Lab-1 Foreground Segmentation

Kush Gupta and Sergio Avello

I. INTRODUCTION

This laboratory report presents foreground segmentation methods, written in C++ using the OpenCV library on eclipse. The input video sequences are assumed to be captured from a still camera. We have implemented various foreground segmentation methods like frame difference, progressive update of the background using the blind, and the selective mode. Removal of objects from the foreground which are not moving after some frames. We also implemented the shadow suppression method based on chromaticity. To enhance the results, we tried advanced methods based on Gaussian models.

II. METHOD

For foreground and background segmentation of the provided images/video sequences various methods were implemented. A brief description of the methods will be provided below.

A. Method 1 - Shadow suppression based on the chromaticity (Task 3).

Firstly, the RGB image (current frame) and the background image is converted to HSV. The main reason is that the HSV color space explicitly separates chromaticity and luminosity and has proven easier than the RGB space to set a mathematical formulation for shadow detection. Afterwards, individual components of the HSV color space of the image and background are considered to determine if a pixel is a shadow pixel. A foreground pixel is expected to show chromaticity distortion or a large brightness distortion, while a cast shadow pixel is expected to show darker brightness than it's corresponding background model. [1]

B. Method 2- Single Gaussian Model (Task 4)

The single Gaussian model is a parametric approach. For each pixel, there is a gaussian distribution characterised by its mean $\mu[x, y]$ and its standard deviation $\sigma[x, y]$. If the intensity value of the pixel lies within three σ then that pixel is considered as a background pixel, otherwise it's considered as a foreground pixel.

C. Method 3- Gaussian Mixture Model (Task 5)

Gaussian mixture model is the method of modelling pixels as a weighted sum of gaussians. GMMs are widely used to cluster data, where each point in the n-dimensional feature space gets associated with each of the clusters with a certain probability, unlike in k-means clustering, where a point in the feature space gets associated with only a single cluster. Each of these clusters are parameterized by the cluster mean $\mu[x, y]$, the covariance $\sigma[x, y]$ and weight (w).

III. IMPLEMENTATION

In the submission zip file there are three folders, corresponding to Task 2, Task 3 and Task 4. These three folders work independently and have their own source files in order to simplify the review of each task.

A. Functions for Method 1

This method corresponds to the Task 3 folder. For the Shadow suppression model we implemented the function removeShadows(), where we used the HSV planes of each original image and some thresholds for creating the shadow mask. After that, we obtained the foreground mask subtracting the shadow mask from the background mask .

B. Functions for Method 2

This method corresponds to the Task 4 folder. For the Simple Gaussian model we implemented the function unimodalGaussian(cv::Mat Frame, int it) both for gray scale and RGB. In this function we compare the value of a pixel with its corresponding gaussian in order to determine if it belongs to foreground or background. If the pixel belongs to background we update the mean and the variance of its gaussian.

C. Running the code

In order to compile, link and run each of the tasks, it is only needed to run make in the terminal in the src folder of each task and then run ./(built object). For every task, it is also possible to change which model (from Task 1) will be used, but for that you must edit the main source file following the instructions that are commented on it.

IV. DATA

The data set analyzed contains five video sequences named empty office, eps hotstart, eps shadows, hall, stationary objects. The changedetection dataset's website http://changedetection.net/ wasn't working so we had to use the above mentioned videos.

V. RESULTS AND ANALYSIS

All the results were obtained with the Task 1.1.3 model since we got the best results with it.

A. Task 2: Evaluation of Results from Lab 1

Figure 1 contains the evaluation results obtained for different set of parameters for the algorithm (1.1.3)- Suppression of stationary objects that appeared or removed performed on highway, office, pedestrians, PETS2006 data sets.

The third set of parameters showed good results overall, as it can be seen on Figure 2. So, we used those parameters for the subsequent tasks.

Parameters	Recall	Specificity	Precision	FMeasure	Overall
tau = 18; alpha=0.2; selective_bkg_update = true; threshold_ghosts2=25; bool rgb = true;	0.48532	0.9915	0.594	0.4709	0.8872
tau = 18; alpha=0.2; selective_bkg_update = true; threshold_ghosts2=25; bool rgb = true;	0.5746	0.9847	0.5045	0.4764	0.926
tau = 23 alpha=0.5; selective_bkg_update = true; threshold_ghosts2=25; rgb = true;	0.1875	0.9982	0.7394	0.2775	0.9154

Fig. 1. Evaluation results obtained for different set of parameters for the algorithm- Suppression of stationary objects that appeared or removed.

Fig. 2. The above figure shows an example of the evaluation results.

Since the link http://jacarini.dinf.usherbrooke.ca/results2012 was not working we could not compare the results.

