Smooth Foreground-Background Segmentation for Video Processing

Konrad Schindler and Hanzi Wang*

Electrical and Computer Systems Engineering, Monash University
{Konrad.Schindler|Hanzi.Wang}@eng.monash.edu.au

Abstract. We propose an efficient way to account for spatial smoothness in foreground-background segmentation of video sequences. Most statistical background modeling techniques regard the pixels in an image as independent and disregard the fundamental concept of smoothness. In contrast, we model smoothness of the foreground and background with a Markov random field, in such a way that it can be globally optimized at video frame rate. As a background model, the mixture-of-Gaussian (MOG) model is adopted and enhanced with several improvements developed for other background models. Experimental results show that the MOG model is still competitive, and that segmentation with the smoothness prior outperforms other methods.

1 Introduction

A basic requirement for video processing tasks with static cameras, such as surveillance and object tracking, is to segment the objects of interest from the permanently observed background. To this end, a model is estimated which describes the background, and parts of a frame which do not fit the model within a certain tolerance are labeled as foreground. What makes the task difficult is the fact that the background dynamically changes over time. Toyama et al. have termed the task "background maintenance" to point out the dynamic aspect of keeping the model up to date, and have presented a taxonomy of possible difficulties [1]. These include gradual and sudden illumination changes, shadows, vacillating background, foreground objects which share the characteristics of the background, foreground objects which remain static and must be merged into the background model, and the situation where no training images without foreground objects are available. Examples for these difficulties can be found in the test sequences in Sect. 4.

The literature about background maintenance can be broadly classified into two main approaches. *Non-predictive* methods recover a probability density function (*pdf*) of the observations at each pixel, and classify pixels as foreground, which do not match the function. The *pdf* can be approximated by a single Gaussian [2], a mixture of Gaussians [3] or a non-parametric distribution [4]. Some authors use not only intensities, but also higher-level information such as optical

* KS has been funded by the *Monash Institute for Vision Systems*, HW by the *ARC*.

flow [5]. A few methods do not work on single pixels: in [6], the background model is compressed to a set of codebook vectors, while [7] uses a simple mean image as background model, and normalized cross-correlation of small windows to measure how well two regions match.

A second class of methods uses *prediction* rather than density estimation to predict the pixel value, and classifies pixels as foreground, which do not match the prediction. Linear prediction is the basis of [1]. That paper also introduced the notion that background maintenance has to take into account different spatial scales: the initial result is improved using information at region-level for hole-filling, and at frame-level by maintaining several background models and switching between them, such that the foreground does not become too large. Prediction can also be performed with a Kalman filter [8], through projection onto a PCA-basis [9], or with an autoregressive model [10].

A classical statistical model, which is able to deal with many difficulties, is the mixture-of-Gaussian (MOG) model introduced by Stauffer and Grimson [3]. It describes the values of each background pixel throughout the sequence with a mixture of Gaussian distributions. Since several Gaussians are used, it correctly models multi-modal distributions due to periodic changes (e.g., a flag in the wind or a flickering light source), and since the parameters of the Gaussians are continually updated, it is able to adjust to changing illumination, and to gradually learn the model, if the background is not entirely visible in the beginning.

A straight-forward implementation of the MOG method has been shown to fail on several of the difficulties described above [1]. One goal of this paper is to show that most of these failures can be avoided, if the improvements suggested for different other background maintenance algorithms are incorporated into the MOG model, too. If the method is implemented carefully, the results are at least as good as for other standard methods. Firstly, the difficulties due to shadows and highlights can be solved using chromaticity coordinates, as already proposed in [11]. The second difficulty is more deep-rooted: the method uses a single learning rate to control two distinct phenomena, the adaptation to *changing illumination* and the fading of *static foreground objects* into the background. Therefore, foreground objects which stop moving are absorbed into the background too quickly. To overcome this limitation, a learning delay is introduced, which explicitly states how long a static object should remain foreground. Thirdly, we show that information, which can only be detected at frame-level (e.g. sudden changes in global illumination), can easily be fed back into the MOG model via the learning rate.

