Lab 4

# Q1

clf,clc,clear;

%1. Get PCB.gif using UI. keep for late use

%[f\_name,f\_path]=uigetfile();

%filepath=[f\_path f\_name];

%I=imread(filepath);

I=imread("PCB.gif");

PCB\_fig=figure();

imshow(I,[]);

title("PCB.gif");

saveas(PCB\_fig,"0PCB\_fig",'jpg');

Hist\_I\_fig=figure();

%showing the histogram

Hist\_I=imhist(I(:));

bar(Hist\_I);

title("hist of I");

saveas(Hist\_I\_fig,"1Hist\_I\_fig",'jpg')

%we found that the major pixels are around 0 and 80. we will take 40.

thresh=graythresh(Hist\_I)\*64

%the value is close. we got 31.8745

%2. binary convert

I\_bin\_fig=figure();

%All values that are larger than thresh will be 1, else 0.

I\_bin=I>thresh;

imshow(I\_bin);

title("I binary")

saveas(I\_bin\_fig,"2I\_bin\_fig",'jpg');

%Inverse image, using NOT

not\_I\_bin\_fig=figure();

not\_I\_bin=not(I\_bin);

imshow(not\_I\_bin);

title("inverse of I");

saveas(not\_I\_bin\_fig,"3not\_I\_bin\_fig",'jpg');

%4.

%creating one mask for the background, meaning that the edges are white,

%rest is zero. This will allow us do get rid of the background

blackmask=zeros(size(I));

blackmask(1:end,1)=1;

blackmask(1:end,end)=1;

blackmask(1,1:end)=1;

blackmask(end,1:end)=1;

blackmask\_fig=figure();

imshow(blackmask);

title("The mask");

saveas(blackmask\_fig,"4blackmask\_fig",'jpg');

%making one label for the background using the blackmask and OR

not\_I\_bin2=blackmask|not\_I\_bin;

%label the masked image

Label\_not\_I\_bin=bwlabel(not\_I\_bin2);

labeled\_not\_fig=figure();

imshow(Label\_not\_I\_bin,[]);

title("Labeled NOT(I)");

saveas(labeled\_not\_fig,"5labeled\_not\_fig",'jpg');

%binary convert for the holes, the value of the background is 1 so the

%holes will be extracted.

Label\_not\_I\_bin=Label\_not\_I\_bin>1;

holes\_fig=figure();

imshow(uint8(Label\_not\_I\_bin),[]);

title("holes");

saveas(holes\_fig,"6holes\_fig",'jpg');

output\_fig=figure();

output=Label\_not\_I\_bin|I\_bin;

imshow(output);

title("Output");

saveas(output\_fig,"7output",'jpg');

%5.

output\_fig\_cirlces=figure();

imshow(output);

title("Output with cirlces");

%getting the stast of the objects.

stats = regionprops('table',Label\_not\_I\_bin,'Centroid',...

'MajorAxisLength','MinorAxisLength');

%centers- all the centers of the object

centers = stats.Centroid

%getting the mean of major and the minor axis diameters.

diameters = mean([stats.MajorAxisLength stats.MinorAxisLength],2);

radii = diameters/2

% %another way to calculate:

% radii=stats.MajorAxisLength/2;

hold on

viscircles(centers,radii);

hold off

saveas(output\_fig\_cirlces,"8output\_with\_circles",'jpg');

# Q2

vidobj = VideoReader('Weather\_Cam.avi');

numFrames = get(vidobj, 'NumberOfFrames');

%getting the best correlation

%normxcorr2(

%fit the result

%tform=fitgeotrans

%imwarp

Frame\_Avrg=im2double(read(vidobj,1));

PrelabFrame\_Avrg=Frame\_Avrg;

% (161, 198) [0.44 0.25 0.26]

% (194, 216) [0.27 0.26 0.29]

