

实验一 序列图像的运动估计实验

一、实验目的和任务

学习在 MATLAB 编程环境下（或用其他编程语言）实现对标准序列图像的多帧图像间进行运动估计，并统计得到的运动矢量。

二、实验原理

运动估计的基本思想是将图像序列的每一帧分成许多互不重叠的宏块，并认为宏块内所有像素的位移量都相同，然后对每个宏块到参考帧某一特定搜索范围内，根据一定的匹配准则找出与当前块最相似的块，即匹配块（预测块），匹配块与当前块的相对位移即为运动矢量。预测块与当前块逐像素的差值组成残差块。

利用 MATLAB 图像处理工具箱中的函数，在 MATLAB 编程环境下，用穷尽块匹配算法和一种快速搜索算法实现序列图像的帧间运动估计。

三、实验内容和步骤

1. 将标准序列图像的前后多帧图像文件存入计算机，运行 MATLAB 集成开发环境。
2. 编程实现序列图像的穷尽块匹配算法和一种快速搜索算法，得到不同运动估计方法的

运动矢量图和残差帧图像，比较两种方法的计算复杂度和 PSNR。

3. 撰写实验报告。

【实验内容】

此次采用穷尽匹配算法和三步快速搜索算法实现序列图像的帧间运动估计，两种方法分别得到残差帧图像和运动矢量图，对两种算法进行对比并得出结论。

I. 实验算法原理

基于块匹配的运动估计算法依据一定的匹配准则，通过在两个帧之间的像素域搜索程序找到最佳的运动矢量估算。块匹配的算法在实现时，在最佳匹配原则、匹配块搜索过程以及选择块的大小等方面有差异。但最佳匹配准则在不同运动估计方法中的差异并不是必须的，即不同的运动估计方法可以使用相同的最佳匹配准则。因此，匹配块的搜索过程，即寻找最佳匹配块的搜索路径是区分不同于运动估计方法的主要依据。

穷尽搜索算法又称全搜索算法(FS)，是对搜索窗内所有像素点计算 MAD 值，从中找出 MAD 值最小的点，其对应的偏移量即为所求的运动矢量。

三步快速搜索算法，它采用一种由粗到细的搜索模式，从原点开始，按一定步长取周围 8 个点构成每次搜索的点群，然后进行匹配计算，跟踪 MBD 点。

II. 穷尽搜索算法(FS)

源代码:


```

                                if Eij<A
                                    A=Eij;
                                    xrecord(x+1,y+1)=p;
                                    yrecord(x+1,y+1)=q;
                                end
                            end
                        end
                    end
                A=999999999999999999;
                for mx=1:blocksize
                    for ny=1:blocksize
diff(row+mx,col+ny)=I2(row+mx,col+ny)-I1(row+mx+dm+xrecord(x+1,y+1),col+ny+dm+yrecord(x+1,y+1));
                    end
                end
            end
        end
    end
    toc

    figure,
    imshow(I1,[]);
    title('The first frame');

    figure,
    imshow(I2,[]);
    title('The second frame');
    III=I2-I1;

    figure,
    imshow(III,[]);
    title('残差帧图像');

    figure;
    quiver(1:16,1:16,yrecord,xrecord);
    grid on;
    title('运动矢量图');

```

FS 算法结果:

