EE569 Digital Image Processing

HOMEWORK #3

Issued: 02/13/2022   Due: 11:59PM, 03/10/2022

Problem 1: Geometric Image Modification (25%)

1.1 Motivation

Geometric image modification means changing the geometry of an image. A set of image transformations where the geometry of image is changed without altering its actual pixel values are commonly referred to as "Geometric" transformation. In general, you can apply multiple operations on it, but, the actual pixel values will remain unchanged. In these transformations, pixel values are not changed, the positions of pixel values are changed.

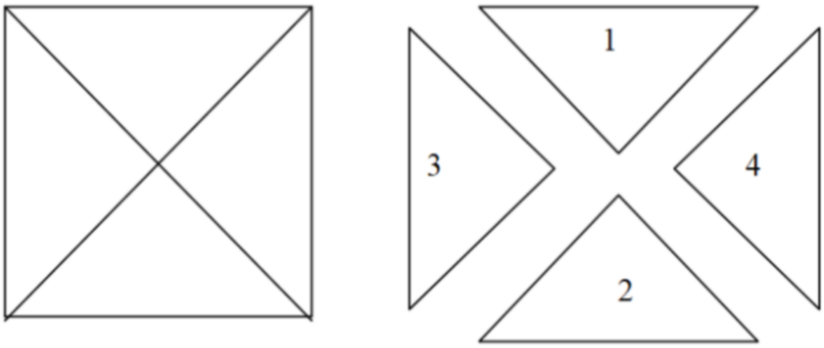
Image warping is the technique of digitally modifying an image so that any shape displayed in the image is extremely lossy. Warping can be used to compensate for picture loss and for some artistic objectives (e.g., distortion). While pictures can be changed in a variety of ways, pure image distortion indicates a point-to-point mapping without modifying the color. This can be based on any mathematical function from (at least partially) one plane to another. The original file can be rebuilt if the function is one-shot, and the picture can be changed inversely if the function is two-way one-shot. Here are a few options for changing this: Simultaneous distortion and transformation of the picture is possible. Consider the picture to be projected onto a curved or mirrored reflecting surface. The image may be broken into polygons, each of which can be deformed.

In the image warping process, we can choose at least two ways to achieve the image change: (forward mapping) a direct mapping application from the source to the image. (Inverse mapping) For a given mapping from source to image, the source can be found in the image.

1.2 Approach

Step 1: Normalize the image so that the coordinate values of the original image become between -1 and 1. This is a linear change, making the one with the smallest index become -1, making the one with the largest index become 1, making the one with the middlemost index value become 0, and then the other indexes are mapped evenly between minus one and 1.

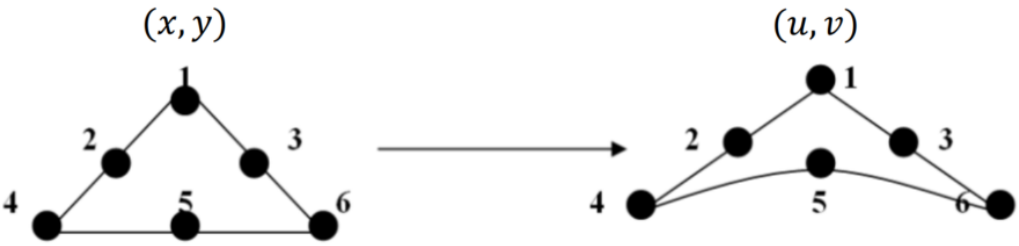
Step 2: Input image is divided into 4 triangles:



And for each triangle, the point after the image warping is set as (u, v), and the relationship with the original coordinates is:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 \\ b_0 & b_1 & b_2 & b_3 & b_4 & b_5 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \\ x^2 \\ xy \\ y^2 \end{bmatrix}$$

There are 12 unknown parameters and we find 6 pairs to solve the linear equation:



Step 3: We can get the value of 12 known parameters by inverse operation:

$$\begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 \\ b_0 & b_1 & b_2 & b_3 & b_4 & b_5 \end{bmatrix}$$

$$= \begin{bmatrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \\ v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{bmatrix} \Big/ \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ y_1 & y_2 & y_3 & y_4 & y_5 & y_6 \\ x_1^2 & x_2^2 & x_3^2 & x_4^2 & x_5^2 & x_6^2 \\ x_1y_1 & x_2y_2 & x_3y_3 & x_4y_4 & x_5y_5 & x_6y_6 \\ y_1^2 & y_2^2 & y_3^2 & y_4^2 & y_5^2 & y_6^2 \end{bmatrix}$$

