Foreground Segmentation

María Fernanda Herrera and David Savary

I. INTRODUCTION

The objective of this report is to document the implementation and results of different foreground segmentation techniques. Basic foreground segmentation was implemented using a simple background model, background update and stationary object supression. Advanced foreground methods include using single gaussian and gaussian mixture models to define the background model and background update. A module for shadow supression according to image chromacity was also implemented. These methods were evaluated using the "*VideoSlidesAVSA*" dataset and the Baseline, Shadow and Dynamic Background categories from the "*changedetection*" dataset [4].

II. METHOD AND IMPLEMENTATION

Different implemented methods were developed during this project including a basic segmentation method, Single Gaussian foreground segmentation and segmentation through Gaussian Mixture background model. As a result, three different versions of the algorithm were obtained. All of the implemented functionalities are able to process grayscale and RGB images, except for the Gaussian Mixture Models method that only works with grayscale images. Each of the versions defines a series of modules in a similar way with respect to the order of steps required to perform a foreground segmentation. These steps include the initialization of background, background subtraction and background update. However, there are specific modules that depend on the version that is used. In this way, the basic segmentation algorithm contains an additional module for suppression of stationary objects, while the basic segmentation algorithm and the algorithm that uses single gaussian background model contain an additional module for removing shadows. Shadow removal module has not been implemented within the version corresponding to Gaussian Mixture background model because it requires an implementation in the three RGB channels.

A. BASIC FOREGROUND SEGMENTATION

The first implemented version was the basic foreground segmentation. For this version, the different modules were implemented as follows:.

1) Background initialization: init_bkg: .

This module is used to initialize the background model. To achieve this, the function takes a frame as a single input argument and stores the given frame as the background model. Since the first background model normally correspond to the first frame of the video, the function is called immediately after reading the first frame of the sequence and it is only called once per video sequence. Besides, this function initializes a counter that is used for stationary object supression function.

2) Background subtraction: bkgSubtraction: .

This module obtains an initial foreground mask by comparing all the pixels of the current frame with the background model and measuring if the difference is larger than a given parameter (τ). To achieve this, the function takes a frame as a single input argument and stores it as the current frame. This function is called once for every frame in the video sequence.

3) Stationary Object Suppression: ghostSupression: .

The next step of the algorithm defines a function that is able to remove stationary objects. This category refers to objects that used to be moving objects, but have been in the same position for a predefined period of time and therefore should no longer be classified as moving objects. This is achieved by modifying the background model according to a counter that measures for how long an object has been classified as foreground.

This function does not consider any input argument since it is based on the variables that have already been initialized, including a counter defined in background initialization, the foreground mask obtained from background subtraction and a "*threshold_ghosts2*" defined within the constructor object. The "*ghostSupression*" function increases the counter for each pixel that is classified as foreground in the image, and resets the counters for each pixel classified as background. Finally, every pixel that surpasses the "*threshold_ghosts2*" updates the value of the background model by replacing the specified pixel with the value of the stored frame.

4) Shadows suppression: removeShadows: .

.

This module is able to remove the shadows from the foreground mask, transforming both the stored frame and the background model to the HSV color space, as described in [1]. A pixel is marked as a shadow and removed from the foreground mask if it satisfies the equations (1).

$$
\begin{aligned}
\alpha \leq \frac{BV_t[x,y]}{IV_t[x,y]} \leq \beta \wedge \|IS_t[x,y]-BS_t[x,y]\| \leq \tau_S \wedge D_H \leq \tau_H \\
D_H = min(\|IH_t[x,y]-BH_t[x,y]\|, 360-\|IH_t[x,y]-BH_t[x,y]\|) \\
Image_t[x,y] = (IH_t[x,y], IS_t[x,y], IV_t[x,y] \\
BKG_t[x,y] = (BH_t[x,y], BS_t[x,y], BV_t[x,y]
\end{aligned}
\tag{1}
$$

This function consider five input arguments that correspond to α, β, τ_S or saturation threshold and τ_H or hue threshold and an additional parameter that corresponds to a flag allowing or not to apply shadow removal.

