CPSC 4040/6040 Computer Graphics Images

Joshua Levine
levinej@clemson.edu

Lecture 20
Morphing

Nov. 5, 2015

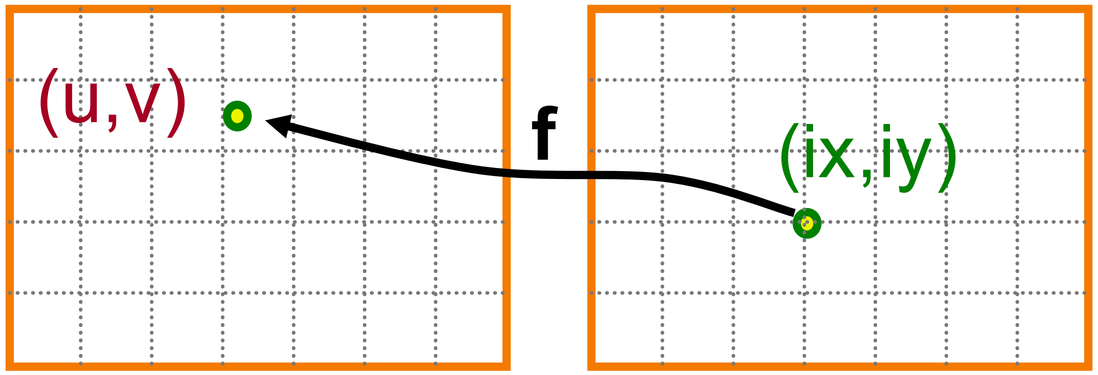
Slide Credits:
Szymon Rusinkiewicz
Frédo Durand
Alexei Efros
Yung-Yu Chung

Some Final Thoughts on Warping and Sampling

A Simple Example

Possible implementation of image scale:

```
Scale(src, dst, sx, sy) {
  w ≈ max(1/sx,1/sy);
  for (int ix = 0; ix < xmax; ix++) {
    for (int iy = 0; iy < ymax; iy++) {
      float u = ix / sx;
      float v = iy / sy;
      dst(ix,iy) = Resample(src,u,v,k,w);
    }
  }
}
```

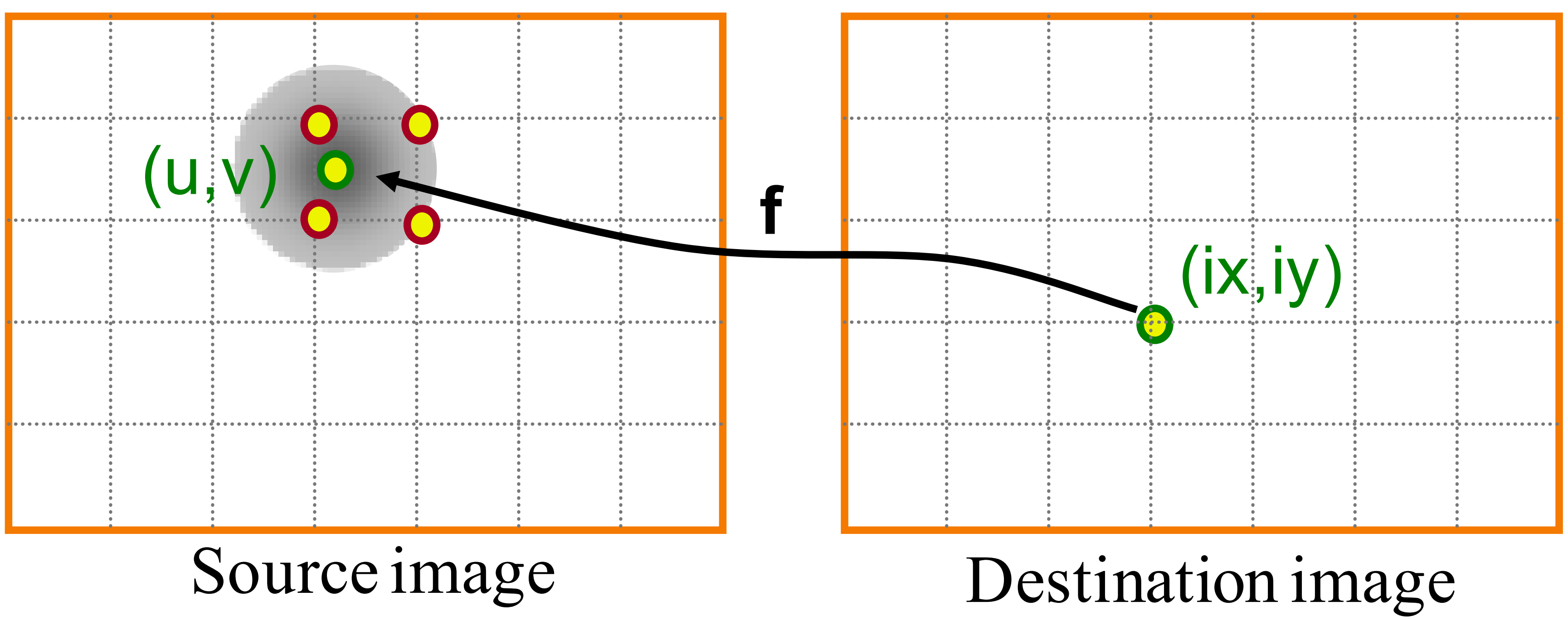


Source image   Destination image

Forward vs. Reverse Mapping

Reverse mapping:

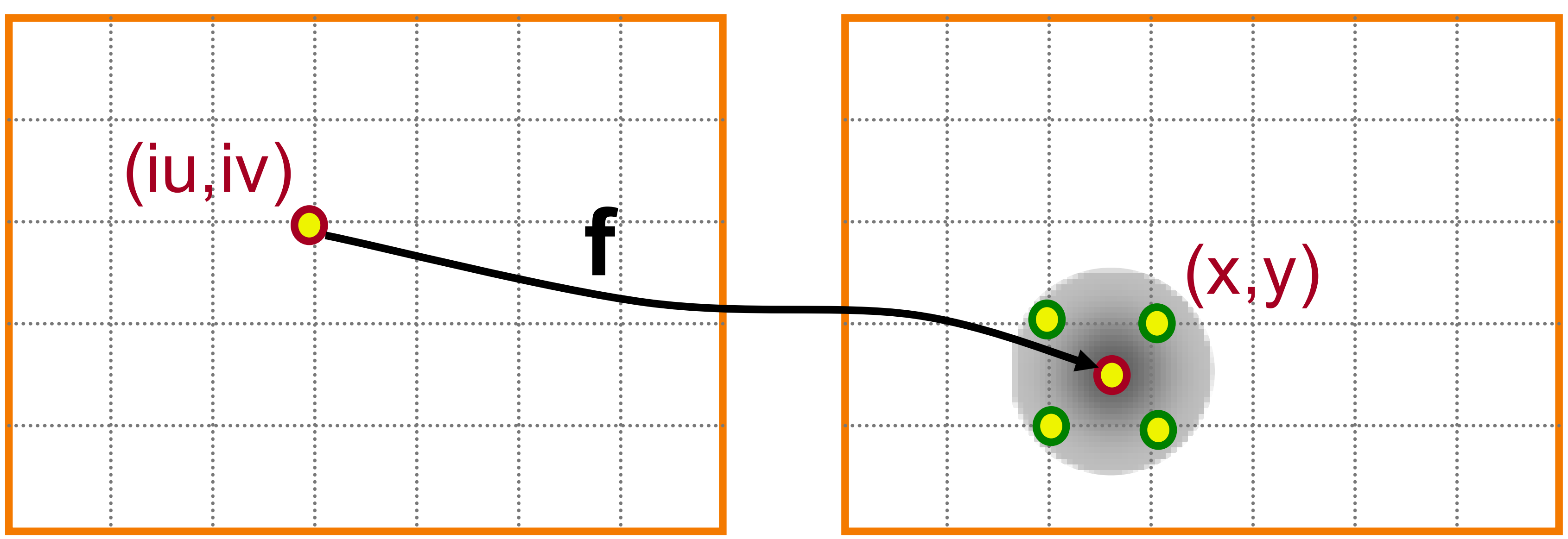
```
Warp(src, dst) {
   for (int ix = 0; ix < xmax; ix++) {
      for (int iy = 0; iy < ymax; iy++) {
         float w ≈ 1 / scale(ix, iy);
         float u = fx^-1(ix,iy);
         float v = fy^-1(ix,iy);
         dst(ix,iy) = Resample(src,u,v,w);
      }
   }
}
```



Forward vs. Reverse Mapping

Forward mapping:

```
Warp(src, dst) {
  for (int iu = 0; iu < umax; iu++) {
    for (int iv = 0; iv < vmax; iv++) {
      float x = fx(iu,iv);
      float y = fy(iu,iv);
      float w ≈ 1 / scale(x, y);
      Splat(src(iu,iv),x,y,k,w);
    }
  }
}
```



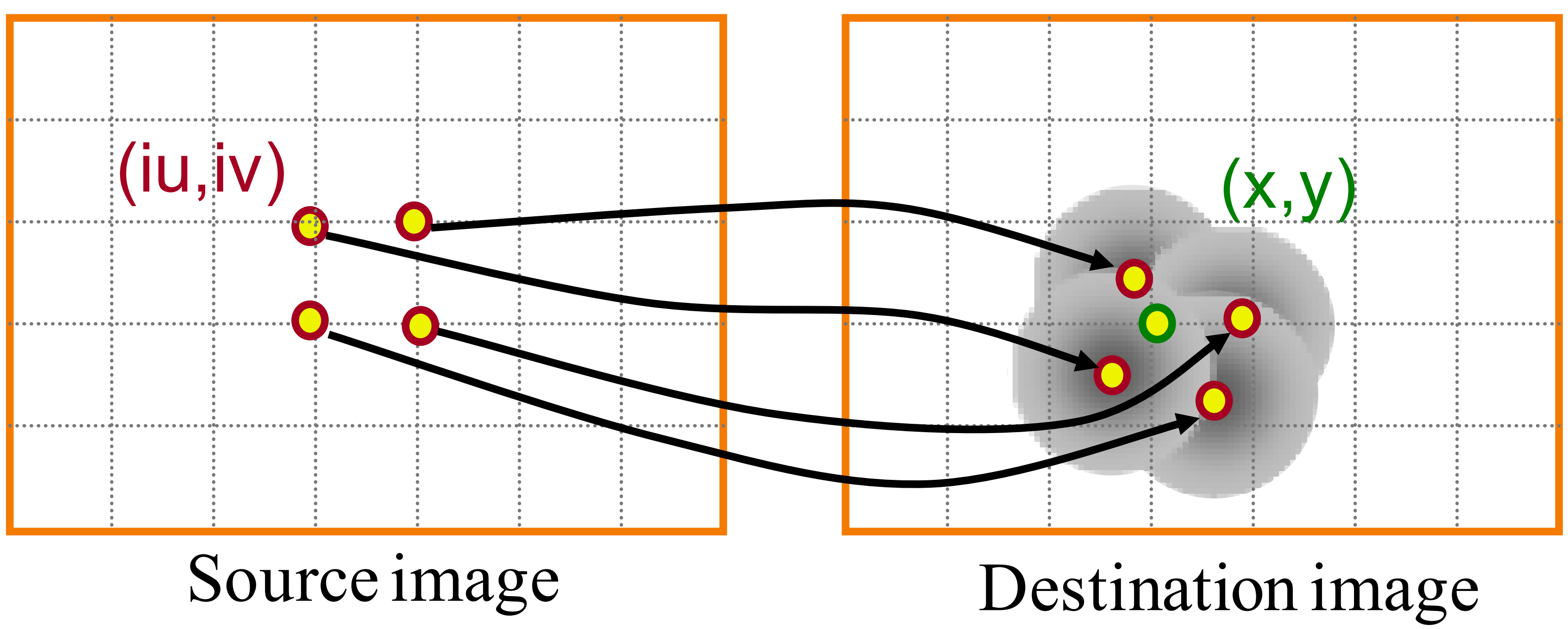
Source image

Destination image

Forward vs. Reverse Mapping

Forward mapping:

```
Warp(src, dst) {
   for (int iu = 0; iu < umax; iu++) {
      for (int iv = 0; iv < vmax; iv++) {
         float x = fx(iu,iv);
         float y = fy(iu,iv);
         float w ≈ 1 / scale(x, y);
         Splat(src(iu,iv),x,y,k,w);
      }
   }
}
```