Take Triangle 4 as an example, the 6 pairs points are as below:

(x, y)	(u, v)
(0, 0)	(0, 0)
(0.5, 0.5)	(0.5, 0.5)
(0.5, -0.5)	(0.5, -0.5)

(1, 1)	(1, 1)
(1, -1)	(1, -1)
(1, 0)	(0.609, 0)

Step 4: Modify the parameters according to the different triangles and then do the image warping respectively.

Step 5: After image warping, set the point of output image is (p, q), transfer image to original size.

Step 6: Recover warped image to original image through inverse address mapping.

1.3 Results

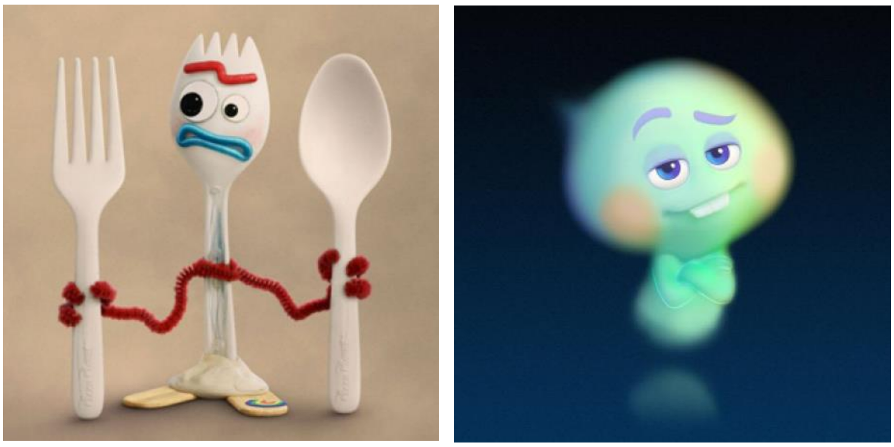


Figure 1: Original Image (Forky, 22)

(1) My approach is described in section 1.2. The resulting images are shown below:

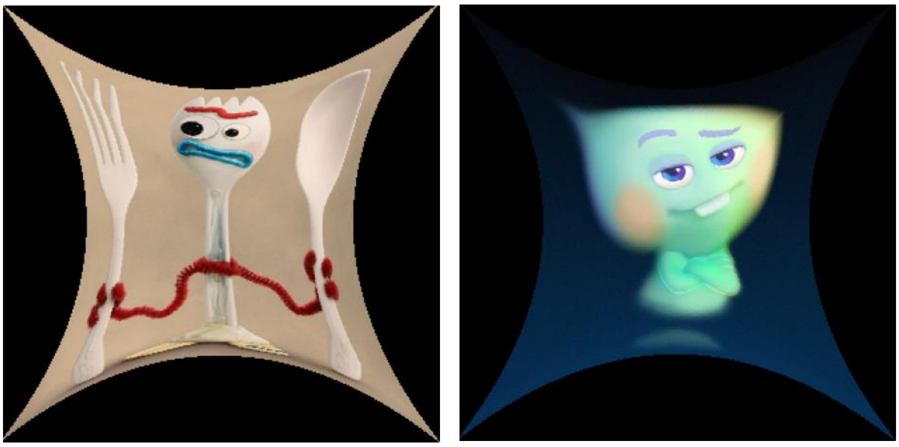


Figure 2: Star-shaped Image (Forky, 22)

(2)

