CSE 5524 - Homework #3
09/16/2013
Manjari Akella

1) Using the grayscale images (walk.bmp, bg000.bmp) provided on the WWW site, perform background subtraction 1 using simple image subtraction to identify the object. Experiment with thresholds.

- Used the graythresh() function which returns a normalized threshold value
- Used the following formula to convert normalized value from [0-1] to [0-255]
  level = $e_n (e_{max} - e_{min}) + e_{min}$, where e_n= normalized intensity
  e_{min} = minimum intensity
  e_{max} = maximum intensity
- Threshold computed using graythresh() function was 0.4980. After converting it, the 'level' was set to 126.0039
- Thresholds were tested ranging from (level-50) to (level+20)
- After about (level+20=146.0039), the bottom half was indiscernible
- At about (level-50=76.0039), the top and bottom halves of the girl were *almost* connected

Output

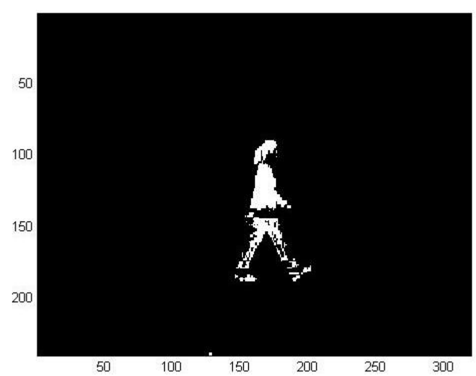

Figure 1: Threshold = 76.0039

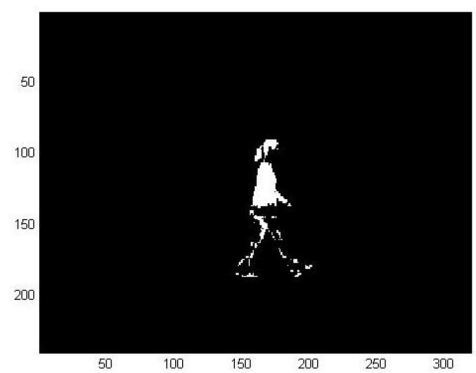

Figure 2: Threshold = 126.0039

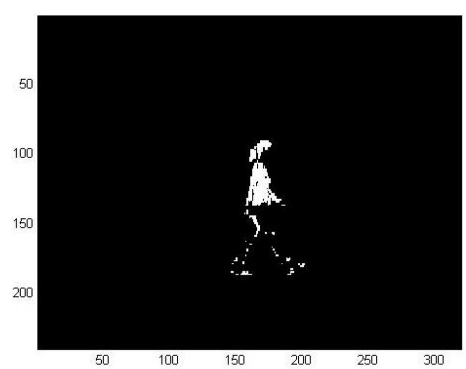

Figure 3: Threshold = 146.0039

2) Using the grayscale images (walk.bmp, bg[000-029].bmp) provided on the WWW site, perform background subtraction 2 using statistical distances. Experiment with thresholds.

- Thresholds were tested ranging from $(3*sigma)^2$ to $(20*sigma)^2$, sigma = standard deviation
- The top and bottom halves were divided. This might possibly be due to same color intensity at the waist area of the girl and the background wall (strip of gray)
- As sigma was increased a lot of stray pixels cleared up
- Background Subtraction 1 gave cleaner and better defined images

