2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance

Splitting Gaussians in Mixture Models

Rubén Heras Evangelio, Michael Pätzold and Thomas Sikora
Communication Systems Group, Technische Universität Berlin
Einsteinufer 17, 10587 Berlin, Germany
heras,paetzold,sikora@nue.tu-berlin.de

***Abstract*—Gaussian mixture models have been extensively used and enhanced in the surveillance domain because of their ability to adaptively describe multimodal distributions in real-time with low memory requirements. Nevertheless, they still often suffer from the problem of converging to poor solutions if the main mode stretches and thus over-dominates weaker distributions. Based on the results of the Split and Merge EM algorithm, in this paper we propose a solution to this problem. Therefore, we define an appropriate splitting operation and the corresponding criterion for the selection of candidate modes, for the case of background subtraction. The proposed method achieves better background models than state-of-the-art approaches and is low demanding in terms of processing time and memory requirements, therefore making it especially appealing in the surveillance domain.**

***Keywords*-Background subtraction; Gaussian mixture models; Video surveillance**

I. Introduction

The detection of change is a low-level vision task used as a first step in many computer vision applications in order to reduce the computational load of further processing steps as object detection, object tracking and scene analysis. Therefore, the results obtained at this first processing step are of crucial importance for the success of higher-level tasks based on them. Background subtraction is a frequently adopted approach, especially in static camera setups, aiming to accomplish the task of change detection. Basically, background subtraction algorithms use a model of the static scene, the background model, in order to distinguish between background and foreground, i.e. units of relevant change, in video sequences.

There have been many different proposals for the task of background subtraction [1], [2]. Among them, Gaussian Mixture Models (GMMs) [3] have proven their outstanding suitability in the surveillance domain because of their ability to achieve many of the requirements of a surveillance system, e.g. adaptability and multimodality, in real-time with low memory requirements. GMMs model the history of each pixel by a mixture of K Gaussian distributions, which are updated by means of an Expectation Maximization (EM)-like algorithm.

The method in [3] has been enhanced in many directions. In [4] the use of a negative prior evidence was introduced in order to discard the components that are not supported by the data, therefore being able to constantly adapt the number of components of the mixture used for each pixel. In [5] the use of an adaptive learning rate calculated for each Gaussian at every frame was proposed, therefore being the convergence rate improved without compromising the model stability. Recently, in [6], a windowed weight update scheme, which is also suitable for a hardware implementation, was proposed to further reduce execution time. A comprehensive study of the original method and its derivations can be found in [7], where the authors gather the improvements that have been published in over 150 papers.

The EM [8] algorithm is a general approach used to iteratively compute maximum likelihood estimates for models with latent variables. However, due to its greedy nature, the EM algorithm is sensitive to initialization when fitting finite mixtures, thereby suffering from two main problems: singularities and local maxima. In [9], split and merge operations were proposed in order to overcome the local maxima problem, when fitting mixture models to stationary distributions.

The EM variant used to update GMMs for the task of background subtraction may also converge to local maxima if the main mode stretches, thus over-dominating weaker distributions. As a consequence, detection results decline. Following the same guiding principle as in [9], in this paper we propose to split over-dominating modes. Therefore, we derive an appropriate splitting operation and the corresponding criterion for the selection of candidate modes for the background subtraction approach, i.e., for the case when the underlying distribution is non-stationary. The selection criterion is based on a novel adaptive variance controlling value, which is also used in order to properly initialize new created modes. In Section II we thoroughly explain the proposed method. In Section III we present some experimental results, showing that our method leads to more accurate models of the background. Section IV concludes our paper.

II. GMM based Background Subtraction

A. State of the Art

State of the art GMMs follow the formulation of Stauffer and Grimson [3], thereby modelling the history of each pixel by a mixture of K Gaussian distributions. The probability of