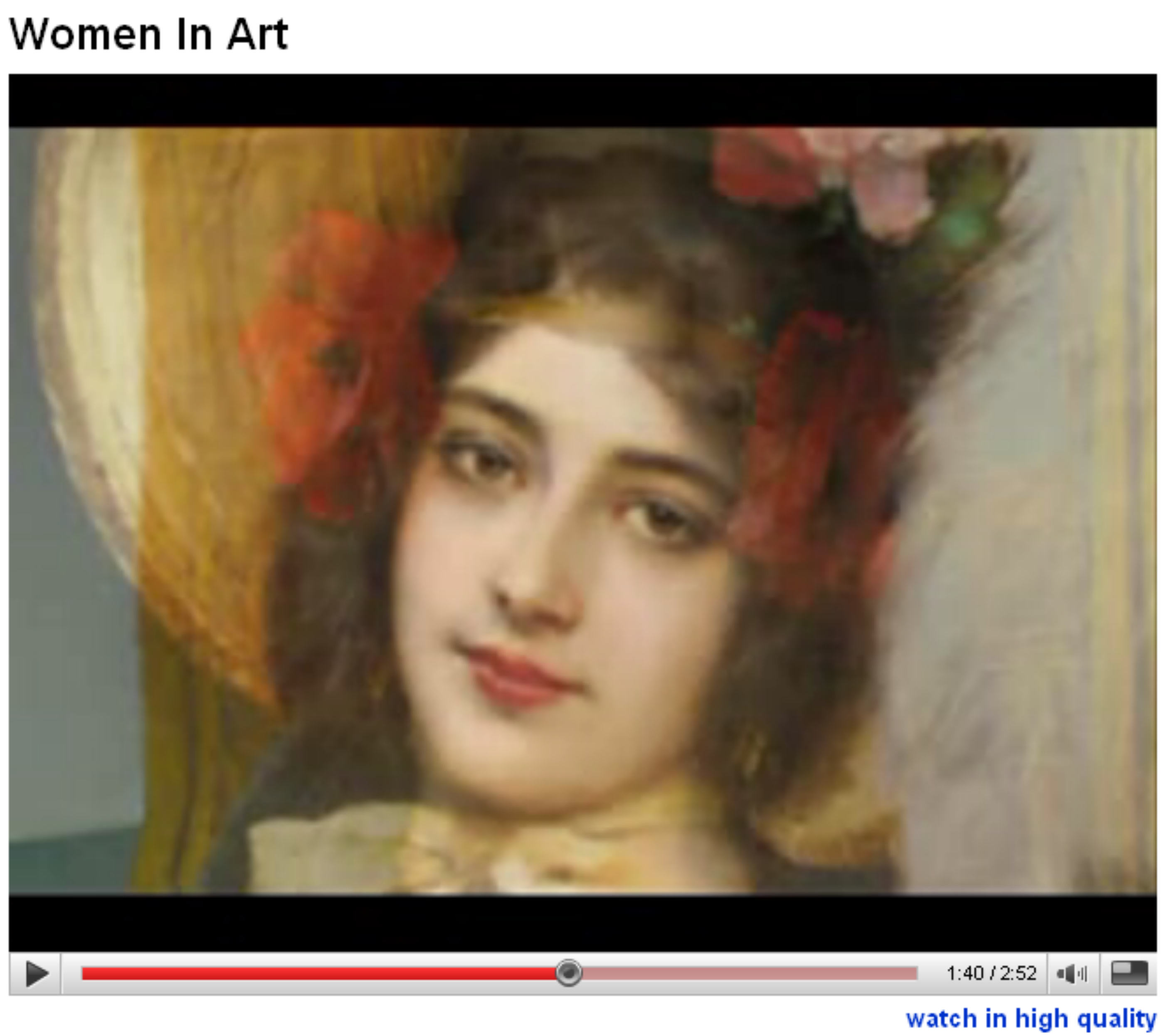


Morphing

Morphing in Film: Willow

http://www.telegraph.co.uk/culture/culturevideo/filmvideo/film-clips/9937381/Willow-a-clip-of-the-morphing-sequence.html

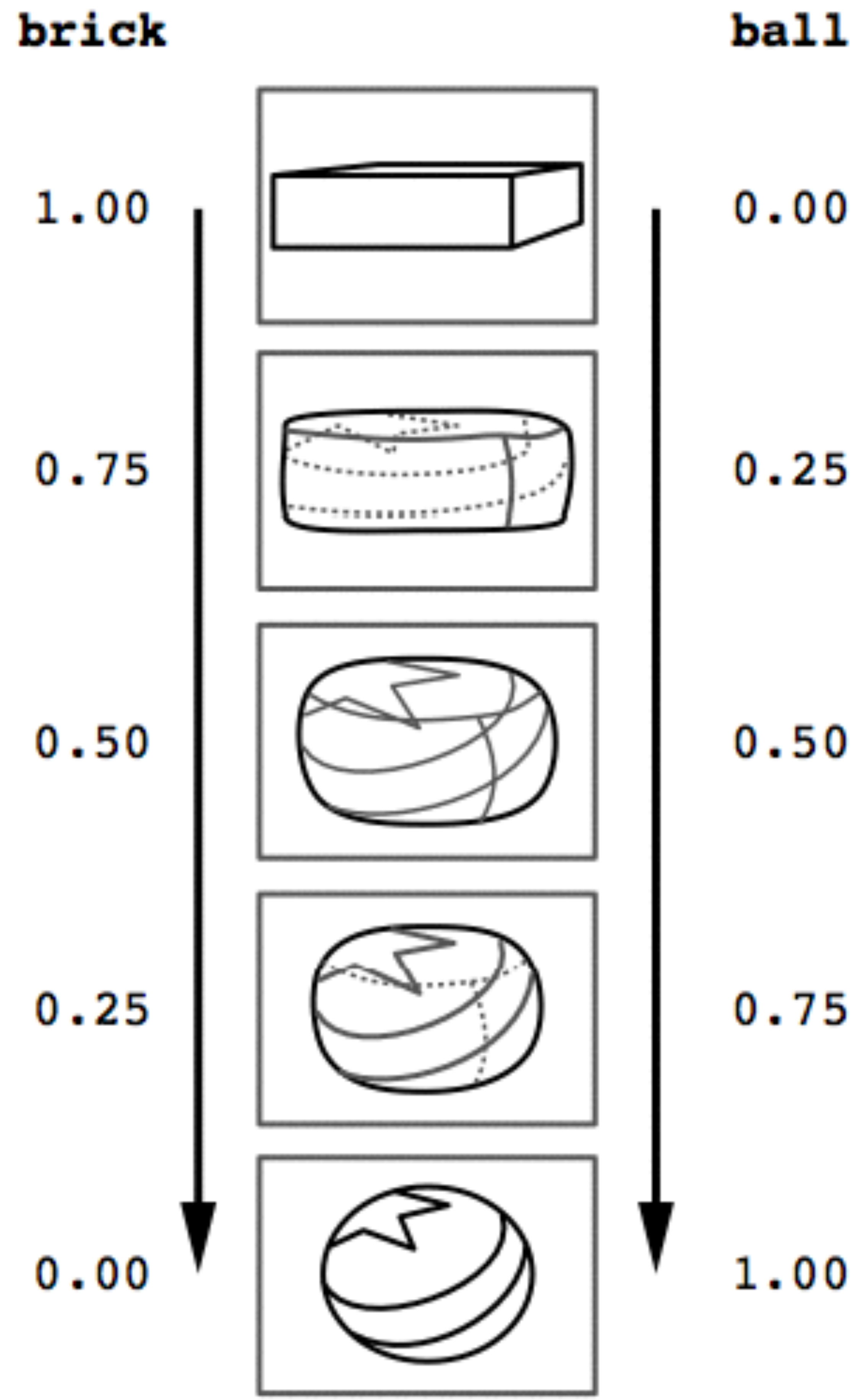
Women in Art



http://youtu.be/nUDIoN-_Hxs

Steps In a Morph

- We have been discussing warping as a start and a finish
- With morphing, we want the intermediate steps as well
- Typically use interpolation over some “time” t to get them!



Cross Dissolving

The Challenge

- “Smoothly” transform a face into another
- Related: slow motion interpolation interpolate between key frames

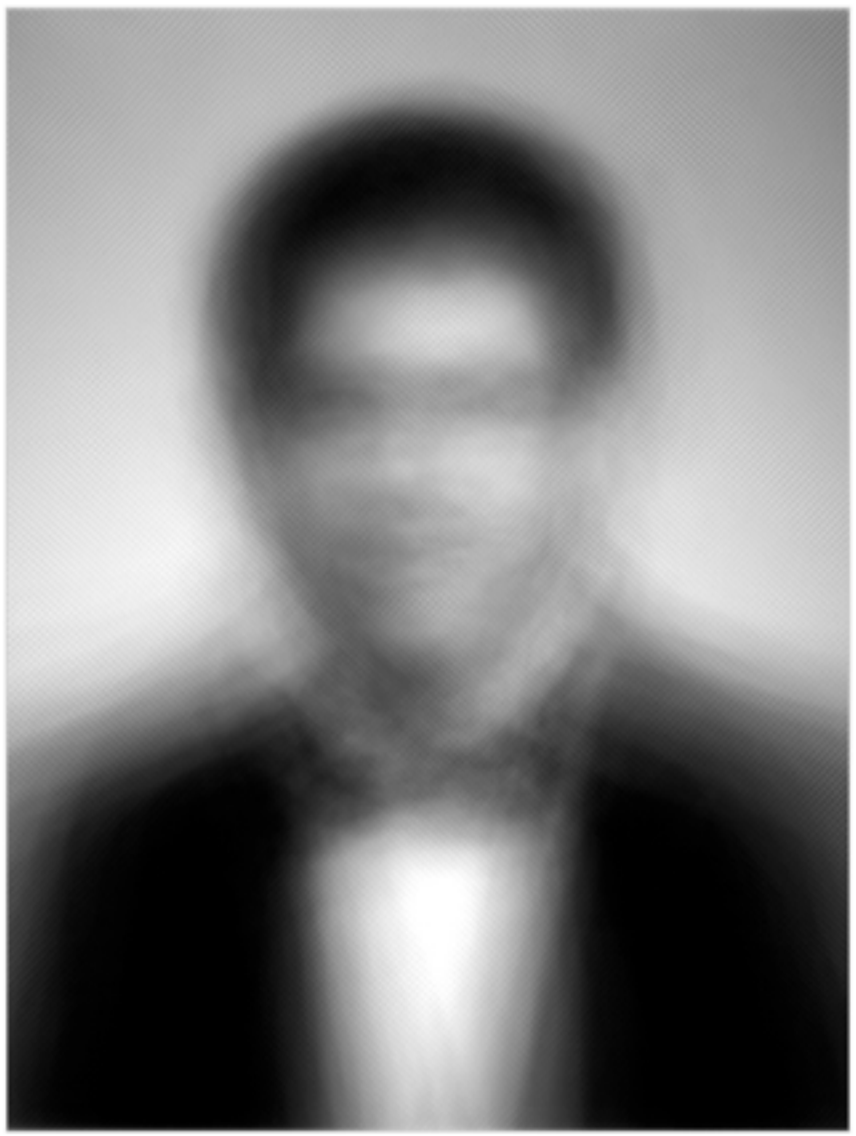
Averaging Images

- Interpolate whole images:
- Image(t) = (1-t)*Image1 + t*image2

Problems with Cross Fading

- 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
    - Domain transformation = warp!