<u>Output</u>

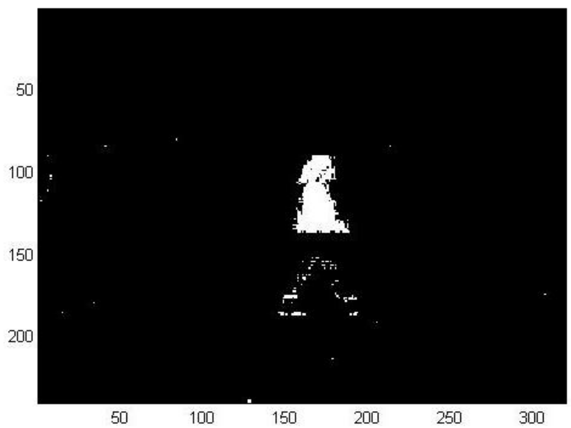

Figure 4: Threshold = $(20*sigma)^2$

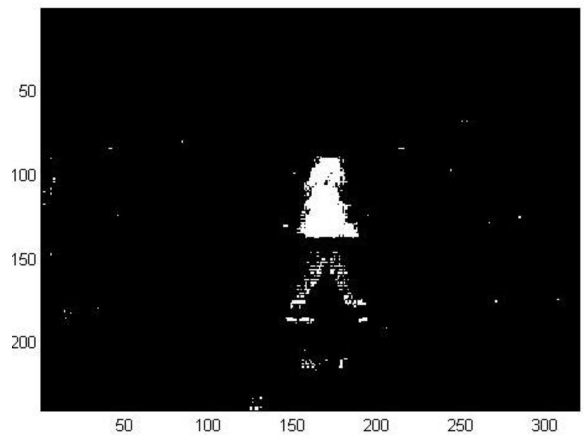

Figure 5: Threshold = $(12*sigma)^2$

3) Dilate the binary image resulting from step 2) using bwmorph() function.

- At T = $(20*sigma)^2$, the slight curve on the face were visible. However, the bottom half was almost lost.
- At T = $(12*sigma)^2$, the curves on the face weren't visible but most of the bottom half was preserved.

Output

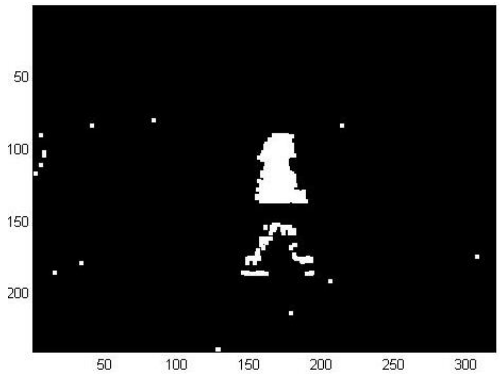

Figure 4: Dilated Image at T=(20*sigma)^2

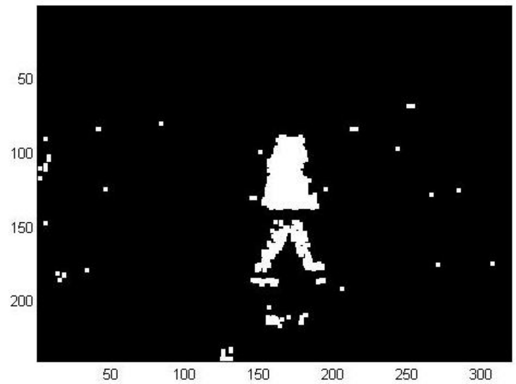

Figure 5: Dilated Image at (T=12*sigma)^2

4) Perform a connected components algorithm, and keep only the largest region in L, (save/display as an image).

- At T = (20*sigma)^2, the slight curve on the face were visible
- At T = (12*sigma)^2, the curves on the face weren't visible . Blob looks more like a bear's head

Output

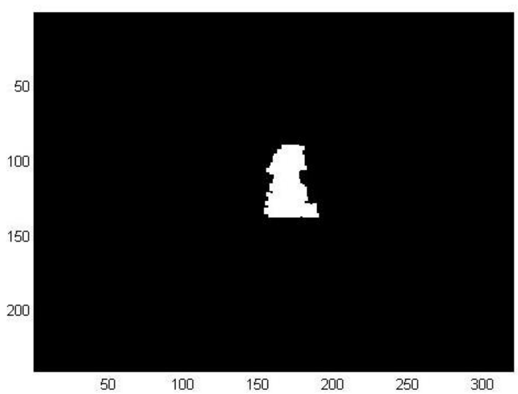

Figure 6: Largest Conncected Component (T=(20*sigma)^2)

