EL-GY 6123 Image and Video Processing
Matlab Assignment 6

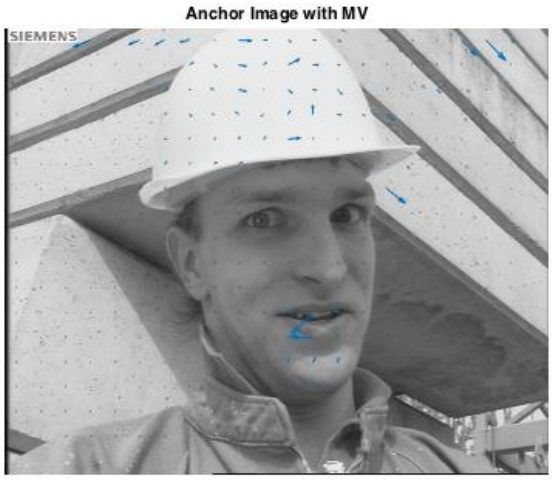
Name: Amitesh Kumar Sah
NYU ID: N19714360

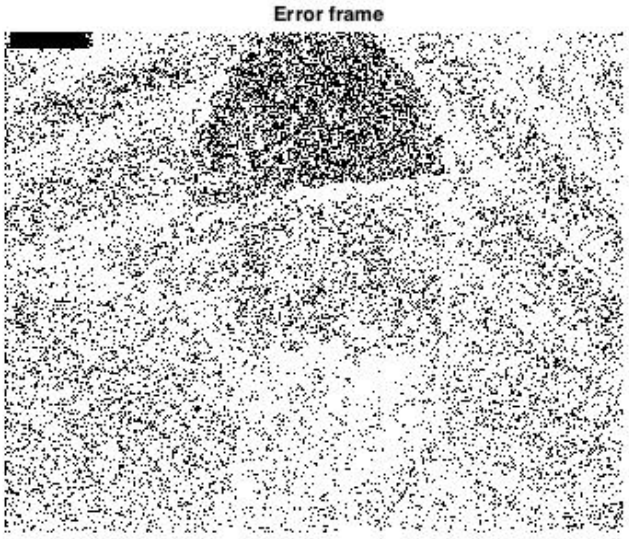
Question 1

```
clc;clear all;close all;

f1 = 'foreman103.Y'; % anchor frame
f2 = 'foreman100.Y'; % target frame
R = 32; % search range
W = 352; % frame width
H = 288; % frame height
N = 16; % block size
fp = zeros(H, W); % predicted frame
mvx = zeros(H/N, W/N);
mvy = zeros(H/N, W/N);
[X, Y] = meshgrid(N/2:N:W, N/2:N:H);
X=X(1:end-1,1:end-1);
Y=Y(1:end-1,1:end-1);
% read frames
fid1 = fopen(f1, 'r');
fid2 = fopen(f2, 'r');
f1 = fread(fid1, [W, H], 'uint8=>double')';
f2 = fread(fid2, [W, H], 'uint8=>double')';
fclose(fid1);
fclose(fid2);
figure,
subplot(1,2,1),imshow(f1,[]),title('Anchor Image');
subplot(1,2,2),imshow(f2,[]),title('Target Image');

%f1: anchor frame; f2: target frame, fp: predicted image;
%mvx,mvy: store the MV image
%widthxheight: image size; N: block size, R: search range
mvx=0;mvy=0;
for i=1:N:H-N,
 for j=1:N:W-N %for every block in the anchor frame
 MAD_min=256*N*N;


 for k=-R:1:R
    if i+k < 1 || i+k+N-1 > H % check vertical boundary
        continue;
    end

for l=-R:1:R
    if j+l < 1 || j+l+N-1 > W % check horizontal boundary
        continue;
    end

MAD=sum(sum(abs(f1(i:i+N-1,j:j+N-1)-f2(i+k:i+k+N-1,j+l:j+l+N-1))));
% calculate MAD for this candidate
    if MAD<MAD_min
        MAD_min=MAD;
        dy=k;
        dx=l;
    end;
 end;
 end;

 %put the best matching block in the predicted image
 fp(i:i+N-1,j:j+N-1)= f2(i+dy:i+dy+N-1,j+dx:j+dx+N-1);
```

```
 iblk=floor((i-1)/N+1);
 jblk=floor((j-1)/N+1); %block index
 mvx(iblk,jblk)=dx;
 mvy(iblk,jblk)=dy; %record the estimated MV
%  arrow([i j],[i+dy j+dx], 3);
end;
end;

 figure,imshow(f1,[]),title('Anchor Image with MV');
 hold on
 quiver(X,Y,mvx,mvy);
hold off

figure,imshow(fp,[]),title('Predeicted frame');

fd = f2-f1; % the direct difference between f1 and f2
% calculate the error frame bysubtracting the predicted frame from the
% anchor frame
errorframe=imabsdiff(f1,fp);
figure,imshow(errorframe),title('Error frame');

ferr = fp-f1; % the motion compensation error image
% compute the variances
var_f1 = mean(mean((f1-mean(mean(f1))).^2))
var_fd = mean(mean((fd-mean(mean(fd))).^2))
var_fe = mean(mean((ferr-mean(mean(ferr))).^2))

% Calculate PSNR
PSNR_full=10*log10(255*255/mean(mean(errorframe.^2)))
```