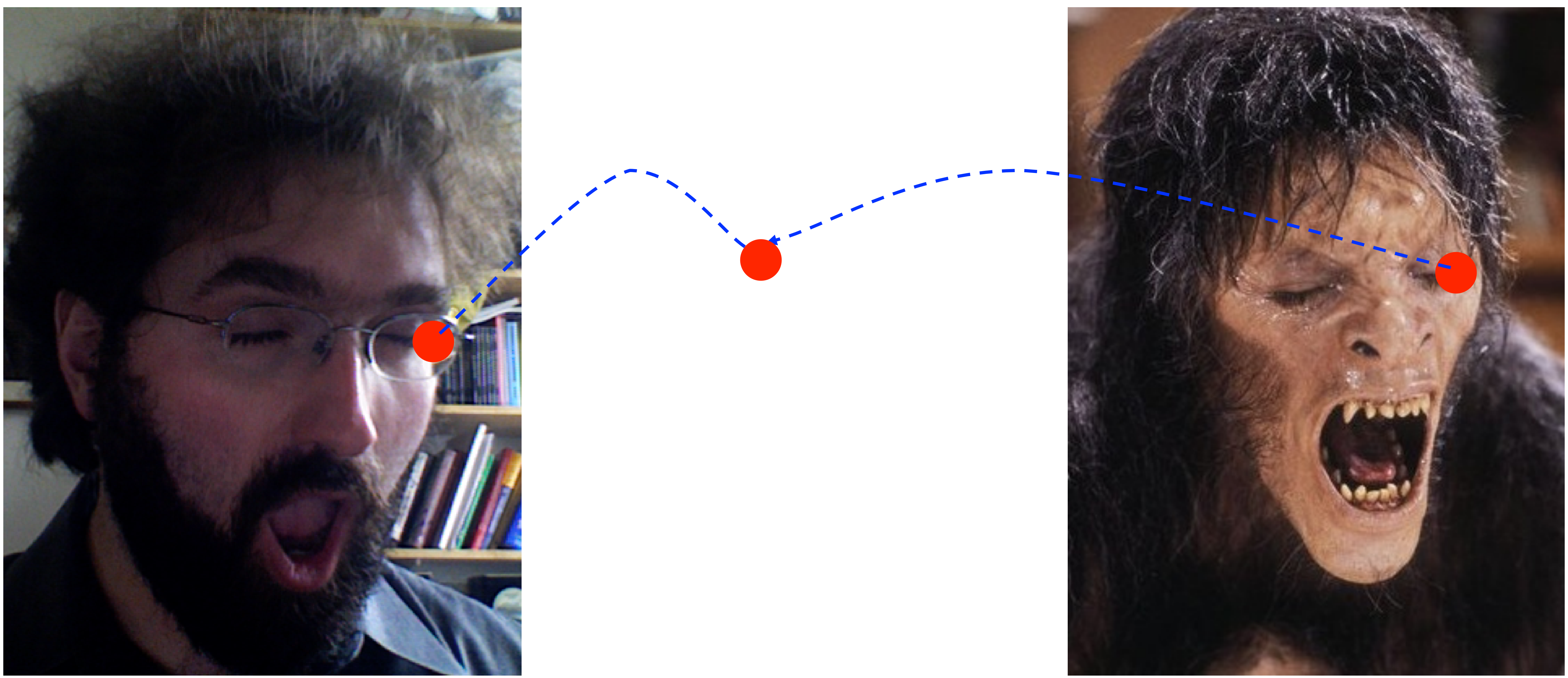
Artistic Uses of Cross-Dissolving



http://www.salavon.com/work/Class/

Morphing: Combines Warping and Color Interpolation

- For each pixel
  - Transform its location like a vector (domain warp)
  - Then linearly interpolate colors (interpolation)



Sometimes Image Alignment is Sufficient

- Align first, then cross-dissolve
- Alignment using global warp

Sometimes Image Alignment is Sufficient

- Align first, then cross-dissolve
- Alignment using global warp

Sometimes Image Alignment is Sufficient

- Align first, then cross-dissolve
- Alignment using global warp

Sometimes Image Alignment is Sufficient

- Align first, then cross-dissolve
- Alignment using global warp

Sometimes Image Alignment is Sufficient

- Align first, then cross-dissolve
- Alignment using global warp

Sometimes Image Alignment is Sufficient

- Align first, then cross-dissolve
- Alignment using global warp

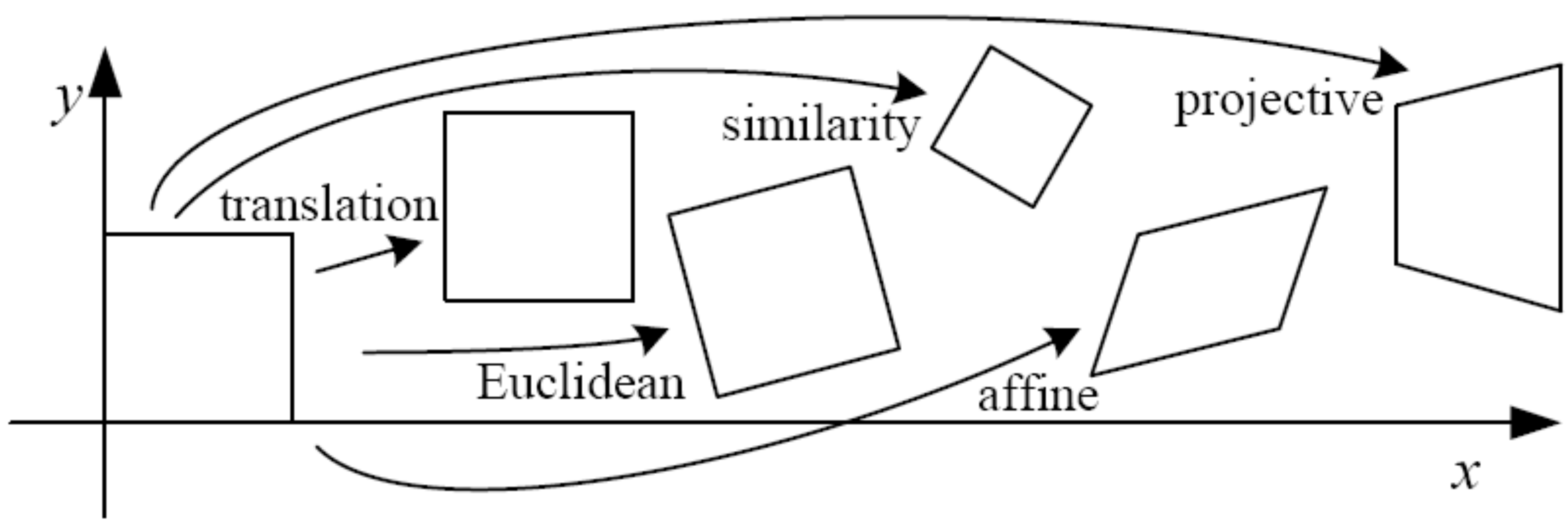
Morphing: Object Averaging

Image Average

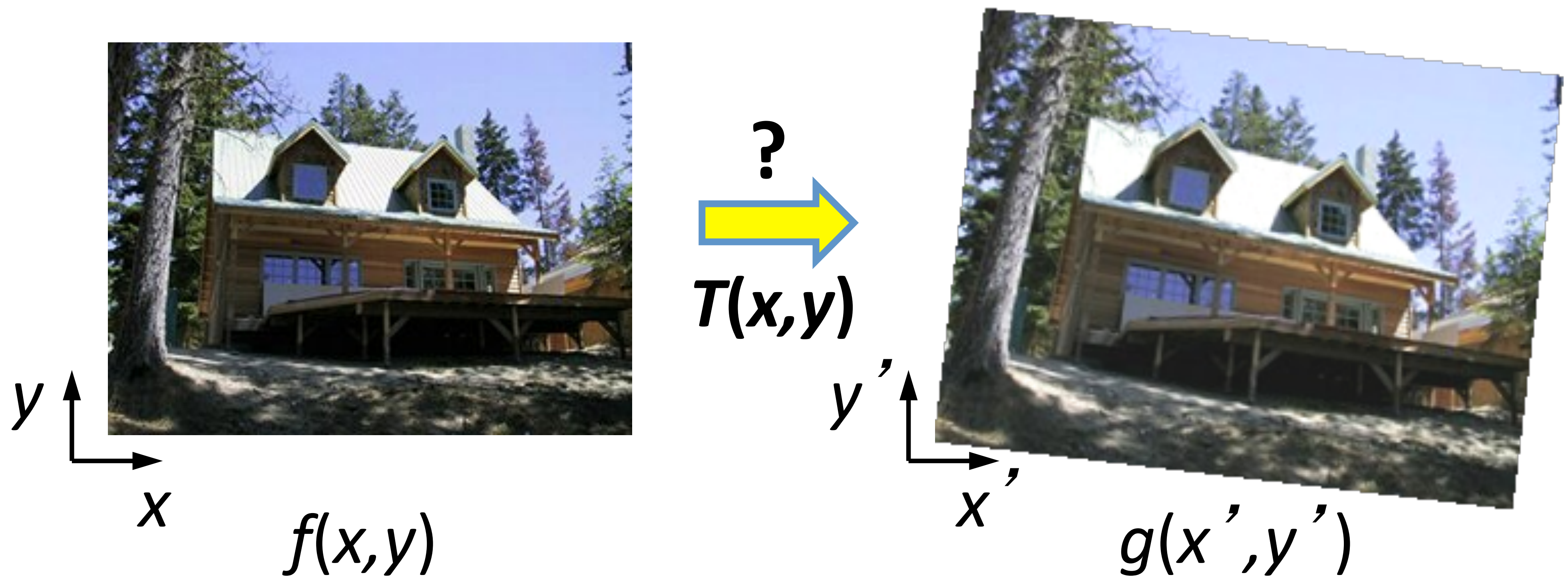
Object Average

- The aim is to find "an average" between two 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

