Project 2 Review

Image Morphing

Content

- Overview
- Correspondence
- Triangulation
- Thin Plate Spline
- Midterm Review

Overview

- Goal:
  - Understand and implement two morphing techniques, namely, Triangulation and TPS.
- Results: (4 videos in total)
  - Customized face morphing (YOUR face -> someone's face)
    - One for triangulation, one for TPS
    - 60 frames of animation. (.avi movie)
  - Standard testing morphing (a square consisting of 4 triangles)
    - One for triangulation, one for TPS

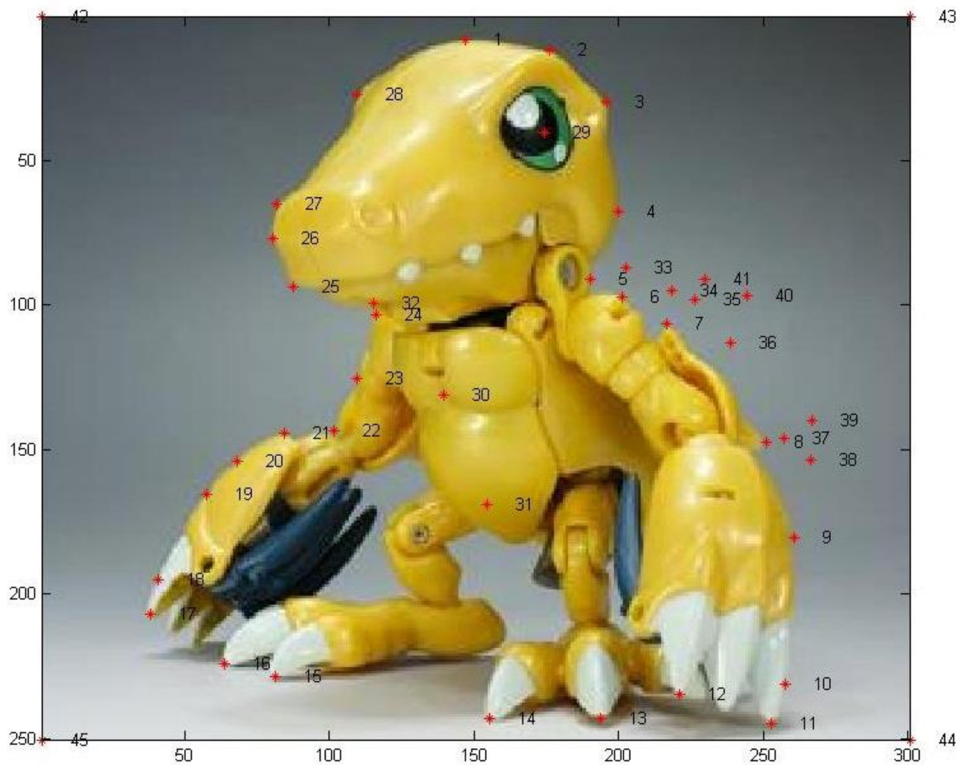
Procedure

- Task 1: Define Correspondences
  - [im1 pts, im2 pts] = click_correspondences(im1, im2)
- Task 2: Image Morph via Triangulation
  - morphed_im = morph(im1, im2, im1_pts, im2_pts, warp_frac, dissolve_frac);
- Task 3: Image Morph via Thin Plate Spline
  - [a1, ax, ay, w] = est_tps(ctr_pts, target_value)
  - morphed_im = morph_tps(im_source, a1_x, ax_x, ay_x, w_x, a1_y, ax_y, ay_y, w_y, ctr_pts, sz)
  - morphed_im = morph_tps_wrapper(im1, im2, im1_pts, im2_pts, warp_frac, dissolve_frac)

Example

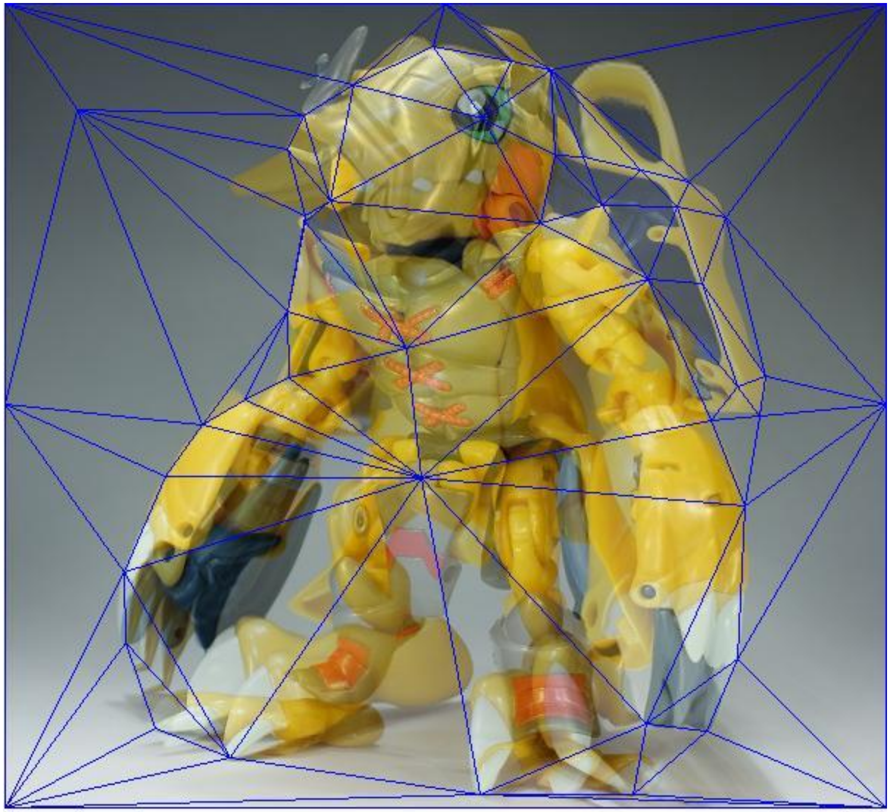
- (You need a face instead of a cute dinosaur…)

Define Correspondences



- Recommended function: cpselect(moving,fixed,CPSTRUCT_IN)

