Self-Adaptive Gaussian Mixture Models for Real-Time Video Segmentation and Background Subtraction

Nicola Greggio [a,c], Alexandre Bernardino [c], Cecilia Laschi [a], Paolo Dario [a,b], José Santos-Victor [c]

[a] *ARTS Lab - Scuola Superiore S.Anna, Polo S.Anna Valdera*
Viale R. Piaggio, 34 - 56025 Pontedera, Italy

[b] *CRIM Lab - Scuola Superiore S.Anna, Polo S.Anna Valdera*
Viale R. Piaggio, 34 - 56025 Pontedera, Italy

[c] *Instituto de Sistemas e Robótica, Instituto Superior Técnico*
1049-001 Lisboa, Portugal

$nicola.greggio@ieee.org$

***Abstract*— The usage of Gaussian mixture models for video segmentation has been widely adopted. However, the main difficulty arises in choosing the best model complexity. High complex models can describe the scene accurately, but they come with a high computational requirements, too. Low complex models promote segmentation speed, with the drawback of a less exhaustive description. In this paper we propose an algorithm that first learns a description mixture for the first video frames, and then it uses these results as a starting point for the analysis of the further frames. Then, we apply it to a video sequence and show its effectiveness for real-time tracking multiple moving objects. Moreover, we integrated this procedure into a foreground/background subtraction statistical framework. We compare our procedure against the state-of-the-art alternatives, and we show both its initialization efficacy and its improved segmentation performance.**

Index Terms - Real-Time Video Segmentation, Self-Adapting Gaussian Mixtures, Online EM, Background Subtraction.

I. Introduction

Nowadays, the new computer generation enlarged scientist opportunities of using more complex algorithms in image processing. In the past decades, real-time video segmentation has suffered from the low computational power of the last generation machines. Segmentation of single frames resulted too slow or with an acceptable frame rate, but subjected to excessive restrictions. To this aim, the usage of adaptive Gaussian mixture models had become a standard in the recent years, due to their theoretical foundations and analytical representation.

A. *Related Work*

Video segmentation techniques can be focussed on two main categories, mainly: Those based on the content-based video retrieval (CBVR), and those based on the Foreground/Background (F/B) segmentation.

Some relevant exponents of CBVR are [1], [2], which consider a segmentation of pixel volumes in the whole 3D set, analyzing the content in the extended domain by combining the information across all frames, and [3], [4], [5], which perform a frame-by-frame tracking analysis. However, all of these methods suffer of elevated computational complexity, both for the amount of data they consider at each iteration, and for the usual large number of Gaussian needed for a precise segmentation, which together make them too complicated and not suitable for real-time applications.

Foreground/Background (F/B) segmentation has received many efforts in the last decade, due to their less computational burdensome requirements. In 1999, Stauffer and Grimson [6] published what become the standard formulation for the mixture approach in the field. Here, a recursive filter is used to train a mixture background model together with an K-means approximation. However, the adaptation rate depends on a global parameter α, which ranges between $(0, 1)$, fixed experimentally. Moreover, a small amount for α is required, therefore slowing the learning procedure. In 2002 Lee *et Al.* [7] proposed to segment the background in a different way from Stauffer and Grimson [6]. Here, the segmentation problem is decomposed as two independent problems: The first one is to estimate the distribution of all observations at a single pixel, as a Gaussian mixture, and the second one is to estimate the probability that each Gaussian in the mixture constitutes the background. However, the issue of using experimental values for the learning rate still remain unsolved, also for the F/B segmentation. In 2004 Zivkovic [8] proposed a method for adapting the scene to light changes, by adding new samples and discarding the old ones a reasonable time period. A constant α describes an exponentially decaying envelope that is used to limit the influence of the old data. Nevertheless, this approach uses an even higher number of heuristic thresholds featured by the others, while featuring a slow convergence. Finally, in 2005 Lee [9] started from the base of Stauffer and Grimson [6], i.e. considering a recursive filter for training the mixture, introducing a new variable learning rate for each components, as previously suggested by Lee *et Al.* [7] three years before. However, the required computational complexity is still high, together with the number of variables needed to be set in order to perform the computation.

We chose to adopt a (F/B) approach. Nevertheless, a common problem influencing all the above discussed algorithms is their initialization, regarded both as the number of mixture components selection and their initial values. Besides, despite all the efforts made in the last decade to improve the original work of Stauffer and Grimson [6], others problems still remain unsolved, such as the learning rates decision, and the high computational complexity.

