

EE 569: Homework #3

Issued: 2/13/2022 Due: 11:59PM, 3/10/2022

General Instructions:

1. Read *Homework Guidelines* and *MATLAB Function Guidelines* for the information about homework programming, write-up and submission.
2. If you make any assumptions about a problem, please clearly state them in your report.
3. You need to understand the USC policy on academic integrity and penalties for cheating and plagiarism. These rules will be strictly enforced.

In the first two problems, you will apply geometric modification and spatial warping techniques to do some interesting image processing tricks. During this process, you may need to solve some linear equations to get the matrix parameters.

Problem 1: Geometric Image Modification (25%)

Design and implement a spatial warping technique that transforms an input square image into an output image of a star-shaped image, out of the square image. An example is given in Figure 1.



Figure 1: Warp the original image to star-shaped image.

The warped image should satisfy the following requirements:

- Pixels that lie on boundaries of the square should still lie on the boundaries of the star.
- Pixels on the two diagonals of the square should remain where they are.
- The thickness (height) of each black arc region is 64 pixels.

Apply the same developed spatial warping algorithm to Forky and Twenty-two images in Figure 2.

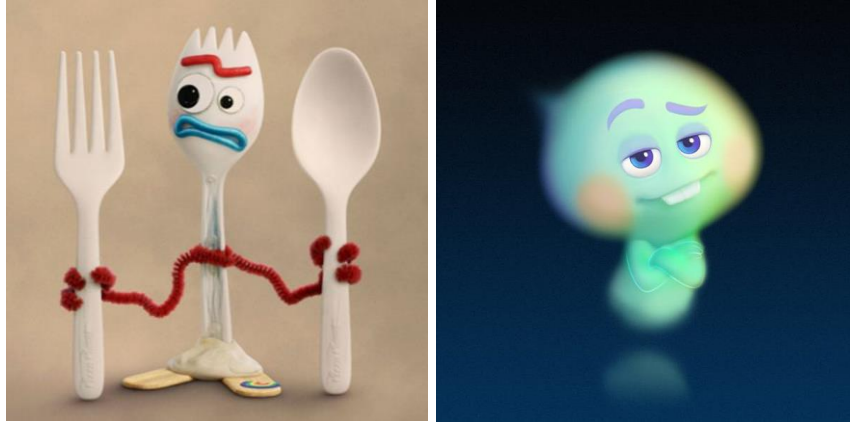


Figure 2: Forky and Twenty-two images.

- (1) Describe your approach as clearly as possible and show the resulting images.
- (2) Apply the reverse spatial warping to each warped image to recover its original image.
- (3) Compare the recovered square image with the original square image. Is there any difference between two images? If any, explain sources of distortion in detail.

Problem 2: Homographic Transformation and Image Stitching (25%)

One can use homographic transformation and image stitching techniques to create panorama that consisting of multiple images. One example (Taken from MATLAB examples [1]) is shown in Figure 3. The left images were taken with an uncalibrated smart phone camera by sweeping the camera from left to right along the horizon to capture all parts of the building. The right panorama is the desired output by stitching transformed images.



Figure 3: An example of image stitching to make panorama [1].

This example involves five images. However, the basic principle is to process in terms of consecutive pair of images. It could be achieved by following these steps:

- Select control points from both images. You can use SIFT/SURF to detect control points. You are allowed to use OpenCV or other open source code for selecting control points, online source refer to [2].
- Apply homographic transformation to find a homograph mapping (described below).
- Warp one image onto the other using the estimated transformation.
- Create a new image big enough to hold the panorama and composite the warped image into it.

You can composite by simply averaging the pixel values where the two images overlap. The homographic transformation procedure is stated below. Images of points in a plane, from two different camera viewpoints, under perspective projection (pin hole camera models) are related by a homography:

$$P_2 = HP_1$$

where H is a 3×3 homographic transformation matrix, P_1 and P_2 denote the corresponding image points in homogeneous coordinates before and after the transform, respectively. Specifically, we have

$$\begin{bmatrix} x'_2 \\ y'_2 \\ w'_2 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \frac{x'_2}{w'_2} \\ \frac{y'_2}{w'_2} \end{bmatrix}$$

To estimate matrix H , you can proceed with the following steps:

- Fix $H_{33} = 1$ so that there are only 8 parameters to be determined.
- Select four-point pairs in two images to build eight linear equations.
- Solve the equations to get the 8 parameters of matrix H .
- After you determine matrix H , you can project all points from one image to another by following the backward mapping procedure and applying the interpolation technique.

