Santa Clara University

School of Engineering

Digital Image Processing I: Project Part 3

ELEN 640
December 2016

Summited by
Victor Figueroa

Course Instructor
Sally Wood
Professor of Electrical Engineering

Table of Contents

1. Best Methods Used to Identify the Coins

1.1. Image Set #1

The images presented in the set #1 are grayscale images with only coins of different sizes and different types of contrast and illuminations as shown in Figure 1.1

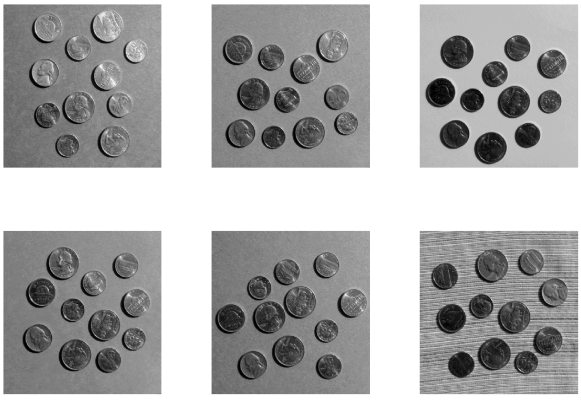


Figure 1.1. Images of Set #1

1.1.1. Best Method for Set #1

For Image set #1, the best method found to find the coins in the image is using **Hough Transform on the edges of the gray scaled image,** because there is no distractors elements in the image we can easily perform a hough transform after applying an appropriated edge filter to get the centers and the radii of the coins.The process is illustrated in the block diagram in Figure 1.2.

We picked this method because we do not have distraction elements and only circular elements

a. Block Diagram

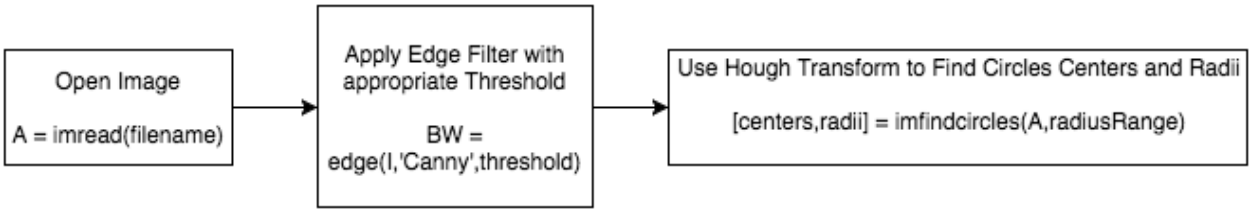


Figure 1.2. Hough Transform Process for Image Set #1

.

b. Selected Images Result

For image 563 and Image 579 the following table summarize the result using the procedure of Table 1.1

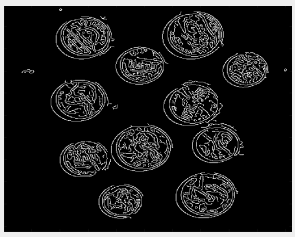
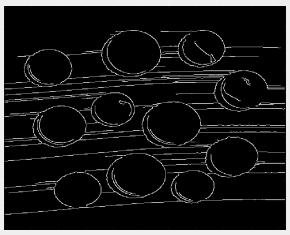
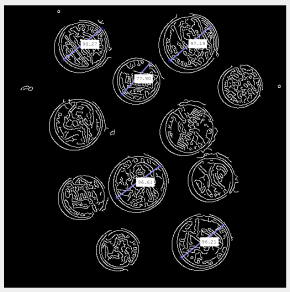
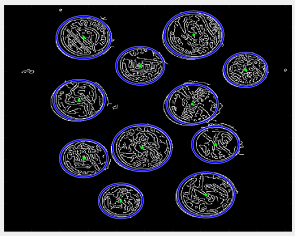
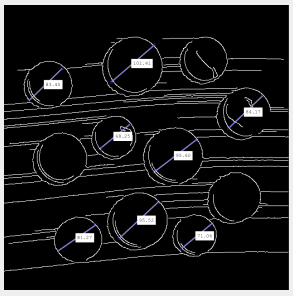
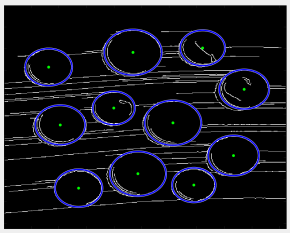
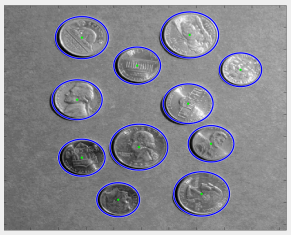
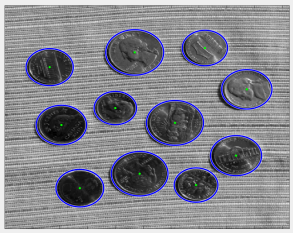
Original Image	Edges of the Image	Original Image	Edges of the Image
			
