CSE585/EE555: Digital Image Processing

Computer Project 2

Homotopic Skeletonization and Shape Analysis

Yifei Xiao, Ling Zhang, Jiaming Chai

February 21, 2020

A Objectives

We conduct the following topics in this project:

1. The algorithm of homotopic skeleton and the process to implement it
2. Calculation for different parts for shape analysis and the qualitative results including: size distribution, pattern spectrum and complexity

B Methods

In this project, our images are all in binary. So, we do not need to convert those images as we did in the first project.

For the first part, we first implement the hit-or-miss of eight structuring elements, follow the steps as they are well defined in the handout. Our method is a little different than the general approach due to the special feature of the structuring elements: We pass the index of the structuring elements when the pixel value is 1 for foreground and background. This way, when we slide through the image, it would be easier to calculate the exact position of that pixel we are comparing to. This approach makes our code and converting process from back to fore ground easier. For the process of iterating through the image by the eight filters, we simply follow the pseudo code in the handout that has a stopping criteria when the new image is the same as the previous one. We compress the eight structuring elements into a tensor for easy data transfer between the functions and loops. For the image subtracting the result from hit-or-miss, we convert the image from "logical" to "double" and apply subtraction operation. The hit-or-miss is done by following the definition. Besides, we apply padding to the original image for easier index computation. The intersection is done by the logical operator.

For the second part, there are two images we take into consideration, and each image contains several objects that require image cropping for easier analysis of their shapes.

B.1 Homotopic skeleton

We introduce our method in details with codes in MATLAB. To give a general idea of how we implement our method, a flow chart is presented below.

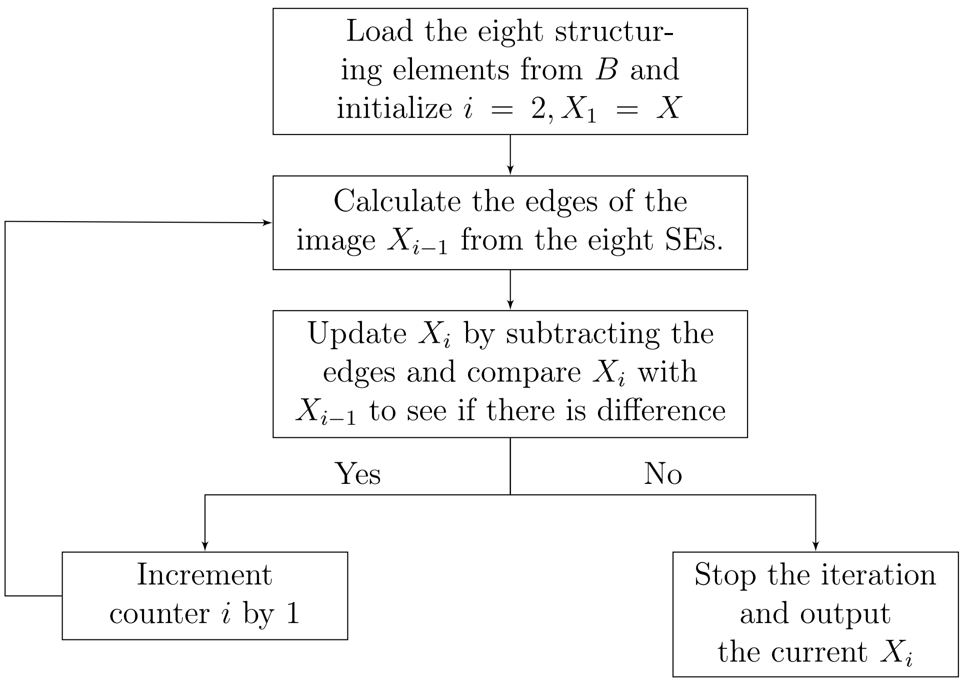

Figure 1: Homotopic skeleton method flow chart for our implementation.

B.1.1 sk_main.m

In this file, we construct the constructing elements B1 to B8. And run the homotopic skeleton algorithm recursively until the image cannot thinning anymore. The final image is the skeleton of the object in the image.

B.1.2 *homotopic_ skeleton.m*

We build eight structuring elements by taking the index of the 3 x 3 shape object when the pixel value indicates 1, and iterate through many steps which we will mention in next subsection in this MATLAB file. Recall that the equation of the thinning the object X by b is:

$$X \bigcirc B = X - X \circledast B_i$$

For here $X \circledast B$ is *Hit-or-Miss* operation equals $(X \ominus B_i^f) \cap (X^c \ominus B_i^b)$. In our project we have eight structuring elements. Each Hit-or-Miss operation, we separate one structuring element B_i into two structuring element, B_i^f for foreground and B_i^b for background. After *Hit-or-Miss*, we use previous object X_i to do a set subtraction. In this iteration, the object will burn off the first layer of the object X and preserving the homotopic of the object. Then we do the same step until the object cannot change anymore. The points left is the skeleton of the object.

B.2 Shape Analysis for 'match1' and 'match3'

B.2.1 cropImage.m

Use build-in 'bwconncomp' function to isolate the original image and mark separately.

B.2.2 isolateImage.m

Based on the given pixels, calculated the minimum bounding box of each objects.

B.2.3 complexity.m

First we do *Size Distribution*. We have the equations:

$$U(r) = m(X_r B), r \geq 0$$

X_rB means opening of X by rB, $m(X_rB)$ means the area of X_rB. For discrete case, the area of the object equals the number of pixels in the object.

The next is *Pattern Spectrum(Pecstrum)*:

For $r > 0$

$$f(r) = \frac{-\frac{du(r)}{dr}}{m(X)}$$

This function gives amount of are in X per component. Our project is discrete space, so:

$$\sum_{i=0}^{N} f(i) = 1$$

Then we can get the measure of *Shape Complexity* for X:

$$H(X|B) = -\sum_{i=0}^{N-1} f(i) \log f(i)$$

B.2.4 Shape_Analysis.m

This file combine the cropImage function, isolatedImage function, size distribution, pecstrum and complexity. We plot all the result for match 1 and match3 in this file.

B.2.5 patternRecognition.m

This file implement the distance algorithm which is the most important part to finding the matches. We calculate the distance d between each image in the 'match1' and each image in the 'match3', d can be written as

$$d_i = \left[\sum_{n=0}^{N-1} C_n \left(f(n) - f_{R_i}(n) \right)^2 \right]^{1/2} \tag{1}$$

where $f(n)$ is the pecstrum for image in 'match3' and $f_{R_i}(n)$ is the pecstrum for image in 'match1'.

The shortest distance between two images will be considered as match, weights C_n is set to $[1.0, 0.8, 0.6, 0.4, 0.2, 0.1, 0.1, 0.1, 0.1, 0.1]$.

B.3 Shape Analysis for 'shadow1' and automatically match pairs. Shape_Analysis2.m

Similar to the previous section which we did 'automatically' match step by step. The main idea that automatically match object is that calculate the separate objects' complexity and pair two objects with the smallest distance.

C Results

In this section, we show all the results of our work.

C.1 Homotopic Skeletonization

C.1.1 penn256

The original image is shown in Figure 2. After we apply thinning on this image, we can get results from Figure 3 to Figure 5, which show the intermediate X_2, X_5, X_{10} step of thinning. Figure 6. The pink background shows the original graph and the white foreground shows the skeletonized image. Notice that X_{10} is the same as the final skeletonized image, because the thinning process has converged at the 7^{th} step. Figure 7 shows the skeleton of original image.

Figure 2: Original penn256.

Figure 3: X_2 of thinning.

Figure 4: X_5 of thinning.

Figure 5: X_{10} of thinning.

Figure 6: Final result of thinning with original image.

Figure 7: Skeleton of original image.

C.1.2 bear

Similarly, we apply the same process to *bear.gif*. The original image is shown in Figure 8. After we apply thinning on this image, we can get results from Figure 9 to Figure 11, which show the intermediate X_2, X_5, X_{10} step of thinning. Figure 12. The pink background shows the original graph and the white foreground shows the skeletonized image. In this case, *bear.gif* needs 22 steps to converge, which is longer than *penn256.gif*. Figure 13 shows the skeleton of original image.

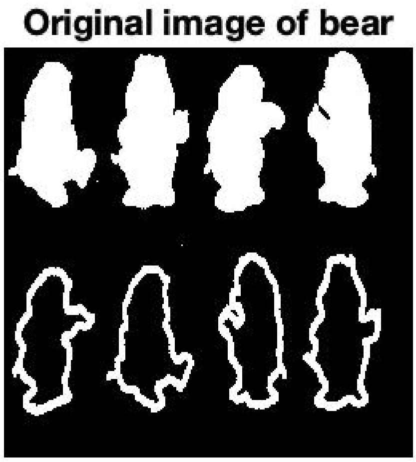

Figure 8: Original bear.

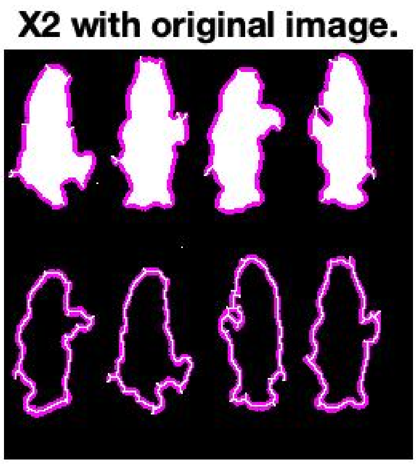

Figure 9: X_2 of thinning.

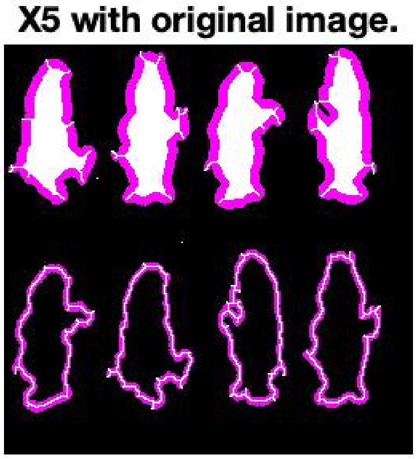

Figure 10: X_5 of thinning.

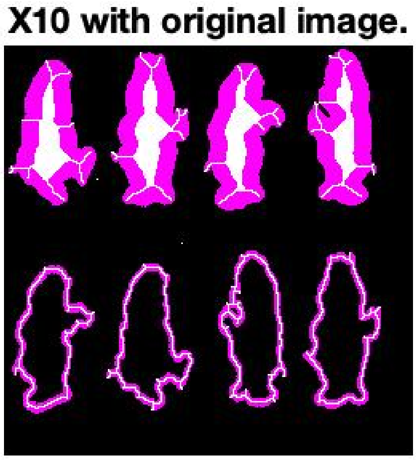

Figure 11: X_{10} of thinning.

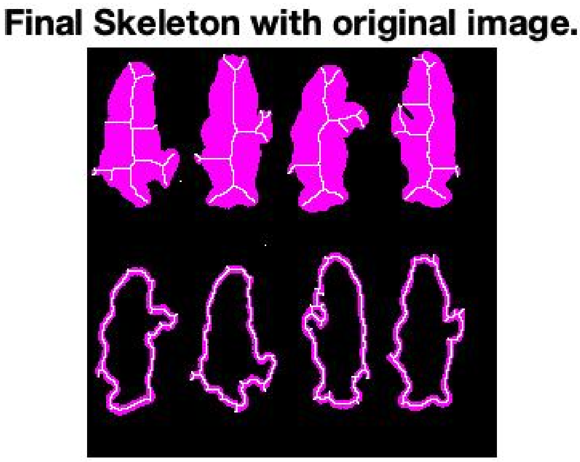

Figure 12: Final result of thinning with original image.

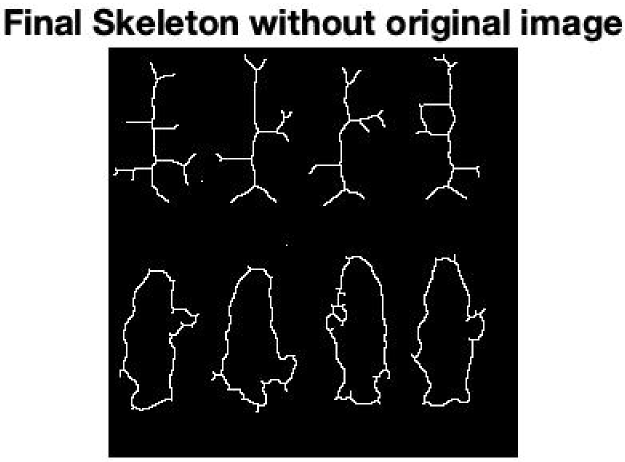

Figure 13: Skeleton of original image.

C.2 Shape Analysis

C.2.1 match1 and match3

The original 'match1' is shown in Figure 14. First we cropped objects from the original picture and the edges are the minimum bounding boxes of each object. Figure 15. Then we calculate the size distribution, pectrum and complexity for each object separately. For Clover, size distribution shows in Figure 16, Pectrum shows in Figure 17 and complexity is 0.681563. For Steer, size distribution shows in Figure 18, Pectrum shows in Figure 19 and complexity is 0.850099. For Spade, size distribution shows in Figure 20, Pectrum shows in Figure 21 and complexity is 0.710317. For Airplane, size distribution shows in Figure 22, Pectrum shows in Figure 23 and complexity is 0.697545. Table 1 shows the summary of complexity of each image.

Image	**Complexity**
Clover	0.681563
Steer	0.850099
Spade	0.710317
Airplane	0.697545

Table 1: Complexity of original image.

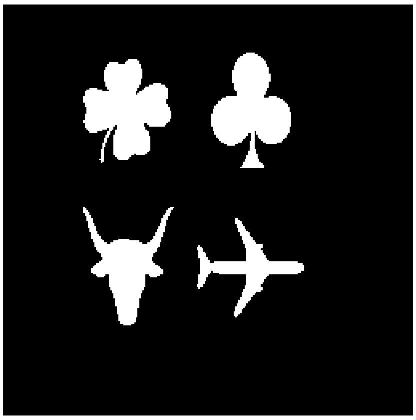

Figure 14: Original match1.