Define Triangular Mesh



- Recommended function: TRI = delaunay(X,Y)
- Note: Better to use control points at the midway shape (mean)

Image Morphing

- We know how to warp one image into the other, but 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

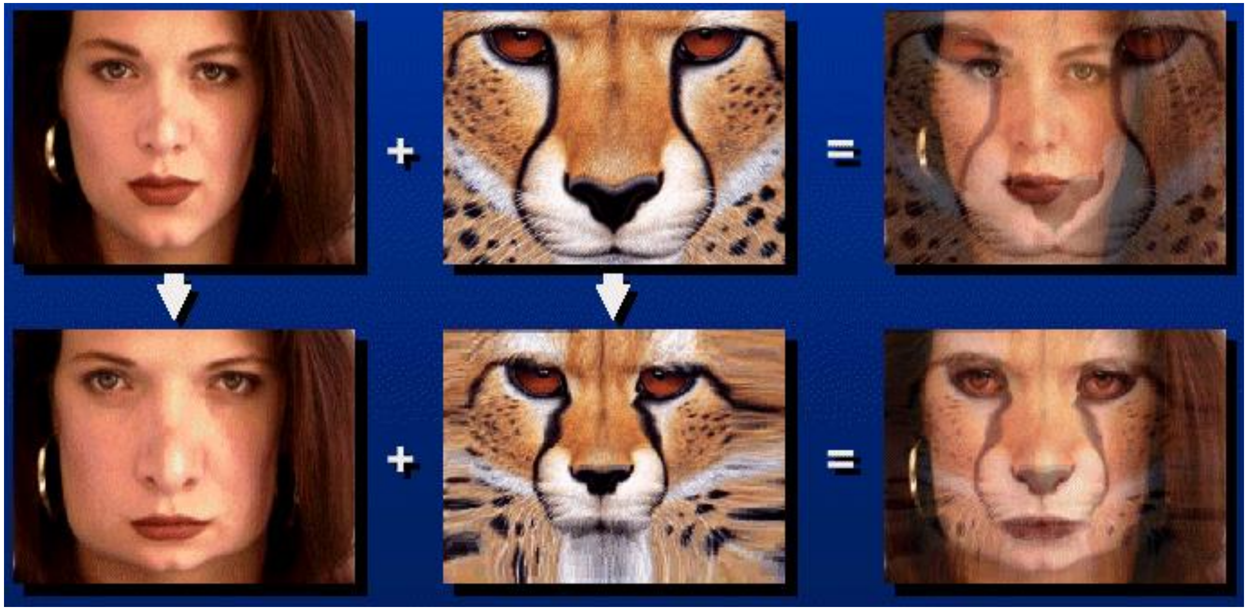
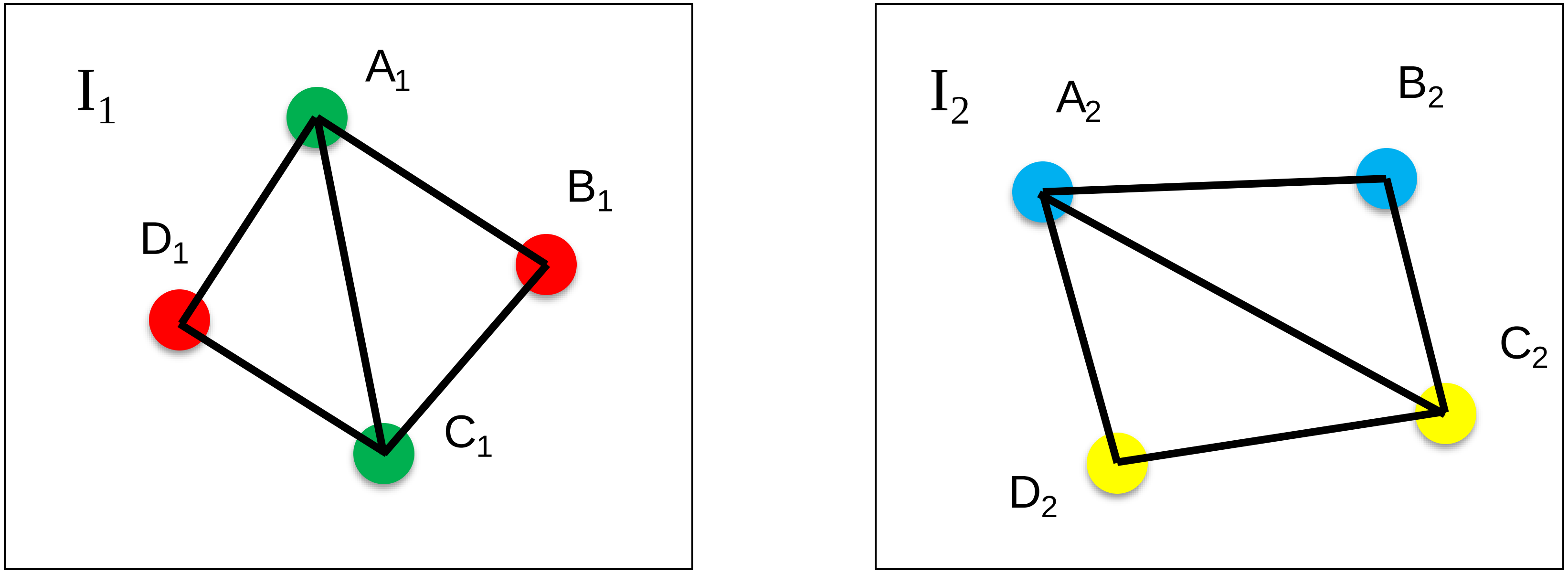


