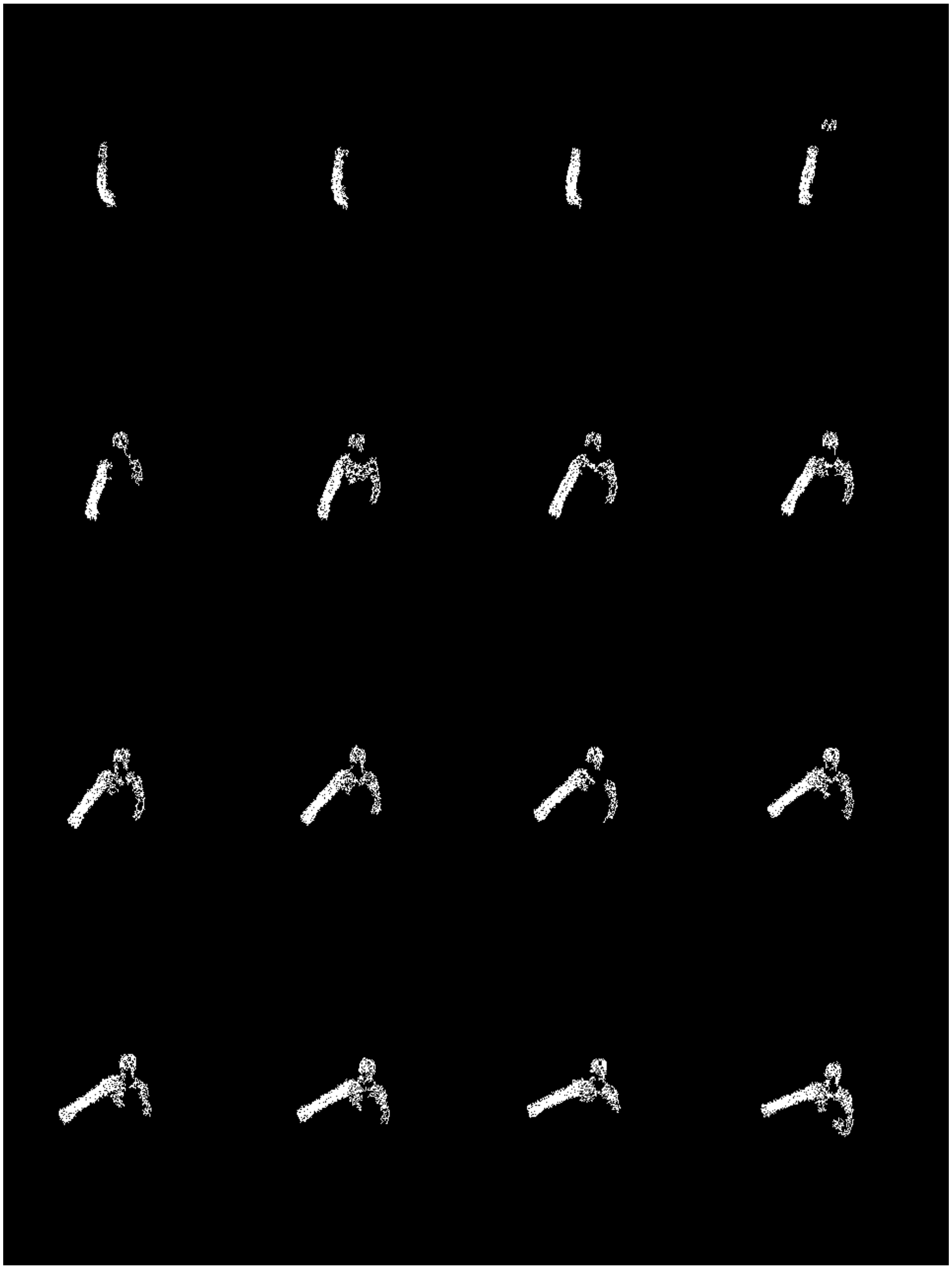
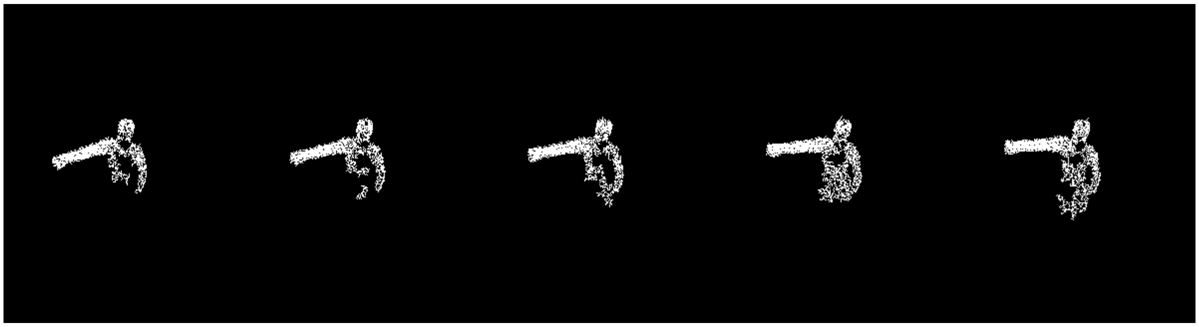
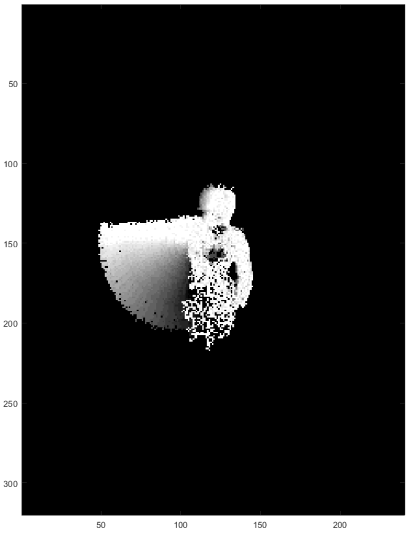
1) Using the images (aerobic-[001-022].bmp) provided on the class materials site, experiment with simple "motion detection" between consecutive frames using (abs) image differencing. Clean-up and remove any tiny regions (e.g., use techniques such as bwareaopen, median filtering, etc.). Experiment with different thresholds. [2 pts]