B. *Our Contribution*

In this paper we propose an algorithm for real-time video segmentation based of Gaussian mixture models (GMMs), and foreground/background segmentation. Our approach first learns a description mixture that best describes the first video frames, and then it uses these results to describe the further frames. To this aim, we make use of a previous work of ours for the learning initialization procedure [10], [11]. The key point is that instead of starting with fixed *a-priori* set mixture complexity, we learn a proper one automatically. Then, we apply a foreground/background segmentation procedure we derived from the approach of Lee [9].

C. *Outline*

In sec. I-A we describe the current findings in this field. In sec. II we introduce our technique for the mixture initialization. Subsequently, we propose our approach for the video segmentation in sec. III, and for the foreground/background segmentation in sec. IV. In sec. V we describe our validation experiments. After that, in sec. VI we make some final comparisons and considerations regarding the tested approaches. Finally, in sec. VII we conclude.

II. Mixture Initialization and Number of Components Selection

It is worth noticing how none of the algorithms previously described mention some approach for the selection of the initial number of components. For instance, Stauffer and Grimson [6] and Lee [9] sidestepped this problem by using a compromise, i.e. by using always 3, 4, 5 components since the beginning, during all the computation.

We initialize the description mixture by running an algorithm that both learns the mixture distribution while automatically select the best number of components, based on [10] and [11]. However, we will consider a static input data set, not a video sequence, i.e. the first video frame. In this way, we will have a mixture that *best* approaches the video segmentation initialization in some ways. Assuming that the video scenario does not change too much during first stage - the initialization the learned mixture resulting from this stage can be assumed as a valid input for the second stage - the video segmentation. Therefore, a fast algorithm is essential, otherwise we cannot longer assume the variance from the first video frame to the rest of the sequence as negligible. This can be considered analogous to the initial background learning process of [6] and [9] , while some identical frames representing the background are processed in order to learn the mixture before starting the process.

A. *Learning a finite mixture model from static data*

Different approaches for describing a static input data set by means of Gaussian mixtures can be found in literature. Here, the computational complexity is a strict requirement: In fact, the longer the initialization procedure is, the less unaltered the scenario will be, resulting in a mixture no longer describing the video frames with accurately enough. The number of components is what constrains the EM time performance most.

Hence, greedy algorithms seem to be a reasonable choice. They are characterized by making the locally optimal choice at each stage with the hope of finding the global optimum. In 2002 Vlassis and Likas introduced a greedy algorithm for learning Gaussian mixtures [12]. The algorithm starts with a single component covering all the data, while subsequently splitting an element and perform the EM locally for optimizing only the two modified components. Unfortunately, the total complexity for the global search of the element to be split is $O(n^2)$. In 2003, Verbeek *et Al.* developed an improved greedy method, in which the search for the new components is claimed to take $O(n)$ [13]. However, the main problem of greedy algorithms mostly (but not always) is that they fail to find the globally optimal solution, because they usually do not operate exhaustively on all the data.

Another approach is that of Figueiredo [14], which starts with a high number of components much more than necessary, and then progressively reduces it while adapting the remaining ones. However, although demonstrating a good precision, it is very slow, as demonstrated by the comparison in [11].

B. *Our Approach*

We choose to adopt the technique we presented in [11]. This is a greedy algorithm for unsupervised learning a mixture while selecting the best number of components at the same time. Our methodology overcomes the previous cited limitations solving the new component search with a full binary thee that both takes only $O(\log n)$ and exploits the whole possible solutions' set. The structure we use has the particularity that only the leaves contain a mixture component.

The data structure organization is as follows:

- The initial tree starts with the root, only;
- Each node has no children (so that it is a leaf) or two children;
- Only the leaves can contain the mixture components; when a class is inserted by a leaf replication, the latter was the father and now it becomes a child together with the new inserted, creating a new parent without mixture components.
- The nodes eligible for being replicated are those of the last level only.

III. Mixture updating: Video Segmentation

Once the initial mixture has been learned, it is necessary to update it. Despite all the approached techniques, each of them presents several similarities among them. The base algorithm is that of Stauffer and Grimson [6]. Here, an on-line K-means approximation is used against the the exact EM, due to its major speed. None of the approaches present in literature uses the exact EM formulation, while each of them features different techniques for updating the mixture as a recursive filter, considering the three color dimensions as independent (for reducing the components' covariance matrix, and then the whole algorithm burden).

We follow a different approach. In our formulation, we want to balance the accuracy of estimation with the computational complexity in an opposite way. Starting from the assumption that each frame is correlated with the previous one, like the other techniques, an upgrade of the current mixture description rather than a completely new one would be adopted. However, instead of using a recursive filter, we prefer running a fixed number of the EM algorithm iterations. Scene can vary abruptly, therefore instead of waiting until full convergence for a single frame, we adapt the current mixture in order to follow the video. Moreover, the likely lack of accuracy lost by stopping the EM before some iterations can be compensated by both using the EM algorithm and the full covariance matrix for each component, i.e. by considering the statistical dependence among each color dimension. However, despite the assumption that the frame do not chance one from another sensibly, we found experimentally that our approach is robust to sudden changes in the scene, like the appearance of new objects.