Result





var_f1 = 2.9781e+03
var_fd = 277.9028
var_fe = 2.3070e+03
PSNR_full = 14.1040

Question 2

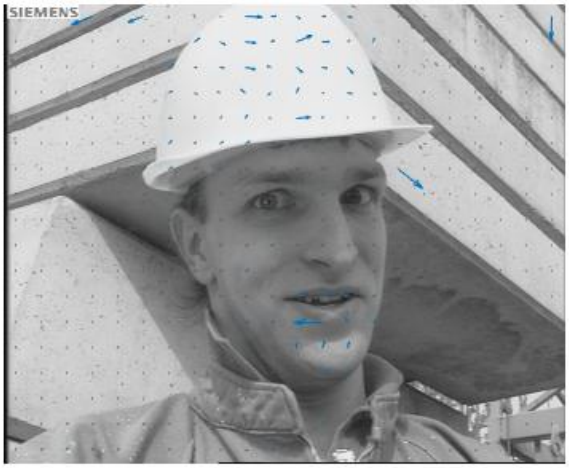
```
close all,clear all,clc
f1 = 'foreman103.Y'; % anchor frame
f2 = 'foreman100.Y'; % target frame
R = 32; % search range
W = 352; % frame width
H = 288; % frame height
N = 16; % block size
fp = zeros(H, W); % predicted frame
mvx = zeros(H/N, W/N);
mvy = zeros(H/N, W/N);
[X, Y] = meshgrid(N/2:N:W, N/2:N:H);
% read frames
fid1 = fopen(f1, 'r');
fid2 = fopen(f2, 'r');
f1 = fread(fid1, [W, H], 'uint8=>double')';
f2 = fread(fid2, [W, H], 'uint8=>double')'; fclose(fid1);
fclose(fid2);
fd = f2-f1; % the direct difference between f1 and f2
figure,
```

```
subplot(1,2,1),imshow(f1,[]),title('Anchor Image');
subplot(1,2,2),imshow(f2,[]),title('Target Image');
% EBMA with half-pel accuracy
%mvx,mvy: store the MV image
%first upsample f2 by a factor of 2 in each direction
f3=imresize(f2, 2,'bilinear');
mvx=0;mvy=0;
for i = 1:N:H
for j = 1:N:W
MAD_min = 255*N*N;
    for k = -R:0.5:R
        if i+k < 1 || i+k+N-1 > H % check vertical boundary
            continue;
        end
    for l = -R:0.5:R
         if j+l < 1 || j+l+N-1 > W % check horizontal boundary
            continue;
         end
MAD=sum(sum(abs(f1(i:i+N-1,j:j+N-1)-f3(2*(i+k):2:2*(i+k+N-1),2*(j+l):2:2*(j+l+N-1)))));
        if MAD < MAD_min
            MAD_min = MAD;
            dy = k;
            dx = l;
        end
    end
    end
fp(i:i+N-1, j:j+N-1) = f3(2*(i+dy):2:2*(i+dy+N-1),2*(j+dx):2:2*(j+dx+N-1));
iblk = (i-1)/N+1;
jblk = (j-1)/N+1;
mvx(iblk, jblk) = dx;
mvy(iblk, jblk) = dy;
end
end
ferr = fp-f1; % the motion compensation error image

figure,imshow(f1,[]),title('Anchor Image with MV');
    hold on
 quiver(X,Y,mvx,mvy);
    hold off

figure,imshow(fp,[]),title('Predeicted frame');

fd = f2-f1; % the direct difference between f1 and f2
% calculate the error frame bysubtracting the predicted frame from the
% anchor frame
errorframe=imabsdiff(f1,fp);
figure,imshow(errorframe),title('Error frame');

ferr = fp-f1; % the motion compensation error image
% compute the variances
var_f1 = mean(mean((f1-mean(mean(f1))).^2))
var_fd = mean(mean((fd-mean(mean(fd))).^2))
var_fe = mean(mean((ferr-mean(mean(ferr))).^2))

% Calculate PSNR
PSNR_full=10*log10(255*255/mean(mean(errorframe.^2)))
```

