Evaluation of Background Subtraction Algorithms using MuHAVi, a Multicamera Human Action Video Dataset

Jorge Sepúlveda  
Departamento de Ingeniería Informática  
Universidad de Santiago de Chile  
Santiago, Chile  
Email: jorge.sepulvedao@usach.cl

Sergio A. Velastín  
Departamento de Ingeniería Informática  
Universidad de Santiago de Chile  
Santiago, Chile  
Email: sergio.velastin@ieee.org

Abstract—MuHAVi is a human action video dataset that provides manually annotated silhouette data for evaluating action recognition methods that are based on silhouettes. The manual silhouette data is intended to separate segmentation difficulties from action recognition performance evaluation. However, the number of annotated images is limited to a small set of sequences and to generate new manually annotated silhouettes is a very complex and costly process. We propose to use state of art segmentation algorithms that automatically generate silhouettes for the whole dataset. This paper addresses the performance evaluation of algorithms that separate foreground and background in the MuHAVi dataset, particularly for methods that create a statistical model of background. It discusses background subtraction methods based in mixture of Gaussians, and presents the metrics commonly used in segmentation evaluation. The best evaluated algorithm is used to automatically generate a set of silhouettes.

I. INTRODUCTION

Human motion analysis is a growing research area in computer vision, that tries to detect, identify and classify different human activities. The application domains in this field include visual surveillance, navigation of autonomous vehicles, action and face recognition, among others. Action recognition typically involves a combination of computer vision and machine learning tools, the vision tools arrange a set of features obtained from a sequence of images and machine learning classifies those features according some learning rules. Detection of human activities based on silhouette, requires segmentation of the human shapes from the background. This process, for example, can be carried out by means of background subtraction methods.

Background subtraction, is a technique that separates foreground from background in a sequence of images. The foreground comprises a set of a representative pixels that defines an object or some region of the image, which particularly can be moving in front of the background. A background subtraction algorithm typically operates at pixel level, labeling each pixel either as background or foreground according the shape of moving object or some other defined property. There are several approaches for segmenting foreground from background. The simplest method consists in modeling background using averages, median, or histograms of the frames over time. Other methods build a statistical model of background pixels, where the pixels are considered foreground if they do not fit in such model. The outcome of these algorithms are the basis for human motion analysis, i.e. a set of binary images (silhouette masks) corresponding to human body doing some activity.

The computer vision community has built several public datasets for testing, evaluating, and comparison of new methods. In the case of action recognition, a dataset mainly provides a set of human actions (e.g. actors performing some human activity), where algorithms can be evaluated in a real environment under different views and illumination conditions. The evaluation methods are mainly based in comparative methodologies, i.e. the binary masks are compared with a reference image or ground truth. In this context, MuHAVi [1] is a dataset of common human activities, with a set of annotated images, which can be used as ground-truth for a comparative evaluation of segmentation methods.

This paper describes a comparative performance evaluation of some background subtraction algorithms, exercised over MuHAVi [1]. Testing was specially focused on algorithms based on a Gaussian Mixture Model. Various evaluation metrics commonly used in computer vision and information retrieval, were combined and applied for performance evaluation. The goal is to find an optimal operating point of these algorithms and to compare each other, with the purpose to generate a full set of silhouettes with the best algorithm. As the results will be representative of a good but imperfect segmentation algorithm, researchers in human action recognition might be able to test the robustness of their methods to segmentation errors. Although this work presents here a relatively small number of methods, the paper provides a baseline against which other researchers can compare work with a more extensive evaluation of other relevant methods.

This paper is organized as follows, section two describes background subtraction methods that use a statistical model of background, specifically as mixture of gaussian components. Section three explains evaluation methods making distinction between evaluations based on discrepancy and visual perception. Section four provides an overview of the MuHAVi dataset,
section five presents experimental results of the tests conducted using the human action sequences with ground truth images, and section six concludes this work.

II. BACKGROUND SUBTRACTION METHODS

This section describes methods for foreground and background segmentation, based on GMM (Gaussian Mixture Models). Different flavors of these methods were selected to perform evaluation tests with the MuHAVi dataset. A complete description of background modeling for foreground detection can be found in a survey prepared by Bouwmans [2]. The study states that GMM has been one of the most referenced methods in the literature.