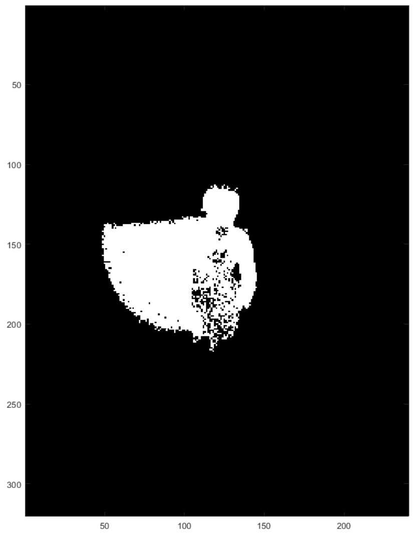


I have tried the T (threshold) from 2 to 10. If the T is too small, there will be a lot of noisy points. But if the T is too big, the images will loss information. So I set the T as 5. And I used bwareaopen to remove the tiny regions.

2) Compute an MEI and MHI on the image sequence (using your best motion differencing approach from problem #1 above for each image pair *i* and *i*-1), simulating the current MHI "timestamp" for each image pair using the larger of the image pair index values (i.e., use *i*, not *i*-1). Therefore, you will have difference images from *i*=2 to 22. The MEI/MHI duration should include all image diff results in the sequence into the final template. Use imagesc (Matlab) to show your results. Compute the 7 similitude moments for the final MEI and the MHI (make sure to *normalize* the MEI and MHI values to be between 0-1 before computing the moments using the given formula in the class notes: max[0, (*i*-1.0)/21.0] for this example). [4 pts]

