



哈尔滨工业大学

Harbin Institute of Technology

运动目标的检测

课程名称: 新概念成像制导信息处理系统

院 系: 计 算 机 学 院

班 级: 1303101

设 计 者: 杨 尚 斌

学 号: 1130310128

设计时间: 2014. 11. 28

一、实验目的

利用 MATLAB 软件对视频进行处理，检测场景中的运动目标。利用统计的方法得到背景模型，并实时更新以适应场景的变化，检测当前图像和背景图像中对应像素点的差异，如果差值大于一定阈值，则判定该像素为前景运动目标。然后对图像进行后处理，去除噪声和背景中运动物体的影响。

二、实验内容

运动目标检测是指在序列图像中检测出变化区域并将运动目标从背景中提取出来。但由于后期干涉（天气、光照等）的干扰，使得运动目标的检测变得困难。常用的方法有帧差法、背景减弱法和光流法。

1、运动目标检测方法

①帧差法：属于运动目标检测最常用的方法之一，基本原理就是在图像序列相邻两帧或三帧间采用基于像素的时间差分通过闭值化来提取图像中的运动区域，利用后一帧与前一帧的差值来计算是否存在运动目标。后一帧的灰度值减去前一帧的灰度值，取绝对值，当有运动存在时下一帧与前一帧肯定会有亮度的变化，当这种变化大于一定值时就认为是运动。不过这种做法对环境噪声叫为敏感，差值的选择相当关键，选择过低不足以抑制图像中的噪声，过高则忽略了图像中有用的变化。如此，对于比较大的、颜色一致的运动目标，有可能在目标内部产生空洞，无法完整地提取运动目标。

②背景减除法：利用一些参数得到背景模型来近似背景图像的像素值，将当前帧与背景图像进行差分比较实现对运动区域的检测，其中区别较大的像素区域被认为是运动区域，而区别较小的像素区域被认为是背景区域。但是为得到背景模型需要对大量帧进行操作，占内存较多；而且对于背景扰动较大时，应该也并不理想。

③光流法：利用光流方程计算出每个像素点的运动状态矢量，从而发现运动的像素点，并且能够对这些像素点进行跟踪。（被观察物体表面带光学特征的部位的移动给人们提供了运动和结构的信息。当相机与场景目标间有相对运动时所观察到的亮度模式运动称之为光流）该方法是对每一个像素点进行操作，对比他们的位置，得到其中运动的像素点，然后对运动像素点进行跟踪它能同时完成运动

目标检测和跟踪,但是计算复杂度高这样就影响了他的实时性,同时,光流场的计算非常容易受到噪声、光照变化和背景扰动的影响。

2、运用帧差法实现运动目标的检测：

- 1) 本此实验是利用帧差法对视频进行处理，得到运动的目标并对其进行跟踪操作。在操作过程中取绝对值。

$$D=|\text{Imag}(x,y,k+1)-\text{Imag}(x,y,k)|$$

2) 实验结果



图 1 实验视频

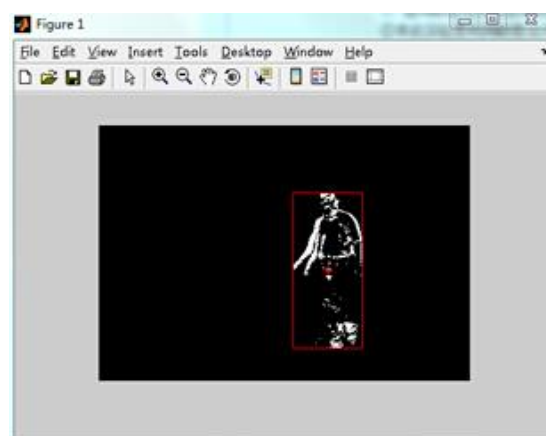


图 2 实验处理结果

- 3) 结果分析及实验结论：在图 2 中我们可以看到手臂处有明显的空洞，且轮廓不太理想，这是帧差法的缺点。



图 3 轮廓放大展示

三、设计感想

运动目标的检测常用方法有三种，即帧差法、背景减除法和光流法。每种方法否有其优点和缺点，我们选用的是第一次帧差分，其中差值的选定算是一个难点，也正是我们这份代码中存在的问题。

