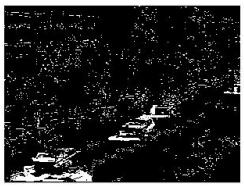
Image and Video Processing
Matlab Homework 5

Name: Amitesh Kumar Sah
ID: N19714360

Question 1

```
% 1.Write a program that can:
% i) read in a short video clip (keep up to only, say 30 frames);
% ii)Compare each two successive frames to extract pixels with significant
%    change (absolute difference greater than a threshold T), set such pixels
%    to white, while keeping other pixels as black;
% iii) Display the successive thresholded difference images as a movie;
% iv) Save the successive difference images as an ".avi" file.


xyloObj=VideoReader('Street.MPG');
nFrames=xyloObj.NumberOfFrames;
vidHeight=xyloObj.Height;
vidWidth=xyloObj.Width;
mov(1:30)= struct('cdata', zeros(vidHeight, vidWidth, 3, 'uint8'),'colormap', []);
%select each 2 successive frames
for i=1:1:30
frame1 =read(xyloObj,i);
frame2=read(xyloObj,i+1);
frame1=uint8(frame1);
frame2=uint8(frame2);
diff=abs(frame1-frame2);
mov(i).cdata=diff;
end
%display the difference of two frames
diff=rgb2gray(diff);
diff=(diff>30)*255;
diff=uint8(diff);
subplot(1,3,1);imshow(frame1,[]);
subplot(1,3,2);imshow(frame2,[]);
subplot(1,3,3);imshow(diff,[]);
% show the difference as a movie at framerate = 5
% movie(figure, mov, 1, 5);
% save the difference as a ".avi" file
myObj=VideoWriter('newfilr.avi');
myObj.FrameRate=5;
open(myObj);
for i=1:30
temp=mov(i).cdata;
writeVideo(myObj,temp);
end
close(myObj);
hf = figure; % Create New Display Window
set(hf, 'position', [150 150 vidWidth vidHeight]) % Set window Location
movie(hf, mov, 1, xyloObj.FrameRate); % Display video sequence
or:
```

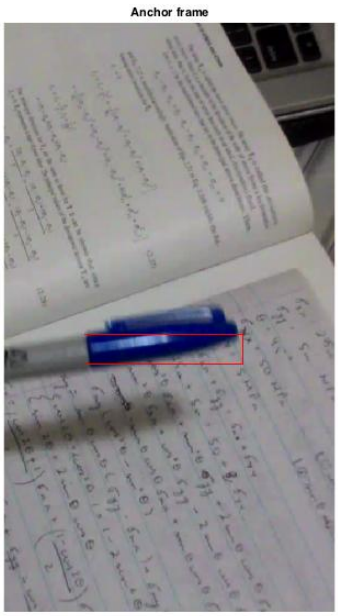
```
v = VideoReader('Street.MPG');
mov = read(v);
mov_new = diff(mov,1,4);% if 4 =>1 then it is the x ax diff, 4=>2 y ax diff
new_vid = VideoWriter('Street_new');
open(new_vid);
writeVideo(new_vid,mov_new);
close(new_vid);
```

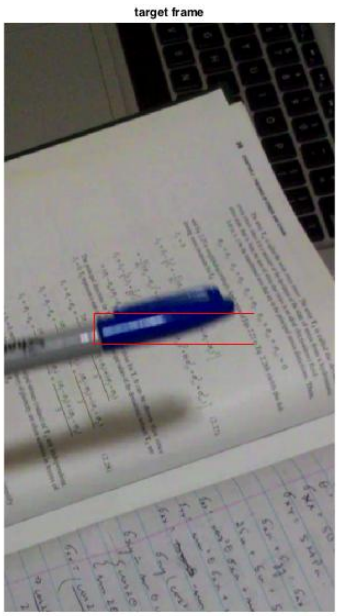


```
% Question 2:
% Read the sample program for template matching given in the lecture note (templatematching( ))%
and understand how it works. Select two video frames from a video that contain the same object or
person with shifted positions. Identify a bounding box for the object in one frame. Write your own
program that can help find its corresponding position in another frame. Was the result accurate? If not,
give some reasons that may have contributed to the error.


xyloObj = VideoReader('pen4.mp4'); % Create Object
nFrames = xyloObj.NumberOfFrames; % Obtain Frame Number
vidHeight = xyloObj.Height; % Obtain Image Height
vidWidth = xyloObj.Width; % Obtain Image Width
mov(1:nFrames)= struct('cdata', zeros(vidHeight, vidWidth, 3, 'uint8'),'colormap', []);
target_frame = read(xyloObj, 5); % Read in 5th Frame
for k = 1 : nFrames-1
```

```
mov(k).cdata = read(xyloObj, k); % Load video into MOV
end
 hf = figure; % Create New Display Window
set(hf, 'position', [150 150 vidWidth vidHeight]) % Set window Location
 movie(hf, mov, 1, xyloObj.FrameRate); % Display video sequence
a=mov(2).cdata;
b=mov(20).cdata;
subplot(2,2,1);imshow(a);
subplot(2,2,2);imshow(b);
subplot(2,2,3);imshow(mov(40).cdata);
subplot(2,2,4);imshow(mov(55).cdata);
imwrite(a,'a.jpg');
imwrite(b,'b.jpg');
figure,imshow(a,[])
figure,imshow(b,[])

a1=rgb2gray(a);
b1=rgb2gray(b);
a2=int16(a1);
b2=int16(b1);

figure, imshow(a2);
figure, imshow(b2);
x0=590;y0=860;
 x1=850;y1=910;
 Rx=600;Ry=300;
 template=a2(y0:y1,x0:x1);
 [xm,ym,matchblock]=EBMA(template,b2,x0,y0,Rx,Ry);
disp('xm=');
disp(xm);
disp('ym=');
disp(ym);
disp('true location');
disp('x= 495,y=580');
[r,c]=size(matchblock);
 sh1 = insertShape(a1, 'rectangle', [x0 y0 x1-x0 y1-y0],'Color', 'red');
figure,subplot(1,2,1),imshow(sh1,[]); title('Anchor frame');
 sh2 = insertShape(b1, 'rectangle', [xm ym c r],'Color', 'red');
subplot(1,2,2),imshow(sh2,[]);title('target frame');


% EMBA (templatematching())function:
function [xm,ym,matchblock]=EBMA(template,img,x0,y0,Rx, Ry)
%x0,y0 define the location of the top left corner of the template in the previous image
%xm,ym define the matched position of the template in the current image
%R is the search range
%template and img should be saved in integer or float
[H,W]=size(img);
[BH,BW]=size(template);
maxerror=BH*BW*255;
xm=x0;
```

```
ym=y0;
for (k=max(1,x0-Rx):min(W-BW,x0+Rx))
for (l=max(1,y0-Ry):min(H-BH,y0+Ry))
candidate=img(l:l+BH-1,k:k+BW-1);
error=sum(sum(abs(template-candidate))); %SAD
if (error<maxerror)
xm=k;ym=l;matchblock=candidate;
maxerror=error;
end
end
end
```





Result
xm= 496
ym= 574

true location
x= 495,y=580

The result obtained is approximately close to the bounding box of the anchor image. There is small error due to scaling, and rotation.