Selecting Range for Radii	**Results**	**Selecting Range for Radii**	**Results**
			
Results on Original Image	**Centers and Radii**	**Results on Original Image**	**Centers and Radii**
	242.3283 137.9206 42.3188 207.9151 443.4820 38.7314 429.4811 147.3092 38.3067 337.4628 67.9250 52.3604 142.1844 347.5893 42.0210 245.5203 323.6171 52.2628 376.1223 316.0352 41.3362 359.0073 430.2063 50.4519 133.6434 215.5848 45.2718 142.2399 74.1031 47.9311 334.6664 224.6516 45.1028		199.5493 235.9498 38.0968 360.6709 98.3689 40.6183 243.3372 386.4502 50.9251 81.4611 142.1834 42.4046 344.7499 412.4534 39.2835 102.5180 274.8503 45.2166 234.4009 108.4148 52.0775 306.8506 269.9180 51.3027 436.0053 193.0489 45.1368 416.6783 344.9854 45.9066 135.7279 419.5910 42.9115

Table 1.1 Result for selected images on set #1

1.2. Image Set #2

The images presented in the set #2 are the color version of the images presented in image set #1, as shown in Figure 1.2. We can see now the different colors of the coins and the color of the backgrounds allowing us to perform a better segmentation of the image based on the color information for some of the images.

Figure 1.2. Images of Set #2

1.2.1. Best Method for Set #2

Image Set #2 are the colored images of set #1 therefore we can use the same method as in set #1 to find the coins centers and radii but first we need to obtain the grayscale version of the images.

1.3. Image Set #3

The images presented in the set #3 are color images with coins of different sizes and non circular distractions objects with different shapes, colors of contrast and illuminations as shown in Figure 1.3.

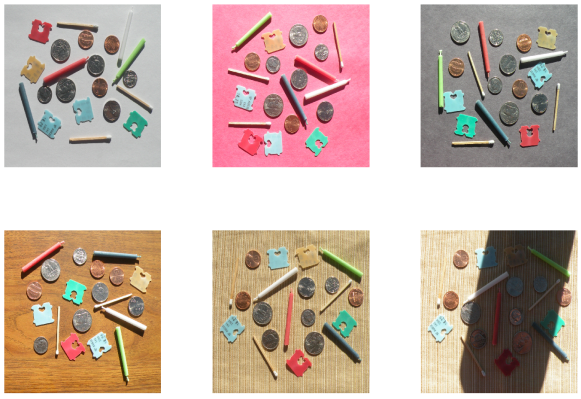


Figure 1.3. Images of Set #3

1.3.1. Best Method for Set #3

For Image set #3, the best method found to find the coins in the image is using **Hough Transform on a BW image** we were able to separate almost all of the images using a combination of RGB or HSV color segmentation and thresholding, combined with morphological operation in order to eliminate the non circular distracting element and then performing a hough transform to find the centers and the radii of the coins.The process is illustrated in the block diagram in Figure 1.4.

We picked this method because we were able to separate the foreground and the background and eliminate the distraction element based in the shaped of the distractor objets.

c. Block Diagram

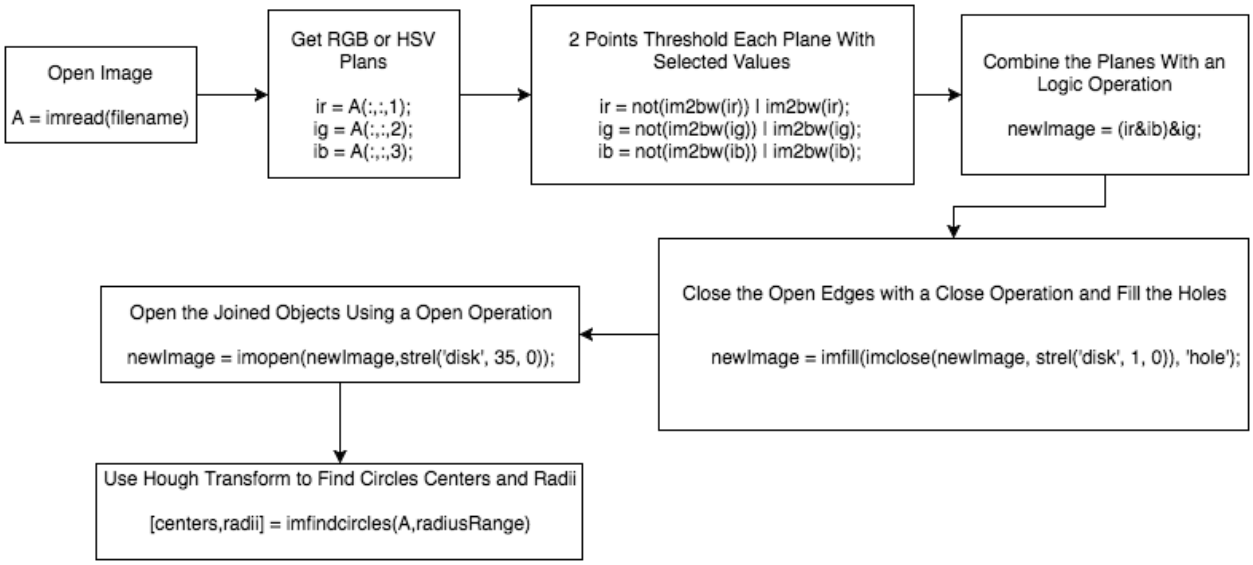


Figure 1.4. BW Hough Transform Process for Image Set #3

d. Selected Images Result

For image 616 and 619 the following table summarize the result using the procedure of Table 1.2