978-0-7695-4797-8/12 $26.00 © 2012 IEEE
DOI 10.1109/AVSS.2012.69

observing a given pixel value X_t at time t is estimated as:

$$P(X_t) = \sum_{k=1}^{K} \omega_k \mathcal{N}(X_t, \mu_k, \Sigma_k) \quad (1)$$

where ω_k are the weights, and $\mathcal{N}(X_t, \mu_k, \Sigma_k)$ is a normal density of mean μ_k and covariance matrix Σ_k, which is assumed to be the diagonal matrix $\sigma_k^2 I$. The components are sorted according to their relevance and the background model is approximated by the first B components such that

$$B = \arg\min_k \left(\sum_{k=1}^{B} \omega_k > T \right) \quad (2)$$

where $B \leq K$ and T is a predefined threshold indicating the minimum portion of the data that should be assumed to be background. The model is continuously adapted by means of an EM-like algorithm, usually adopting a winner-takes-all update strategy. This means, that only the parameters of the selected matching mode are updated at a time. The matching mode is selected by computing the distance of the observed pixel value X_t to the modes of the model in a descendant order and assuming the first mode m which distance is lower than a given threshold τ to be the best match. If none of the available modes matches the current pixel value X_t, a new mode is created with X_t as its mean, a default value for the variance and a low prior weight. If none of the modes in the model is free, this new created mode replaces the one with the lowest prior.

Although the winner-takes-all updating has become a *de facto* standard for efficiency reasons, it falls into the trap of allowing some Gaussians to stretch, especially in crowded environments, therefore over-dominating weaker ones. This pitfall can be even emphasized if new modes are not initialized with adequate parameters. We propose to overcome this problem by introducing a rule to split stretching modes, which is based on the estimation of a novel adaptive variance controlling value. This value is also used in order to properly initialize new modes.

B. Proposed Method

Our proposed method follows the standard formulation of GMMs, incorporates some of the recently proposed enhancements and further improves existing methods by properly initializing new modes and avoiding over-dominating modes by means of a splitting rule. For every new frame, the GMM corresponding to each pixel is updated as follows:

$$\omega_{k,t} = (1 - \alpha)\, \omega_{k,t-1} + \alpha M_{k,t} - \alpha c_T \quad (3)$$

where $k = 1 \ldots K$, $M_{k,t}$ is a binary function with value 1 for the matched mode and 0 otherwise, and c_T is the bias introduced by [4] to select the number of modes needed to describe each pixel; furthermore:

$$\mu_{m,t} = (1 - \rho_{m,t})\, \mu_{m,t-1} + \rho_{m,t} X_t \quad (4)$$

$$\sigma_{m,t}^2 = (1 - \rho_{m,t})\, \sigma_{m,t-1}^2 + \rho_{m,t} \delta_{m,t}^T \delta_{m,t} \quad (5)$$

where $m \in \{1 \ldots K\}$ is the matched mode, $\delta_{m,t} = (X_t - \mu_{m,t})$ and $\rho_{m,t}$ is a learning rate calculated individually for each mode as introduced in [5]:

$$\rho_{m,t} = \frac{1 - \alpha}{\eta_{m,t}} + \alpha \quad (6)$$

where $\eta_{m,t}$ is a variable used to count the number of observations for each mode. $\eta_{m,t}$ is set to 1 when a mode is created and consecutively incremented when the parameters of the mode are updated. Therefore, the parameters of recently created modes are updated approximately as based on sufficient statistics ($\rho_{m,t} \approx 1/\eta_{m,t}$) while older modes are updated in a recursive fashion ($\rho_{m,t} \approx \alpha$). We chose this kind of learning rate because of its ability to provide fast convergence at early learning stages while guaranteeing the same speed of learning for every mode throughout the whole system. Observe that in the update equations of [4] $\rho_{m,t} = \alpha/\omega_{m,t}$, therefore depending the update of each mode on the whole GMM instead of on its age. By ensuring the same kind of convergence along the whole system, feedback based systems as proposed in [10] and more recently in [11] can be more reliably build upon the background model. After updating the parameters of each matched mode, we check if the splitting rule as defined in section II-B2 should be applied.

If none of the available modes matches the current pixel value X_t, a new mode is created as explained in the next section.

1) Initialization of New Modes: New modes represent observations that were not contained in the model. Therefore, they are created with a low prior weight, a mean equal to the value X_t of the observation and an initialization value for the variance, which is adaptively computed as explained in the following.

The variance term of each mode accounts for the variation of the values corresponding to the given distribution. This variations are introduced by the camera noise, the kind of surface and the kind of object (moving objects usually exhibit a higher variance than static ones). Correctly initializing this parameter is of central importance since it has a significant implication on the behaviour of the model; too low a value may lead the model to over-fit some boundary of the feature space, while a too large value may lead the model to under-fit the underlying distribution.

