Assignment 1

Activity Analytics

Li Pei, AndrewID: lip
Zheng Lei, AndrewID: zlei

1. Extraction of Moving People in Video
2. Motion Energy Image
3. Motion History Image
4. Heat Map
5. Find the static girl

Extraction of Moving People in Video

The key to solve this problem is the same as the hint, removing the background. We use the method based on the "Approximate Median algorithm"(*referenced from page 150 in Ambient Diagnostics, Yang Cai*). Described as the pseudo code below,

```
read first frame
median_frame = first frame % assume the first frame is the median
while (frames remaining)
        current_frame = read next_frame
                for (every pixel)
                        delta = | current_frame – median_frame|
                        if (current_frame_pixel > median_pixel)
                                median_pixe l + 1
                        else
                                median_pixel - 1 end
                        if (delta > threshold)
                                foreground_pixel = current_frame_pixel
                        else
                                foreground_pixel = 0
                        end
                end
```

and we apply this algorithm in Matlab with the code in "poorRemovingBackground.m" shown below:

```
%removing back ground to make people in blobs
%duplicate some program in sample code "runVideoReader.m"
%a poor application because the rightmost man stayed for a while

%-----------read in video and get the attributes------------------
inputObj = VideoReader('Shopping Mall Wing People.mp4');

nFrames = inputObj.NumberOfFrames;%get the number of frames

frame = read(inputObj,1);

[height,width,d] = size(frame);%get the size of each frame

backGround = rgb2gray(frame);%get the approximate median background

%----------------------processing!-----------------------------

foreGround = zeros(height,width);%initializing the output

threshold = 50;%set the threshold of comparison

for k = 1:5:nFrames
   inputFrame = read(inputObj, k);
   inputGray = rgb2gray(inputFrame);
   frameDiff = abs(double(inputGray) - double(backGround));
   for i = 1:width
      for j = 1:height
         if(frameDiff(j,i) > threshold)

            foreGround(j,i) = 255;
         else

            foreGround(j,i) = 0;
         end

      end

   end
   %show in figure(1)
   figure(1),subplot(2,1,1),imshow(inputFrame);
   subplot(2,1,2),imshow(uint8(foreGround));
end
```

For your information, this code is according to the sample code “runVideoReader.m” from CH7 folder. This code applied the pseudo code and the threshold came from an iterative searching.

However, this method left a problem that the man in the rightmost corner in the first frame stayed for a while in the foreground, which was not synchronized to the read-in video. This description is shown below in figure 1.1 to 1.3.

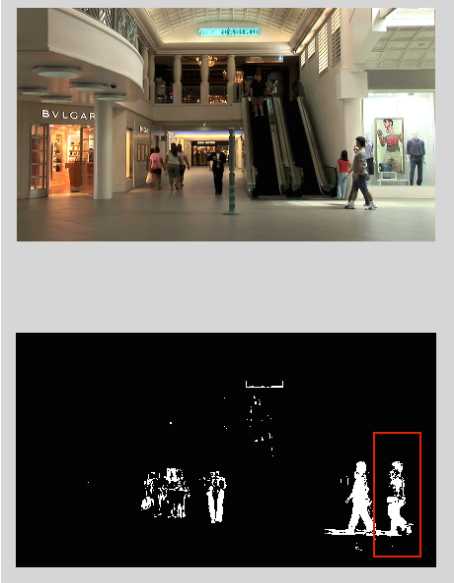

Fig 1.1

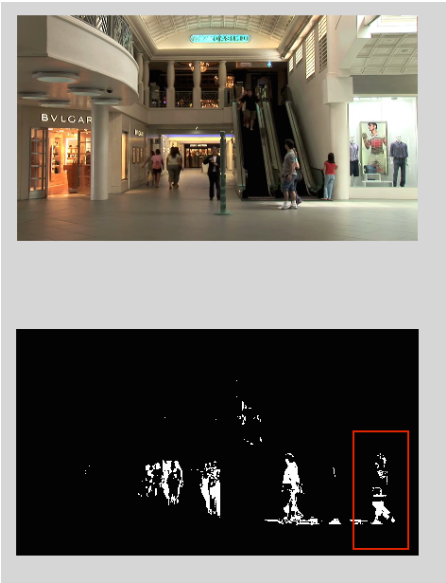

Fig 1.2

Fig 1.3

Our assumption is that this algorithm using first frame as background, and the absolute value of subtraction between each frame and background as standard for extraction. However, for example, the rightmost man in the first frame did not appear in the same or near position in the next consecutive frames, so he become the foreground in the next frames at that position, instead of a background for subtraction.