The image is then modeled as a Gaussian mixture distribution in the spatial and color space space, i.e. each pixel is represented as $\bar{x}^i \in [x, y, R, G, B]$ or $\bar{x}^i \in [x, y, H, S, V]$, depending on the employed color space, being $\mathcal{X} = \left\{\bar{x}^1, \bar{x}^2, \cdots, \bar{x}^k\right\}$ the input data set for each frame, with k the total number of each frame pixel (supposing this to not change for different frames). We briefly describe next the basic steps of the EM algorithm for the case of Gaussian mixture model. For a single pixel $\bar{x}^i \in \mathbb{R}^d$, with $d = 5$ in our case, the probability of being observed is:

$$p(\bar{x}^i) = \sum_{c=1}^{nc} w_c \cdot p_c(\bar{x}^i) \tag{1}$$

where nc being the number of Gaussian components, and $p_c(\bar{x}^i)$ the probability of the pixel $\bar{x}^i$ given the Gaussian component C_c, expressed as:

$$\begin{aligned} p_c(\bar{x}^i) &= p(\bar{x}^i|C_c) \\ &= \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_c|^{\frac{1}{2}}} \cdot e^{-\frac{1}{2}(\bar{x}^i-\bar{\mu}_c)^T|\Sigma_c|^{-1}(\bar{x}^i-\bar{\mu}_c)} \\ &= \eta\left(\bar{x}^i, \bar{\mu}_c, \Sigma_c\right) \end{aligned} \tag{2}$$

Each Gaussian is described by a parameter set $\bar{\theta} = \{w_c, \bar{\mu}_c, \Sigma_c\}$, where:

- $w_c > 0$, with $\sum_{c=1}^{nc} w_c = 1$, represents the *a-priori* probability of the class C_c;
- $\bar{\mu}_c$ represents the mean of the class C_c;
- Σ_c represents the full covariance matrix of the class C_c.

Given a set of nc feature vectors $g_1, g_2, \cdots, g_{nc}$, the maximum likelihood estimation of $\bar{\theta}$ is:

$$\begin{aligned} \theta_{ML} &= \arg\max_{\bar{\theta}} L\left(\bar{\theta}|g_1, g_2, \cdots, g_{nc}\right) \\ &= \arg\max_{\bar{\theta}} \sum_{c=1}^{nc} \log p_c(\mathcal{X}|\bar{\theta}) \end{aligned} \tag{3}$$

The EM algorithm is proven to converge to an estimation of the optimum (thought local and not global) for $\bar{\theta}$ after a certain number of iterations. Each iteration of the EM algorithm re-estimates the parameter set $\bar{\theta}$, according to the following two steps:

- <u>*E-step:*</u> for each data sample evaluate the probability that the input sample $\bar{x}^i$ belongs to the class c, i.e. that $P\left(C_c = 1|\bar{x}^i\right)$ for each class $c \in [1, nc]$, as:

$$\begin{aligned} P\left(C_c = 1|\bar{x}^i\right) &= P\left(C_c|\bar{x}^i\right) \\ &= \frac{p\left(\bar{x}^i|C_c\right) \cdot P\left(C_c\right)}{p\left(\bar{x}^i\right)} = \frac{w_c \cdot p_c\left(\bar{x}^i\right)}{\sum_{c=1}^{nc} w_c \cdot p_c\left(\bar{x}^i\right)} \\ &\triangleq \pi_c^i \end{aligned} \tag{4}$$

- <u>*M-step:*</u> re-estimate the parameter vector $\bar{\theta}$, which at the $n+1$ iteration will be $\bar{\theta}^{(n+1)}$. This, in case of a gaussians mixture distribution, the means and the covariances are evaluated by weighting each data sample by the degree in which it belongs to the class as:

$$\begin{aligned} \bar{\mu}_c^{(n+1)} &= \frac{\sum_{i=1}^{k} \pi_c^i \bar{x}^i}{\sum_{i=1}^{k} \pi_c^i} \\ \Sigma_c^{(n+1)} &= \frac{\sum_{i=1}^{k} \pi_c^i \left(\bar{x}^i - \bar{\mu}_c^{(n)}\right)\left(\bar{x}^i - \bar{\mu}_c^{(n)}\right)^T}{\sum_{i=1}^{k} \pi_c^i} \end{aligned} \tag{5}$$

Finally, re-estimate the *a-priori* probabilities of the classes, i.e. the probability that the data belongs to the class c as:

$$w_c^{(n+1)} = \frac{1}{k} \sum_{i=1}^{k} \pi_c^i, \quad \mathit{with} \;\; c = \{1, 2, \ldots, nc\} \tag{6}$$

Using EM, the parameters representing the Gaussian mixture are found. In this study we use a fixed number of iterations. Usually the bigger gradient in the mixture components learning occur within the fist iterations, while as long as the EM reaches its convergence only small refinements occur. Then, it is possible to maintain a real-time process, without relying on the unknown convergence time of the original EM, with a negligible lack of accuracy.