In a traffic monitoring system, Friedman and Russell [3] proposed to model the background, using a fixed number of Gaussian components. They model pixel value only with three Gaussians: vehicles, shadows, and road (background). However, pixel values which do not adjust any of these three components are not considered part of the model. A similar approximation was done by Stauffer and Grimson [4], but instead of modeling a pixel with a fixed number of Gaussians, they selected $K$ mixtures of Gaussian ($K$ is typically between 3 and 5). Thus, the method could incorporate more than one component for the background model. This method also added two important parameters: learning rate and the pixel proportion factor that might be part of background. Zivkovic and Heijden [5] approached the problem from a Bayesian perspective, selecting in real time a suitable number of mixture Gaussians for a pixel. The algorithm, using a Dirichlet prior distribution, estimates in real time the parameters of mixtures and selects the number of Gaussians. In this case the number $K$ is adapted automatically to each pixel distribution (multi mode), a pixel could be represented by just one or more than one Gaussian components. In an urban traffic monitoring context, Z. Chen [6] improved the model proposed by Zivkovic and Heijden [5], taking into account illumination changes and the shadows cast by cars moving. He proposed an auto-adaptive Gaussian mixture model, introduces a global illumination factor to dynamically modify learning rate and reduce sudden changes in global illumination. He also added a temporal-spatial filter to decrease noise and vibration problems produced by wind on the cameras. Salvadori and Petracca [7] proposed to incorporate the GMM model in micro-controllers. In order to reduce memory use and computational cost, they make an approximation in the precision of integer numbers to overcome the lack of floating point in low cost processors. The updating rules of Gaussian components were also modified as approximations they call Linear and Staircase.

A. General model

The main idea of GMM, is to model the behavior of a pixel over time, using a mixture of Gaussian distribution. The method assumes, the background of the images (no moving objects) presents a regular behavior which can be described by such statistical model. Any unusual event (e.g. a walking person) could be identified through pixels that have changed their values, and will not fit the background model. In this way, the behavior of each pixel is modeled by a probability density function (pdf). The background subtraction process is constantly updating its background model with new images of the sequence, any new pixel would be background only if the intensity value (three values in case of RGB color) is described by the background probability function, otherwise is considered a foreground pixel.

Figure 1 shows one pixel (red spot in both frames) at two different times in a MuHAVi sequence (Kick Camera 3 Person 4). In the first case, the pixel is part of the person playing an action and the pixel assumes the values associated to the color of the actor’s clothing. In the second case, the pixel lies on the stage floor (background), and the value is the color of the stage where the action is being played. Figure 2 shows the histogram of the entire sequence and two fitted Gaussian curves. The background pixel values exhibit low variance, because the stage is relatively stable except for illumination variations.

$$
\hat{p}(x|X_T, BG + FG) = \sum_{k=1}^{K} \hat{\pi}_k \mathcal{N}(x; \hat{\mu}_k, \hat{\Sigma}_k) \quad (1)
$$

$$
\sum_{k=1}^{K} \pi_k = 1 \quad (2)
$$

The estimated density function that describes the behavior of a single pixel over a time adaptation period of $T$ frames ($X_T$), is defined by equation 1 (detailed formulation in [5]). This is a combination of $K$ gaussian components $\mathcal{N}(x; \mu_k, \Sigma_k)$, weighted by factors $\hat{\pi}_k$. Each estimated weight factor $\hat{\pi}_k$ is the degree of influence of each component in the general behavior of the pixel. The algorithm is constantly updating the $K$ estimated values: $\hat{\theta} = \{ \pi_1, ..., \hat{\pi}_k, \hat{\mu}_1, ..., \hat{\mu}_k, \hat{\Sigma}_1, ..., \hat{\Sigma}_k \}$. The parameters describing each pixel are typically estimated
by a Maximum Likelihood method. In general, the background model is estimated by a training set $X_T$, which keeps a recent historical sample of each pixel. Samples contain values that belongs to both, the background (BG) and foreground (FG). The length of the training set is fixed, oldest samples are removed after this has been updated with new images.

