Assignment 5: Geometry

Harry Kim

Handed out: October 3, 2022
Due: 11:59pm, October 21, 2022
Handed in: 11:59pm, October 22, 2022

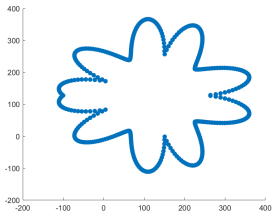
Important Notes:

- Feel free discuss the homework with the instructor or the TAs.
- Handwritten solutions will not be accepted.
- Turn in a PDF report and .m/.py files through Canvas as a compressed (.zip) file; turn in a hardcopy of PDF printout in class)

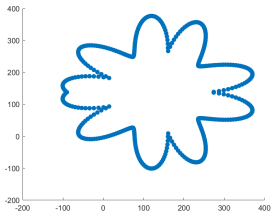
Question 1: Transformations

(a) Write a function that applies a given translation (x-shift, y-shift) to a given 2D shape in the homogeneous coordinates. Report results for the following shift values (10,10), (20,-10), (-10,-20) and (-30,10).

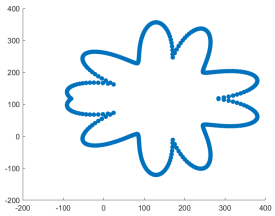
shape1-question1.mat



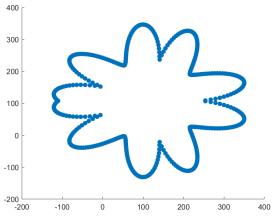
(a) Original image



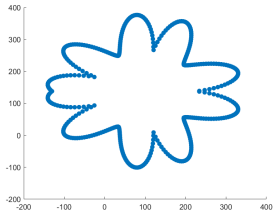
(b) Shift (10,10)



(c) Shift (10,-10)

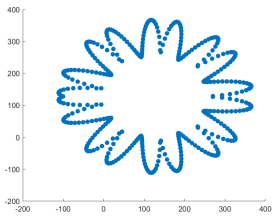


(d) Shift (-10,-20)

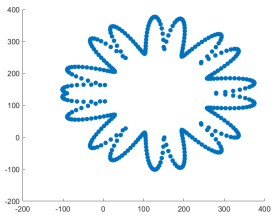


(e) Shift (-30,10)

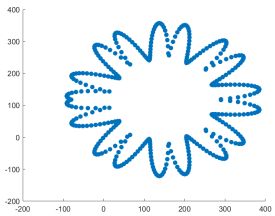
shape2-question1.mat



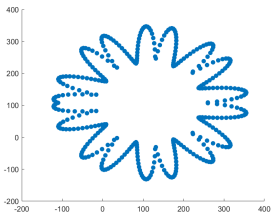
(a) Original image



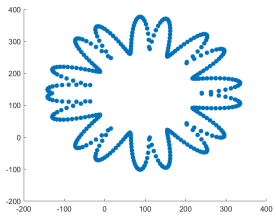
(b) Shift (10,10)



(c) Shift (10,-10)



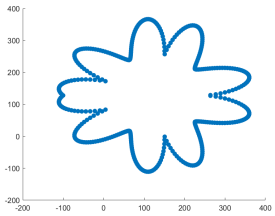
(d) Shift (-10,-20)



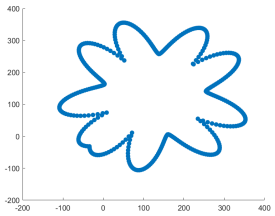
(e) Shift (-30,10)

(b) Write a function that applies a given rotation by an angle theta to a given 2D shape in the homogeneous coordinates. Report the results on the following rotation values 45, 90, -45, -90 degrees.