A. *Modified EM*

As noted in [14], the EM formulation presents several drawbacks. Particularly, methods that require mixture estimates for various number of components may converge to the boundary of the parameter space. This means that, e.g. for Gaussian mixtures, the covariance matrixes may became singular. A way to address this problem is to use adequate priors for the parameters.

We use the negative Dirichlet prior, because it promotes configurations where the prior probability is either 0 or 1, [14]. Besides, it also speeds-up the components' adaptation process also, due to its "sharped characteristics" with respect to other priors, e.g. the Jeffrey's prior or the minimum entropy prior, therefore leading to a faster convergence.

This results in a modified EM procedure. Following [14], we considered the cost function as a posterior density due to the adopting of improper Dirichlet priors for the classes probabilites [15]:

$$p\left(\vartheta\right) \propto e^{\left(-\frac{N}{2}\sum_{i=1}^{c}\ln w_i\right)} \tag{7}$$

where $w_{i=1,2,\ldots,c}$ are the prior probabilities of each class C_c. Therefore, the EM that minimizes the cost function - with a fixed number of component c - in our case results being [10]:

- *E-step*:

$$\tilde{\pi}_z(t+1) = \frac{\max\left\{\left(\sum_{i=1}^{k}\pi_z^i\right) - \frac{N}{2}\right\}}{\sum_{j=1}^{c}\max\left\{\left(\sum_{i=1}^{k}\pi_j^i\right) - \frac{N}{2}\right\}} \tag{8}$$

  where π_z^k is given by the E-step in eq. (17).
- *M-step*:

$$\tilde{\bar{\vartheta}}_z^{(n+1)} = arg\max_{\bar{\vartheta}_z} Q\left(\bar{\vartheta}^n, \tilde{\bar{\vartheta}}^{(n-1)}\right) \tag{9}$$

both for $z = 1, 2, \ldots, c$, with N being the number of parameters specifying each component.

IV. Background Subtraction

Background segmentation can be interpreted at each single pixel level as a binary classification problem, where a probability function is used to determine how much that pixel belongs to the background or foreground. Considering that being part of the background or the foreground can be assumed complementary for each generic pixels $\bar{x}^i$ at the time t, we can write:

$$p(F|\bar{x}^i) + p\left(B|\bar{x}^i\right) = 1 \tag{10}$$

where F and B are the foreground and background classes, respectively.

The probability $p\left(B|\bar{x}^i\right)$ can be expressed as [9]:

$$p\left(B|\bar{x}^i\right) = \sum_{c=1}^{nc} p(B|C_c)p(C_c|\bar{x}^i) \tag{11}$$

Then, we adopted the same Bayesian derivation as in eq. (4), obtaining:

$$p(C_c|\bar{x}^i) = \frac{p(\bar{x}^i|C_c)p(C_c)}{p(\bar{x}^i)} = \pi_c^i \tag{12}$$

When applied to $p\left(B|\bar{x}^i\right)$, this results in:

$$p\left(B|\bar{x}^i\right) = \sum_{c=1}^{nc}\frac{p(B|C_c)p(\bar{x}^i|C_c)p(C_c)}{p(\bar{x}^i)} \tag{13}$$

Then, being:

$$p(\bar{x}^i) = p(\bar{x}^i|C_c)p(C_c) \tag{14}$$

we have:

$$p\left(B|\bar{x}^i\right) = \frac{\sum_{c=1}^{nc}p(B|C_c)p(\bar{x}^i|C_c)p(C_c)}{\sum_{c=1}^{nc}p(\bar{x}^i|C_c)p(C_c)} \tag{15}$$

There have been proposed different approaches for determining $p(B|C_c)$ in literature. For instance, Friedman and Russel [16] manually labeled the Gaussian components, maintaining them fixed during all the computation. Two years later, Stauffer and Grimson [6] considered to segment the background not as a single pixel formulation, but as a component formulation. They first ordered the Gaussians by the value of $w_c/|\Sigma_c|$, and then they selected the components that represent the background as the first B ones such that $B = \arg\min_b\left(\sum_{c=1}^{nc} w_c > T\right)$, where T is an empirical threshold determined experimentally.

