Morphing Intro

Morphing in film: Willow

https://www.youtube.com/watch?v=IKzbsDG58pc

500 years of female portraits in western art

http://youtu.be/nUDIoN-_Hxs

Morphing overview

- A morph from one image to another involves
  - Warping each image to an intermediate shape
  - Cross-dissolving (blending) the two warped image

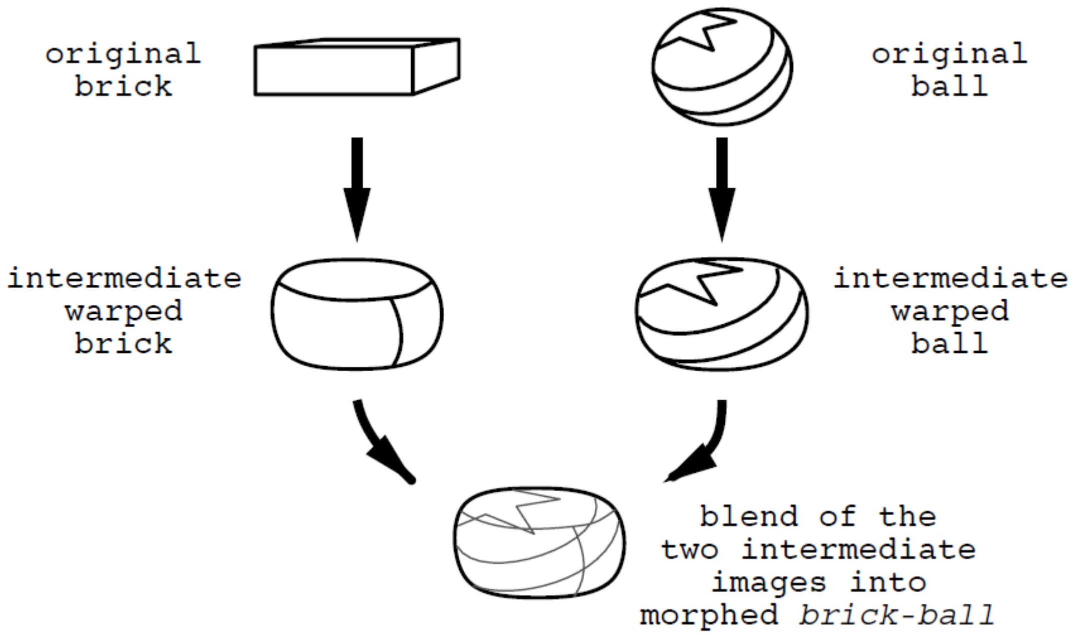
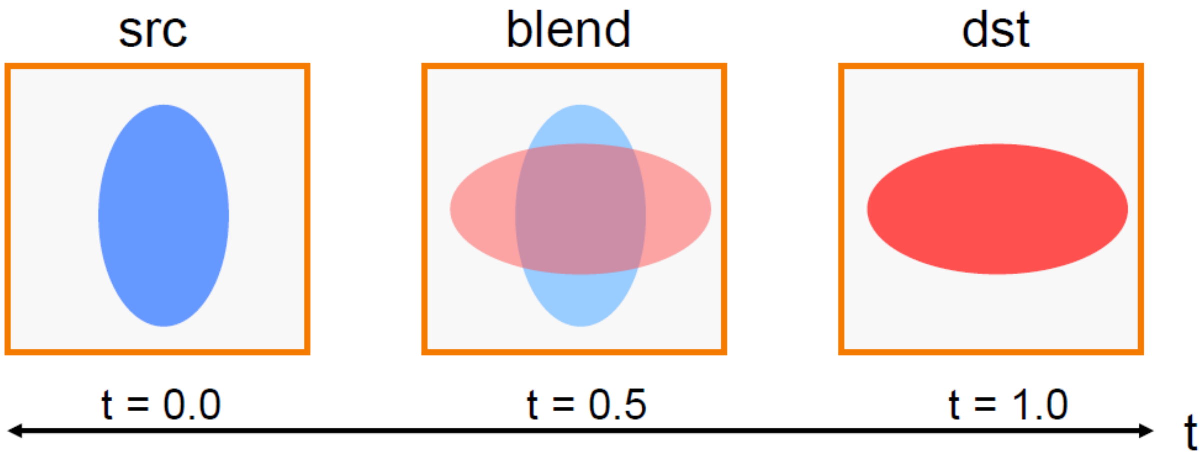


Figure 13.1: A Step in a Morph of a Brick into a Ball

Cross-Dissolving

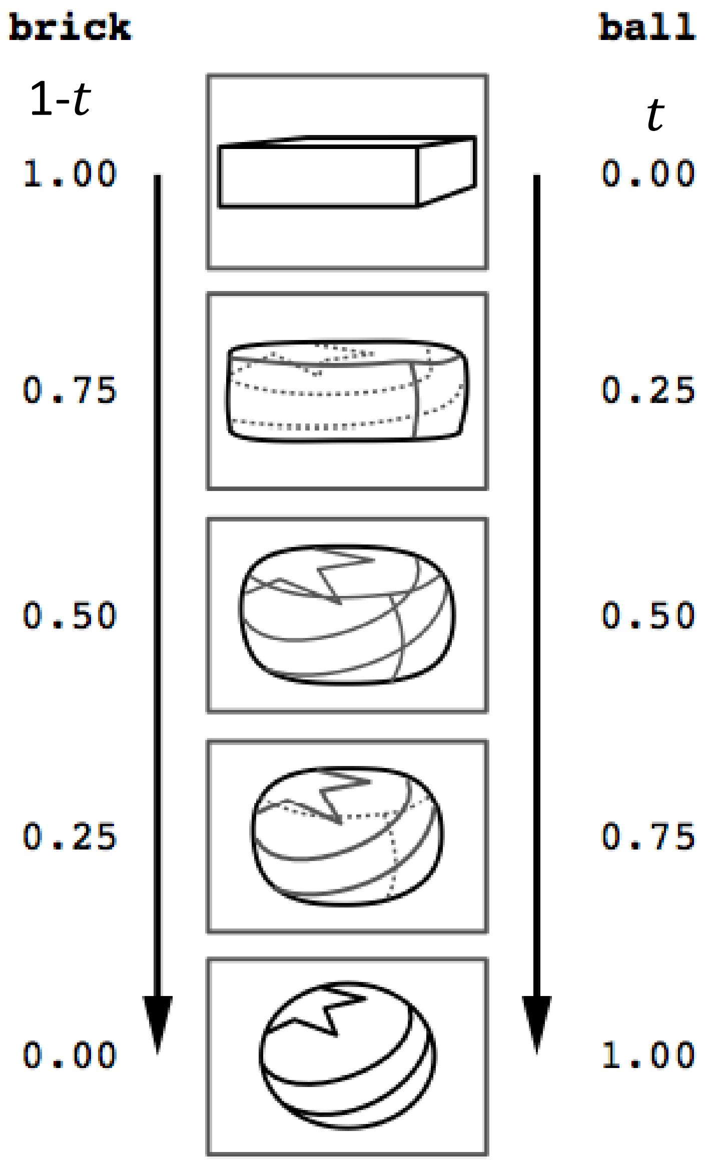
Blend images with over operator

$$\text{blend}(i,j) = (1\text{-}t)\ \text{src}(i,j) + t\ \text{dst}(i,j) \quad (0 \le t \le 1)$$