Image Morphing



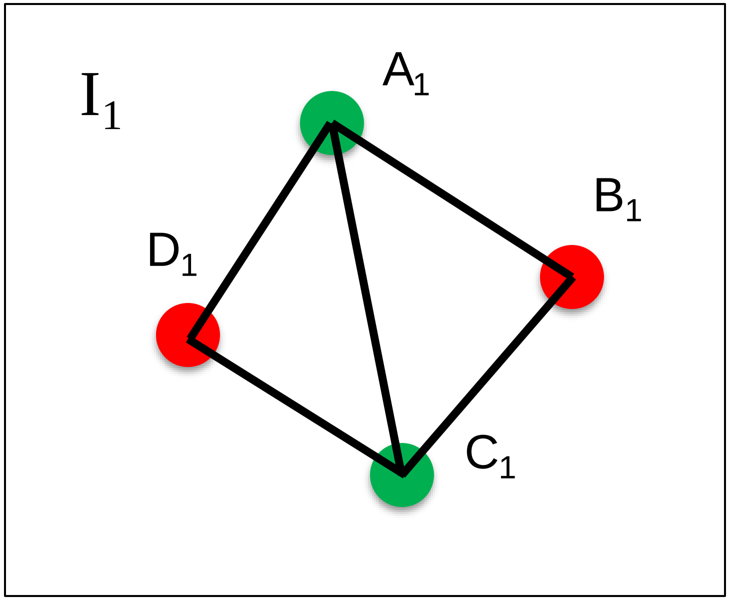
- Corresponding points:
  - A1 – A2, B1 – B2, C1 – C2, D1 – D2

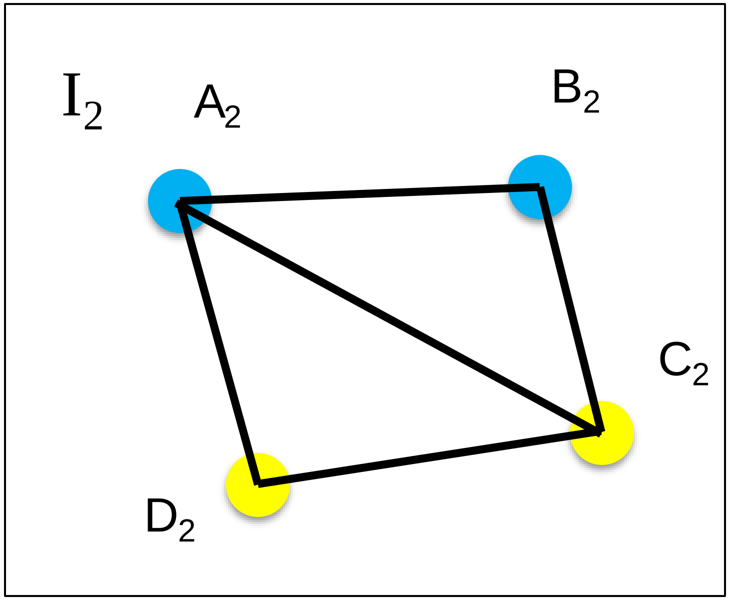
•Step1: Create an intermediate shape (by interpolation)

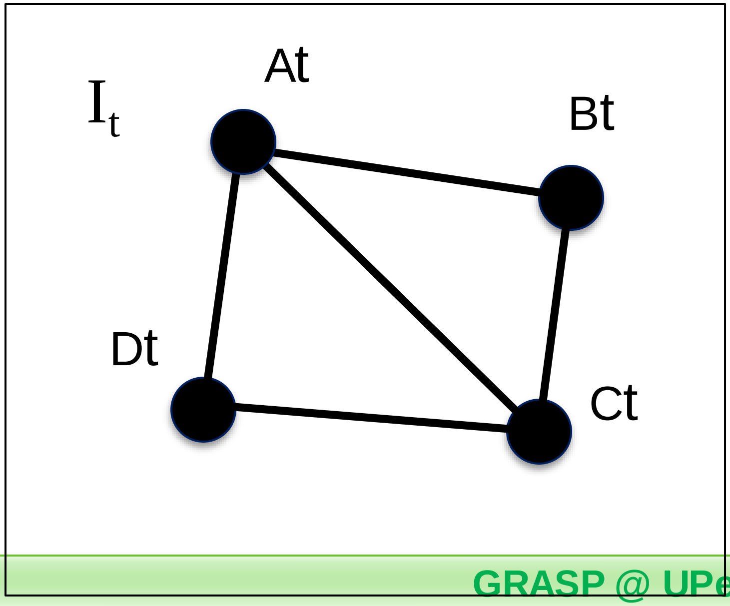
•Step2: Warp both images towards the shape

•Step3: Cross-dissolve the color

Image Morphing: Intermediate Shape







$$A_t = tA_1 + (1-t)A_2$$
$$B_t = tB_1 + (1-t)B_2$$
$$C_t = tC_1 + (1-t)C_2$$
$$D_t = tD_1 + (1-t)D_2$$
$$0 \le t \le 1$$