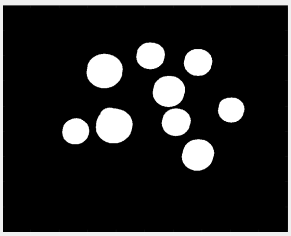
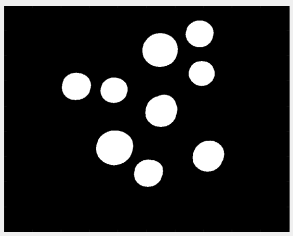
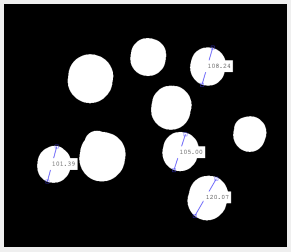
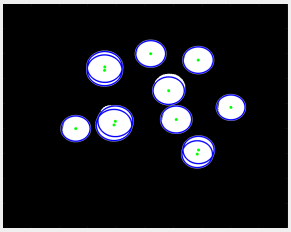
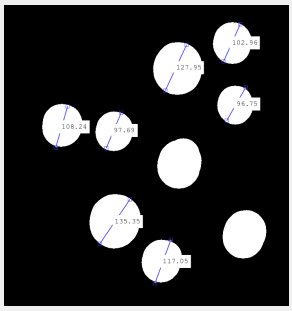
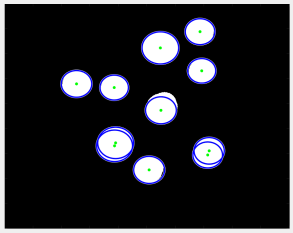
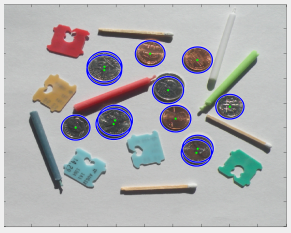
Original Image	Segmented Image	Original Image	Segmented Image
			
Selecting Range for Radii	**Results**	**Selecting Range for Radii**	**Results**
			
Results on Original Image	**Centers and Radii**	**Results on Original Image**	**Centers and Radii**
	519.7953 200.4177 52.3256 683.1419 601.5355 53.5093 582.9936 348.0584 54.8359 686.6015 225.9459 52.8018 687.7254 584.9032 54.5342 609.5259 463.1647 53.8162 391.0323 485.0418 62.2220 358.0248 266.6106 61.4776 358.6938 253.2327 61.5957 256.9178 499.4723 50.4310 801.2699 414.7685 49.7150 395.9131 470.3646 61.1877		547.4355 177.6980 64.3624 252.9359 321.8691 53.4492 686.8374 112.6636 51.8688 718.8634 590.0446 53.1336 385.3520 336.6177 49.8606 549.4195 427.6463 54.0613 713.7194 606.5444 52.0350 390.2278 558.1761 62.9726 386.4999 569.9432 63.1255 507.9907 666.3561 54.2486 692.7375 269.5300 49.1694

Table 1.2 Result for selected images on set #3

1.4. Image Set #4

The images presented in the set #4 are color images with coins of different sizes and circular distractions objects with different colors and sizes, contrast and illumination as we can see in Figure 1.5.

Figure 1.5. Images of Set #4

1.4.1. Best Method for Set #4

For Image set #4, the best method found to find the coins in the image is using **Matching Filters** performing a normalized cross correlation with a different templates for each of the coins and finding the respective peaks. The process is illustrated in the block diagram of Figure 1.6.

We picked this method because if we try to separate the foreground and the background there is no way to differentiate whats was a coin and was is not a coin only based in the shaped of the object as the distractor elements are also circular.