To solve this problem, we use the average frame as the approximate median to calculate, because the average frame will take convenience of all the information from whole video, instead just use one frame, which make the background more precise. Here is the code for calculate the average frame as background in “framebackgroundavg.m”.

```
%calculate the average frame background of the whole video

%read in video and get the attributes
inputObj = VideoReader('Shopping Mall Wing People.mp4');

nFrames = inputObj.NumberOfFrames;%get the number of frame

frame = read(inputObj,1);

[height,width,d] = size(frame);%get the size of each frame

%get the average value of every pixel in the frame

backgroundAvg = zeros(height,width);

for i = 1:nFrames
    background = double(rgb2gray(read(inputObj,i)));
    backgroundAvg = backgroundAvg + background;
```

and we use the average background to extraction people, the code was in "removingbackground.m" below:

```
%removing back ground to make people in blobs
%duplicate some program in sample code "runVideoReader.m"

%-----------read in video and get the attributes--------------
inputObj = VideoReader('Shopping Mall Wing People.mp4');

nFrames = inputObj.NumberOfFrames;%get the number of frame

frame = read(inputObj,1);

[height,width,d] = size(frame);%get the size of each frame

%-----get the average value of every pixel in the frame-------
load('backgroundAvg.mat');

%---------------------processing!----------------------------

foreGround = zeros(height,width);%initializing the output

threshold = 50;%set the threshold of comparison

for k = 1:1:nFrames
    inputFrame = read(inputObj, k);
    inputGray = rgb2gray(inputFrame);
    %notice we use average frame for extraction!
    frameDiff = abs(double(inputGray) - backgroundAvg);%get diffrence

    for i = 1:width
        for j = 1:height
            if(frameDiff(j,i) > threshold)
                foreGround(j,i) = 255;
            else

                foreGround(j,i) = 0;
            end
        end

    end
    %show in figure(1)
    figure(1),subplot(2,1,1),imshow(inputFrame);
    subplot(2,1,2),imshow(uint8(foreGround));
```

With this code we complete the extraction of people question with the rightmost man synchronized to the read-in video. And we take some screenshots in figure 1.4 to 1.6