5) Background update: bkgUpdate: .

.

The last step of one iteration correspond to the background update, which adapts the background model to accommodate progressive variations, such as changes of lighting. This is performed according to two different modalities defined as blind update and selective update. Both metodologies are defined by the equation (2), while the difference among them is the pixels that are selected or not to be updated. For the case of blind update, all the pixels in the background model are updated, while for the selective update method, only pixels classified as background are updated.

$$BKG_{t+i}[x,y] = \alpha Image_t[x,y] + (1-\alpha)BKG_t[x,y] \quad (2)$$

This function does not consider any input argument since it is based on the stored frame, background model and α parameter defined within the constructor object.

B. SINGLE GAUSSIAN FOREGROUND SEGMENTATION

Two robust background models were also implemented in later versions of the algorithm. The first one was an Unimodal Gaussian model [2], which was implemented as follows:

1) Background initialization: init_bkg: .
The background model is defined by generating two matrices filled with one mean and one standard deviation value per pixel. Thus, the background initialization module was modified accordingly. To achieve this, mean matrix was initialized according to the values of the first frame within the video sequence and standard deviation matrix was initialized to a high value which will be updated automatically according to the scene variations.

2) Background subtraction: bkgSubtraction: .
This module obtains an initial foreground mask by comparing all the pixels of the current frame with the background model and measuring if the difference is larger than three times its corresponding standard deviation value. Features such as input arguments and function result remain the same as in the basic foreground segmentation.

3) Shadows suppression: removeShadows: .
This function shows the same characteristics as in the basic foreground segmentation version, except for the fact that the matrix considered as background model for the HSV frame and HSV background comparison correspond to the mean matrix.

4) Background update: bkgUpdate: .
The function is also defined according to blind and selective methods as in the basic foreground segmentation. However, two different equations are used for updating values as shown in (3), since both mean and standard deviation values are required to be updated. Other characteristics of the function remain the same as in the basic foreground segmentation.

$$\begin{aligned} \mu_{t+1}[x,y] &= \alpha Image_t[x,y] + (1-\alpha)\mu_t[x,y] \\ \sigma^2_{t+1}[x,y] &= \alpha(Image_t[x,y] - \mu_t[x,y])^2 + (1-\alpha)\sigma^2_t[x,y] \end{aligned} \quad (3)$$

C. GAUSSIAN MIXTURE MODELS FOREGROUND SEGMENTATION

The second robust background model implemented in the algorithm was the Gaussian Mixture Model [3]. This model is similar to the Single Gaussian model, but is able to cope with background flickering, new or removed objects in the background, progressive background changes and image noise by storing multiple Gaussian distributions for every pixel. This model was implemented as follows:

1) Background initialization: init_bkg: .
The background model is defined by generating three gaussians per pixel value in the grayscale image. This is implemented by initializing one mean matrix, one standard deviation matrix and one weight matrix per gaussian. The first gaussian is initialized using the first frame values as mean values, a standard deviation of 10 per pixel and 0.5 weight per pixel. For the other two gaussians, mean values are set randomly, a standard deviation of 30 per pixel and 0.25 weight per pixel.

2) Background subtraction: bkgSubtraction: .
First, the module stores the current frame and orders the three gaussian models according to their weight using bubble sort algorithm. After, an initial foreground mask is obtained by comparing all the pixels of the current frame with the background model and measuring if the difference is larger than 2.5 times its corresponding standard deviation value. Then, the different segmentation masks are ORed to obtain the foreground mask. Two additional masks are generated during this stage in order to define which gaussians are going to be updated in the next module. The first mask is defined by selecting only the gaussian with heighest weight that fulfills the standard deviation criterion. The second mask is defined by returning the upper gaussians according to their weights and a threshold defined within the constructor object. Finally, both masks are ANDed.