The first frame



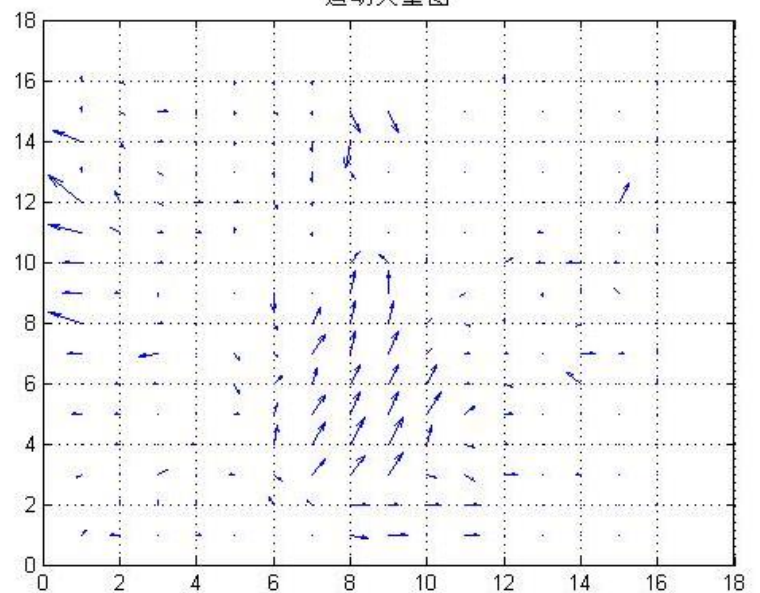
The second frame



残差帧图像



运动矢量图



III. 三步快速搜索算法(TSS)

源代码:

```
clear all;  
  
I1=imread('claire1.bmp'); %read the first frame  
I2=imread('claire2.bmp'); %read the second frame  
  
dm=7;  
  
%给图像扩边, 每个边都扩 dm 大小  
  
I1=double(I1);  
I2=double(I2);  
  
[rownum colnum] = size(I1);
```



```

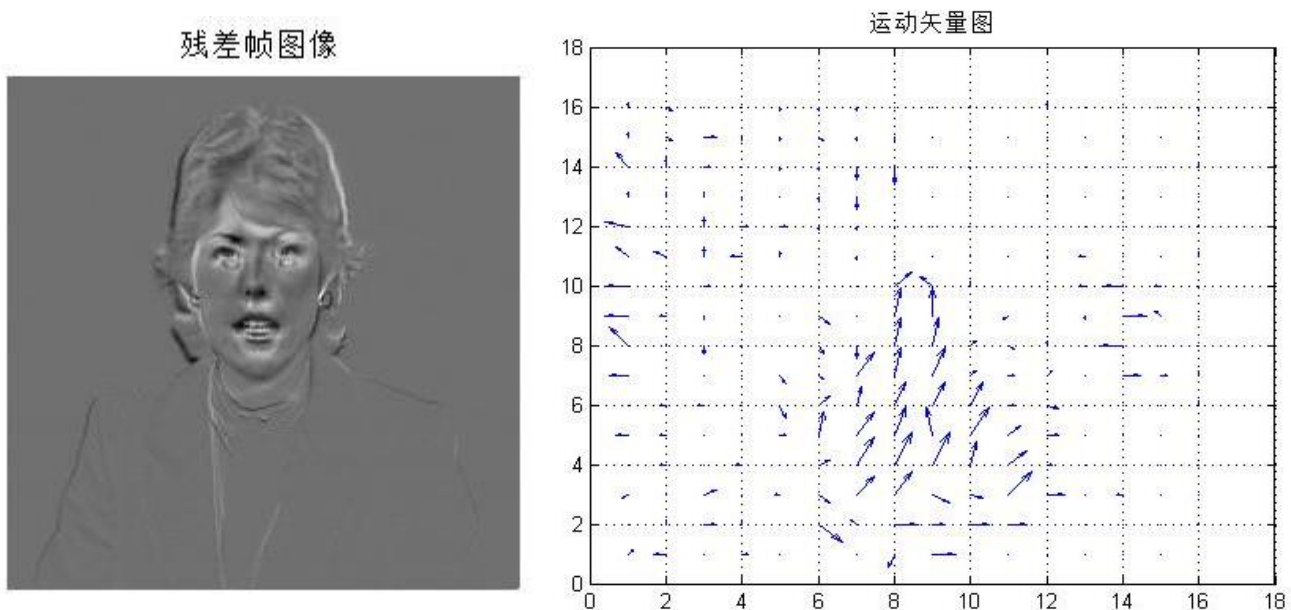
p1=xrecord(x+1,y+1);
q1=yrecord(x+1,y+1);
for p2=p1-2:2:p1+2          % 第二步
    for q2=q1-2:2:q1+2
        if p2~=p1 | q2~=q1
            Eij=0;
Eij=sum(sum((I2(row+1:row+blocksize,col+1:col+blocksize)-II(row+dm+p2+1:row+dm+p2+blocksize,col+dm
+q2+1:col+dm+q2+blocksize)).^2))/(blocksize^2);
            if Eij<A
                A=Eij;
                xrecord(x+1,y+1)=p2;
                yrecord(x+1,y+1)=q2;
            end
        end
    end
end
end

p2=xrecord(x+1,y+1);
q2=yrecord(x+1,y+1);
for p3=p2-1:1:p2+1          % 第三步
    for q3=q2-1:1:q2+1
        if p3~=p2 | q3~=q2
            Eij=0;
Eij=sum(sum((I2(row+1:row+blocksize,col+1:col+blocksize)-II(row+dm+p3+1:row+dm+p3+blocksize,col
+dm+q3+1:col+dm+q3+blocksize)).^2))/(blocksize^2);
            if Eij<A
                A=Eij;
                xrecord(x+1,y+1)=p3;
                yrecord(x+1,y+1)=q3;
            end
        end
    end
end
end

A=999999999999999999999999;

for mx=1:blocksize
    for ny=1:blocksize

```

VI.实验结论:

通过对比实验发现,运动矢量反映了参考帧相对于当前帧的运动趋势,残差帧反映了预测帧与当前帧之间的差值,它们都在主观和是哪个反映了不同算法的效果。其中,穷尽块匹配算法(FS)得到的运动矢量图反映的运动趋势最为准确,残差帧的值也是最小的,因为它穷尽搜索窗口内的每一个点,三步快速搜索算法(TSS)次之。

穷尽块匹配算法(FS)的平均 PSNR 最高,即 FS 算法进行运动估计搜索精度最高,但是 FS 算法的块平均搜索点数却是最多的,因此它需要消耗大量的计算量,不适合实时压缩应用。相比 FS 算法,三步快速搜索算法(TSS)的平均 PSNR 是最低的,匹配效果最差,这是由于 TSS 第一步搜索步长过大,容易导致陷入局部极小点,使搜索精度下降。

四、参考文献

- [1] 郑运昌,张连连,张红岭,张克辉,黄晓英,张静静,王磊.基于块匹配的视频帧间运动估计算法研究[J].科学技术创新,2019(11):57-58.
- [2] 但鸿键,汪伟.基于高效视频编码的运动估计算法综述[J].计算机时代,2020(01):1-4+9.
- [3] 罗军辉. MATLAB7.0 在图像处理中的应用[M].北京:机械出版社.2005.

五、实验心得

通过此次实验用穷尽匹配算法和三步快速搜索算法实现序列图像的帧间运动估计,更加深刻的理解视频信息的获取、搜索和保存的基本方法。基于块的运动估计,是先将视频图像分成一个个规则的图像块,然后对每个图像块估计运动矢量,基于块的运动估计和运动补偿已经广泛应用于各种视频压缩编码标准中。