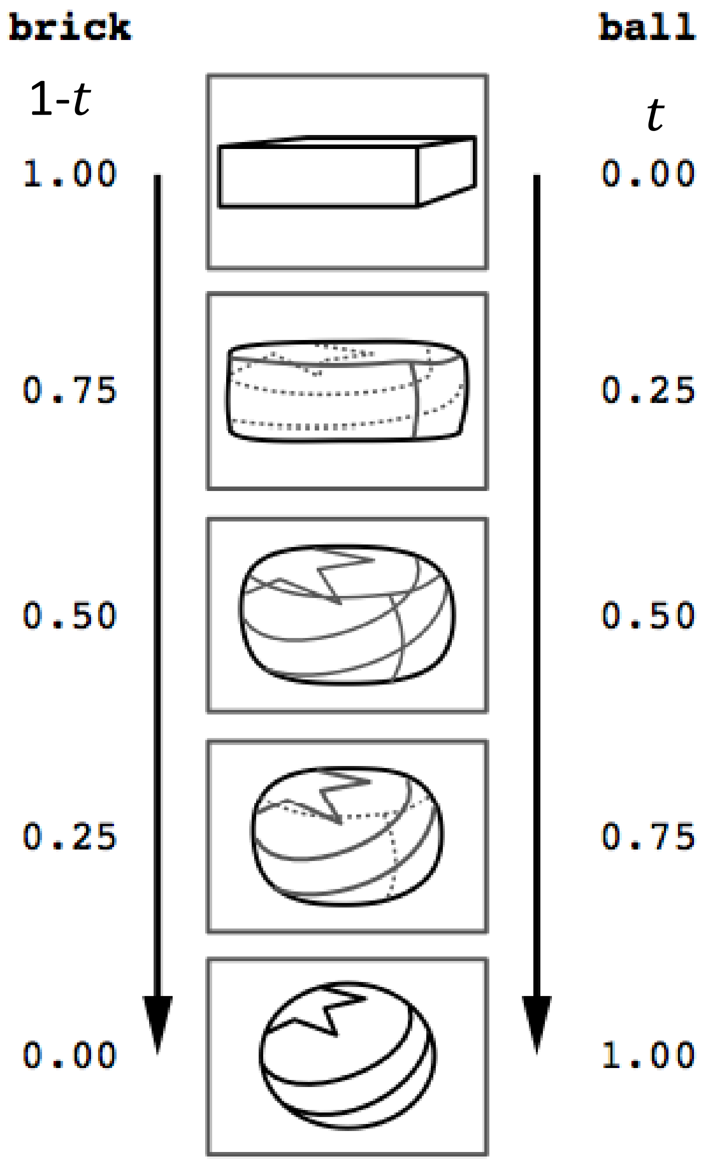
Steps in a morph

- Morphing includes a series of warping/blending steps
- Typically use interpolation over some "time", t, to get them!



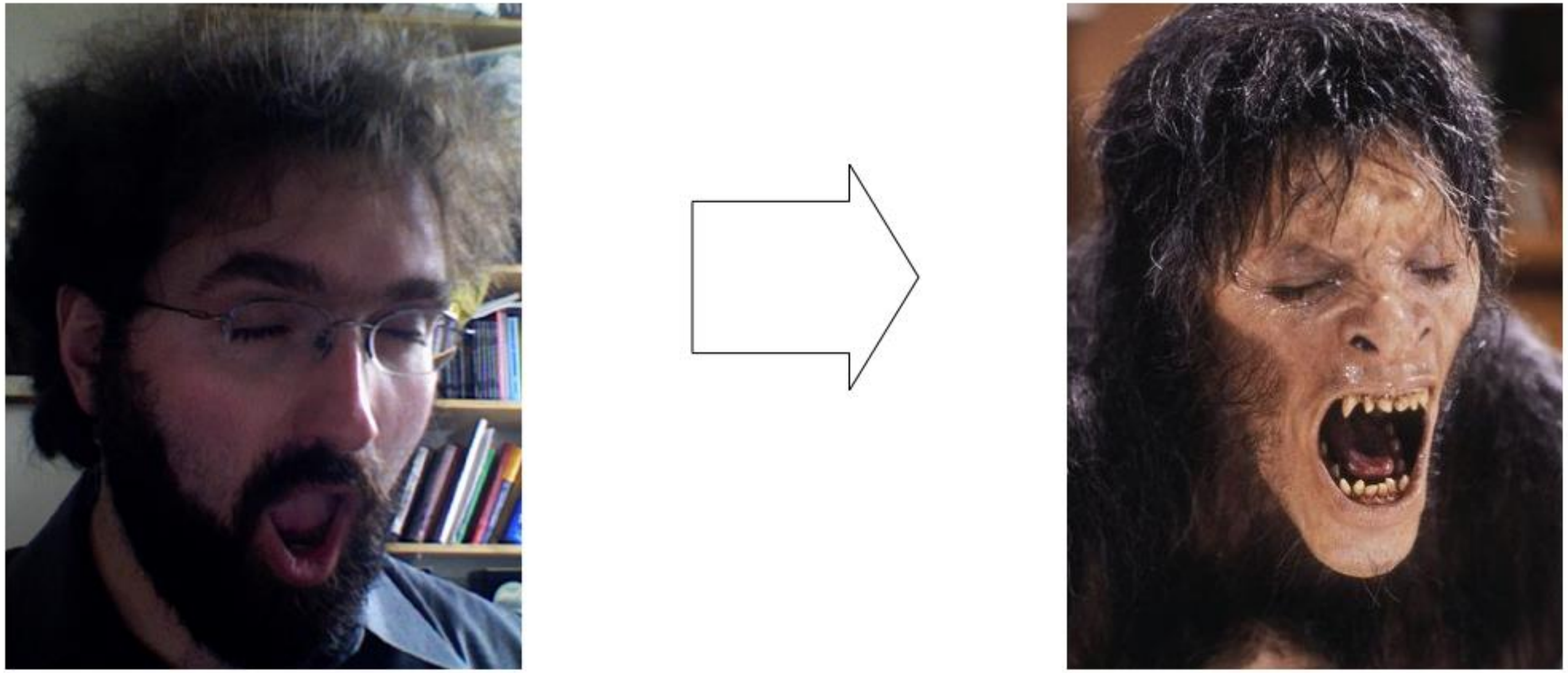
Steps in a morph

- Morphing includes a series of warping/blending steps
- Typically use interpolation over some “time”, t, to get them!



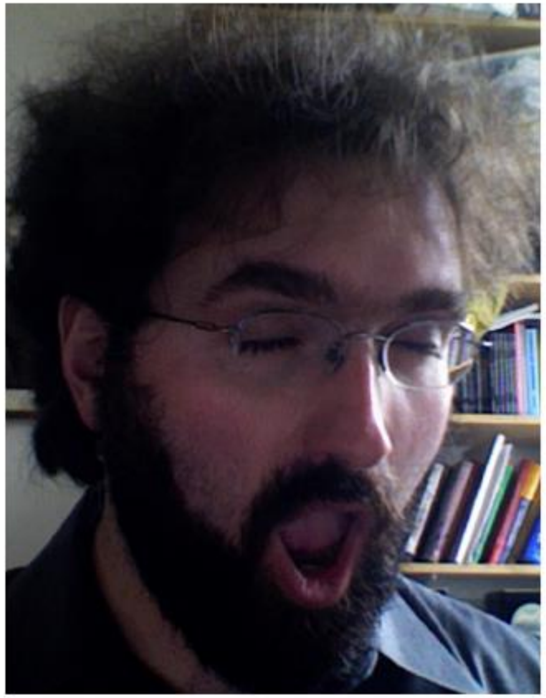
The challenge

- Smoothly transform a face into another



Why not just blend?

- Interpolate whole images:
  - Image(t) = (1-t) $*$ Image1 + t $*$ Image2



Why not just blend?

