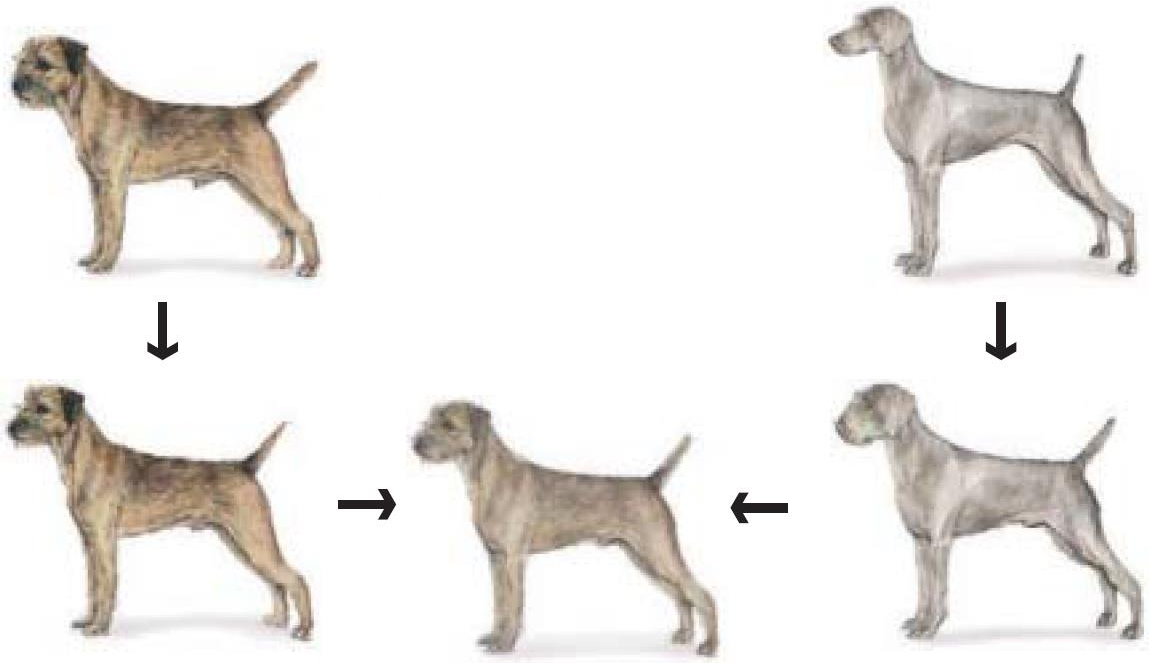
Idea #3: Local warp, then cross-dissolve



Morphing procedure:

for every t,

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

Image Morphing, ***select features***

1) Generating correspondence points (by hand)
    1) In matlab, use “cpselect” function, or write your own with “ginput” + “plot” (with “hold on”, “hold off”)

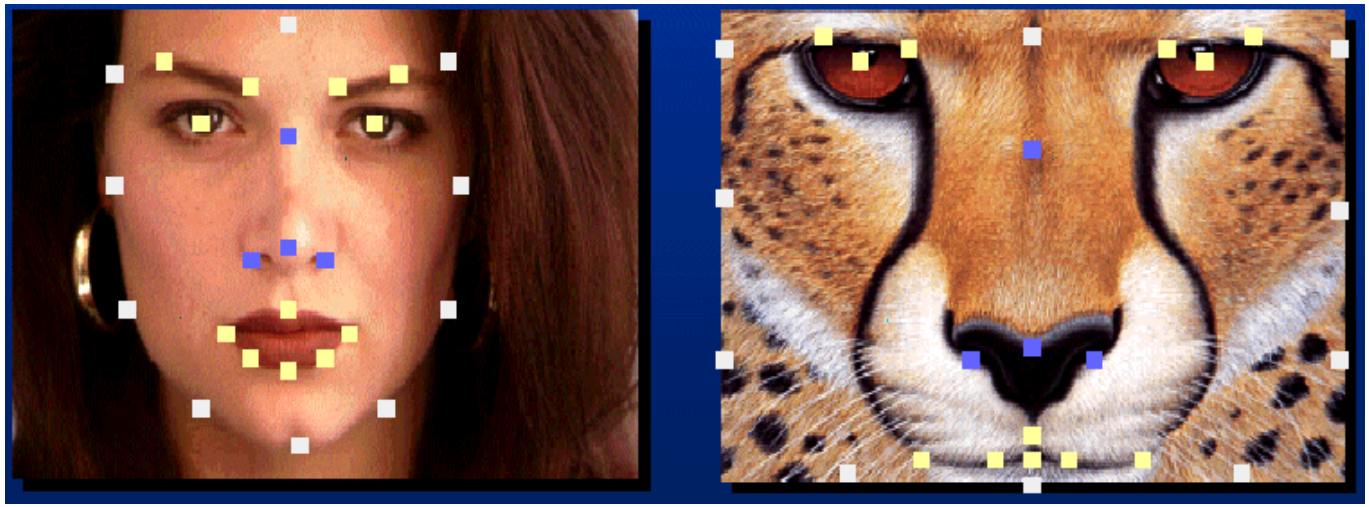
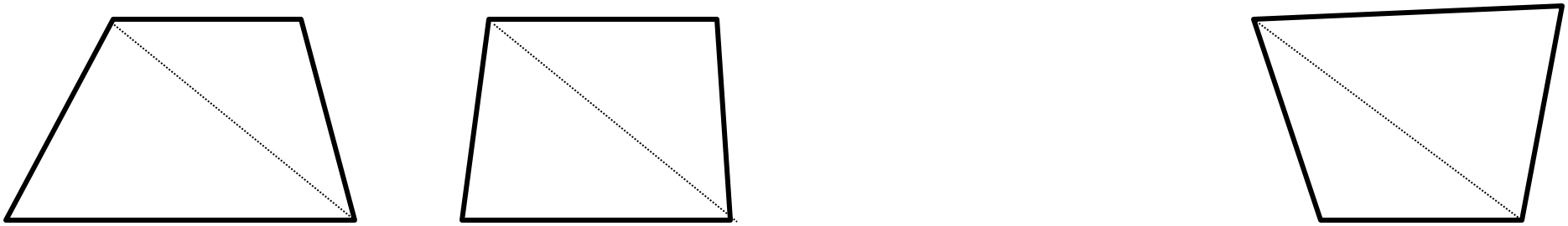


Image Morphing, generating ***intermediate shape*** (t-average)

2) Compute a weighted average shape

- Assume t = [0,1]
- Simple linear interpolation of each feature pair
- (1-t)*p1+t*p0 for corresponding features p0 and p1

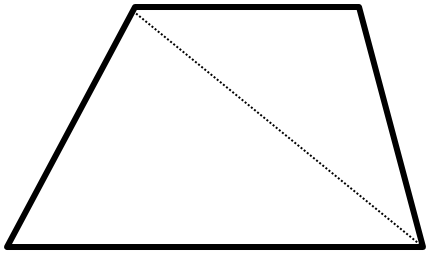


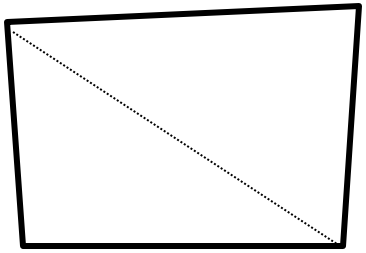
t=0.4

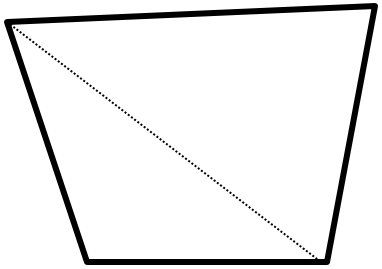
Image Morphing, generating ***intermediate shape***

2) Compute a weighted average shape

- Assume t = [0,1]
- Simple linear interpolation of each feature pair
- (1-t)*p1+t*p0 for corresponding features p0 and p1