3) Background update: bkgUpdate: .
This module is implemented in two stages.The first stage involves updating values of selected gaussians using the same equation defined for unimodal gaussian method. Besides, the weight of selected gaussians is increased, while the weight of non selected gaussians is decreased. In the second stage,

Dataset	Purpose	Video	Characteristic
VideoSlidesAVSA	General	Hall Empty office Stationary Object EPS Shadows EPS Hotstart	Colour camouflage Illumination change Ghosts Shadows Hotstart
Baseline	General	Highway Office Pedestrians PETS2006	Shadows, tree leafs Static foreground Shadows Shadows, reflections
Shadow	Shadow detection	Backdoor Bungalows BusStation CopyMachine Cubicle PeopleInSHade	Shadows Shadows Shadows Shadows Shadows Shadows
DynamicBackground	Flikering filtering	Boats Canoe Fall Fountain01 Fountain02 Overpass	Moving water Moving water Moving leafs Moving water Moving water Moving leafs

TABLE I

VIDEO SEQUENCES USED TO ANALYZE THE PERFORMANCE OF THE ALGORITHM.

gaussians with lower weight and classified as foreground are assigned the current frame value.

III. RUNNING THE CODE

To run any of the versions of the algorithm, it is necessaary to compile the code in a linux machine with opencv installed by using the makefiles. These makefiles will generate a "*Lab1.0AVSA2020*" executable that does not require any argument to run. Parameters and flags that can be modified before compiling the code varies according to the different versions. Finally, the paths of the videos and the results folder can also be modified within the "*Lab1.0AVSA2020.cpp*" file.

1) Basic foreground segmentation: .
Threshold τ is defined for measuring degree of difference between background subtraction and frame. Threshold α is defined for background update rate. Threshold "*threshold_ghosts2*" is defined for stationary object suppression rate. Flag named "*selective_bkg_update*" is used to select among blind and selective background update. Flag named "*RGB*" is used to select among RGB and grayscale methods. Other parameters to consider incude the threshold values defined for shadow supression.

2) Single Gaussian foreground segmentation: .
Threshold α, "*selective_bkg_update*" flag, "*RGB*" flag and thresholds for shadow supression are defined as before.

3) Gaussian mixture models foreground segmentation: .
Threshold α and "*RGB*" flag are defined as before. Threshold τ is defined fo selecting upper gaussians.

IV. DATA

The different versions of the algorithm were tested against four datasets. Each dataset is designed to either test the performance of the algorithm in a general way, or to test one specific functionality of the algorithm, such as shadow removal or the performance with dynamic backgrounds and flickering pixels. All of the used video sequences are described in the table (I). The "*baseline*", "*shadow*" and "*dynamicBackground*" datasets are categories from the "*ChallengeDetection 2012*" dataset [4].

Fig. 1. Mask extracted from the video after an ilumination change.

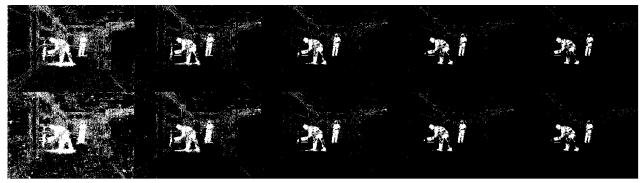

Fig. 2. Comparison between the masks extracted with low and high τ values for both grayscale (top images) and RGB (bottom images) frames. The range of τ goes from 10 in the left column to 50 in the right column

V. RESULTS AND ANALYSIS

In order to evaluate the performance of the algorithm, the simpler versions were tested on a general dataset to obtain a rough idea of the capabilities of the code, meanwhile, the versions that aim to improve a specific problem of the algorithm were tested on specific datasets that suit that problem.

A. SUBJECTIVE VIDEOSlIDESAVSA DATASET ANALYSIS

The first step in the evaluation of the algorithm was to subjectivly test the basic foregrund segmentation by visually comparing the obtained results and study the effect of each parameter with the "*VideoSlidesAVSA*" dataset.