When starting the system, the background model for each pixel has to be initialized. Therefore, we use the observed value at each pixel as its mean value and make a guess for the initialization of the variance parameter. To do that, we use the two first frames and compute for each pixel the deviation from the first to the second frame ($X_{t=1}$, $X_{t=2}$). We assume that most of the pixels in consecutive frames, respectively, belong to the same distribution. Furthermore,

let us assume that most of the pixels belong to the background and can, therefore, be described by Gaussian distributions $\mathcal{N}(\mu, \Sigma)$ with similar covariance matrices $\sigma_b^2 I$. If our assumptions hold, then the distribution of the deviations is also Gaussian $\mathcal{N}\left(0, 2\sigma_b^2 I\right)$. Therefore, we can use the median of the absolute deviations med to robustly estimate the standard deviation of the former distributions as:

$$\hat{\sigma}_b = \frac{med}{0.68\sqrt{2}} \tag{7}$$

and use $\hat{\sigma}_b$ to initialize our background model.

A similar method was used in [12] in order to estimate the bandwidth of the kernel for each pixel independently. In our estimation, we extrapolated the computation to the frame level by assuming that most of the pixels belong to the background. While this is certainly not always the case, the only consequence of including some foreground pixels in this computation would be an over-estimation of the variance corresponding to background pixels. The higher the number of moving objects in the scene, the higher the over-estimation. In practice, this does not affect much further detection results since, after this first estimation, the variance of each pixel is individually updated to match the underlying distribution. As we may show in the experimental section, our method converges to appropriate values even if this first estimation drifts because of violation of our assumptions.

Further modes are initialized using the value $\sigma_{i,t}$, which is set equal to $\hat{\sigma}_b$ at system initialization and continuously updated so as to fit to the dynamic of the scene. In order to update the value of $\sigma_{i,t}$, we observe the behaviour of the system from two different perspectives. On one hand, we consider the absolut deviation of the observations belonging to background pixels $\mathcal{D}_b^{abs} := \{|\delta_{p,m,t}| : p \in \mathcal{P}_b\}$, being $\mathcal{P}_b$ the set of pixels belonging to the background, with respect to $\sigma_{i,t}$. Following the arguments leading to eq. 7, $\sigma_{i,t}$ should have a similar value to the median of $\mathcal{D}_b^{abs}$, but, since the deviations in $\mathcal{D}_b^{abs}$ are affected by the value of $\sigma_{i,t}$ at the initialization time of the individual modes, this similarity is conditioned on past values of $\sigma_{i,t}$. Therefore, we consider on the other hand the absolute deviation of the observations belonging to recently created modes $\mathcal{D}_f^{abs} := \{|\delta_{p,m,t}| : p \in \mathcal{P}_f\}$, being $\mathcal{P}_f$ the set of pixels matching recently created modes, with respect to $\sigma_{i,t}$, which provide us the instant behaviour of the system. In order to evaluate the behavior of the system from these two different perspectives, we define two indicators, ν and $\hat{\sigma}_f$, and adapt the value of $\sigma_{i,t}$ as follows.

The first indicator, ν, is a counter of the number of positions between the median of the absolute deviation of the background pixels $\mathcal{P}_b$ with respect to their corresponding matching modes m at time t and the position that would occupy $\sigma_{i,t}$ if considered among $\mathcal{D}_b^{abs}$. That means, for every new frame we set $\nu = 0$ and for every updated background pixel $p \in \mathcal{P}_b$ we compare the variance of the matched mode $\sigma_{p,m,t}$ with $\sigma_{i,t}$ and set ν to:

$$\nu = \begin{cases} \nu + 1, & \text{if } \sigma_{p,m,t} > \sigma_{i,t} \\ \nu - 1, & \text{if } \sigma_{p,m,t} < \sigma_{i,t} \end{cases} \tag{8}$$

The second indicator, $\hat{\sigma}_f$, is an approximation of the median absolute deviation of recently created modes $\mathcal{P}_f$. To obtain this value, for every new frame we set $\hat{\sigma}_f = \sigma_{i,t}$ and for every new updated mode we compare the variance of the mode σ_m with $\hat{\sigma}_f$ and set $\hat{\sigma}_f$ to:

$$\hat{\sigma}_f = \begin{cases} \hat{\sigma}_f + 0.1, & \text{if } \sigma_m > \hat{\sigma}_f \\ \hat{\sigma}_f - 0.1, & \text{if } \sigma_m < \hat{\sigma}_f \end{cases} \tag{9}$$

where a new updated mode is a mode with $\eta_{m,t} < M$, with M being a small natural value usually set to 2. Eq. 9 is a recursive approximation on the median of a serie of values similar to the one proposed in [13].