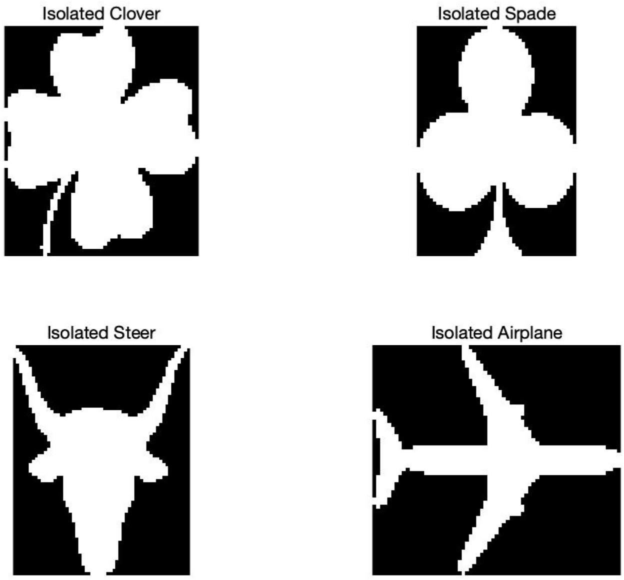

Figure 15: Isolated Object.

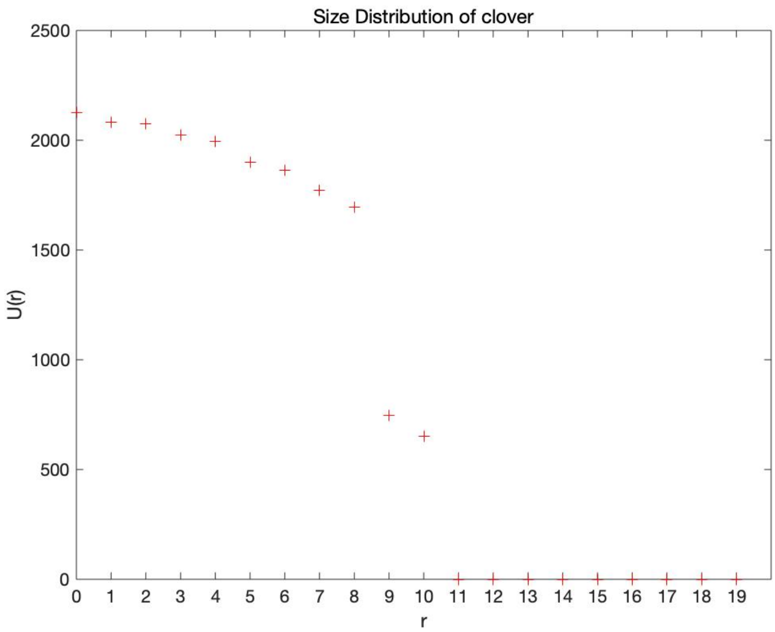

Figure 16: Size Distribution of Clover.

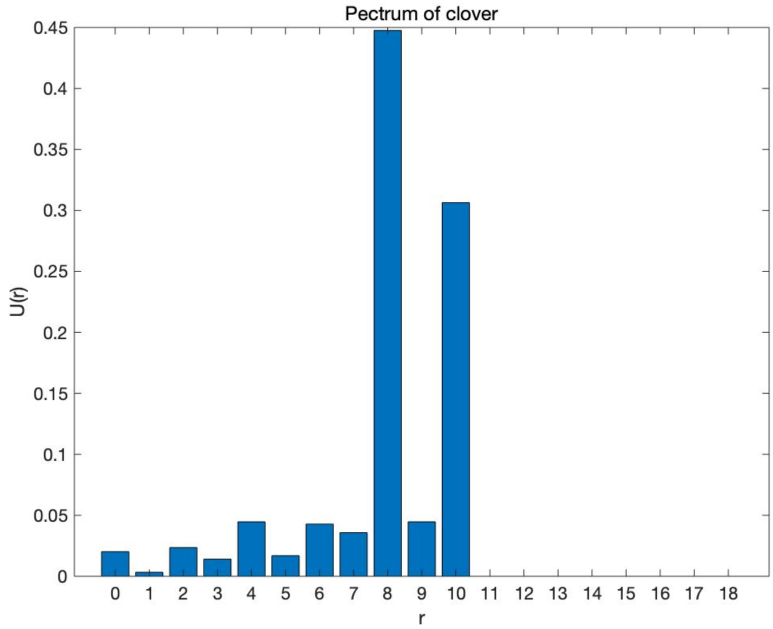

Figure 17: Pectrum of Clover.

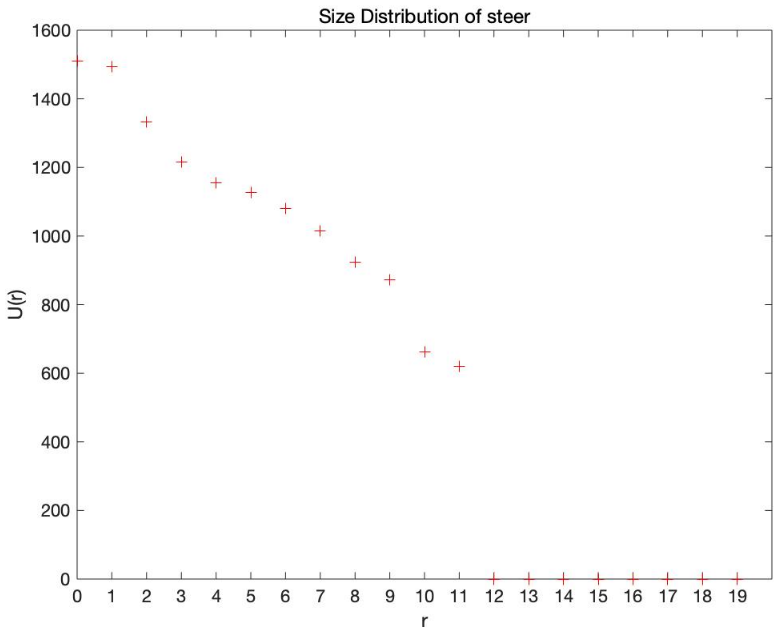

Figure 18: Size Distribution of Steer.

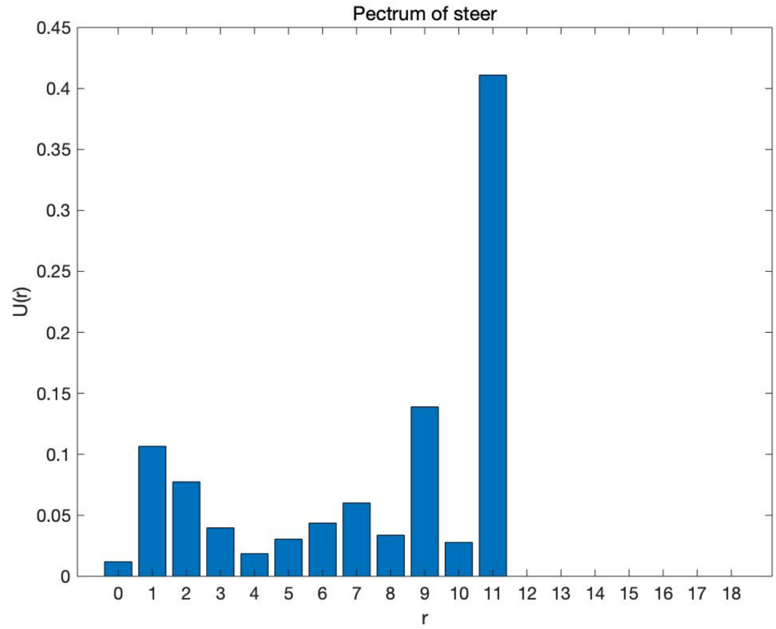

Figure 19: Pectrum of Steer.

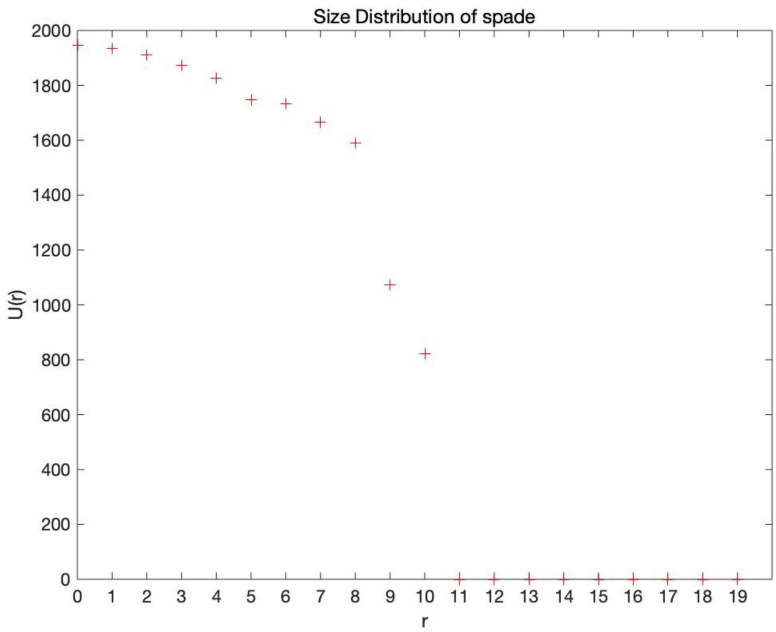

Figure 20: Size Distribution of Spade.

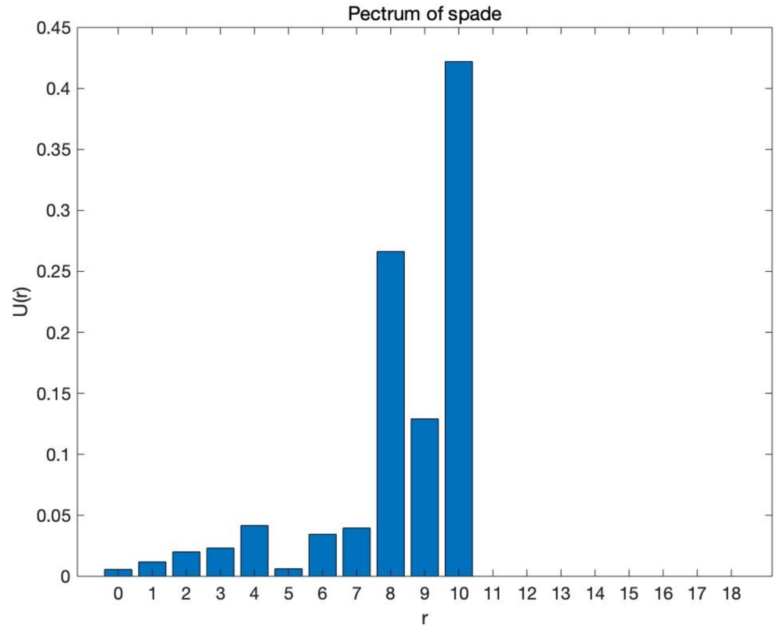

Figure 21: Pectrum of Spade.

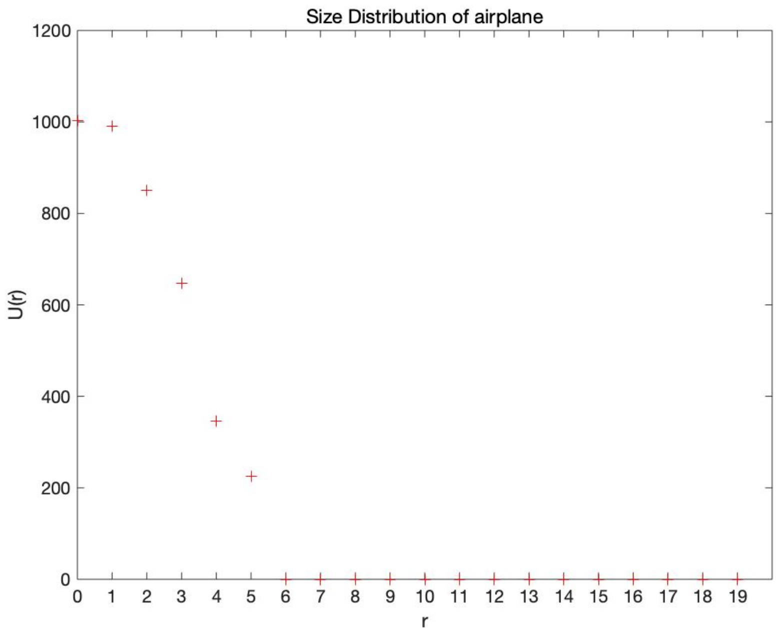

Figure 22: Size Distribution of Airplane.

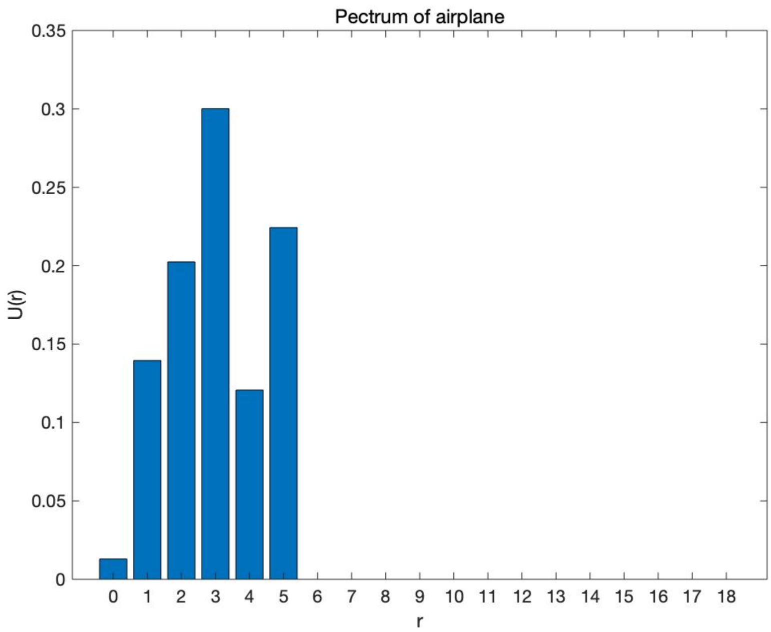

Figure 23: Pectrum of Airplane.

The original 'match3' is shown in Figure24 which is rotated a little from 'match1'. Same as above we cropped objects from the original picture and the edges are the minimum bounding boxes of each object. Figure 25. Then we calculate the size distribution, pectrum and complexity for each object separately. For rotated Clover, size distribution shows in Figure 26, Pectrum shows in Figure 27 and complexity is 0.726108. For rotated Steer, size distribution shows in Figure 28, Pectrum shows in Figure 29 and complexity is 0.863044. For rotated Spade, size distribution shows in Figure 30, Pectrum shows in Figure 31 and complexity is 0.721768. For rotated Airplane, size distribution shows in Figure 32, Pectrum shows in Figure 33 and complexity is 0.618795. Table 2 shows the summary of complexity of each image.

Image	**Complexity**
Rotated Clover	0.726108
Rotated Steer	0.863044
Rotated Spade	0.721768
Rotated Airplane	0.618795

Table 2: Complexity of rotated image.

