TP1-INF6803
VIDEO PROCESSING AND APPLICATIONS

Jinling Xing 1915481

February 9, 2018

1 Introduction

This TP is about video segmentation, including GMM and optical flow. I will perform them on the Matlab.

2 Questions

2.1 Question 1

Description: Identify, based on your theoretical understanding of the two methods, which one should be the best of the two in at least THREE specific use cases. For example, which is the best method to use if the camera is not fixed? Why? And if the contrast is low?

Presentation of the two methods(general principles):

GMM(Gaussian mixture method) is a technique to extract the interest region. The value of each pixel in the image is distributed by a Gaussian *pdf*. The X_t (pixel samples at time t) belongs to the background is given by $w_{i,t}$(the weight of the i-th Gaussian distribution in time t) and $\eta(X_t, \mu, \Sigma)$ (the i-th Gaussian distribution in time t).

Generally, we need to calculate the mean value for each channel of every pixel and the variance. According to this feature, each pixel value X_t is compared with the current K distributions, if the pixel value is far from a value $\mu_{k,t-1}$, then the pixel value belongs to the foreground. If the pixel value is close to the value $\mu_{k,t-1}$ (within a certain range of variance: $2.5 * \sigma$), then this pixel belongs to the background.

Then, the GMM did the parameter update. If the X_t is the background pixel, $M_{k,t} = 1$, otherwise $M_{k,t} = 0$, and then the weights of each distribution is adjusted. The mean value μ and standard deviation σ of the unmatched mode(the distribution in which X_t doesn't belong) are unchanged. The parameters of the matched mode(the distribution in which X_t belong) are updated.

Optical flow is a technique to extract the interested region by means of analyzing two continuous frames in a small time interval, compare the difference between the two images. It should have a obvious movement between the object and background from the observer angle. We did the Taylor expansion of the I(x,y,t) and got an equation of the images. In this case, we have one equation with two unknown variables. On the slides(page 45) of Chapter 2, introducing a Lucas-Kanade optical flow method to solve this problem.

Lucas-Kanade optical flow: we assume the optical flow V_x, V_y in a small region is a constant value and use the least-square method to find V_x, V_y.

Case 1. camera is not fixed

Optical flow is best.

When the camera is not fixed, we now assumed that the object is moving or not moving, relatively speaking, the object is moving for the camera. In this case, we can calculate the difference in a small continuous time interval by means of optical flow method and then we can detect whether the object is moving or not.

Case 2. Contrast is low

I am not sure about that case, but I guess GMM is best.

In general, the contrast is high, the image clearer and more eye-catching, more vivid and beautiful colors; and the contrast is low, it will make the entire picture are gray. If the contrast is low, it will become difficult to distinguish the background and foreground. Maybe we can use parameters in the GMM to adjust the contrast as high, but for optical flow, we can do nothing.

Case 3: light changing

GMM is best.

The assumption on the slides (page 17) of Chapter 2 is that *"Background is varying because of sensor noise, and small motion or changes are possible in the background."*. Because of that assumption, we can guess that the GMM is good to apply on the light changing. But it is not enough for us to believe the GMM is the best method.

In this case, first, We assumed that the only changed thing is light. Analyzing the GMM, we found that the parameter update is the key point to solve light changing problem. Aiming at the problem of light changing, GMM compares the current pixel value with the k distributions $|X_t - \mu_{k,t-1}|$ and then updates weight w_k of each distribution and selects a certain weight coefficient to update parameters in order to adapt to slow changes of light.

2.2 Question 2

Description: Describe in detail the experiments realized to test the hypotheses of the previous point. Which dataset did you use? What are the difficulties in this datasets' videos? Which evaluation criteria did you use? Did you rely on an external framework to test your code?

Test the hypotheses:
In order to test these three hypotheses, I use the same dataset to test the two methods and then compare the difference. Based on the hypotheses on the question 1, I need to change some parameters, which I thought can be used to verify the hypothesis I given at question 1. **Dataset:** My dataset is from the website changedetection, the name of the dataset on this website is highway, which including 1700 frames. The dataset is a car running on the highway.