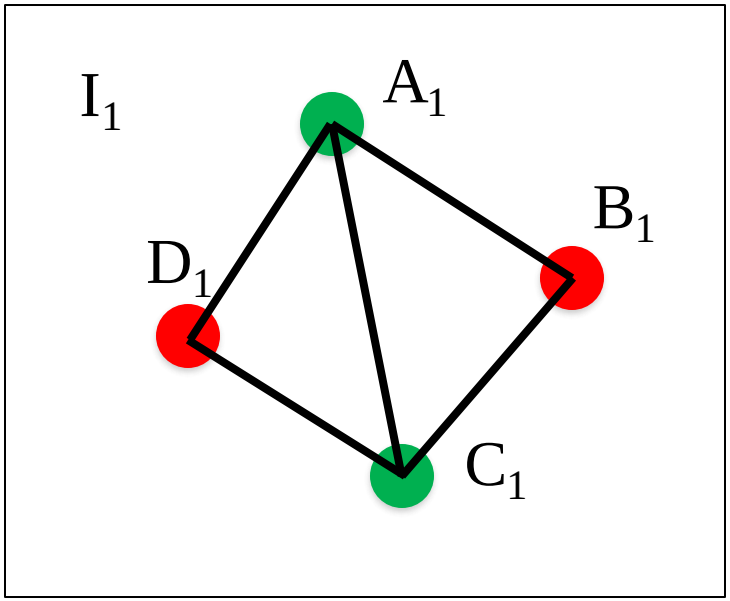


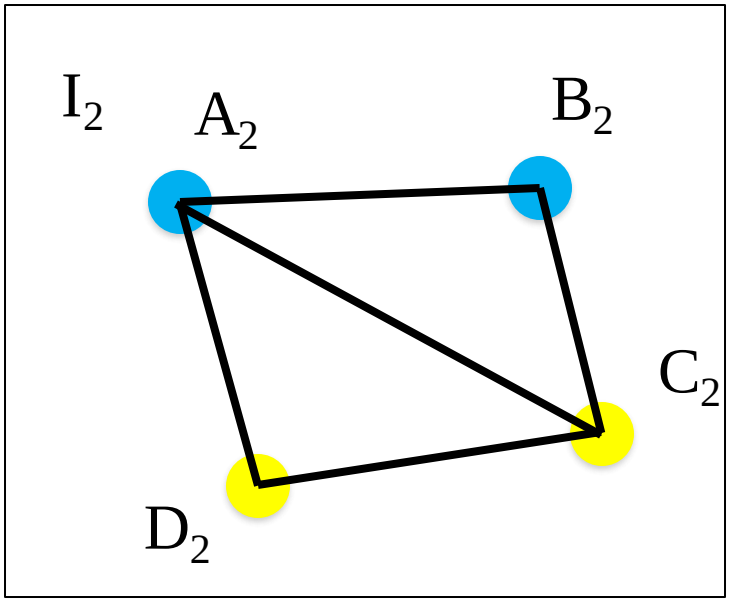




t=0.7

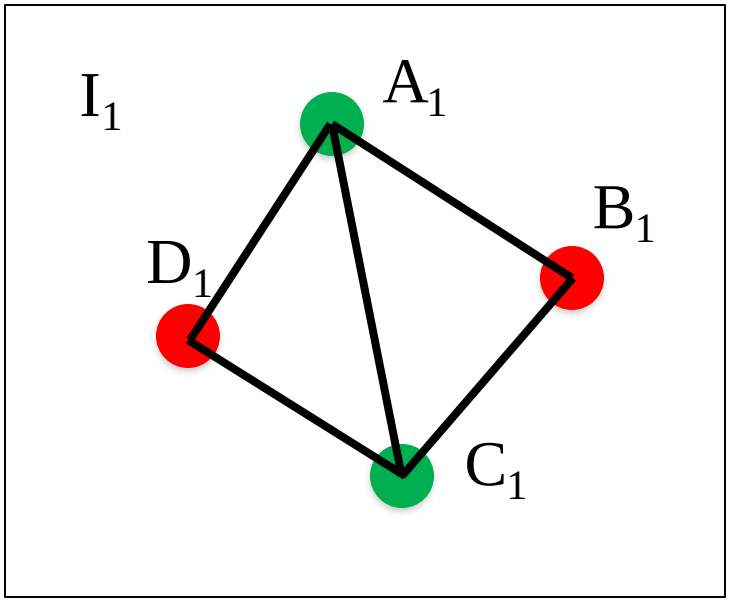
Image Morphing

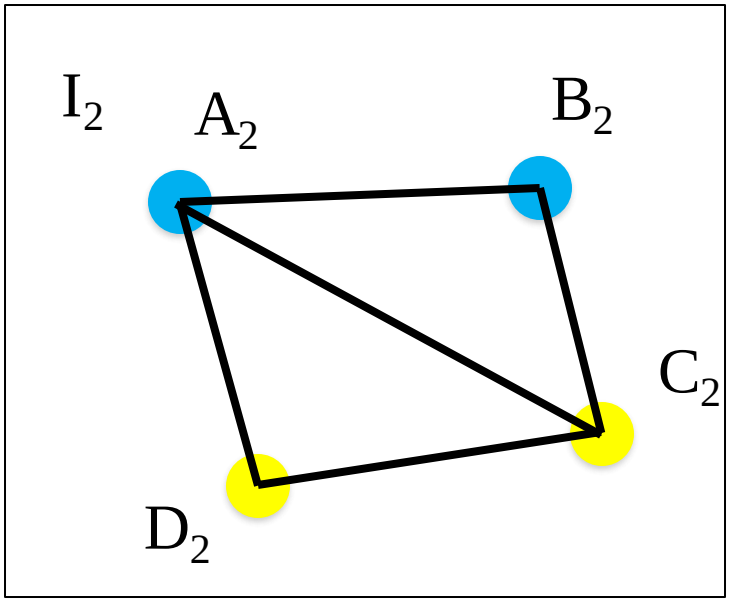


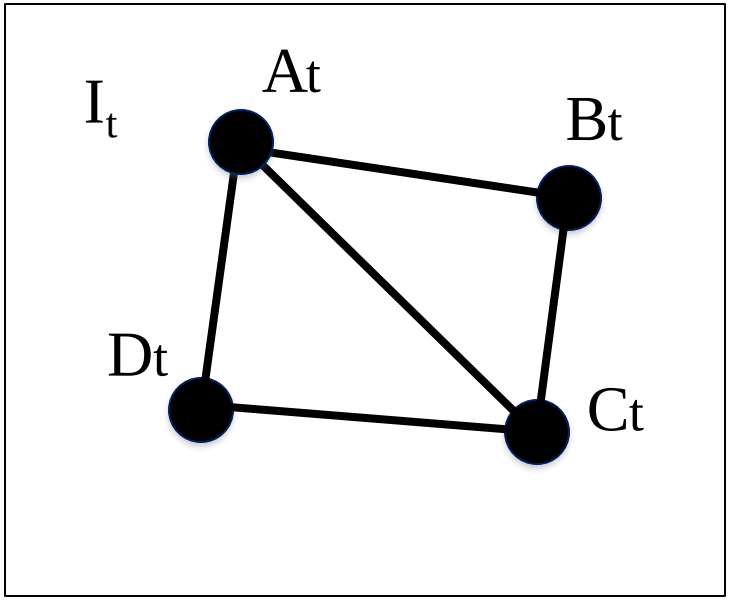


- Corresponding points:
  - A1 – A2, B1 – B2, C1 – C2, D1 – D2
- Step1: Create an intermediate shape (by interpolation)
- Step2: Warp both images towards the shape
- Step3: Cross-dissolve the color

Image Morphing: Intermediate Shape







$$At = tA1 + (1-t)A2$$
$$Bt = tB1 + (1-t)B2$$

$$Ct = tC1 + (1-t)C2$$

Image Morphing: Warping

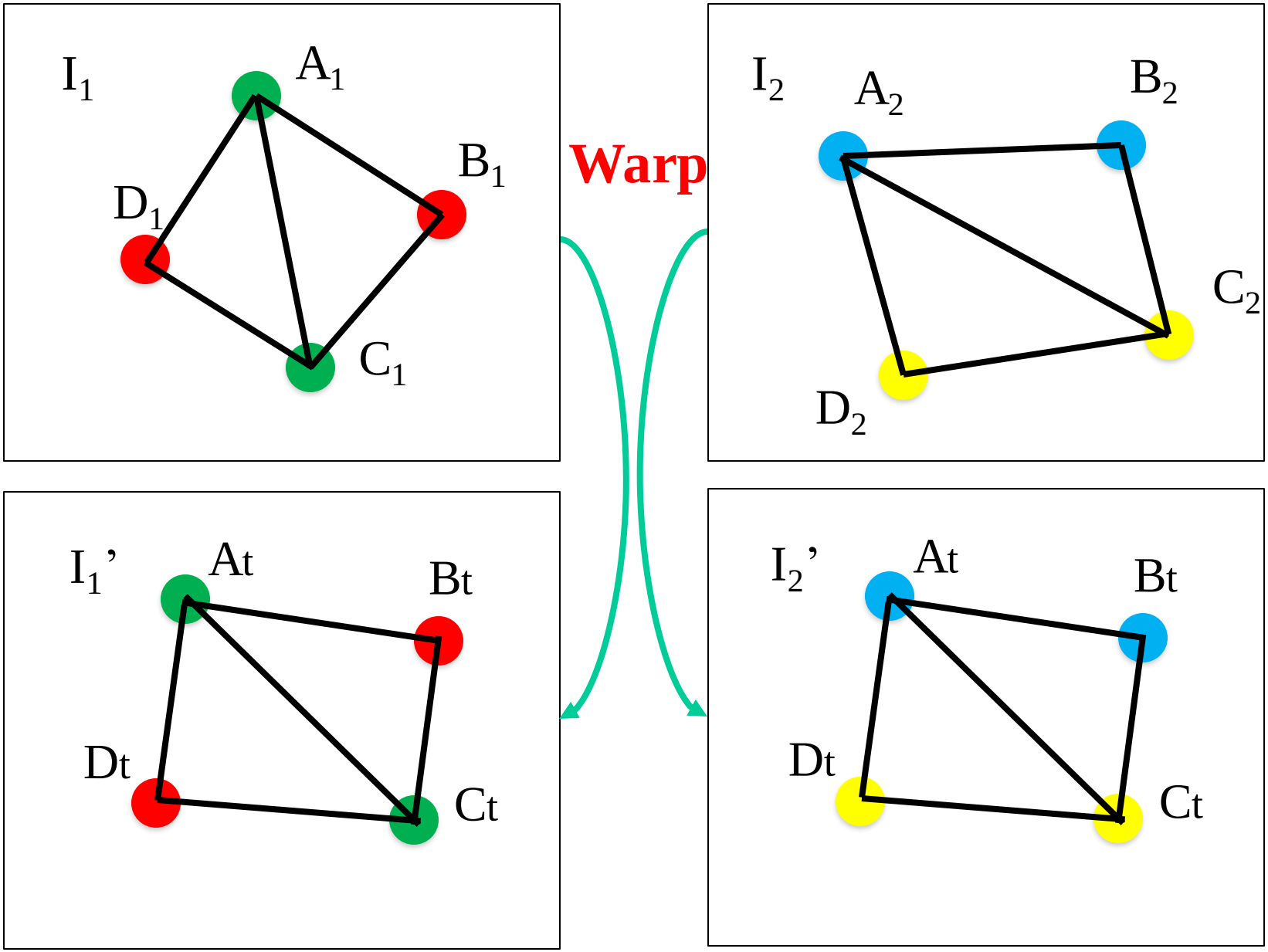
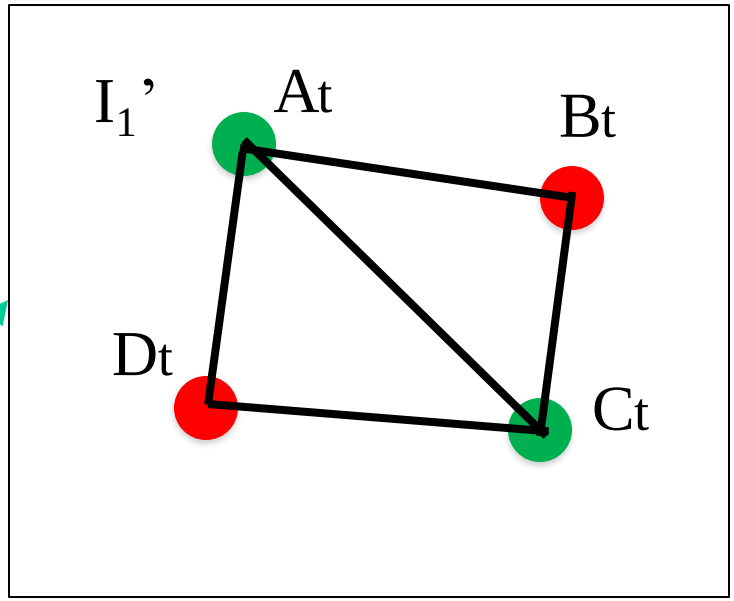
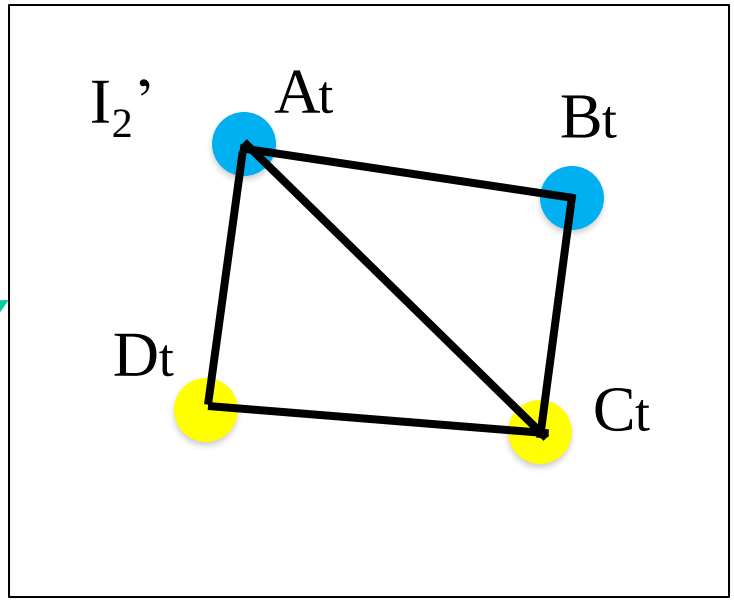


