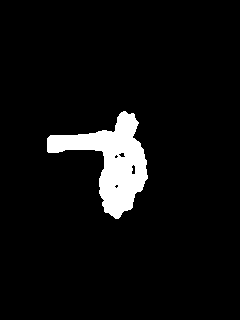
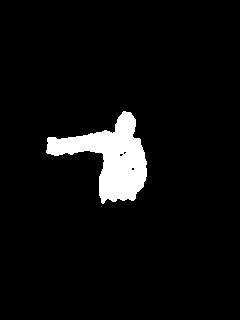
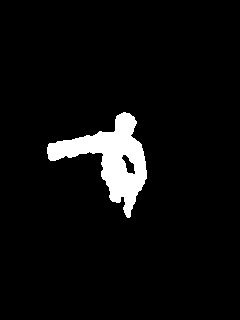
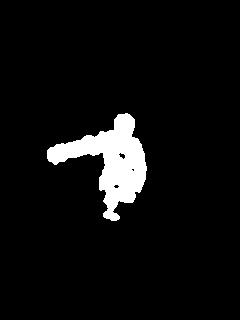
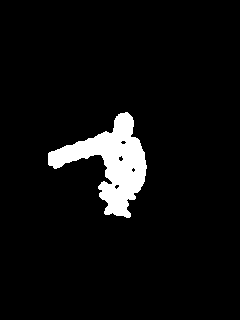
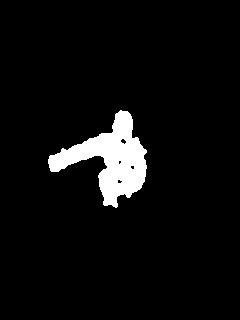
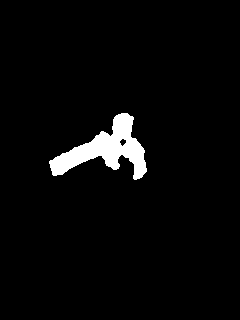
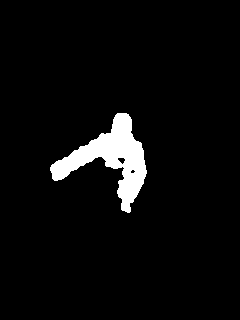
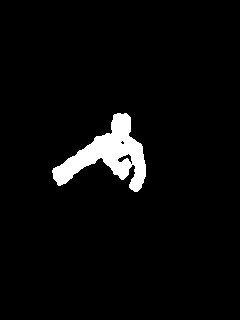
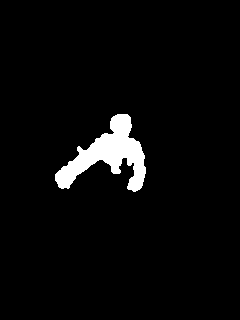
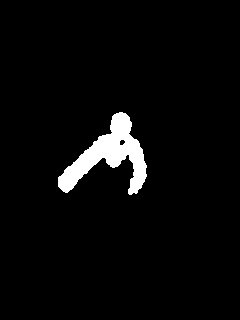
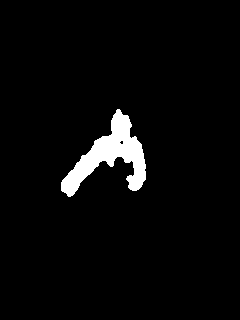
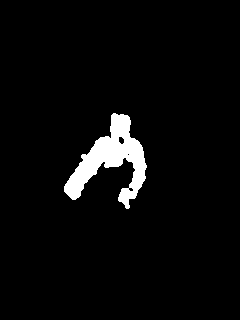
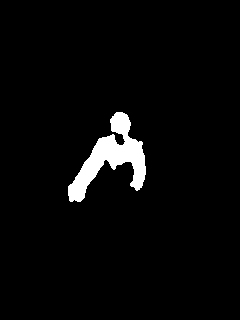
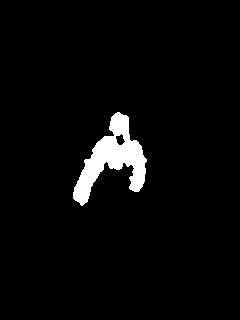
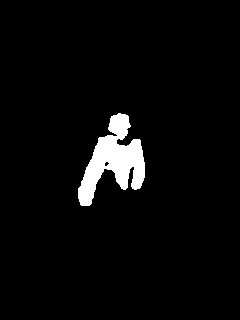
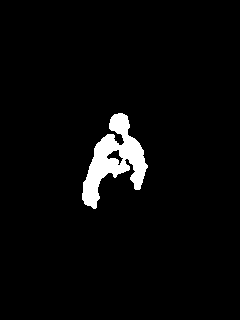
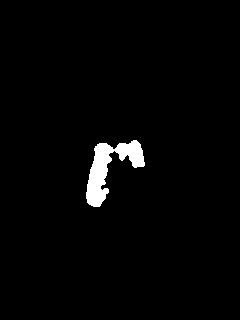
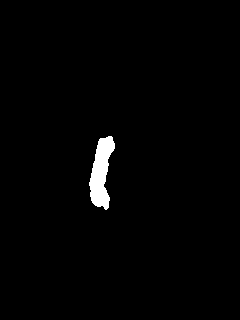
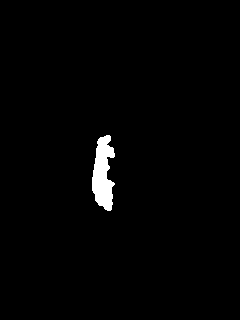
CSE 5522 HW5