Image Morphing: Warping

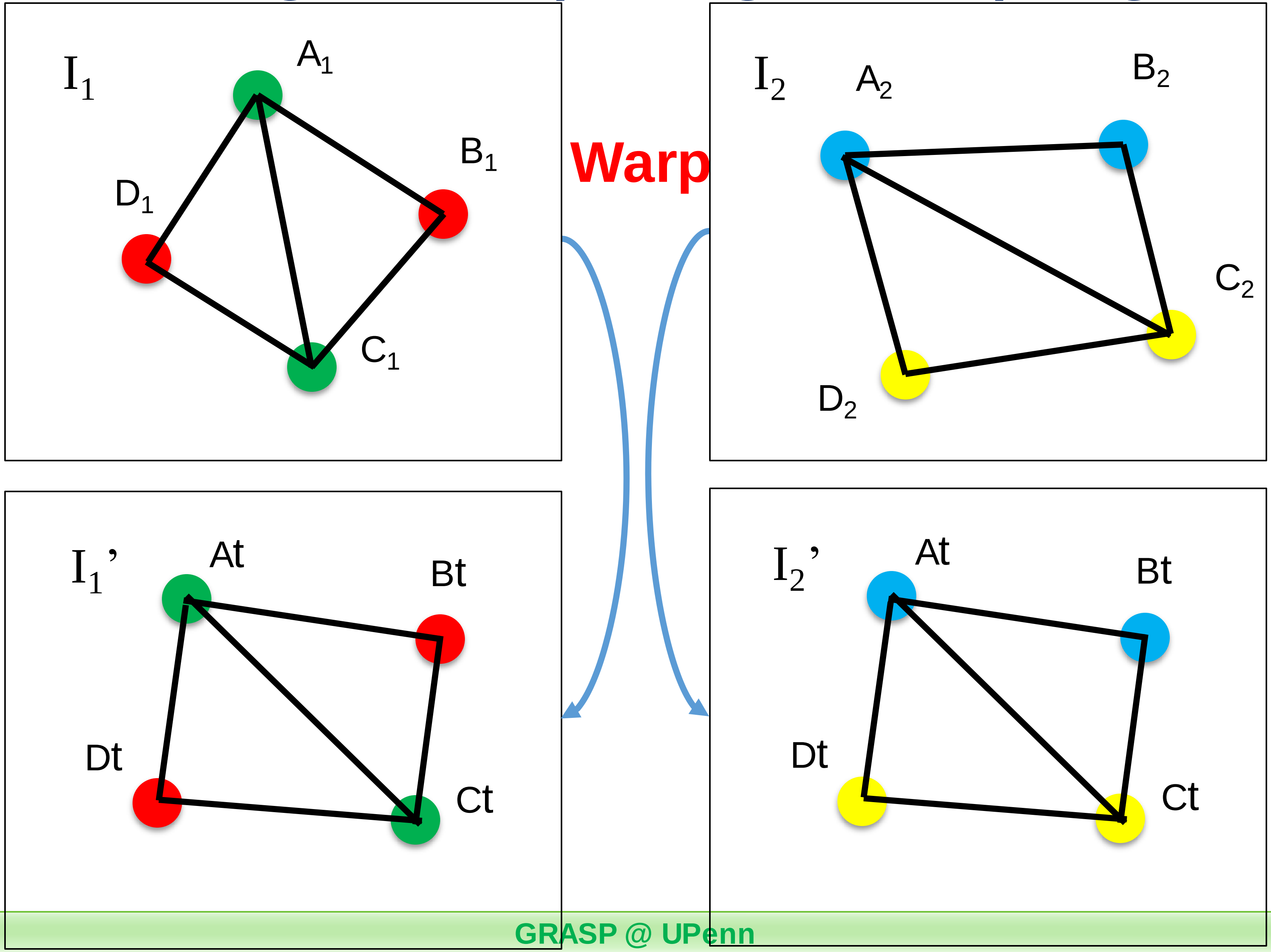
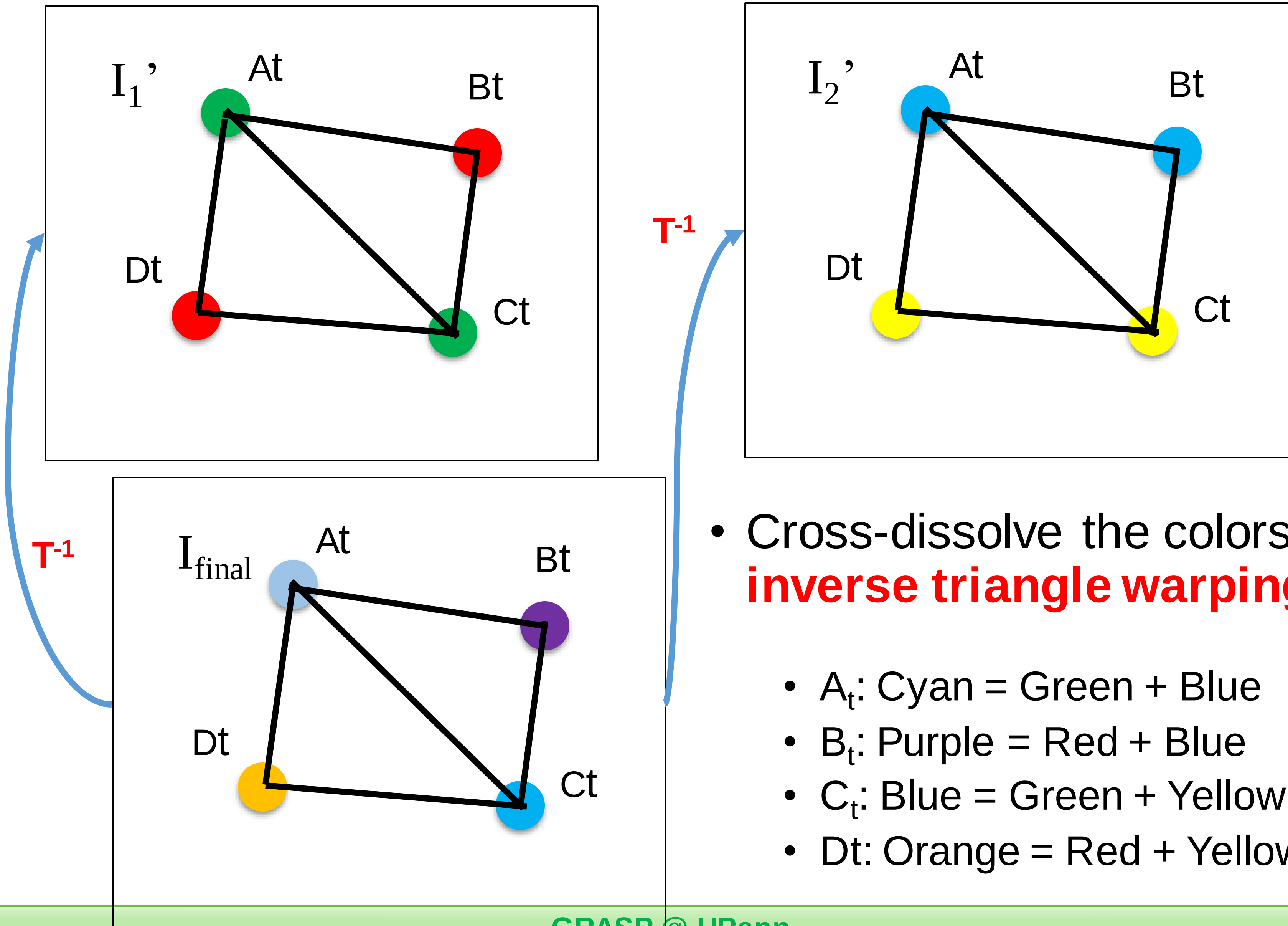