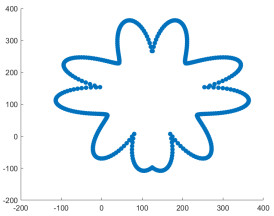
shape1-question1.mat



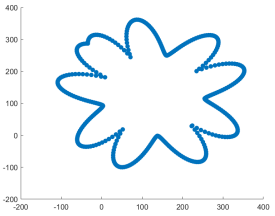
(a) Original image



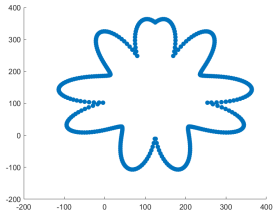
(b) Rotate 45 degrees



(c) Rotate 90 degrees

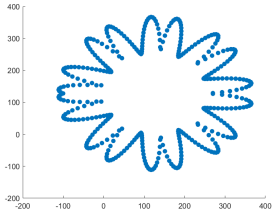


(d) Rotate -45 degrees

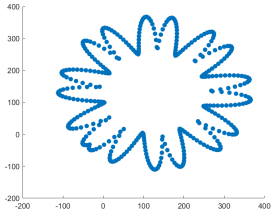


(e) Rotate -90 degrees

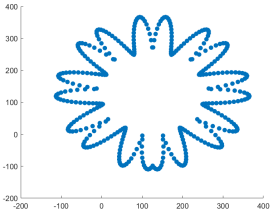
shape2-question1.mat



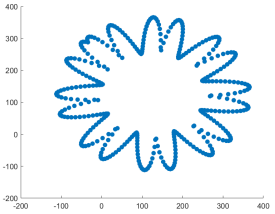
(a) Original image



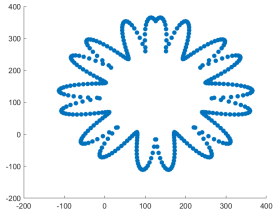
(b) Rotate 45 degrees



(c) Rotate 90 degrees



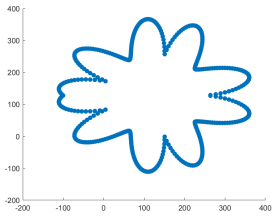
(d) Rotate -45 degrees



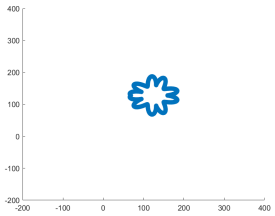
(e) Rotate -90 degrees

(c) Write a function that applies a given uniform scaling by a value s to a given 2D shape in the homogeneous coordinates. Report the results on the following scaling values 0.25, 0.5, 2, 8.