We followed the approach of [9], i.e. training a sigmoid function on $w_c/|\Sigma_c|$ to approximate $p(B|C_c)$ using logistic regression:

$$\hat{p}(B|C_c) = f\left(\frac{w_c}{|\Sigma_c|}, a, b\right) = \frac{1}{1+e^{\left(-a\frac{w_c}{|\Sigma_c|}+b\right)}} \tag{16}$$

Then, we choose to identify the background at each single pixel level as a binary classification problem, i.e. selecting the Gaussian components representing the background as those that:

$$p\left(B|\bar{x}^i\right) > Th \tag{17}$$

Here, Th is still heuristic, although not defined an a fixed default value remaining unaltered during all the computation, e.g. as $T = 0.5$ in [9], but varying on $p\left(B|\bar{x}^i\right)$ at each iteration:

$$Th(\gamma) = \min_c p\left(B|\bar{x}^i\right) + \left(\max_c p\left(B|\bar{x}^i\right) - \min_c p\left(B|\bar{x}^i\right)\right)\cdot\gamma \tag{18}$$

with γ being an experimental value in the range $[0, 1]$. The latter can be interpreted as a decision balance that shifts the threshold Th more toward the $p\left(B|C_c\right)$ lowest values ($\gamma < 0.5$), or higher values ($\gamma > 0.5$), with regard to the $p\left(B|C_c\right)$ average among all the components (i.e. with $Th(\gamma = 0.5)$). This allows to adjust the F/B classification during all the process, adapting to the new Gaussian components automatically. We found that the mere usage of $T = 0.5$ did not work while using our modified EM computation instead of the recursive filter proposed in [9]. In this work we adopted $\gamma = 0.1$ as the best compromise.

V. Experimental Results and Discussion

A. *Experimental set-up*

*1) **Experiments**:* We applied the proposed algorithm to a video sequence registered in a in-door contexts (objects handling and manipulation). This is common situations for e.g. robotics object tracking and grasping. Then, we applied the approaches of [6] and [9] to the same video sequence in order to compare them and validate our approach.

Our first aim is to test whether our technique can perform faster than the reference state of the art approaches [6] and [9], intended as both learning speed (valuable with a log-likelihood faster increase) and higher image segmentation accuracy (valuable by visual images inspection). Then, our second aim is to evaluate a possible hybrid technique, which combines our mixture learning algorithm, FSAEM together with the recursive filters mixture learning procedure in [6] and [9].

In the followings we then illustrate the behavior of our mixture learning algorithm [10], the comparison of the video and F/B segmentation among our approach, [6] and [9], and finally our hybrid technique.

*2) **Parameters selection**:* Unfortunately, [6] and [9] do not have a unique initialization criterion, relying on the user's model complexity selection. Therefore, in order to make the experiments the fairest as possible, we adopt the same number of components detected by FSAEM also for [6] and [9], so that to not have disparities regarding the image segmentation and the whole Log-likelihood due to different mixture models complexity.

Our technique selected a model with 3 Gaussian for the first video, and 4 for the other one. Then, we set the logistic regression parameters in eq. (16) as $a = 20 \cdot 10^7$ and $b = 10$, respectively. The same values have been employed for the algorithm in [9], together with the value of $\gamma = 0.014$ in eq. (18). Our approach does not have other parameters, while [6] and [9] also need to set the learning ruler and the threshold for the matching function, which we set as $\alpha = 0.0001$ and $T_\alpha = 2.5$, respectively.

B. *Mixture Learning Initialization: Image segmentation by means of Gaussian Mixture*

In this section we show the results of the mixture learning initialization on the first video frame. We use a previous work of ours, called FSAEM [10].

Fig. 1 shows some examples of color image segmentation, obtained by processing real images. These represent an in-door and a out-door contexts. For each row, from left to right, there are the source image, the Gaussian segmented one, the Log-likelihood of the whole computation as function of the number of iterations, the cost function as function of the number of iterations, and the same as function of the number of components.

For a more exhaustive FSAEM description together with its application to synthetic distributions see [10].

C. *Video and F/B segmentation*

In this section we compare our F/B segmentation with those of [6] and [9]. As reported by Lee in his original work [9], its improvement with respect to [6] regards the first learning phase, mainly. In fact, it is possible to see how [9] adapts faster to the video. Then, the performances of the two algorithms are quite similar. Fig. 2 shows the Log-likelihood of the three approaches, [6], [9] and our, namely. Our technique is drawn in blue, while [6] in green, and [9] in red.