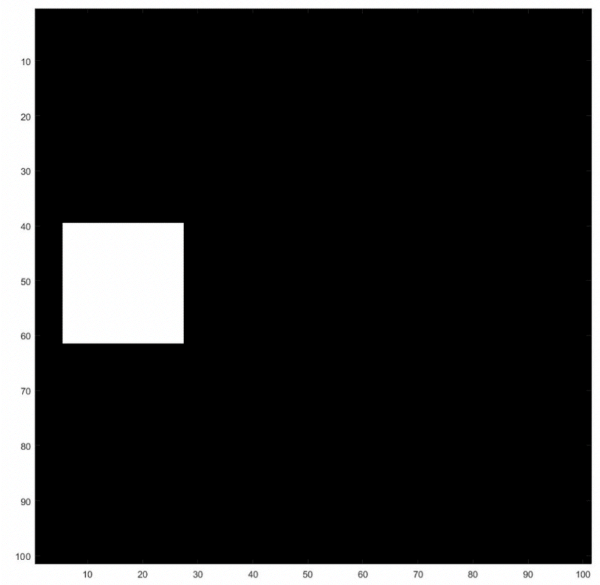


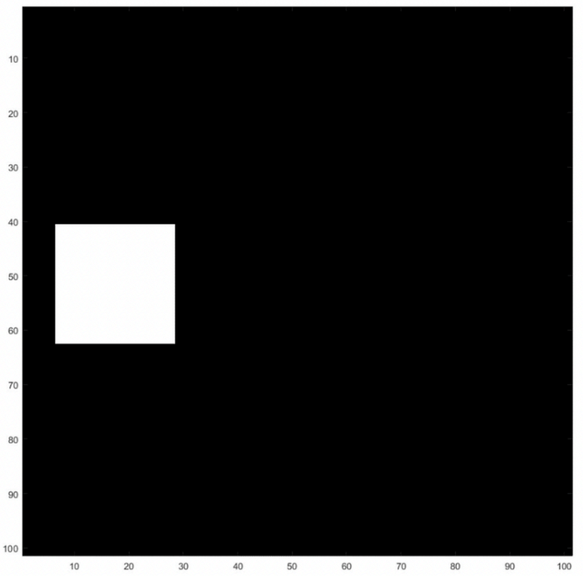
Similitude moments for MHI = 0.1030   0.0080   0.0057   0.0130   0.1611   -0.0011   -0.0192

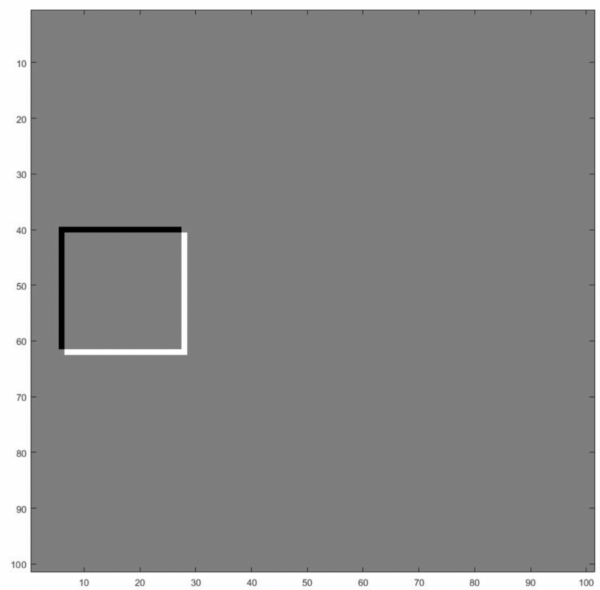


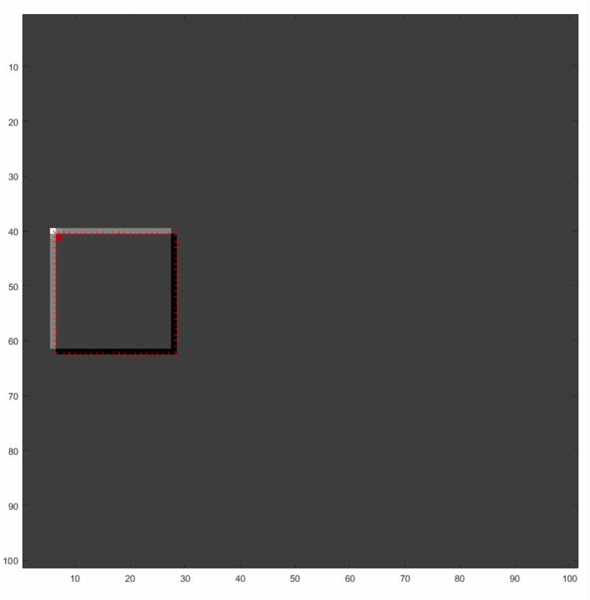
Similitude moments for MEI = 0.0806   -0.0002   -0.0043   0.0072   0.1084   -0.0035   -0.0045

3) Create a 101x101 image with a black (0) background and a white (255) box of size 21x21, placing the upper-left corner at pixel (row=40, col=6). Create another new box image, but shift the box 1-pixel to the right and 1-pixel down. Compute the **normal flow** between the images. Use MATLAB's quiver function to draw the motion vectors on the image (call *imagesc*, then '*hold on*', and lastly call *quiver*). (Make sure your gradient mask orientations/directions and the plot axes are consistent!!!) Make sure all masks are "correct" with proper scaling/ normalization. Is the result what you expected? Why or why not? Comment on the flow for the 4 sides of the box and also for the 4 corners. [5 pts]









Is the result what you expected?： No!