Anchor Image

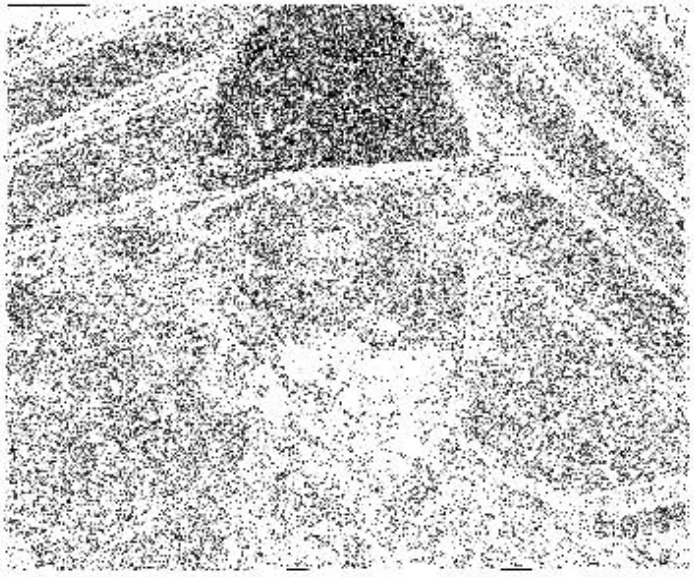
Target Image

Anchor Image with MV



Predeicted frame

Error frame



var_f1 =2.9781e+03
var_fd = 277.9028
var_fe = 39.6733
PSNR_full = 32.1200
The method of half-pel EBMA gives me more accurate prediction, however, it requires more computation time.

Question 3

```
fid1=fopen('foreman100.Y');
B1 = fread(fid1, [352, 288], 'uint8=>double')';

fid2=fopen('foreman103.Y');
B2 = fread(fid2, [352, 288], 'uint8=>double')';

N=16;
R=8;
[height, width]=size(B1);

fi1=imresize(B1,0.5,'bilinear');
fi2=imresize(B2,0.5,'bilinear');

[row,col]=size(fi1);

f1=zeros(row+2*R,col+2*R);
f2=f1;
f1(R+1:R+row,R+1:R+col)=fi1;
f2(R+1:R+row,R+1:R+col)=fi2;
fp=zeros(size(f1));
    mvx=0;
        mvy=0;
for i=1+R:N:row+R-N
    for j=1+R:N:col+R-N
        MAD_min=256*N*N;

        for k=-R:R
            for l=-R:R
                MAD=sum(sum(abs(f1(i:i+N-1,j:j+N-1)-f2(i+k:i+k+N-1,j+l:j+l+N-1))));
                if MAD<MAD_min
                    MAD_min=MAD;
                    dy=k;
                    dx=l;
                end
            end
        end
        fp(i:i+N-1,j:j+N-1)=f2(i+dy:i+dy+N-1,j+dx:j+dx+N-1);
        iblk=floor((i-1-R)/N)+1;
        jblk=floor((j-1-R)/N)+1;
        mvx(iblk,jblk)=dx;
        mvy(iblk,jblk)=dy;
    end
end

fps=fp(9:152,9:184);
fi11=imresize(fps,2,'bilinear');

f11=zeros(height+2*R,width+2*R);
f22=f11;
f11(R+1:R+height,R+1:R+width)=fi11;
f22(R+1:R+height,R+1:R+width)=B2;
fpp=zeros(size(f11));
    mvx2=0;
        mvy2=0;
for i=1+R:N:height+R-N
    for j=1+R:N:width+R-N
        MAD_min2=256*N*N;

        for k=-R:R
            for l=-R:R
                MAD2=sum(sum(abs(f11(i:i+N-1,j:j+N-1)-f22(i+k:i+k+N-1,j+l:j+l+N-1))));
                if MAD<MAD_min2
```

```
                    MAD_min2=MAD;
                    dy2=k;
                    dx2=l;
                end
            end
        end
        fpp(i:i+N-1,j:j+N-1)=f22(i+dy2:i+dy2+N-1,j+dx2:j+dx2+N-1);
        iblk2=floor((i-1-R)/N)+1;
        jblk2=floor((j-1-R)/N)+1;
        mvx2(iblk2,jblk2)=dx2;
        mvy2(iblk2,jblk2)=dy2;
    end
end

MSE=mean2((f22-fpp).^2);
PSNR=10*log10(255^2/MSE)

figure,subplot(2,2,1),imshow(f11,[]),title('Anchor');
subplot(2,2,2),imshow(f22,[]),title('Target');
subplot(2,2,3),quiver(mvx2,mvy2),title('Motion Vector');
subplot(2,2,4),imshow(fpp,[]),title('Predicted image');
```

Result:
PSNR =

11.4006