LAB 1 - FOREGROUND SEGMENTATION

APPLIED VIDEO SEQUENCES ANALYSIS

Zsófia Sára Budai and Juan Manuel Peña Trapero
{budai9902, juanmaptcg}@gmail.com

I. INTRODUCTION

This laboratory is composed of four independent tasks and it builds on the previous assignment (Task 0 and Task 1). The first task focuses on understanding and using of a metric used by a real detection competition. The second task is to implement a shadow suppression module and then evaluate it. In the last two assignments, two advanced background subtraction algorithms are implemented and tested: Single-Gaussian and Gaussian Mixture Model.

This report is divided into a method section describing the theoretical bases used, a section describing and explaining the implementation carried out, another section with results and their analysis and, finally, the conclusion of the work carried out.

II. TASK 2: EVALUATING OUR ALGORITHM

First, we have evaluated and compared with the ChangeDetection dataset 2012[1] our implementations for the Task1 of this Laboratory.The evaluation code provided by the competition organisers calculates and displays a considerable number of metrics that are briefly described on the ChangeDetection website. These metrics are based on comparing the results obtained with the Ground Truth to obtain the amount (how many pixels of the mask) of False Positives, False Negativs, Precission, ... All these parameters make the comparison very simple and are detailed in the results section.

III. METHOD

This section describes briefly the methods we have implemented.

A. Shadow suppression (Task 3)

For this task the basic shadow detection algorithm presented on the course slides is implemented. This simple algorithm uses the chromacity values in the HSV space to detect the pixels that are changing due to the light change produced when a object (part of the foreground) is projecting a shadow over the frame. The final goal of this shadow is to subtract it from the background final mask. This is because the intent of the segmentation algorithm is not to incorporate all the variations to the foreground: the noise, the shadows and the reflections should not included.

From the evaluation perspective, the performance evaluation provided by ChangeDetection includes a measure called Shadow Error (SE) that evaluates how many shadow pixels has been classified as foreground. The determination of SE is possible because the ground truth used by the evaluation script includes not only the result of the ideal motion detection mask but also other grey levels in order to evaluate whether shadows are being misclassified. As can be seen in Figure 7 , the shadow cast by the pedestrian has a softer grey level, indicating that this part of the image corresponds to the shade.

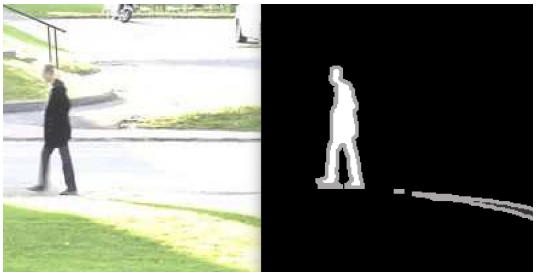

Fig. 1. Frame sample of Pedestrian sequence (left) and its corresponding ground truth (right).

In this point our algorithm was not improving the performance of the original version (without shadow rejection) due to a mistake in the shadow mask computation. The problem was that we were not disregarding the background completely so the shadow mask had many artifacts as shown in the Figure 4. This behaviour was due to the fact that the shadow detection algorithm detects small variations in intensity and color, therefore the image noise was being missclassified as shadow.

Before finding the error we decided to try to improve the shadow masking by using OpenCV's"image_opening" function.

```
opening = cv.morphologyEx(img, cv.MORPH_OPEN,
    kernel)
```

This function performs an erosion followed by a dilation to remove the salt noise as shown in Figure 3. The obtained results with one of the baseline images is displayed in the Figure 2.

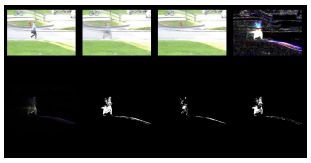

Fig. 2. Implemented shadow suppression.

Fig. 3. Example of morphological opening.

Fig. 4. Shadow mask comparison of a PETS2006 abalyzed frame before and after opening.

B. Single Gaussian Background Subtraction (Task 4)

This is the simplest way of employing Gaussian functions to model the foreground/background behaviour of image pixels. For each pixel the mean and standard deviation will be stored and dynamically updated. When the pixel has a new value that is not expected (a deviation larger than the standard deviation of the Gaussian), this pixel will be considered as a changing pixel that should trigger classification. The necessary changes will therefore depend on each pixel, since, as the sequence is analysed, the Gaussian becomes narrower if a pixel remains completely constant. For example, if an area of the image is overexposed (the pixels of the sensor are saturated so they are always assigned the highest value 255 in a 8-bit B&W image) these pixels will have a Gaussian with mean 255 and a standard deviation of zero so it will be sufficient to lower a single grey level to trigger detection of a change in those pixels.

