Lab1: Foreground Segmentation

Tamás BUKITS and Balázs Márk BÓDIS

I. INTRODUCTION

In this assignment we developed multiple algorithms based on different background models to perform foreground segmentation. Firstly, we both developed algorithms, which work based on frame difference, with static and updating background models and finally stationary object suppression. Secondly we evaluated our models and chose Tamás' algorithm as a starting point for the following tasks. Next, we implemented the shadow suppression module for the chosen algorithm. We implemented the Single Gaussian algorithm for the segmentation task in both gray-level and color versions. Lastly we implemented a Multimodal model for the background (Mixture of Gaussians). This last implementation had major problems which we will go in detail. Overall, we achieved the best result with our basic background model enhanced with shadow suppression (task 3).

II. METHODS

As we did not deviate from the suggested methods in the exercises we will give a very brief overview in this section of the models used. All methods (except the last) can be called for gray-level or rgb modes using the **(rgb)** flag. Other, method specific input parameters are highlighted for each model.

A. Frame difference - static and updating background model

The segmentation is based on the absolute difference between the background model and the current frame. A threshold **(tau)** is given, if the difference is greater than this threshold we assign the pixel to the foreground class, otherwise it is background. In the updating background model we change all or currently background pixels of the background model **(blind or selective update)**. The new pixel value is the linear combination of the previous background and the current frame. The ratio is decided by the **(alpha)** factor.

B. Stationary object suppression

This builds upon the last method. If a pixel was assigned for a set amount of time it is assumed to be background and the model will be updated. This approach combats hot-start ghosts but can introduce false negative detections on stationary foreground objects. The number of allowed foreground frames can be set via the **(ghost_threshold)** parameter.

C. Shadow suppression

The detection of shadows based on chromaticity. For this we require the HSV values of the pixels. A pixel is considered shadow if all three statements described in the slides are met. For this 4 extra parameters were required **(alpha_sh)** **(beta_sh)** **(saturation_sh** and **(hue_sh)**.

D. Single Gaussian Background model

Each pixel has a gaussian model, represented by a mean and a standard deviation value. The model is updated based on the occurrence of pixels. If a pixel falls outside of the distribution it is considered foreground, otherwise it is background. The gaussian is initialized so that the mean is the pixel value and the deviation is relatively high (100 in our case).

E. Multimodal Background model (Mixture of Gaussians

Each pixel has **(K)**$\in\{3,4,5\}$ gaussians.

1) Implementation: We have K gaussian classes. Each class contains a set of means and deviations for each pixel. A ranking is made for each pixel based on the weight of the distributions. The main difficulty in implementing this method came from the high amount of parameters required.

III. DATA

For the exercises we used the CDNET (ChangeDetection.NET)[1] database. The databases was specifically created to test algorithms based on change detection methods. For our testing the results we used the 2012lite version of the dataset, which was provided to us by the course administration through the university's moodle platform.

IV. QUALITATIVE ANALYSIS OF METHODS

A. Frame difference and Stationary Object suppression

As we can see from *Figure 1* there is a blurring effect present for moving objects. We suspect this is the result of the background update model. As setting the alpha parameter lower (less than 0.1) reduces this effect. We also experimented with the ghost_threshold parameter, but as it was already high enough (around 25) it did not cause this effect (the cars are moving fast "enough" here). From *Figure 2* we deducted that the background update successfully adjusts to slow changes (light changes etc.) in the background. In the top right corner we can see the light difference between the current frame and the starting frame. While we can detect individuals moving, they move slower than cars on the highway, thus reducing the ghost_threshold can lead to less false negative detections. We reason that the "hole" in

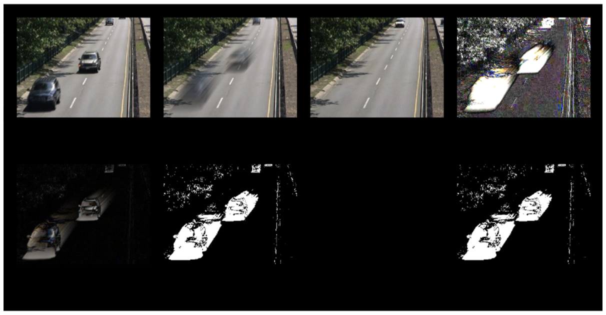

Fig. 1. Highway sequence with subtraction

Fig. 2. Pedestrians sequence with subtraction

the mans stomach who is wearing a red shirt in *Figure 3* is caused by the stationary object suppression part of our algorithm. Another problem this algorithm introduces is disappearing subjects. As the man starts reading he becomes stationary, thus in a few frames he is almost completely undetected. This can be observed in *Figure 4*. The results of the evaluation can be seen in *Table 1*.