Recovering Information in Warps

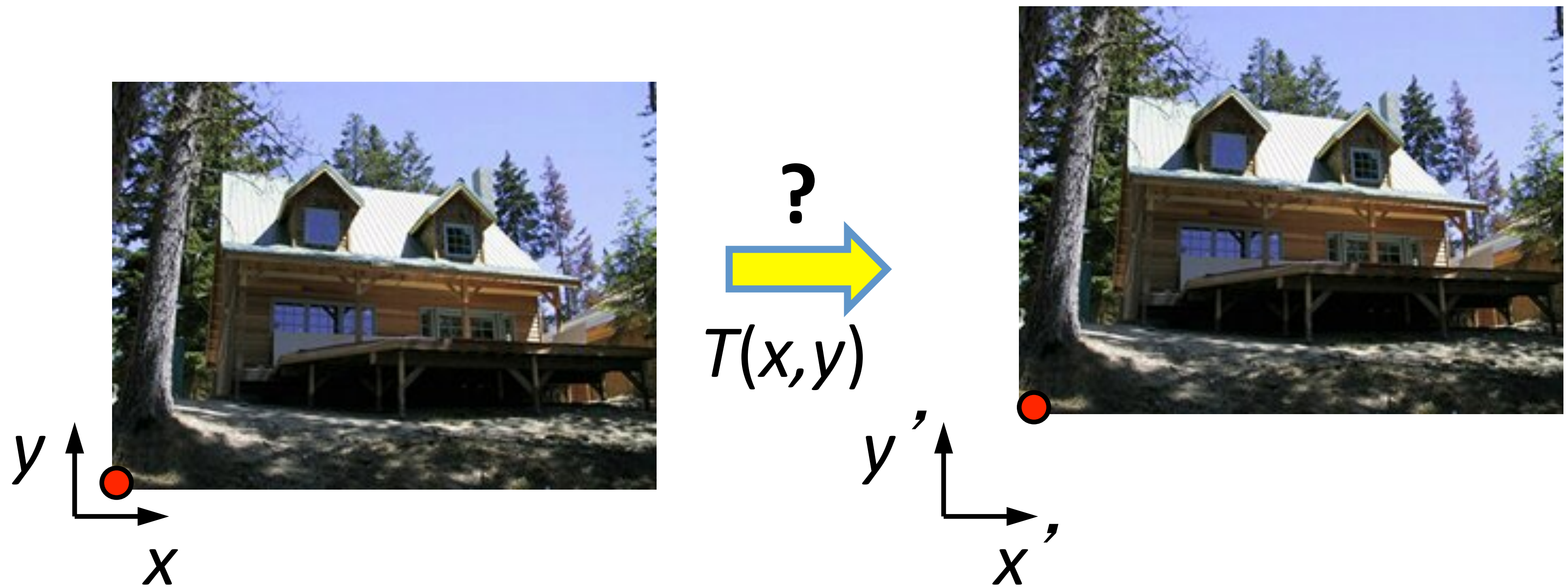


Recovering Transformations



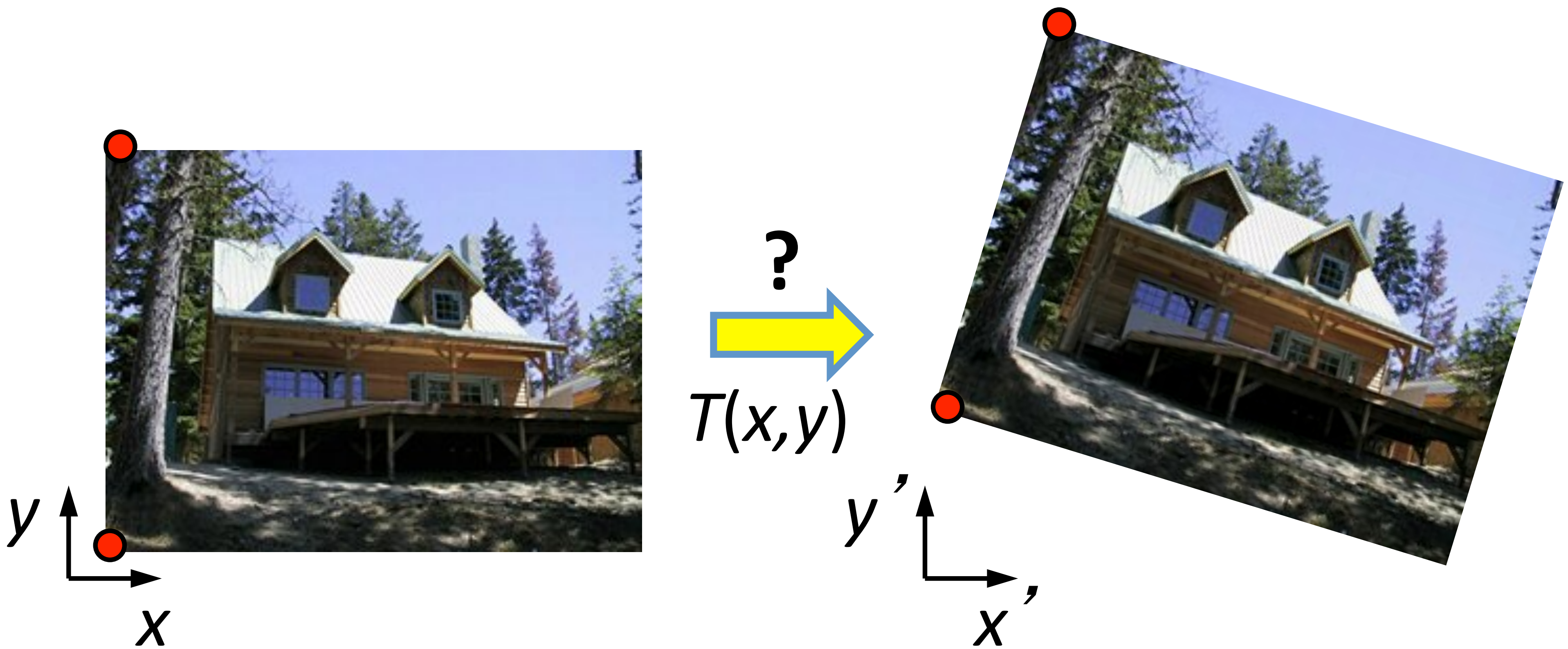
- What if we know f and g and want to recover the transform T?
  - Willing to let user provide correspondences
- How many do we need?

Translations: # of Correspondences?



- How many correspondences needed for translation?
  - How many Degrees of Freedom?
- What is the transformation matrix?

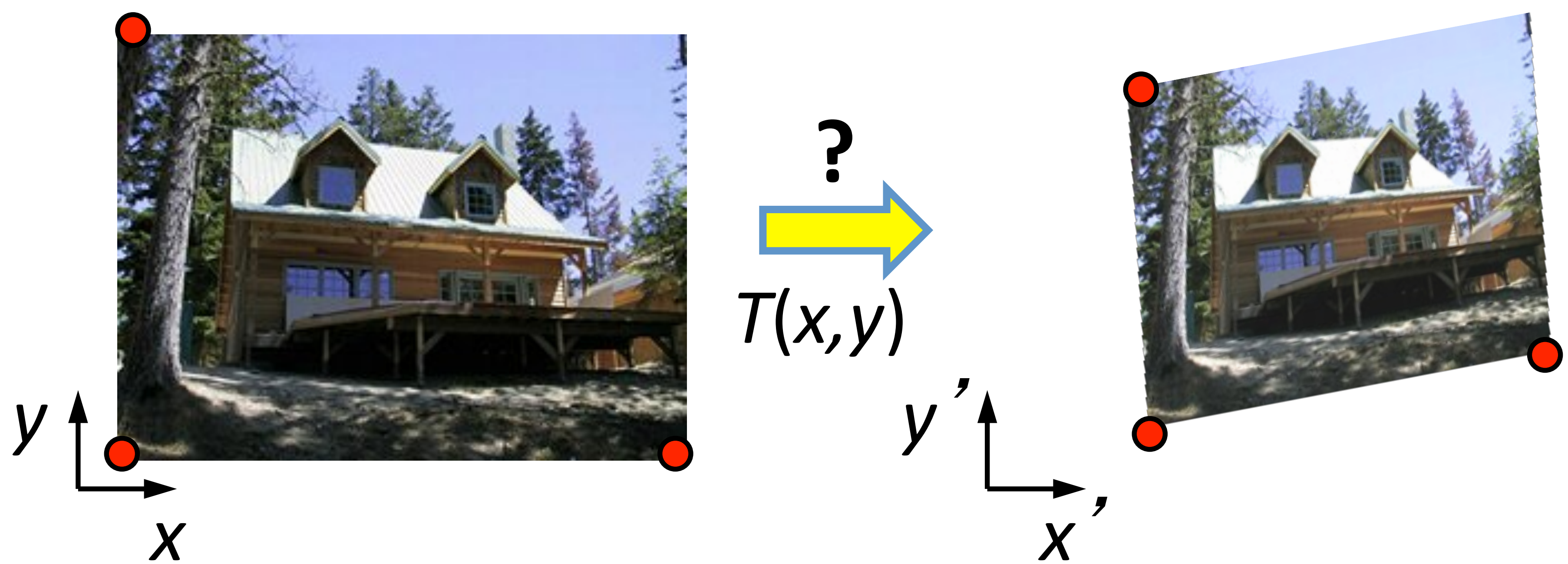
Euclidian: # correspondences?



- How many correspondences needed for translation +rotation?

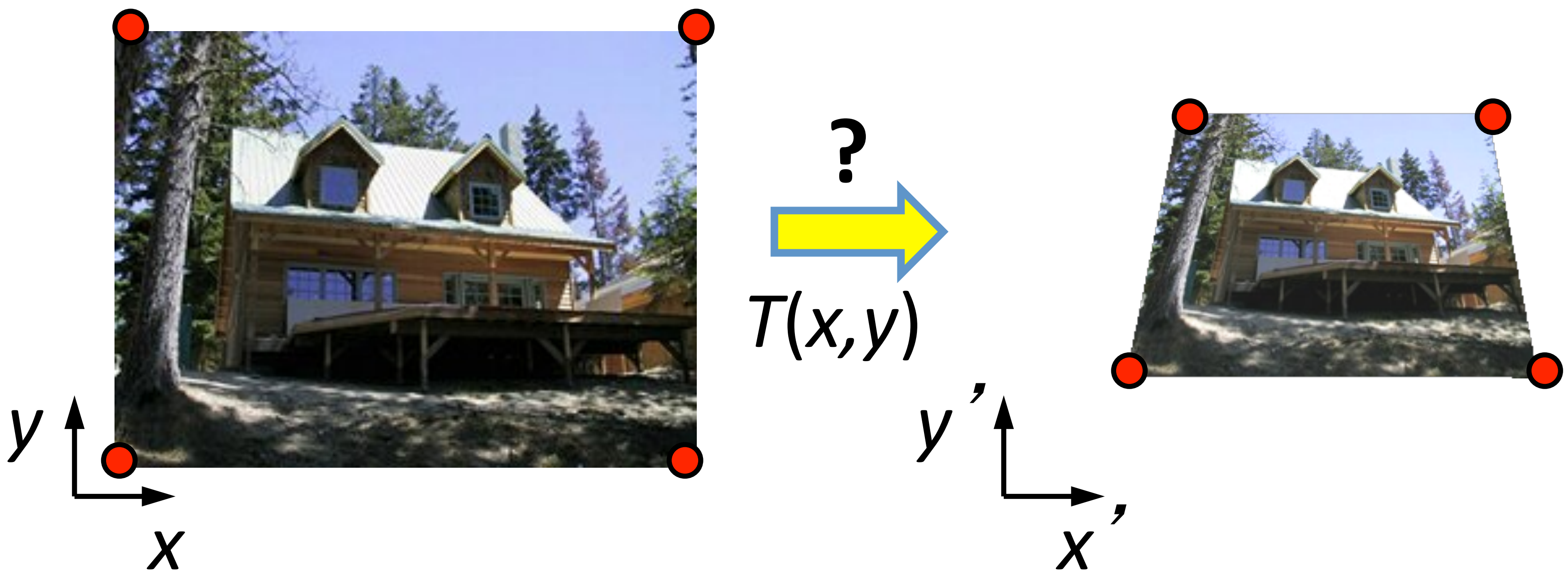
- How many DOF?

Affine: # of Correspondences?



- How many correspondences needed for affine?
- How many DOF?

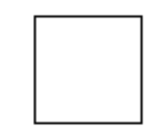
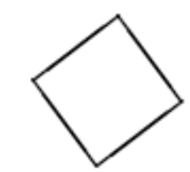
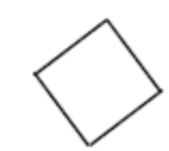
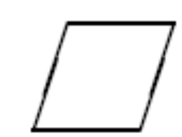
Projective: # of Correspondences?



- How many correspondences needed for projective?

- How many DOF?