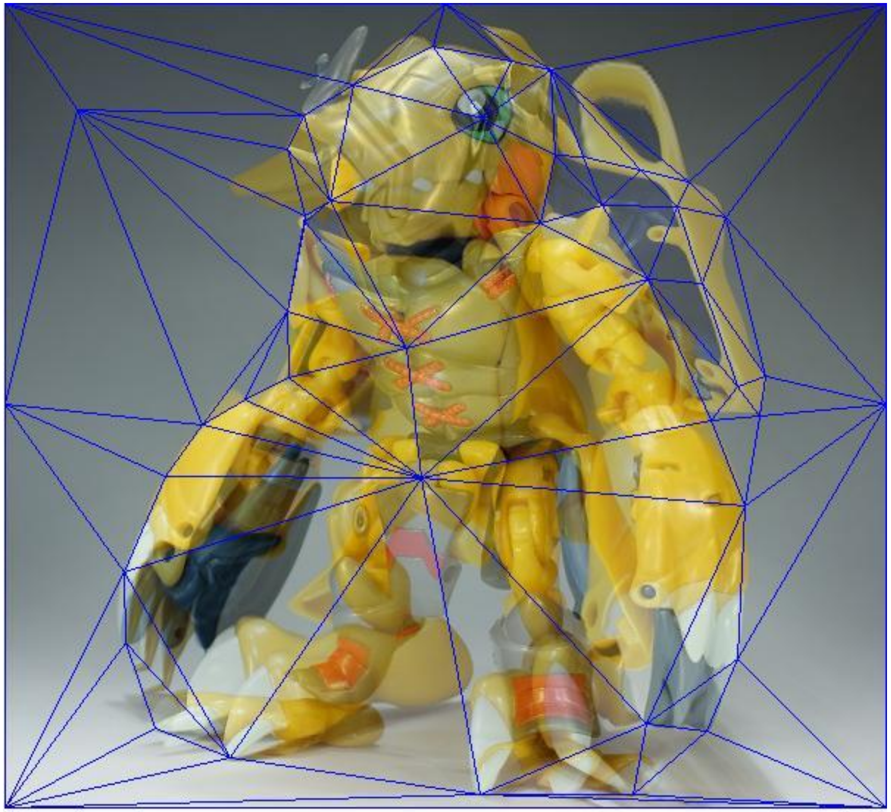
Image Morphing: Cross Dissolve Colors



- 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

Compute Intermediate Shape

- impoints = (1-warpFrac)*im1points + warpFrac*im2points;



Compute Barycentric Coordinate

- Find which triangle a point is in:
  - t = tsearchn(X,TRI,XI)
  - Note: You are NOT allowed to use this function directly for barycentric coordinate.

- Compute barycentric coordinate in target image (α, β, γ)
  - Note: compute the inverse matrix for each triangle once.

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

Inverse Warping

- Use barycentric coordinate to get the pixel values in two source images (im1 & im2)

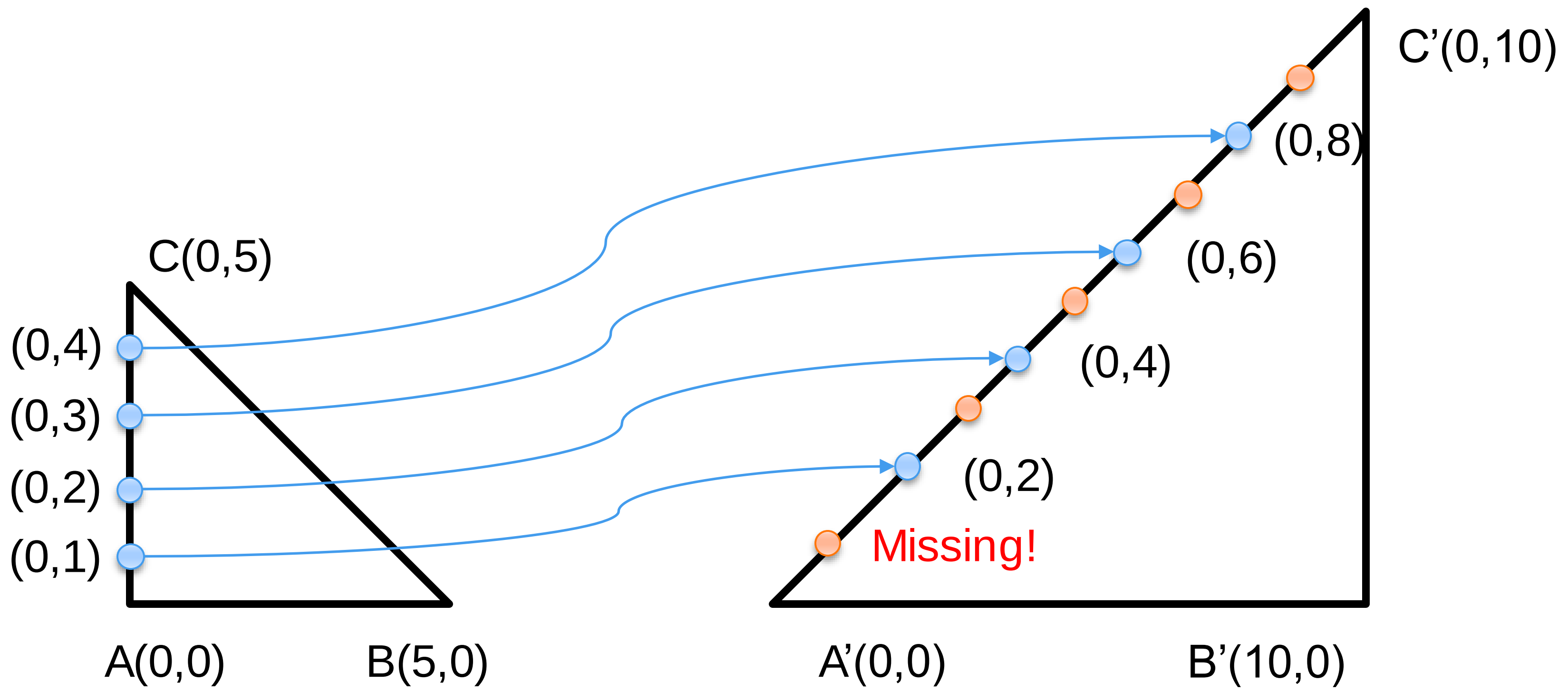
$$\begin{bmatrix} a_x^s & b_x^s & c_x^s \\ a_y^s & b_y^s & c_y^s \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Round the pixel location or use interp2

- What will happen if you don't use inverse warping?