After processing a whole frame we evaluate ν and $\hat{\sigma}_f$. A negative value of ν means that the median of the deviation of the updated modes is lower than the initialization variance $\sigma_{i,t}$. Therefore, we conjecture that $\sigma_{i,t}$ is too high. Conversely, a positive value means that the median of the deviation of the updated modes is higher than $\sigma_{i,t}$. In this case, we conjecture that $\sigma_{i,t}$ is too low. In order to verify this conjectures, we use $\hat{\sigma}_f$. If the median of the deviation of the recently created modes $\hat{\sigma}_f$ is lower than $\sigma_{i,t}$ we can corroborate that $\sigma_{i,t}$ is too high, otherwise we can corroborate that it is too low. By impossing the condition that both indicators ν and $\hat{\sigma}_f$ agree, we are able to lead $\sigma_{i,t}$ converging to the median of the deviation of the observations corresponding to background modes without being conditioned by their respective initialization settings.

If $\sigma_{i,t}$ is too high ($\nu < 0$ and $\hat{\sigma}_f < \sigma_{i,t}$), we update its value as:

$$\sigma_{i,t+1} = \sigma_{i,t} + \left(\frac{\sigma_{i,t}}{\hat{\sigma}_f} - 1\right)\frac{\nu}{N} \tag{10}$$

where N is the total number of pixels in a frame. That means, we decrease the value of $\sigma_{i,t}$ according to $\hat{\sigma}_f$ and ν.

If $\sigma_{i,t}$ is too low ($\nu > 0$ and $\hat{\sigma}_f > \sigma_{i,t}$), we update its value as:

$$\sigma_{i,t+1} = \sigma_{i,t} + \left(\frac{\hat{\sigma}_f}{\sigma_{i,t}} - 1\right)\frac{\nu}{N}\frac{c}{u} \tag{11}$$

where c is the number of created modes and u the number of updates. That means, we update the value of $\sigma_{i,t}$ according to $\hat{\sigma}_f$ and ν. We introduced the factor c/u in (11) in order to penalize higher values of $\sigma_{i,t}$, i.e., as the number of created modes decreases and the number of updated modes increases, $\sigma_{i,t}$ grows slower.

This process is repeated for every new frame. Observe that we use an approximation of the median absolute deviation of recently created modes in order to update $\sigma_{i,t}$. Since new modes mostly correspond to moving objects,

we are deliberately setting $\sigma_{i,t}$ slightly higher than the expected variance of most of the modes corresponding to background distributions, so as to be able to span a wide range of possible underlying distributions. Since the variance of each pixel is individually further updated, new modes corresponding to background distributions are expected to achieve appropriated variance values when they become part of the background.

The value $\sigma_{i,t}$ is also used to set a selection criterion for the splitting rule as we explain in the next section.

2) Splitting Over-Dominating Modes: The Split and Merge EM algorithm (SMEM) [9] was introduced in order to escape from local maxima when fitting a GMM with a fixed number of components to a given distribution. The intuition behind it is that the Gaussian modes can be better distributed over the feature space by simultaneously splitting a Gaussian in an underpopulated region while merging two Gaussians in an overpopulated region. The split and merge operations are followed by a *partial EM procedure* and the *full EM procedure* and repeatedly performed until convergence.

Our proposed splitting rule finds its roots in the SMEM algorithm. Nevertheless, there are two important differences that hinder a straightforward transfer of the SMEM algorithm to the background subtraction domain. First, the distribution that we try to fit in order to perform background subtraction is a non-stationary distribution. And second, the number of modes that we use is limited, but not fixed. Moreover, the winner-takes-all updating strategy and the matching mode selection scheme favour the update of dominating modes. Therefore, we can consider that the merging operation is implicitly done in the variant of the EM used for background subtraction and will only need to define an appropriate splitting rule.