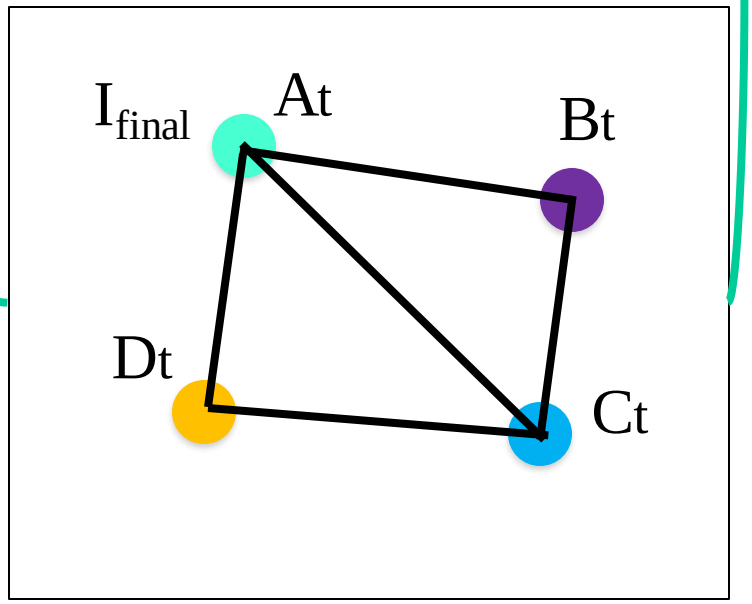
Image Morphing: Cross Dissolve Colors





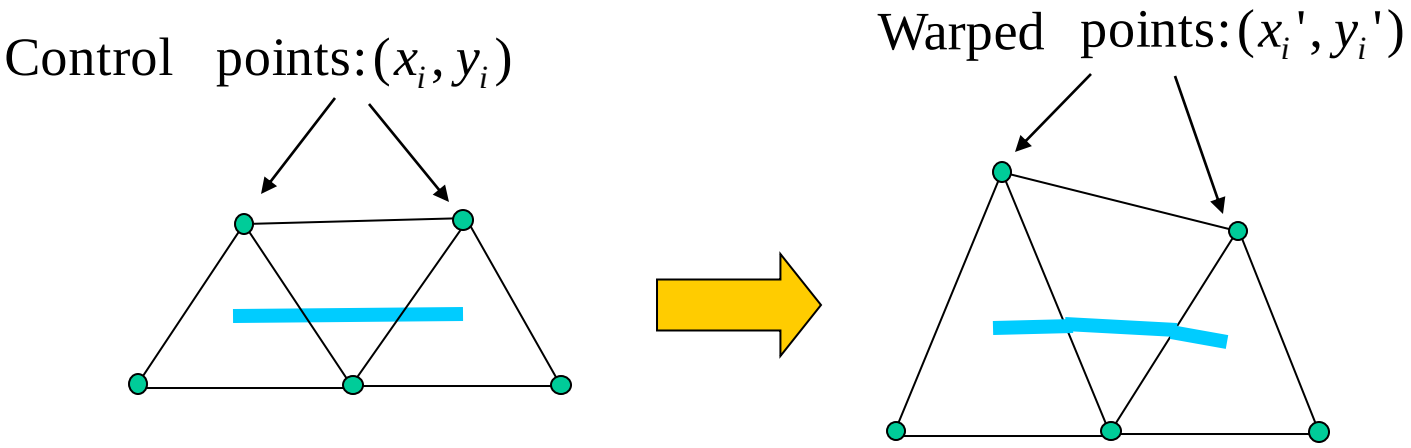
T^{-1}

T^{-1}



- Cross-dissolve the colors by **inverse triangle warping**
  - A_t: Cyan = Green + Blue
  - B_t: Purple = Red + Blue
  - C_t: Blue = Green + Yellow
  - Dt: Orange = Red + Yellow

Interpolation using Triangles

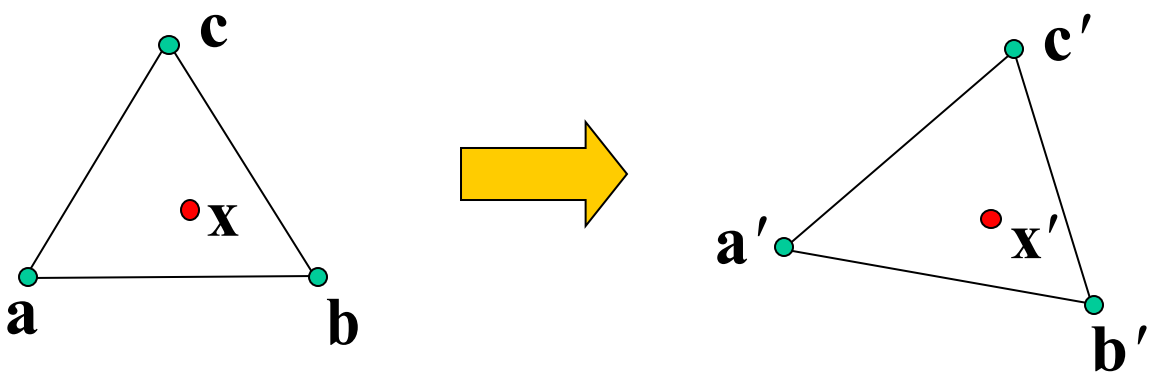


Region of interest enclosed by triangles.

Moving nodes changes each triangle

Just need to map regions between two triangles

Barycentric Co-ordinates



$$\mathbf{x} = \alpha\mathbf{a} + \beta\mathbf{b} + \gamma\mathbf{c}$$

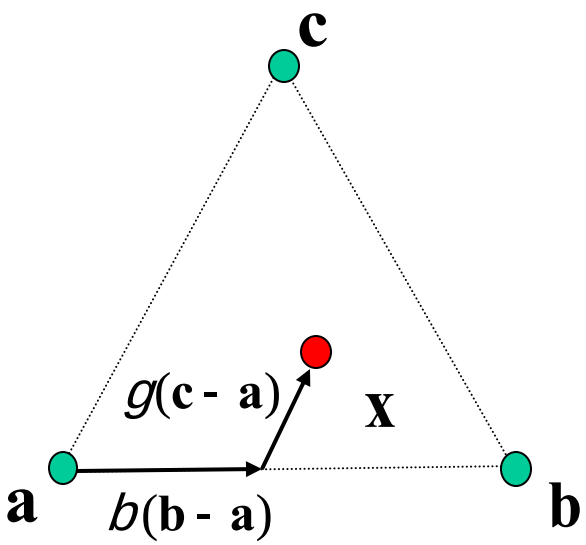
$$\alpha + \beta + \gamma = 1$$

$$\mathbf{x}' = \alpha\mathbf{a}' + \beta\mathbf{b}' + \gamma\mathbf{c}'$$

How do we know if a point is inside of a triangle?