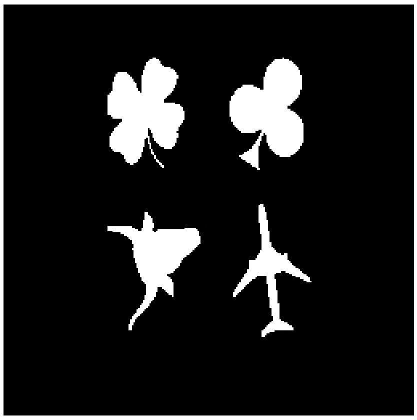

Figure 24: Original match3.

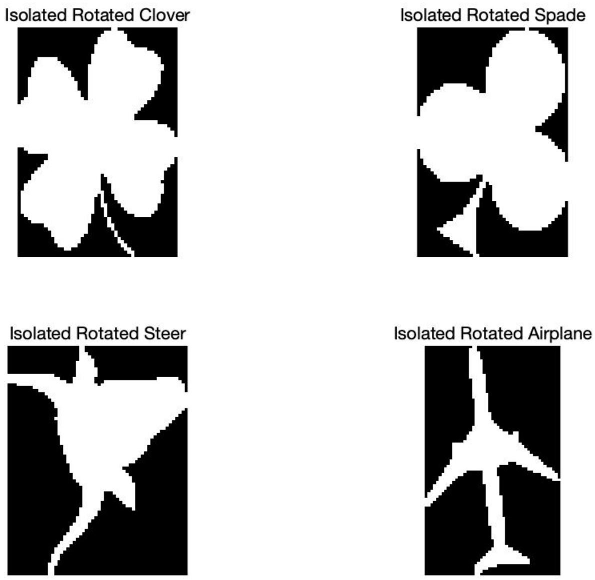

Figure 25: Isolated Rotated Object.

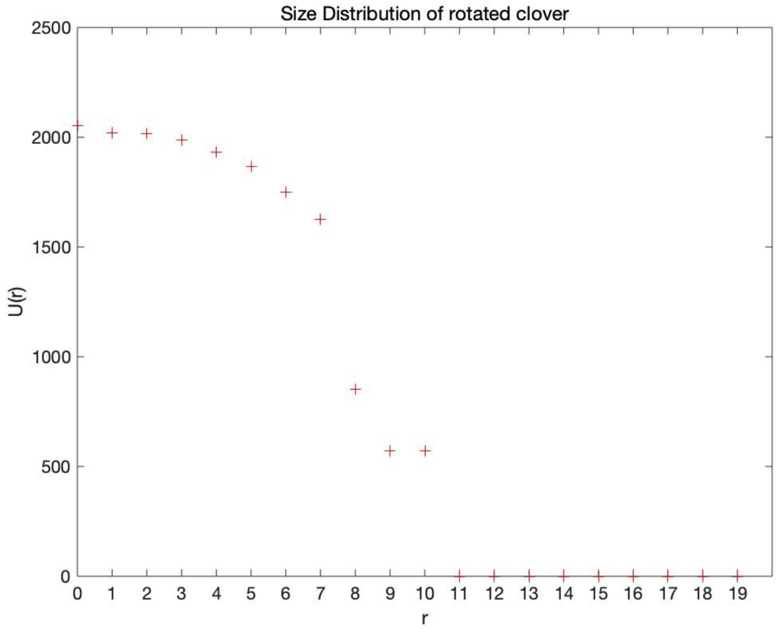

Figure 26: Size Distribution of Rotated Clover.

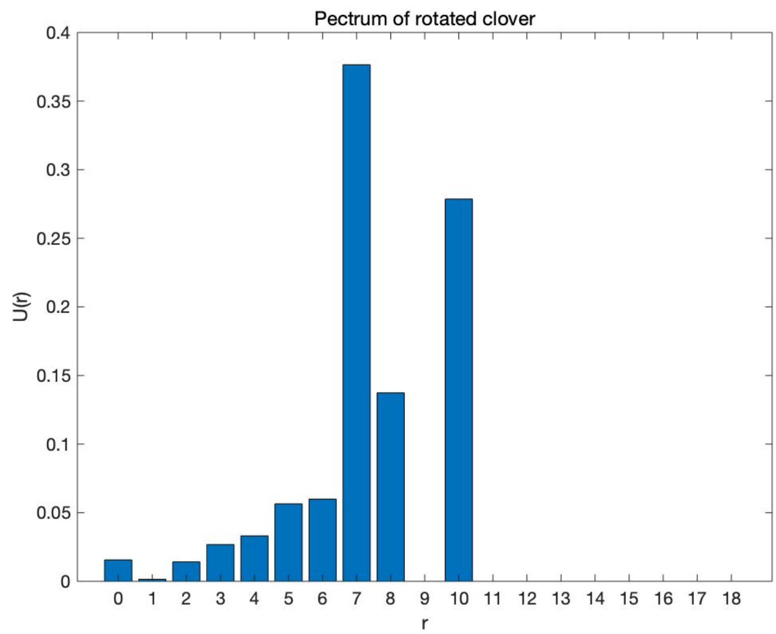

Figure 27: Pectrum of Rotated Clover.

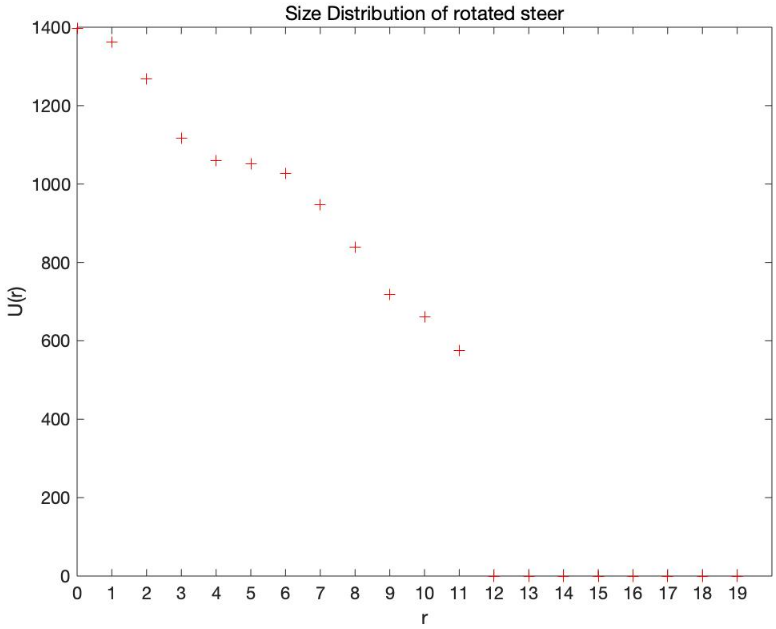

Figure 28: Size Distribution of Rotated Steer.

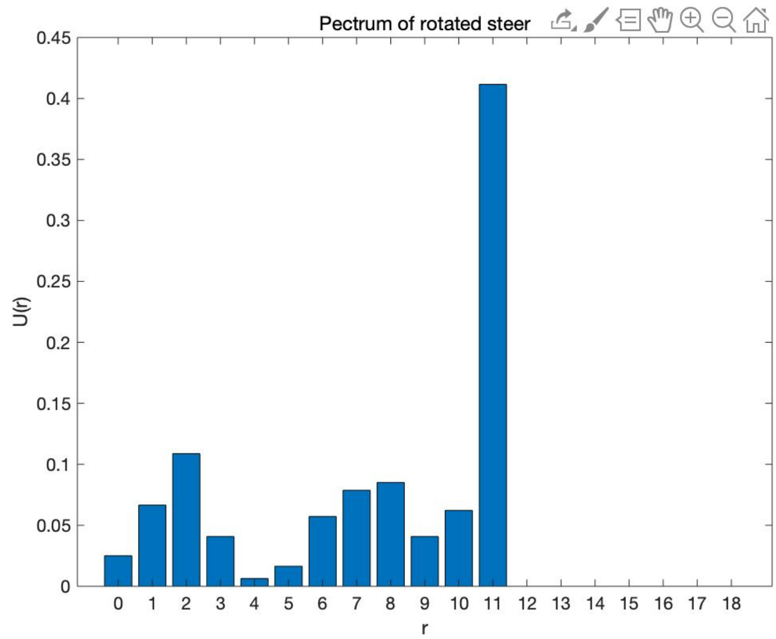

Figure 29: Pectrum of Rotated Steer.

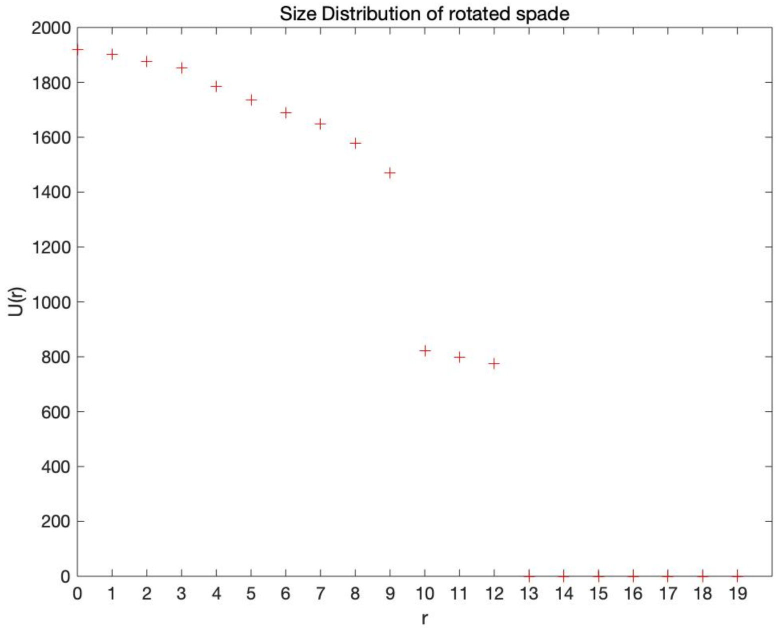

Figure 30: Size Distribution of Rotated Spade.

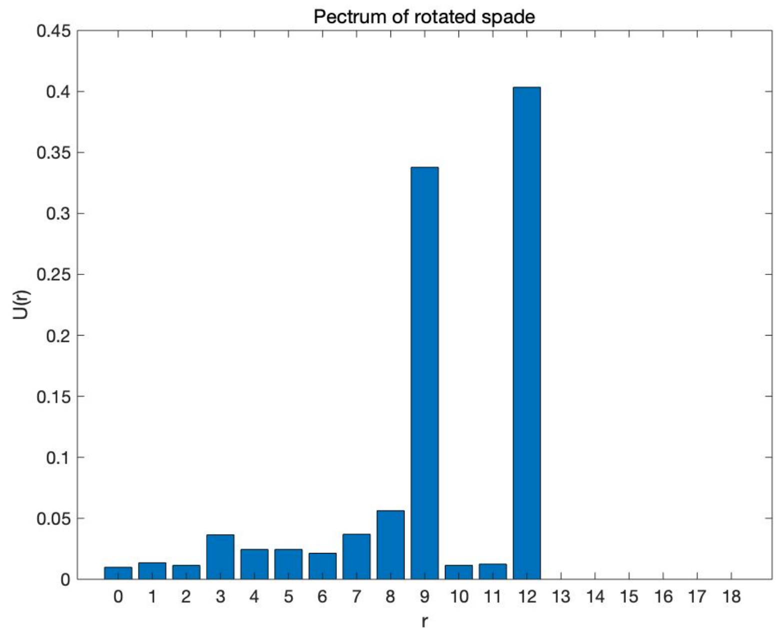

Figure 31: Pectrum of Rotated Spade.

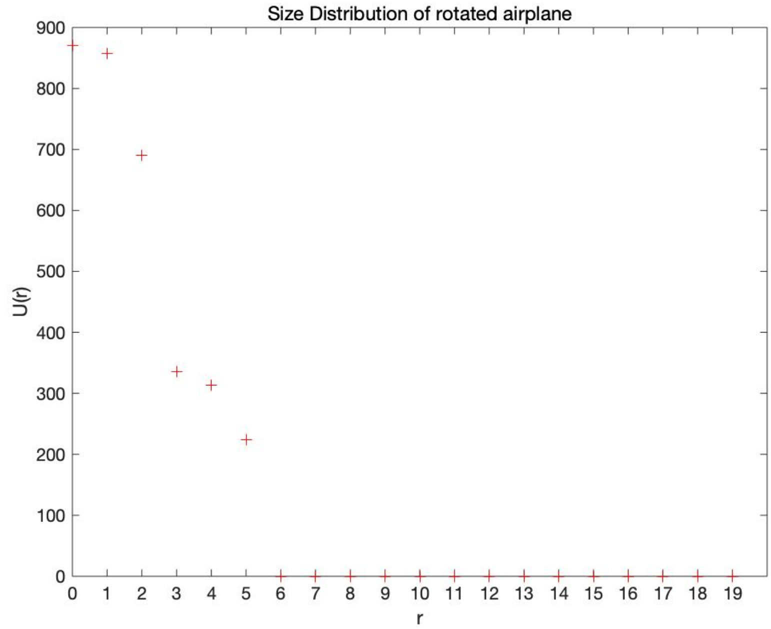

Figure 32: Size Distribution of Rotated Airplane.

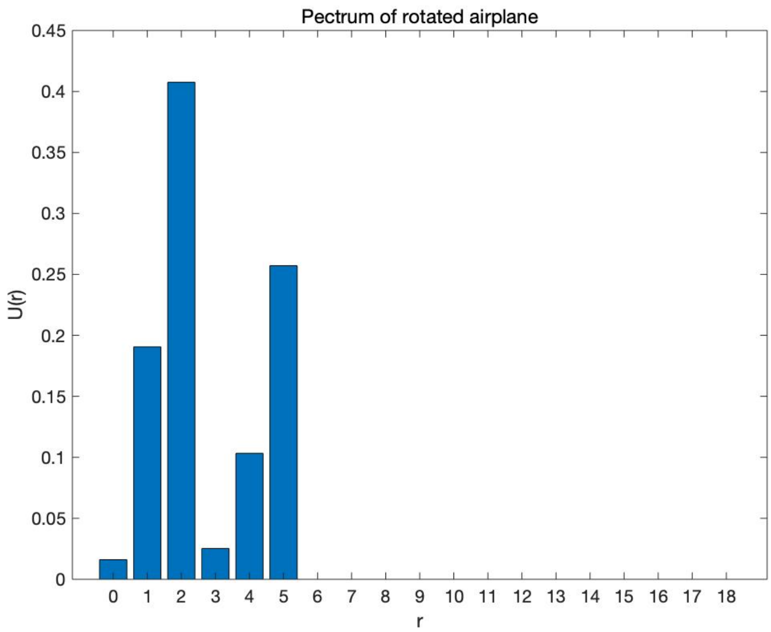

Figure 33: Pectrum of Rotated Airplane.