To select candidate modes for the splitting operation we use the value $\sigma_{i,t}$ as calculated in the former section and set $\sigma_c^2 = c\sigma_{i,t}^2$, with $c > 1$. Updated modes m with $\sigma_m^2 > \sigma_c^2$ are selected for splitting into the m' and the m'' Gaussians. By setting $c > 1$ we account for a certain variation of the variance of background pixels. For $c \to \infty$ the behaviour of the system is the same as state-of-the-art GMMs with adaptive setting of the initialization variance. Selected Gaussians m are splitted as follows:

$$\omega_{m',t} = \omega_{m,t} \tag{12}$$

$$\omega_{m'',t} = \alpha \tag{13}$$

$$\mu_{m',t} = \mu_{m,t} \tag{14}$$

$$\mu_{m''} = X_t \tag{15}$$

$$\sigma_{m',t} = \sigma_{m'',t} = \sigma_{i,t} \tag{16}$$

That means, we use m' to represent the background and m'' to represent the foreground. Furthermore, we assume that the observed value X_t at the moment of splitting Gaussian m corresponds to a foreground pixel and that the mean value $\mu_{m,t}$ can still be considered as a good description of the background even if shrinking the variance of m' to the value of $\sigma_{i,t}$. Since the initial parameter values given to m' are often poor, we set its counter $\eta_{m',t}$ to a small value.

III. Experimental Results

To assess the proposed system, in the next referred to as SGMM for brevity, we first validated the proposed technique for the estimation of $\sigma_{i,t}$, then, measured the overall computational load and, finally, evaluated the segmentation results.

For the validation of the proposed technique for the estimation of $\sigma_{i,t}$, we used three video sequences exhibiting three different behaviours concerning the amount of foreground activity and lighting conditions, so as to proof that the parameter $\sigma_{i,t}$ is able to follow the charasteristics of the scene. The first sequence, *Lobby*, contains 70000 frames ($\approx$ 2 h.) recorded in the lobby of a crowded public building, which has both natural and artificial light. As it gets darker outside, it is easy to appreciate how the camera noise raises. The second sequence, *Winter*, contains 65000 frames ($\approx$ 1 h. 50 min.) recorded in a sparsely crowded yard in winter. At the beginning of the scene it is snowing and, therefore, measurements are very noisy; at the end of the scene stops snowing and, therefore, the noise shrinks. The third sequence, *Underground*, is a public sequence taken from the i-LIDS dataset supplied to AVSS 2007, containing 5223 frames ($\approx$ 3 min.). It contains a scene in an underground; the field of view is short and therefore the moving objects large. The noise is nearly constant during the whole scene.

In order to evaluate the estimated $\sigma_{i,t}$ values, we computed the absolute deviation for consecutive values (X_t, X_{t+1}) of each pixel for each pair of consecutive frames and estimated the standard deviation of the modes representing the background as in eq. (7). We used this value as groundtruth, σ_{GT}, and evaluate $\sigma_{i,t}$ with respect to it. In sparsely crowded environments, e. g. *Winter*, σ_{GT} approaches the variance of most of the pixels belonging to the background. In crowded environments, e.g. *Lobby*, σ_{GT} has a slightly higher value than most of the pixels belonging to the background. Therefore, our searched value $\sigma_{i,t}$ should be slightly higher or similar to σ_{GT}, depending on the kind of scene. Figure 1 shows the results obtained for the three above mentioned sequences. The blue line, $\sigma_{i,t}$, shows the behaviour of the algorithm as described in this paper. The proposed system is able to correctly follow the dynamic of the scene and finds values near to σ_{GT}. The dashed cyan, $\sigma_{o,t}$, and green, $\sigma_{u,t}$, lines, show that the algorithm also converges to suitable values even in the hypothetic case of a wrong initialization (this case was forced, since the algorithm started well for the three sequences).