C. Gaussian Mixture Model Background Subtraction (Task 5)

In this part, the simplified version implemented in the previous task is extended. Now, for each pixel, 3 to 5 gaussians are defined to determine the expected modes or ranges that each pixel will be expected to have. In order to control the different gaussians, the concept of weight is introduced. The purpose of this weight is to determine the creation of new Gaussians. When a pixel is outside these distributions it will be classified as foreground and it might be added to the background model. It will be necessary to include it in the background model (create a new gaussian with mean of the pixel value) when the pixel stays outside the distributions for a sufficiently long period of time. The term sufficient time should be determined during the implementation. In terms of design decisions, GMM has the following key points:

- Initialisation of Gaussians: which are the default values of the standard deviation and how many Gaussians are going to be used.
- Choice of when to include pixel as a new mode of the background model: default value of standard deviation.
- Type of update: blind or selective.

In the implementation section the decided implementation will be described.

IV. IMPLEMENTATION

A. Functions for Task 3: Shadow suppression module

As can be seen in the header of the function below , this code takes as input parameters the 3 values used to compare the changes in hue, saturation and luminance. The code implemented in this function iterates over each pixel stored in the matrix Frame and decides, depending on the change produced in each channel of the HSV and the background, if the pixels are considered shadows or not. The result of this decision for each pixel is stored in the private variable $shadowMask$ of the fgseg class.

```
_shadowMask = update_shadowmask(Mat Frame,
                  double alpha_shdw,
                  double beta_shdw,
                  double tau_shdw_s,
                  double tau_shdw_h)
```

The parameter setting at this point is complex, as not only do you have to select the parameters for the background update, but now you have to add four new parameters for shadow detection. This is especially intricate since, for each video sequence employed in the baseline, there will be a different set of optimal parameters. These will depend on the illumination (how strong the shadows are), how noisy the sensor is (depending on the ambient illumination all image sensors will add noise in both luminance and colour). For this reason parameters that work ideally for some videos will perform worse for other videos within the baseline group we are using to perform the objective evaluation of our shadow detection model.

On the other hand, the necessary logic has been implemented to perform a selective filtering of the background areas in which, if no shadow movement is detected, there is no way to calculate whether the pixel is a shadow or not.

B. Functions for Task 4: Single Gaussian Backgorund Model

For this task, a new function is created to initialise the structure necessary to store, for each pixel, the parameters of the Gaussian that is going to be shaped. In the case of design choices, these can be divided into three parts:

- Data structure: The mean and the variance were stored in cv::mat with CV_32F values.
- Initialisation of Gaussian's parameters: The matrices are initialized with zeros in the dimensions of the frame. The mean gets its initial value for the fist frame from the frama.
- Hyper-parameters of the model: alpha = 0.1, double alpha_shdw = 0.5, double beta_shdw = 0.9, double tau_shdw_s = 90, double tau_shdw_h = 100.
- Update of Gaussian's parameters in each frame with the following equation: $\mu_{t+1}[x, y] = \alpha Image_t[x, y] +$

$(1-\alpha)\mu_t[x,y]$ and $\sigma^2_{t+1}[x,y] = \alpha(Image_t[x,y] - \mu_t[x,y])^2 + (1-\alpha)\sigma^2_t[x,y]$

The algorithm was very noisy and it effected the foreground mask as well as it is seen in the 5 figure, even with the implemented shadow suppression in case of rgb images.

Fig. 5. Implemented simple Gaussian.

C. Task 5: Multiple Mixture Gaussian Backgorund Model

This is the most complex of the models we have implemented so far, and for this reason we decided to create a data structure to meet the challenge.

From the outset, our aim was to make the code as readable as possible. For this, we considered different ways of creating a structure that would contain, for each of the up to 5 Gaussians, the necessary mean, deviation and weight values. After sketching several methods using one object per gaussian and playing with the use of struct we decided that the most efficient and easy to understand would be to create a new class called GaussiansMM. This class houses a vector of Mat matrices for the parameters needed to perform the pixel by pixel operations as well as auxiliary matrices to facilitate the calculation and readability of the code. In addition to these variables, the class hosts, apart from the constructor that initialises the structure, 4 private functions, 3 pubic functions and 4 getters. With hindsight, we think that it would have been much better to apply some design pattern to simplify and improve the code.

As shown in the Listing I containig the class fields, a set of three vectors of Mats is placed to keep all the parametersof the K Gaussians.

```
//Store gaussian parameters for each pixel
vector<Mat> _meanGMM;
vector<Mat> _sigmaGMM;
vector<Mat> _weightsGMM;

Mat _Ngaussians;//N active Gaussians of pixel
Mat _ForegroundGMM; //Resulting mask
Mat _GaussLessWeight;//Index least weight Gaussian
```

Listing 1. Variables of class GaussiansMM.