Why or why not? : As I expected, the top and bottom vectors are going down, and the left and right vectors are going right. I thought all four corner would be pointing down to the right, but the lower left vector is pointing down to the left, and the upper right vector is pointing up to the right.

4) As usual, submit all code, images, printouts of images, and report/ discussion of results to Carmen.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Problem 1
for i=1:22
filename = sprintf('aerobic-%03d.bmp', i); % if starts with 001
Ims(:,:,i) = double(imread(filename));
end

T = 5;
for i = 2:22

    Im_diff(:,:,i-1) = abs(Ims(:,:,i)-Ims(:,:,i-1)) >= T;
    Im_diff(:,:,i-1) = bwareaopen(Im_diff(:,:,i-1),50,8);

    imwrite(Im_diff(:,:,i-1), sprintf('diff-%03d.bmp', i-1));
    imagesc(Im_diff(:,:,i-1));
    axis('image');
    colormap('gray');

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Problem 2

%MHI
d = 21;
[r,c] = size(Im_diff(:,:,1));
MHI = zeros(r,c);
for i = 2:22
    temp = Im_diff(:,:,i-1);
    [y,x] = size(temp);
    for row = 1:y
        for column = 1:x
            if temp(row,column) ~= 0
                MHI(row,column) = i;
            elseif MHI(row,column) < i -d
                    MHI(row,column) = 0;
            end
        end
    end


end

%MEI
MEI = max(Im_diff(:,:,1),Im_diff(:,:,2));
for i = 3:21
    MEI = max(MEI, Im_diff(:,:,i));
end
%normailize

for row = 1:r
    for column = 1:c
        MHI(row,column) = max(0, (MHI(row,column) - 1)/21.0);
    end
end
```

```
%similitude moments
NvalsMHI = similitudeMoments(MHI);
NvalsMEI = similitudeMoments(MEI);
disp(NvalsMHI);
disp(NvalsMEI);
    imagesc(MHI);
    axis('image');
    colormap('gray');
    %pause;
    imagesc(MEI);
    axis('image');
    colormap('gray');
    %pause;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Problem 3

whitebox1 = zeros(101,101);
whitebox1(40:61,6:27) = 255;

whitebox2 = zeros(101,101);
whitebox2(41:62,7:28) = 255;

Ft = whitebox2 - whitebox1;

imagesc(whitebox2);
axis('image');
colormap('gray');
%pause;
imagesc(whitebox2);
axis('image');
colormap('gray');
%pause;
imagesc(Ft);
axis('image');
colormap('gray');
%pause;

Fx = -fspecial('sobel')'/8;
Fy = -fspecial('sobel')/8;

Fx = imfilter(whitebox2,Fx);
Fy = imfilter(whitebox2,Fy);

deli = sqrt(Fx.^2 + Fy.^2);
deli (deli == 0) =1;
Fx = Fx./deli;
Fy = Fy./deli;
Ft = (-Ft)./deli;


u = Fx .* Ft;
v = Fy .*Ft;

imagesc(Ft);
axis('image');
colormap('gray');
```

```
hold on
quiver(u,v,2,'r')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Problem 2
%similitude moments

function n_pq =  spatial_central_moments(image, p, q)
    m00 = sum(sum(image));
    [y,x] = size(image);

    m10 = 0;
    m01 = 0;
    for row = 1 :y
        for column=1:x
            m10 = m10 + column * image(row,column);
            m01 = m01 + row * image(row,column);

        end
    end
    x_bar = m10/m00;
    y_bar = m01/m00;

    u_pq = 0;
    for row = 1 :y
        for column=1:x
            u_pq = u_pq + ((column - x_bar)^p) * ((row - y_bar)^q) *
image(row,column);
        end
    n_pq = u_pq / (m00 ^ (1+ ((p+q)/2)));
    end
end
function Nvals =  similitudeMoments(image)

       Nvals = [spatial_central_moments(image,0, 2),
spatial_central_moments(image, 0, 3),...
           spatial_central_moments(image, 1, 1),
spatial_central_moments(image, 1, 2),...
           spatial_central_moments(image, 2, 0),
spatial_central_moments(image, 2, 1),...
           spatial_central_moments(image, 3, 0)];
end
```