Fig. 3 shows the results of the comparison. Rows (a) and (b), are composed by the segmentation of the approaches in [6] and [9], respectively, row (c) shows our segmentation technique outcomes, and row (d) represents the original images with our object detection superimposed. It is possible to see how [6] learns slower than [9] in the first frames, while performing equally after the initial stabilization. Nevertheless,

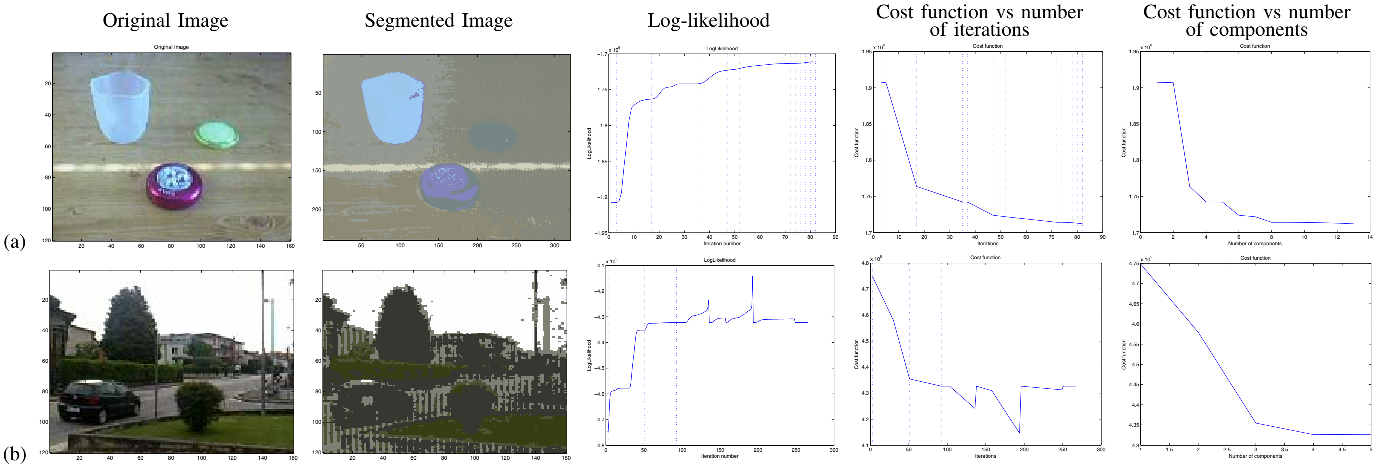

Fig. 1. Image segmentation in the RGB color space. We considered an in-door (a) and a out-door (b) situation. For each image we have: From left to right: the source image, the Gaussian segmented one, the Log-likelihood of the whole computation as function of the number of iterations, the cost function as function of the number of iterations, and the cost function as function of the number of components.

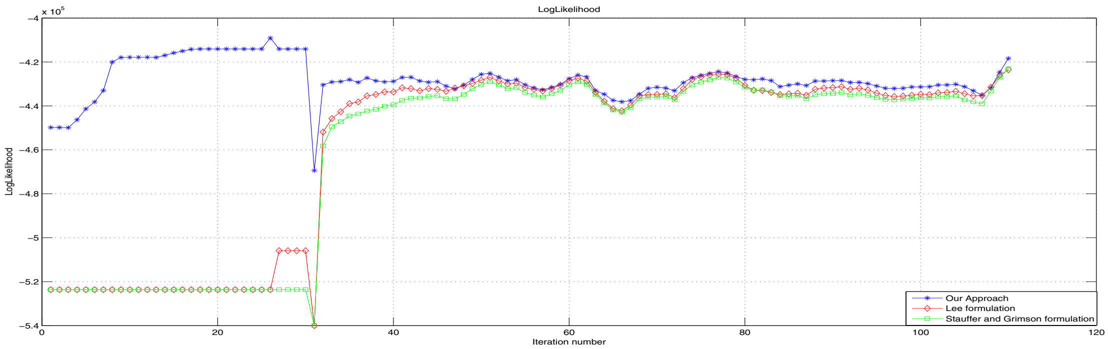

Fig. 2. Log-Likelihood of the objects handling and manipulation video: Comparison among our approach and [6] and [9]. The proposed algorithm is showed in blue, while [6] in green and [9] in red.