$\mathbf{x}$ is inside the triangle if $0 \le \acute{\alpha} \le 1$ and $0 \le \hat{\alpha} \le 1$

Barycentric Co-ordinates



$$\mathbf{x} = \mathbf{a} + \beta(\mathbf{b} - \mathbf{a}) + \gamma(\mathbf{c} - \mathbf{a})$$
$$= (1 - \beta - \gamma)\mathbf{a} + \beta\mathbf{b} + \gamma\mathbf{c}$$
$$= \alpha\mathbf{a} + \beta\mathbf{b} + \gamma\mathbf{c}$$

$$\mathbf{x} = \alpha\mathbf{a} + \beta\mathbf{b} + \gamma\mathbf{c}$$
$$\alpha + \beta + \gamma = 1$$

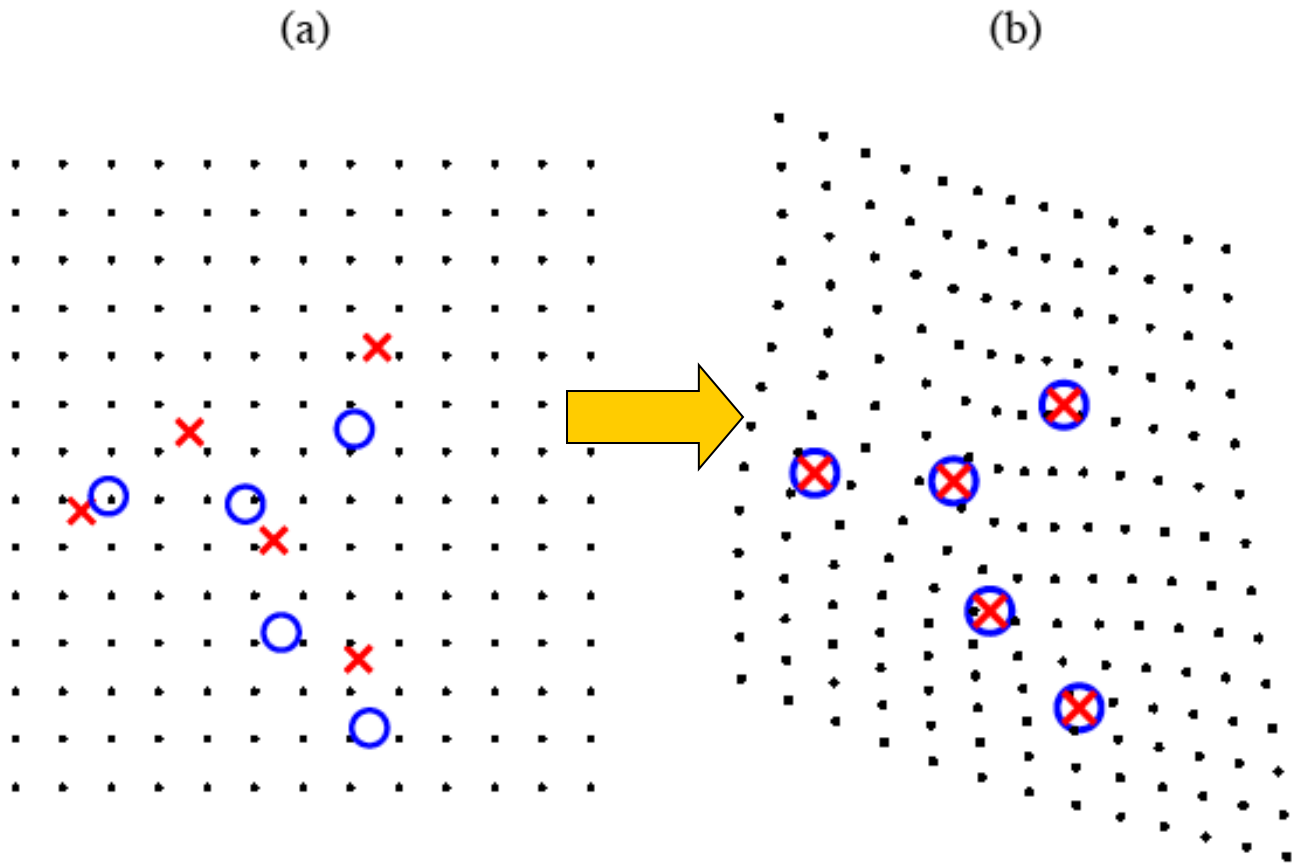
$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} a_x & b_x & c_x \\ a_y & b_y & c_y \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}$$

Three linear equations in 3 unknowns

Interpolation using Triangles

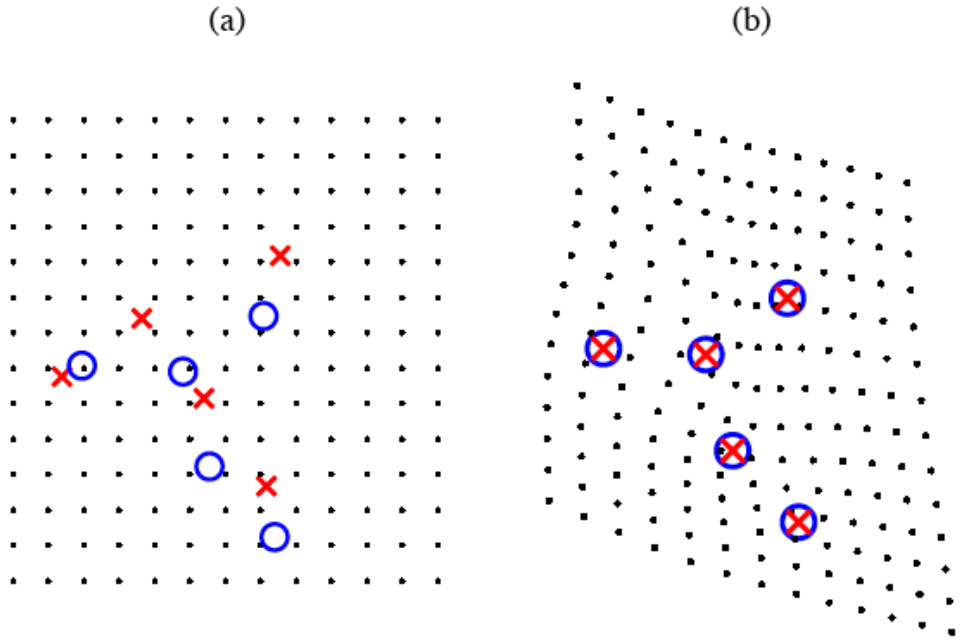
- To find out where each pixel in new image comes from in old image
  - Determine which triangle it is in
  - Compute its barycentric co-ordinates
  - Find equivalent point in equivalent triangle in original image
- Only well defined in region of `convex hull' of control points

Thin-plate spline TPS



Sparse and irregular positioned feature points, and smooth interpolation

Simple example of coordinate transformation using TPS



Let's consider two sets of points for which we assume the correspondences to be known (a). The TPS warping allows a perfect alignment of the points and the bending of the grid shows the deformation needed to bring the two sets on top of each other (b). Note that in the case of TPS applied to coordinate transformation we actually use two splines, one for the displacement in the x direction and one for the displacement in the y direction. The displacement in each direction is considered as a height map for the points and a spline is fit as in the case of scattered points in 3D space. And finally the two resulting transformations are combined into a single mapping.

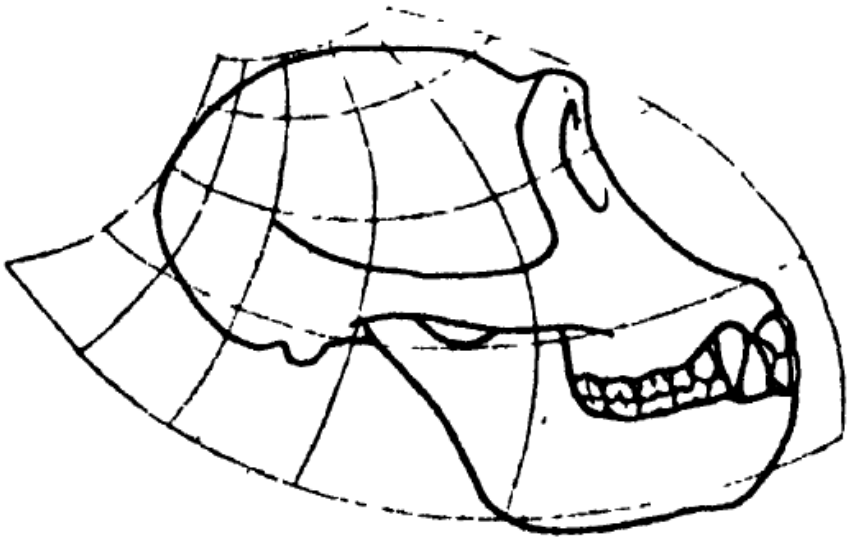


Fig. 550. Skull of chimpanzee.

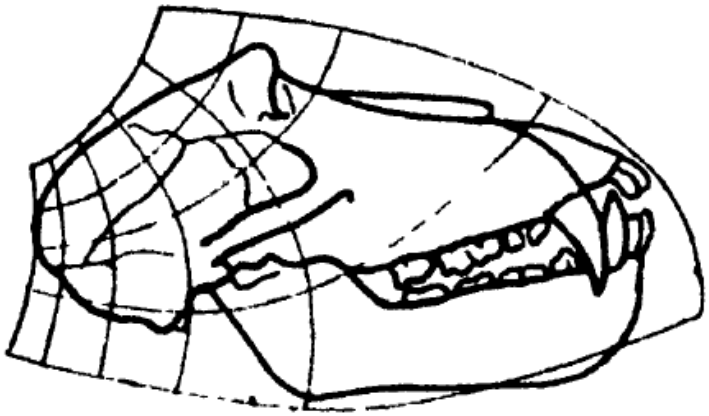


Fig. 551. Skull of baboon.