B. Shadow Suppression

Our shadow detection works best on the PETS2006 sequence, a frame from this sequence can be seen on *Figure 5*. The shadows are all correctly detected and subtracted from the background and the foreground masks. In the pedestrians sequence it works similarly on few individuals but mostly performs like it is show on *Figure 6*. Some of the shadow is detected, but the outline remains. This is probably due to the texture of the ground (grass). On *Figure 7* we can see how in the office sequence we pick up shadows on the

Fig. 3. Office sequence with subtraction

Fig. 4. Office sequence with subtraction

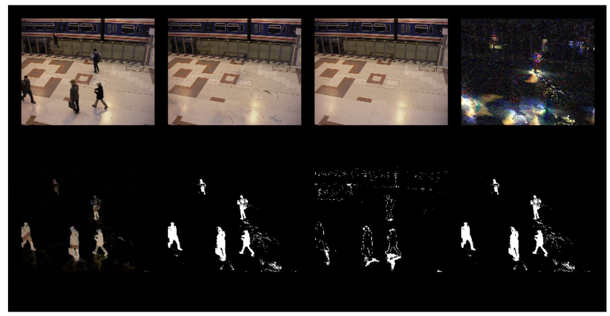

Fig. 5. PETS2006 sequence with shadow suppression

foreground subject himself. This results in reduces accuracy. Lastly, our shadow reduction does not work well on concrete, the chromaicity difference is hard to detect on this type of gray ground with the given lighting conditions. However the interior of some cars are detected as shadow. An example frame this sequence, where both of these occur can be seen on *Figure 8*.

C. Single Gaussian

Our gaussian algorithm works well on both the highway and pedestrian sequences. Examples of these can be seen in *Figures 9&10*. However it struggles to detect humans when they walk over the darker areas of the flooring in the PETS2006 sequence (*Figure 11*). We suspect our scores are lower on this method because of the lack of stationary foreground suppression. We will mention this again in the final comparison between our solutions.