- Interpolate whole images:
  - Image(t) = (1-t) $*$ Image1 + t $*$ Image2

Just blending is not enough

- Features (eyes, mouth, etc.) are not aligned
- It is probably not possible to get a global alignment
- We need to also interpolate the LOCATION of features (warp!)

How to average a dog?

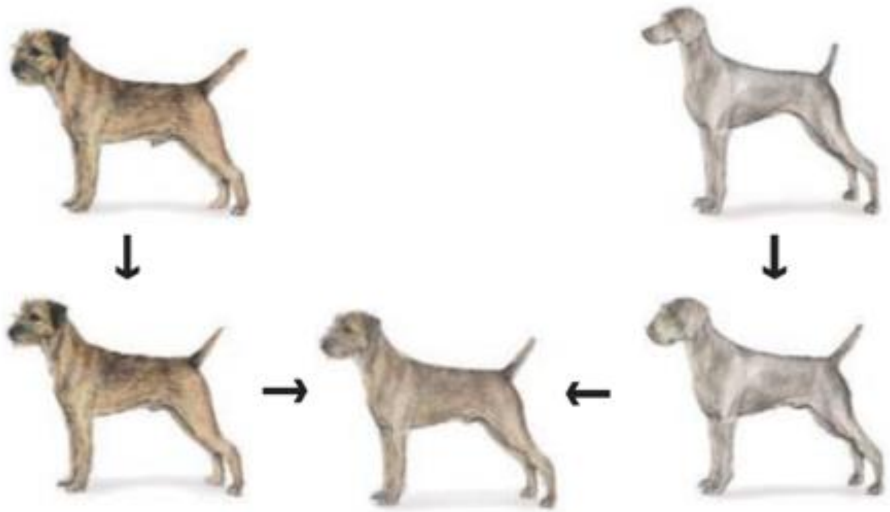
- What to do?
  - Cross-dissolve doesn’t work
  - Global alignment doesn’t work

- Feature matching
  - Nose to nose, tail to tail, etc.
  - This is a local warp!

Morphing: warp first, then cross-dissolve

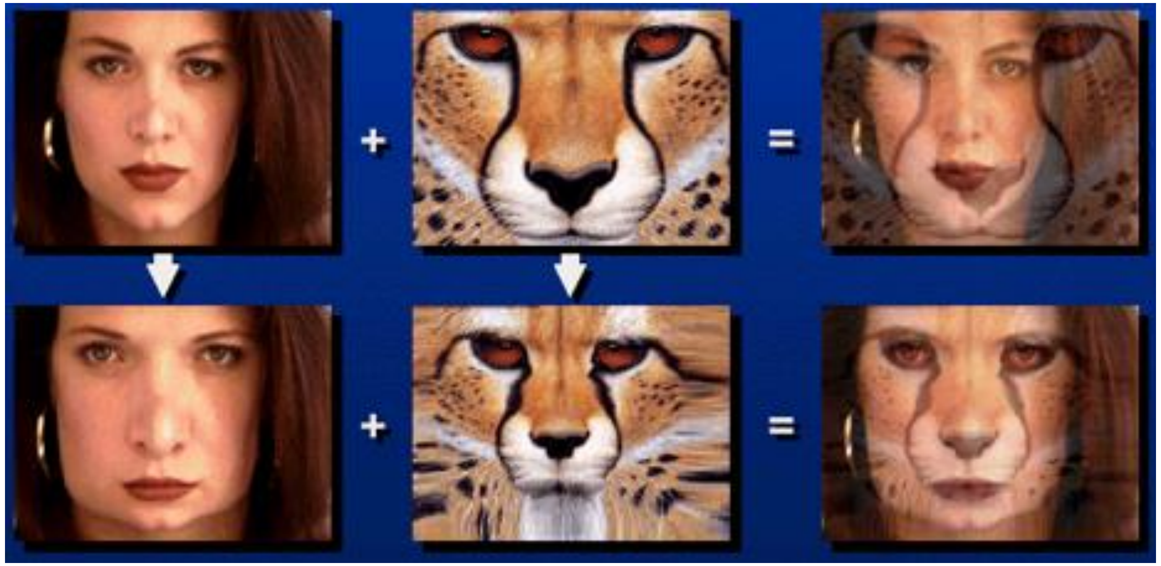
For an intermediate step t

1. Find the average shape (the “mean dog”)
    - local warping
2. Find the average color
    - Cross-dissolve the warped images



Summary: morphing sequence

- How do we create a morphing sequence?
    1. Create an intermediate shape (by interpolation)
    2. Warp both images towards it
    3. Cross-dissolve the colors in the newly warped images



Morphing is object averaging

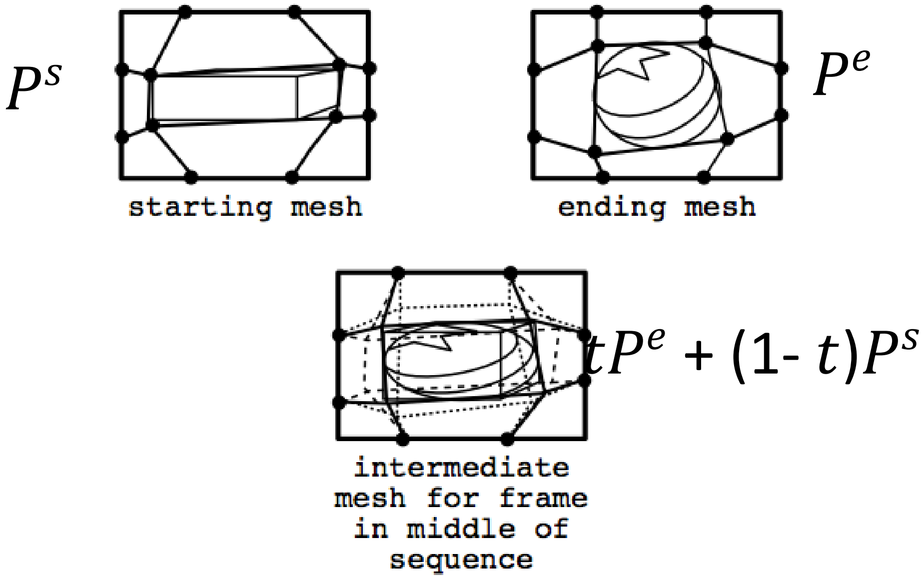
- The aim is to find "an average" between 2 objects
  - Not an average of two images of objects
  - …but an image of the average object!
- How do we know what the average object looks like?
  - We haven't a clue!
  - But we can often fake something reasonable, usually with required user/artist input

Morphing Schemes

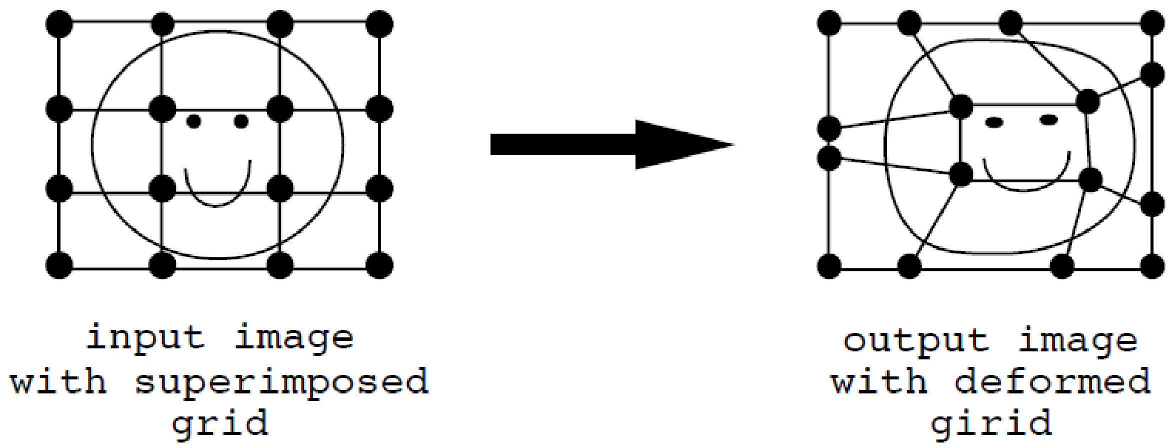
1. Mesh-based morph

Mesh-based warping

1. Specify a grid on top of each image
2. Deform grid vertices
3. At each time step
   - Interpolate the positions of vertices to get an intermediate mesh
   - Warp each image to this mesh (e.g., bilinear warp, separable mesh warp)
   - Blend the warped images