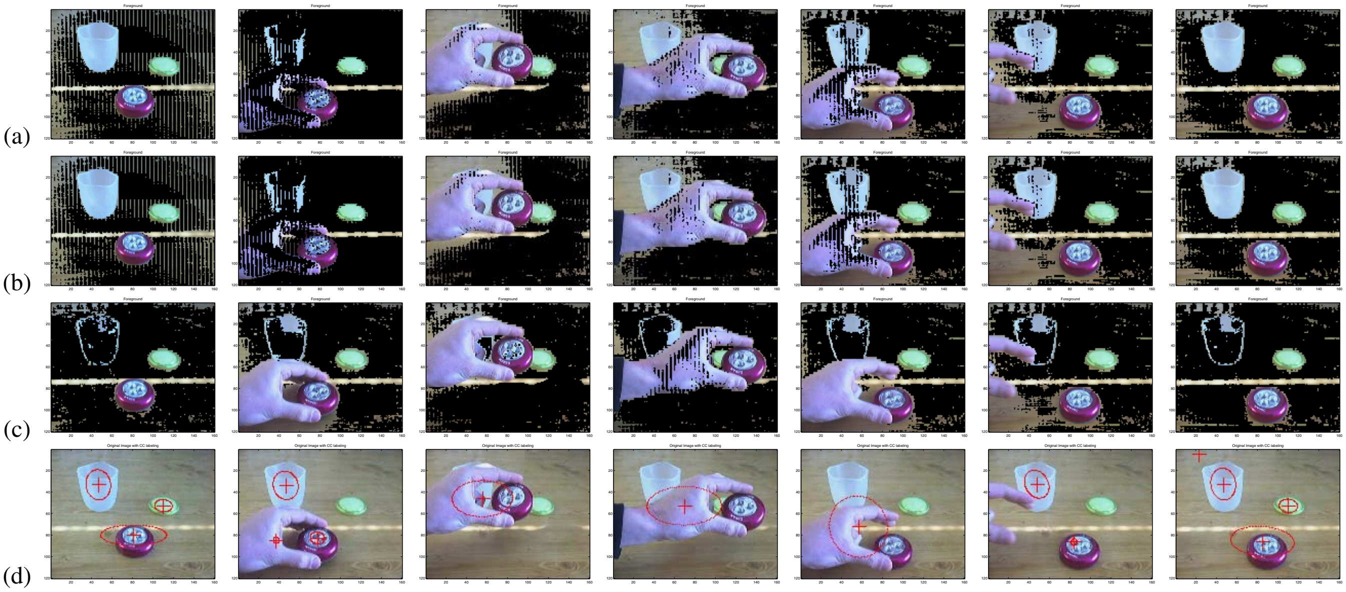

Fig. 3. Video segmentation in the RGB color space. Row (a) shows the results obtained applying [6], rows (b) those obtained with [9], rows (c) show our outcomes, and the last ones (d) the original frames with the object detection superimposed..

our approach demonstrates to behave better since the first images, maintaining this during the whole video processing. More specifically, the table is almost all recognized as background, together with the glass on the left. The other two algorithms both perform worse. It is worth noticing that we experimentally tried the best parameters for [6] and [9], and the best α specifically. In fact, this greatly affect the table recognition for instance, at the top left of each image. Higher values lead to a faster learning, but with instability and worse detection of the table as foreground at the end, and viceversa.

D. Hybrid Structure: Our Initialization with [9] procedure

Finally, we want to make another test: Comparing Lee's formulation with our initialization. This will bring about to an hybrid approach.

We tested it on the same previous videos. Fig. 4 shows the results. After the first mixture learning by means of the FSAEM procedure, the two algorithms work equally well, though our approach returns a slightly higher Log-likelihood.

Actually, the [9] performance does not change with respect to the previous section results once the mixture has been learned sufficiently. This is obvious, since the learning phase determines the succeeding iterations. We do not mention [6] here because its behave really similar to [9], with the solely difference of a slower initial learning procedure.

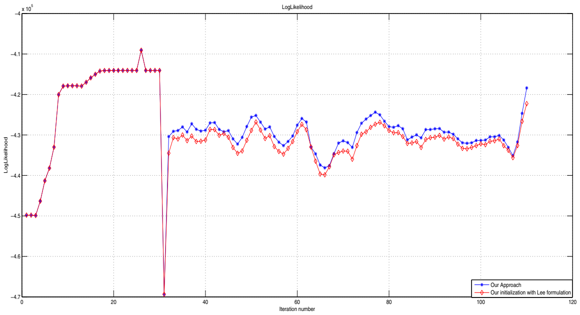

Fig. 4. Log-likelihood of the in-door video situation. The hybrid approach (red) versus our technique (blue) is employed for the video segmentation. The mixture learning at the beginning is the same for both methods, while during the further computations a slightly higher Log-likelihood is achieved with the proposed algorithm.

VI. Final Considerations

One can ask which one between our technique and [9] learns faster. This depends on the image to be segmented. It may happen that the best compromise between the number of components goes toward a small value, therefore ending FSAEM earlier, or viceversa. Therefore, on one hand our method would result in a slower initial performing.