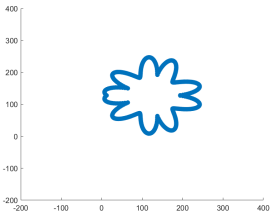
shape1-question1.mat



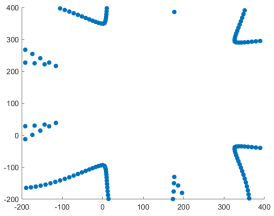
(a) Original image



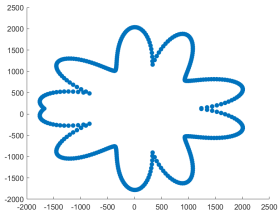
(b) Scale 0.25



(c) Scale 0.50

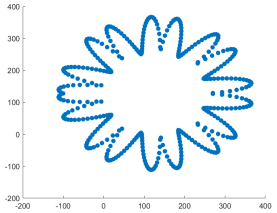


(d) Scale 2

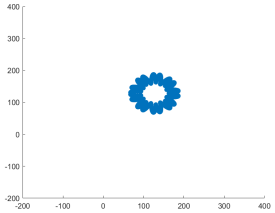


(e) Scale 8

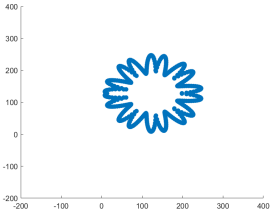
shape2-question1.mat



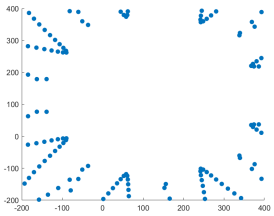
(a) Original image



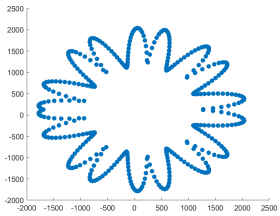
(b) Scale 0.25



(c) Scale 0.50



(d) Scale 2



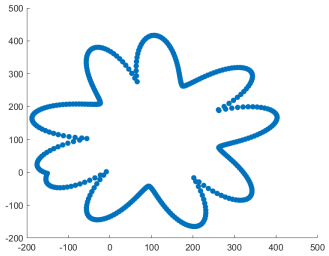
(e) Scale 8

For the last scaling, it would not fit on the graph, so I am showing the plot with the bounds that fits the shapes.

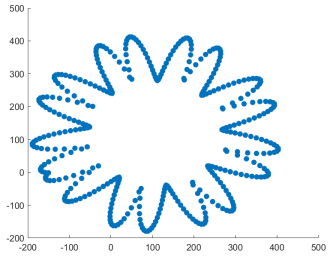
Question 2: Sequence of transformations

(a) Write a function that loads a shape file (.mat) then applies the following sequences of transformations to that shape. Report the result (as a scatter plot) of each intermediate step.

(i) Shift by (-20,-10), rotate the points by 25 degrees, then scale the point by 1.25 factor.

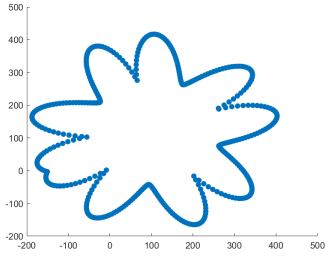


(a) Shape 1

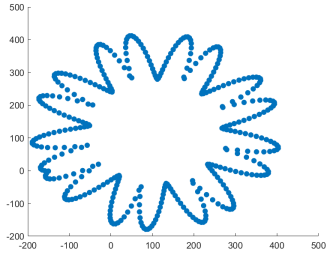


(b) Shape 2

(ii) Scale the point by 1.25 factor, rotate the points by 25 degrees, then shift by (-20,-10).



(a) Shape 1



(b) Shape 2

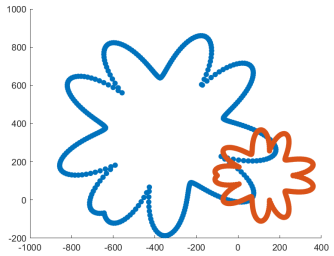
(b) Compare the results from the above experiments. What do these results indicate about the properties of these transformations?

They are the same. The transformations are linearly independent.

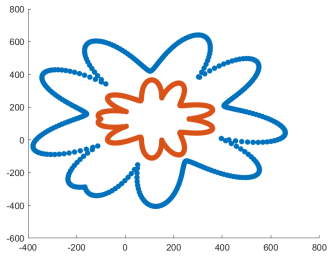
Question 3: Procrustes alignment

(a) Implement Procrustes alignment algorithm from scratch. You cannot use the Procrustes alignment function available in Matlab or Python, but you can use other functions like SVD. Show results at intermediate steps. Report both the shapes before alignment and after each step of alignment. Report the average error in the aligned points.

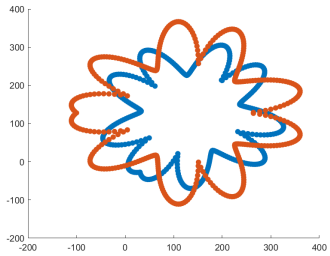
Question3-shape1a.mat & Question3-shape1b.mat