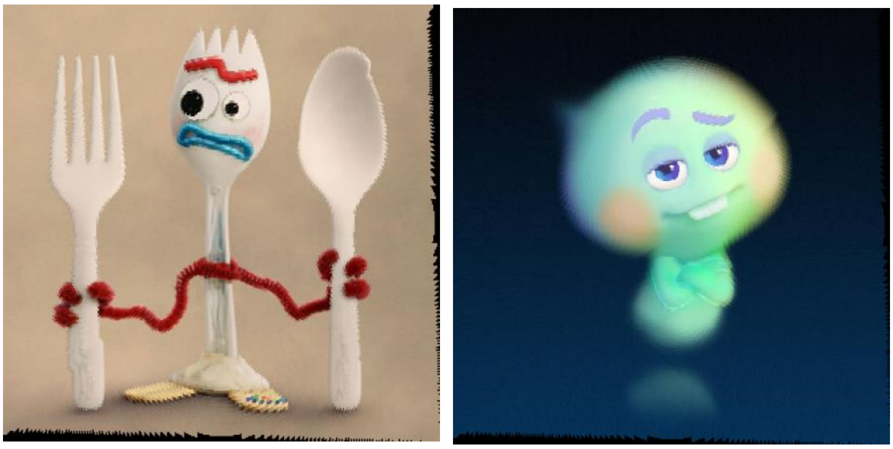


Figure 3: Reverse Image (Forky, 22)

(3) There is little difference. I use inverse address mapping, compared with forward address mapping, there is no black dots in warped image. But you can see there are some missing pixels at the edges of the image. I believe this is because I utilized basic and well-defined mathematical equations to produce a one-to-one mapping, and the extrema were occasionally unable to translate back to specific values.

1.4 Discussion

To image warping, there are forward warping and inverse warping. There are differences between forward warping and inverse warping. The result of forward warping, which is forward address mapping, will miss some pixels in target image, and warped images will have a lot of black dots. On the contrary, inverse warping, which is inverse address mapping, will not cause such a result. The resulting image is almost identical to the original image.

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

2.1 Motivation

Image stitching is a popular image processing method. Multiple photos with tiny views can be stitched into one large-view image via mutual matching of feature points, and it is utilized in wide-angle photo compositing, satellite photo processing, medical image processing, and other disciplines. The pixel value matching approach was widely utilized in early picture stitching. Later, people searched for stable characteristics like inflection points and edges in two different photos and patched them together using feature matching.

2.2 Approach

Step 1: Load all images, convert them to double grayscale image.

Step 2: Detect feature points in both images using Harris corner detector.

Step 3: Calculate descriptors for every key point in both images using SIFT extraction.

Step 4: Calculate the feature space distance between each pair of feature points in each two images.

Step 5: Select putative matches based on the distance matrix above.

Step 6: Run RANSAC algorithm, which apply homographic transformation to find a homograph mapping to find the model fitting the matches.

$$\begin{bmatrix} x_2' \\ y_2' \\ w_2' \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \frac{x_2'}{w_2'} \\ \frac{y_2'}{w_2'} \end{bmatrix}$$

Step 7: Generate the panorama and composite all images into it.

2.3 Results

Figure 4: Scenery Image

(1) I used 200 control points by RANSAC algorithm. Yellow points are control points. Blue points are the corresponding control points.