Then we use function that we defined to calculate the distance between each pairs, and we choose the closed pair and get the result for each objects in 'match1'. The distance d_i among each image are shown in Table 3. As we can see, all images in 'match1' get corresponding matches in 'match3'. The pair results are in Figure 34, Figure 35, Figure 36 and Figure 37.

Distance	Clover	Steer	Spade	Airplane
Rotated Clover	**0.264660**	0.295584	0.278019	0.758905
Rotated Steer	0.351512	**0.141182**	0.227351	0.541294
Rotated Spade	0.242022	0.272068	**0.171750**	0.727226
Rotated Airplane	0.718510	0.513533	0.679728	**0.399621**

Table 3: Distance among original images and rotated images.

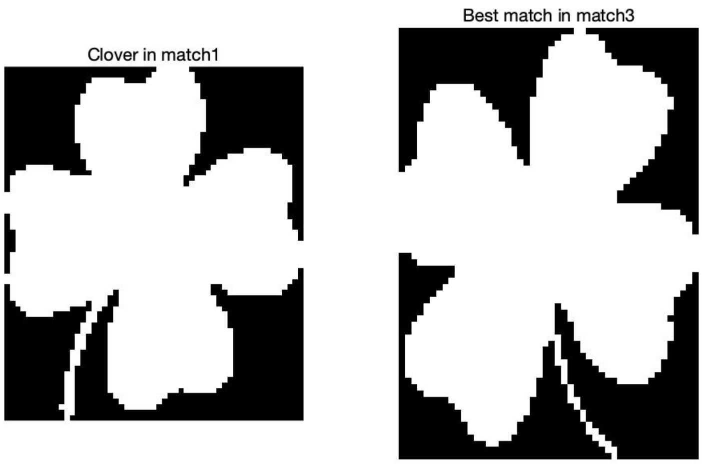

Figure 34: The pair of clovers.

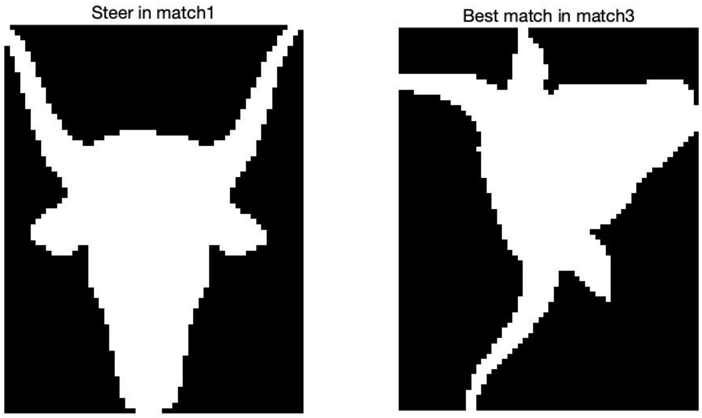

Figure 35: The pair of steers.

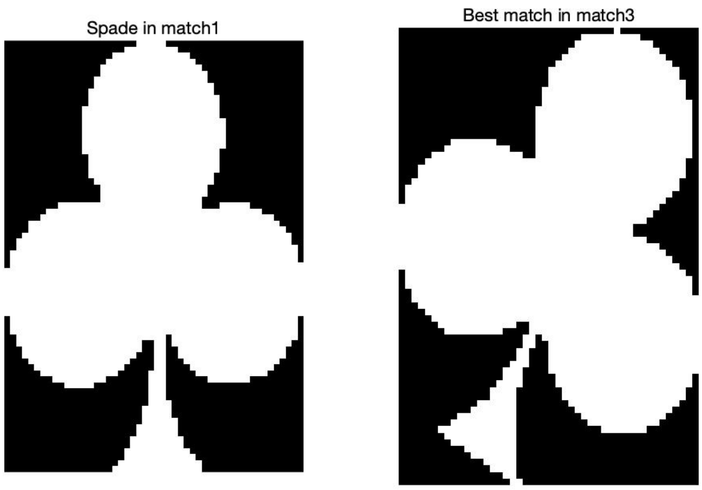

Figure 36: The pair of spades.

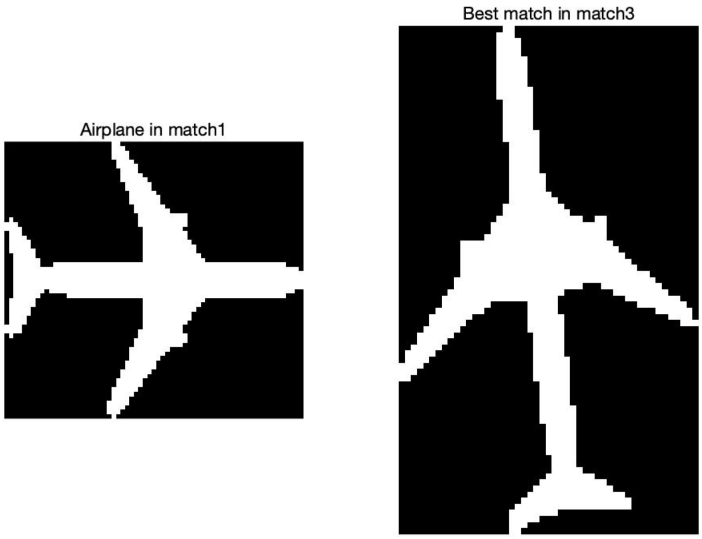

Figure 37: The pair of Airplanes.

C.2.2 shadow1 and shadow3

We can analyze 'shadow1' and 'shadow3' in a similar way. The original image is in Figure 38. As shown in 39, from left to right we label each of them as 'black1', 'black2', 'black3', 'black4' and 'white1', 'white2', 'white3', 'white4'. Figure 40to Figure 55 show their corresponding size distribution and pectrum. The summary of complexity is shown in Table 4.

Image	**Complexity**
Black1	0.327779
Black2	0.419872
Black3	0.208678
Black4	0.415503
White1	0.732552
White2	0.791118
White3	0.668677
White4	0.763048

Table 4: Complexity of original image.

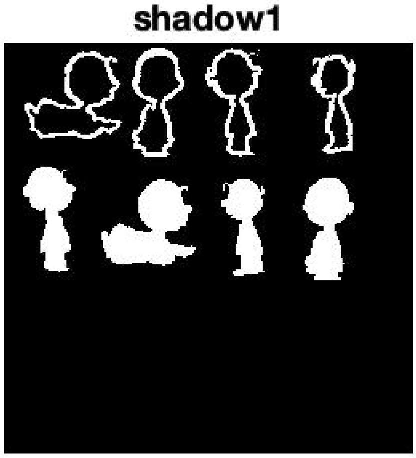

Figure 38: Original image of shadow1.

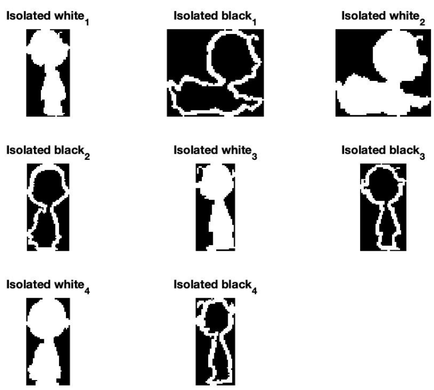

Figure 39: Isolated shadow1.

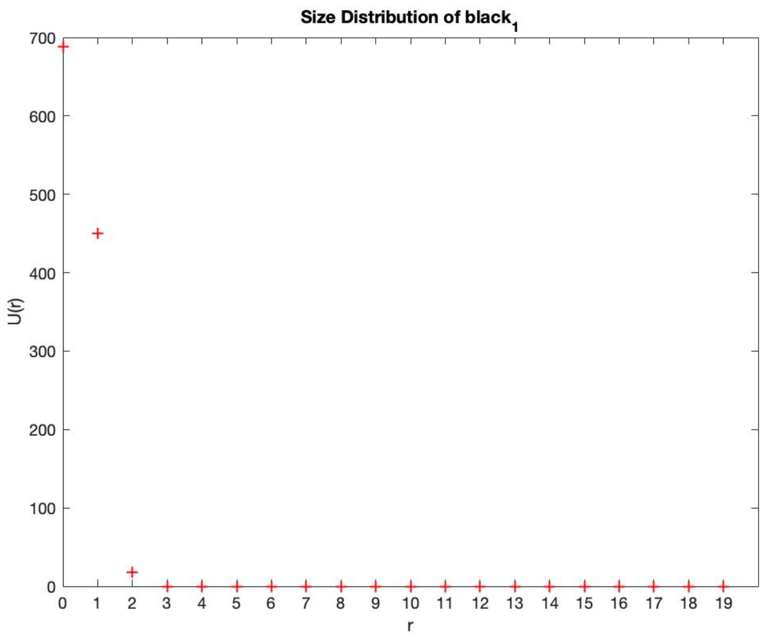

Figure 40: Size distribution of black1.

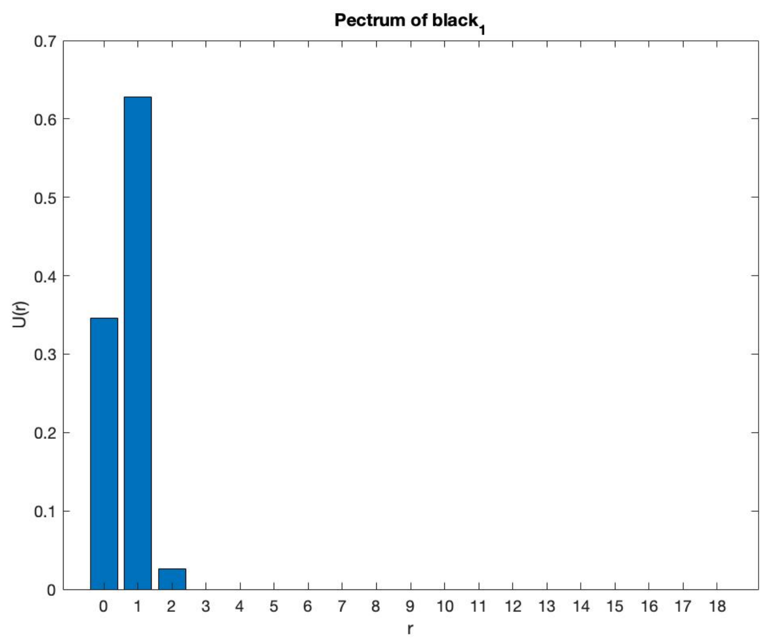

Figure 41: Pectrum of black1.

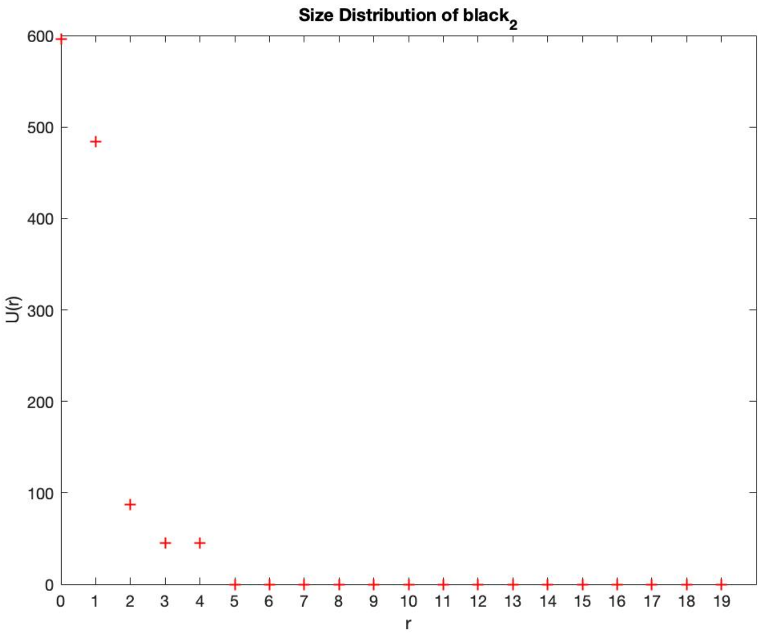

Figure 42: Size distribution of black2.

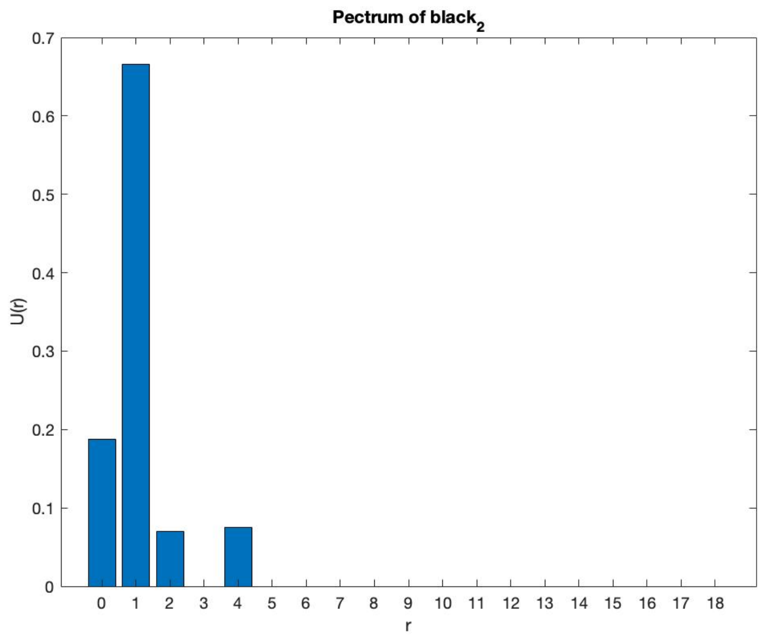

Figure 43: Pectrum of black2.