Forward Warping (Wrong)

- What will happen if you use forward warping?

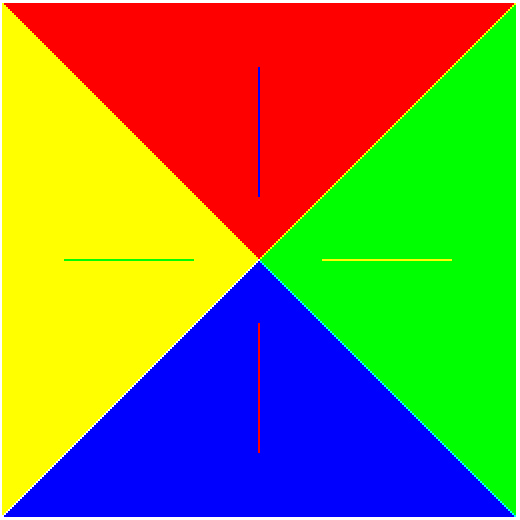


Cross Dissolve

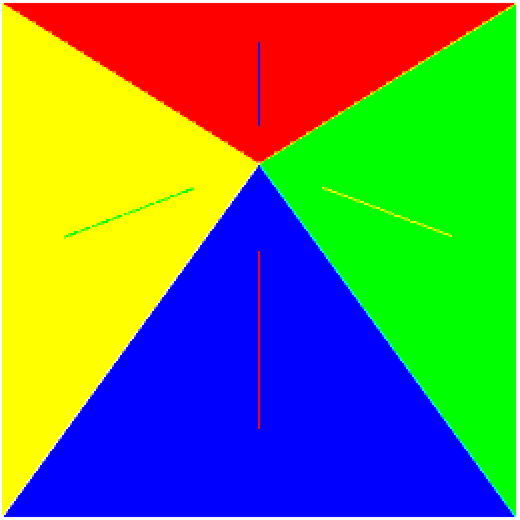
morphed_im = (1-dissolve_frac)*morphed_im1 + dissolve_frac*morphed_im2;

Example

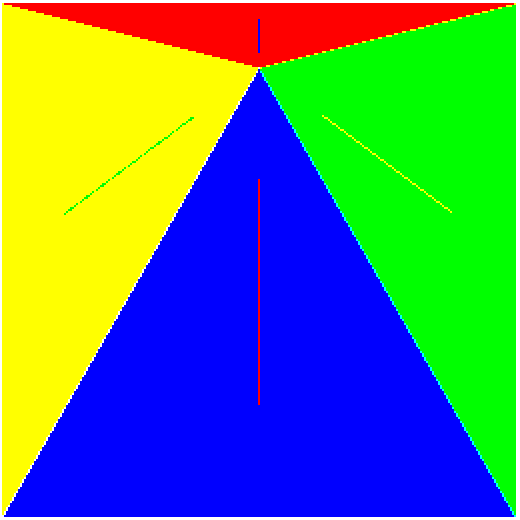
Frac = 0



Frac = 0.5



Frac = 1

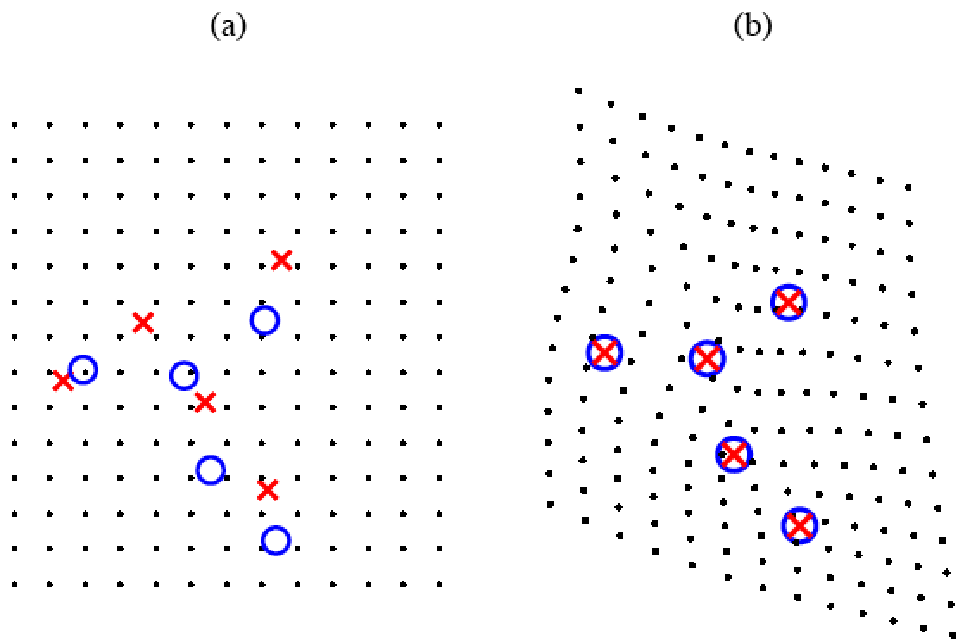


Last Year's Testing Results

- Triangulation
- https://www.youtube.com/watch?v=48LHPImEPws

- TPS
- https://www.youtube.com/watch?v=23S8B38rD_E

Thin Plate Splines



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

Compute TPS Coefficients (est_tps)

$$f(x, y) = a_1 + a_x \cdot x + a_y \cdot y + \sum_{i=1}^{P} W_i U(||(x_i, y_i) - (x, y)||),$$

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

- Compute two sets of coefficients for x and y:
  - $x' = f_x(x, y)$
  - $y' = f_y(x, y)$

- Note: Take care of r = 0.