a. Block Diagram

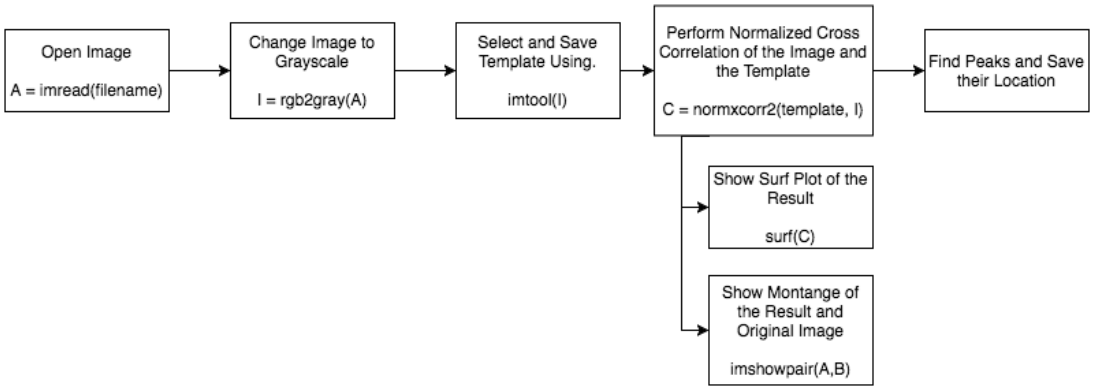


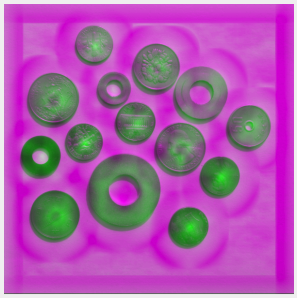
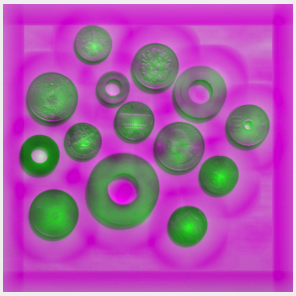
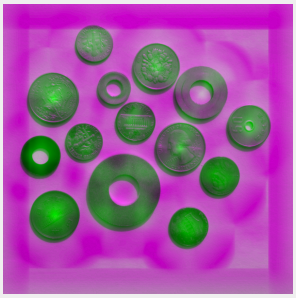
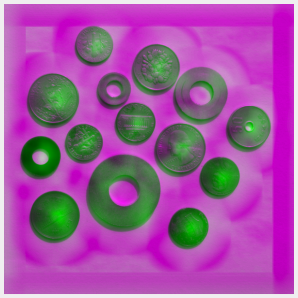
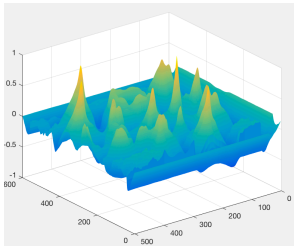
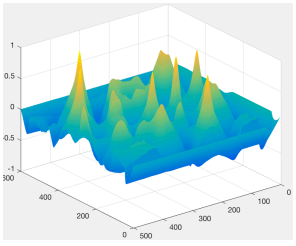
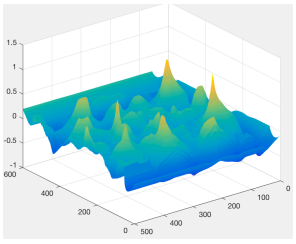
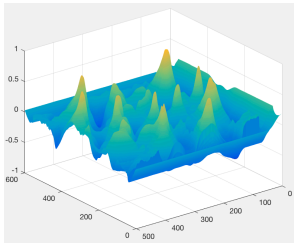
Figure 1.6. Match Filter process for Image Set #4

b. Selected Images Result

For image 582 using the following table summarize the result using the procedure of Table 1.3.

Original Image	Gray Scaled Image
	

Templates			
			

Results			
			
			

Peaks >= 0.7			
171.0000 290.0000 1.0000 388.0000 463.0000 0.8808 453.0000 359.0000 0.8762 192.0000 88.0000 0.8386	386.0000 466.0000 1.0000 450.0000 362.0000 0.9616 190.0000 91.0000 0.8989 169.0000 292.0000 0.8578 276.0000 250.0000 0.8343 81.0000 319.0000 0.8005	105.0000 198.0000 1.0000 108.0000 439.0000 0.8250 367.0000 300.0000 0.7170	512.0000 258.0000 1.0000 117.0000 436.0000 0.7423

Table 1.3 Result for selected images on set #4

For image 582 using the following table summarize the result using the procedure of Table 1.4

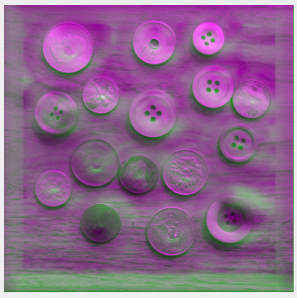
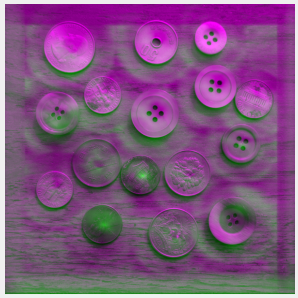
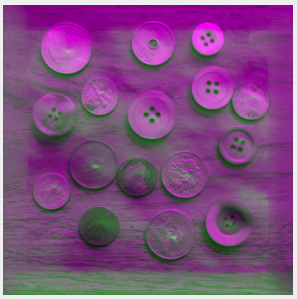
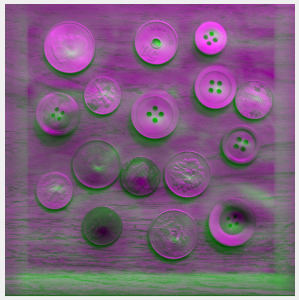
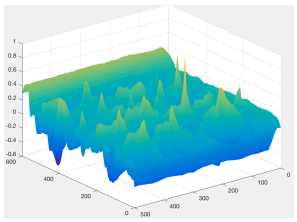
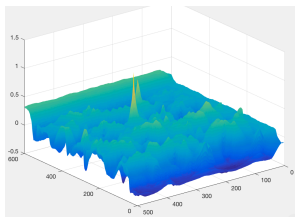
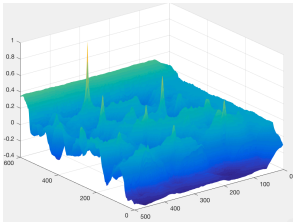
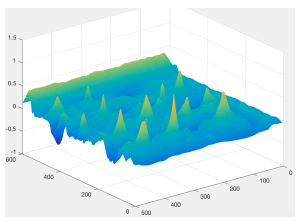
Original Image	Gray Scaled Image
	