Difficult of dataset:

Generally speaking, the difficult thing i think in this dataset is that the car speed is uncontrollable, it may affect the performance. In several frames, it maybe appear several cars, so some of cars maybe cannot be detected.
Evaluation criteria:

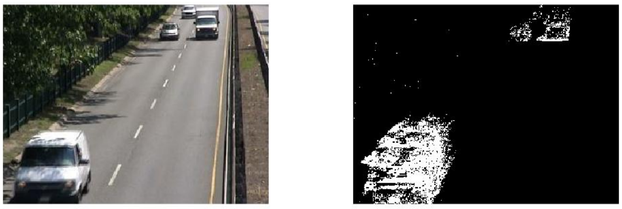

Figure 1: The output of frame 165 without multiplying alpha

I compared the ground truth and the results which I got. what we are different is the 1-469 frames of ground truth is all black. My results has the detection of cars. The quality of my code is based the performance.
External framework:

I don't need it.

2.3 Question 3

Description: Describe the implementation of the two studied methods. If you did not write all the code yourself, where does it come from? Did it require modifications? Otherwise, from which papers or websites did you inspire yourself to write it? In all cases, what are the primary parameters of your methods? How did you set their values?

Implementation of GMM:

- First, I imported all of the frames using a for loop, and took out all the images with 'jpg' as end from the folder, and then took out the images in order. At this way, we can see on the left side on Matlab, our dataset is imported.
- Second, we initialized the RGB channel C=3, the Gaussian component K=3. I set the first frame as background. We had three important matrices, mean matrix, weight matrix and standard deviation matrix. The most important thing in the second step is to construct the background mode and update the background model. When we verified if measurement X_t os inside one of the K distributions, we used the difference between the $X_t - \mu_{k,t-1}$ to make sure whether we need to update the model. Before I used an "if-else" construction to update it, I calculating the difference from mean for each pixel in each Gaussian distributions. How to update the parameters, the theoretical background is based on the paper "Adaptive background mixture models for real-time tracking" and the slides on the page 19-23 of chapter 2.
- The detailed description of updating the parameters is that I set a stable parameter α as study rate and combine the match(whether the pixel is belong to the background or foreground). We still need to update the mean matrix and the standard deviation(variance)

matrix. The beta is the product of the alpha and Gaussian distribution, we use the $beta = alpha * normpdf(double(bg(i,j)), mean(i,j,k), variance(i,j,k))$; to define beta and then use it to update the variance and mean matrices.

- The output of the result, I exported the background and the images we compared for 1700 times and saved them on a folder. Then, I can compare the difference between my output results and the ground truth.

The running time of GMM is as below:
$>> INF6803TP1$
Elapsed time is 3406.607561 seconds.

Implementation of optical flow:

- First, I imported all of the frames using a for loop, and took out all the images with 'jpg' as end from the folder, and then took out the images in order. At this way, we can see on the left side on Matlab, our dataset is imported.
- Second, I extracted the derivatives of x and y and compared the continuous frame, the first one with the second one, the second one with the third one and so on.
- Third, we implemented the Lucas–Kanade algorithm. For matrix form Av=b, we cannot inverse it, because it is not a square matrix.
- I output the results in a folder.

How to write code?

For the GMM: This code is built from existing code at "http://areshmatlab.blogspot.ca/2010/05/high-complexity-background-subtraction.html". The structure is similar, but the computation is different and adapted to the slides of GMM. How to do the background model update, we are totally different. I also read the paper"Adaptive background mixture models for real-time tracking" to understand the detailed information of GMM.

For optical flow: This code is built on the example which professor shown on the course. what I have modified is to compare the continuous frames of all the dataset, not only two frames, and I also rewrote some part of the implementation of lucas-kanade algorithm.