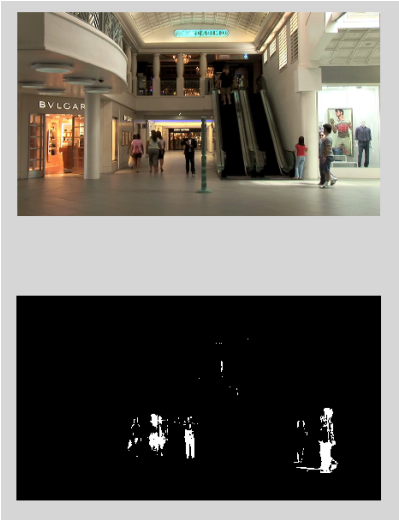

Fig 1.4

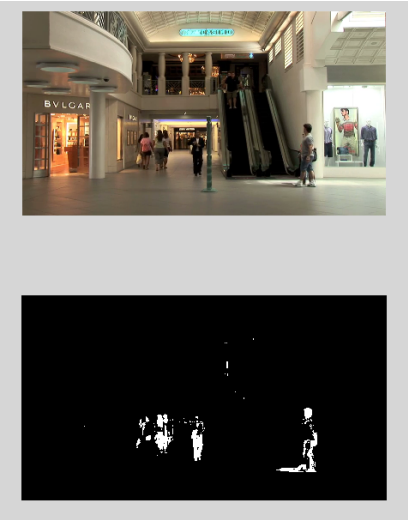

Fig 1.5

Fig 1.6

Motion Energy Image

The solution of Motion Energy Image is based on the previous extraction process. Instead of showing each frame' extraction output, we accumulate them together and then present. The code was in "motioneneryimage.m" below:

```
%removing back ground to make people in blobs
%adding all frames before to make motion enery image

%---------read in video and get the attributes-------------
inputObj = VideoReader('Shopping Mall Wing People.mp4');

nFrames = inputObj.NumberOfFrames;%get the number of frame

frame = read(inputObj,1);

[height,width,d] = size(frame);%get the size of each frame

%----get the average value of every pixel in the frame-----

load('backgroundAvg.mat');

%-----------------------processing!------------------------

foreGround = zeros(height,width);%initializing the output

threshold = 25;%set the threshold of comparison

for k = 1:nFrames
   inputFrame = read(inputObj, k);
   inputGray = rgb2gray(inputFrame);
   frameDiff = abs(double(inputGray) - backgroundAvg);%get diffrence

   for i = 1:width
      for j = 1:height
         if(frameDiff(j,i) > threshold)
            foreGround(j,i) = 255;
         end
      end
   end
%show in figure(1)
   figure(1),subplot(2,1,1),imshow(inputFrame); subplot(2,1,2),imshow(uint8(foreGround));
end
```

and the result was in figure 2.1 to 2.3.