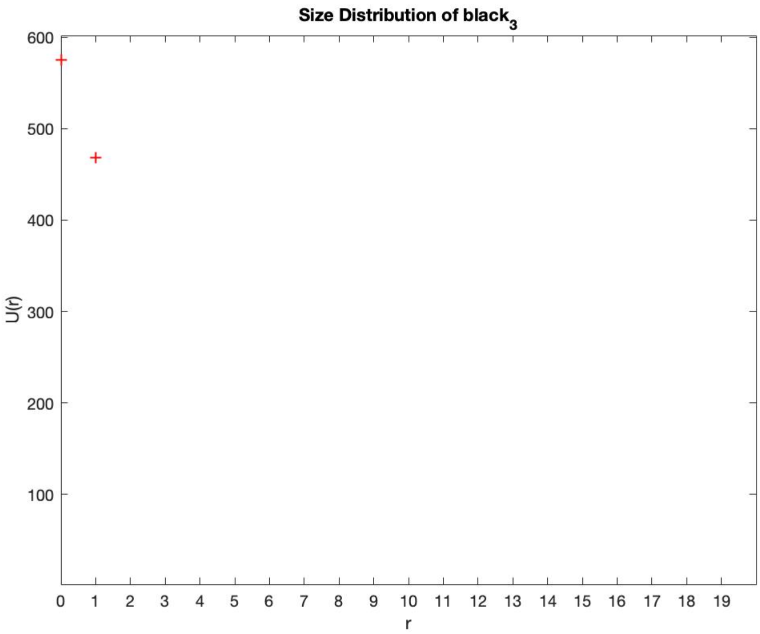

Figure 44: Size distribution of black3.

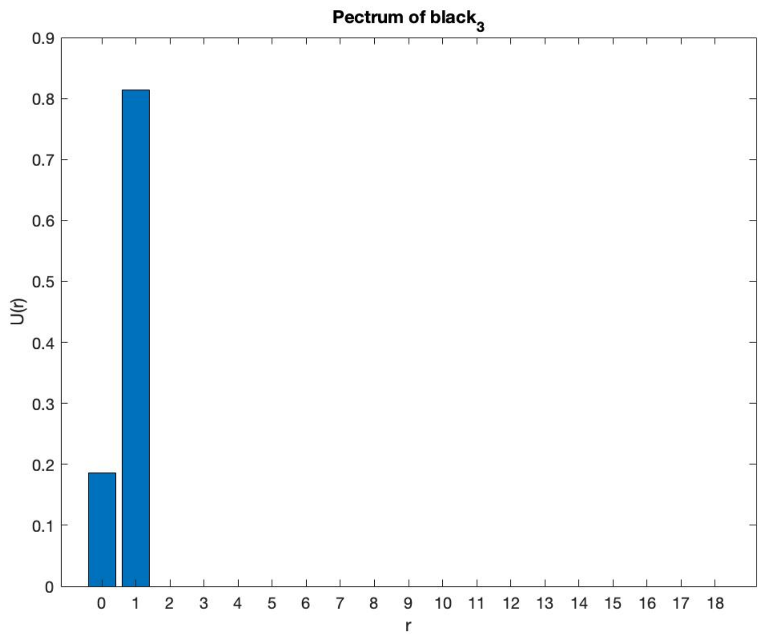

Figure 45: Pectrum of black3.

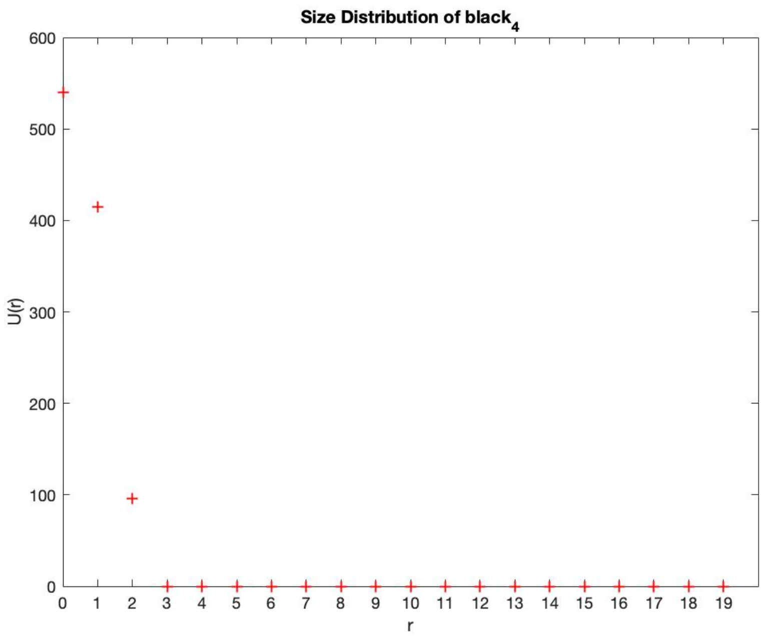

Figure 46: Size distribution of black4.

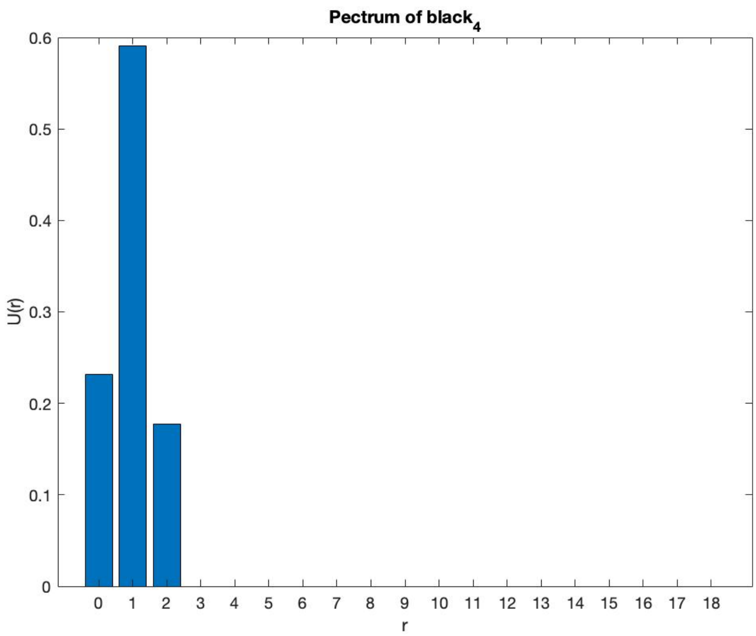

Figure 47: Pectrum of black4.

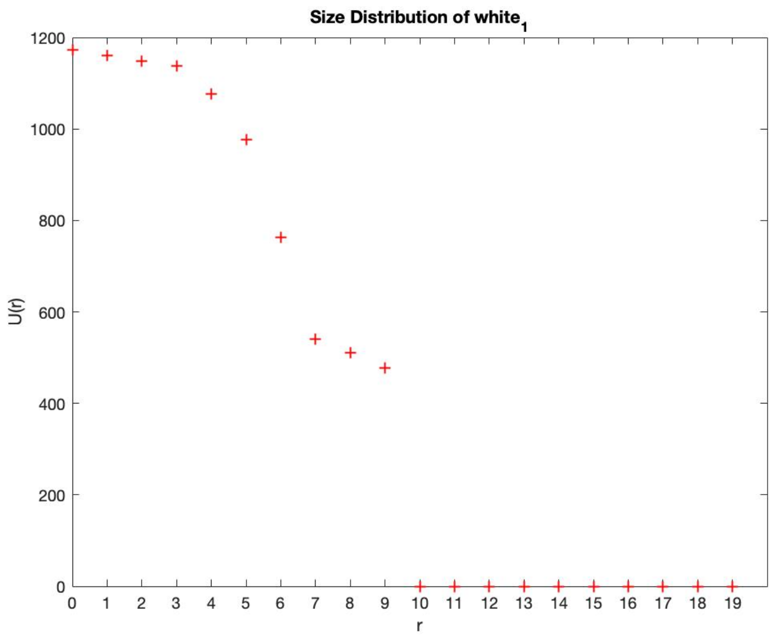

Figure 48: Size distribution of white1.

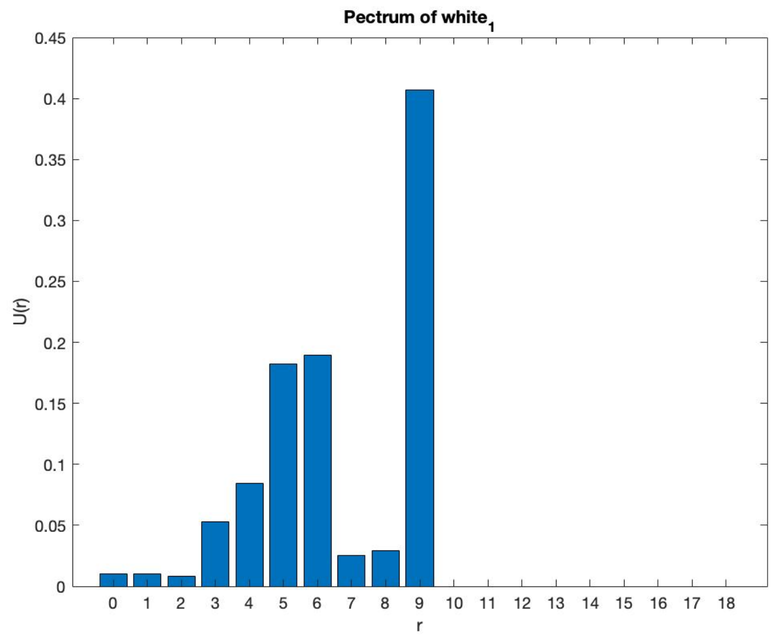

Figure 49: Pectrum of white1.

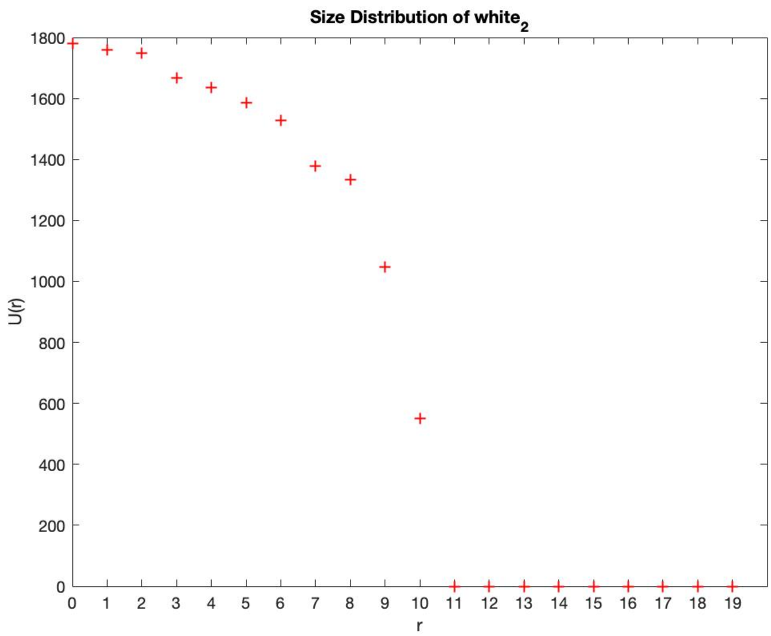

Figure 50: Size distribution of white2.

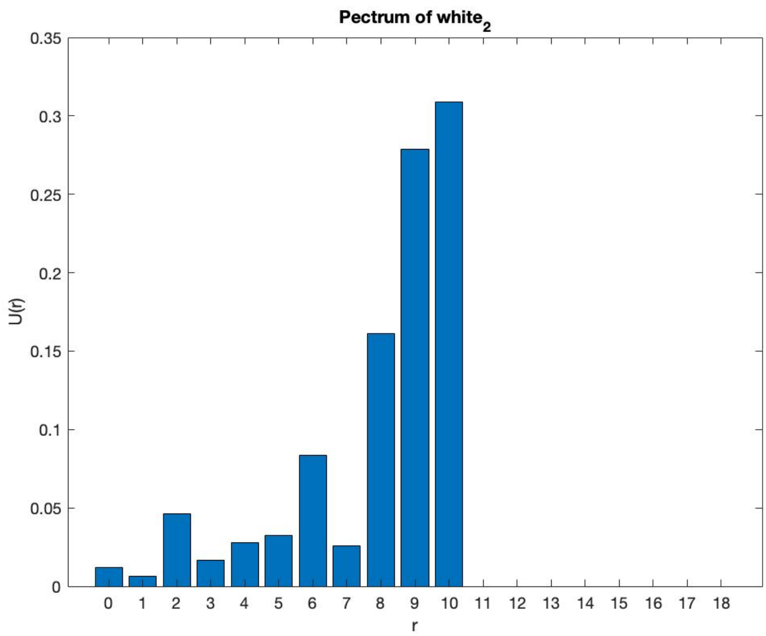

Figure 51: Pectrum of white2.

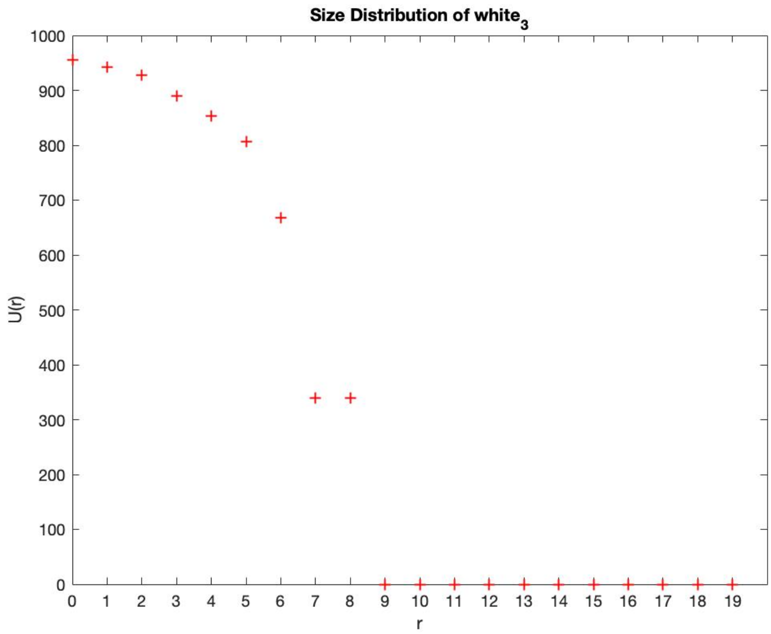

Figure 52: Size distribution of white3.