后一步需要优化的就是差值的选取，确保适应不同的环境。

四 、源代码

```
clear data
disp('input video');
avi = aviread('samplevideo.avi');
video = {avi.cdata};
for a = 1:length(video)
    imagesc(video{a});
    axis image off
    drawnow;
end;
disp('output video');
tracking(video);

function d = tracking(video)
if ischar(video)
    % Load the video from an avi file.
    avi = aviread(video);
    pixels = double(cat(4,avi(1:2:end).cdata))/255;
    clear avi
else
    % Compile the pixel data into a single array
    pixels = double(cat(4,video{1:2:end}))/255;
    clear video
end

% Convert to RGB to GRAY SCALE image.
```

```

nFrames = size(pixels,4);
for f = 1:nFrames

%      F = getframe(gcf);
%      [x,map]=frame2im(F);
%      imwrite(x,'fln.jpg','jpg');
% end
    pixel(:,:,f) = (rgb2gray(pixels(:,:,f)));
end
rows=240;
cols=320;
nframes=f;
for l = 2:nframes
d(:,:,l)=(abs(pixel(:,:,l)-pixel(:,:,l-1)));

k=d(:,:,l);
% imagesc(k);
% drawnow;
% himage = imshow('d(:,:,l)');
% hfigure = figure;
% impixelregionpanel(hfigure, himage);

% datar=imageinfo(imagesc(d(:,:,l)));
% disp(datar);

    bw(:,:,l) = im2bw(k, .2);

    bw1=bwlabel(bw(:,:,l));
    imshow(bw(:,:,l))
    hold on

% %      for h=1:rows
% %      for w=1:cols
% %
% %          if(d(:,:,l)< 0.1)
% %              d(h,w,l)=0;
% %          end
% %      end
% %
% % end

% % disp(d(:,:,l));
% % size(d(:,:,l))

```

```

cou=1;
for h=1:rows
    for w=1:cols
        if(bw(h,w,l)>0.5)

%            disp(d(h,w,l));
            toplen = h;

                if (cou == 1)
                    tpln=toplen;

                end
                cou=cou+1;
            break
        end

    end
end

disp(toplen);

coun=1;
for w=1:cols
    for h=1:rows
        if(bw(h,w,l)>0.5)

            leftsi = w;

                if (coun == 1)
                    lftln=leftsi;
                    coun=coun+1;

                end
                break
            end

        end
    end

    disp(leftsi);
    disp(lftln);

% % drawnow;

```

```

% %      d = abs(pixel(:, :, l), pixel(:, :, l-1));
% %      disp(d);

%      s = regionprops(bw1, 'BoundingBox');
% %      centroids = cat(1, s.Centroid);
%
% %      ang=s.Orientation;
%
% %      plot(centroids(:,1), centroids(:,2), 'r*')
%      for r = 1 : length(s)
%          rectangle('Position',s(r).BoundingBox,'EdgeColor','r');
%
% %      plot('position',s(r).BoundingBox,'faceregion','r');
%      end
%

% %      disp(ang);
% %      imaqmontage(k);

width=leftsi-lftln;
heig=toplen-tpln;

widt=width/2;
disp(widt);
heit=heig/2;
with=lftln+widt;
heth=tplen+heit;
wth(1)=with;
hth(1)=heth;

disp(heit);
disp(width);
disp(heig);
rectangle('Position',[lftln tpln width heig],'EdgeColor','r');
disp(with);
disp(heth);
plot(with,heth, 'r*');
drawnow;
hold off

end;
% wh=square(abs(wth(2)-wth(nnames)));
% ht=square(abs(hth(2)-hth(nnames)));
% disp(wth(1

```

```
% distan=sqrt(wh+ht);  
%  
% disp(distan);
```