**Lab 7: Machine Vision and Image Processing via MATLAB**

Andrew Jiang, Sharanya Pulapura, Nabib Ahmed

**3.2.2 Forming the Binary Image**

(a) Code:

% load pictures and convert to black and white

back = imread('background.bmp');

imbg = rgb2gray(back);

coins1 = imread('coins1.bmp');

imcoins1 = rgb2gray(coins1);

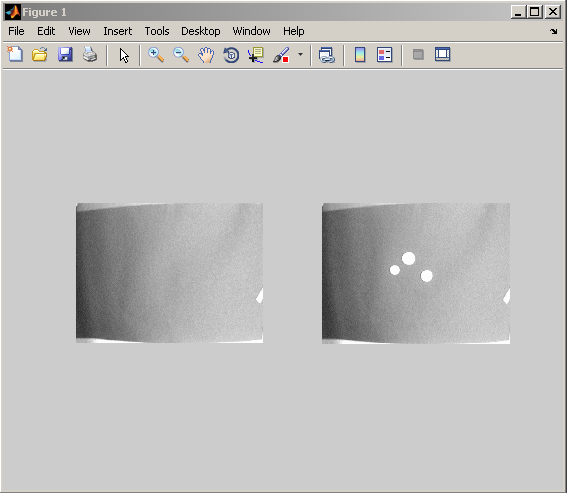
% create subplot of background and coins1

subplot(1,2,1), imshow(imbg);

subplot(1,2,2), imshow(imcoins1);

imdiff = abs(imcoins1-imbg);

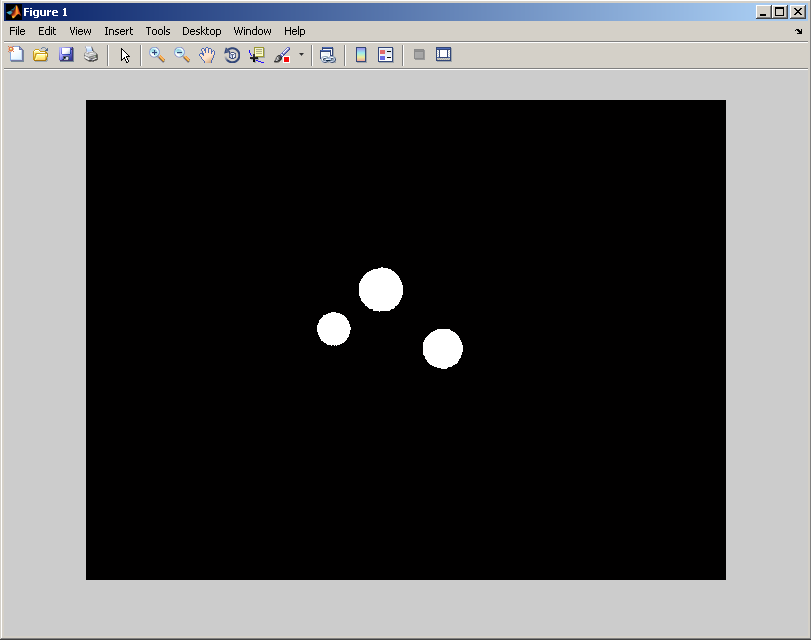
Image:



**3.2.4 Thresholding**

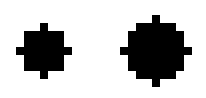
(a) Threshold values between 40 and 60 generally work well.

(b) Image:



**3.2.5 Eliminating Noise**

(a) Left: Radius 3; Right: Radius 4:



(b) (i) The structuring element members comprise all pixels whose centers are no greater than R away from the origin.

**3.2.6 Dilation and Erosion**

(a) Since our original imdiffb has no false positives, we can use the radius as 1 to have a correct image.

(b) Code:

imclean2 = imerode(imdilate(imdiffb, sec), sec);

(c) Once again, our original image had no false positives, so radius of 1 produces a correct image. This is the same value because our original image had no holes or other errors.

**3.2.7 Displaying Your Results**

(a) Code:

% load pictures and c;onvert to black and white

back = imread('background.bmp');

imbg = rgb2gray(back);

coins1 = imread('coins1.bmp');

imcoins1 = rgb2gray(coins1);

% create subplot of background and coins1

subplot(1,3,1), imshow(imbg);

subplot(1,3,2), imshow(imcoins1);

% create difference image b/w background and coins1

imdiff = abs(imcoins1-imbg);

imdiffb = imdiff > 40;

