Assignment #6

- Adwaith Venkataraman (Andrew ID: adwaithv)

Question 1

Project Proposed **:** Eye Tracking using OpenCV
Name **:** Adwaith Venkataraman
Data Source **:** Self
Method **:** Utilizing the Eye Tracking device which Prof. Cai possesses.
Expected Results **:** Detection of eye blinking, a few emotions, etc.

Note: The project information provided in this assignment is a very brief one as it was discussed with Prof. Cai only on 21st October and yet to receive/discuss more on this with him. The exact expected result will be specified in the next submission.

Question 2

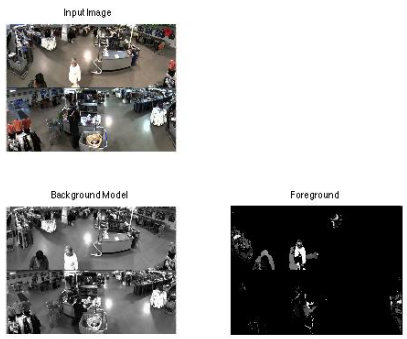
Analysis:

1) The video utilized was read, and the number of frames was calculated.
2) The first frame of the video was then stored as the background, and was subsequently converted to grayscale.
3) Then, the dimension of the background image was calculated and the threshold was set.
4) Traversing through each frame of the video, and in that each pixel of every frame, the difference between that frame and the background was calculated as delta.
5) Delta was then compared with the set threshold.
6) Based on the above comparison, the value of the foreground's corresponding pixel was assigned a value.
7) Finally, the average pixel value was computed and the foreground was displayed.
8) Based on the number of frames and the average, the motion energy image was displayed.

Results:

The input image from the video used is shown below, along with the background and foreground images derived.

Also, the obtained Motion energy Image is shown.

	
Input, Background and Foreground Images	Motion Energy Image

Matlab Code:

```
input_vid = VideoReader('store1(1).mp4'); %read the video from file
no_frames = input_vid.NumberOfFrames; %compute the number of frames
stable_bg = read(input_vid, 1); %store the background image as the
first frame of the video
stable_bg_gr = rgb2gray(stable_bg); %convert the background to
grayscale
[height width] = size(stable_bg_gr); %calculate the dimensions of the
background
foreground = zeros(height, width); %calculate the dimensions of the
foreground with zeros function
average = zeros(height, width); %compute the average height and width
threshold = 20; %set a threshold

for i = 2:no_frames %traverse from the second frame to the last frame
of the video

     current_frame = read(input_vid, i); %read the 'i'th frame
     current_frame_gr = rgb2gray(current_frame); %convert the frame to
grayscale
     delta = abs(double(current_frame_gr) - double(stable_bg_gr));
%compute the difference between the absolute value and type casted
double value

     for j = 1:width %traverse through the pixels of the width

          for k = 1:height %traverse through the pixels of the height

               if (delta(k, j) > threshold) %compare the difference with
the threshold
                  foreground(k, j) = current_frame_gr(k, j); %if the
difference is greater, set the current frame accordingly
               else
                   foreground(k, j) = 0; %otherwise zero out the
foreground
               end

          end
     end
```

```
     average = average + foreground; %update the average
     figure(1), subplot(2, 2, 1), imshow(current_frame); %plot the
input image i.e., the current frame
     title('Input Image')
     subplot(2, 2, 3), imshow(uint8(stable_bg_gr)) %display the
grayscale background
     title('Background Model')
     subplot(2, 2, 4), imshow(uint8(foreground)) %display the
foreground
     title('Foreground')

     stable_bg_gr = current_frame_gr; %update the background by
assigning the current frame to it

end

result = average/no_frames; %let the resultant image be the average
image array of all frames
figure(2), imshow(result); %display the motion energy image
title('Motion Energy Image');
```

Question 3

Analysis:

1) The video utilized was read, and the number of frames was calculated.
2) The first frame of the video was then stored as the background, and was subsequently converted to grayscale.
3) Then, the dimension of the background image was calculated and subsequently the threshold and speed values were set.
4) Traversing through each frame of the video, and in that each pixel of every frame, the difference between that frame and the background was calculated as delta.
5) Delta was then compared with the set threshold.
6) Based on the above comparison, the value of the motion history image's corresponding pixel was assigned a value.
7) The foreground image was then filtered out.
8) Finally, the average pixel value was computed and the foreground was displayed.
9) Based on the number of frames and the average, the motion history image was displayed.
10) A motion energy image displays or highlights regions of the image where any movement or motion was detected. However, a motion history image consisting of brighter pixels indicate regions of the image where movement or motion was more recent.
11) A motion history image contains information regarding how the object has moved during the video. But a motion energy image indicates only how the object has moved, but not when or the sequence of motion.

Results:

<table>
<tr>
<td> 
</td>
<td> 
</td>
</tr>
<tr>
<td>Input, Background and Foreground Images</td>
<td>Motion History Image</td>
</tr>
</table>

Matlab Code:

```
input_vid = VideoReader('store1(1).mp4'); %read the video from file
no_frames = input_vid.NumberOfFrames; %compute the number of frames
stable_bg = read(input_vid, 1); %store the background image as the
first frame of the video
stable_bg_gr = rgb2gray(stable_bg); %convert the background to
```

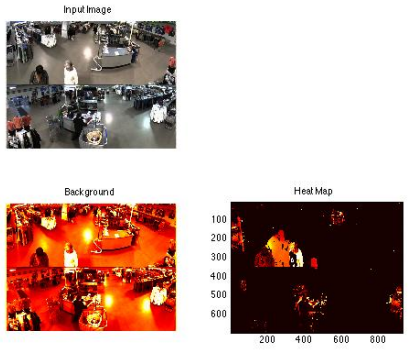
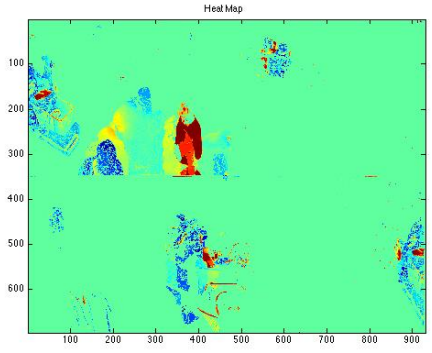
```
grayscale
[height, width] = size(stable_bg_gr); %calculate the dimensions of the
background
foreground = zeros(height, width); %calculate the dimensions of the
foreground with zeros function
mhi = zeros(height, width); %compute the average height and width

speed = 1.5; %setting the decremental value for updating the motion
history image
pause on;
threshold = 20; %set a threshold

for i = 2:no_frames; %traverse from the second frame to the last frame
of the video

    current_frame = read(input_vid, i); %read the 'i'th frame
    current_frame_gray = rgb2gray(current_frame); %convert the frame to
grayscale

    delta = abs(double(current_frame_gray) - double(stable_bg_gr));
%compute the difference between the absolute value and type casted
double value

    for j = 1: width %traverse through the pixels of the width

        for k = 1:height %traverse through the pixels of the height

            if (delta(k, j) > threshold) %compare the difference with
the threshold
                mhi(k, j) = current_frame_gray(k, j); %if the
difference is greater, set the motion history image accordingly
            else
                if (mhi(k, j) ~= 0) %check if the array value is not
equal to zero
                    mhi(k,j) = mhi(k,j) - speed; %if true, decrement
the value by 'speed'
                    foreground(k, j) = mhi(k, j); %subsequently, set
the foreground
                else
                    foreground(k, j) = 0; %if not true, set the array
value to zero
                end
            end
        end
    end
    fg_filtered = medfilt2(foreground, [4, 4]);
    figure(1), subplot(2, 2, 1), imshow(current_frame);
    title('Input Image');
    subplot(2, 2, 3), imshow(uint8(stable_bg_gr))
    title('Background');
    subplot(2, 2, 4), imshow(uint8(fg_filtered));
    title('Motion History Image');
    stable_bg_gr = current_frame_gray; %update the background by
assigning the current frame to it
end
figure (2), imshow(mhi);
title('Motion History Image');
```