Implement above homographic transformation and stitching techniques to composite the scenery images in Figure 4. Show the results and make discussion on the following questions.

- (1) How many control points were used? Show the corresponding control points between left and middle pair, middle and right pair.
- (2) How did you select control points? Clearly specify how you use matched feature from SURF/SIFT to create panorama.



Figure 4. The scenery images (left, middle, right) (credit to Zhiruo Zhou)

Problem 3: Morphological processing (50%)

In this problem, you will implement the morphological processing operation: **thinning**. A pattern table (patternables.pdf) is attached for your reference. For **shrinking** and **skeletonizing**, you can use Matlab or OpenCV functions if needed. Please show outputs for all following parts in your report and discuss them thoroughly. Please state any assumptions you make in your solution.

Note: the provided greyscale images may not be binary images. You need to binarize them before doing morphological processing. Binarize the image in the following way: suppose the maximum pixel intensity is x , then any locations with original intensity greater than $0.5x$ are labeled as 1 in the binary image. Otherwise, 0.

(a) Basic morphological process implementation (15%)

Please apply the “thinning” filter to the pattern images (spring, flower, jar) below. Show your results, including intermediate processing result (around the 20th iteration) and give reasonable explanation.



Figure 5: flower, jar, spring (modified from [3])

(b) Defect detection and counting (20%)

A greyscale image of a deer is shown in Figure 6. The deer image is supposed to be all white; however, there are black defects of different sizes all over the deer. Suppose that black holes whose sizes are smaller than 50 pixels are defects. Please apply morphological processing algorithm for automatic defect counting

and detection. For each question, please explain your method and include intermediate results if there are any:

- (1) Count the total number of defects in the image.
- (2) How many different defect sizes are present in the image? What is the frequency of these defect sizes?
(Hint: The defect size is the number of pixels in a defect. Please state any further assumption you take regarding defect size.)
- (3) Correct the defects and show the clear deer image.

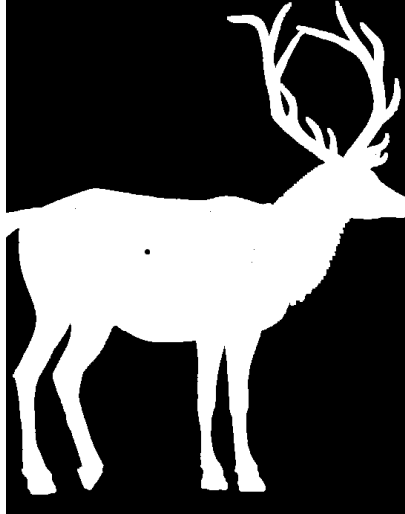


Figure 6: deer with defect.

(c) Object Segmentation and Analysis (15%)

Figure 7 displays different types of beans.



Figure 7: beans (modified from [4])

For each question, please explain your method and include intermediate results if there are any:

- (1) Count the total number of beans via morphological processes.
- (2) Generate the segmentation mask for different types of beans. Then compare the size of different types of beans. Rank the bean's size from small to large in terms of type.

Note: You need to convert the incoming RGB image to grayscale image first and then apply morphological processes. Some pre-processing or post-process may be needed to deal with unwanted dots and holes.

Appendix:

Problem 1&2: Geometric Image Modification

Forky.raw	328x328	24-bit	color (RGB)
22.raw	328x328	24-bit	color (RGB)
left.raw	576x432	24-bit	color (RGB)
middle.raw	576x432	24-bit	color (RGB)
right.raw	576x432	24-bit	color (RGB)

Problem 3: Morphological processing

spring.raw	252x252	8-bit	greyscale
flower.raw	247x247	8-bit	greyscale
jar.raw	252x252	8-bit	greyscale
deer.raw	550x691	8-bit	greyscale
beans.raw	494x82	24-bit	color (RGB)

Note: "550x691" means "width=550, height=691".

References

- [1] Matlab panorama example: <https://www.mathworks.com/help/vision/ug/feature-based-panoramic-image-stitching.html>
- [2] OPENCV feature matching example: https://docs.opencv.org/3.4/d5/d6f/tutorial_feature_flann_matcher.html
- [3] MPEG-7 Shape dataset: <http://www.dabi.temple.edu/~shape/MPEG7/dataset.html>
- [4] <https://wdrfree.com/stock-vector/download/beans-legumes-set-vector-illustration-397505954>