Primary parameters

For optical flow, there are not too much parameters we needed to set. i set the windows size is 25, according to the example on the course.
For the GMM:
$K = 3$; this is the number of Gaussian components, normally the components is 3-5. I set it as 3.
$alpha = 0.01$; learning rate, this we can find on paper.
$beta = alpha * normpdf(double(frame_b g(i,j)), mean(i,j,k), variance(i,j,k))$; using Gaussian pdf to calculate the beta.
mean, weight, variance(standard deviation): when I initialized the background model, I used the first frame pixels value as the mean, and equal weights, and the variance is the initial variance, it it 5. when I did the background model update, I set the mean, weight, variance based on match.

```
if verify_K_distribution(i,j,k)<=2.5*variance(i,j,k)
    m=1;
    %m is the match on slides page 21
    %the parameters update according to the formulas on
    %slides on page 22
    weight(i,j,k) = (1-alpha)*weight(i,j,k) + alpha*m;
    beta = alpha * normpdf(double(frame_bg(i,j)),mean(i,j,k),variance(i,j,k));
    %gaussian pdf
    mean(i,j,k) = (1-beta)*mean(i,j,k) + beta*double(frame_bg(i,j));
    variance(i,j,k)=sqrt((1-beta)*variance(i,j,k).^2)
    +beta * transpose(double(frame_bg(i,j))-mean(i,j,k))
    *(double(frame_bg(i,j))-mean(i,j,k));
else
    m=0;
    %%%%the adjustment when m==0, is according to the
    %%%%description on slides on page 21

    %the distributions with the smallest weight is replaced
    %with a new one with the value of the first frame as the
    %mean, a large arbitrary variance(the initial one) and a
    %small weight
    weight(i,j,:) = weight(i,j,:)./sum(weight(i,j,:));
    [min_weight,min_weight_index] = min(weight(i,j,:));
    mean(i,j,min_weight_index) = double(frame_bg(i,j));
    variance(i,j,min_weight_index) = variance_init;
end
```

2.4 Question 4

Description: Provide the evaluation results from your experiments related to the hypotheses of the first point. Use a proper format for their presentation — tables, figures, ...

For the contrast is low. I tested it by changing the parameter: beta on GMM. I set the beta as two ways:

```
%alpha==0.01
beta = alpha * normpdf(double(frame_bg(i,j)),mean(i,j,k),variance(i,j,k));
beta = normpdf(double(frame_bg(i,j)),mean(i,j,k),variance(i,j,k));
```

Let's looking at difference by comparing same image with different beta. We can see that: the figure 4 has more noise points apart from the detecting the car. We know that the contrast is the ratio of the black figure to white figure. When the beta become smaller, the black part is becoming smaller(the white noise points become more), the contrast is becoming smaller. The compared images are figure.

Figure 2: The output of frame 50 without multiplying alpha

Figure 3: The output of frame 50 with multiplying alpha

For the light is changing: light may have slight changing. We can compare the continuous frames for both GMM and optical flow. For each method, we compared three frames, it seemed that the car is not changing so fast, but the light can be changed. What we want to see is that the detect effect has not affected. From the continuous frames, although we can not know the light changing directly, a slight changing is possible, we can guess if the light changing, the effect is same. This hypotheses need to be totally supported, we need to change our dataset, because the light changes are not obvious. figure 4-9.

Figure 4: The GMM output of frame 100

Figure 5: The GMM output of frame 101

For the camera is not fixed: We didn't know whether the camera is fixed or not fixed, so we cannot compared this situation. But we can compare the same frame of these two method. When the camera is not fixed, we now assumed that the object is moving or not moving, relatively speaking, the object is moving for the camera. If the optical flow is best, the quality of the output images should be better. For the same frame 100, we can look at the figure 4 and figure 7. The GMM method has more noise points, the optical flow is more clear than GMM.

Figure 6: The GMM output of frame 102

Figure 7: The Optical flow output of frame 100

2.5 Question 5

From the figures above:

Discuss the results of the fourth point in relation with the hypotheses of the first point.
I have already talked it on the question 4.
camera is not fixed, the optical flow is best.
contract is low, the GMM is best.
light is changing, not supported.
For this reason, we need to change our dataset, which has a obvious light changing, we can use to assess these two methods.

Figure 8: The Optical flow output of frame 101

Figure 9: The Optical flow output of frame 102