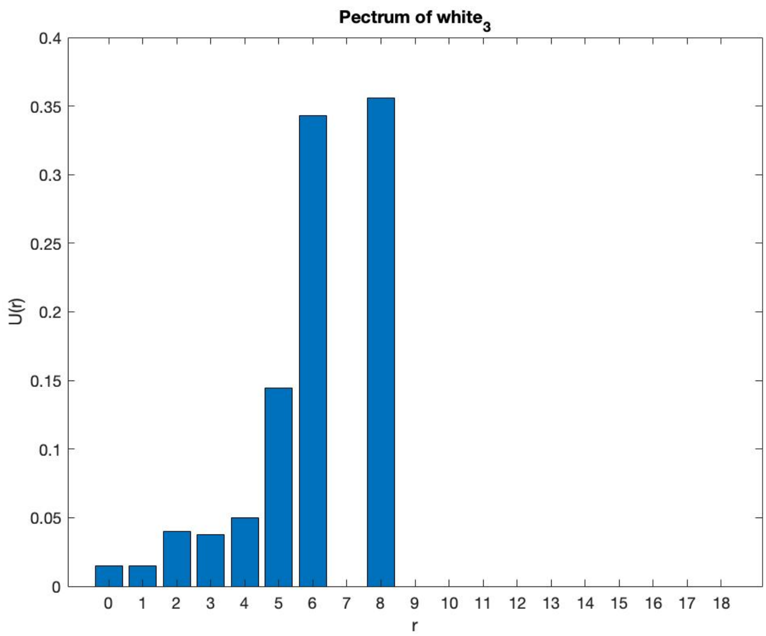

Figure 53: Pectrum of white3.

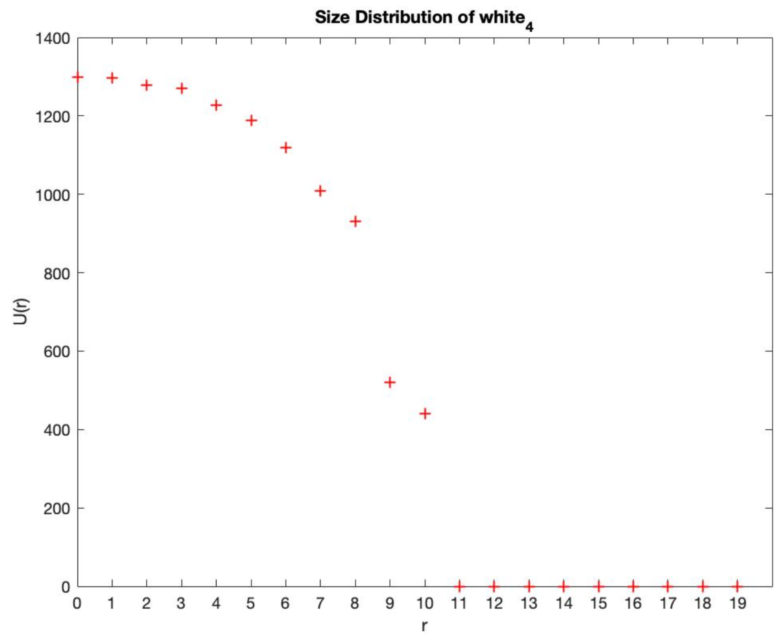

Figure 54: Size distribution of white4.

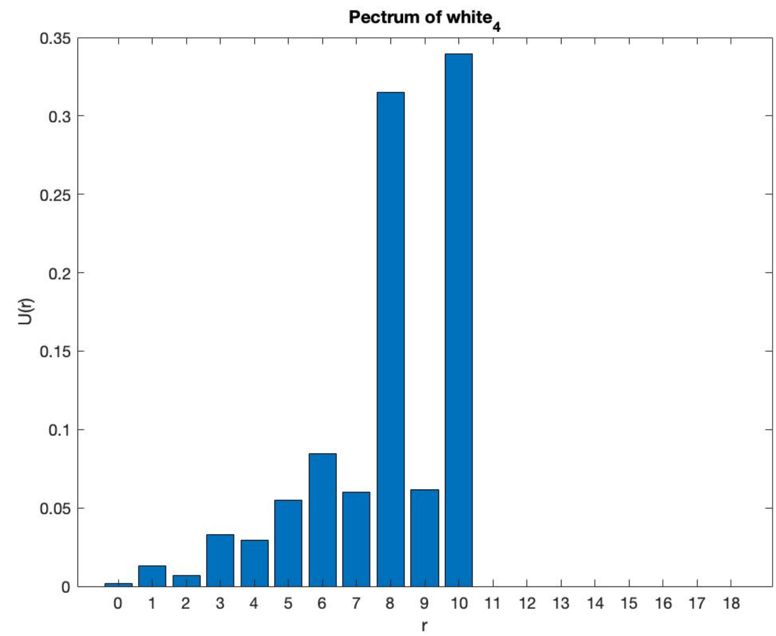

Figure 55: Pectrum of white4.

The same process is applied on ‘shadow3’. Regardless of the location changing, we still use ‘black1’ to ‘black4’ and ‘white1’ to ‘white4’ in ‘shadow1’. The original image is shown in Figure 56, the isolated image is shown in Figure 57. Figure 58 to Figure 73 show their corresponding size distribution and pectrum. The summary of complexity is shown in Table 5.

Image	**Complexity**
Rotated Black1	0.363777
Rotated Black2	0.451152
Rotated Black3	0.229828
Rotated Black4	0.363705
Rotated White1	0.727436
Rotated White2	0.834501
Rotated White3	0.715469
Rotated White4	0.731866

Table 5: Complexity of rotated image.

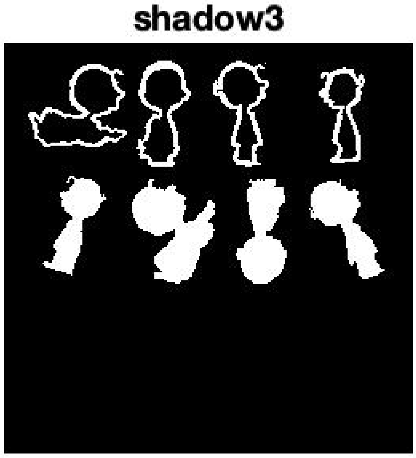

Figure 56: Original image of shadow3.

Then we use function that we defined to calculate the distance between each pairs, and we choose the closed pair and get the result for each objects in ‘shadow1’. The distance d_i among each image are shown in Table C.2.2. As we can see, there is an obvious gap between white and black images, which is excepted. However, White4 cannot match Rotated White4 correctly, and it matches to Rotated White3. This mismatch is probably because White3 and White4 have very similar structures, after rotating, the distances between them might change. All match results are shown in Figure 74 to Figure 81.

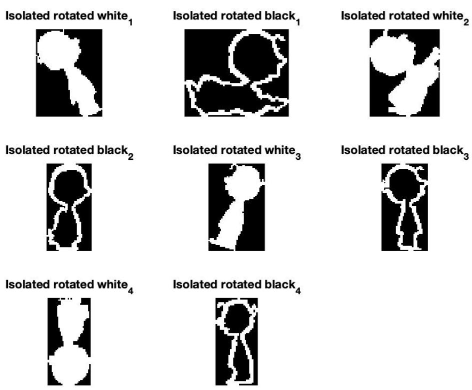

Figure 57: Isolated shadow3.

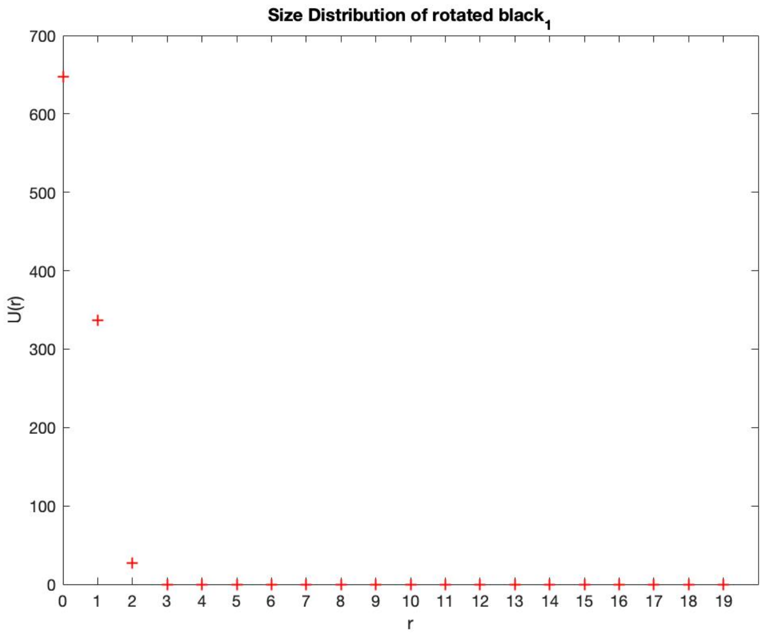

Figure 58: Size distribution of rotated black1.

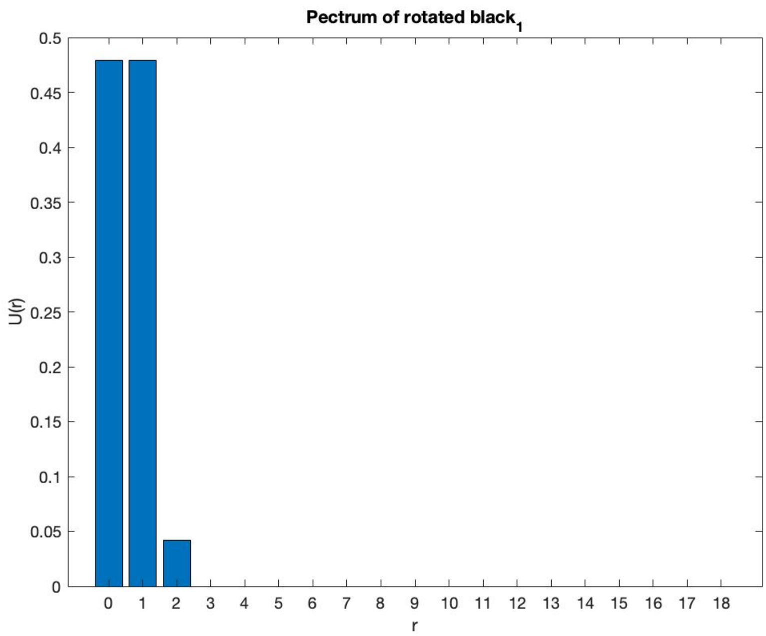

Figure 59: Pectrum of rotated black1.

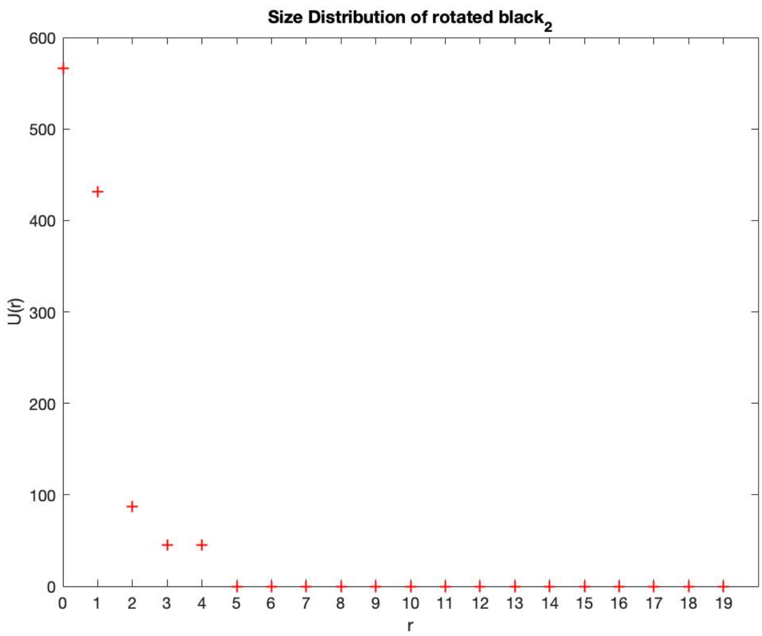

Figure 60: Size distribution of rotated black2.

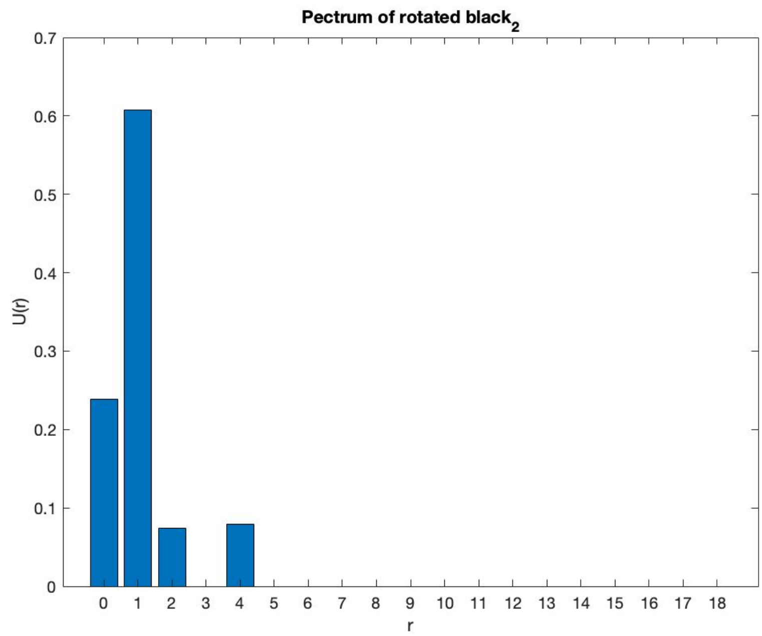

Figure 61: Pectrum of rotated black2.

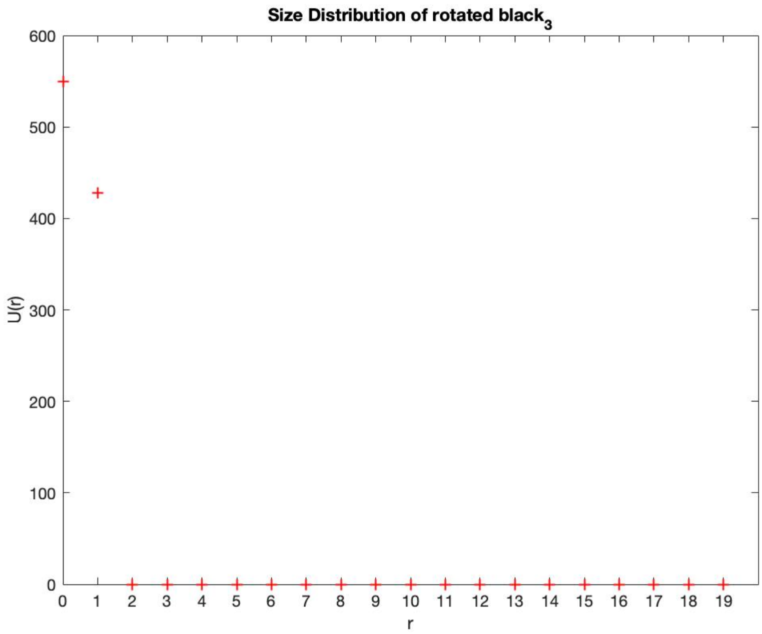

Figure 62: Size distribution of rotated black3.