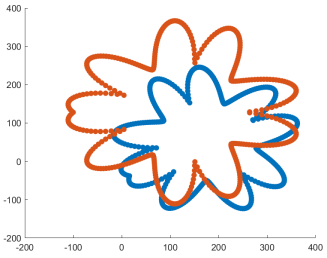
(a) Starting Positions



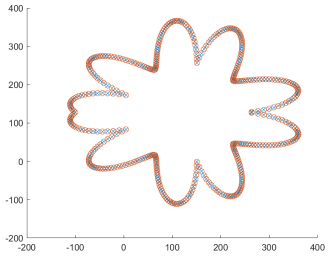
(b) Translation



(c) Scaling



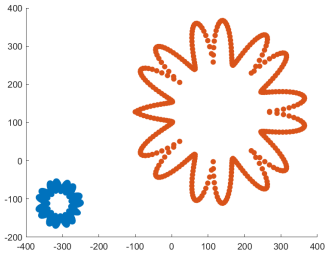
(d) Rotation



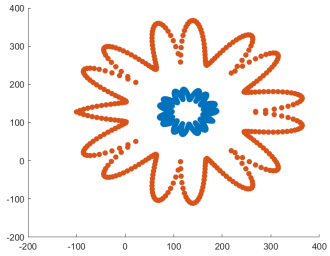
(e) Matlab Solution

(b) Repeat the results with Question3-shape2a.mat and Question3-shape2b.mat from the files shared and report result in the file. Do you observe performance differences between the two sets of shapes?

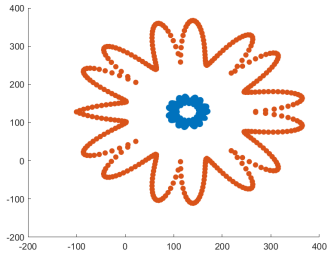
Question3-shape2a.mat and Question3-shape2b.mat



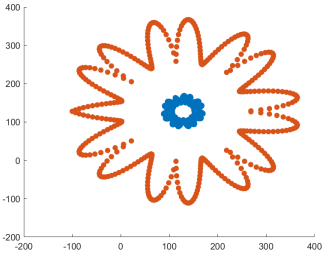
(a) Starting Positions



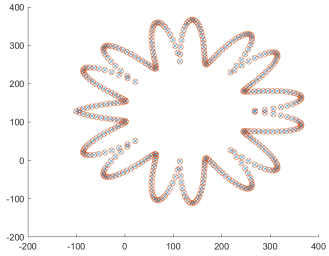
(b) Translation



(c) Scaling



(d) Rotation



(e) Matlab Solution

I really tried to implement Procrustes alignment to the best of my ability, but even though I followed the equations given in the slides and the textbook to the teeth, I could not get it to work.

I do not see any significant performance differences between the two sets of shapes.

The average error for both aligned shapes, reported by matlab, is 0.

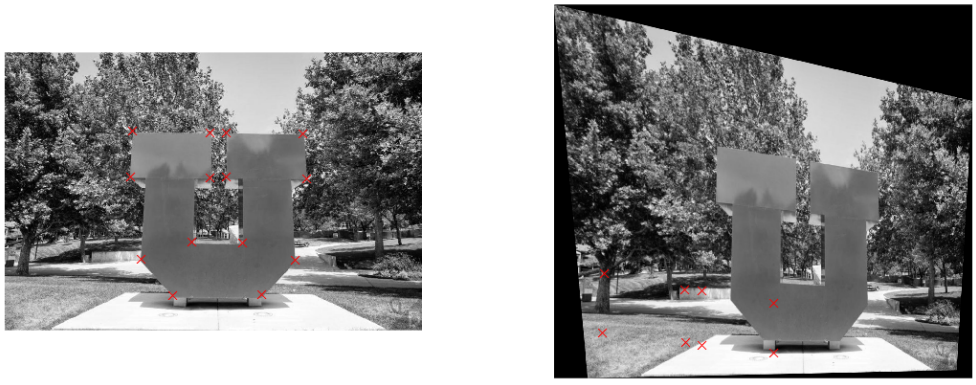
Question 4: Projective transformation

(a) Write a function to find the projective transformation using the given source and destination points. Report the projective transformation matrix obtained.

The projective transformation matrix that I got was the following:

$$\begin{bmatrix} 2.6679 & 0.1439 & -515.0319 \\ 0.8715 & 3.3041 & 535.3812 \\ 0.0004 & 0.0002 & 1 \end{bmatrix}$$

(b) Using a warping function in Python or Matlab, find the transformed image using the projective matrix found in part (a). Plot the src and dst points on scatter plots with images shown. Show the final transformed image.