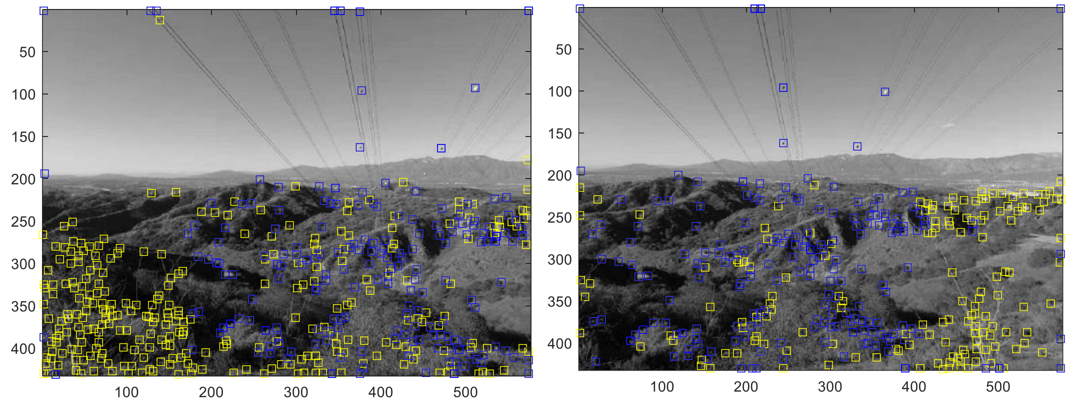


Figure 5: Left and Middle Control Points

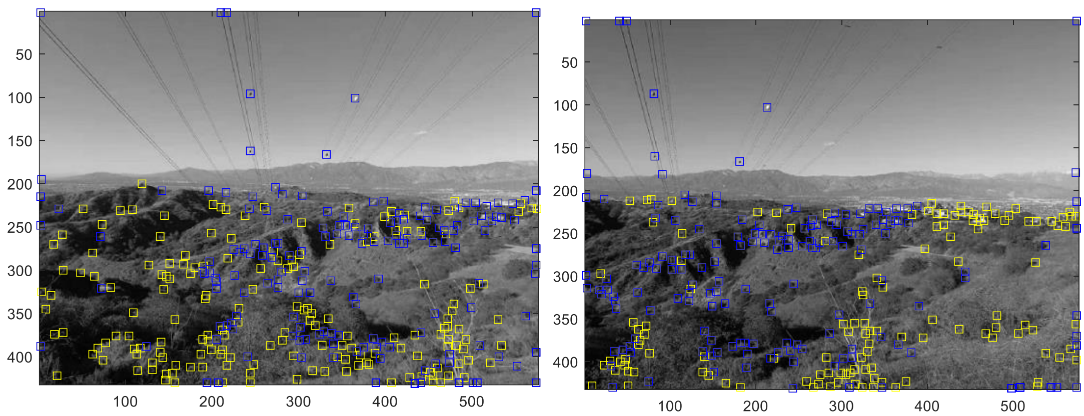


Figure 6: Middle and Right Control Points

(2) As I describe in section 2.2, I applied Harris corner detection function to find control

points. After I got matched feature from SIFT, I calculated the feature space distance between each pair of feature points in each two images. Then I selected putative matches based on the distance matrix above. After that, I ran RANSAC algorithm, and there are 200 matches for the RANSAC algorithm to fit and RANSAC iterates 5000 times to optimize the transformation. Finally, I generated the panorama and composited all images into it.

2.4 Discussion

The Random Sample Consensus (RANSAC) technique is used to match the critical points. Using one image as a guide, 8 points are randomly picked from it at a time, and the corresponding 8 points are discovered in the other image. The 8 pairs of points are used to generate a homography, and the remaining feature points in the base picture are projected onto the other image using the homography transformation, and the number of paired points is tallied. To obtain the homography with the most precise pairings, the preceding stages are repeated 2000 times.

We would obtain a number of crucial points after utilizing the Harris corner detection tool. To choose a certain number of essential points on merit, we may utilize the Adaptive Non-Maximal Suppression (ANMS) approach.

Problem 3: Morphological processing (50%)

3.1 Motivation

Morphology, also called mathematical morphology, is a widely used image processing technique for extracting image components from images that are meaningful for expressing and depicting the shape of regions, so that subsequent recognition can capture the most important (most discriminative) shape features of the target object, such as boundaries and connected regions. Discriminative shape characteristics include distinguishing features such as boundaries and adjacent regions. Picture pre-processing and post-processing methods such as refining, pixelation, and burr trimming are frequently utilized as a potent supplement to image enhancement approaches.

3.2 Approach

3.2.1 Basic morphological process implementation

Step 1: Binarize the image before morphological processing.

Step 2: Generate patterns according to Hit Masks Table 14.3-1, Table 14.3-2.

TABLE 14.3-1. Shrink, Thin and Skeletonize Conditional Mark Patterns [$M = 1$ if hit]

Table	Bond	Pattern							
S	1	0 0 1 0 1 0 0 0 0	1 0 0 0 1 0 0 0 0	0 0 0 0 1 0 1 0 0	0 0 0 0 1 0 0 0 1				
S	2	0 0 0 0 1 1 0 0 0	0 1 0 0 1 0 0 0 0	0 0 0 1 1 0 0 0 0	0 0 0 0 1 0 0 1 0				
S	3	0 0 1 0 1 1 0 0 0	0 1 1 0 1 0 0 0 0	1 1 0 0 1 0 0 0 0	1 0 0 1 1 0 0 0 0	0 0 0 1 1 0 1 0 0	0 0 0 0 1 0 1 1 0	0 0 0 0 1 0 0 1 1	0 0 0 0 1 1 0 0 1
TK	4	0 1 0 0 1 1 0 0 0	0 1 0 1 1 0 0 0 0	0 0 0 1 1 0 0 1 0	0 0 0 0 1 1 0 1 0				
STK	4	0 0 1 0 1 1 0 0 1	1 1 1 0 1 0 0 0 0	1 0 0 1 1 0 1 0 0	0 0 0 0 1 0 1 1 1				