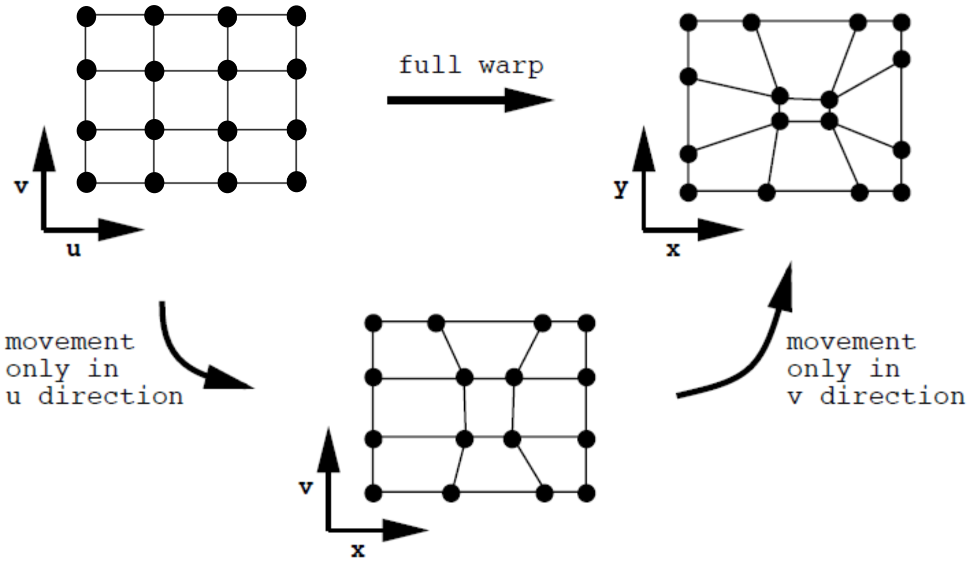
How to do the warp?



- Specify a grid on top of the image
- Deform vertices of the grid
- Warp image by treating each quadrilateral in the grid separately

Separable mesh warp

- Decompose the warp into
  - Horizontal pass
  - Vertical pass

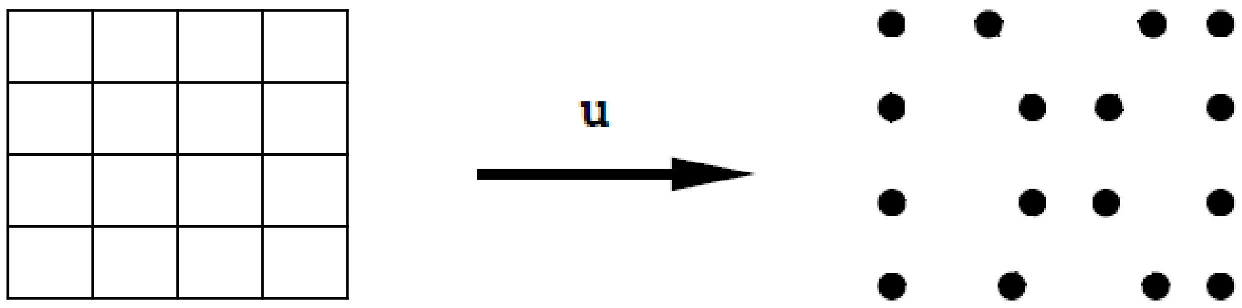


b) full mesh warp in two steps

Figure 12.10: Mesh Warp Implementation Using Horizontal and Vertical Displacements

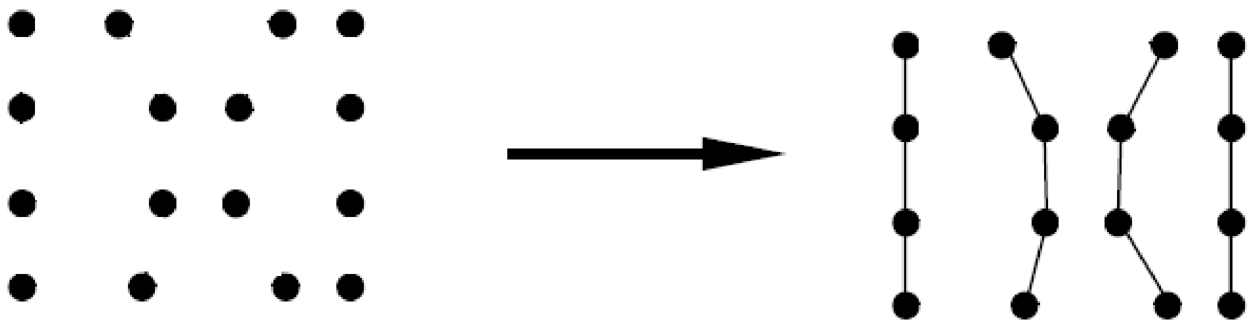
Horizontal pass

- Apply horizontal movement to each mesh point in the u direction



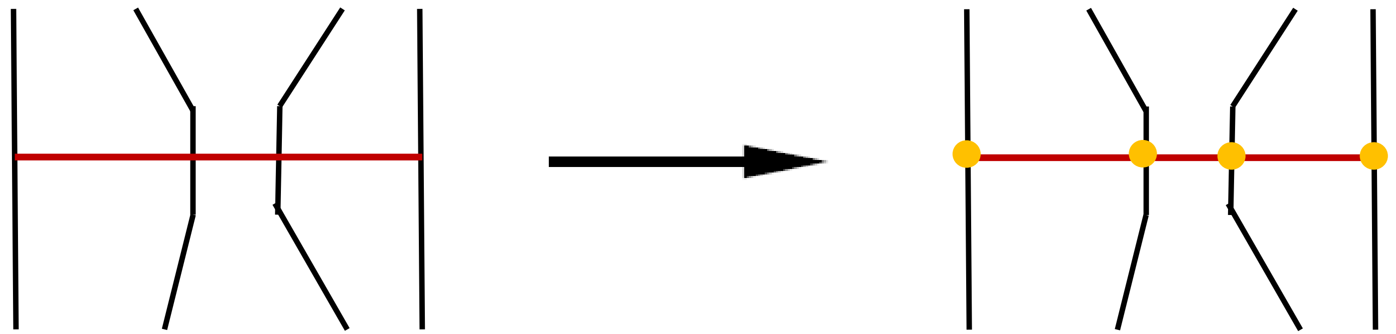
Horizontal pass

- Construct vertical line segments



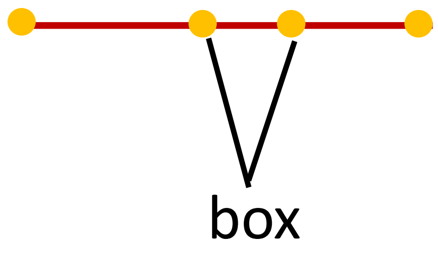
Horizontal pass

- Intersect scanlines with line segments



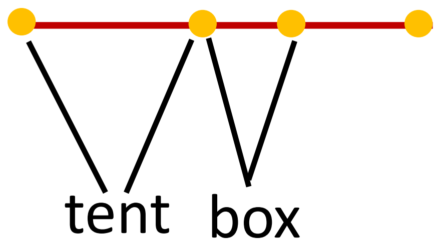
Horizontal pass

- Resample across each scanline
  - Use box filter to account for overlapping



Horizontal pass

- Resample across each scanline
  - Use box filter to account for overlapping
  - Use tent filter to fill in gaps