### III. PERFORMANCE EVALUATION METRICS

Performance evaluation of background subtraction methods, measures segmentation quality and provides a common criteria to compare general performance of the algorithms. There are several evaluation metrics in the literature, but no general agreement for using a common set of metrics. The measures often mentioned in classification are Precision, Recall and F1-score, all generated by a mix of true/false positive/negative measures (TP, TN, FP, FN). In the same way, a BMC workshop [8] proposed evaluation criteria for evaluating such algorithms, with static quality and visual perception metrics.

In general, the evaluations methods have been classified (depending on the evaluator criteria), as subjective [9] or objective evaluation. Subjective evaluation is based on a group assessment, requiring a large number of the evaluations for statistical relevance. Moreover, objective evaluation (not requiring human intervention) at the same time is divided into analytics and empirical metrics, with analytic needs deep knowledge of algorithms for evaluating complexity and structure and the empirical metrics just measuring the outcome of segmentation compared to some reference or ground truth.

The most common evaluation methods are called empirical discrepancy and are based on comparative measures, mostly comparing the binary mask (foreground) and its ground-truth (reference image). A ground-truth image is also a binary mask, which labels a pixel as background or foreground, providing a set of reference images for comparing segmentation results. Figure 3 shows an example of a ground-truth frame (obtained from WalkTurnBack Camera3 Person1 sequence) and a segmented silhouette mask. The True Positive (TP), in this example, is the number of silhouette pixels (foreground) the system got right. True Negative are the pixels of background image correctly selected. False Positive is the number of background pixels wrongly selected as silhouette and False Negative are the silhouette pixels wrongly classified as background. The result of these pixel measures populate a contingency table to build all base metrics used for performance evaluation.

$$\text{True Positive Rate} = \frac{TP}{TP + FN} \quad (3)$$

$$\text{False Positive Rate} = \frac{FP}{FP + TN} \quad (4)$$

$$F1 - \text{Score} = \frac{1}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}} \quad (5)$$

The true positive Rate is a ratio of silhouette pixels correctly selected and this metric is also called recall or sensitivity. The false positive Rate or false alarm rate is the negatives ratio of background pixels classified incorrectly. The measures of specificity ($1 - FPR$) and sensitivity (recall) provide a general classification index, they illustrate the ability of any algorithm to identify silhouette (sensitivity) and background (specificity) pixels. Both measures define a single point in a ROC [10] curve for comparing performance of different algorithms under the same conditions.

The Matthew Correlation Coefficient (MCC) is a correlation index between reference and result image and provides a number between -1 to 1 to shows difference between reference and results.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (6)$$

A. Receiver Operating Characteristic (ROC)

The receiver operating characteristic curve [10] is a graph to visualize global classification performance of an algorithm; the algorithms classify a pixel as either foreground or background. Any point of this curve is a relationship of classification response between pixels correctly classified and negative pixels incorrectly classified. It describes a relative balance between benefits (true positives) and costs (false positives). The curve helps to compare performance of the algorithms, a curve located near of top left corner presents better performance compared to another one that is further away.

B. Visual perception measures

These approaches try to quantify the visual perception of an independent observer weighing up segmentation results according to the distance of classified pixels (background/foreground) to the edge of the segmented objects. The higher the distance of a misclassified pixel to the border, the higher the weight of the measure. In [11] [12] log functions were used to weight false positive and linear functions for false negatives. In contrast, [13] proposed that false negatives are more relevant for longer distances, then a sigmoid function is used to weight up both FP and FN.

The perceptual measure SSIM (Structural SIMilarity) [14] is a similarity measurement between two images and is based...
on the hypothesis that the human eye gets some structural information from its visual field, defining the concept of structural distortion. This similarity measure is a mix of three factors: correlation loss, average of distortion, and variance of the distortion.

$$SSIM(S,G) = \frac{1}{n} \sum_{i=1}^{n} \frac{(2\mu_{S_i}\mu_{G_i} + c_1) \cdot (2\sigma_{S_iG_i} + c_2)}{\mu_{S_i}^2 + \mu_{G_i}^2 + c_1} \cdot \frac{(\sigma_{S_i}^2 + \sigma_{G_i}^2 + c_2)}{\sigma_{S_i}^2 + \sigma_{G_i}^2 + c_2}$$

The D-Score [15] only considers measurement errors of reference and input image, the error cost is related to the distance of the reference object, the errors are bigger in the intermediate range. This is because such errors could have a bigger effect when recognizing the shape of the objects. In consequence, the near and far errors would have D-Score about zero.

$$D-score(x) = \exp(- (\log_2(2.DT(x)) - \alpha)^2)$$

IV. MuHAVi Dataset

MuHAVi [1] is a video dataset mainly aimed for human action recognition algorithms based in silhouettes. A set of manually annotated silhouettes has been specially prepared for evaluation of these type of methods, providing accurate silhouette data of the video frames. Nevertheless, the manually annotated silhouettes can be also used as ground truth for evaluation of segmentation methods (the purpose of this paper).

