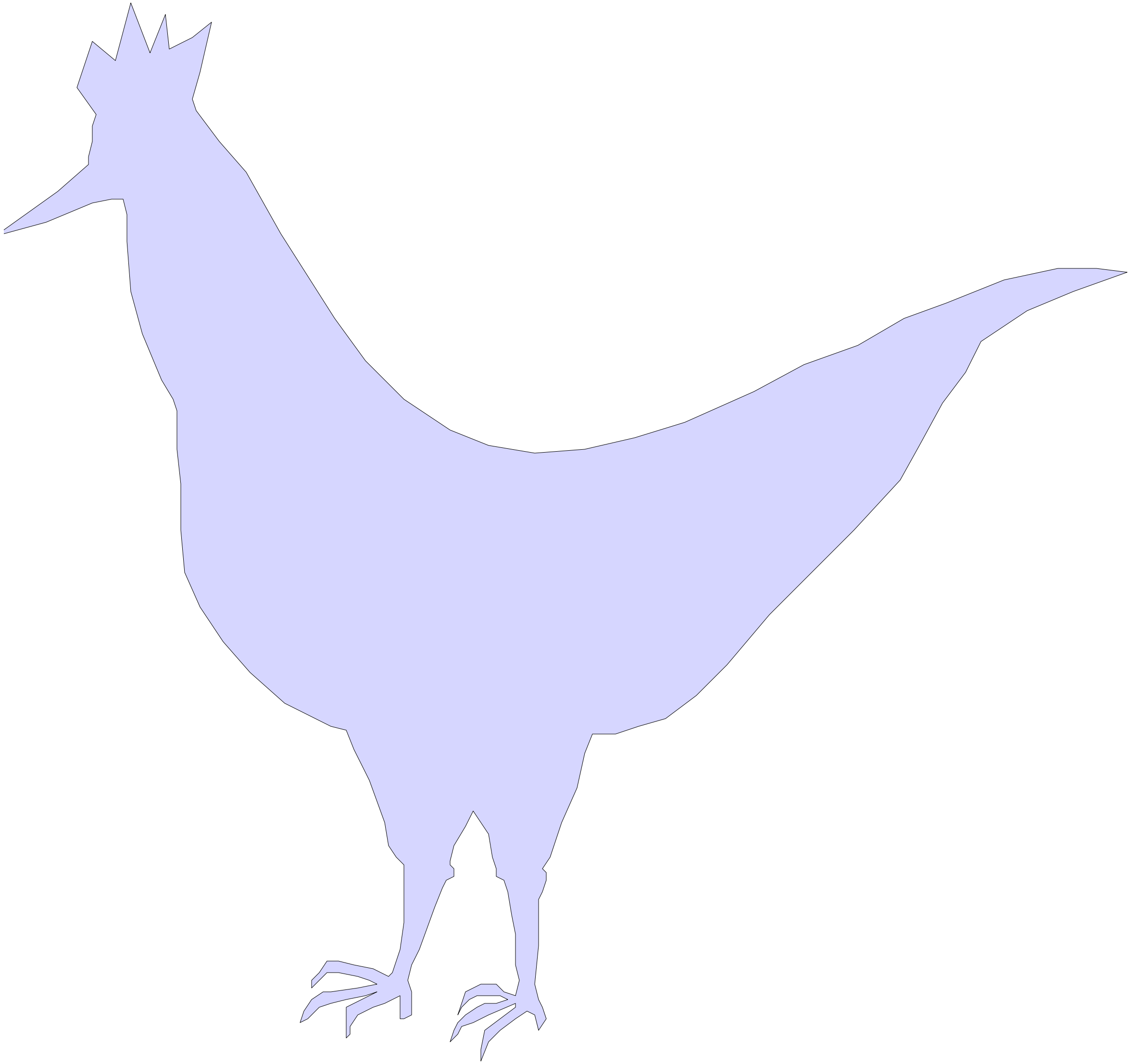
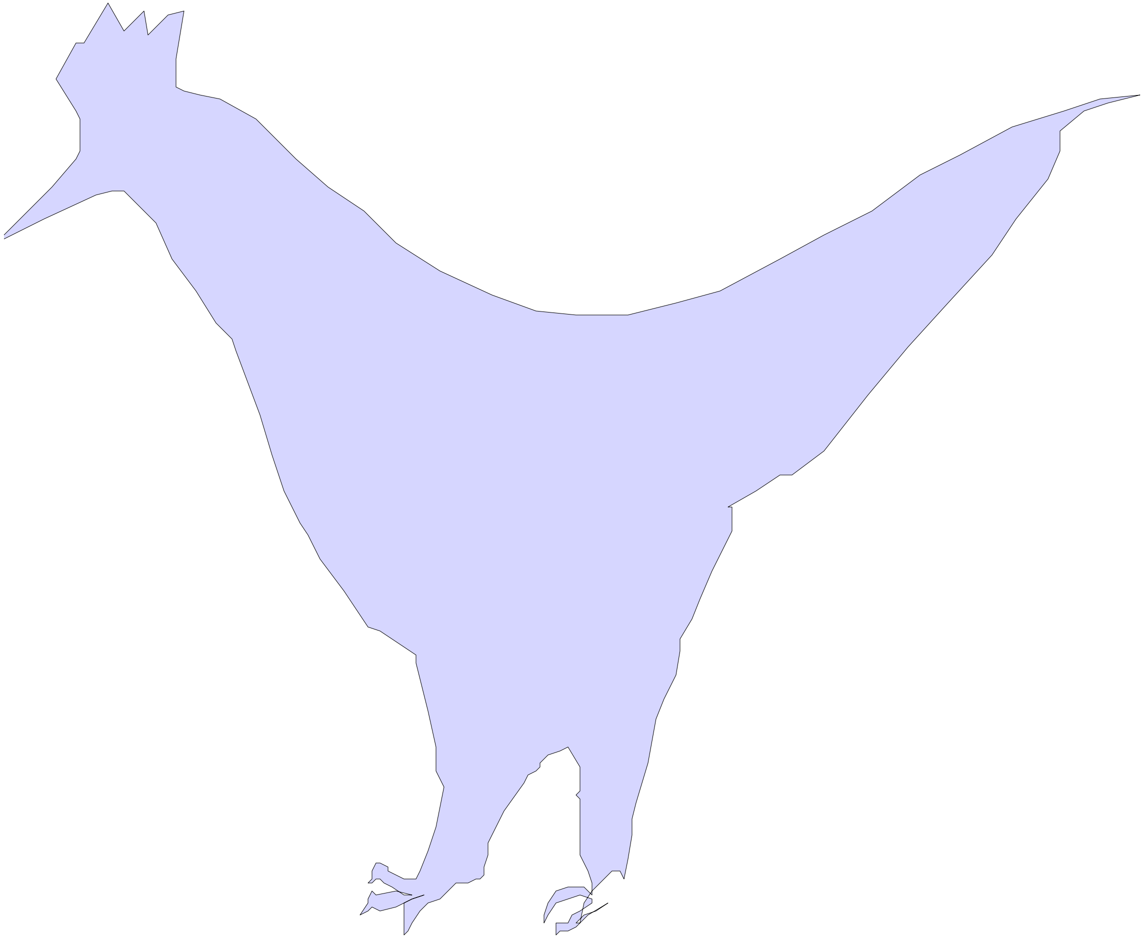