%(418, 161) 0.07

%(443, 191) 0.24

%create masks

mask = rgb2gray(Frame\_Avrg);

mask1=mask([198:216], [161:194]);

mask2=mask([161:191],[418:443]);

masks=figure();

subplot(2,1,1);

imshow(mask1);title("mask1");

subplot(2,1,2);

imshow(mask2);title("mask2");

saveas(masks,"9masks",'jpg');

%choose points on frame

x1 = ((161 + 194)/2);

y1 = ((198 + 216)/2);

x2 = ((418 + 443)/2);

y2 = ((161 + 191)/2);

original\_point= [y1 x1; y2 x2];

%transform and avrg

N\_avg=16;

for i=2:N\_avg

Frame=im2double(read(vidobj, i));

PrelabFrame=Frame;

Frame\_bw=rgb2gray(Frame);

%getting the correlation

corr1\_frame=normxcorr2(mask1,Frame\_bw);

corr2\_frame=normxcorr2(mask2,Frame\_bw);

%getting the shifted objects corrdinates

[shifted\_y1, shifted\_x1] = find(corr1\_frame == max(max(corr1\_frame)));

[shifted\_y2, shifted\_x2] = find(corr2\_frame == max(max(corr2\_frame)));

%creating a shifted point

shifted\_point = [shifted\_y1 shifted\_x1; shifted\_y2 shifted\_x2];

%getting the transform using the original point and the shifted point

tform = fitgeotform2d(shifted\_point,original\_point,"similarity");

%affine output

sameAsInput = affineOutputView(size(Frame),tform,"BoundsStyle","SameAsInput");

%fixing the image

imwarp(Frame,tform,"OutputView",sameAsInput);

%summing the frames

Frame\_Avrg= Frame\_Avrg+Frame;

PrelabFrame\_Avrg=PrelabFrame\_Avrg+PrelabFrame;

end

%getting the average, dividing by the number of frames

PrelabFrame\_Avrg=PrelabFrame\_Avrg/N\_avg;

Frame\_Avrg= Frame\_Avrg/N\_avg;

%calculate std

std\_Frame = std(Frame\_Avrg(:));

std\_Prelab = std(PrelabFrame\_Avrg(:));

std\_Original = std(Frame(:));

%results show

montages=figure();

montage({Frame,PrelabFrame\_Avrg,Frame\_Avrg},"Size",[1 3]);

title('Original Frame(left)---Pre Lab Frame(center)---After Transform(right)');

saveas(montages,"10montage\_of\_3\_imgs",'jpg');

hists=figure();

subplot(3,1,1);

hist(Frame(:),255);

title(['Original hist {\sigma} = ' num2str(std\_Original)]); grid on;

subplot(3,1,2);

hist(Frame\_Avrg(:),255);

title(['Transform And Avrg hist {\sigma} = ' num2str(std\_Frame)]); grid on;

subplot(3,1,3);

hist(PrelabFrame\_Avrg(:),255);

title(['Prelab Algorithm hist {\sigma} = ' num2str(std\_Prelab)]); grid on;

saveas(hists,"11histograms",'jpg');

window\_crop=figure();

subplot(3,2,1);

imshow(Frame(250:350,:,:));

title('Original Pic Window Crop');

subplot(3,2,2);

plot(Frame(340,:));grid on;

title('Power plot Original Window Crop')

subplot(3,2,3);

imshow(Frame\_Avrg(250:350,:,:));

title('Transform And Avrg Window Crop');

subplot(3,2,4);

plot(Frame\_Avrg(340,:));grid on;

title('Power plot Transform And Avrg Window Crop');

subplot(3,2,5);

imshow(PrelabFrame\_Avrg(250:350,:,:));

title('Prelab Algorithm Window Crop');

subplot(3,2,6);

plot(PrelabFrame\_Avrg(340,:)); grid on;

title('Power plot Prelab Algorithm Window Crop');

saveas(window\_crop,"12window\_crop",'jpg');