Separable mesh warp

- Better results using splines
- After horizontal pass, repeat the process over image columns

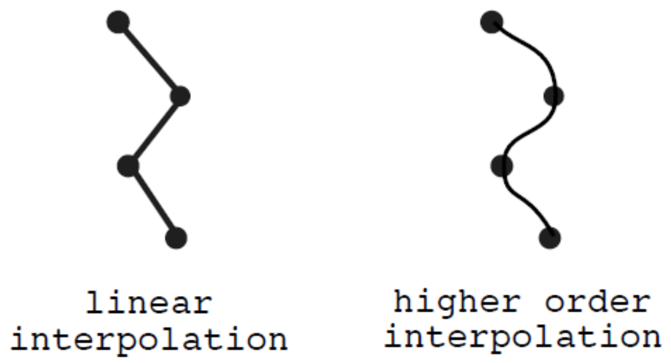


Figure 12.13: Linear Interpolation vs. Cubic Polynomial Interpolation

Mesh-based warping

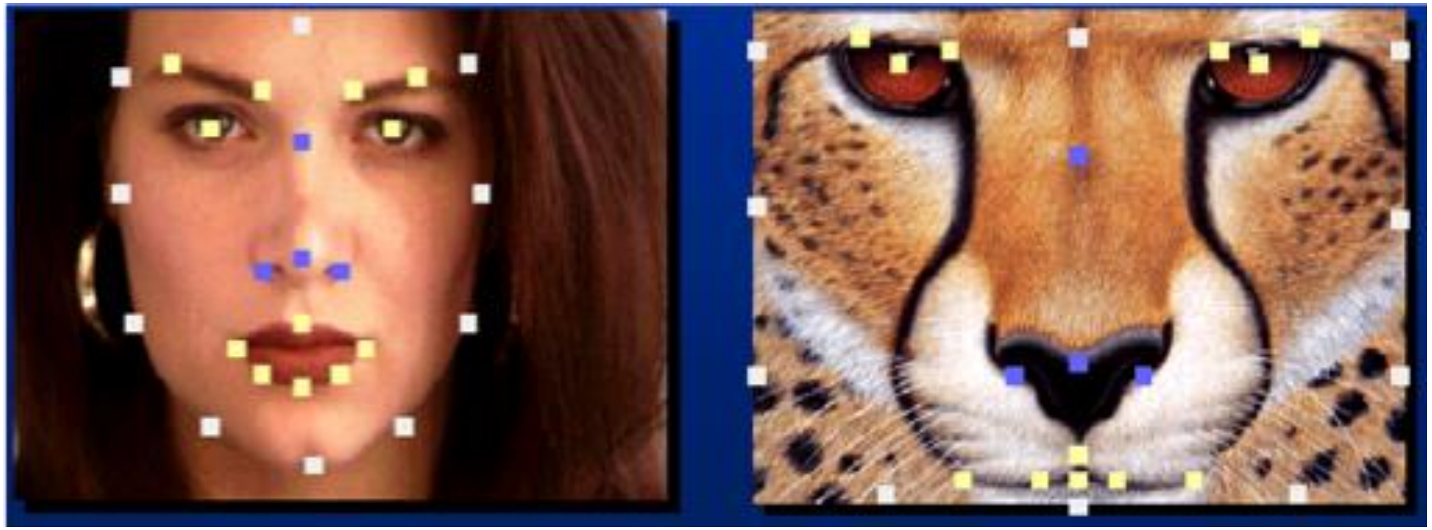
- How big of a grid is necessary?
- Really, we'd rather specify just a few points, not a grid

Morphing Schemes

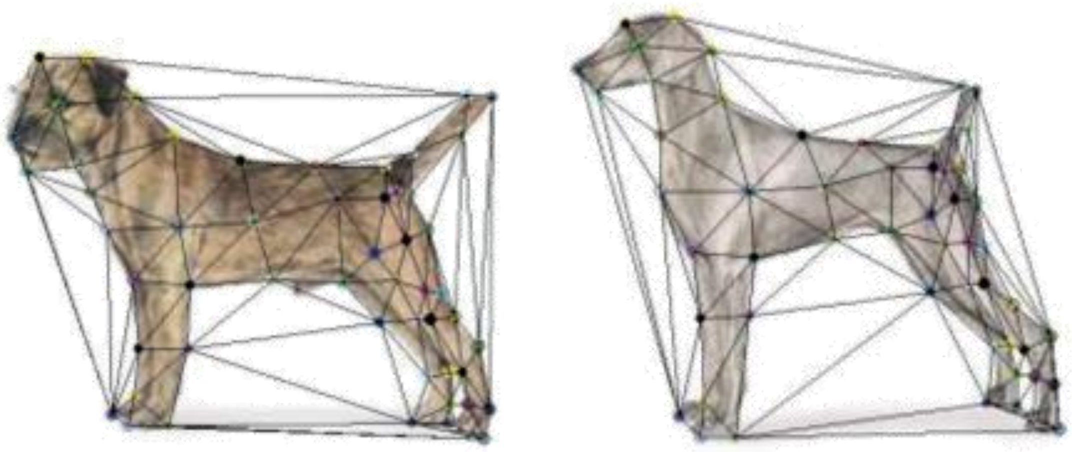
2. Triangle-based morph

Sparse specification for morphs

- Specify corresponding points
- Interpolate to a complete warping function
- How do we go from feature points to pixels?