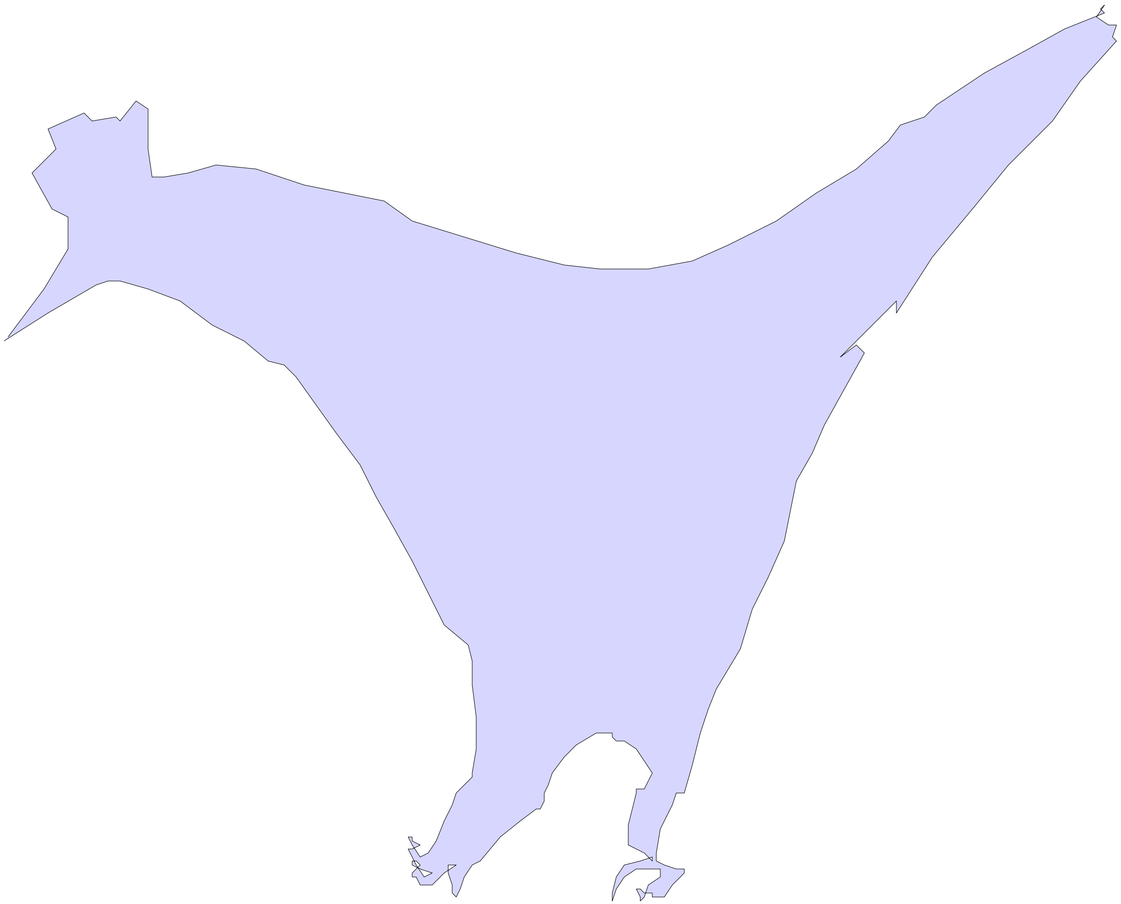
Animation