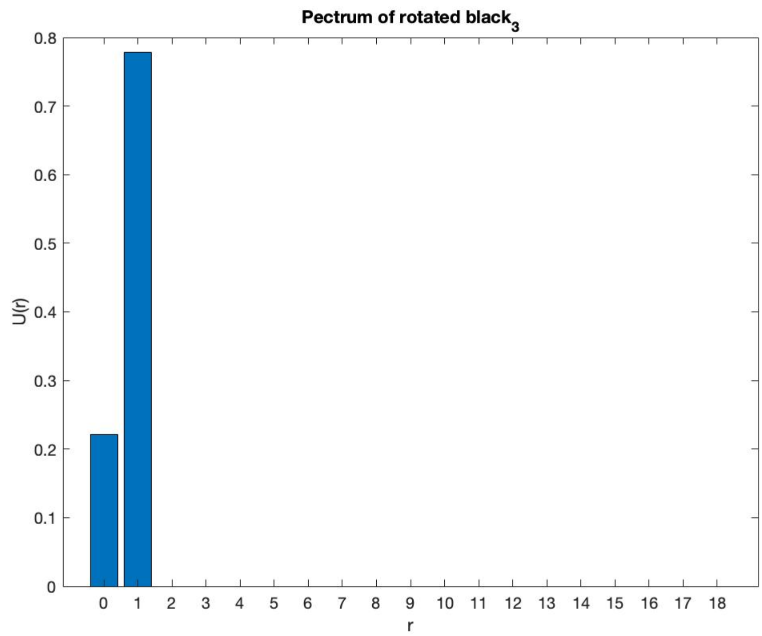

Figure 63: Pectrum of rotated black3.

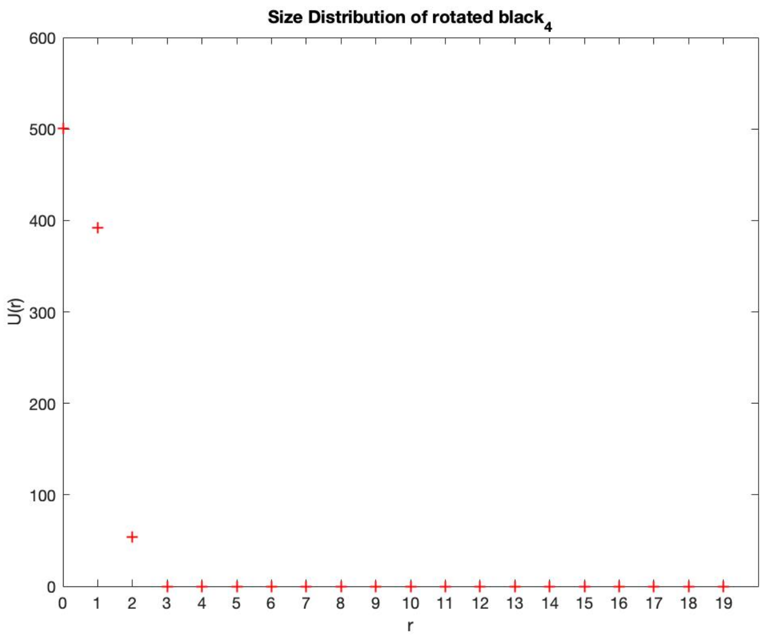

Figure 64: Size distribution of rotated black4.

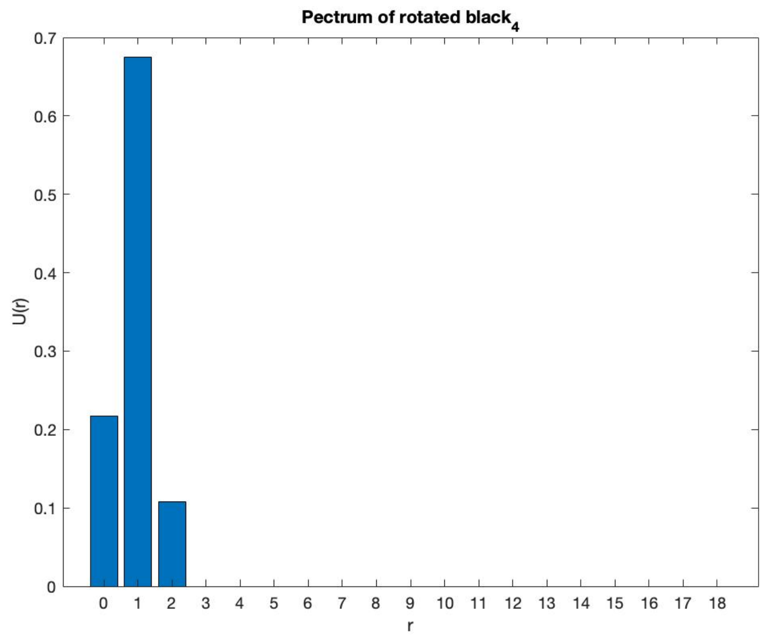

Figure 65: Pectrum of rotated black4.

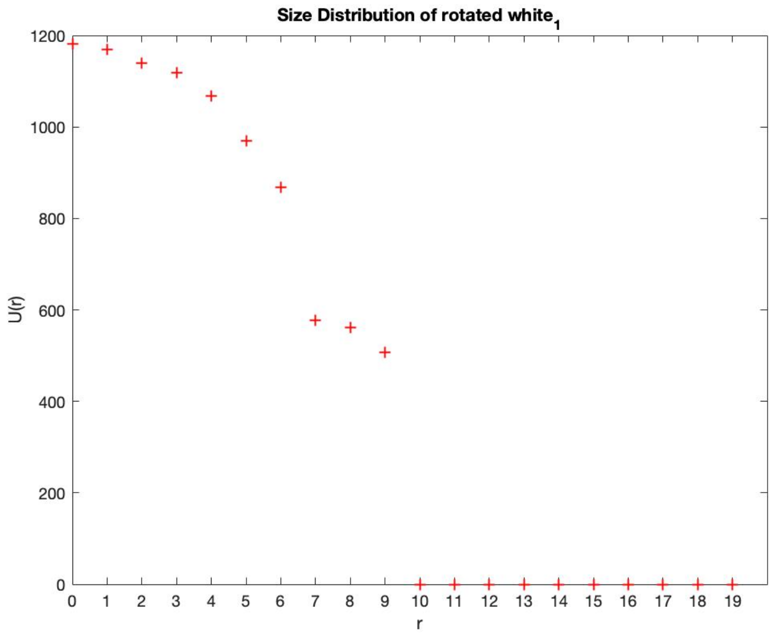

Figure 66: Size distribution of rotated white1.

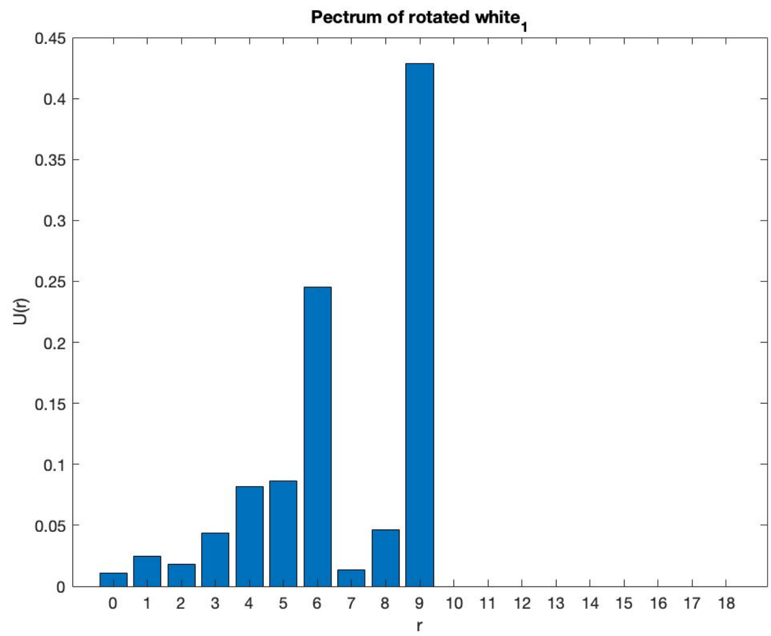

Figure 67: Pectrum of rotated white1.

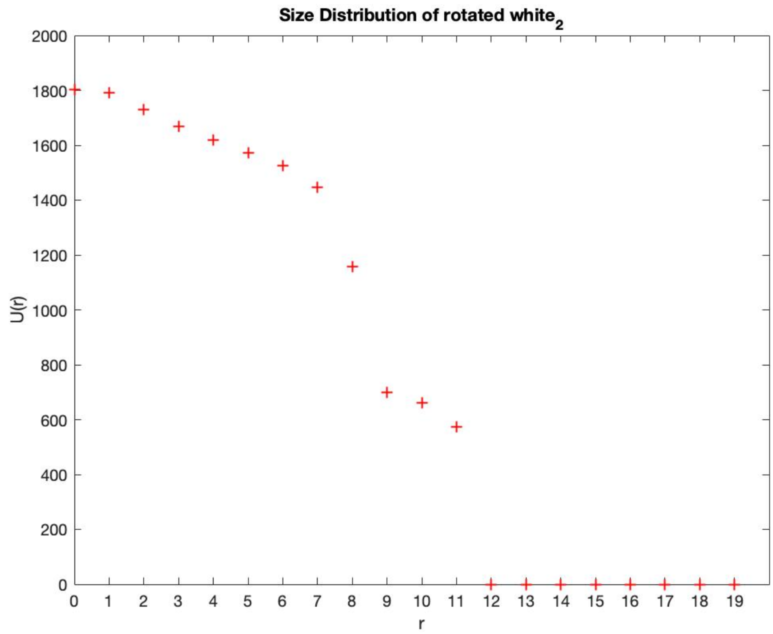

Figure 68: Size distribution of rotated white2.

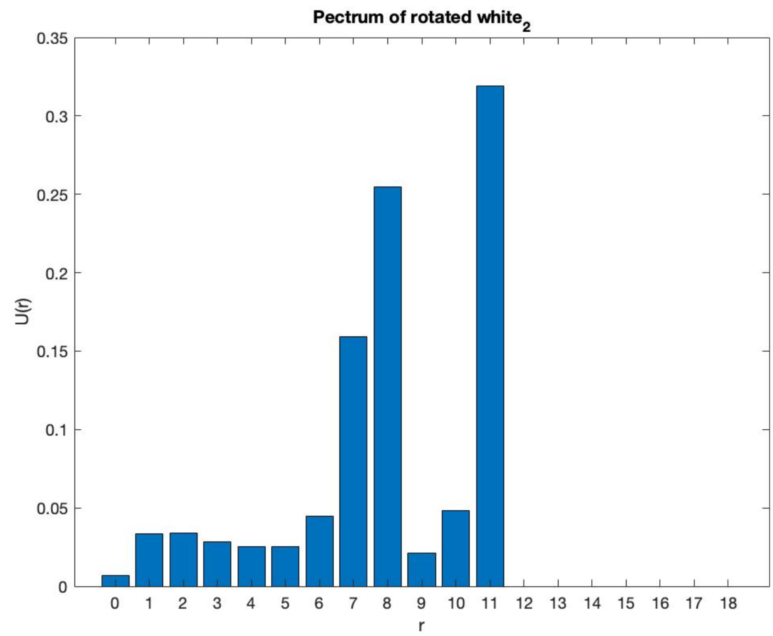

Figure 69: Pectrum of rotated white2.

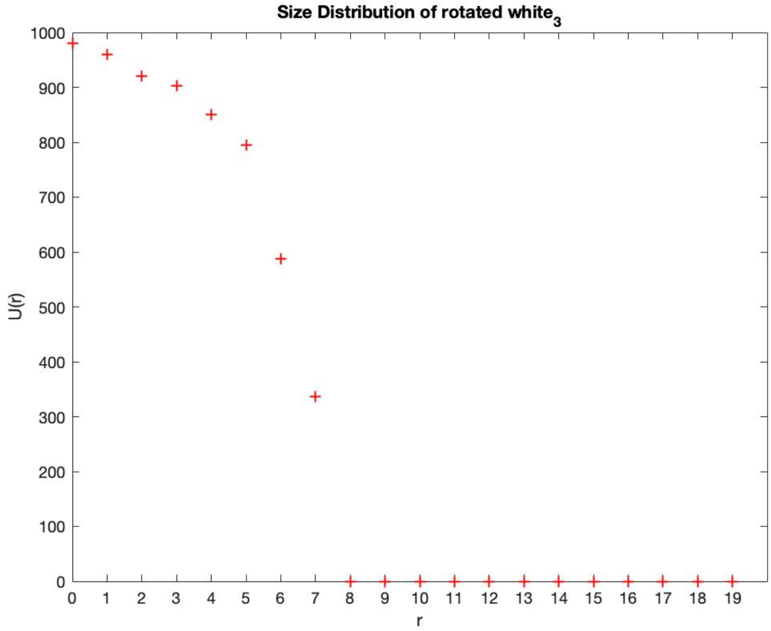

Figure 70: Size distribution of rotated white3.

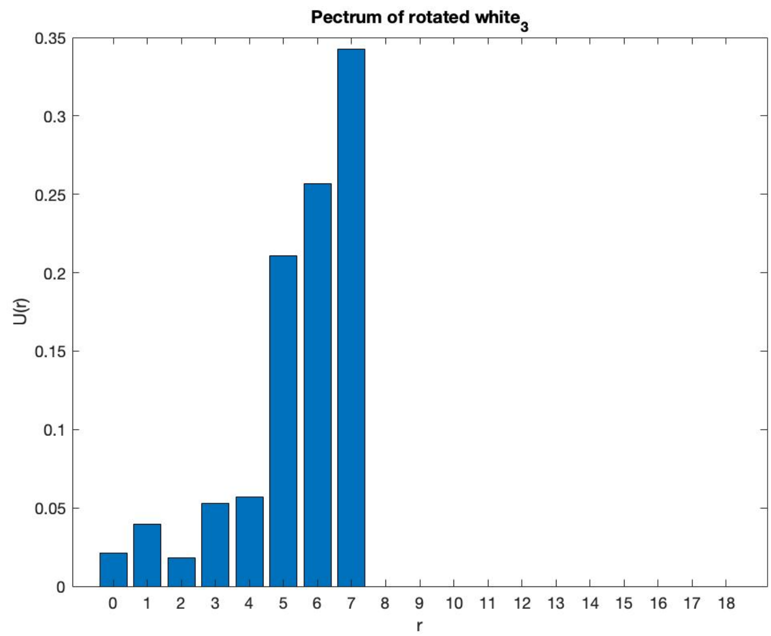

Figure 71: Pectrum of rotated white3.