Summary

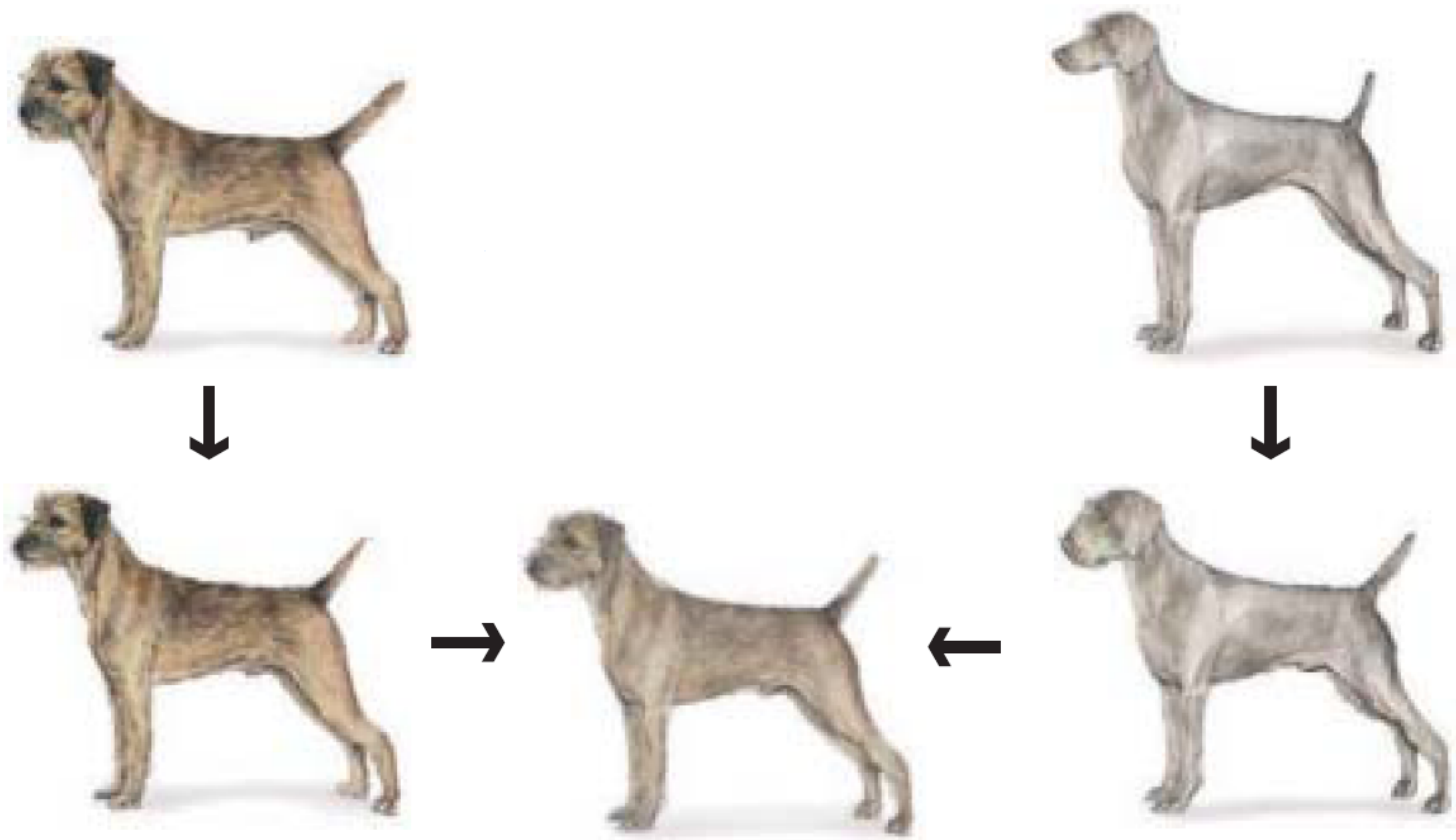
Name	Matrix	# D.O.F.	Preserves:	Icon
translation	$\begin{bmatrix} \boldsymbol{I} & \vert & \boldsymbol{t} \end{bmatrix}_{2\times 3}$	2	orientation $+ \cdots$	
rigid (Euclidean)	$\begin{bmatrix} \boldsymbol{R} & \vert & \boldsymbol{t} \end{bmatrix}_{2\times 3}$	3	lengths $+ \cdots$	
similarity	$\begin{bmatrix} s\boldsymbol{R} & \vert & \boldsymbol{t} \end{bmatrix}_{2\times 3}$	4	angles $+ \cdots$	
affine	$\begin{bmatrix} \boldsymbol{A} \end{bmatrix}_{2\times 3}$	6	parallelism $+ \cdots$	
projective	$\begin{bmatrix} \tilde{\boldsymbol{H}} \end{bmatrix}_{3\times 3}$	8	straight lines	

Mesh-Based Warping

How to Average a Dog?

- What to do?
  - Cross-dissolve doesn’t work
  - Global alignment doesn’t work
- Cannot be done with a global transformation (e.g. projective)
- Feature matching?
  - Nose to nose, tail to tail, etc.
  - This is a local warp

Idea: Warp First, Then Cross-Dissolve



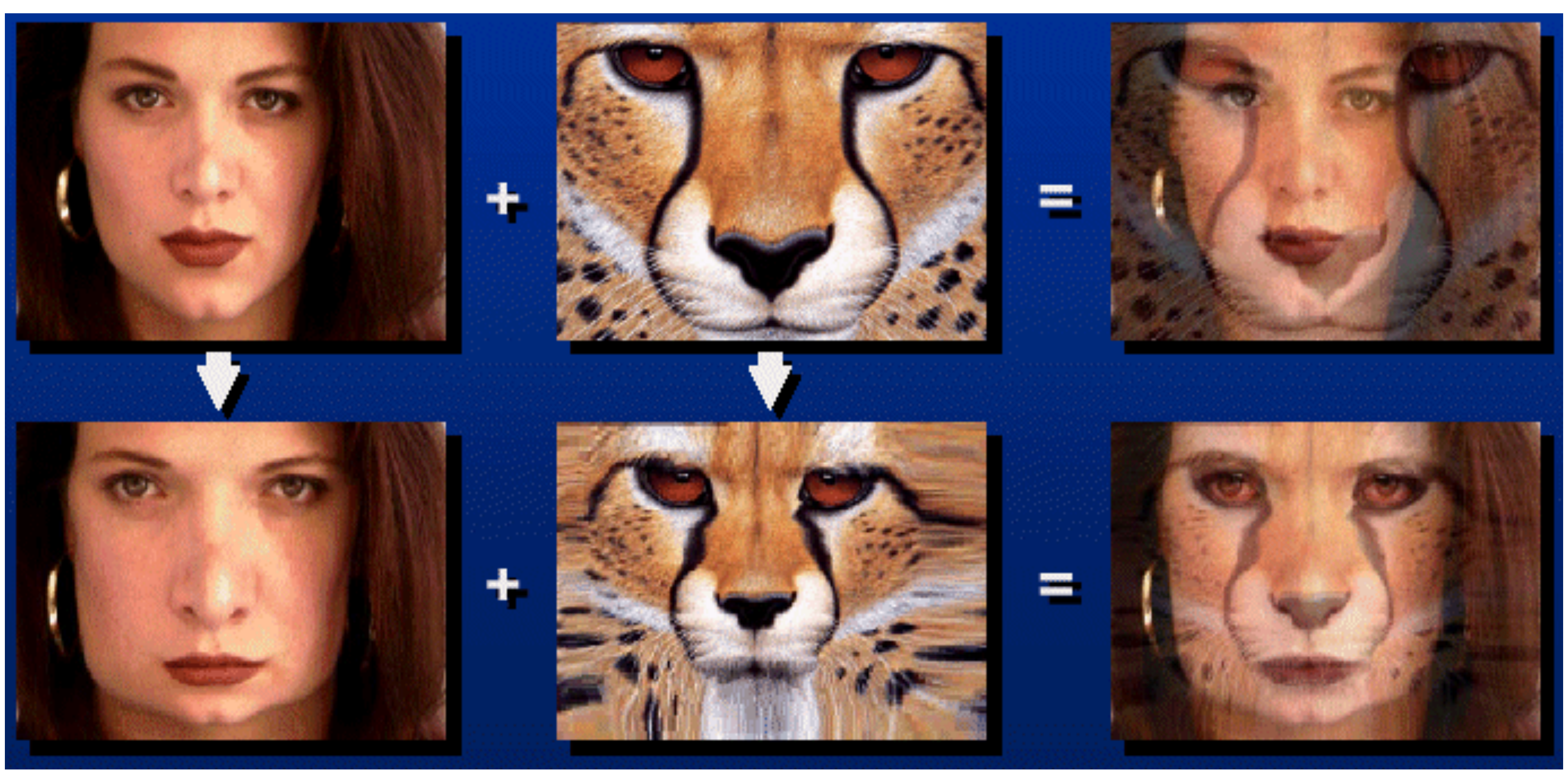
Morphing procedure:

For every intermediate step t,

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

Morphing Sequences: Summary

- If we know how to warp one image into the other, 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

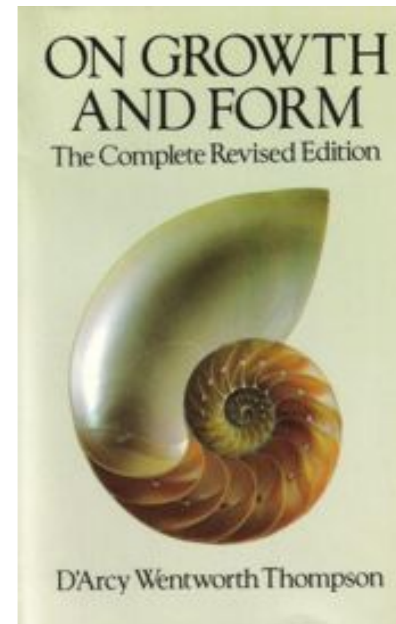


Local Warping

- Need to specify a more detailed warp function
- Global warps were functions of a few (2,4,8) parameters
- Non-parametric warps $u(x,y)$ and $v(x,y)$ can be defined independently for every single location x,y!
- Once we know vector field u,v we can easily warp each pixel (use backward warping with interpolation)

Image Warping in Biology

- D'Arcy Thompson
- http://www-groups.dcs.st-and.ac.uk/~history/Miscellaneous/darcy.html
- http://en.wikipedia.org/wiki/D%27Arcy_Wentworth_Thompson
- Importance of shape and structure in evolution