Table I shows the processing time for the above mentioned sequences (each frame containing $720 * 576$ RGB pixels) in

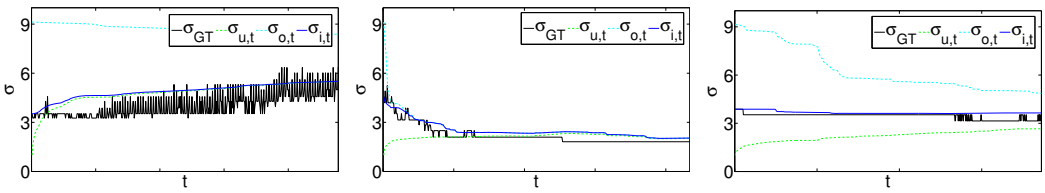

Figure 1. Behaviour of the proposed heuristic for the test video sequences *Lobby*, *Winter* and *Underground*.

a 3GHz PC without software optimization. For comparison, we also measured the processing time needed by the system in [4], in the next AGMM. AGMM is able to automatically select the number of needed components per pixel in order to adapt to the observed scene, but does not have any means to initialize and control the variance parameter of the Gaussian modes. The processing times of both systems are very similar for sparsely crowded scenarios. In fact, our proposed method converges to similar background models as AGMM in sequences of sparsely crowded scenarios, where over-dominating modes rarely appear. In crowded scenarios SGMM needs more processing time than AGMM. This is not a surprise, since AGMM often converges to models where over-dominating modes cover a wide range of the possible pixel values. Particularly, in the case of the *Lobby* sequence, many of the GMMs obtained by the AGMM converged to unimodal mixtures, therefore not being able to properly segment foreground objects. On the other hand, the proposed system was able to correctly select and split over-dominating modes and thus provided useful segmentation results. To summarize, in comparison to the reference system, the proposed system achieved similar segmentation results at similar processing times in sparsely crowded environments, while achieving significantly better results in crowded scenarios at the price of a slightly higher processing time.

Sequence	SGMM	IGMM
Lobby	43,96	37,52
Winter	34,15	33,65
Underground	34,84	35,45

Table I
PROCESSING TIME IN MS. OF THE THREE COMPARED SYSTEMS.

To evaluate the segmentation results, we used the dataset of the IEEE Workshop on Change Detection, held in conjunction with the CVPR 2012. The dataset consists of 31 surveillance videos divided into six categories covering most of the challenges regarding background subtraction for the task of video surveillance. The dataset is provided with a set of human-annotated ground truth and a toolkit to compute the performance metrics used, so as to enable a quantitative comparison and ranking of foreground segmentation algorithms. More information on the dataset can be found at www.changedetection.net. Our proposed algorithm was tested through the whole dataset and ranked against the algorithms that were provided as benchmark at the time of the workshop proposal, namely SOBS [14], ViBe [16], KDE [17], the seminal GMM formulation in [3] (in the table referred to as GMM), a GMM with a two phases kind of learning and shadow detection as proposed in [15] (in the table, TPGMM-SD), a GMM with automatic selection of number of components per pixel as proposed in [4] (in the table, AGMM), Mahalanobis distance [18] (in the table, MD) and Euclidean distance [18](in the table, ED). For the computation of the performance metrics used for ranking we used the provided toolkit. The performance metrics are: Recall (Re), Specificity (Sp), False Positive Rate (FPR), False Negative Rate (FNR), Percentage of Wrong Classifications (PWC), F-Measure and Precision. The results of the benchmark methods were taken from the website of the workshop. Table II shows the average results along the dataset, the ranking considering the average results, and the average ranking accross the different categories. The detailled results obtained for the individual categories will be provided to the organizers of the workshop, who aim to update the ranking of methods for years to come so that the dataset becomes like the Middlebury dataset for optical flow and stereo vision, upon publication of this article. The proposed method outperformed not only the GMM methods already evaluated as benchmark, but also every other evaluated method.

IV. CONCLUSIONS

The proposed system contibutes two main improvements to the GMMs for the task of background subtraction. First, we propose the incorporation of a heuristic in order to adaptively compute a value for the correct initialization of the variance parameter of new created modes, which leads models to faster converge to meaningful representations of the observed scene. Second, we derive a splitting rule in order to avoid over-dominating nodes, therefore significantly improving segmentation results in crowded environments. Moreover, since the computation of the variance controlling value only requires three global variables and a very reduced set of computationally light operations, the method complies the requirements of a surveillance system and can be straightforward extended to account for hardware considerations as proposed in [6].