(a) Projective transformation

I followed the projective transformation outlined in the coding examples, but the points don't exactly line up and I didn't know how to adjust it. Judging from eyeballing it, it seems like the points would line up in the correct spot.

(c) Use the same projective transformation obtained in (a) and write the code to warp the image yourself using the below techniques. Which one is better and why?

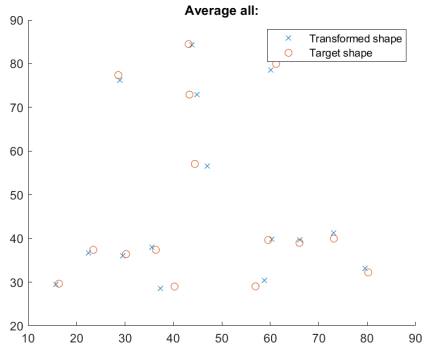
I couldn't figure out how to do this.

Question 5: Alignment for population statistics

(a) Align all the face shapes in the Male and Female folders to the base shape. You can use a built-in Procrustes alignment function in Matlab or Python. Report the mean face shape for all shapes, males, and females.

(a) Female average pts

(b) Male average pts



(c) All average pts

I have plots for all the face shapes and how they line up after Procrustes alignment in the output_images folder.

(b) Use a warp function in Matlab or Python to obtain a transformed face image for each sample. Compute the mean (average) image for all face images before and after alignment. Which mean image is better and why?

(a) Before alignment (b) After alignment

The aligned is better because the tightness of the features displayed in the after aligned image shows that it is better. This can be shown with the (my) right eye that has less distortion and within a more compact area.

(c) Compute the mean (average) image for all male images and a mean image for all female images, before and after alignment. Report how the mean shape of men is different from the mean shape of females.

(a) Average female (b) Average male

The average face for men seems to be frowning more with a more general angry expression while the average face for female seem to be more neutral or happier with a faint smile and less angled eyebrows.

(d) [2 points] Pick any two shape-image pair from each folder and blend images and landmarks using the following equation. Vary a from 0 to 1, with a difference of 0.1. Report all the images in the report.

Female faces:

(a) a = 0 (b) a = 0.1 (c) a = 0.2 (d) a = 0.3 (e) a = 0.4 (f) a = 0.5

(g) a = 0.6 (h) a = 0.7 (i) a = 0.8 (j) a = 0.9 (k) a = 1.0

Male faces:

(a) a = 0 (b) a = 0.1 (c) a = 0.2 (d) a = 0.3 (e) a = 0.4 (f) a = 0.5

(g) a = 0.6 (h) a = 0.7 (i) a = 0.8 (j) a = 0.9 (k) a = 1.0

(e) Using the same shape-image pairs you picked in (d), first align landmarks to the overall mean shape (from part a) using the Procrustes algorithm (you can use built-in functions). Warp the face images using the transformation matrix obtained from the Procrustes algorithm. Now compute the blended images using the transformed/warped images using the same equation as (d) and same values for a. Report all the blended images.

Female faces:

(a) a = 0 (b) a = 0.1 (c) a = 0.2 (d) a = 0.3 (e) a = 0.4 (f) a = 0.5

(g) a = 0.6 (h) a = 0.7 (i) a = 0.8 (j) a = 0.9 (k) a = 1.0

Male faces:

(a) a = 0 (b) a = 0.1 (c) a = 0.2 (d) a = 0.3 (e) a = 0.4 (f) a = 0.5

(g) a = 0.6 (h) a = 0.7 (i) a = 0.8 (j) a = 0.9 (k) a = 1.0

(f) Which blending algorithm is better? Why? Comments should be supported by results from part d and part e results.

I would imagine that the algorithm form e would be better, but, like what happened in question 4, I couldn't get the points to line up exactly, but the transformation is correct. Therefore, out of the results that I got, d is the better algorithm.