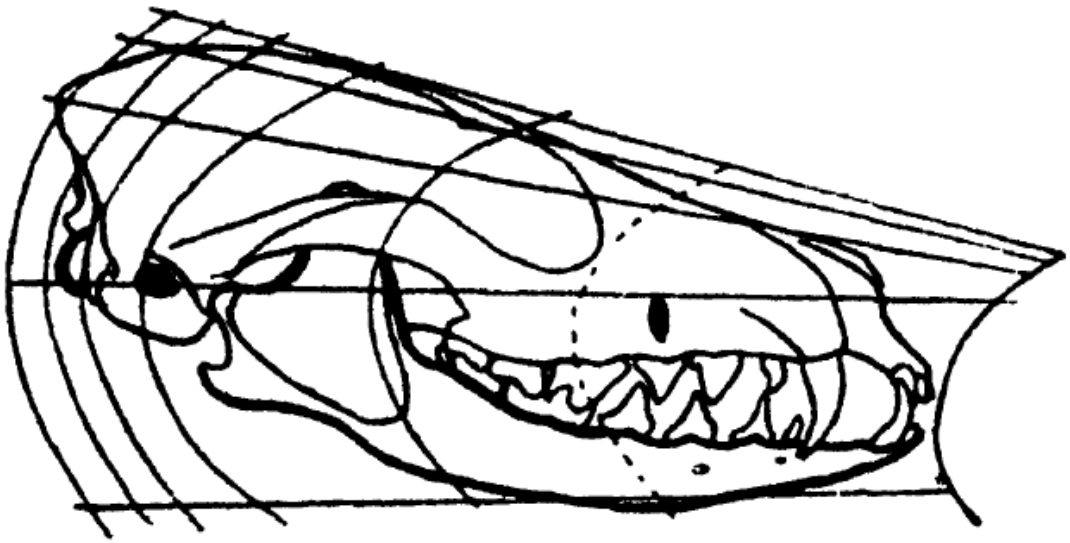


Fig. 552. Skull of dog, compared with the human skull of Fig. 548.

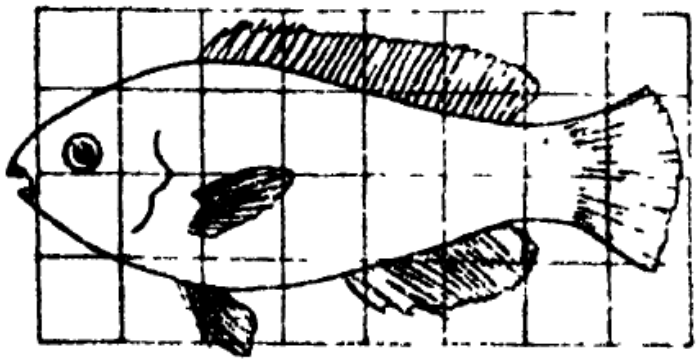


Fig. 519. *Scarus* sp.

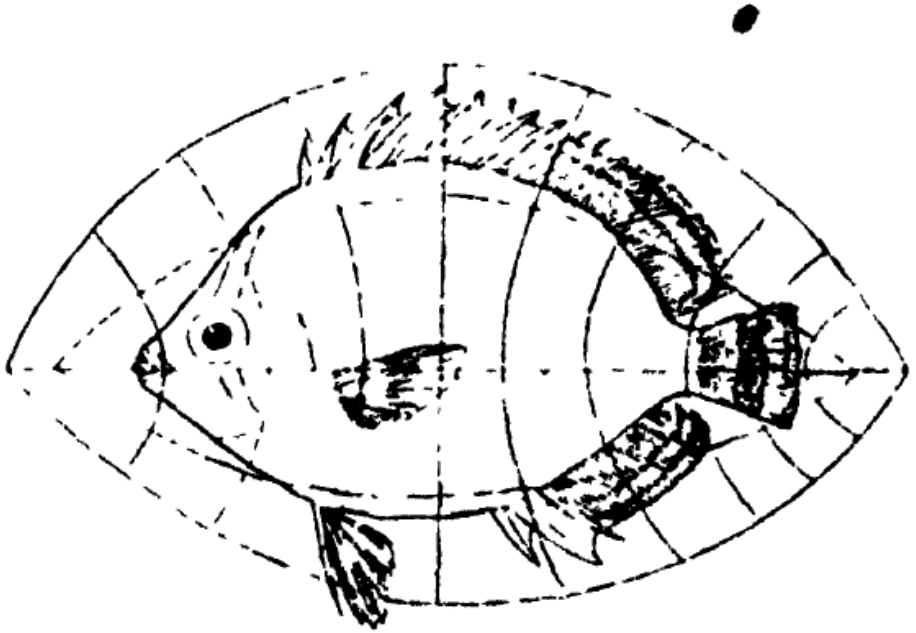
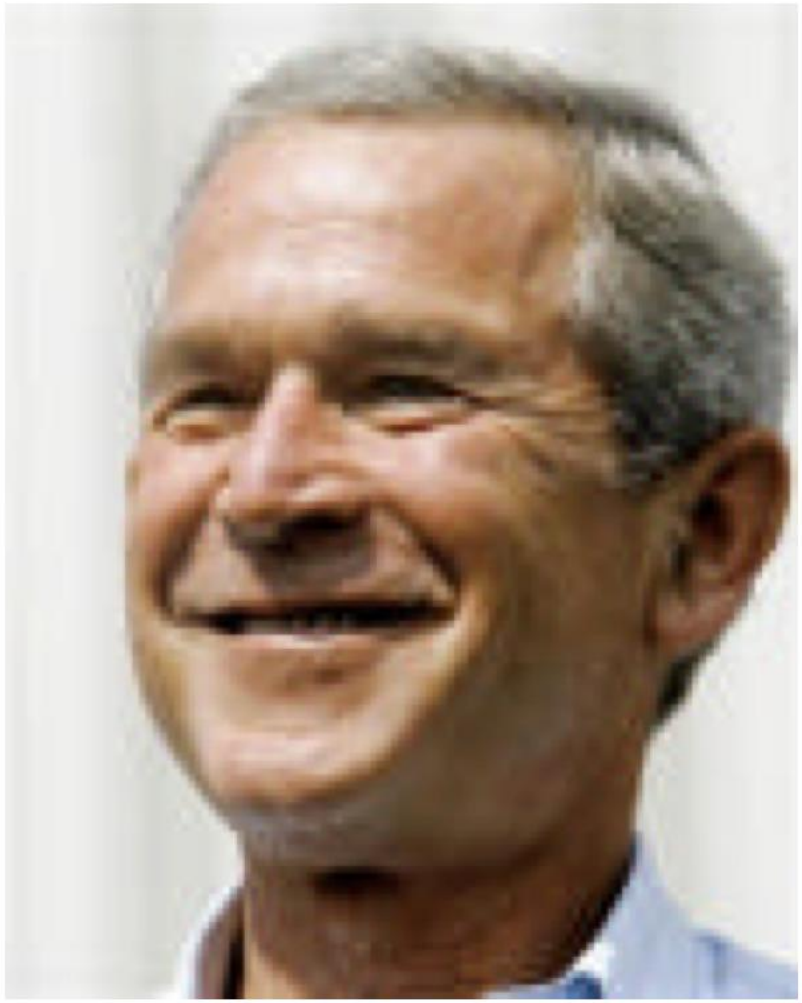