Templates	
   	
Results	
   	
   	
Peaks >= 0.7	
205.0000 193.0000 1.0000	285.0000 356.0000 1.0000
352.0000 479.0000 1.0000	317.0000 83.0000 1.0000 439.0000 174.0000 0.7012

Table 1.4 Result for selected images on set #4

2. The Ideal Image (Most Robust)

The Image 569 shown in Figure 2.1 was identify as the most robust image across all the methods. The Image is the image number 3 from the image set #1, and therefore is also image number 3 from the set #2

Figure 2.1. Most Robust Image

2.1. Characteristic

The image defined as most robust presented a high contrast between the background and foreground, as shown in the histogram in Figure 2.2 and the background is a solid light gray color, the coins have enough space in between and there is not overlapping even in the minimum shadow it present.

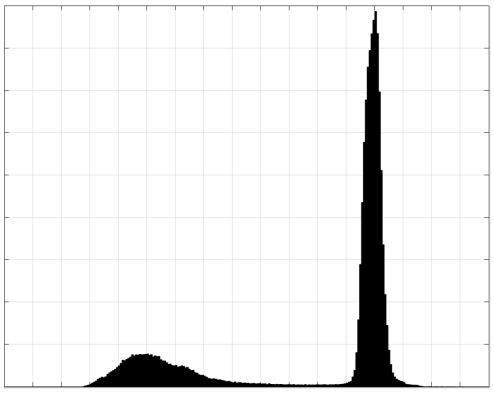


Figure 2.2. Histogram of most robust Image

Most of the features of the inside the coins are not underexpose, but for this case we only care about the size thus these features are not needed.

2.2. Processing Steps

In order to find the circles on this image we didn't require a lot of prepossessing as most of the methods worked very well as we see below.

2.2.1. Segmentation

Using simple thresholding, because the high contrast of the background and foreground we found a range of value that worked well from 0.5 to 0.73.

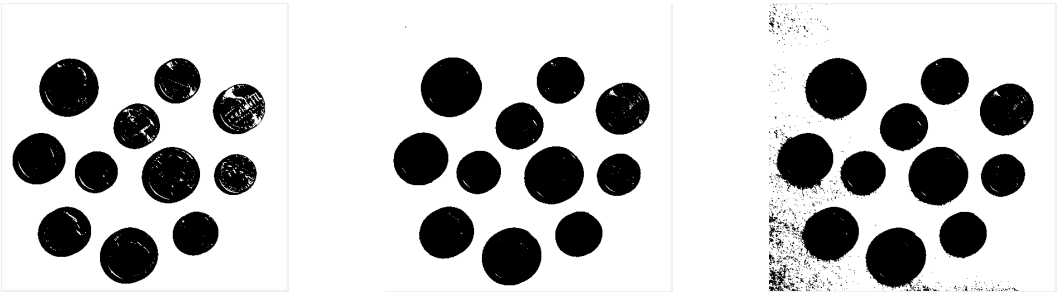


Figure 2.3 Range of values for Thresholding

We also start noticing some detail of the background start to appear and is at this point the process would not work anymore.

Also because of this high contras we can easily detect the edges using the different method, Sobel, Prewitt, Roberts, LoG and Canny, with the automatic thresholding, being Canny with manual thresholding the most reliable one

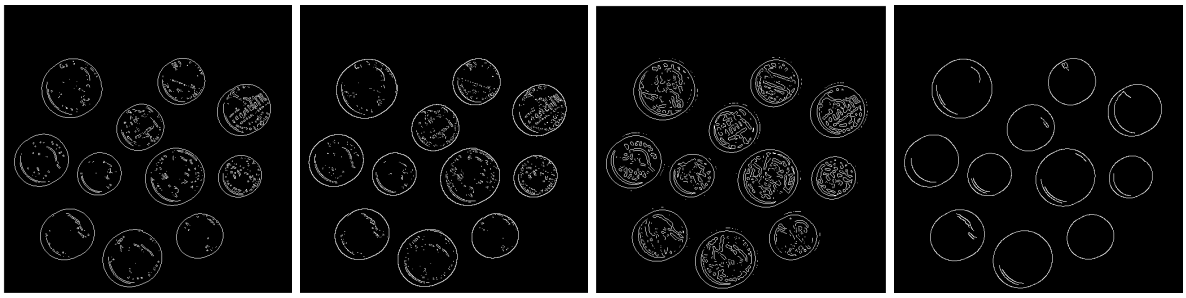


Figure 2.4 Range of values for Thresholding

2.3. System Design

2.3.1. Image Resolution

According to the theory we need at least 2 pixel in the smallest feature in this case a smaller image should work as well, as see in Figure 2.5 with a lower resolution image.