(Continued)

Table	Bond	Pattern							
ST	5	1 1 0 0 1 1 0 0 0	0 1 0 0 1 1 0 0 1	0 1 1 1 1 0 0 0 0	0 0 1 0 1 1 0 1 0				
ST	5	0 1 1 0 1 1 0 0 0	1 1 0 1 1 0 0 0 0	0 0 0 1 1 0 1 1 0	0 0 0 0 1 1 0 1 1				
ST	6	1 1 0 0 1 1 0 0 1	0 1 1 1 1 0 1 0 0						
STK	6	1 1 1 0 1 1 0 0 0	0 1 1 0 1 1 0 0 1	1 1 1 1 1 0 0 0 0	1 1 0 1 1 0 1 0 0	1 0 0 1 1 0 1 1 0	0 0 0 1 1 0 1 1 1	0 0 0 0 1 1 1 1 1	0 0 1 0 1 1 0 1 1

TABLE 14.3-1. (Continued)

Table	Bond	Pattern							
		1 1 1	1 1 1	1 0 0	0 0 1				
STK	7	0 1 1	1 1 0	1 1 0	0 1 1				
		0 0 1	1 0 0	1 1 1	1 1 1				
		0 1 1	1 1 1	1 1 0	0 0 0				
STK	8	0 1 1	1 1 1	1 1 0	1 1 1				
		0 1 1	0 0 0	1 1 0	1 1 1				
		1 1 1	0 1 1	1 1 1	1 1 1	1 1 1	1 1 0	1 0 0	0 0 1
STK	9	0 1 1	0 1 1	1 1 1	1 1 1	1 1 0	1 1 0	1 1 1	1 1 1
		0 1 1	1 1 1	1 0 0	0 0 1	1 1 0	1 1 1	1 1 1	1 1 1
		1 1 1	1 1 1	1 1 1	1 0 1				
STK	10	0 1 1	1 1 1	1 1 0	1 1 1				
		1 1 1	1 0 1	1 1 1	1 1 1				
		1 1 1	1 1 1	1 1 0	0 1 1				
K	11	1 1 1	1 1 1	1 1 1	1 1 1				
		0 1 1	1 1 0	1 1 1	1 1 1				

TABLE 14.3-2. Shrink and Thin Unconditional Mark Patterns
$[P(M, M_0, M_1, M_2, M_3, M_4, M_5, M_6, M_7) = 1$ if hit$]^a$

Pattern							
Spur		Single 4-connection					
0 0 *M*	*M* 0 0	0 0 0	0 0 0				
0 *M* 0	0 *M* 0	0 *M* 0	0 *M M*				
0 0 0	0 0 0	0 *M* 0	0 0 0				
L Cluster							
0 0 *M*	0 *M M*	*M M* 0	*M* 0 0	0 0 0	0 0 0	0 0 0	0 0 0
0 *M M*	0 *M* 0	0 *M* 0	*M M* 0	*M M* 0	0 *M* 0	0 *M* 0	0 *M M*
0 0 0	0 0 0	0 0 0	0 0 0	*M* 0 0	*M M* 0	0 *M M*	0 0 *M*
4-Connected offset							
0 *M M*	*M M* 0	0 *M* 0	0 0 *M*				
M M 0	0 *M M*	0 *M M*	0 *M M*				
0 0 0	0 0 0	0 0 *M*	0 *M* 0				
Spur corner cluster							
0 *A M*	*M B* 0	0 0 *M*	*M* 0 0				
0 *M B*	*A M* 0	*A M* 0	0 *M B*				
M 0 0	0 0 *M*	*M B* 0	0 *A M*				

(Continued)