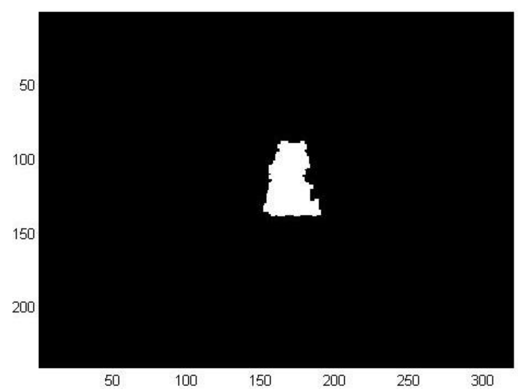

Figure 7: Largest Connected Component(T=(12*sigma)^2)

5) Use the regionprops function on boxIm1.bmp (provided on the class website) to compute its 'Area', 'Centroid', and 'BoundingBox'. Plot the centroid and bounding box on the image.

- Used regionprops() on the image after converting to binary
- Used the 'basic' parameter which returns area, centroid and bounding box features
- All coordinates are w.r.t image axis

Output

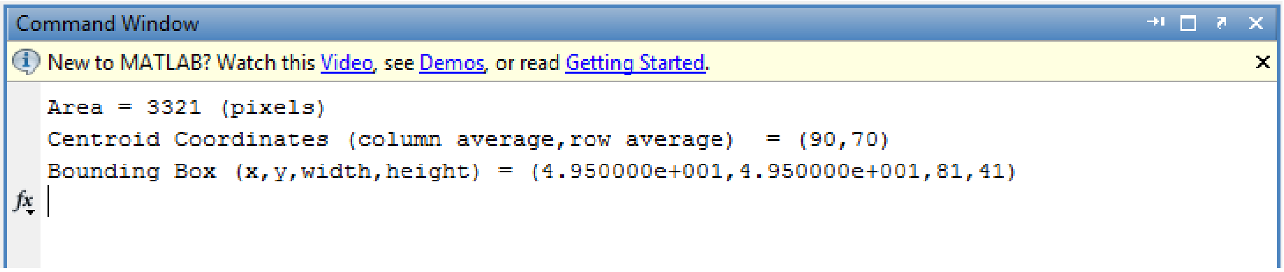

Figure 8: Area, Centroid and Bounding Box Coordinates

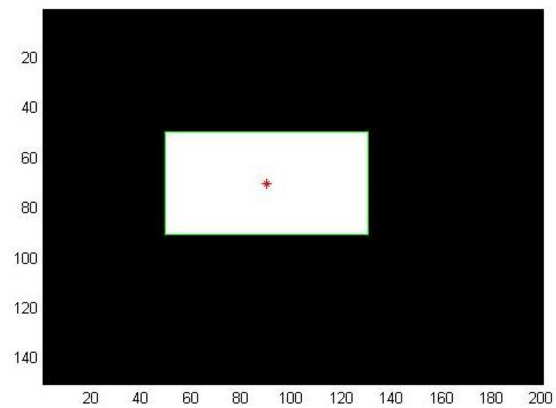

Figure 9: Centroid (red) and Bounding Box (green)

6) Write a function to compute the seven similitude moment shape descriptors. Test and compare results on the rectangle box images 'boxIm[1-4].bmp' on the website. How do they change across the box images?

- Output vector N is of the form [n_{02} n_{03} n_{11} n_{12} n_{20} n_{21} n_{30}]
- For boxIm1 and boxIm2, these values were exactly the same. This is because boxIm2 is just a translated version of boxIm1 and similitude moments are invariant to translation
- For boxIm3, the values were *almost* (very slight deviation) the same as boxIm1 and boxIm2. The slight difference in value may have arisen due to the factor of scale. For the purpose of discussion we can ignore the slight error and say that these are equal. This is because similitude moments are invariant to scale
- For boxIm4, these values are different that the previous ones. In fact they are exactly opposite to the previous result. This is because the image is rotated by 90 degrees
- Further for boxIm1, boxIm2, boxIm3 the value of $n_{02} < n_{20}$. This is because the image is spread over columns(x-axis), i.e.-length along this axis is more
- For boxIm4, this is exactly opposite as the length of the image stretches over the rows(y-axis)