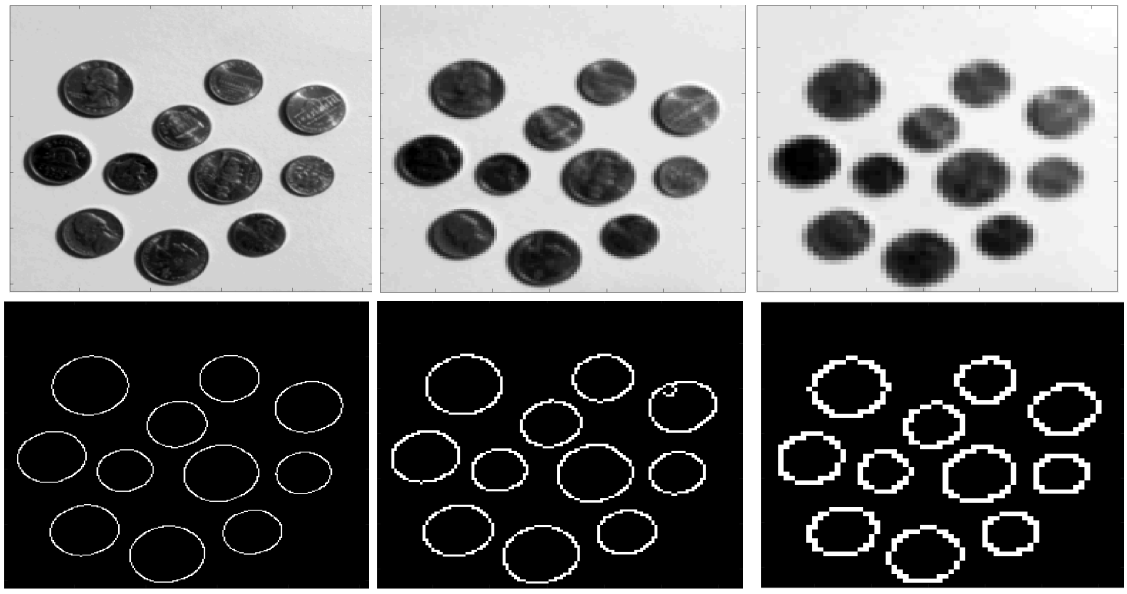


Figure 2.5. Lower resolution Images and their detected edges

As we can see using lower resolution image we can still detect the edges of the images and therefore we would be able to find the center and the radii of the images as seen in Figure 2.6. and the computational complexity decrease.

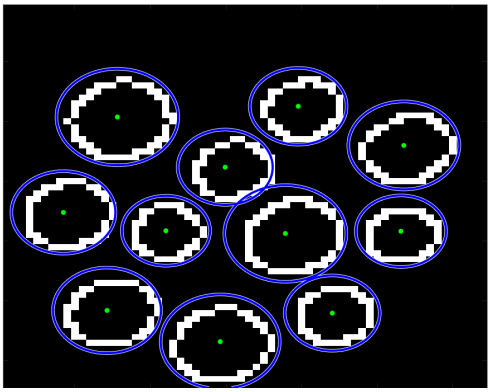


Figure 2.6 Centers and radii of low resolution Image

2.3.2. Illumination Source

I would recommend a system a diffuse illumination right above the coins to reduce the shadows and detect the radii with more accuracy.

2.3.3. Background

I would use a solid and uniform background with such as it would produce a high contrast between the foreground and the backgroun.

Because the algorithm we are using depend on the edges of the image, a patterned background would create edges other than the coins, making it harder to detect the circles.

3. The Image 632

The Image 632 shown in Figure 3.1 was identify as the most complex image across all the methods. The Image is the image number 6 from the image set #3.

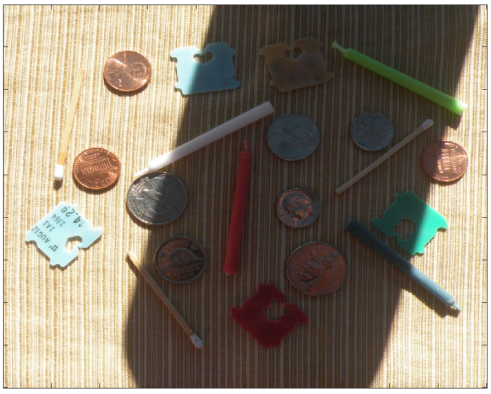


Figure 3.1. Image 632

3.1. Characteristics

The image defined as most complex presented a noticeably bad illumination due to the big shadow in the middle of the image, also with a patterned background and with non circular distraction element.

We can see that the simpler processing method of the most robust image will fail here, as we don't have a very well define high contrast between the background and the foreground objects, so the simple two point thresholding will fail, as well as the edge detection.