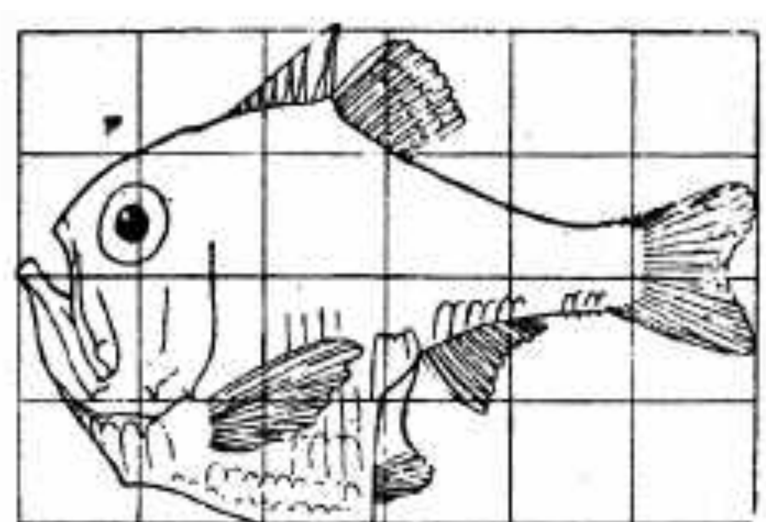


Fig. 517. *Argyropelecus Olfersi.*

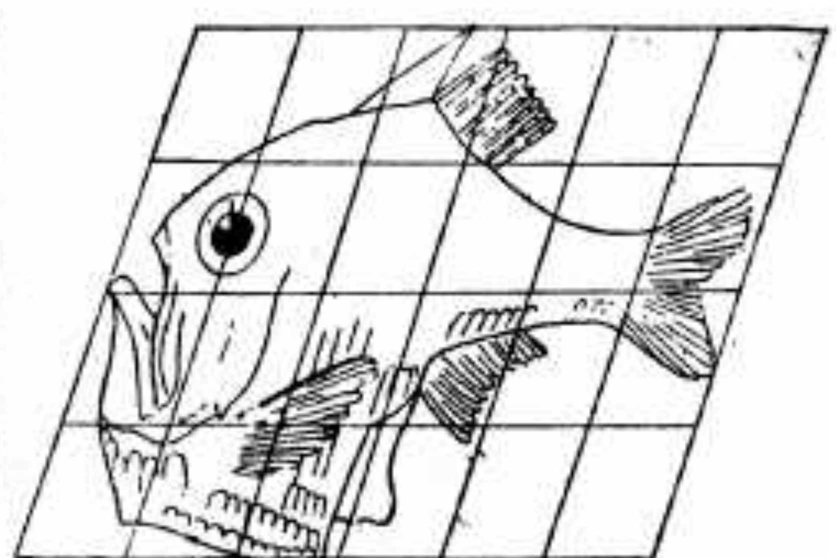
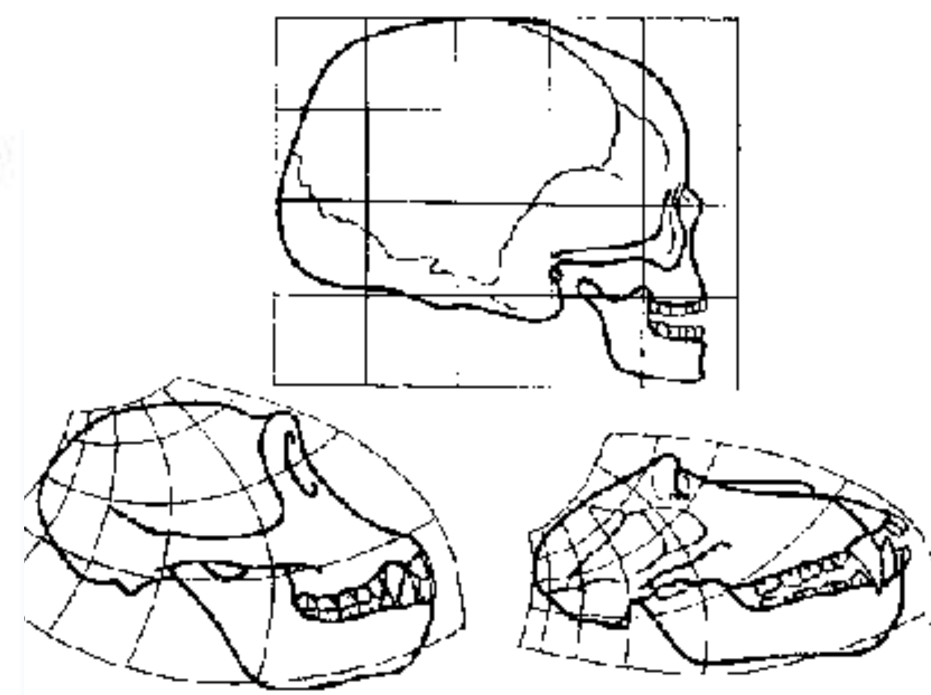
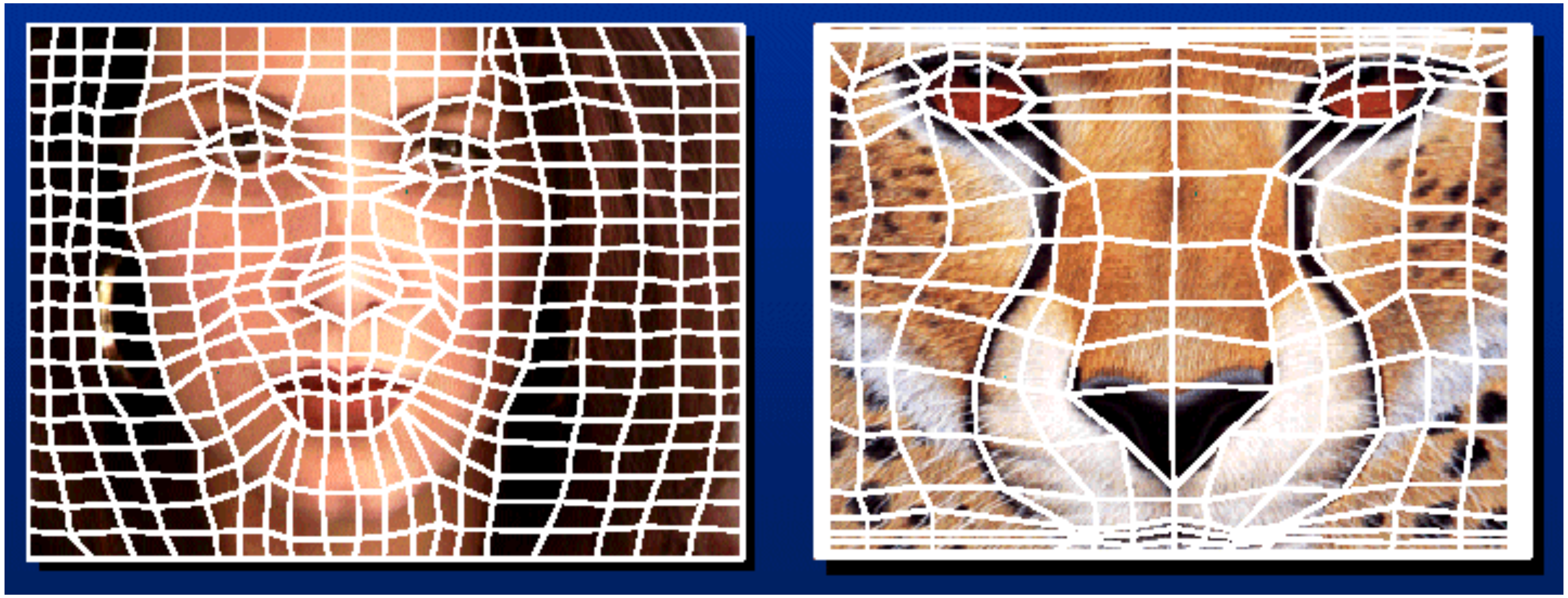


Fig. 518. *Sternoptyx diaphana.*



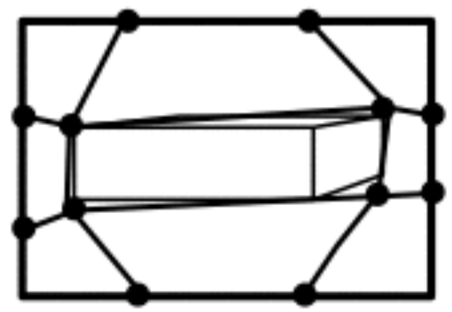
Skulls of a human, a chimpanzee and a baboon and transformations between them

Mesh-Based Warping

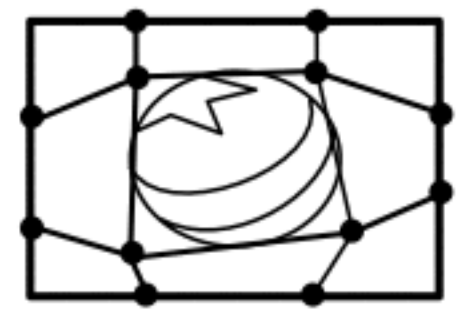


- Specify a grid on top of the image
- Deform vertices of the grid
- Treat each quadrilateral in the grid separately:
  - Interpolate the positions of vertices for each time step
  - Interpolate colors of before and after pixels