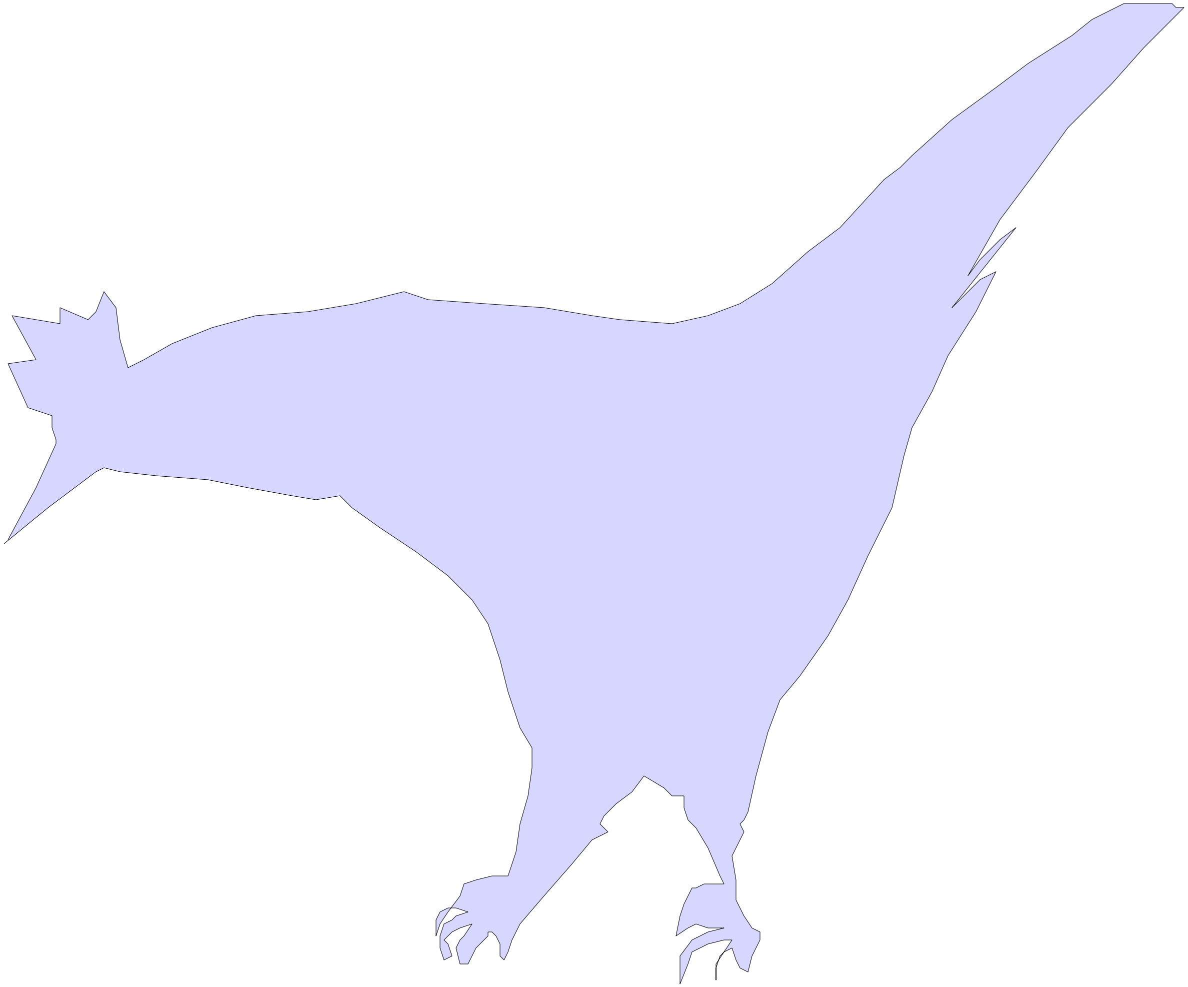
CS418 Computer Graphics

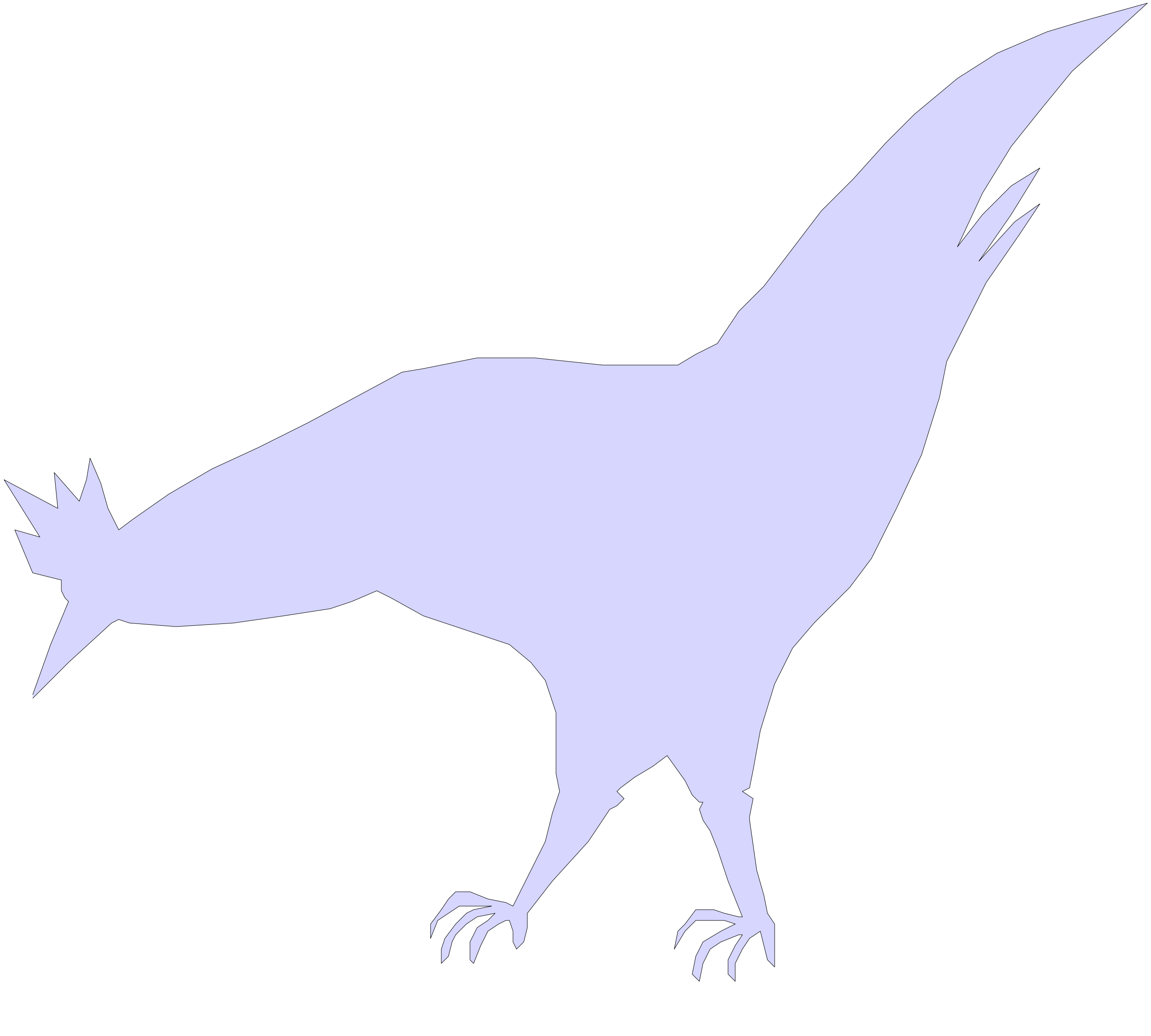
John C. Hart





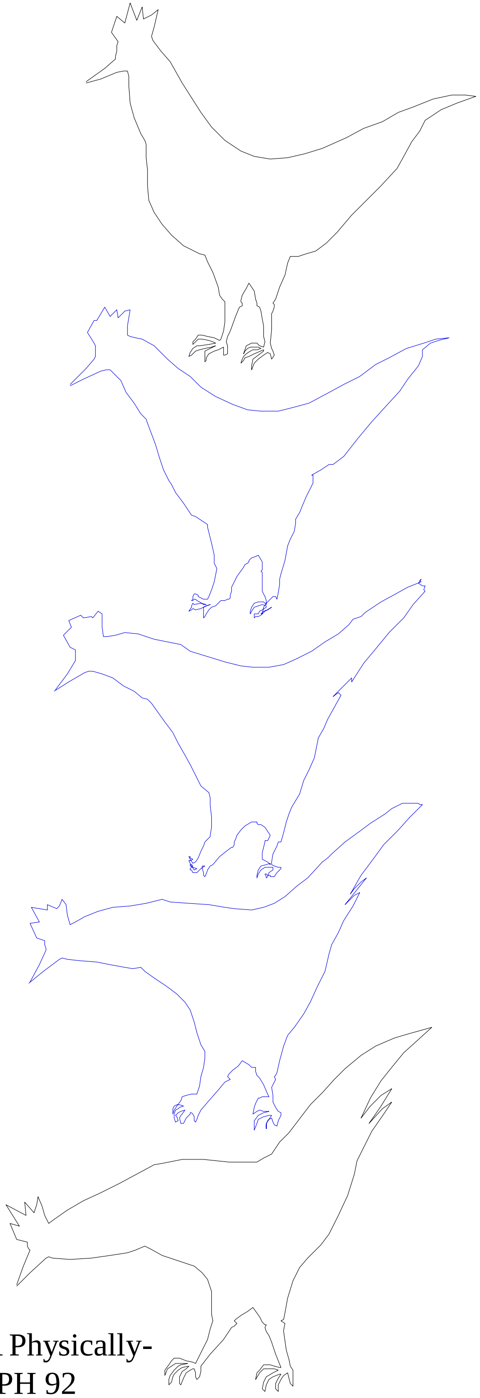






Keyframe Animation

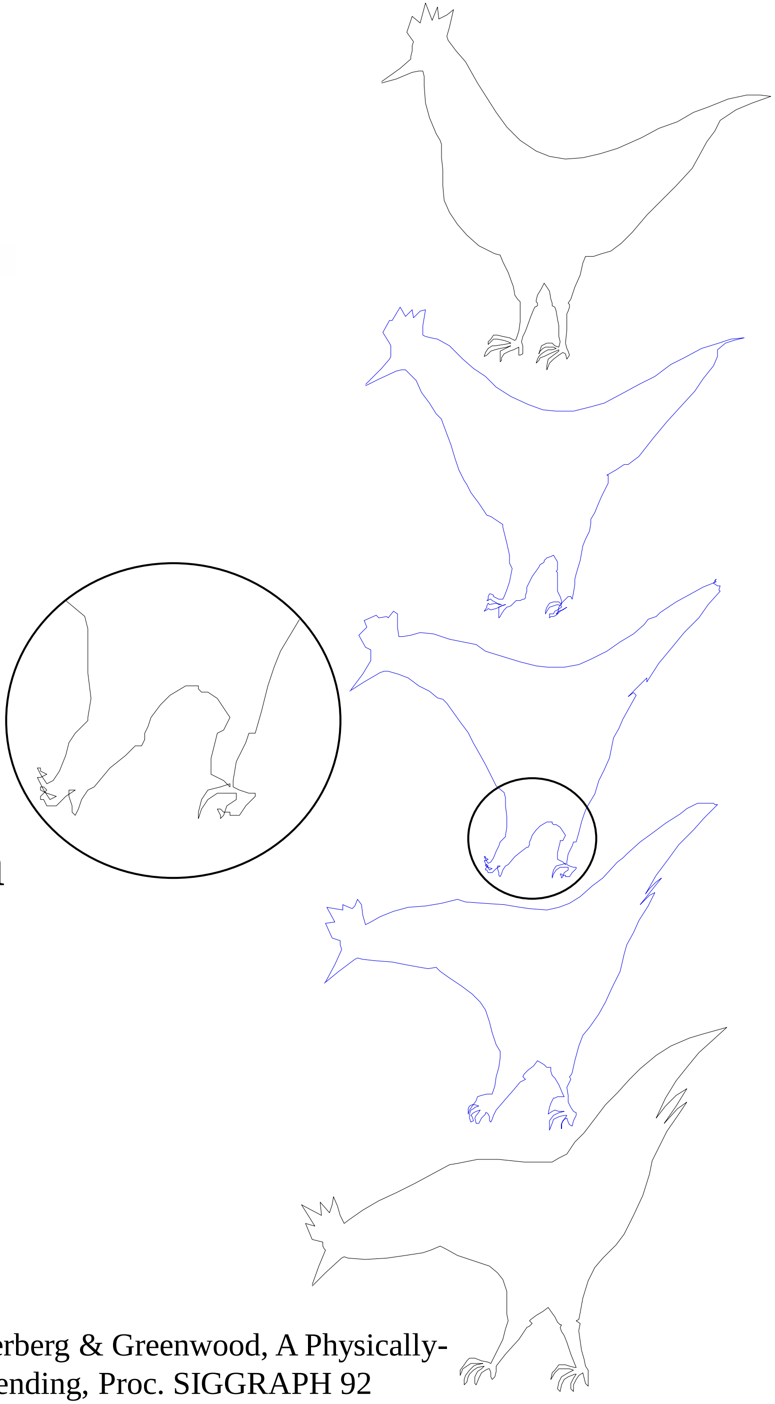
- Set target positions for vertices at "key" frames in animations
- Linearly interpolate vertex positions between targets at intervening frames
- Lots can go wrong (like the feet)
- Can be fixed by adding key frames
- Piecewise linear approach to animation