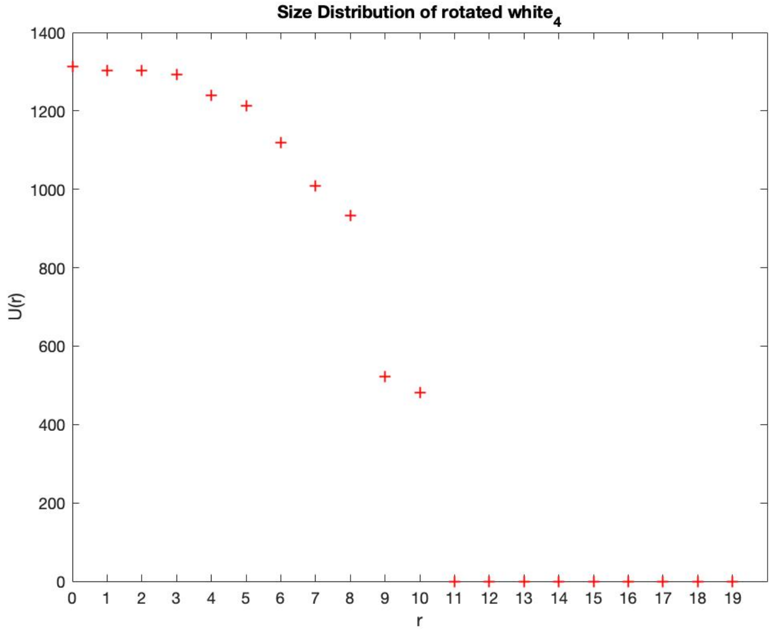

Figure 72: Size distribution of rotated white4.

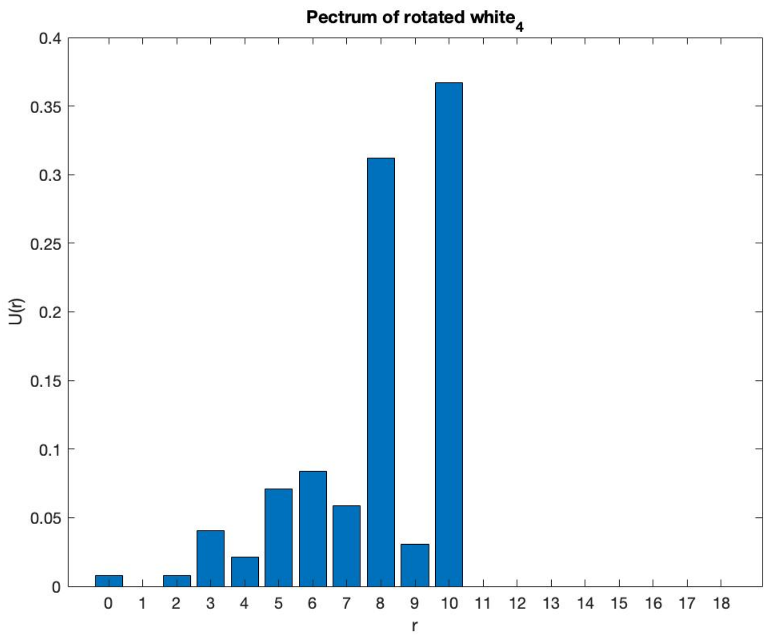

Figure 73: Pectrum of rotated white4.

Distance	Black1	Black2	Black3	Black4	White1	White2	White3	White4
Rotated Black1	**0.198191**	0.278330	0.278788	0.233398	1.205142	1.066148	1.171959	1.125180
Rotated Black2	0.514476	**0.099957**	0.222493	0.107454	1.081853	0.975689	1.065144	1.050349
Rotated Black3	0.624807	0.329840	**0.067690**	0.239527	1.218926	1.092939	1.186645	1.123238
Rotated Black4	0.452858	0.143185	0.315022	**0.138052**	1.174892	1.024015	1.141708	1.094929
Rotated White1	1.248851	1.112890	1.239905	1.213321	**0.091839**	0.416700	0.325898	0.336500
Rotated White2	1.100992	1.004300	1.143477	1.058745	0.236228	**0.206120**	0.447371	0.214874
Rotated White3	1.194359	1.084601	1.233773	1.158829	0.334716	0.327701	0.267043	**0.206799**
Rotated White4	1.137661	1.046626	1.126238	1.102132	0.335138	0.133866	0.389865	**0.043070**

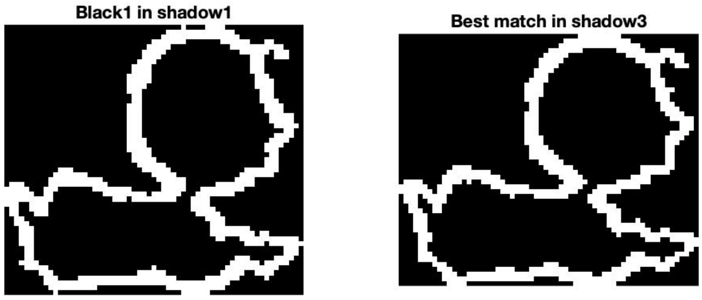

Figure 74: The pair of Black1.

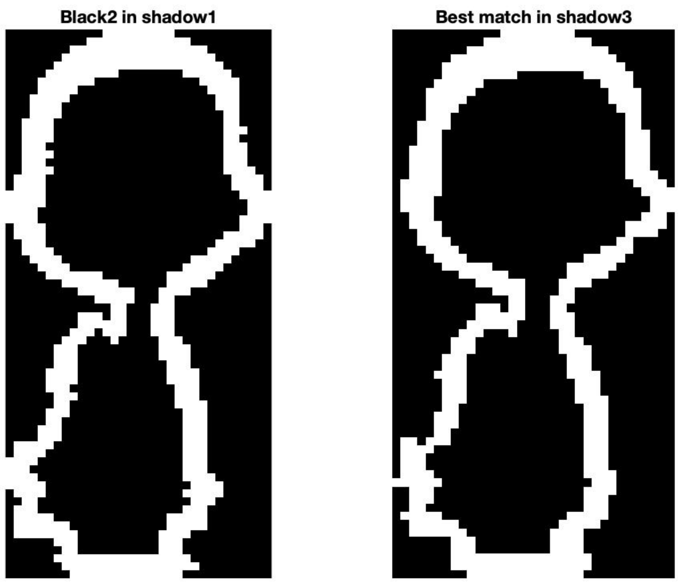

Figure 75: The pair of Black2.

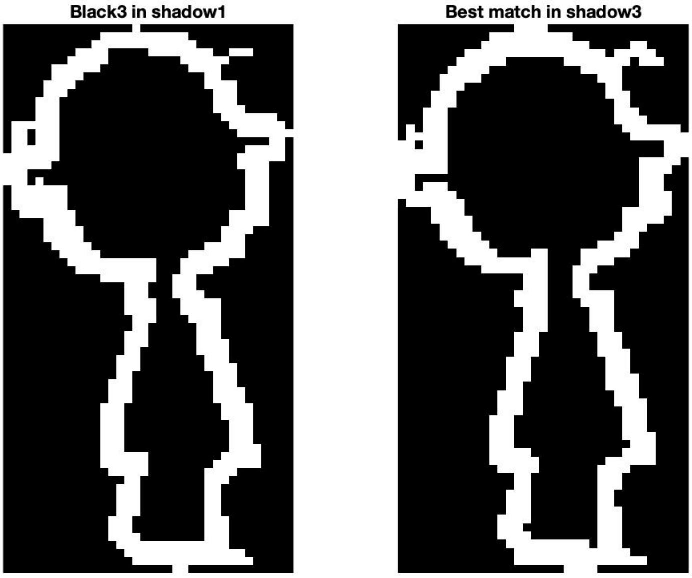

Figure 76: The pair of Black3.

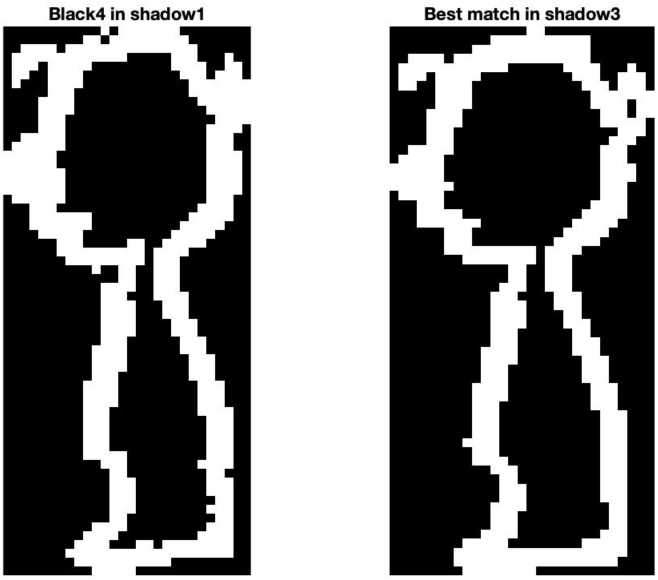

Figure 77: The pair of Black4.

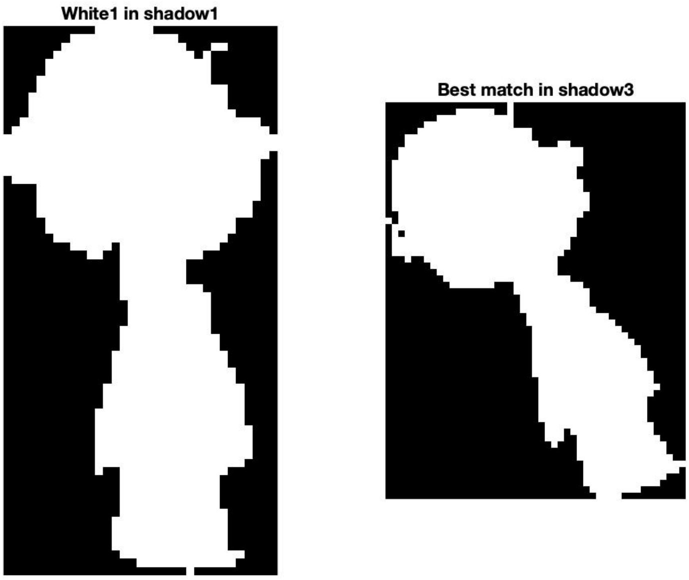

Figure 78: The pair of White1.

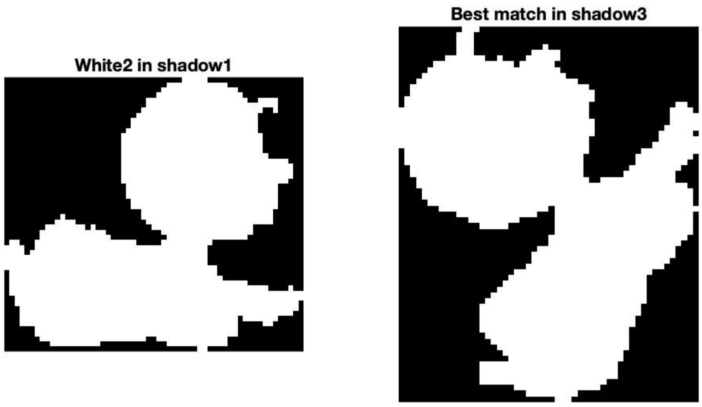

Figure 79: The pair of White2.

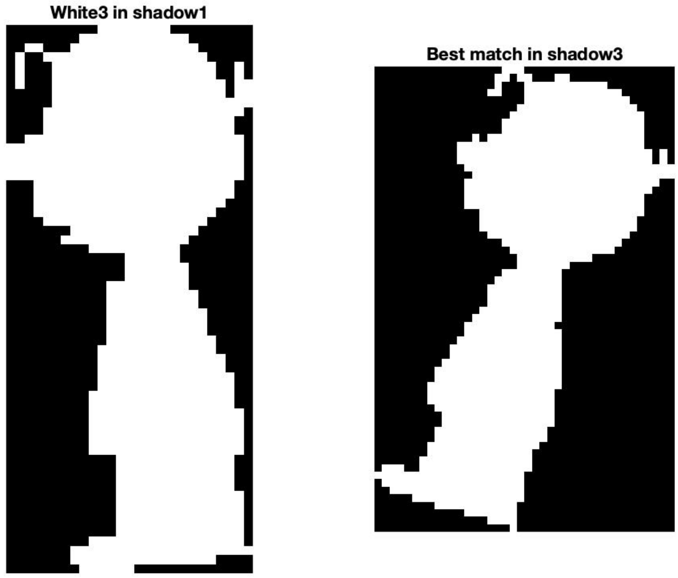

Figure 80: The pair of White3.

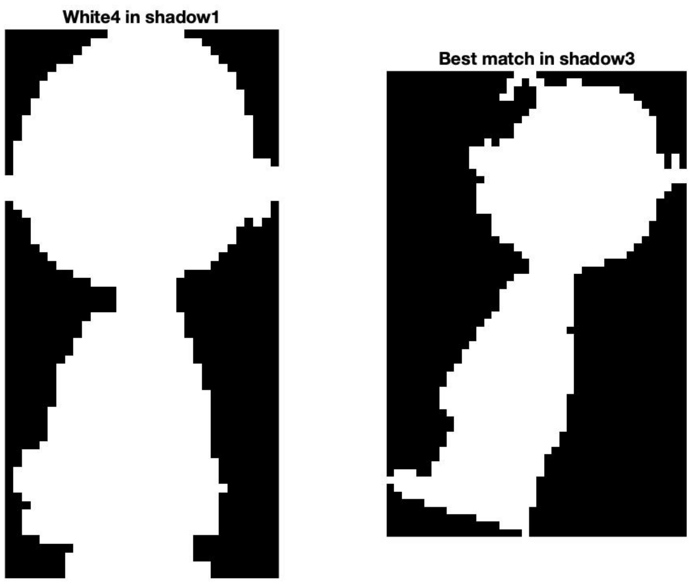

Figure 81: The pair of White4.

D Conclusions

In this project, we think that we applied what we have learned in the class from theoretical operations to the actual implementation. The first part performs edge detection by using a homotopic skeleton technique that uses hit-or-miss during iterations to extract edges and does a set subtraction. We successfully extract the skeleton by using the eight predefined filters. Besides skeleton extraction, the second part tends to make a comprehensive shape analysis for the objects in the images. We perform the shape analysis strictly following the rules in the lecture notes. Last but not the least, we quantitatively match the objects to the proper objects. Considering the difficulty of this project, we spent more time than we expected but the results are satisfying.