This object is manipulated it created through the series of functions listed bellow:

1) gaussianMM: This is the constructor of the "gaussianMM" class. It initializes the Gaussian parameters with the selected mode (uniform, random, tailored... only uniform is impelmented). The other parameters specify the number of Gaussian components (K), the mean an deviation updating rate (alpha), the initial standard deviation assign to new Gaussian, and the weight delta ratio applied when a gaussian is erased to place a newcomer.

2) update_GMM_mask: updates the foreground mask based on the current Gaussian parameters. It has as input parameter the threshold needed to trigger the detection.

3) getforegroundGMM: returns the foreground mask that was calculated by update_GMM_mask().

4) update$_w eights(private)$: updates the weights of selected and non-selected Gaussian distributions at a particular pixel location (i,j) for the c-th color channel.

5) is_pixel_in_gaussian(private): checks whether the given pixel value at the pixel location (i,j) for the c-th color channel belongs to the g-th Gaussian distribution.

6) refine_gaussian(private): updates the parameters (mean and variance) of the g-th Gaussian distribution to better fit the given pixel value at the pixel location (i,j) for the c-th color channel.

7) normalize_weights(private) normalizes the weights of all the Gaussian distributions at a particular pixel location (i,j) for the c-th color channel.:

8) update_params: updates the mean, deviation, and weights of all the Gaussian distributions based on the given frame and a weight ratio between old and new Gaussian distributions.This is the most complex function and the one that draws on all the declared private functions. This function completes all the logic for indexing and updating the structure, destroying and creating Gaussian and getting everything ready for the generation of the segmentation mask.

V. DATA

For the evaluation and comparison of the models, 5 video sequences from the baseline of the ChangeDetect Challenge 2012 are used. These videos present very different situations with a set of different challenges:

- PETS2006: dynamic background, busy area, noisy image, low resolution...
- Highway: dynamic background, fast variations, small shadows..
- Office: flickering light source, noisy images,..
- Pedestrians: slightly overexposure, large shadows, dynamic background,

VI. RESULTS AND ANALYSIS

A. Task 2: Comparaison of our basic difference method

The results of this part are summarized in the Table I. As can be observed, Zsófia's can detect more true positive events (higher recall, actual movements) than Juanma's. However, it also means that Zsofia has a higher false negative rate, which indicates that it may miss some actual movements. Both detectors have high specificity rates, which means that they can accurately identify true negative events. Overall, Zsófia's detector has better performance compared to Juanma detector. However, it's important to note that the results may vary depending on the specific parameters used in the algorithm, and further analysis and tuning may be required to optimize the performance.

TABLE I

OVERALL RESULTS COMPARAISON OF OUR DIFFERENT APPROACHES.

Parameter	Zsófia $\tau = 25$ $\alpha = 0.15$ $Th_{ghost} = 50$	Juanma $\tau = 50$ $\alpha = 0.13$ $Th_{ghost} = 50$
Recall	**0.6**	0.52
Specifity	0.97	**0.98**
Precission	**0.55**	0.39
Overall	**0.74**	0.67

B. Task 3: Evaluation the suppression of shadows method

Once the shadow suppression module has been implemented, the evaluation of the ChangeDetect Challenge has been run with several parameters. As can be seen in Table II the number of Shadow Errors detected by the evaluation script decreases considerably in those sequences where the shadow ground thruth is available. Despite this, for certain parameter settings, the overall performance has been found to worsen. In this sense, it is key to understand the difficulty of shadow detection at certain frequencies and to evaluate, for each scenario, whether it is really worthwhile to eliminate from the original model those parts that are detected as shadows.

TABLE II

SHADOW SUPPRESION PERFORMANCE RESULTS COMPARAISON FOR DIFFERENT CONFIGURATION SETTINGS.

Shadow Configuration				***ChangeDetect Results***		
α	β	τ_s	τ_h	**Overall Performance**	**Precision**	**Shadow Error**
No shadow suppression				0.74	0.55	**373629**
0.7	0.9	80	70	0.67	0.51	**348375**
0.5	0.9	80	100	0.6	0.32	**358375**

C. Task 4: Evaluation of the Single Gaussian advanced background subtraction algorithm

Our attempt to implement the algorithm as presented in the lecture slides did not produce the anticipated outcome. Upon further investigation, we discovered that the issue lay with the background mask, which significantly impacted the results we obtained. The noise present in the frame was interfering with the selection of the background, and consequently, the foreground as well. However the moving people and objects are detected as foreground and the shadow suppression is able to detect some parts of the shadow. One example of the noise is shown in the Figure 6.

