Project Progress IX

Archana Kumari

Under the Guidance of
Dr. Pulak Mondal

28 March 2016

As Full search (FS) and Logarithmic search (LS) were implemented in MATLAB with macroblock size 8x8 and search area 24x24 for 10 consecutive frames of the video sequence of "Foreman", to get 10 residual frames after proper motion compensation while forming the predicted frames, so after the analysis of results given by these motion estimation algorithm in performance, it was observed that FS is far better in predicting the frames but brings a lot of computations into picture, whereas LS was found to be less computational but less accurate. So to exploit the advantages of both the search algorithms, a new algorithm for motion estimation is proposed here which is better in prediction than LS and less computational than FS.
It is proposed that if some frames in between of the sequence is searched by FS and the rest by LS then the overall performance for a video will surely increase.
Here this method is implemented in MATLAB such that after every three consecutive video frames searched by LS, the fourth one is searched by FS.
The following MATLAB code was implemented:

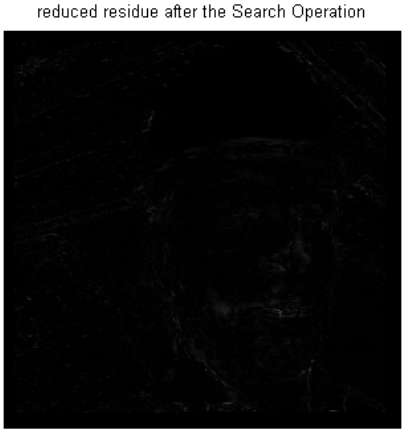
```
clc
clear all
close all
tstart=cputime;
f_ref(1:300,1:300)=0;
Im=imread('E:\foreman_10frames\f001.pgm');
f_ref(9:288,9:288)= Im(9:288,9:288);

srcFiles = dir('E:\foreman_10frames\*.pgm');

f_p=zeros(300,3000);

X=zeros(35,350);
Y=zeros(35,350);
fg=1;
change=0;
MSE=zeros(1,9);

for frameNo=1:9
    filename = strcat('E:\foreman_10frames\',srcFiles(frameNo+1).name);
    f_2(1:300,1:300)=0;
    Im2= imread(filename);
    f_2(9:288,9:288)= Im2(9:288,9:288);

    X_motion=zeros(35);
    Y_motion=zeros(35);
```

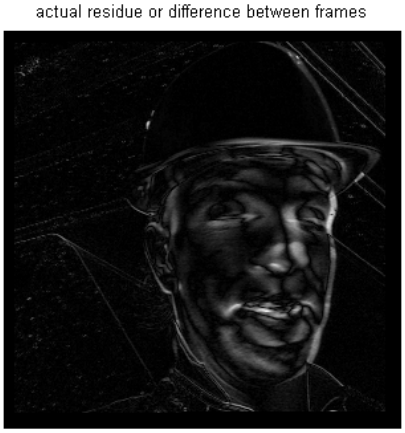
```
f_pre(1:300,1:300)=0;

if fg==1
    s=1;
    for i=9:8:288
        t=1;
        for j=9:8:288

            img_abs=zeros(8,8);
            img_24=f_ref(i-8:i+7+8,j-8:j+7+8);
            img_8=f_2(i:i+7,j:j+7);

            for p=1:17
                for q=1:17

                    img_abs(p,q)=sum(sum((img_24(p:p+7,q:q+7)- img_8).^2));
                end
            end

            [M,I] = min(img_abs(:));
            [row_cor, col_cor] = ind2sub(size(img_abs),I);


            f_pre(i:i+7,j:j+7)=img_24(row_cor:row_cor+7,col_cor:col_cor+7);

            X_motion(s,t)= row_cor -9;
            Y_motion(s,t)= col_cor -9;
            t=t+1;
        end
        s=s+1;
    end

elseif fg==0
    s=1;
    X_motion= zeros(22,22);
    Y_motion=zeros(22,22);
    for i=9:8:288
        t=1;
        for j=9:8:288
            img_abs=[0 0 0 0 0];
            img_24=f_ref(i-8:i+7+8,j-8:j+7+8);
            img_8=f_2(i:i+7,j:j+7);
            flag = 1;
            I=1;
            Rc=9;
            Cc=9;
            step_size=4;

            while flag
                r=[Rc,Rc-step_size,Rc,Rc,Rc+step_size];
                c=[Cc,Cc,Cc-step_size,Cc+step_size,Cc];

                for g=1:5
```

```
                    if r(g)<=0 || r(g)>=17
                        img_abs(g)=255*64*255;
                    elseif c(g)<=0 || c(g)>=17
                        img_abs(g)=255*64*255;
                    elseif g==I && g-1>0
                        img_abs(g)=255*64*255;
                    else
                        img_abs(g)=sum(sum((img_24(r(g):r(g)+7,c(g):c(g)+7)- img_8).^2));
                    end
                end


                [M,I] = min(img_abs);
                switch (I)
                    case 1
                        step_size=step_size/2;
                    case 2
                        Rc=Rc-step_size;
                    case 3
                        Cc=Cc-step_size;
                    case 4
                        Cc=Cc+step_size;
                    case 5
                        Rc=Rc+step_size;
                end
                if step_size<1
                    flag=0;
                    break;
                else
                    continue;
                end
            end
            f_pre(i:i+7,j:j+7)=img_24(Rc:Rc+7,Cc:Cc+7);
            X_motion(s,t)= Rc-9;
            Y_motion(s,t)= Cc-9;

            t=t+1;
        end
        s=s+1;
    end

end

change=change+1;
if change<3
    fg=0;
elseif change==3
    change=0;
    fg=1;
end

f_p(1:300, 1+(300*frameNo):300*(frameNo+1))=f_pre;
X(1:35, 1+(35*(frameNo-1)):35*frameNo)=X_motion;
Y(1:35, 1+(35*(frameNo-1)):35*frameNo)=Y_motion;
```

```
    residu1=abs(f_2-f_pre);
    MSE(frameNo)=(sum(sum((residu1).^2)))/90000;
    figure,imshow(uint8(residu1));
    title('reduced residue after the Search Operation');
    figure,imshowpair(f_2,f_ref,'diff');
    title('actual residue or difference between frames');
    f_ref=f_2;
end
telapsed=cputime-tstart;
Frame=[1 2 3 4 5 6 7 8 9];
figure,FullLogSearch=plot(Frame,MSE);
title('Mean Square Error [MSE] Vs Frames Plot');
ylabel('MSE found in Proposed Search');
xlabel('Frame number');
display('time elapsed in search');
display(telapsed);
display(MSE);
```

```
time elapsed in search

telapsed =

   30.3750


MSE =

  Columns 1 through 7

   16.1771   18.5147   21.4357   14.1686   15.1979   20.9555   13.9718

  Columns 8 through 9

   24.6960   25.6278
```