Method	Average ranking accross categories	Average ranking	Average Re	Average Sp	Average FPR	Average FNR	Average PWC	Average F-Measure	Average Precision
SGMM	3,00	2,86	0.7074	0.9910	0.0090	0.0191	2.5299	0.7009	0.7813
SOBS	3,00	3,00	0.7854	0.9805	0.0195	0.0097	2.7049	0.7039	0.7040
TPGMM-SD	4,17	4,86	0.5075	0.9946	0.0054	0.0294	3.1296	0.5871	0.8182
KDE	4,17	5,29	0.7371	0.9749	0.0251	0.0147	3.5974	0.6607	0.6749
ViBe	4,33	5,00	0.6758	0.9825	0.0175	0.0182	3.2035	0.6599	0.7301
GMM	5.50	4.57	0.7070	0.9864	0.0136	0.0206	3.0962	0.6561	0.6987
AGMM	6,17	5,57	0.6942	0.9846	0.0154	0.0194	3.1498	0.6542	0.7045
MD	7,00	6,71	0.7584	0.9576	0.0424	0.0112	4.8771	0.6143	0.5904
ED	7,67	7,14	0.7020	0.9683	0.0317	0.0173	4.4509	0.6016	0.6110

Table II
SEGMENTATION RESULTS.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Community FP7 under grant agreement number 261743 (NoE VideoSense).

REFERENCES

[1] S. Brutzer, B. Hoferlin, and G. Heidemann, "Evaluation of background subtraction techniques for video surveillance," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado Spring, USA, June 2011, pp. 1937–1944.

[2] D. H. Parks and S. Fels, "Evaluation of background subtraction algorithms with post-processing," in *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2008, pp. 192–199.

[3] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, Fort Collins, CO, USA, Jun 1999, pp. 246–252.

[4] Zivkovic, "Improved adaptive gausian mixture model for background subtraction," in *Proceedings of the International Conference on Pattern Recognition*, 2004.

[5] D.-S. Lee, "Effective gaussian mixture learning for video background subtraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 827–832, 2005.

[6] P. Gorur and B. Amrutur, "Speeded up gaussian mixture model algorithm for background subtraction," in *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2011.

[7] T. Bouwmans, F. E. Baf, and B. Vachon, "Background modeling using mixture of gaussians for foreground detection - a survey," *Recent Patents on Computer Science*, vol. 1, pp. 219–237, 2008.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, 1977.

[9] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton, "Split and merge em algorithm for improving gaussian mixture density estimates," *The Journal of VLSI Signal Processing*, vol. 26, pp. 133–140, 2000.

[10] M. Harville, "A framework for high-level feedback to adaptive, per-pixel, mixture-of-gaussian background models," in *In Proceedings of the European Conference on Computer Vision*, 2002, pp. 543–560.

[11] R. Heras Evangelio and T. Sikora, "Complementary background models for the detection of static and moving objects in crowded environments," in *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2011.

[12] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," in *Proceedings of the IEEE*, vol. 90, no. 7, July 2002, pp. 1151–1163.

[13] N. J. B. McFarlane and C. P. Schofield, "Segmentation and tracking of piglets in images," *Machine Vision and Applications*, vol. 8, pp. 187–193, 1995.

[14] L. Maddalena and A. Petrosino, "A self-organizing approach to background subtraction for visual surveillance applications," *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1168 –1177, july 2008.

[15] P. Kaewtrakulpong and R. Bowden, "An improved adaptive background mixture model for realtime tracking with shadow detection," in *Proceedings of the 2nd European Workshop on Advanced Video Based Surveillance Systems*. Kluwer Academic Publishers, September 2001.

[16] O. Barnich and M. Van Droogenbroeck, "Vibe: A universal background subtraction algorithm for video sequences," *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1709 –1724, june 2011.

[17] A. Elgammal, D. Harwood, and L. S. Davis, "Non-parametric model for background subtraction," in *Proceedings of the 6th European Conference on Computer Vision*, 2000.

[18] Y. Benezeth, P.-M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger, "Comparative study of background subtraction algorithms," *J. Electronic Imaging*, vol. 19, 2010.