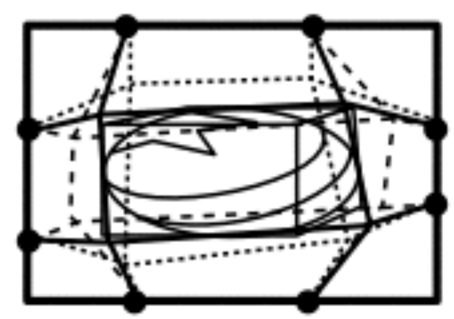
Mesh-Based Warping



starting mesh



ending mesh

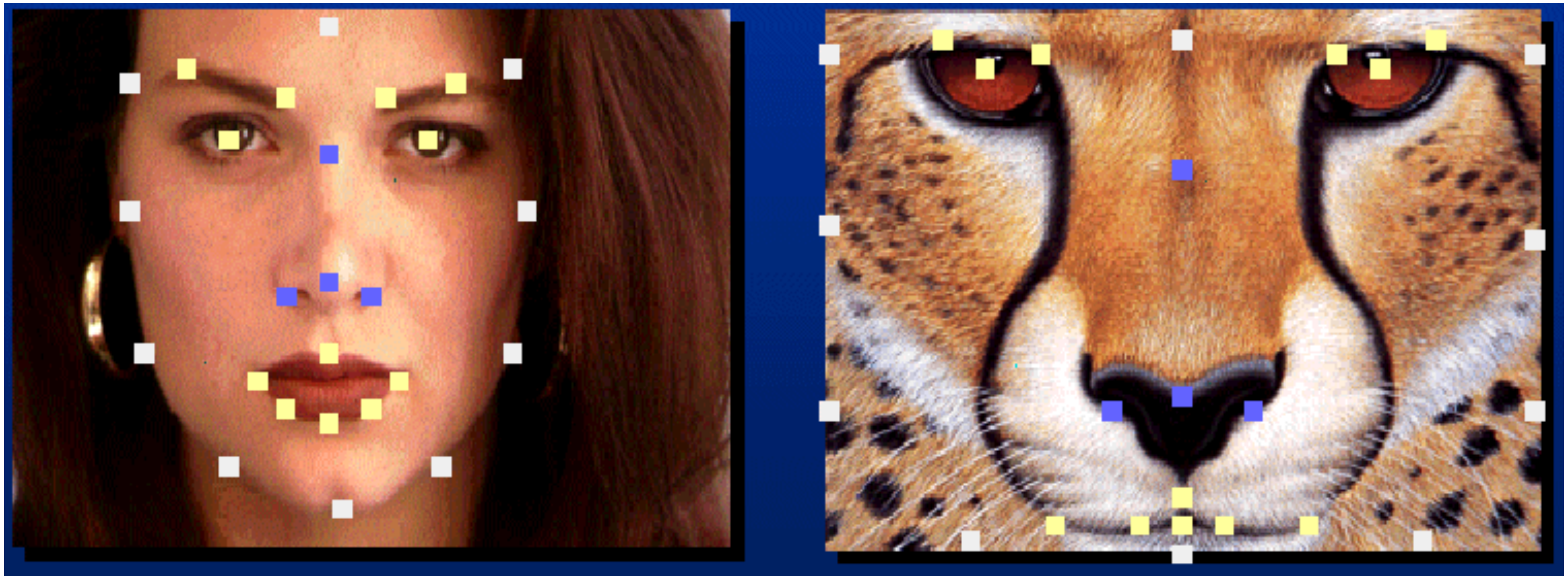


intermediate mesh for frame in middle of sequence

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

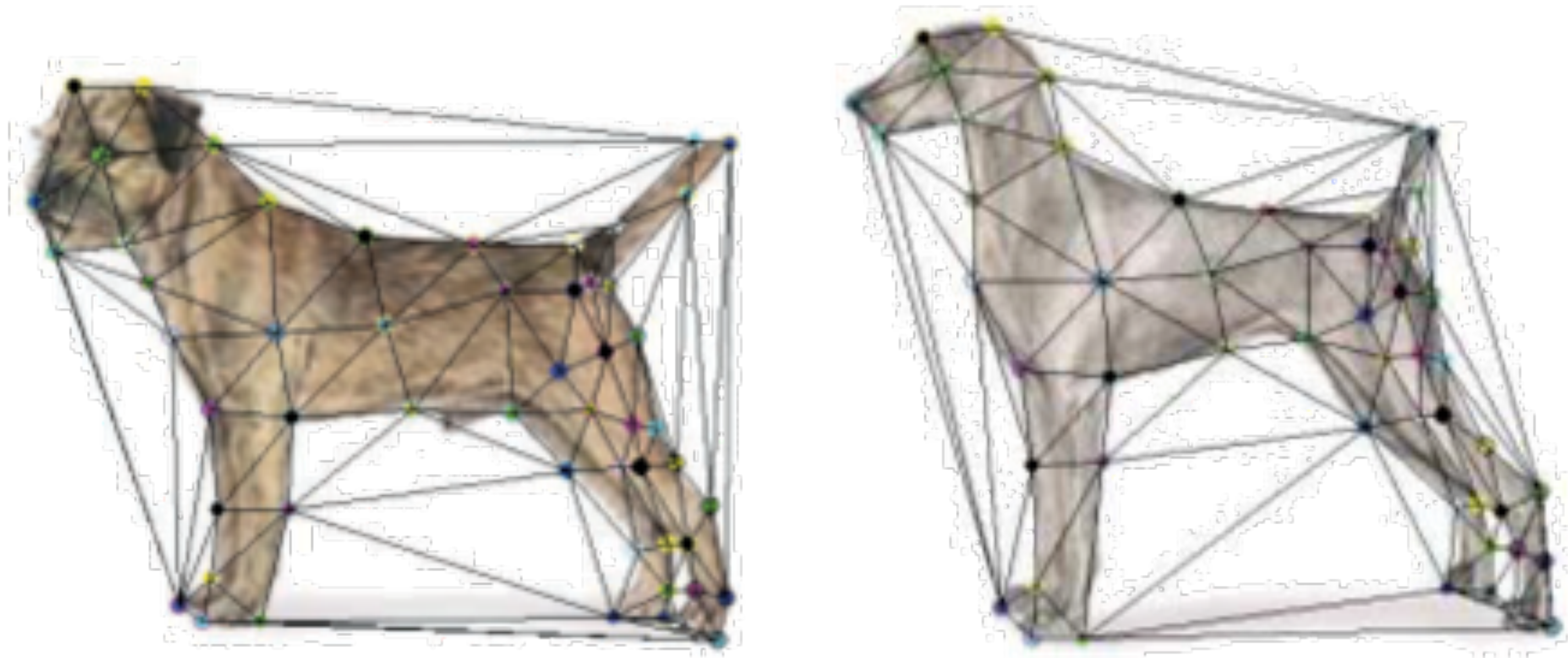
Triangle-Mesh Based Warping

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

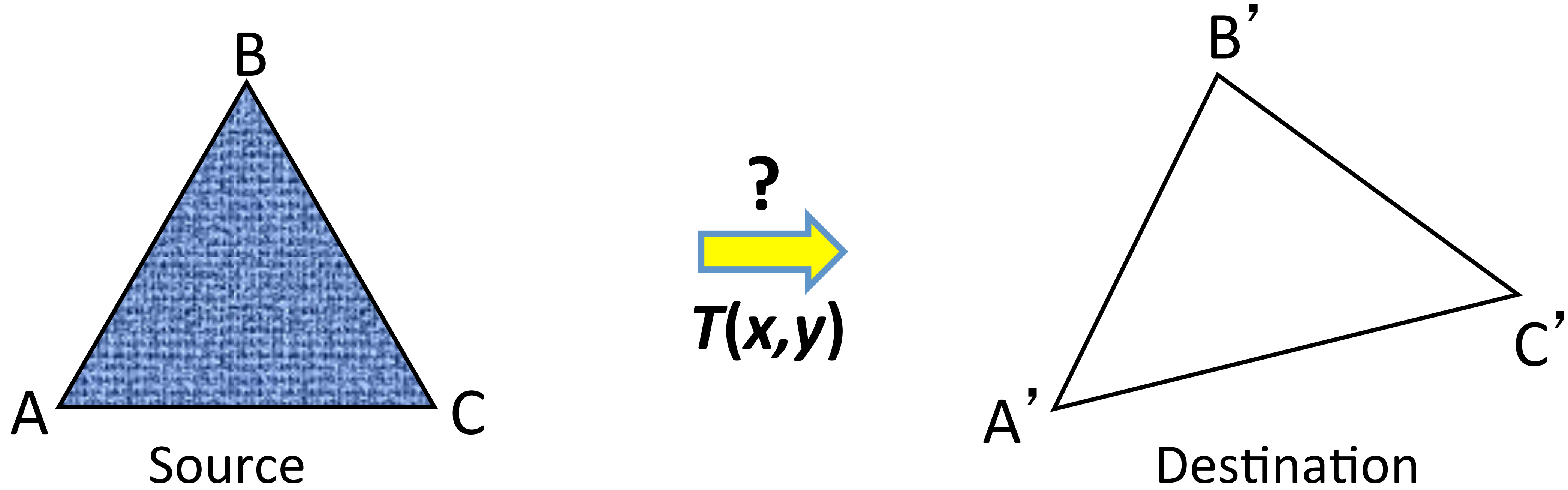


1. Input correspondences at key feature points

2. Define a triangular mesh over the points
   - Same mesh in both images!
   - Now we have triangle-to-triangle correspondences

3. Warp each triangle separately from source to destination
   - Warping a triangle means using 3 correspondences = affine warp!
   - Just like texture mapping

How to Warp a Triangle



- 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!

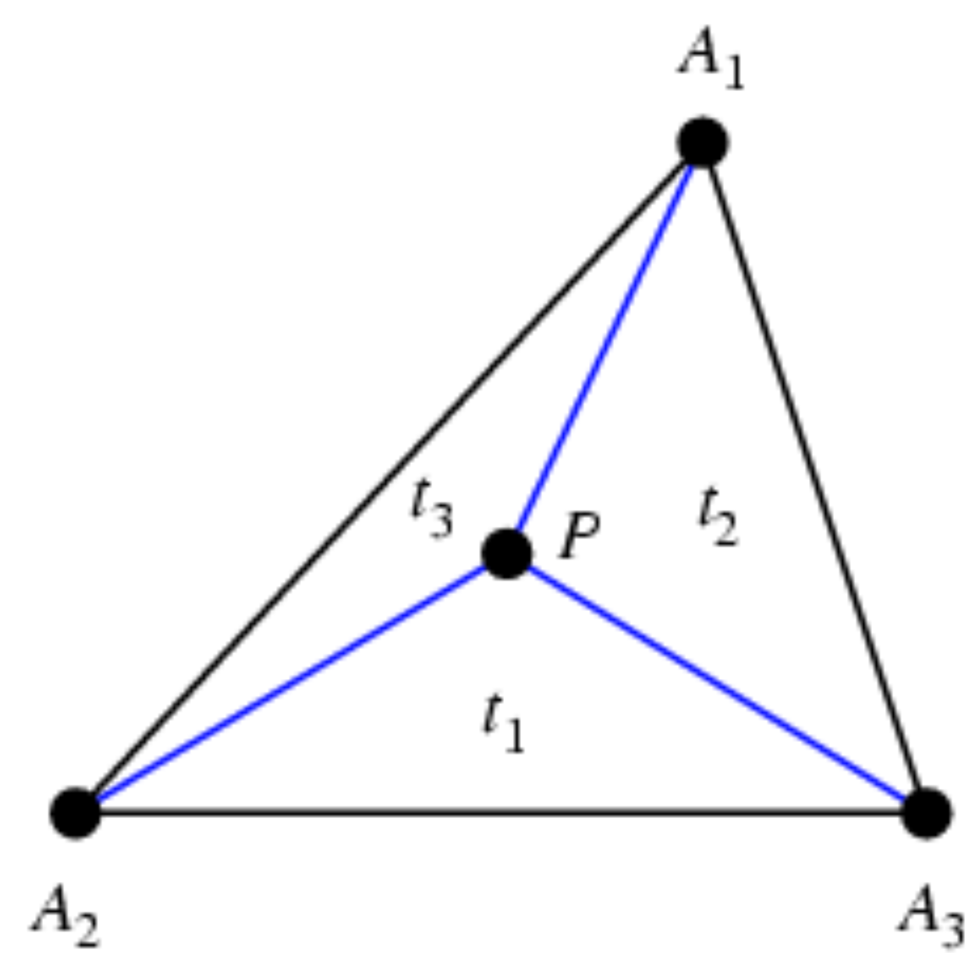
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

How to Warp a Triangle, Option #2

- Use barycentric interpolation
- Each point P is an area-weighted average of the vertices of the triangles
- Not unlike the strategy of bilinear interpolation of quads for warping