% filter 'noise' in images

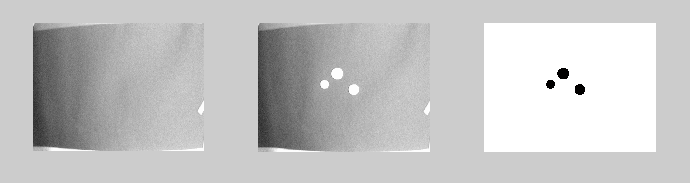
sec = strel('disk',2,0);

imnoholes = imerode(imdilate(imdiffb, sec), sec);

subplot(1,3,3);

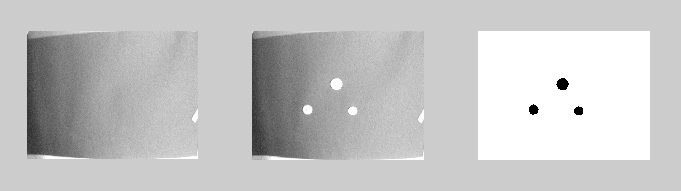
imshow(~imnoholes); 'Detected Coins';

Image:

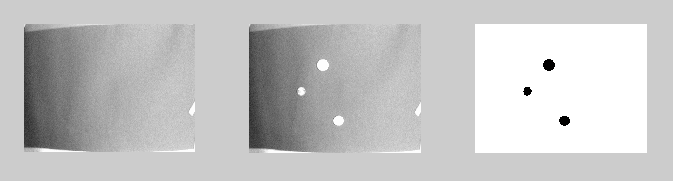


**3.2.8 Analyzing Your Other Coin Images**

(a) coins2.BMP:



coins3.BMP:



**3.3.1 Computing Areas**

(a) The areas of coins 1, 2, and 3 are 874, 1536, and 1259, respectively.

**3.3.2 Streamlined calculation of areas: looping**

(a) It produces the correct values for coins1.BMP

coins2.BMP:

1110

1541

888

coins3.BMP:

774

1528

1154

(b) The average areas are as follows:

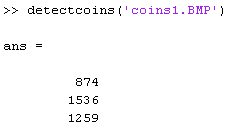
Dime: 845

Nickel: 1174

Quarter: 1535

**4.1 Creating a MATLAB Function**

1. The output of the function is:



The function code is:

function coinareas = detectcoins(coinfilename, threshold, radius1, radius2)

% load pictures and c;onvert to black and white

back = imread('background.bmp');

imbg = rgb2gray(back);

coins1 = imread(coinfilename);

imcoins1 = rgb2gray(coins1);

% create subplot of background and coins1

subplot(1,3,1), imshow(imbg);

subplot(1,3,2), imshow(imcoins1);

% create difference image b/w background and coins1

imdiff = abs(imcoins1-imbg);

imdiffb = imdiff > threshold;

% filter 'noise' in images

seo = strel('disk',radius1,0);

imer = imerode(imdiffb, seo);

imdi = imdilate(imer, seo);

sec = strel('disk',radius2,0);

imnoholes = imerode(imdilate(imdiffb, sec), sec);

subplot(1,3,3);

imshow(~imnoholes); 'Detected Coins';

[imcc, n ] = bwlabel(imnoholes);

coinareas = zeros(n,1); %allocate array for areas

for i=1:n %repeat for each coin

coinareas(i) = sum(imcc(:) == i); %area of ith coin

end;

**4.2 Counting loose change**

(a) Code:

function howmuch = detectcoins(imname)

% load pictures and c;onvert to black and white

imname

back = imread('background.bmp');

imbg = rgb2gray(back);

coins1 = imread(imname);

imshow(coins1)

imcoins1 = rgb2gray(coins1);

imshow(imcoins1)

% create subplot of background and coins1

%subplot(1,3,1), imshow(imbg);

%subplot(1,3,2), imshow(imcoins1);

% create difference image b/w background and coins1

imdiff = abs(imcoins1-imbg);

imdiffb = imdiff > 50;

% filter 'noise' in images

seo = strel('disk',2,0);

imer = imerode(imdiffb, seo);

imdi = imdilate(imer, seo);

sec = strel('disk',10,0);

imnoholes = imerode(imdilate(imdiffb, sec), sec);

%subplot(1,3,3);

imshow(~imnoholes); 'Detected Coins';

[imcc, n] = bwlabel(imnoholes);

n

howmuch = 0; %monetary value of coins

coinareas = zeros(n,1); %allocate array for areas

firstupperbound = 1000; %approx# of pixels in your smallest coin

secondupperbound = 1300; %apprx# of pixels in 2nd smallest coin

for i = 1:n % repeat for each coin

coinareas(i) = sum(imcc(:) == i); %area of ith coin

if coinareas(i) < firstupperbound

howmuch = howmuch + 10 %its a dime

elseif (coinareas(i) >= firstupperbound) & (coinareas(i) < secondupperbound)

howmuch = howmuch + 5 %its a nickel

elseif (coinareas(i) > secondupperbound)

howmuch = howmuch + 25 %its a quarter

end

end