3.2. Processing Steps

First we used some smoothing filter in all three planes, hopping to make the background patter more uniform, as well as the different colors. We apply a median filter size 15x15 and then we apply an averaging filter 5x5, the result are shown in Figure 3.2

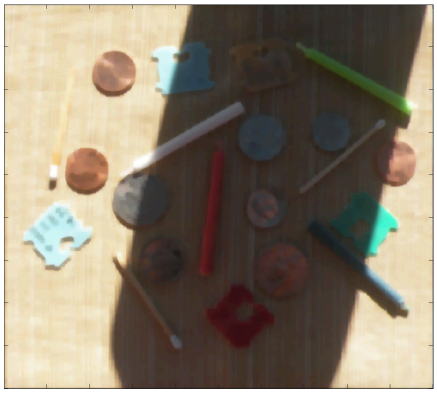


Figure 3.2 Smoothed Image

Now we proceed to use different color segmentation in order to remove the shadow, and we obtained after several trial is show in Figure 3.3

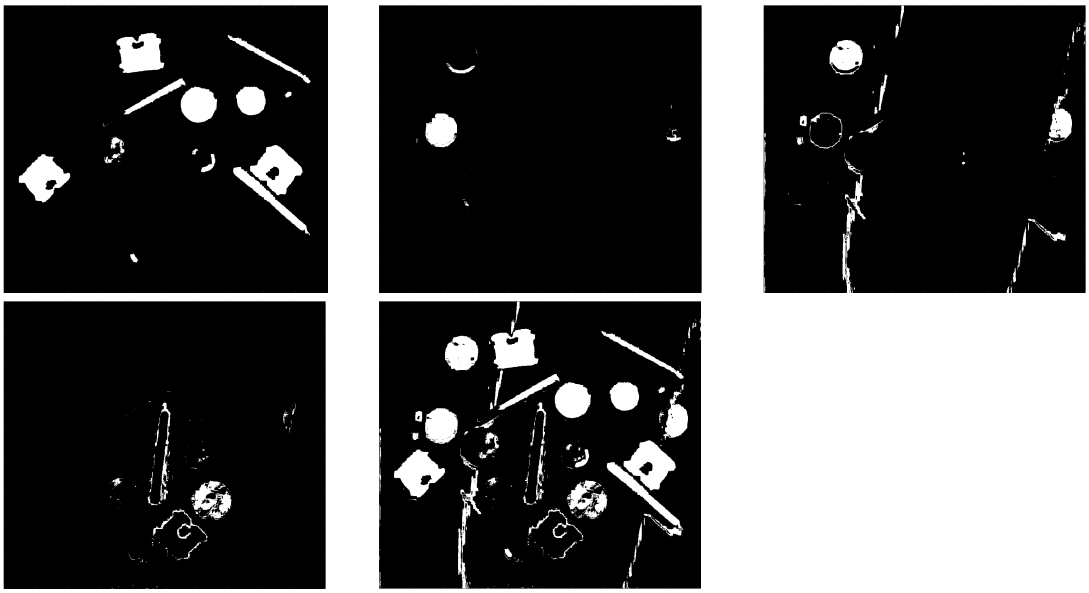


Figure 3.3 Several color segmentation and final result thresholding

As we can see, we were successful in separate some of the objects inside the shadow as well the ones in the well illuminate part.

3.3. Postprocessing

Now we could try to use different types of morphological operation in order to have better circular shapes and then use hough transform to find the centers and radii of the circles. The result of the morphological operation is show in Figure 3.4