All the actions in the dataset were registered with CCTV cameras from distinct observation angles at similar distances from the actors. The cameras are not calibrated. The stage was also illuminated with night street lights simulating surveillance monitoring in an uneven background. The dataset is composed of 17 different classes of human actions, where the first 5 actions are common actions such as run, walk, run, punch, kick, (shotgun) collapse/fall. The rest of the other activities are a bit more complicated from the human point of view, e.g look in a car, drunk walk, jump over gap, etc. Each action is performed by 14 actors, collected by eight cameras located at the 4 sides and 4 corners of a rectangular stage, as shown in figure 4. The dataset contains 8x17x14=1904 video segments in total. The annotated silhouettes subset is composed only of 5 actions, played by 2 actors registered by 2 cameras, i.e 5x2x2=20 actions.

V. Experimental Results

The evaluation of the algorithms is based on comparison of ROC curves. An optimal operation point for each algorithm is chosen by selecting the closest to the top left border point in the ROC curve. The metrics of MCC, MSSIM and D-Score (described in section III) were used as complementary evaluation measures.

Every ROC curve of the algorithms was obtained doing extensive experiments. The methods described in section II depend on a set of parameters, which are mainly the learning rate and threshold (Mahalanobis distance). The learning rate was predefined as 0.001 following [5], with this value the algorithms take roughly 100 frames to compute Gaussian parameters of each pixel. Thereby, an object (silhouette) that is static for around 100 frames becomes part of the background. The threshold was increased systematically within a range of 21 values in order to obtain the best threshold value. A ROC curve for a single action was composed by combination of this fixed parameter (learning rate) and variations of the threshold. This procedure was repeated for all MuHAVi sequences with annotated images (20 actions). Finally, the overall ROC curve for a given algorithm was built up by averaging (threshold averaging [10] procedure) the independent ROC curves obtained in the experiment of each action.

The following GMM algorithms described in section II were used to perform the experiments.

- Mixture of Gaussian (MOG) is the original method described in [5] adapted from a OpenCV and implemented in C++.
- Select Adaptive Gaussian Mixture Model (SAGMM) originally implemented in Matlab [6], ported to OpenCV C++.
- UCV GMM [7] provides two alternative versions to be tested: Linear and Staircase.

Figure 6 shows final ROC curves achieved during the experiments. It can be seen that SAGMM [6] has better overall performance with respect to the other algorithms. The MOG [5] ROC curve, looks similar (same trend) to SAGMM, but lower true positives rate. The changes proposed by SAGMM has improved its global performance regarding MOG [5], this just can be checked by the ROC curve located just above MOG [5] over all false positives range. On the other hand, MOG also has better performance of two versions of UCV [7] in the range of false positive rate lower than 5%. However, UCV [7] staircase version gets better of UCV Linear with false alarms higher than 2% and MOG with a false positive rate in 6%. MOG presents 3% of false alarm errors, and 62% of true positive detections with a threshold of 6. The UCV staircase for the operation point selected shows a similar detection of SAGMM (little bit lower) 74% but higher rate of false alarms 9% and Linear version of UCV has same false alarms of Staircase but a 64% of true positives detection.
The ROC curves in figure 6 show that, as the threshold increases the true and false positive rates decreases, this implies lower detection rate (less silhouette pixels detected) and also lower false alarms (more background pixels) as the threshold value increases. The ideal aim is to get higher positives (silhouettes pixels right selected) and lower false alarms (background pixels detected correctly). Therefore, selecting an operation point is a trade off between these measures.