Fig. 520. *Pomacanthus.*

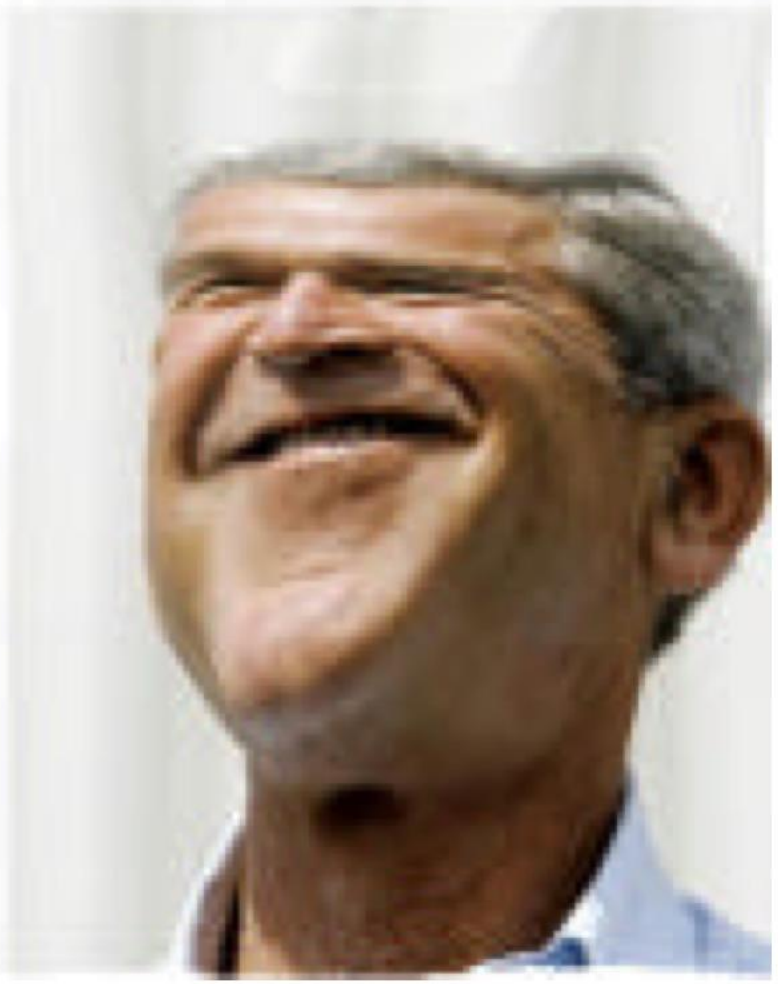
Original

Thin-Plate Splines

Whitbeck & Guo



Original



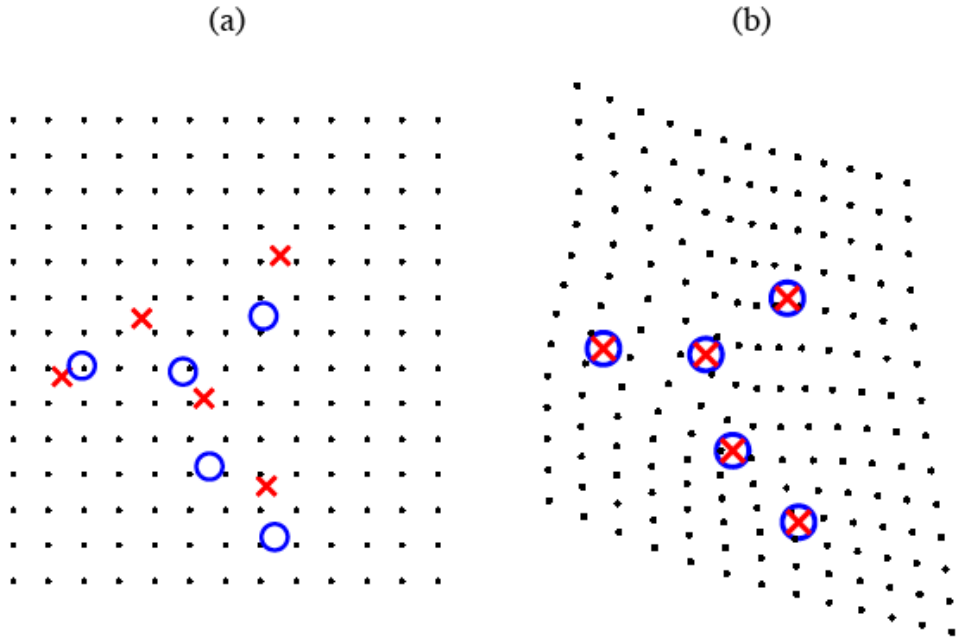
Thin-Plate Splines

Whitbeck & Guo

Original

Thin-Plate Splines

Whitbeck & Guo

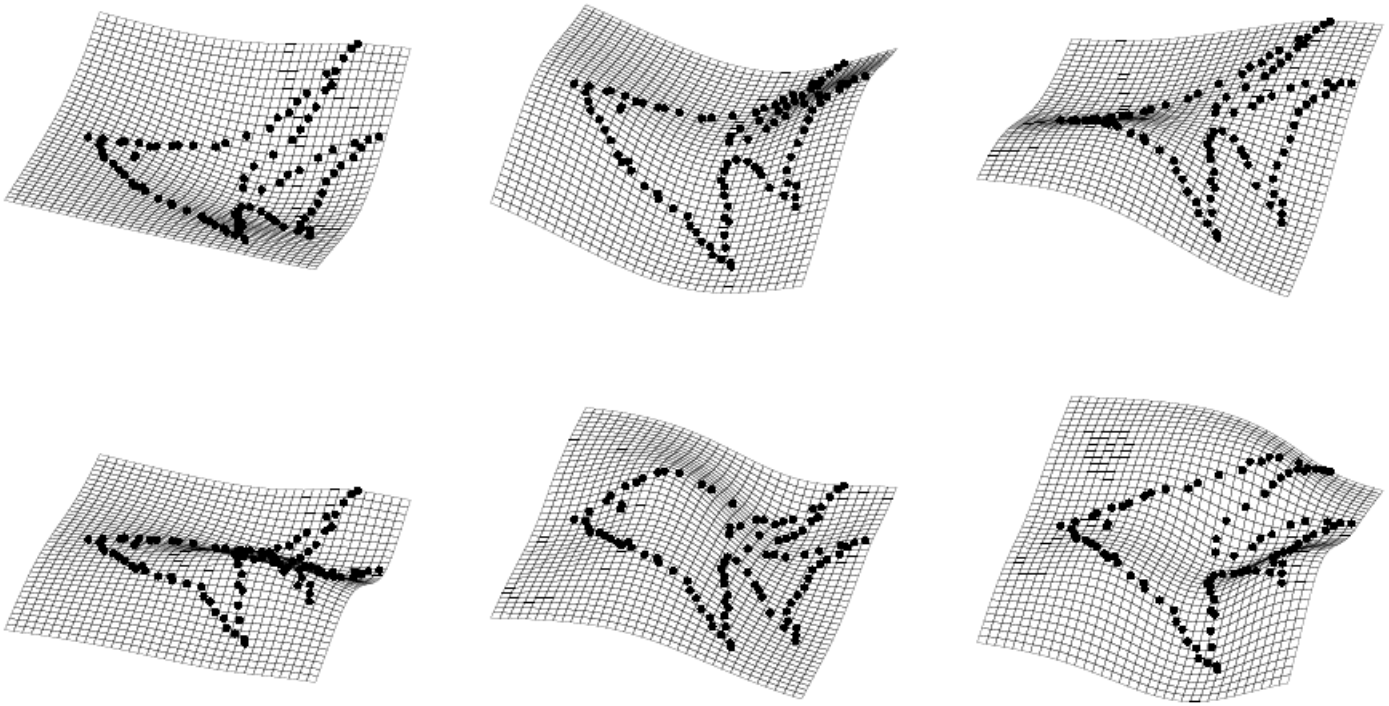


$$\begin{cases} \forall i \ f_x(x_i, y_i) = x'_i \\ f_x = \text{argmin}_{\ g} \left\{ I_g = \int\int_{\mathbb{R}^2} \left(\frac{\partial^2 g}{\partial x^2}\right)^2 + \left(\frac{\partial^2 g}{\partial x \partial y}\right)^2 + \left(\frac{\partial^2 g}{\partial y^2}\right)^2 \right\} \\ f_x(x, y) = v + v_x x + v_y y + \sum_{i=1}^{n} w_i U\left(\|(x_i, y_i) - (x, y)\|\right) \end{cases} \tag{17}$$

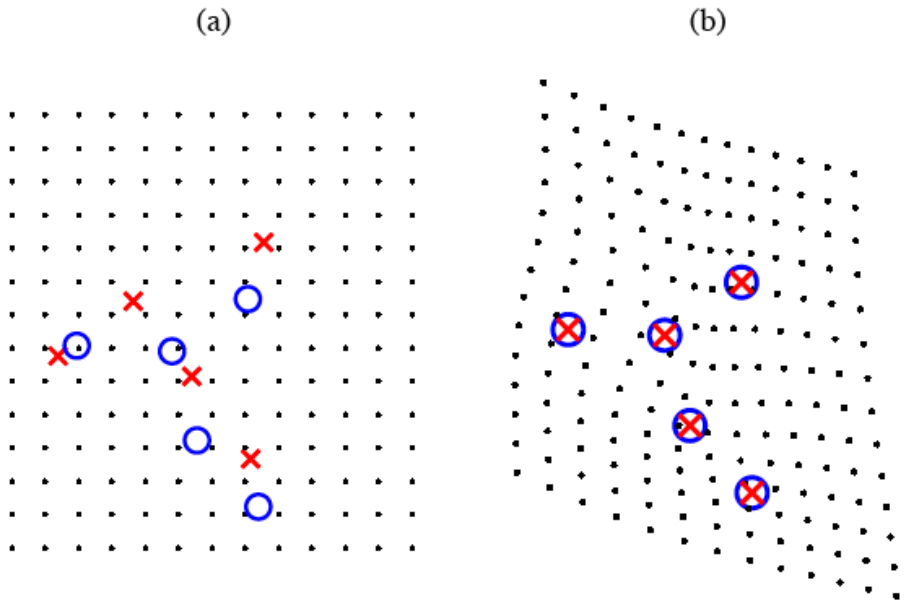
TPS model:

First the equation:

$$f(x,y) \quad = \quad a_1 + a_x x + a_y y + \sum_{i=1}^{p} w_i U\left(\|(x_i,y_i)-(x,y)\|\right)$$



how do we estimate the TPS parameters?



Say the blue circles are the source image features, and red crosses are the target image features, how do we "back-warp" all the image features in the target image back to the source image?

We could compute a TPS model that maps "red" to "blue", and apply it to the rest of the target image pixels.

$$f(x,y) \quad = \quad a_1 + a_x x + a_y y + \sum_{i=1}^{p} w_i U\left(\|(x_i,y_i) - (x,y)\|\right)$$

how do we estimate the TPS parameters?

We would need two functions:

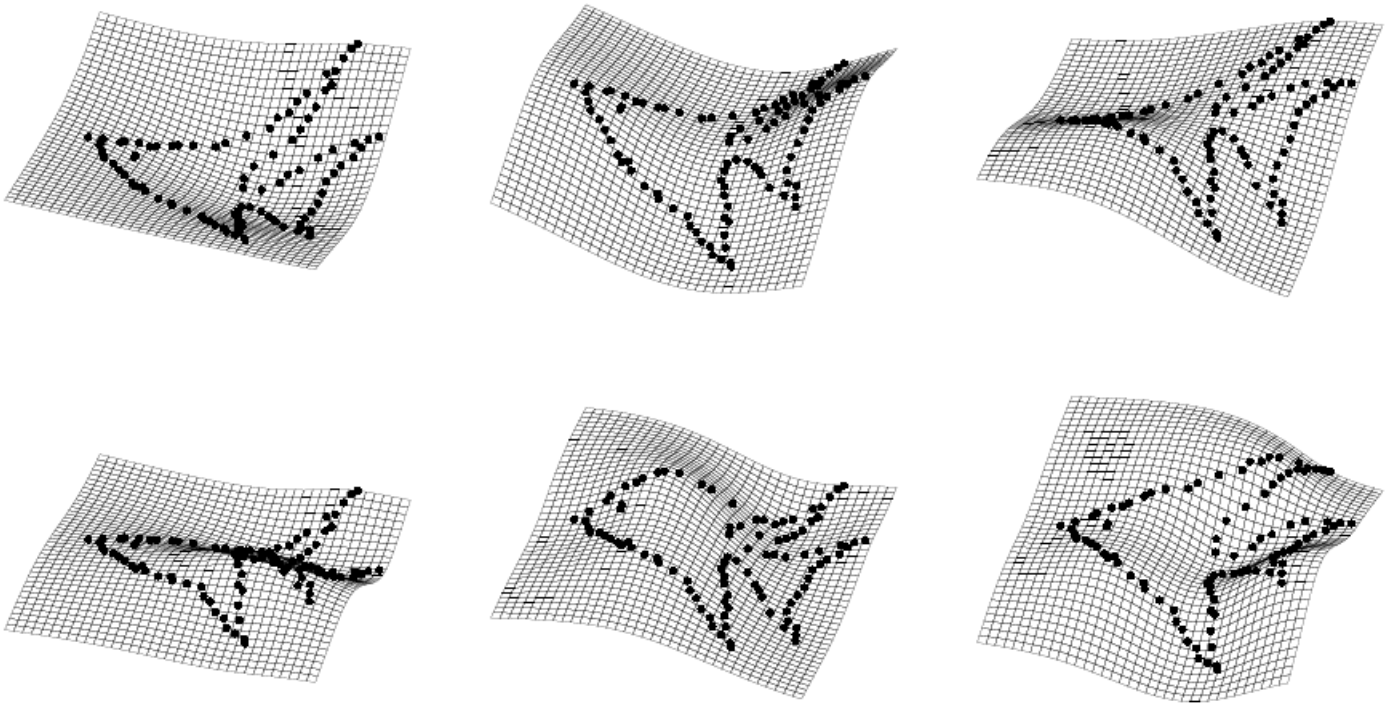
1) **tps_model = est_tps(source_pts, target_pts);**
2) **morphed_im = morph(im_source, tps_model);**

tps_model = (a1, ax, ay, w1, …, wp)

$$f(x,y) = a_1 + a_x x + a_y y + \sum_{i=1}^{p} w_i U\left(\|(x_i,y_i)-(x,y)\|\right)$$