Output

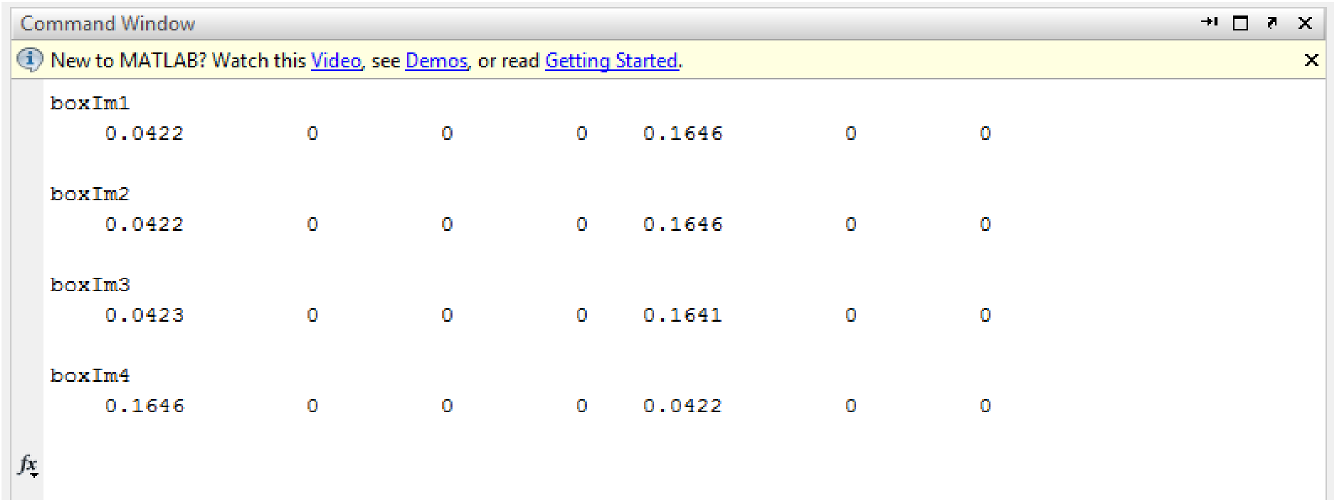

Figure 10: Similitude Moments of the box images

CODE

1). similitudeMoments function

```
function [ Nval ] = similitudeMoments( Im, i, j )
% Computes row and column averages of the given image and then the similitude
moment for
% i,j and returns the value
n=0;
m00=sum(sum(Im));
m10=0;
m01=0;
e=((i+j)/2)+1;
% Calculate row and column averages
for r=1:size(Im,1)
    for c=1:size(Im,2)
        m10 = m10 + r*Im(r,c);
        m01 = m01 + c*Im(r,c);
    end
end
% Row average and column average (Centriod)
rc = m10/m00;
cc = m01/m00;
% Calculate the numerator of the moment
for r=1:size(Im,1)
    for c=1:size(Im,2)
       n = n + ((c-cc)^i)*((r-rc)^j)*Im(r,c);
    end
end
% Calculate the value of the moment by dividing by m00
Nval = n/(m00^e);
end
```