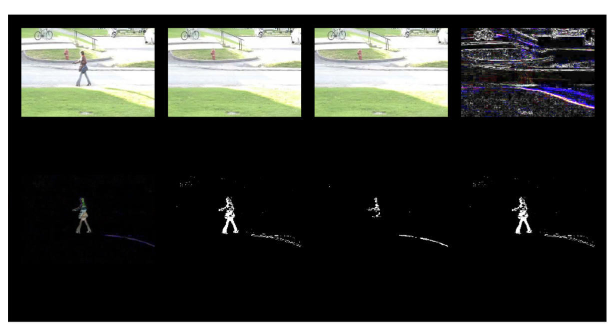

Fig. 6. Pedestrians sequence with shadow suppression

Fig. 7. Office sequence with shadow suppression

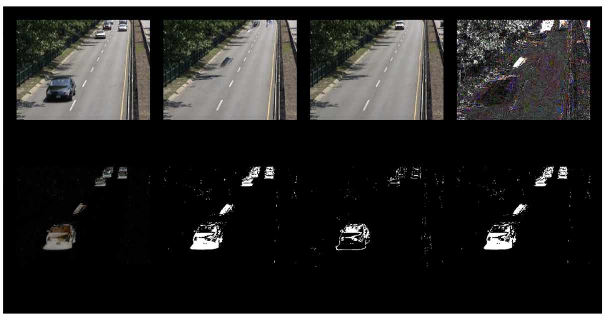

Fig. 8. Highway sequence with shadow suppression

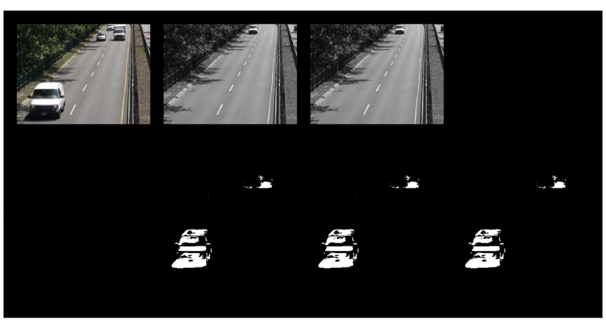

Fig. 9. Highway sequence with Single Gaussian

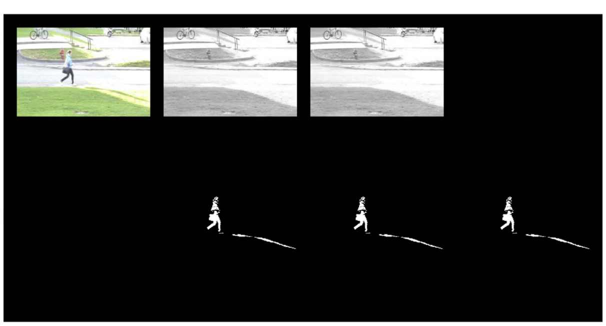

Fig. 10. Pedestrians sequence with Single Gaussian

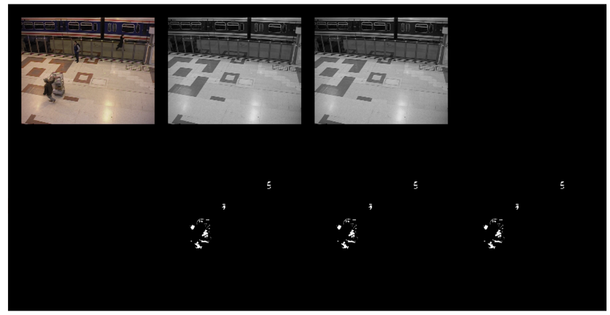

Fig. 11. PETS2006 sequence with Single Gaussian

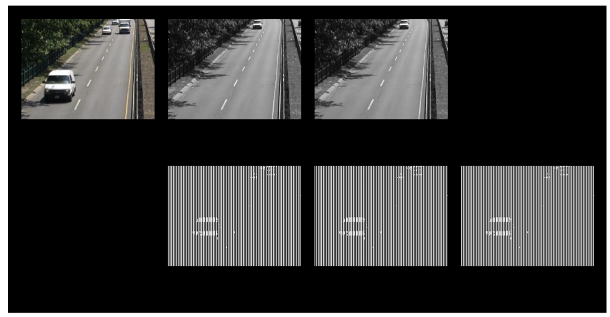

Fig. 12. Highway sequence with Mixture of Gaussians

D. *Mixture of Gaussians*

During the implementation of this model we ran into a logic error that we could not resolve. This causes artifacting on the mask of which we do not know the reason of. The artifacts and the detection (barely visible) can be seen on *Figure 12*.

V. QUANTITATIVE COMPARISON OF METHODS

The results can be seen in *TABLE 1*. We can see that our shadow suppression is a worth adding to the base frame difference + Static Object Suppression, as expected, Precision increases and Recall decreases. Single gaussian performs the best on recall. However it falls short on precision. Both of these values are effected by the error mentioned before. As for mixtured gaussian, the problem of artifacting explains the absolutely abysmal precision. However we expected better Recall, this signals, that while just looking at the frames the error does not seem that large, we might be dealing with a wrong implementation. However we believe our approach is correct (even if it is not the most efficient way of implementing it).

Method	Recall	Specificit	Precision	F	Overall
Frame Dif. + SOS	0.642	0.981	0.508	0.509	0.7470212016
+Shadow suppres.	0.556	0.992	0.727	0.608	0.7719485108
Single Gaussian	0.917	0.869	0.260	0.402	0.7298452717
Mixture Gaussians	0.510	0.467	0.038	0.068	0.3609984677

TABLE I
QUANTITATIVE RESULTS

VI. CONCLUSIONS

We conclude, that our simpler models could get by changing the parameters empirically, but we need to implement a Static Object Suppression for Single Gaussian. As for the Mixture Gaussian model, we need to investigate the source of the logic error.

VII. USAGE

For each task, the dataset location is as follows:

```
string dataset_path ="/home/avsa/
    AVSA2020datasets/dataset2012lite/dataset";
string dataset_cat[1] = {"baseline"};
string baseline_seq[4] =
    {"highway","office","pedestrians","PETS2006"};
string image_path = "/input/in%06d.jpg";
```

The results are saved to, where "X" is the task number:

```
string project_root_path =
    "/home/avsa/Documents/AVSA2020results/";
string project_name = "TaskX"; // project exe name
string results_path =
    project_root_path+"/"+project_name+"/results";
```

VIII. TIME LOG

- Task Comprehension: 2 hours
- Data analysis: 30 minutes
- Coding & Debugging (individual): 5-6 hours
- Coding & Debugging (co-operative): 25 hours
- Report: 5 hours

Total: around 38 hours,
of which around around 10 hours were shared.

REFERENCES

[1] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, changedetection.net: A new change detection benchmark dataset, in Proc. IEEE Workshop on Change Detection (CDW-2012) at CVPR-2012, Providence, RI, 16-21 Jun., 2012.