Yi Zhao

1. I tried lots of T and here I use T = 4 for differencing. After differencing, I removed tiny regions, perform dilation and finally smooth through median filter. Here is the resulted differencing image from Image 2 to image 22. Initially, the only thing in the image is the left hand because other body parts does not move too much. Later, the head and upper body are shown in the image because these parts are moving as her arm moving up.



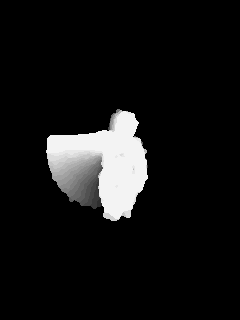
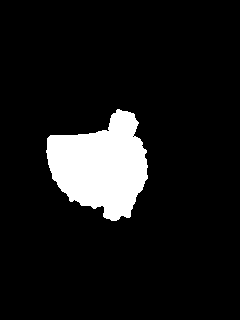
1. Here is the resulted MEI (left) and MHI (right).

Here is the similitude of MEI (top) and MHI (bottom).

**0.0822 0.0002 0.0008 0.0069 0.0907 -0.0035 -0.0076**

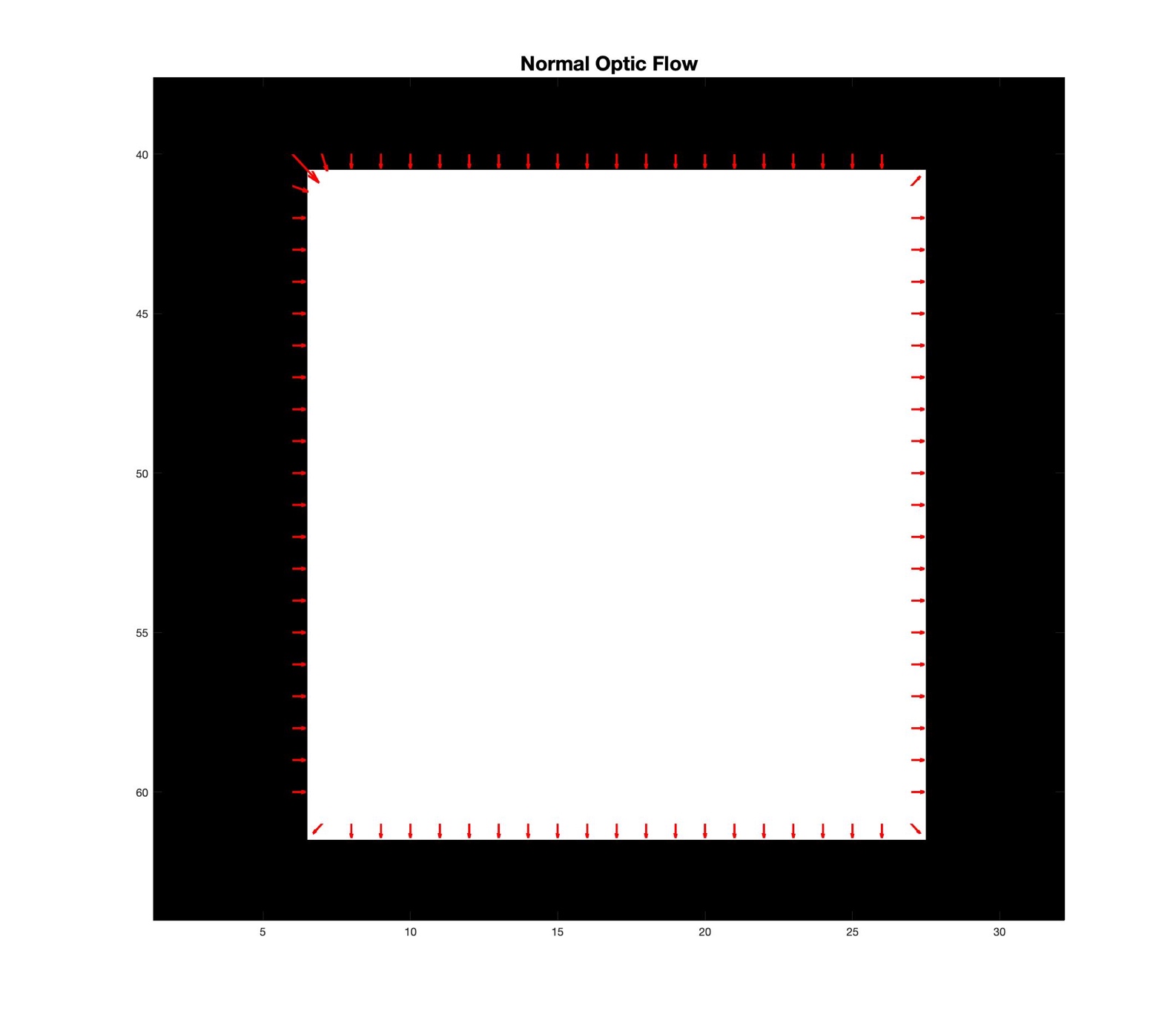
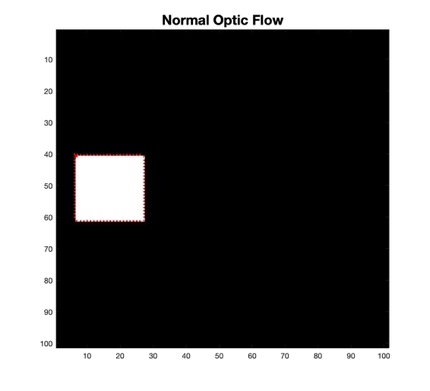
**0.1016 0.0059 0.0098 0.0095 0.1144 -0.0050 -0.0191**

As required, I did not add constraints on delay in MHI so you can see the whole tracks of the arm, from bottom to top. The intensity is faded away from top to bottom because the most recent image has the highest intensity, and the change of intensity indicates the direction to which she moves her arm. The MEI does not provide any information about direction.