2). HW3.m script

```
% Manjari Akella
% CSE5524 - HW3
% 09/16/2013

mkdir('Output');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Question 1

refresh();
R = double(imread('given_pics/bg000.bmp'));
I = double(imread('given_pics/walk.bmp'));
% Computes a threshold for the image
normlevel = graythresh(I);
% Convert normalized value from [0-1] to [0-255] range
level = (normlevel*(max(max(I))-min(min(I))))-min(min(I));
% Check the image for threshold values for -50 to +20 of the returned
threshold
% value
figure('Name','Q1: Background Subtraction 1 Thresholds','NumberTitle','off');
for i=-50:10:20
    fprintf('Threshold=%d\n',level+i);
    bs1Im = (abs((R-I)))>(level+i);
    imagesc(bs1Im);
    colormap('gray');
    pause(1);
end
% Pick Threshold = 76.0039
bs1Im = (abs((R-I)))>(76.0039);
imwrite(bs1Im,'Output/Q1_B_Sub1.bmp');
pause;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Question 2

refresh();
I = double(imread('given_pics/walk.bmp'));
% Read all background images into a 3D matrix (Each slice is an
% image(240x320)
N=30;
for  i=1:N
filename = sprintf('given_pics/bg%03d.bmp', i-1);
Im(:,:,i) = double(imread(filename));
end

% Model mean and standard deviation statistics for all background images
meanIm = mean(Im,3);
sdIm = std(Im,0,3);

% Threshold the image to subtract the background
figure('Name','Q2: Background Subtraction 2 Thresholds','NumberTitle','off');
for i=3:20
    fprintf('Threshold=(%d*sigma)^2\n',i);
```

```
    bs2Im =(((I-meanIm).^2)./(sdIm.^2))>((i.*sdIm).^2);
    imagesc(bs2Im);
    colormap('gray');
    axis('image');
    pause(1);
end
% Pick Threshold = (20*sigma)^2
bs2Im =(((I-meanIm).^2)./(sdIm.^2))>((20.*sdIm).^2);
imwrite(bs2Im,'Output/Q2_B_Sub2.bmp');
pause;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Question 3

d_bs2Im = bwmorph(bs2Im, 'dilate');
figure('Name','Q3: Dialated Image','NumberTitle','off'),imagesc(d_bs2Im);
colormap('gray');
axis('image');
imwrite(d_bs2Im,'Output/Q3_Dilated_B_Sub2.bmp');
pause;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Question 4

% Label the regions
[L, num] = bwlabel(d_bs2Im, 8);
% Find out more about the connected components
CC = bwconncomp(L);
% Find number of pixels in each label
numPixels = cellfun('size',CC.PixelIdxList,1);
% Find label with maximum number of pixels
[biggest,idx] = max(numPixels);
% Label those indices 1, rest all 0
for i=1:size(L,1)
    for j=1:size(L,2)
        if (L(i,j)==idx)
            L(i,j) = 1;
        else
            L(i,j) = 0;
        end
    end
end
figure('Name','Q4: Largest Component','NumberTitle','off'),imagesc(L);
colormap('gray');
axis('image');
imwrite(L,'Output/Q4_Largest_Connected_Component.bmp');
pause;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Question 5

refresh();
Im = im2bw(double(imread('given_pics/boxIm1.bmp')));
% 'Basic' returns area,centroid and bounding box
s = regionprops(Im,'basic');
c = s.Centroid;
```

```
a = s.Area;
bb = s.BoundingBox;
% Coordinates displayed are w.r.t image axis
fprintf('Area = %d (pixels)\n',a);
fprintf('Centroid Coordinates (column average,row average)  =
(%d,%d)\n',c(1,1),c(1,2));
fprintf('Bounding Box (x,y,width,height) =
(%d,%d,%d,%d)\n',bb(1,1),bb(1,2),bb(1,3),bb(1,4));
figure('Name','Q5: Centroid & Bounding Box','NumberTitle','off'),imagesc(Im);
colormap('gray');
axis('image');
hold on
rectangle('Position',bb,'EdgeColor','g','LineWidth',1);
plot(s.Centroid(1,1), s.Centroid(1,2), 'r*');
pause;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Question 6

refresh();
% Loop for each box image
for i=1:4
    Im =
im2bw(double(imread(strcat('given_pics/boxIm',num2str(i),'.bmp'))));
    % Find out the seven moment descriptors
    n02 = similitudeMoments(Im,0,2);
    n03 = similitudeMoments(Im,0,3);
    n11 = similitudeMoments(Im,1,1);
    n12 = similitudeMoments(Im,1,2);
    n20 = similitudeMoments(Im,2,0);
    n21 = similitudeMoments(Im,2,1);
    n30 = similitudeMoments(Im,3,0);
    fprintf('boxIm%d\n',i);
    N = [n02, n03, n11, n12, n20, n21, n30];
    disp(N);
end
%%%%%%%%%%%%%%%%%%%%%% END %%%%%%%%%%%%%%%%%%%%%%
```