Regarding performance parameters, Figure 8 shows trends of MCC, F1-Score, MSSIM, and D-Score measures with respect to threshold. The measures of MCC and F1-Score present similar results (in all algorithms), SAGMM, in both curves, reaches better measures at threshold 6. However, this threshold value lies in a detection rate around 62% and 1% in false alarms rate. Note that results of MCC, F1-Score, and MSSIM for SAGMM are much better than the other three algorithms in all threshold range, the curves are located above all. MOG, as well as the ROC curve, follows same trend that SAGMM curves, but in a lower scale. D-score measures also have same behavior and there is not much difference between the algorithms, SAGMM and MOG are better around the threshold range 5 to 10. The result of D-score can be interpreted in the following, despite of errors, these have low cost and not perturb the recognition of silhouette. This can be checked in Figure 7, the shape of silhouette still is possible to recognize despite of errors.

The operation points selected for each algorithm are given in the table I. The optimal operating point selected for SAGMM corresponds to a threshold value of 3.5. For this point, silhouette pixels and error background pixels classified have a performance of 77% and 2% respectively. Looking at SAGMM ROC curve in more detail, it could also have been possible to chose (threshold = 3) as the operation point. In this case the rate of false and true positives would have been 3% and 78%. A small improvement in detecting rate (1%) but deterioration of false alarms. A user must always make a choice between better detection (TP) with more noise (FP) or sacrifice some detection and reduce noise FP.

### VI. Conclusion

This paper has shown evaluation results of background subtraction algorithms based on Gaussian mixture models. The algorithms were evaluated contrasting the segmented foreground mask (silhouettes) with annotated images (used as ground truth) provided by MuHAVi. The comparative results of each sequence action were averaged and consolidated in global performance metrics, these have helped to compare and evaluate the algorithms in MuHAVi. The receiver operation characteristic curve was used to determine an optimal operation point, this selected point is an agreement between good detection in silhouette pixels and false alarms in background pixels.

The experiments have determined that SAGMM has better performance than other similar algorithms, this can be verified looking at its ROC curve above of the other and comparing performance measures in this operating point. A complete set of silhouettes all MuHAVi sequences has been generated using this algorithm configured with operation point found in the experiments and is being made available to the research community.

### ACKNOWLEDGMENT

Sergio A Velastin gratefully acknowledges the Chilean National Science and Technology Council (Conicyt) for its funding under grant CONICYT-Fondecyt Regular no. 1140209 (“OBSERVE”).

### REFERENCES


<table>
<thead>
<tr>
<th>Threshold</th>
<th>SAGMM</th>
<th>STAIRCASE</th>
<th>LINEAR</th>
<th>MOG</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPR SE(%)</td>
<td>3.73</td>
<td>2.50</td>
<td>2.33</td>
<td>6.27</td>
</tr>
<tr>
<td>FPR SE(%)</td>
<td>0.35</td>
<td>1.21</td>
<td>1.80</td>
<td>0.97</td>
</tr>
<tr>
<td>TPR</td>
<td>0.77</td>
<td>0.74</td>
<td>0.64</td>
<td>0.62</td>
</tr>
<tr>
<td>FPR</td>
<td>0.02</td>
<td>0.09</td>
<td>0.09</td>
<td>0.03</td>
</tr>
<tr>
<td>MCC</td>
<td>0.53</td>
<td>0.29</td>
<td>0.26</td>
<td>0.43</td>
</tr>
<tr>
<td>FMeasure</td>
<td>0.51</td>
<td>0.22</td>
<td>0.21</td>
<td>0.42</td>
</tr>
<tr>
<td>DSCORE</td>
<td>6.1e-03</td>
<td>4.9e-03</td>
<td>5.1e-03</td>
<td>5.5e-03</td>
</tr>
<tr>
<td>MSSIM</td>
<td>0.55</td>
<td>0.36</td>
<td>0.37</td>
<td>0.49</td>
</tr>
</tbody>
</table>

TABLE I. PERFORMANCE MEASURES OF OPERATION POINT SELECTED FOR EACH ALGORITHM.
Fig. 7. Example of silhouette masks for each algorithm in selected operation point given in table I.

(a) SAGMM (threshold = 3.5)

(b) MOG (threshold = 6)

(c) UCV Linear (threshold = 5)

(d) UCV Staircase (threshold = 8)

Fig. 8. Performance measures obtained during experiments. It shows results as threshold value was increased.

(a) MCC

(b) F1-Score

(c) MSSIM

(d) D-Score