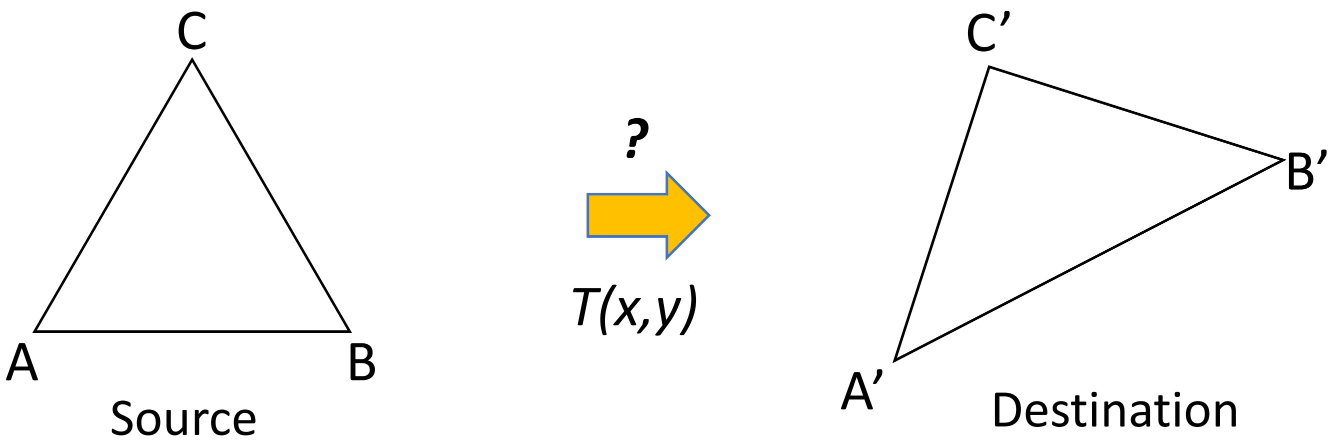


Use a triangle mesh



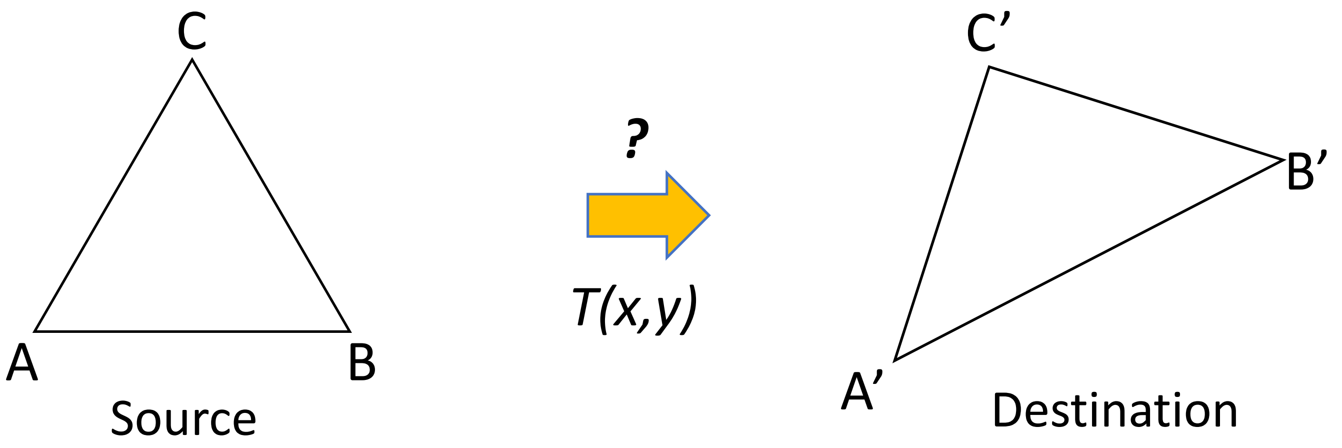
1. Input correspondences at key feature points
2. Define a triangular mesh over the points
   - Same mesh in both images!
   - Now we have triangle-to-triangle correspondences
3. Create an intermediate mesh and warp each triangle from source to destination

Warping a triangle: option #1



- Given two triangles: ABC and A'B'C' in 2D (12 numbers)
- Need to find transform T to transfer all pixels from one to the other
- What kind of transformation is T?

Warping a triangle: option #1

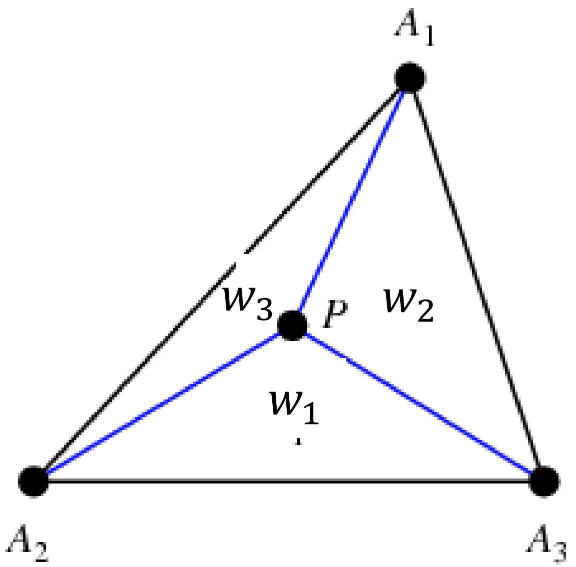


- Given two triangles: ABC and A'B'C' in 2D (12 numbers)
- Need to find transform T to transfer all pixels from one to the other
- What kind of transformation is T?
- How can we compute the transformation matrix?
  - Solve for it!

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ c & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Warping a triangle: option #2

- Use barycentric interpolation
- Each point P is an area-weighted average of the vertices of the triangle