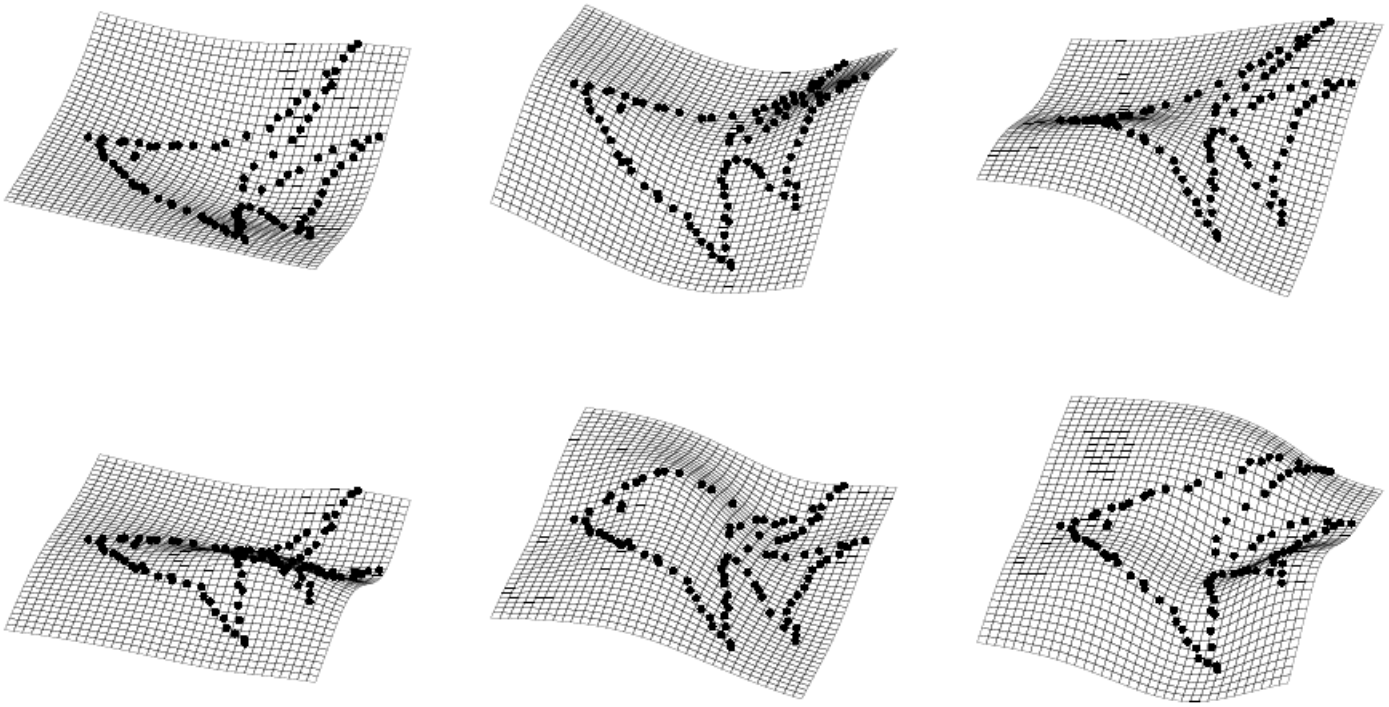
TPS model, special case 1, translation only

$$f(x,y) = \boxed{a_1} + a_x x + a_y y + \sum_{i=1}^{p} w_i U\left(\|(x_i,y_i)-(x,y)\|\right)$$



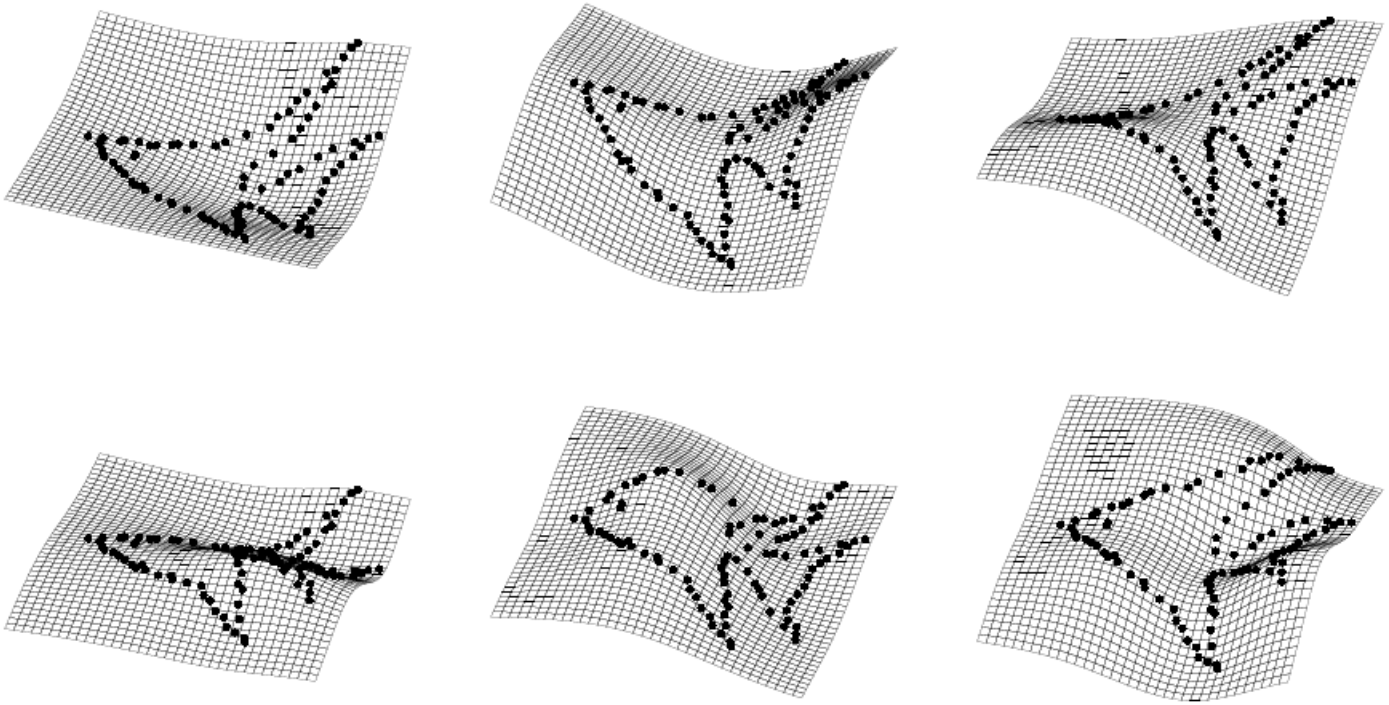
TPS model, case 2: Affine parameters only

$$f(x,y) \quad = \boxed{a_1 + a_x x + a_y y} + \sum_{i=1}^{p} w_i U\left(\|(x_i,y_i)-(x,y)\|\right)$$