Corner cluster

M M D
M M D
D D D

Tee branch

D M 0	0 M D	0 0 D	D 0 0	D M D	0 M 0	0 M 0	D M D
M M M	M M M	M M M	M M M	M M 0	M M 0	0 M M	0 M M
D 0 0	0 0 D	0 M D	D M 0	0 M 0	D M D	D M D	0 M 0

Vee branch

M D M	M D C	C B A	A D M
D M D	D M B	D M D	B M D
A B C	M D A	M D M	C D M

Diagonal branch

D M 0	0 M D	D 0 M	M 0 D
0 M M	M M 0	M M 0	0 M M
M 0 D	D 0 M	0 M D	D M 0

[a] $A \cup B \cup C = 1$ $D = 0 \cup 1$ $A \cup B = 1.$

Step 3: Apply pattern obtained from Table 14.3-1 on image. If match, mark the value on mask to 1.

Step 4: Apply pattern obtained form Table 14.3-2 on Mask. If match, keep the pixel in image. If not match, change pixel to 0 in image.

Step 5: Repeat step 3 & 4, until the desired results are obtained.

3.2.2 Defect detection and counting

Step 1: Binarize the image before morphological processing.

Step 2: Inverse the image.

Step 3: Do shrinking.

Step 4: Check and record holes and background, which is white dots now.

Step 5: Calculate the size (number of pixels) of white dots by BFS method.

0	1	0	1	1
1	1	1	0	0
1	1	0	0	1
0	1	0	1	1

Step 6: Judge whether the white dots are defects.

Step 7: Record the number of defects and their sizes.

Step 8: Correct the defects and inverse the image to show clear image.

3.2.3 Object Segmentation and Analysis
Step 1: Turn color image to grayscale image, then binarize the image before morphological processing.

Step 2: Do shrinking, and turn holes in beans to black.

Step 3: Inverse the image.

Step 4: Compute the number of beans.

Step 5: Generate the segmentation mask and compute the sizes of different beans.

Step 6: Rank the beans' size from small to large and show the image.

3.3 Results
(a)

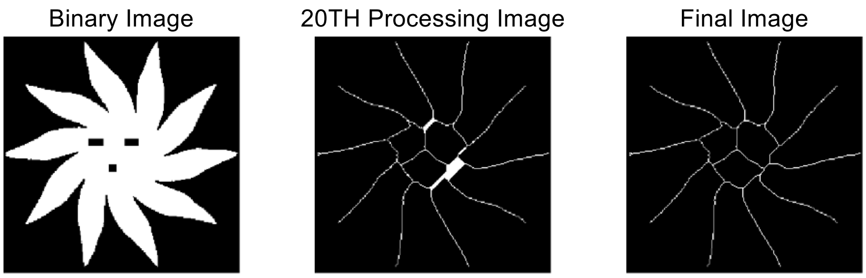


Figure 7: Flower Image

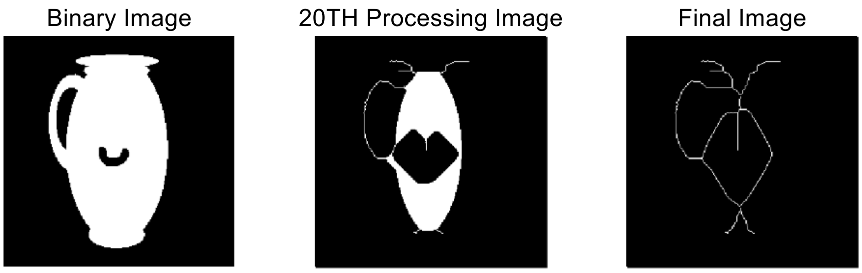


Figure 8: Jar Image

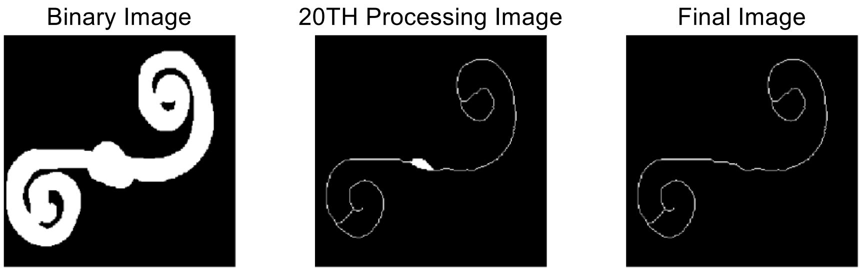


Figure 9: Spring Image

(b)
(1) The number of defects is 6.