1) Background subtraction: .
The background subtraction technique works well as long as there are no illumination changes or foreground objects that also appear on the first frame (hotstart). When an illumination change occurs, all of the pixels are detected as foreground in the foreground mask, like the Fig. 1 shows. The shadows and objects that are removed from or incorporated to the background are always classified as foreground in the foreground mask too. Tuning the τ parameter is a matter of finding a good balance between noise reduction and split objects due to color camouflage. The Fig. 2 shows the difference between a low and high τ threshold both with grayscale and RGB processing (which does not improve the results in a significant manner). It was found that the optimal result was achieved with a τ value close to 40.

2) Background update: .
The progressive background update was mainly tested using the "*empty office*" video, which has a noticeable luminosity change throughout the hole sequence and no foreground. The

Fig. 3. The image in the middle shows two objects although they have been still for a few seconds. The image in the right was generated from the same frame using the "*ghostSupression*" function.

minimum α value in order to achieve an empty foreground mask in all of the frames of the "*empty office*" sequence was 0.03. It is important to find the minimum working value in order to minimize the effect of the technique when it is not needed. Updating the background has little to no negative effect in the foreground masks unless an unnecessary high threshold is selected, in which case some artifacts appear in the masks. If the selective update flag is not set to true, some objects tend to leave a trace when they move or to become empty if they stay relatively still.

3) Stationary Object Suppression: .
The reference for tuning the stationary object suppression was the "*stationary object*" video. There are multiple objects that are moved during that sequence, so the goal was to eliminate those objects if they stay stationary for a while. It was complicated to find a good balance between removing those objects fast and not to introduce artifacts when objects move slow enough to be incorporated to the background image. The selected value for the "*threshold_ghosts_2*" was 85 in order to eliminate the moving objects of the reference video as shown in Fig. 3. The stationary object suppression is also able to correct the hot start of the background in the "*EPS Hotstart*" video.

B. CHANGE DETECTION DATASET ANALISYS

The evaluations of the different versions of the algorithm were performed by taking into account three main classification metrics. This metrics are the precision or percentage of ground truth foreground that was detected, the recall or percentage of detected foreground that is ground truth foreground, and the Fscore, which is obtained from the precision and recall values.

1) Basic foreground segmentation: baseline category: .

The different parameters that can be modified for this version of the algorithm were tested in different combinations that include working in grayscale or RGB colorspace, updating the background blindly or selectively, α values varying from between 0.05, 0.1 and 0.15, and stationary object suppression threshold varying from 18 to 40. From the different combinations, the use of selective update, RGB colorspace, α equal to 0.05, segmentation threshold τ equal to 20 and stationary object suppression threshold equal to 40 is considered the combination with the best results overall. The metrics obtained correspond to 0.76 for recall, 0.75 for precision and 0.74 for f-score. A few samples of the results achieved by applying such parameters are displayed in Fig. 4.

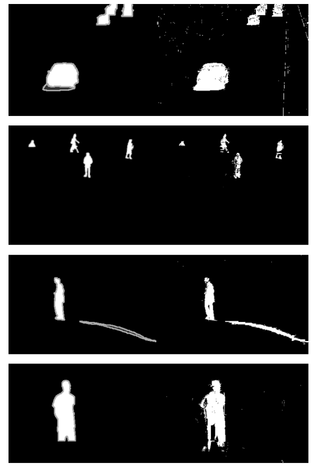

Fig. 4. Results obtained in different video sequences from baseline category, one category per row. On the left side, ground truth and corresponding segmentation result on the right side.

2) Basic foreground segmentation: shadow category: .