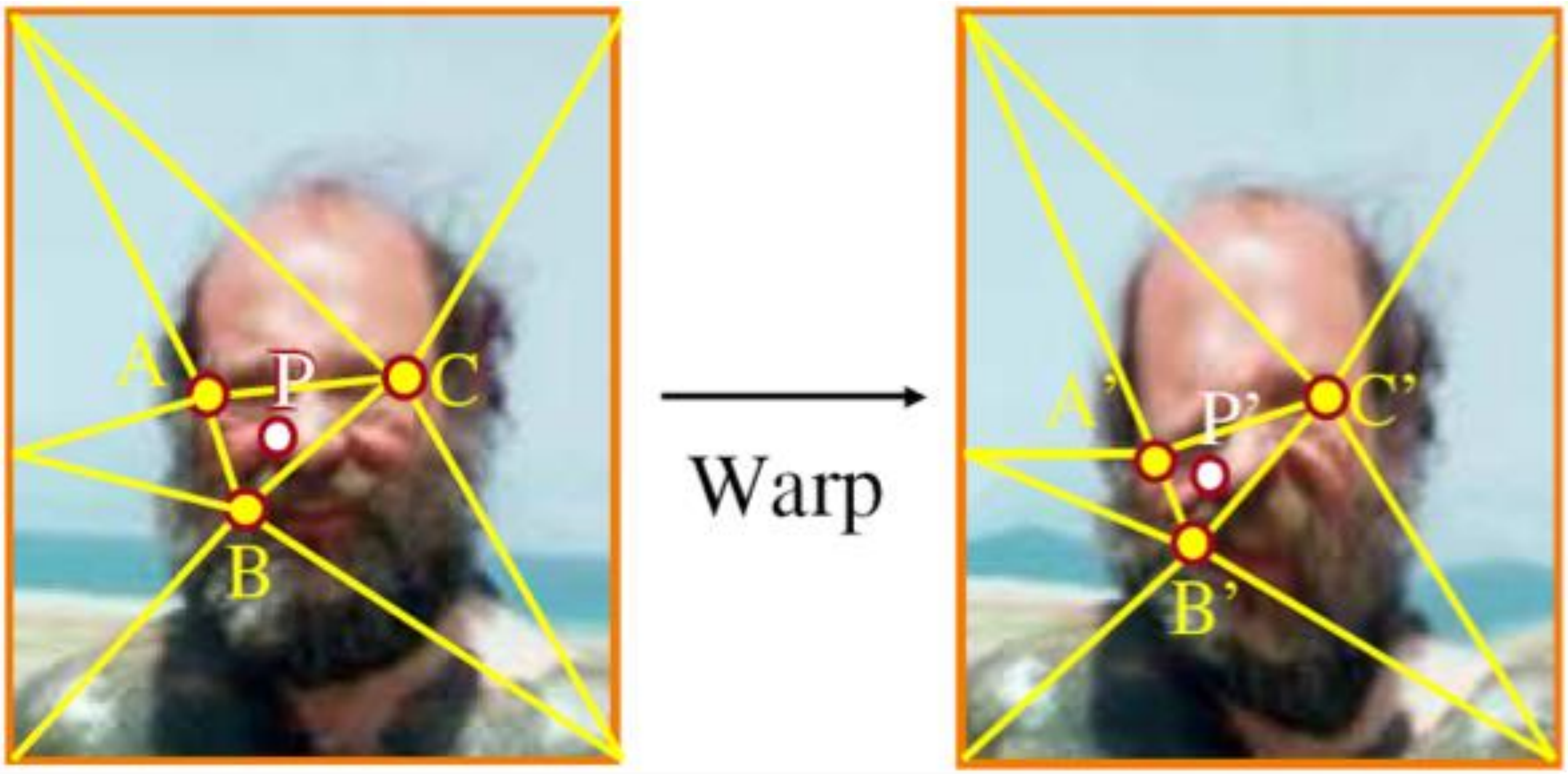
$$P = w_1 A_1 + w_2 A_2 + w_3 A_3$$

$$w_1 + w_2 + w_3 = 1$$

http://mathworld.wolfram.com/BarycentricCoordinates.html

Example: Triangle warping

$P = w_1 A + w_2 B + w_3 C$ $\quad$ $P' = w_1 A' + w_2 B' + w_3 C'$



The inverse map works the same way

Morphing Schemes

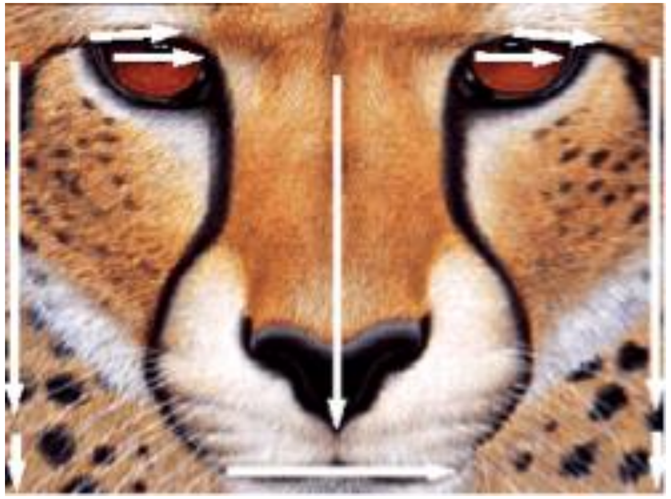
3. Segment-based morph

Michael Jackson's Black or White

http://youtu.be/F2AitTPI5U0?t=5m15s

The Beier & Neely algorithm

- Specify the warp by specifying corresponding vectors
- Interpolate to a complete warping function

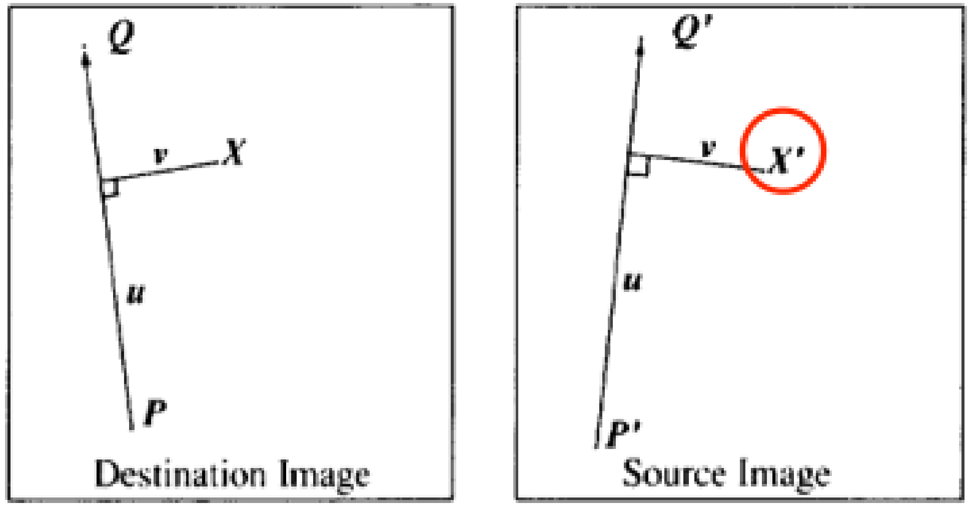


The Beier & Neely algorithm

- Given images A and B, and their corresponding sets of vectors, V_A and V_B, for each time frame t:
  - Interpolate the positions of the corresponding vectors based on t to determine the destination (intermediate) shape
  - Warp image A to the destination shape
  - Warp image B to the destination shape
  - Cross-dissolve using a blending factor of t

Warping with line segments

- Given ***PQ*** and ***P'Q'***, warp a point ***X*** to a point ***X'***
  - Measure distance u along the segment
  - The direction of $\boldsymbol{PQ}$ defines a side, measure distance v from the right side



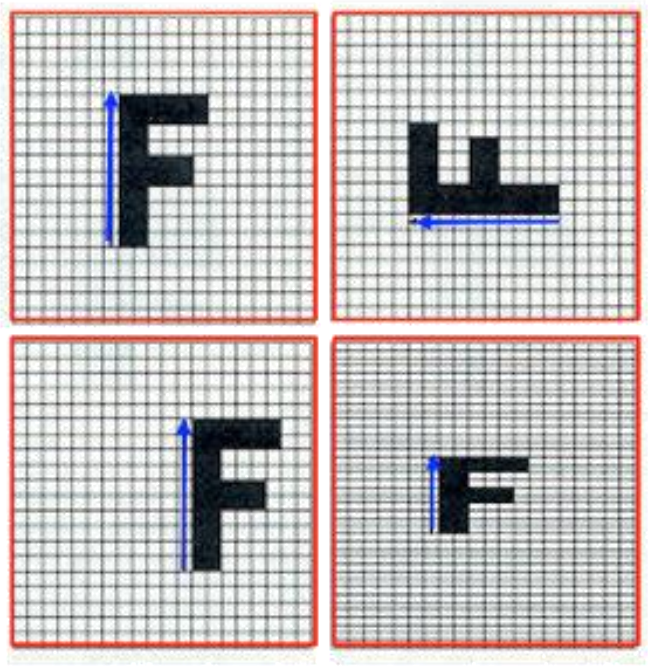
$$u = \frac{(\boldsymbol{X} - \boldsymbol{P}) \cdot (\boldsymbol{Q} - \boldsymbol{P})}{||\boldsymbol{Q} - \boldsymbol{P}||^2}$$

$$v = \frac{(\boldsymbol{X} - \boldsymbol{P}) \cdot perp(\boldsymbol{Q} - \boldsymbol{P})}{||\boldsymbol{Q} - \boldsymbol{P}||}$$

$$\boldsymbol{X}' = \boldsymbol{P}' + u \cdot (\boldsymbol{Q}' - \boldsymbol{P}') + \frac{v \cdot perp(\boldsymbol{Q}' - \boldsymbol{P}')}{||\boldsymbol{Q}' - \boldsymbol{P}'||}$$

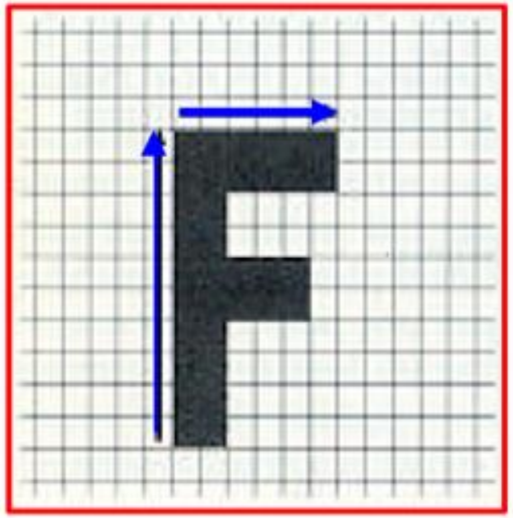
One line segment (globally) warping an image