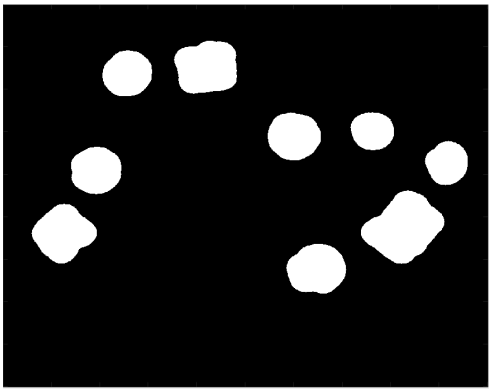


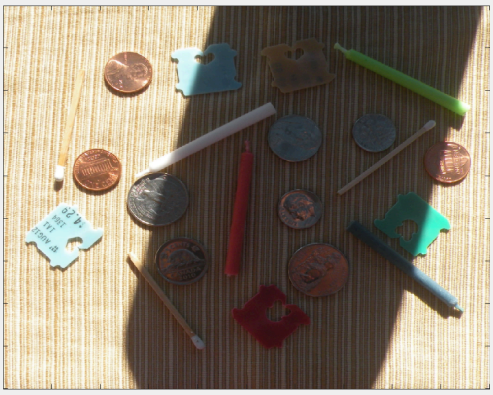
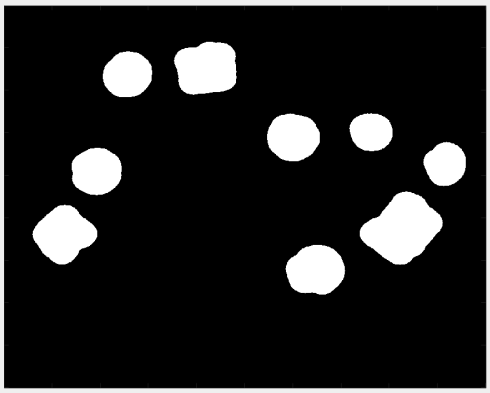
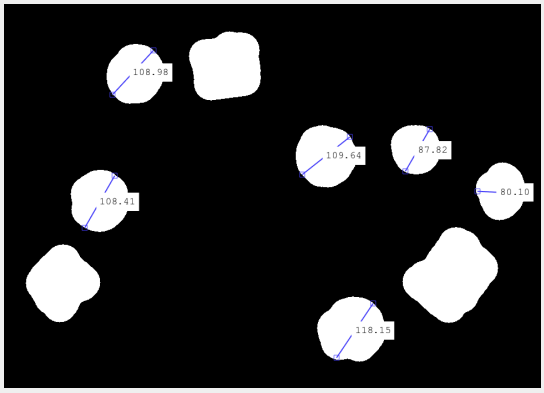
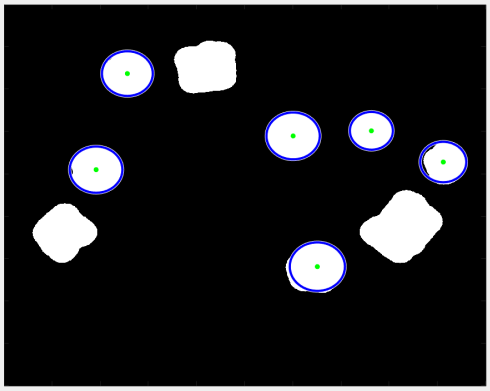
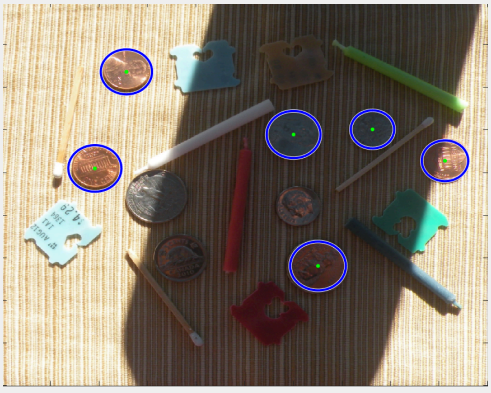
Figure 3.4 Image after Morphological Operation

As we can see of the original 9 coins in the image we can distinguish 6 circular objets, most likely the radii will not be accurate but he centers may be.

We can see that in order of getting some result the simpler step for the ideal image had to be modified starting with the used of smoothing and medial filters and performing several segmentation base on the color of the image, increasing the complexity and the time use in order to get some result.

3.4. Result

Using the Hough Transform in this new Image we get:

Original Image	Segmented Image
	
Selecting Range for Radii	**Results**
	
Results on Original Image	**Centers and Radii**
	763.0635 298.5971 44.6218 600.5865 310.6601 55.3340 256.5649 163.4903 52.7537 191.6701 390.2793 54.6125 650.8919 619.2525 57.2542 912.2500 372.2751 47.7773

At the end with our methods we manage to find 6 out of the 9 coins in the image :)

4. Matlab Functions

`A = imread(filename)`

Reads the image from the file specified by filename, inferring the format of the file from its contents. If filename is a multi-image file, then imread reads the first image in the file.

`I = rgb2gray(RGB)`

Converts the truecolor image RGB to the grayscale intensity image I. The rgb2gray function converts RGB images to grayscale by eliminating the hue and saturation information while retaining the luminance.

`C = normxcorr2(template, A)`

Computes the normalized cross-correlation of the matrices template and A. The matrix A must be larger than the matrix template for the normalization to be meaningful. The values of template cannot all be the same. The resulting matrix C contains the correlation coefficients, which can range in value from -1.0 to 1.0.

`surf(Z)`

Creates a three-dimensional shaded surface from the z components in matrix Z, using x = 1:n and y = 1:m, where [m,n] = size(Z). The height, Z, is a single-valued function defined over a geometrically rectangular grid. Z specifies the color data, as well as surface height, so color is proportional to surface height. The values in Z can be numeric or datetime or duration values.

`[centers,radii] = imfindcircles(A,radiusRange)`

Finds circles with radii in the range specified by radiusRange. The additional output argument, radii, contains the estimated radii corresponding to each circle center in centers.

`BW = edge(I,'Canny',threshold)`

Return all edges that are stronger than threshold. If you do not specify threshold, or if you specify empty brackets ([]), edge chooses the value automatically. threshold is a two-element vector in which the first element is the low threshold, and the second element is the high threshold. If you specify a scalar, edge uses this value for the high value and uses threshold*0.4 for the low threshold.

`B = impyramid(A,direction)`

Computes a Gaussian pyramid reduction or expansion of A by one level, where direction determines whether it's a reduction or an expansion.

5. References

1. Y. Solihin and C. G. Leedham, "Integral ratio: a new class of global thresholding techniques for handwriting images," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 8, pp. 761-768, Aug 1999.
2. http://setosa.io/ev/image-kernels/
3. https://www.mathworks.com/help/images/ref/imfindcircles.html
4. https://www.mathworks.com/help/images/ref/im2bw.html
5. https://www.mathworks.com/help/images/ref/normxcorr2.html
6. https://en.wikipedia.org/wiki/Template_matching