The different parameters that can be modified for this version of the algorithm were tested in different combinations divided into to groups. First, the parameters that correspond to background subtraction and background update were kept constant to the following values: τ set to 18, α set to 0.05, selective update activated, stationary object supression set to 30 and RGB colorspace selected. At the same time, parameters that directly affect shadow removal were modified. The parameters α and β were kept constant to 0.5 and 0.9, while τ_H and τ_S were modified with values from 20 to 80. After this test, the best values obtained for foreground segmentation according to specified metrics are α equal to 0.5, β equal to 0.9, τ_H equal to 40 and τ_S equal to 20. The F-score achieved using this values was 0.67. As a second stage, the obtained shadow parameters were kept fixed and suppression of stationary objects threshold was varied from 25 to 80. It was possible to observe that the F-score measure, as well as recall and precision increased by increased the previously mentioned threshold. The highest F-score measure obtained was 0.72 when the value of the stationary object threshold was set to 80. An explanation for this improvement may be that the dataset shows dynamic objects that are moving most of the time within the scene. However, they may be removed too fast and almost completely in a really small period of time. Giving this effect, a large amount of pixels is removed from the overall evaluation, which drastically penalizes the selected metrics. Another situation that is important to highlight is that different types of scenes may require different parameters values since shadows were appropriately removed in some scenes, but were also not detected in some other scenes. A sample of a good results achieved by applying such parameters are displayed in Fig.

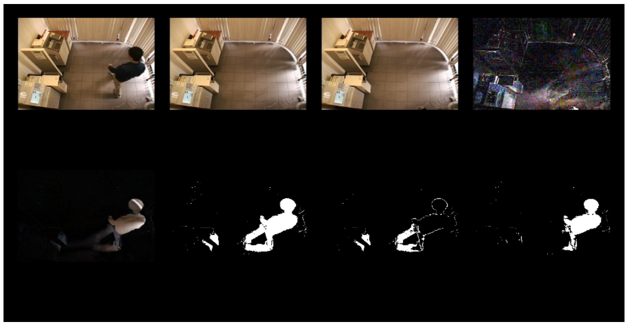

Fig. 5. From left to right in the first row: current frame, background model, initial background model and difference between initial and current background model. From left to right in the second row: Difference between current frame and background, initial background mask, shadow mask and updated background mask.

Fig. 6. Comparison between the masks extracted with different α values for RGB frames and selective background update. The range of α values is 0.01, 0.05, 0.07, 0.09 and 0.11 from left to right. Ground truth is shown as the last image.

5.

3) Single gaussian foreground segmentation: dynamic background category: .

The different parameters that can be modified for this version of the algorithm were tested in different combinations that include working in grayscale or RGB color space, updating the background blindly or selectively, and α values varying from 0.01 to 0.11. From the different combinations, the use of selective update, RGB scale and α equal to 0.05 is considered the combination with the best results overall. The metrics obtained correspond to 0.59 for recall, 0.39 for precision and 0.43 for f-score. Other combinations of values showed that increasing the value of alpha results in a better recall value, but decreases the precision value dramatically. As a result, F- score is also decreased. This indicates that a higher number of α allows to recover most of the foreground, but in turn, introduces noise in the segmentation, as can be observed in Fig. 6. It was also possible to observe that using methods that include RGB colorspace, as well as selective update improve segmentation when alpha is kept constant, which is demonstrated in Fig. 7.

Fig. 7. Comparison between the masks extracted varying flags and α set to 0.05. From left to right the following is shown: grayscale and blind update, RGB colorspace and blind update, grayscale and selective update, RGB colorspace and selective update. Ground truth is shown as the last image

Fig. 8. Comparison between the masks extracted varying α from left to right as 0.01, 0.05 and 0.09 and setting upper gaussians threshold to 0.9. Ground truth is shown as the last image

4) Gaussian mixture models foreground segmentation: dynamic background category: .