Compute TPS Coefficients cont.

$$\begin{bmatrix} K & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \cdots \\ w_n \\ a_x \\ a_y \\ a_1 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ \cdots \\ v_n \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

where

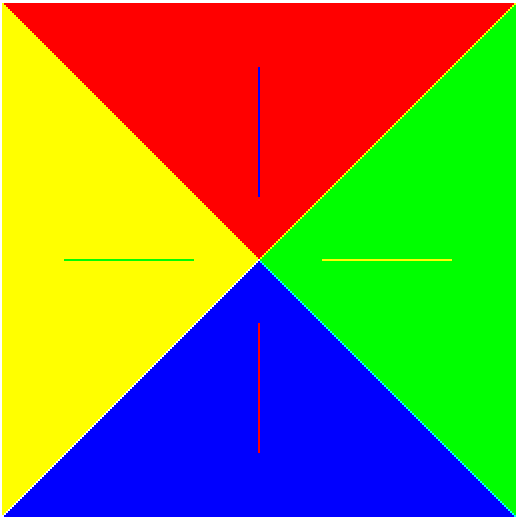
$$K_{ij} = U(||(x_i, y_i) - (x_j, y_j)||),$$

$$\begin{bmatrix} w_1 \\ w_2 \\ \cdots \\ w_n \\ a_x \\ a_y \\ a_1 \end{bmatrix} = inv(\begin{bmatrix} K & P \\ P^T & 0 \end{bmatrix} + \lambda * I(n+3, n+3)) \begin{bmatrix} v_1 \\ v_2 \\ \cdots \\ v_n \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

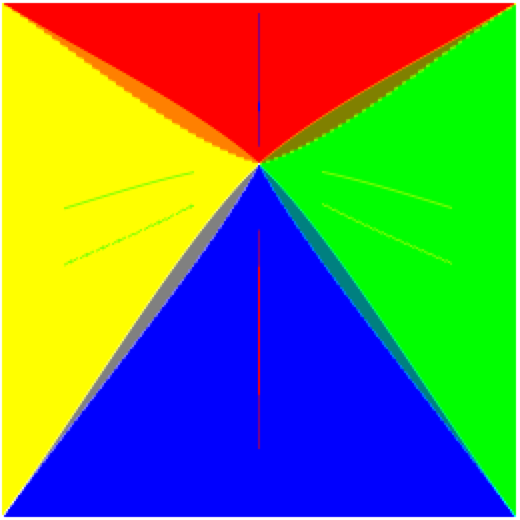
- Note: λ should be very small.