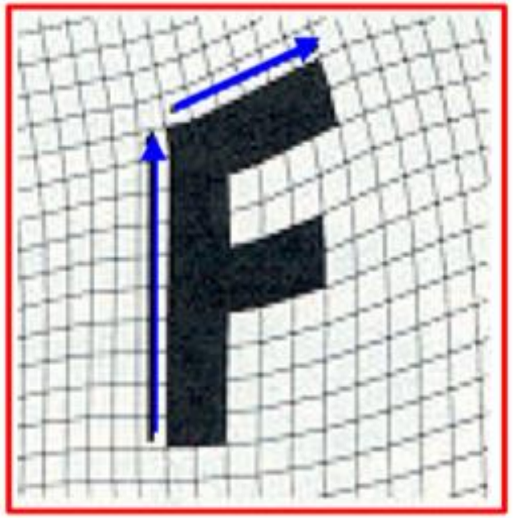
- For each X in the destination image:
  - Find the corresponding u, v
  - Find X' in the source image for that u, v
  - OUT(X) = IN(X')



Multiple line segments

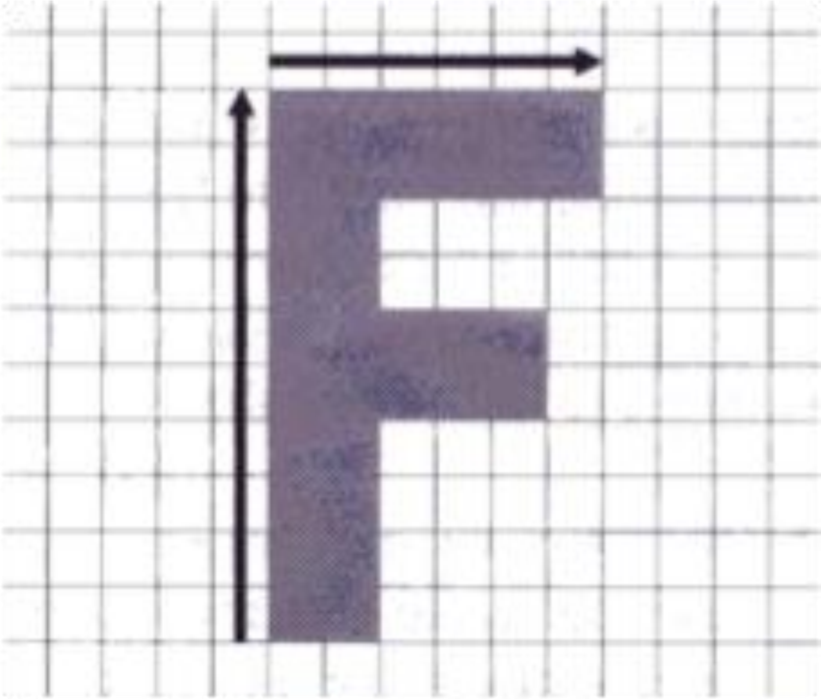
See [Beier and Neely, 1992] for details

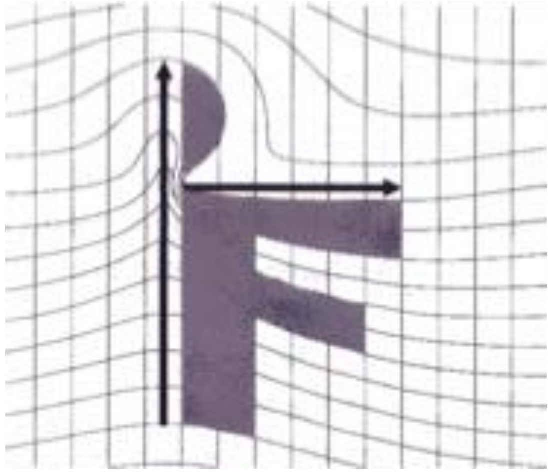




Comparison to mesh morphing

- Pros: More Expressive
- Cons: Speed and Control

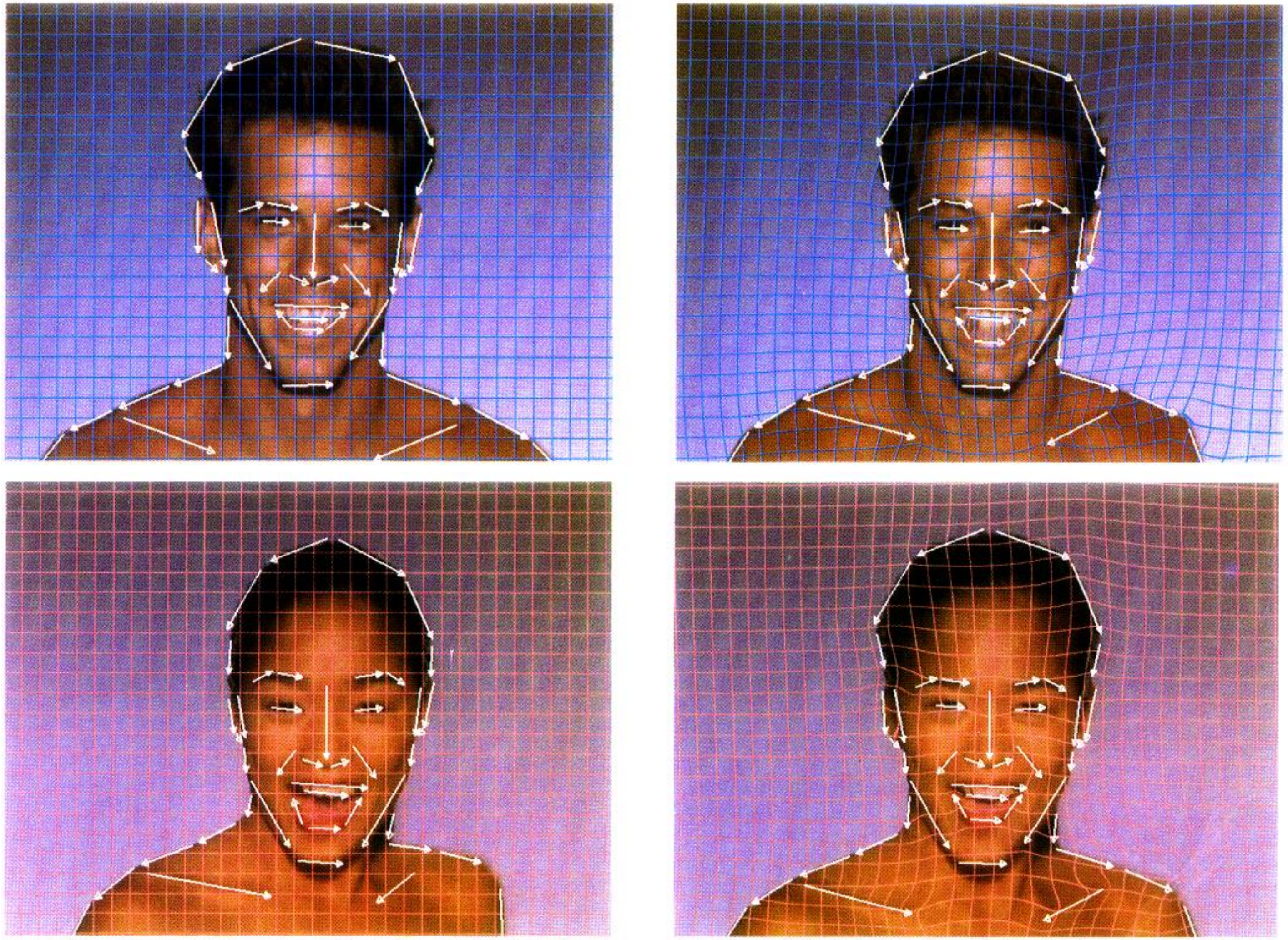




Example from Black or White

initial

t=0.5 (warped)



Example from Black or White

initial

t=0.5 (warped)

Example from Black or White