The different parameters that can be modified for this version of the algorithm were tested in different combinations that include working upper gaussians threshold varying among 0.7, 0.8 and 0.9, as well as α values varying from 0.01 to 0.01. From the different combinations, the use of upper gaussians threshold set to 0.9 and α equal to 0.05 is considered the combination with the best results overall. The metrics obtained correspond to 0.52 for recall, 0.42 for precision and 0.42 for f-score. Other combinations of values showed that increasing the value of α results in higher recall, but lower precision and lower F-score. On the opposite, decreasing the values of α may reduce or increase precision and recall, but always decreases F-score. A sample of the results may be observed in Fig. 8. , where upper gaussians thresholds was set to 0.9, while alpha values are 0.01, 0.05 and 0.09 from left to right and the ground truth as the last image. In this case, both low and high alpha values introduces noise, with alpha set to 0.05 being the best result. However, increasing upper gaussians threshold for a fixed value of α achieves only a small improvement in F-score metric, that may not be visually observed.

VI. CONCLUSIONS

In general, the implemented techniques performed accordingly to our expectations, although it is extremely complicated to tune the parameters in order to perform well in different situations. In the case of ghost suppression, it is very useful in the case of needing to incorporate still objects to the background or deal with hotstarts, but negatively affects the segmentation when people move slowly or perpendicular to the image plane. Similarly, if the shadow suppression is tuned for one specific scene, it can sometimes perform poorly on the rest of the scenes. If tuned to reduce the overall error in all scenes, the shadow suppressor will not produce as good results. Lastly, the advanced foreground models do a good job in reducing the amount of noise in the foreground mask for dynamic backgrounds as water or moving tree leafs compared to the simpler background subtraction method.

VII. TIME LOG

A. *Maria Fernanda Herrera Perez*

1) Algorithm development at its first version (Basic foreground segmentation): : 4 hours.

2) Algorithm development at its second version (Single gaussian foreground segmentation): : 1 hour.

3) Algorithm development at its third version (Gaussian mixture models foreground segmentation): : 6 hours.

4) Algorithm evaluation at its first version with AVSA dataset (Basic foreground segmentation): : 2 hours.

5) Algorithm evaluation at its first version with Change Detection dataset(Basic foreground segmentation): : 5 hours.

6) Algorithm evaluation at its first version with Change Detection dataset (Basic foreground segmentation and shadow supression): : 5 hours.

7) Algorithm evaluation at its second version with Change Detection dataset (Single gaussian foreground segmentation): : 30 minutes.

8) Algorithm evaluation at its second version with Change Detection dataset (Gaussian mixture models foreground segmentation): : 30 minutes.

9) Result reporting: : 5 hours.

B. David Savary Martinez

1) Algorithm development at its first version (Basic foreground segmentation): : 5 hours.

2) Algorithm development at its second version (Single gaussian foreground segmentation): : 3 hour.

3) Algorithm development at its third version (Gaussian mixture models foreground segmentation): : 1 hour.

4) Algorithm evaluation at its first version with AVSA dataset (Basic foreground segmentation): : 2 hours.

5) Algorithm evaluation at its first version with Change Detection dataset(Basic foreground segmentation): : 1 hours.

6) Algorithm evaluation at its first version with Change Detection dataset (Basic foreground segmentation and shadow suppression): : 2 hours.

7) Algorithm evaluation at its second version with Change Detection dataset (Single gaussian foreground segmentation): : 30 minutes.

8) Algorithm evaluation at its second version with Change Detection dataset (Gaussian mixture models foreground segmentation): : 30 minutes.

9) Result reporting: : 5 hours.

REFERENCES

[1] Cucchiara, Rita Grana, Costantino Piccardi, Massimo Prati, Andrea. (2003). Detecting Moving Objects, Ghosts and Shadows in Video Streams. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 25. 1337- 1342. 10.1109/TPAMI.2003.1233909.

[2] C. R. Wren, A. Azarbayejani, T. Darrell and A. P. Pentland, "Pfinder: real-time tracking of the human body," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 780-785, July 1997, doi: 10.1109/34.598236.

[3] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149), Fort Collins, CO, USA, 1999, pp. 246-252 Vol. 2, doi: 10.1109/CVPR.1999.784637.

[4] www.changedetection.net