B. Task 3: Shadow suppression

We implemented the code for shadow suppression on the following data-sets empty office, eps hotstart, eps shadows, hall, stationary objects. The results for shadow suppression were fairly reasonable as shown in Figures 3, 4, 5. The algorithm was able to eliminate shadows as part of the foreground. However, sometimes this approach affected the foreground mask.

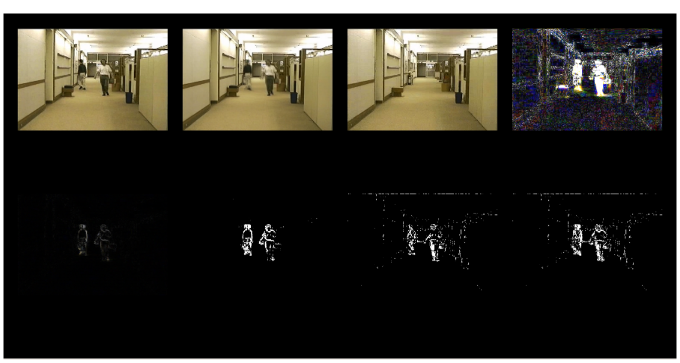

Fig. 3. The figure shows an example of the Shadow suppression in empty office video.

Fig. 4. The figure shows an example of the Shadow suppression in hot start video.

Fig. 5. The figure shows an example of the Shadow suppression in eps shadow video.

C. Task 4: Single Gaussian Model

We tried to implement the algorithm as mentioned in the lecture slides. However, we did not get the expected results. We tried to figure out the issue and it seems that there was a problem with the background mask that deeply affected the obtained results. This malfunction is shown on Figures 6, 7.

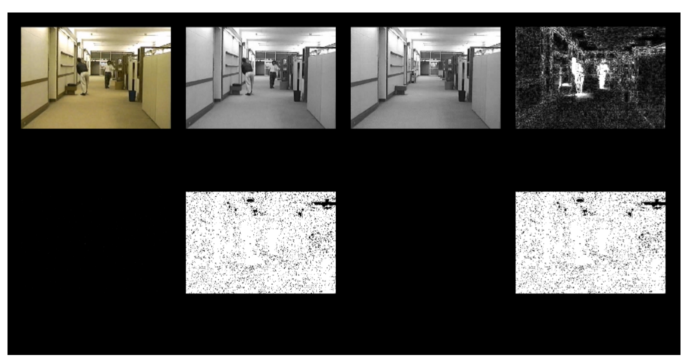

Fig. 6. The figure shows an example of single gaussian model on empty office video.

VI. CONCLUSIONS

In conclusion, the tasks that were performed as part of Lab two were intuitive. The complexity level increased in the subsequent tasks. Upon evaluating the algorithm developed in lab 1, it showed good results, when tested using the

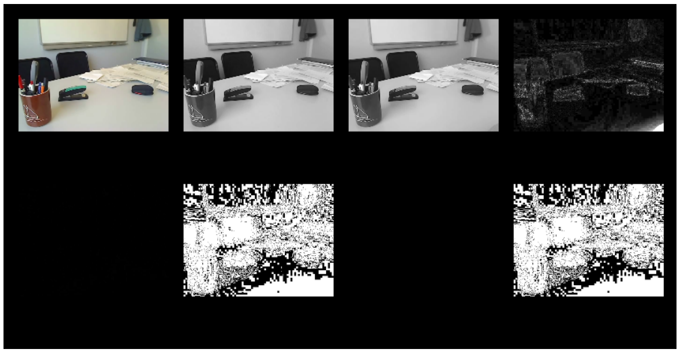

Fig. 7. The figure shows an example of single gaussian model on stationary objects video.

provided matlab script. The removal of shadow from the foreground mask was a challenging task, as we had to find out an optimum set of parameters in order to suppress the shadow. Our suppression algorithm showed quite reasonable results. For single gaussian model, we tried to implement the algorithm as mentioned in the slides. However, the results were not as expected. The problem in this case might be with the update of background mask.

On the other hand, While working on the assignment we learned a lot of new things, with respect to C++ and opencv. The tasks defined in this lab were fairly interesting. However, due to their complexity, the tasks required more time to analyze the problems while coding. Hence, due to time constraints we could not work more to improve our results. We look forward to the next assignment in order to improve our results and learn more advanced concepts to develop our skills in video analysis.

VII. Time Log

Writing code: Around 30 hours, for coding and debugging the errors.
Report: At least 7 hours: writing, taking screenshots of relevant frames, proof-reading, adding and checking references.

References

[1] R. Cucchiara, C. Grana, M. Piccardi and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 10, pp. 1337-1342, Oct. 2003, doi: 10.1109/TPAMI.2003.1233909.