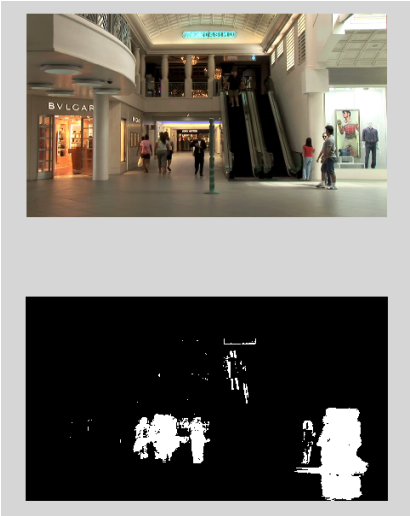

Fig 2.1

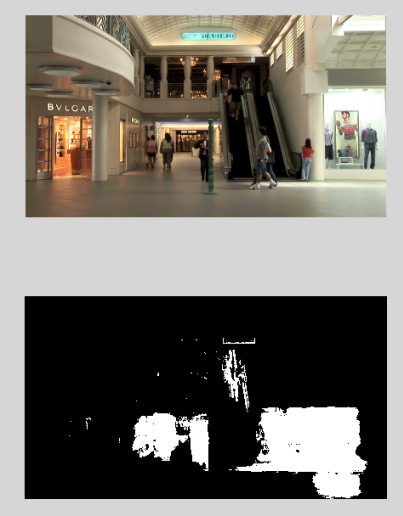

Fig 2.2

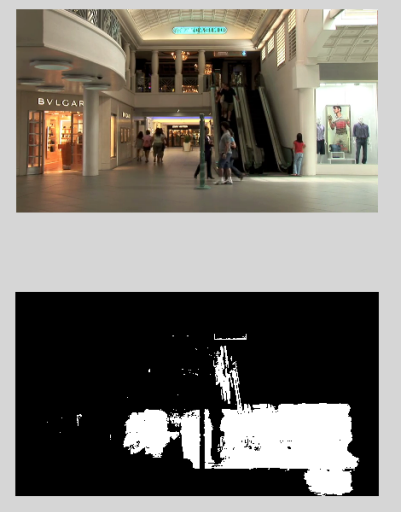

Fig 2.3

Motion History Image

The solution of Motion History Image is based on the Motion Energy Image and the key to process is to make the fade effect in consecutive frames. The code in "motionhistory.m" is listed below:

```
%removing back ground to make people in blobs
%adding all frames before to make motion enery image
%small substraction for frames before to make it fade

inputObj = VideoReader('Shopping Mall Wing People.mp4');
nFrames = inputObj.NumberOfFrames; % Total number of frames

frame = read(inputObj,1); % read in 1st frame as background frame

%-----get the average value of every pixel in the frame-------
load('backgroundAvg.mat');

[height,width,d] = size(frame);%get the size of each frame
% --------------------- process frames ----------------------------------
threshold = 25;
foreGround = zeros(height, width);
for k = 2:nFrames
   inputFrame = read(inputObj, k);
   inputGray = rgb2gray(inputFrame);
   framDiff = abs(double(inputGray) - double(backgroundAvg));
   for i=1:width
      for j=1:height
         if ((framDiff(j,i) > threshold))

            foreGround(j,i) =255;
         else

            if (foreGround(j,i)>0)
               foreGround(j,i)=foreGround(j,i)-10;
            else
               foreGround(j,i)=0;
            end
         end
      end
   end
   figure(1),subplot(2,1,1),imshow(inputFrame);
   subplot(2,1,2),imshow(uint8(foreGround))
end
```

and the result was in figure 3.1 to 3.3.

Fig 3.1

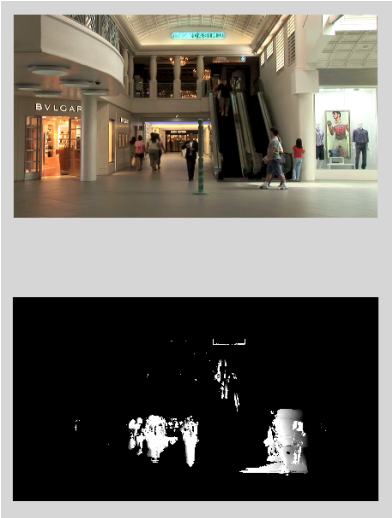

Fig 3.2

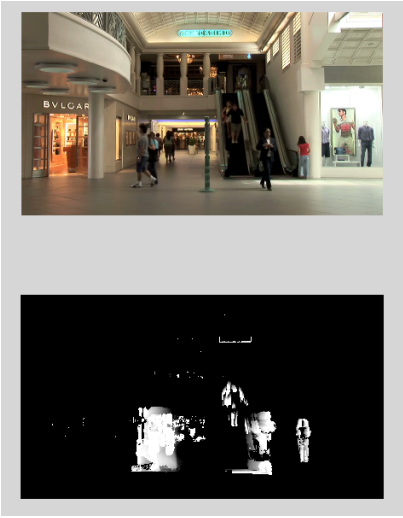

Fig 3.3

Heat Map

The solution of Heat Map is similar to the Motion Energy Image. It is about accumulation of frame but the accumulation was shown in color on the original background. The color was in RGB instead of grey scale, so a method in Matlab for color map was applied. The code in “heatmap.m” was listed below:

```
%Removing background first first
%Accumulate the color in heat map
%Cover the color on original frame
%---------read in video and get the attributes-------------
inputObj = VideoReader('Shopping Mall Wing People.mp4');
nFrames = inputObj.NumberOfFrames;% Total number of frames

frame = read(inputObj,1); % read in 1st frame as background frame

load('backgroundAvg.mat');

[height,width,d] = size(frame);%get the size of each frame

%-------------------------processing!---------------------------
threshold =25;
foreGround = zeros (height, width);

for k = 1:nFrames
   inputFrame = read(inputObj, k);
   inputGray = rgb2gray(inputFrame);
   frameDiff = abs(double(inputGray) - backgroundAvg);
   for i=1:width
      for j=1:height
         if ((frameDiff(j,i) > threshold))
            foreGround(j,i) = foreGround(j,i)+10;
         end
      end
   end
   figure(1)
   imshow(inputFrame)
   hold on
   heat=imagesc(foreGround,[0,255]); %turn double value into color
   set(heat,'AlphaData', 0.5);
   imagesc(heat);
   hold off
end
```

and the result was in figure 4.1 to 4.3.