Nevertheless, it is worth noticing that we tested [9] giving it *a-priori* the same model complexity we estimated using our technique. Since this value greatly affects both the computational complexity (the more components there are, the longer the computation is) and the Log-likelihood (the more components there are, the higher the Log-likelihood results) it is not possible asserting that [9] is better. In fact, we give [9] a fundamental parameter that it *a-priori* does not known, needing to be decided by the user heuristically before each analysis. Finally, with technique does not require the user imposing the learning rate decision a-priori. This is a great advantage, since the latter affects the further computations and video analysis considerably.

VII. Conclusions

In this work we presented a real-time video segmentation algorithm based on a modified EM procedure. The main feature of our approach is an automatic technique, based on a previous work of ours, that learns the starting best mixture from the first frame, subsequently speeding-up the further learning procedure with respect to the state-of-the-art. Then, a fixed number of our modified EM is performed for each frame, in order to adapt the mixture to the new image, while maintaining the property of real-time computation, without requiring the employment of a heuristic learning rate to be decided *a-priori*. Finally we proposed our technique for the foreground/background segmentation, based on a statistical framework. We tested our method against the conventional procedures, [6] and [9].

Acknowledgements

This work was supported by the European Commission, Project IST-004370 RobotCub and FP7-231640 Handle, and by the Portuguese Government - Fundação para a Ciência e Tecnologia (ISR/IST pluriannual funding) through the PIDDAC program funds and through project BIO-LOOK, PTDC / EEA-ACR / 71032 / 2006.

References

[1] H. Greenspan, J. Goldberger, and A. Mayer, “Probabilistic space-time video modeling via piecewise gmm,” *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 26, no. 3, pp. 384–396, 2004.

[2] C. Fowlkes, S. Belongie, and J. Malik, “Efficient spatiotemporal grouping using the nystrom method,” *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 231–238, 2001.

[3] G. Iyengar and A. B. Lippman, “Videobook: An experiment in characterization of video,” *Proc. IEEE Int Conf. Image Processing*, vol. 3, pp. 855–858, 1996.

[4] S.-F. Chang, H. Chen, W. Meng, H. Sundaram, and Z. D., “A fully automated content-based video search engine supporting spatiotemporal queries,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 602–615, 1998.

[5] B. Duc, P. Schroeter, and J. Bigun, “Spatio-temporal robust motion estimation and segmentation,” *Proc. Sixth Int Conf. Computer Analysis Images and Patterns*, pp. 238–245, 1995.

[6] C. Stauffer and W. Grimson, “Adaptive background mixture models for real-time tracking,” *Proc. Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 246–252, June 1999.

[7] D.-S. Lee, E. Berna, and J. J. Hull, “Segmenting people in meeting videos using mixture background and object models,” *Advances in Multimedia Information Processing — PCM 2002*, no. ISBN 978-3-540-00262-8, pp. 393–401, 2002.

[8] Z. Zivkovic, “Improved adaptive gaussian mixture model for background subtraction,” *Proc. ICPR*, 2004.

[9] D.-S. Lee, “Effective gaussian mixture learning for video background subtraction,” *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 27, no. 5, May 2005.

[10] N. Greggio, A. Bernardino, and J. Santos-Victor, “Sequentially greedy unsupervised learning of gaussian mixture models by means of a binary tree structure.” *11-th International Conference on Intelligent Autonomous Systems (IAS-11) 2010 - Aug 30, Sept 1*, 2010.

[11] N. Greggio, A. Bernardino, C. Laschi, J. Santos-Victor, and P. Dario, “Unsupervised greedy learning of finite mixture models.” *IEEE 22th International Conference on Tools with Artificial Intelligence (ICTAI 2010), Arras, France*, October 27-29 2010.

[12] N. Vlassis and A. Likas, “A greedy em algorithm for gaussian mixture learning,” *Neural Processing Letters*, vol. 15, pp. 77–87, 2002.

[13] J. Verbeek, N. Vlassis, , and B. Krose, “Efficient greedy learning of gaussian mixture models,” *Neural Computation*, vol. 15, no. 2, pp. 469–485, 2003.

[14] A. Figueiredo and A. Jain, “Unsupervised learning of finite mixture models,” *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 24, no. 3, 2002.

[15] B. J. and A. Smith, *Bayesian Theory*. Chichester UK: John Wiley and Sons, 1994.

[16] N. Friedman and S. Russel, “Image segmentation in video sequences: A probabilistic approach,” *Proc. 13th Conf. Uncertainty in Artifical Intelligence*, August 1997.