- Need same number and configuration of vertices at key frames for intervening frames to make sense
- Often need to find correspondences between two collections of vertices



A motivating example from: Sederberg & Greenwood, A Physically-Based Approach to 2-D Shape Blending, Proc. SIGGRAPH 92

Keyframe Animation

- Set target positions for vertices at "key" frames in animations
- Linearly interpolate vertex positions between targets at intervening frames
- Lots can go wrong (like the feet)
- Can be fixed by adding key frames
- Piecewise linear approach to animation
- Need same number and configuration of vertices at key frames for intervening frames to make sense
- Often need to find correspondences between two collections of vertices



A motivating example from: Sederberg & Greenwood, A Physically-Based Approach to 2-D Shape Blending, Proc. SIGGRAPH 92

Polar Decomposition

- Linear affine interpolation of transformation matrices does not accommodate rotation
- Let *M* be the upper-left 3x3 submatrix of a 4x4 homogeneous transformation matrix
- Decompose: $M = QS$
  - *Q*: non-linearly varying part (rotation)
  - *S*: linearly varying part (scale, shear)
- Initialize $Q = M$
- Replace $Q = ½ (Q + Q^{-T})$ until it convergence to a 3x3 rotation matrix ($Q^T = Q^{-1}$)
- Then *Q* contains the rotation part of *M*
- And $S = Q^T M$ contains the scaling part
- Interpolate *S* linearly per-element
- Interpolate *Q* using quaternions