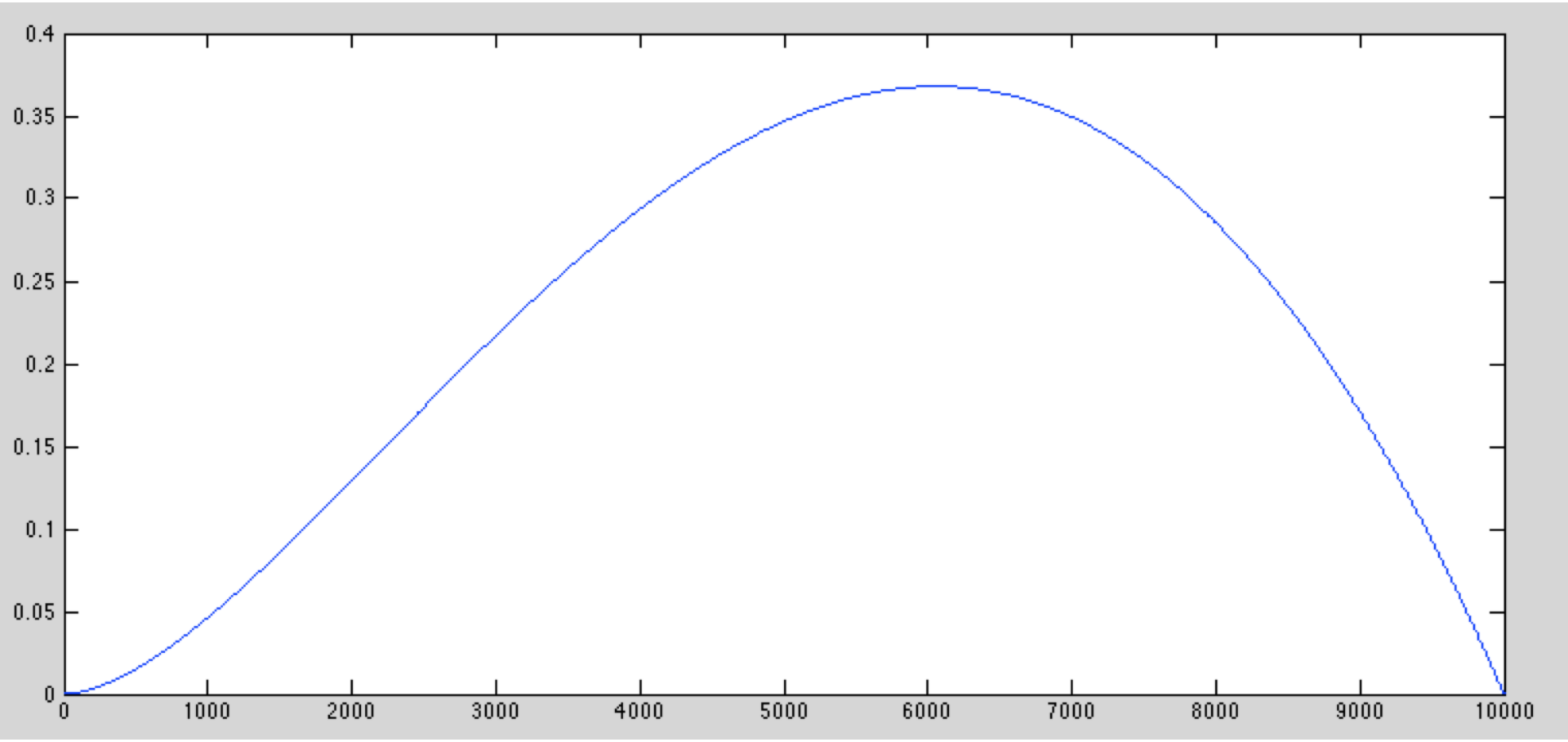


TPS model: Full general case:

$$f(x,y) \quad = \quad a_1 + a_x x + a_y y + \sum_{i=1}^{p} w_i U\left(\|(x_i,y_i)-(x,y)\|\right)$$

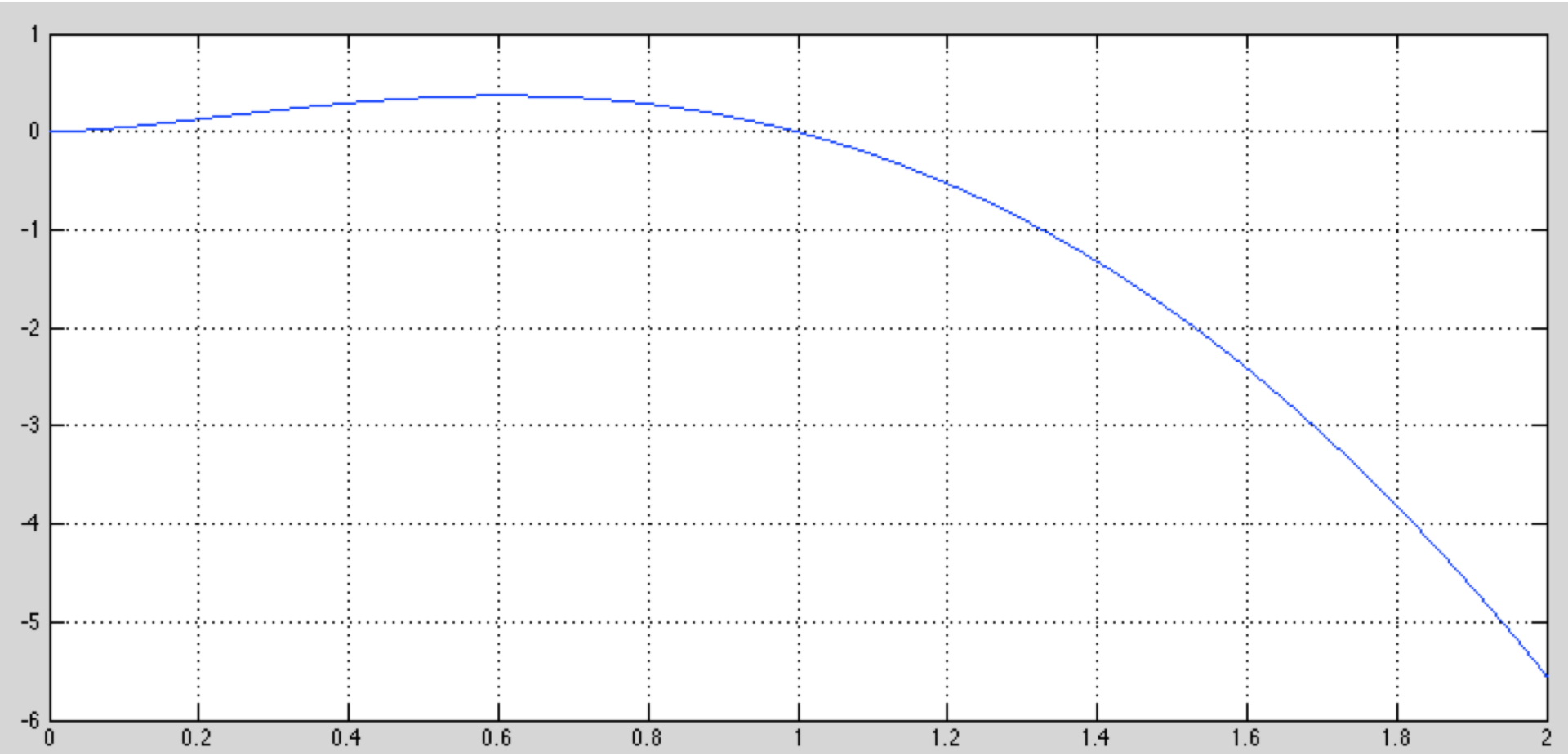


U(r) = - r^2 log(r^2)



r = [0, 1] Max at r = 1/sqrt(e) = 0.607

$$U(r) = -r^2 \log(r^2)$$



$r = [0, 2]$

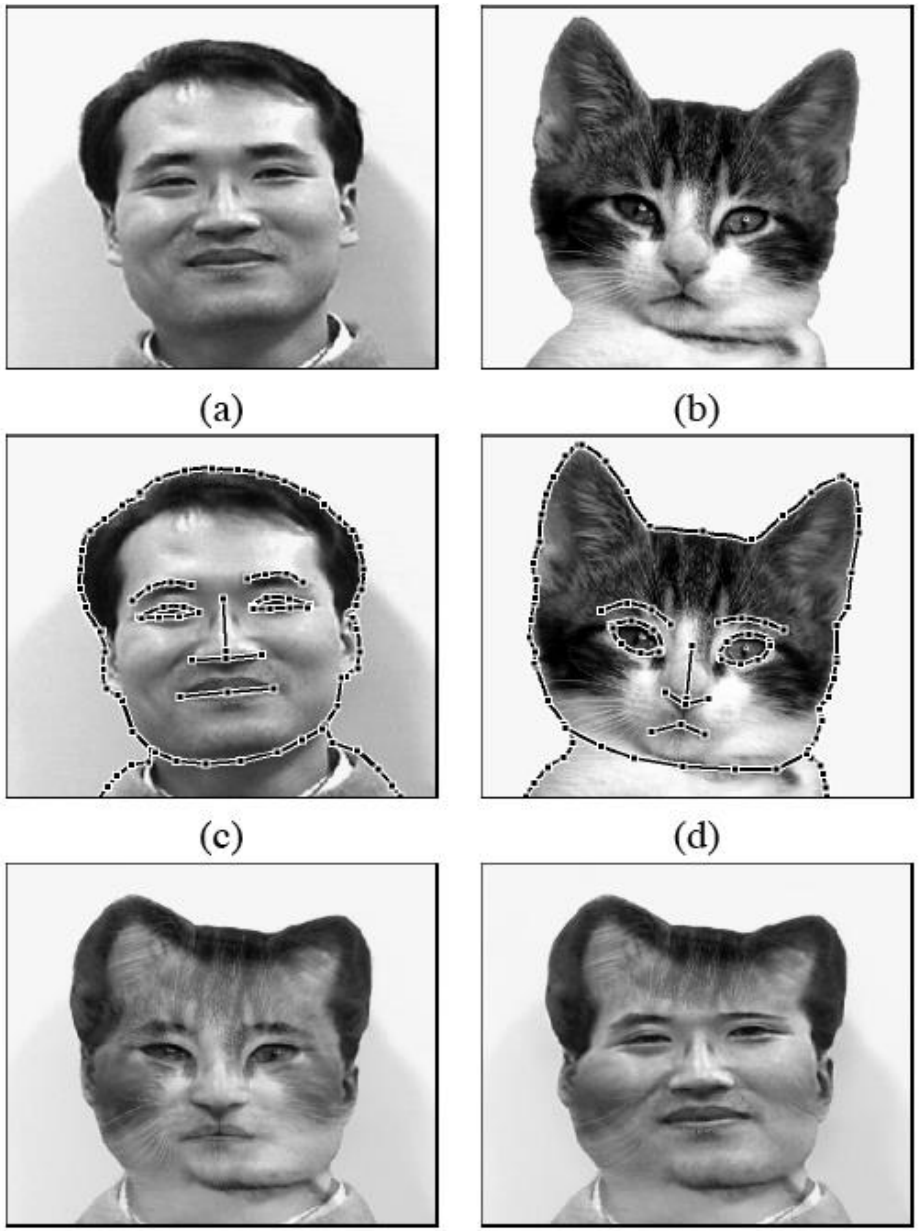
TPS model:

$$f(x,y) \quad = \quad a_1 + a_x x + a_y y + \sum_{i=1}^{p} w_i U\left(\|(x_i,y_i)-(x,y)\|\right)$$

$K_{ij} = U(\|(x_i,y_i)-(x_j,y_j)\|)$,

ith row of P is $(1, x_i, y_i)$,

$$\begin{bmatrix} K & P \\ P^T & O \end{bmatrix} \begin{bmatrix} w \\ a \end{bmatrix} = \begin{bmatrix} v \\ o \end{bmatrix}$$


(a)
(b)
(c)
(d)