$$P = t_1 A_1 + t_2 A_2 + t_3 A_3$$

$$t_1 + t_2 + t_3 = 1$$

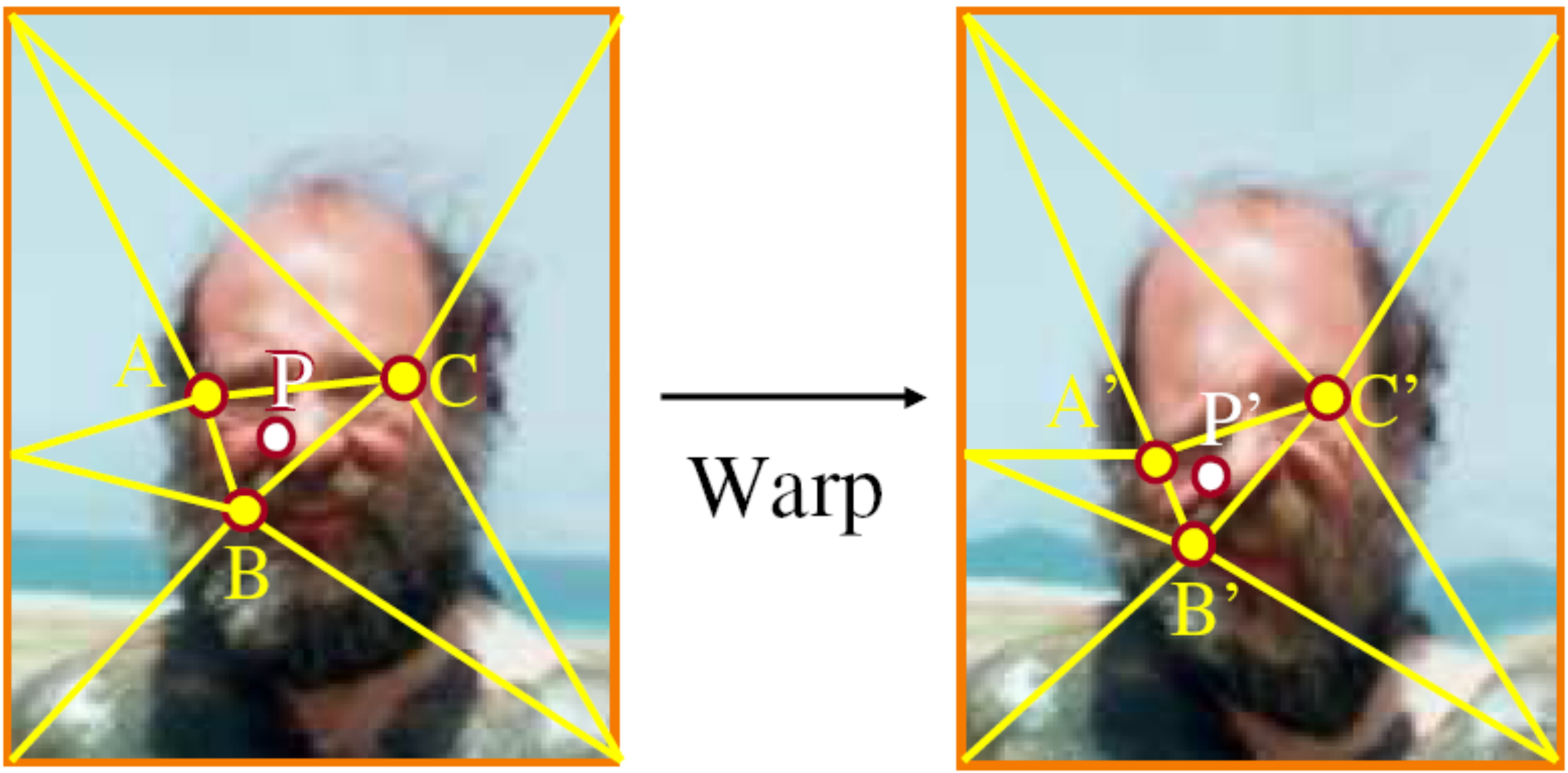


Example: Triangle Warping

$$P = w_A A + w_B B + w_C C$$

$$P' = w_A A' + w_B B' + w_C C'$$

Barycentric coordinate



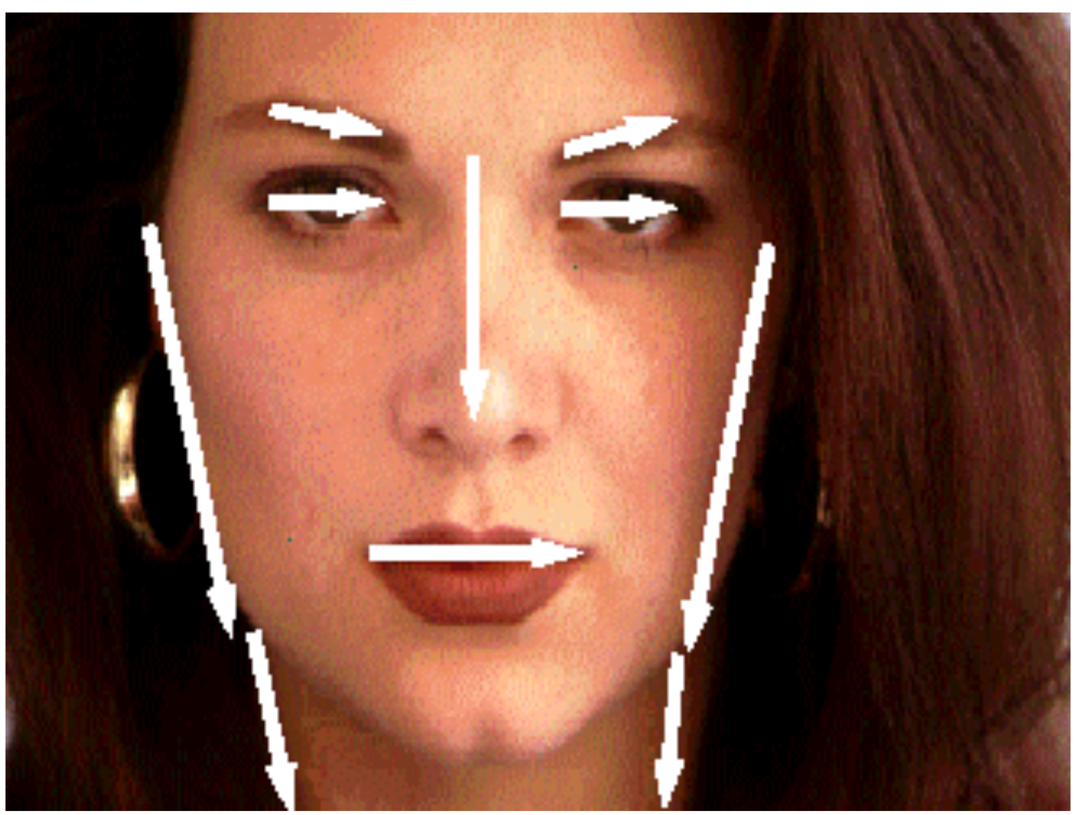
Segment-Based Warping

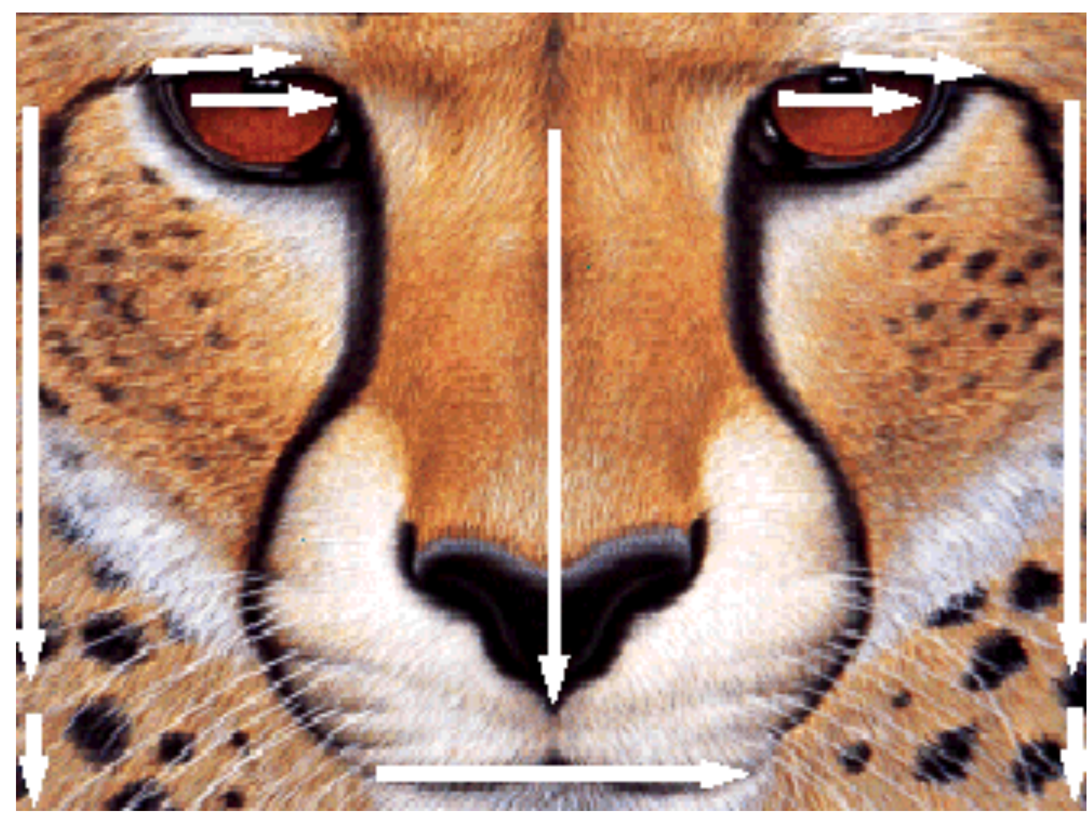
Michael Jackson’s Black or White

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

The Beier & Neely Algorithm

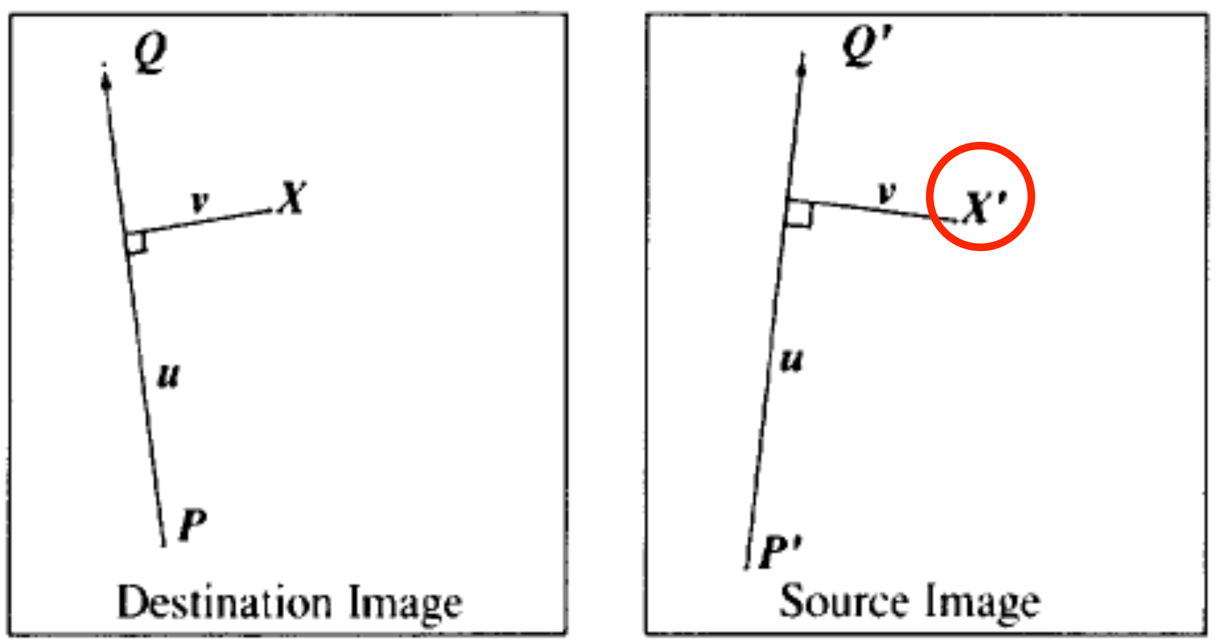
- Specify the warp by specifying corresponding vectors
- Interpolate to a complete warping function





How Line Segments Specify Points

- Given PQ and P’Q’, they define a warp for a point X to a point X’
- Measure distance u along the segment
- The direction of PQ defines a side, measure distance v from the right side



$$u = \frac{(X - P) \cdot (Q - P)}{\| Q - P \|^2} \qquad (1)$$

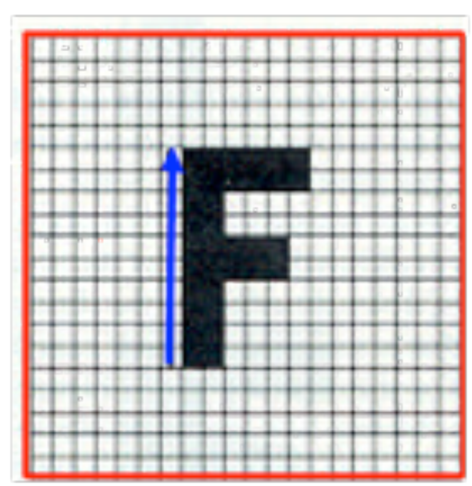
$$v = \frac{(X - P) \cdot Perpendicular\,(Q - P)}{\| Q - P \|} \qquad (2)$$

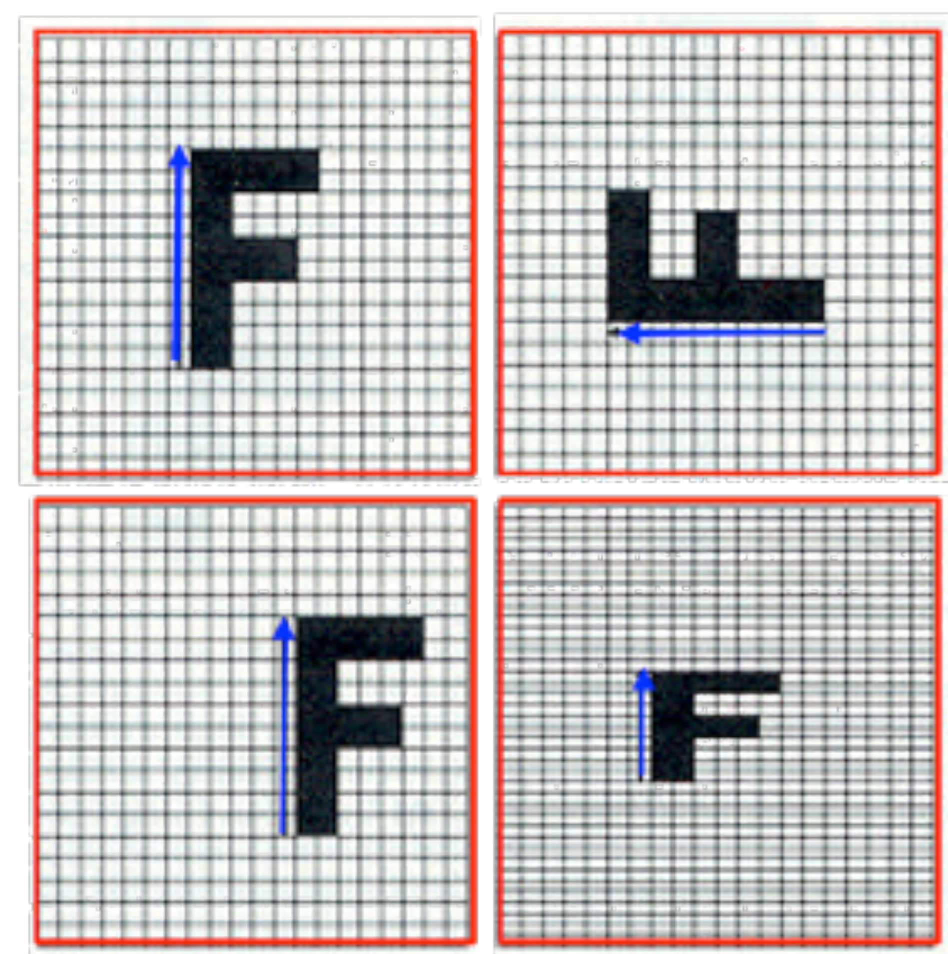
$$X' = P' + u \cdot (Q' - P') + \frac{v \cdot Perpendicular\,(Q' - P')}{\| Q' - P' \|} \qquad (3)$$

One Line Segment (Globally) Warping an Image

- For each X in the destination image:
  1. Find the corresponding u,v
  2. Find X’ in the source image for that u,v
  3. OUT(X) = IN(X’)

Examples





Each of these is an Euclidean warp, why?

Lec22 Required Reading

- Szeliski, Ch. 9