Fig 4.1

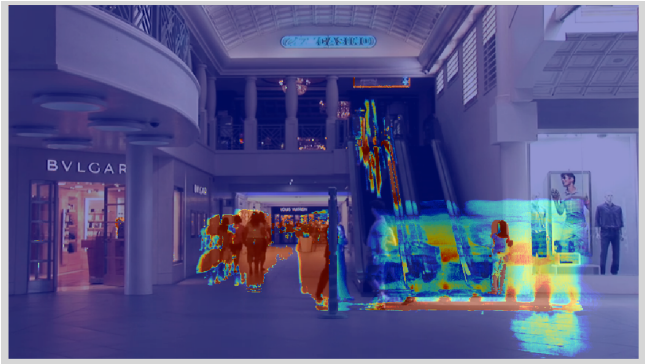

Fig 4.2

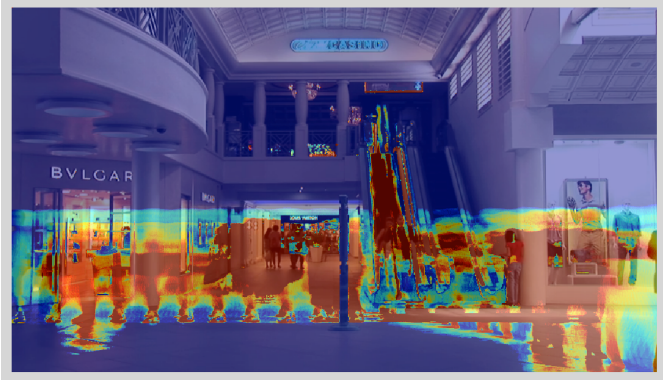

Fig 4.3

Find the static girl

It is a hard task to identify an almost static girl in this video, however, we find to ways to identify the girl.

The first one is to identify the girl with her unique color of shirts and jeans. Use digital color meter from Mac OS, we identify the color of red shirt was within {R(96~140), G(35~60), B(35~60)} and jeans within {R{100~140), G(110~150), B(90~140)}. With the traversal in every frame, we identify the pixel in these color range and identify these pixels as the girl.

The code was listed below from “findgirl.m”

```
%removing back ground to make people in blobs
inputObj = VideoReader('Shopping Mall Wing People.mp4');
nFrames = inputObj.NumberOfFrames; % Total number of frames

frame = read(inputObj,1); % read in 1st frame as background frame

[height,width,d] = size(frame);%get the size of each frame
% ----------------------- set frame size variables -----------------------
load('backgroundAvg.mat');
% --------------------- process frames ----------------------------------


foreGround = zeros(height, width);
threshold = 25;
for i = 1:nFrames
   inputFrame = read(inputObj, i);
   inputGray = rgb2gray(inputFrame);
   frameDiff = abs(double(inputGray) - backgroundAvg);
   for j=1:width
      for k=1:height
         if ((frameDiff(k,j) > threshold))

            foreGround(k,j) =255;
         else
            foreGround(k,j) =0;
         end

         if((inputFrame(k,j,1)<=140)&&(inputFrame(k,j,1)>= 96)&&(inputFrame(k,j,
2)>=35)&&(inputFrame(k,j,2)<=60)&&(inputFrame(k,j,2)>=35)&&(inputFrame(k,j,3)<=60))
            foreGround(k,j)=255;
```

```
        elseif((foreGround(k,j,1)<=140)&&(foreGround(k,j,1)>=100)&&(foreGround(k,j,
2)<=150)&&(foreGround(k,j,2)>=110)&&(foreGround(k,j,3)<=140)&&(foreGround(k,j,1) >=90))
            foreGround(k,j)=255;
        else
            foreGround(k,j)=foreGround(k,j);
        end
      end
    end
    figure(1),subplot(2,1,1),imshow(uint8(inputFrame)); subplot(2,1,2),imshow(foreGround)
end
```

and the screenshot was figure 5.1 to 5.3.