(2) 2 different defect sizes.

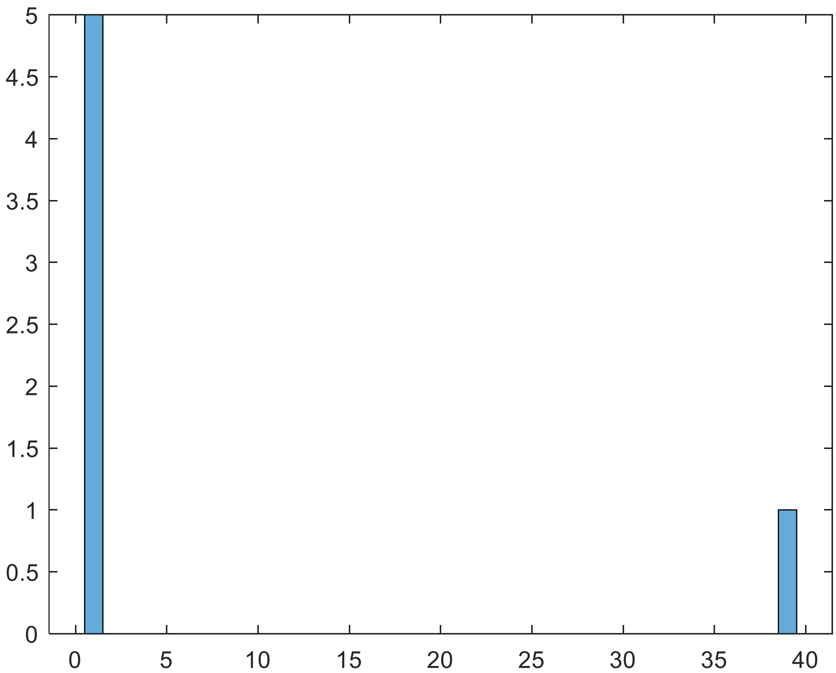


Figure 10: Frequency of Defect Sizes

(3)

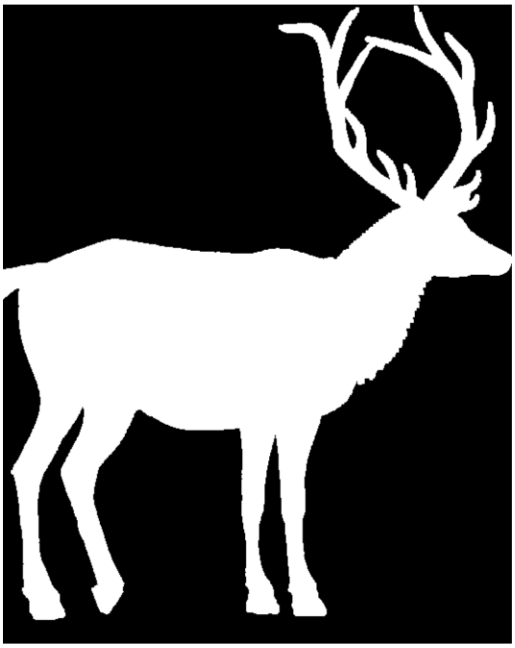


Figure 11: Clear Deer Image

(c)
(1) The total number of beans is 5.

(2)

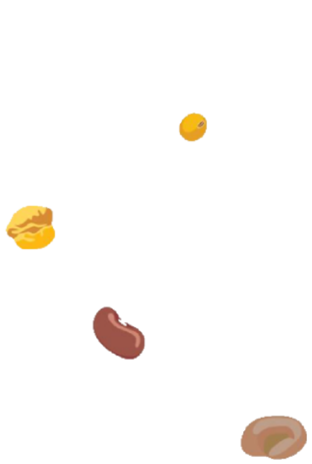


Figure 12: Sorted Beans

3.4 Discussion
Basic morphological process implementation is useful in almost morphological

processing operation. Morphological image processing is very effective when we need to find and correct defects in an image, or to obtain relevant information about the morphological structure of an image for the purpose of image analysis and recognition.