Fig. 6. Noise in advanced background subtraction algorithm.

The Table III shows the performance metrics of a single Gaussian background segmentation algorithm, which is used to identify foreground pixels in an image. The algorithm achieves high specificity, with 99% of the background pixels correctly identified, but poor recall, with only 16% of the foreground pixels correctly identified. The false negative rate is high at 83%, meaning that many foreground pixels have been misclassified as background pixels. The false positive rate is low, at 0.6%, indicating that the algorithm is effective at identifying background pixels. The probability of background classification is high, at 4.01, suggesting that the algorithm is biased towards identifying background pixels. The precision is low, at 36.1%, indicating that many of the pixels identified as foreground are actually background pixels. The F-measure is also low, at 0.2, which suggests that the algorithm is not very effective in identifying foreground pixels. The standard error, or shadow error, is very high, indicating that the estimates of the performance metrics may not be very reliable. Overall, these findings suggest that the single Gaussian background segmentation algorithm may not be the most effective method for identifying foreground pixels in an image.

D. Task 5: Evaluation of the GMM advanced background subtraction

The implementation of the GMM advanced background subtraction was attempted in this lab . However, during testing, it was discovered that the code did not run due to a memory allocation issue. Specifically, a vector of OpenCV Mats with double data types was created, which caused collisions in the data structure during runtime. Despite this issue, the implementation of the algorithm was promising, and future work will be necessary to address the memory allocation problem and fully test the accuracy and effectiveness of the Gaussian Mixture Model for background segmentation based on movement.

E. Overall Evaluation

TABLE III

OVERALL RESULTS FOR THE OPTIMAL PARAMETERS OFTHE SINGLE GAUSSIAN BACKGROUND SUBSTRACTION (TASK4).

Recall	Specificity	FPR	FNR	PBC	Precision	FMeasure	SE
0.16	0.99	0.006	0.83	4.01	0.361	0.2	159047

Finally, all the results are shown in the Table IV. As can be seen, the results are not optimal but they are as expected considering the observed performance. In the case of the Shadow Error, this number is much lower in the Single Gaussian due to the high trheshold chosen in order not to obtain a very noisy output.

TABLE IV

OVERALL RESULT COMPARAISON FOR THE OPTIMAL PARAMETERS OF ALL THE IMPLEMENTED MODELS.

	Task2 Base	Task3 Shadow	Task4 SG	Task5 GMM
Recall	**0.60**	0.34	0.16	–
Specificity	0.98	0.97	0.99	–
FPR	0.014	0.02	0.006	–
FNR	0.39	0.65	0.83	–
PBC	3.31	4.48	4.01	–
Precision	**0.55**	0.3	0.361	–
FMeasure	**0.53**	0.3	0.2	–
SE	373629	346259	**159047**	–

VII. CONCLUSIONS

In this laboratory, different models of motion detection have been studied, implemented and tested. During the development and study we have gained a deep understanding of the internal mechanisms that make these systems possible and have been able to understand the limitations and the importance of the design choices.

In the evaluation of results we have discovered how the evaluation of a real Video Sequence Analysis competition works. These metrics have helped us to understand our results and to objectively identify areas for improvement in our detectors.

In the part of background modelling using Gaussians we have discovered the importance of the types of variables when performing the calculations as we have had to create structures with a large number of dimensions on which to operate. Although the result has not been the expected one, we have learned from our mistakes and we have solved, as far as the time available has allowed us, most of the proposed challenges.

As for the last task of this laboratory, we have not been able to obtain results because we took the wrong path when declaring the data structure, which made them terribly unstable when indexing them and caused collisions in the memory space during run-time.

VIII. TIME LOG

The time invested for this Lab is displayed in the following Table V;

TABLE V

TIME LOG FOR THIS LABORATORY ASSIGNMENT.

Date	***Description***	***Zsófia***	***Juanma***
28/02/2023	Evaluate algorithm using the ChangeDetection datasets}	2	2
02/03/2023	Coding and debugging the Task 3.2	4	2
03/03/2023	Debugging the Task 3.3	3.5	3.5
07/03/2023	Task 3.3 Debugging	8	5
08/03/2023	Task 3.4	10	2
08/03/2023	Task 3.5	2	10
08/03/2023	Report Writting	5	7
		$\sum = 34.5$	$\sum = 34.5$

REFERENCES

[1] ChangeDetection dataset 2012, "Changedetection.net," changedetection.net. [Online]. Available: http://changedetection.net/. [Accessed: 28-Feb-2023].

Fig. 7. Zsófia(left) and Juanma(right) at Mirador del Nen de la Rutlla. Special mention should be made of the fact that part of this lab was (or was not) carried out in Barcelona..