EE 569: Homework #3

Issued: 02/19/2024 Due: 11:59 PM, 03/10/2024

General Instructions:

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

You will apply geometric modification and spatial warping techniques to do some exciting tasks in the first two problems. In 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 given shape from the square image. An example is shown in Figure 1.

Figure 1: Warp the original image to a given shape.

The warped image should satisfy the following requirements:

- Pixels that lie on the 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 Dog and Cat images in Figure 2.

(a) (b)

Figure 2: Dog and Cat 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 the 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 a panorama comprising multiple images. One example (from MATLAB examples [1]) is shown in Figure 3. The left images were taken with an uncalibrated smartphone 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 a panorama [1].

This example involves five images. However, the basic principle is to process a 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 can use OpenCV or other open-source codes for selecting control points [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.

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 (pinhole camera models), are related by a homography:

$$P_2 = HP_1$$

where H is a 3x3 homographic transformation matrix, $P1$ and $P2$ 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 determining matrix H, you can project all points from one image to another by following the backward mapping procedure and applying the interpolation technique.

Implement the above homographic transformation and stitching techniques to composite the scenery images in Figure 4. Show the results and discuss the following.

(1) How many control points do you need? Show the control points between the left and the center pair and between the center and the right pair.

(2) How did you select control points? Specify how you use matched features from SURF/SIFT to create a panorama.

Figure 4. The scenery images (left, center, right) (Photographed by Qingyang Zhou)

Problem 3: Morphological processing (50%)

You need to implement the “thinning” morphological operator on your own in this problem. A pattern table (patterntables.pdf) is attached for your reference. However, you can use Matlab or OpenCV functions for shrinking and skeletonizing operators if needed. Please show outputs for all the 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. Please binarize them before doing morphological processing. Binarize the image in the following way. Suppose the maximum pixel intensity is F_{max}. Then, any location with a value greater than 0.5 F_{max} is labeled 1. Otherwise, it is labeled as 0.

(a) Basic morphological process implementation (15%)

Please apply the “thinning” filter to the pattern images (spring, flower, and jar). Show all intermediate and final converged results (around the 20th iteration). Discuss your observations.

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

(b) Shape detection and counting (20%)

Figure 6 (see below) is target image with circle and square objects (the background is black and the objects are white). Some of these objects have one or more holes in them. You need to design an algorithm that uses morphological and logical operations to solve and answer the questions below. You may use any of these operators that you learned in class or ones that you invent, but all of them must be detailed in your report with respect to how they operate and how their results help you solve the problem. In your report, discuss your algorithm, results, and analysis in detail.:

(1) Find the total number of holes (black circular holes within white objects) in the image.

(2) Find the total number of white objects in the image.

(3) Find the total number of white rectangle objects (with or without holes) in the image.

(4) Find the total number of white circle objects (with or without holes) in the image.

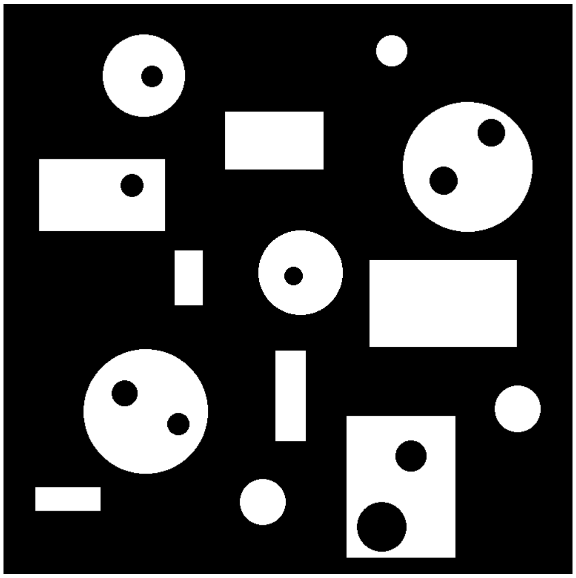

Figure 6: ‘board.raw’ image.

(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 a grayscale image and then apply morphological processes. Some pre-processing or post-process may be required to deal with unwanted dots and holes.

Appendix:

Problem 1&2: Geometric Image Modification

dog.raw	328x328	24-bit	color (RGB)
cat.raw	328x328	24-bit	color (RGB)
toys_left.raw	605x454	24-bit	color (RGB)
toys_middle.raw	605x454	24-bit	color (RGB)
toys_right.raw	605x454	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
board.raw	768x768	8-bit	greyscale
beans.raw	494x82	24-bit	color (RGB)

Note: “494x82” means “width=494, height=82”.

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