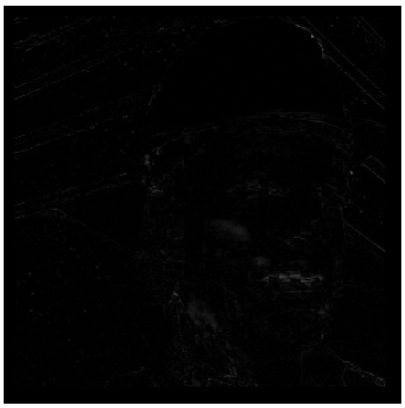




reduced residue after the Search Operation

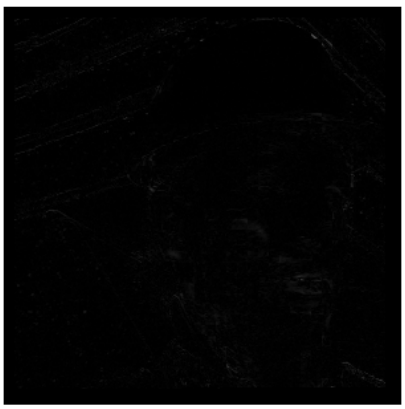
actual residue or difference between frames

reduced residue after the Search Operation



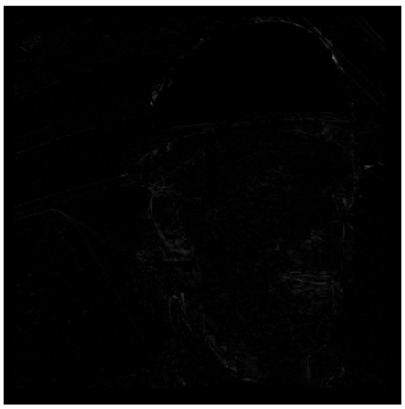
actual residue or difference between frames

reduced residue after the Search Operation



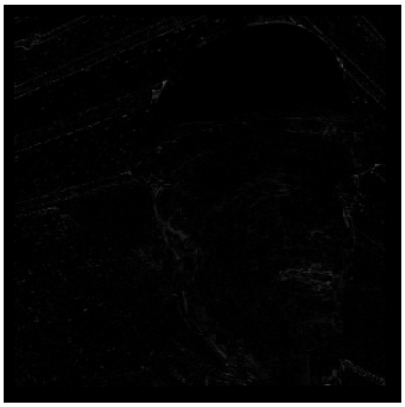
actual residue or difference between frames

reduced residue after the Search Operation



actual residue or difference between frames

reduced residue after the Search Operation

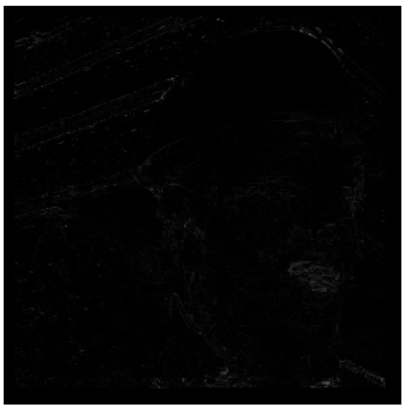


actual residue or difference between frames

reduced residue after the Search Operation

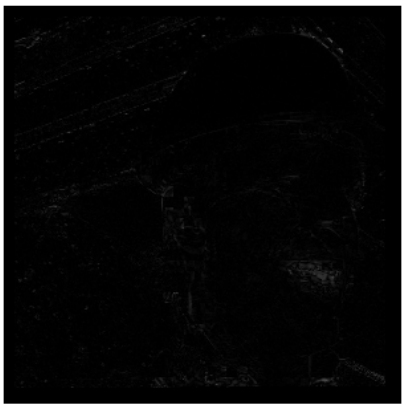
actual residue or difference between frames

reduced residue after the Search Operation



actual residue or difference between frames

reduced residue after the Search Operation



actual residue or difference between frames