Question 4

Analysis:

1) The video utilized was read, and the number of frames was calculated.
2) The first frame of the video was then stored as the background, and was subsequently converted to grayscale.
3) Then, the dimension of the background image was calculated and subsequently the threshold and speed values were set.
4) Traversing through each frame of the video, and in that each pixel of every frame, the difference between that frame and the background was calculated as delta.
5) Delta was then compared with the set threshold.
6) Based on the above comparison, the value of the motion history image's corresponding pixel was assigned a value.
7) The foreground image was then filtered out.
8) From the foreground image, the heat map can be plotted using the 'imagesc' function.
9) From the heat map, we would be able to visualize the three dimensional data of the given video.
10) Apart from the existing Cartesian coordinates, the third dimension in this case was the color i.e., the heat intensity at a region, in comparison to the absolute maximum of the domain dataset. Generally, red represents the maxima and blue represents the minima.
11) In this video, the distance of the human from the camera was obtained based on the heat intensity of the pixels corresponding to each person. It can be seen that one person is shown red and the others are blue, indicating the approximate position or distance of the person from the camera.

Results:

<table>
<tr>
<td> 
</td>
<td> 
</td>
</tr>
<tr>
<td>Input, Background and Heat Images</td>
<td>Heat Map</td>
</tr>
</table>

Matlab Code:

```
input_vid = VideoReader('store1(1).mp4'); %read the video from file
```

```
no_frames = input_vid.NumberOfFrames; %compute the number of frames
stable_bg = read(input_vid, 1); %store the background image as the
first frame of the video
stable_bg_gr = rgb2gray(stable_bg); %convert the background to
grayscale
[height, width] = size(stable_bg_gr); %calculate the dimensions of the
background
foreground = zeros(height, width); %calculate the dimensions of the
foreground with zeros function
mhi = zeros(height, width); %compute the average height and width

speed = 1.5; %setting the decremental value for updating the motion
history image
pause on;
threshold = 20; %set a threshold

for i = 2: no_frames; %traverse from the second frame to the last frame
of the video

    current_frame = read(input_vid, i); %read the 'i'th frame
    current_frame_gray = rgb2gray(current_frame); %convert the frame to
grayscale

    delta = abs(double(current_frame_gray) - double(stable_bg_gr));
%compute the difference between the absolute value and type casted
double value

    for j = 1: width %traverse through the pixels of the width

        for k = 1:height %traverse through the pixels of the height

            if (delta(k, j) > threshold) %compare the difference with
the threshold
                mhi(k, j) = current_frame_gray(k, j); %if the
difference is greater, set the motion history image accordingly
            else
                if (mhi(k, j) ~= 0) %check if the array value is not
equal to zero
                    mhi(k,j) = mhi(k,j) - speed; %if true, decrement
the value by 'speed'
                    foreground(k, j) = mhi(k, j); %subsequently, set
the foreground
                else
                    foreground(k, j) = 0; %if not true, set the array
value to zero
                end
            end
        end
    end

    fg_filtered = medfilt2(foreground, [4, 4]);

    figure(1), subplot(2, 2, 1), imshow(current_frame);
    title('Input Image');

    subplot(2, 2, 3), imshow(uint8(stable_bg_gr))
```

```
    title('Background');

    colormap('hot'); %plotting the heat map of the image

    subplot(2, 2, 4), imagesc(foreground);
    title('Heat Map');

    stable_bg_gr = current_frame_gray; %update the background by
assigning the current frame to it

end

figure (2), imagesc(foreground);
title('Heat Map');
```