Example

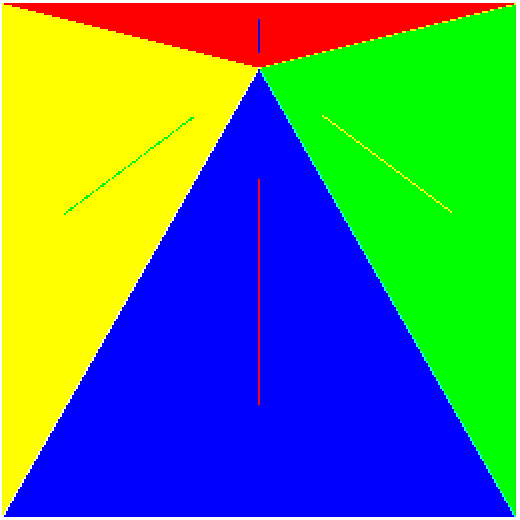
Frac = 0

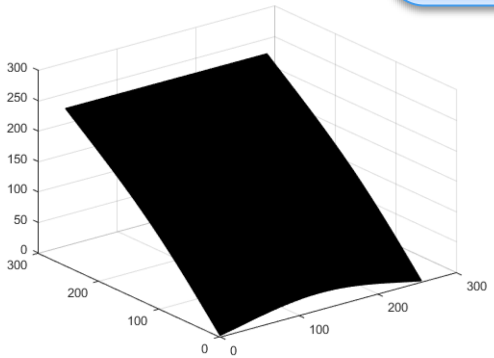


Frac = 0.5

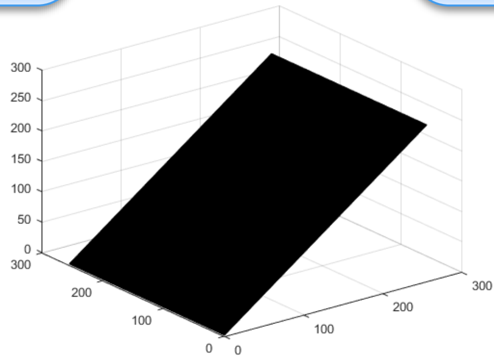


Frac = 1

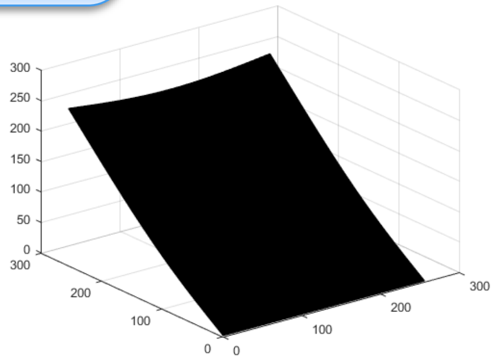




X surface



Y unchanged



X surface

Last Year’s Testing Results

- Triangulation
- https://www.youtube.com/watch?v=48LHPImEPws

- TPS
- https://www.youtube.com/watch?v=23S8B38rD_E

Generate Customized Videos

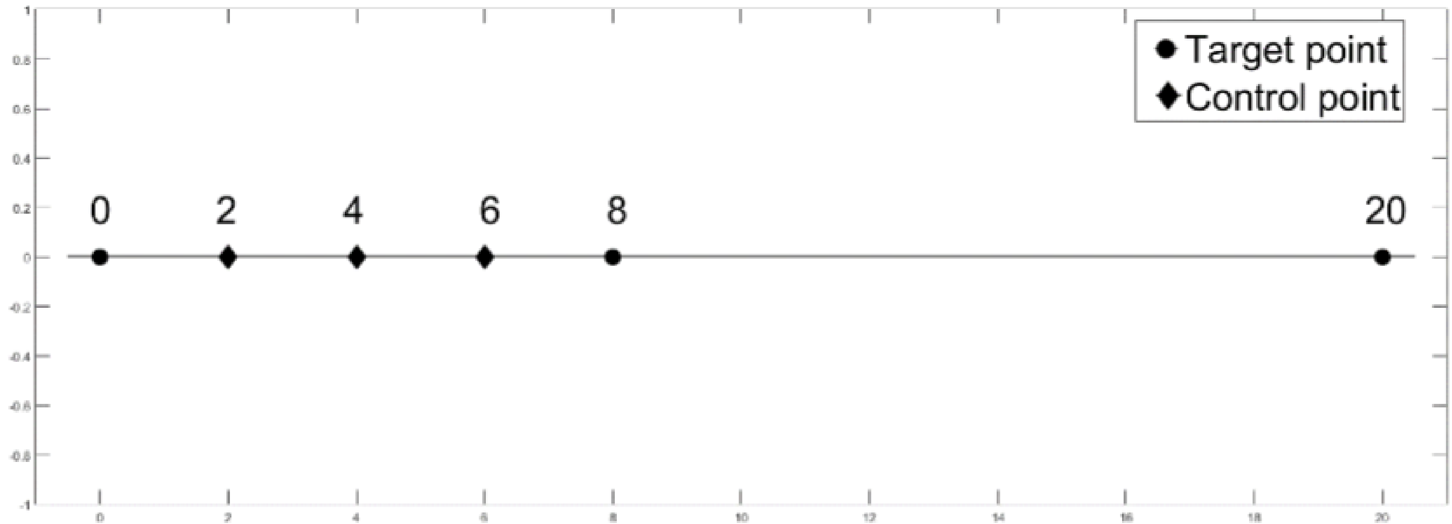
1. VideoWriter
   - video = VideoWriter(filename, 'Uncompressed AVI');
   - video.FrameRate = framerate;
   - video.open();
   - video.writeVideo(image);
   - video.close();

2. avifile
   - video = avifile(filename, 'FPS', framerate);
   - video = addframe(video, image);
   - video = close(video);

Common Mistakes

- Image array data type: should be uint8 -> double -> uint8
- Mess up the coordinate (flip x and y)
- Inconsistent correspondences
- Boundary issue
  - How to deal with points outside of boundaries?
- Forget inverse warping
- λ in TPS is not well chosen

2015 Midterm Review (TPS)

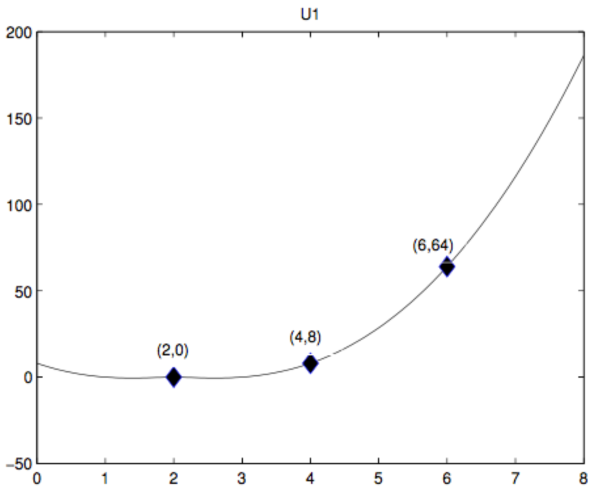


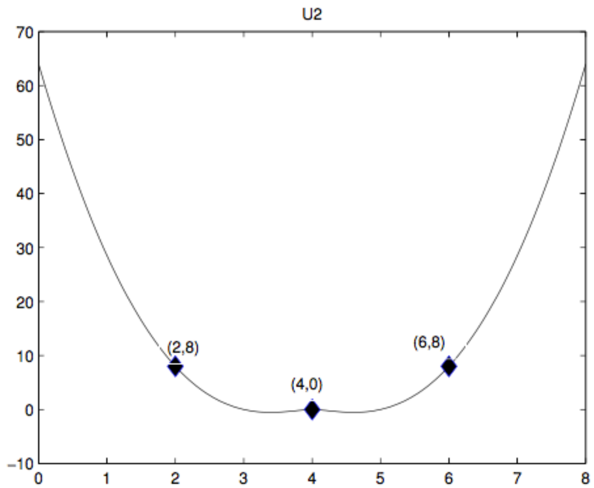
Consider the plot above. We want to find a function that maps the control points to the target points (2 to 0, 4 to 8, 6 to 20). Our model will be:

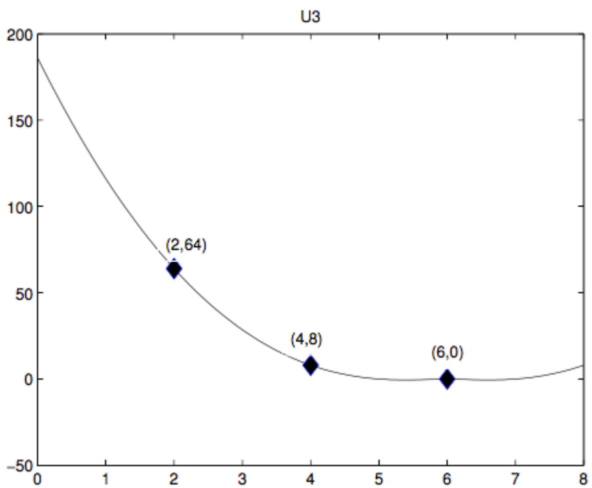
$$f(x) = \sum_{i=1}^{3} w_i U_i(x)$$

Where the U_i are basis functions given by the plots below and the w_i are the parameters in the model. (Note that the affine terms that would be present in a TPS model have been removed to reduce the number of variables)
Solve for w_1, w_2, and w_3.

2015 Midterm Review (TPS) cont.







2015 Midterm Review (TPS) sol.

$$f(2) = w_1U_1(2) + w_2U_2(2) + w_3U_3(2) = 0w_1 + 8w_2 + 64w_3 = 0$$
$$f(4) = w_1U_1(4) + w_2U_2(4) + w_3U_3(4) = 8w_1 + 0w_2 + 8w_3 = 8$$
$$f(6) = w_1U_1(6) + w_2U_2(6) + w_3U_3(6) = 64w_1 + 8w_2 + 0w_3 = 20$$

$$\begin{bmatrix} 0 & 8 & 64 \\ 8 & 0 & 8 \\ 64 & 8 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 8 \\ 20 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \rightarrow \mathbf{w} = \begin{bmatrix} \frac{21}{32} & -\frac{11}{4} & \frac{11}{32} \end{bmatrix}^{\top}$$