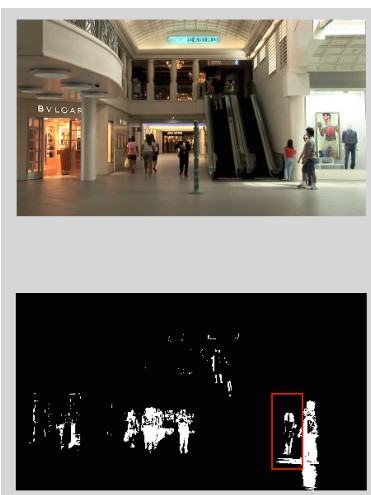

Fig 5.1

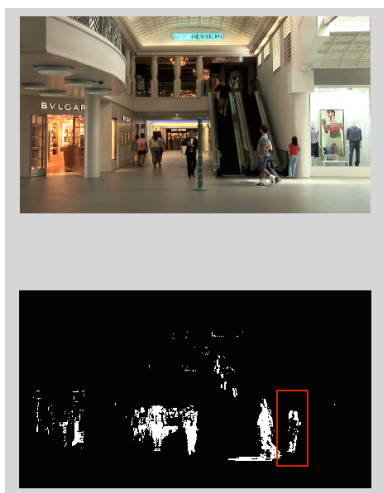

Fig 5.2

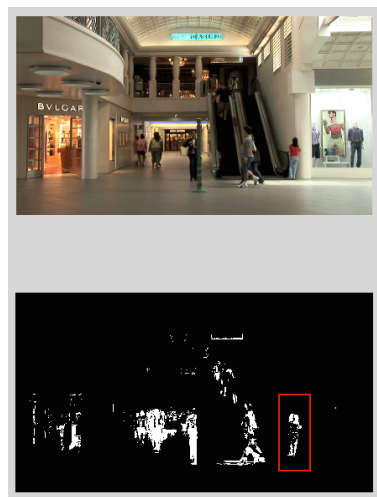

Fig 5.3

Another solution is tricky and may only applicable in this video. We notice that the girl is not completely static. She moved a little, which was not detectable in consecutive frames However, use our average frame as background, her little move will be clear compared with an average background. The code is exactly the same as the average background method in task1, so we do not list here.

The result was in figure 5.4 to 5.6. The result is similar to the color identification method mentioned but it is much easier to apply.

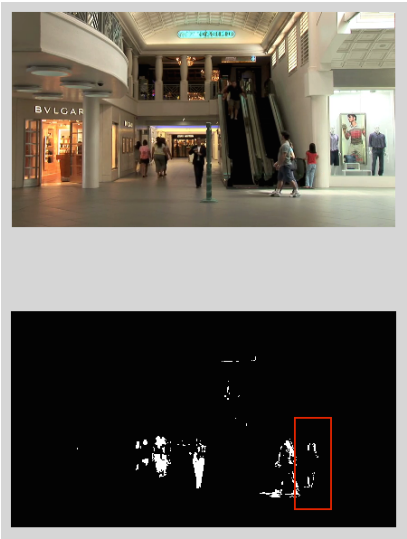

Fig 5.4

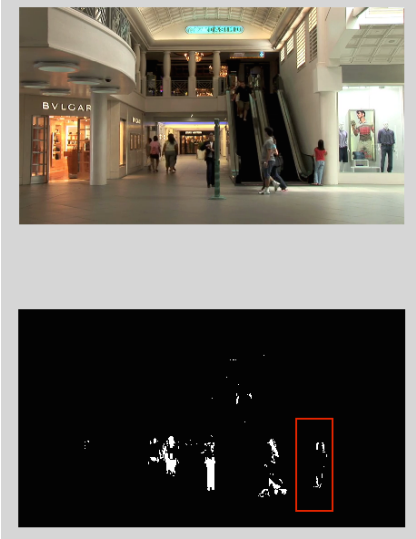

Fig 5.5

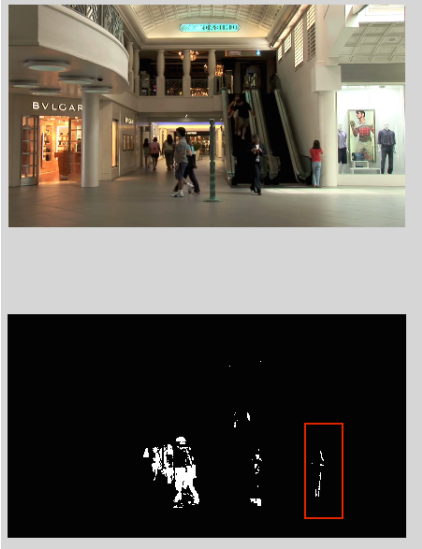

Fig 5.6