1. Normal Optic Flow (origin on left and zoomed on right)

The flow is exactly what I expected. The arrows on x axis are pointing towards bottom, because second picture is moved down by 1 pixel (if we only care about vertical movement). The arrows on y axis are pointing towards right hand side, because second picture is moved towards right by 1 pixel (here we only consider horizontal movement). The left top arrows and right bottom arrows are both pointing towards left bottom side, because these two points on the corner are moved towards right and bottom. However, the point on left bottom side and right top side are moving outwards because the direction is influenced by ft, while the value of left bottom point on ft is positive (so -ft is negative), which directory switched sign of fx and fy (change fx to negative and fy to positive). Thus, the left bottom point is pointing towards left bottom side. Similarly, the fx on right top point is positive originally, and fy is negative. The ft is negative (since it’s 0 on image2 and 1 on image1), which switched the sign of fx and fy as well. The vector is pointing to right top side after sign been switched.



% Author: Yi Zhao

% CSE 5524, HW4

% 09/29/2019

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Problem 1

T = 4;

for i = 2:22

% get image absolute difference

im = double(imread(sprintf('./data/aerobic-%03d.bmp',i)));

im2 = double(imread(sprintf('./data/aerobic-%03d.bmp',i-1)));

dif = abs(im2 - im);

% Threshold, remove tiny regions, dilate, median filter

dif(dif >= T) = 255;

dif(dif < T) = 0;

dif = bwareaopen(dif,150,8);

dif = imdilate(dif, strel('square', 4));

dif = medfilt2(dif);

imwrite(dif, sprintf('./output/Q1/T%d/%d.png',T,i))

% imshow(dif)

% pause;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Problem 2

mei = zeros([320 240]);

mhi = zeros([320 240]);

for i = 2:22

im = imread(sprintf('./output/Q1/T%d/%d.png',T, i));

mei = mei + im;

mhi(im > 0) = i;

end

% normalize mei to binary

mei(mei > 0) = 1;

imagesc(mei)

colormap('gray')

pause;

% normalize mhi between 0 - 1

mhi = max(0, (mhi - 1)/22);

imagesc(mhi)

colormap('gray')

imwrite(mei, './output/MEI.png')

imwrite(mhi, './output/MHI.png')

% compute 7 similityde moments for MEI & MHI

disp(similitudeMoments(mei))

pause;

disp(similitudeMoments(mhi))

pause;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Problem 3

im1 = double(zeros([101 101]));

im2 = double(zeros([101 101]));

box = double(ones([21 21]) \* 255);

im1(40:60, 6:26) = box;

im2(41:61, 7:27) = box;

% sobel filter

hx = [-1 0 1; -2 0 2; -1 0 1]/8;

hy = [-1 -2 -1; 0 0 0;1 2 1]/8;

fx = imfilter(im2, hx);

fy = imfilter(im2, hy);

fxy = sqrt(fx.^2 + fy.^2);

fxy(fxy == 0) = 1; % prevent zero dividing

ft = im2 - im1;

fx = fx./ fxy;

fy = fy./ fxy;

ft = ft./ fxy;

fx = fx.\* ft \* -1;

fy = fy.\* ft \* -1;

xind = repmat(1:size(im2,2),size(im2,1),1); % col => x

yind = repmat((1:size(im2,1))', 1, size(im2,2)); % row => y

imagesc(im2)

colormap('gray')

hold on

quiver(xind,yind,fx, fy, 'color', [1 0 0], 'linewidth', 2)

set(gca,'Ydir','reverse')

title('Normal Optic Flow', 'fontsize', 18)

hold off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Helper function: Calculating Similitude

function Nvals = similitudeMoments(im)

Nvals = [];

% initialize matrix for row index, col index, x average and y average.

xind = repmat(1:size(im,2),size(im,1),1); % col => x

yind = repmat((1:size(im,1))', 1, size(im,2)); % row => y

m00 = sum(im, 'all');

m10 = sum(xind.\*im, 'all');

m01 = sum(yind.\*im, 'all');

xbar = ones(size(im)) \* m10/m00;

ybar = ones(size(im)) \* m01/m00;

% iteratively calculate 7 similitude moments

for i = 0:3

for j = max(0,(2-i)):(3-i)

% 2 <= (i+j) <= 3

nij = sum(((xind - xbar).^i).\*((yind - ybar).^j).\*im, 'all')/(m00.^((i+j)/2+1));

Nvals = [Nvals, nij];

end

end

end