The main contribution of the paper does not concern the maintenance of the background model itself, but the way it is used to label pixels as background or foreground. Commonly, the background likelihood is simply thresholded for each pixel independently. In contrast, we argue that even at a low level the field of background probabilities contains spatial information. For a long time researchers have recognized that even prior to any semantic interpretation the visual world is smooth, in the sense that an image is generated by objects which are mapped to image regions with common properties [12]. This does not require semantic interpretation – even if the objects are unknown, the world is *a priori* more likely to generate a smooth foreground/background pattern than a random pattern (see Fig. 1). To make full use of the estimated likelihoods and add a smoothness prior, we cast the foreground/background segmentation as a labeling problem on a first-order Markov random field (MRF), and show how its optimal configuration can be efficiently found.

Fig. 1. Smoothness as prior belief. Random samples from the posterior distributions of segmentations without (left) and with (right) smoothness prior. Background probabilities are uniformly distributed, there is no semantics. Still the patterns on the right are visually more realistic.

The approaches closest to ours probably are [13], and very recently [14]. The former also model smoothness with a MRF. In their posterior, they combine normalized color and intensity (as advocated in Sect. 2), conventional (R, G, B)-color, and the output of an edge detector. For the resulting complicated energy functional, only a minimum of undetermined goodness is found. We propose a simpler posterior, which uses less information, but can be globally optimized and requires fewer parameters. [14] model both position and appearance in a single *pdf*, estimated with a kernel density method. They also estimate a foreground distribution, assuming smoothly changing foreground, and use a MRF-formulation similar to the one presented here to enforce spatial coherence.

In the last section, experiments on the *Wallflower* benchmark are presented, which show that the enhanced MOG-model is competitive with all other background maintenance methods we are aware of, and that, when used with a smoothness prior, it outperforms all other tested methods.

2 The Mixture-of-Gaussian Model

Principle. The intuition behind the MOG-model is the following: the intensities $\boldsymbol{x}$ of a given pixel form a time series, which can be represented as the mixture of a small number of Gaussians. Let the maximum number of Gaussians for a pixel be K (in our implementation set to $K = 5$). The probability that a pixel assumes a value $\boldsymbol{x}$ at a certain time t is then given by [3]

$$P(\boldsymbol{x}_t) = \sum_{i=1}^{K} \frac{w_{i,t}}{\sqrt{(2\pi)^n |\mathtt{S}_{i,t}|}} e^{-\frac{1}{2}(\boldsymbol{x}_t - \boldsymbol{m}_{i,t})^\top \mathtt{S}_{i,t}^{-1} (\boldsymbol{x}_t - \boldsymbol{m}_{i,t})} \tag{1}$$

where $\boldsymbol{m}_i$ is the mean of the i^{th} Gaussian, $\mathtt{S}_i$ is its covariance matrix, and w_i is its weight (the portion of data it accounts for), all at time t. For computational reasons, the channels of the image are assumed to be independent, so that $\mathtt{S}_k =$

$\operatorname{diag}(\boldsymbol{s}_k^2)$. To determine how many of the K Gaussians are needed for a pixel, the Gaussians are sorted by $\frac{w_k}{\operatorname{mean}(\boldsymbol{s}_k)}$, meaning that distributions based on a lot of evidence and distributions with low uncertainty come first. Only the first B distributions are chosen to represent the background, where

$$B = \arg\min_b \left(\sum_{k=1}^{b} w_k > T \right) \tag{2}$$

The value T determines the minimum fraction of the recent data at the location $\boldsymbol{x}$, which should contribute to the background model. If the background distribution is complicated, a larger value is needed to ensure enough Gaussians to approximate it. We use $T = 0.9$.

The parameters of the model are estimated in an initial training phase, and then continually updated as new data is observed. If the new pixel value $\boldsymbol{x}_t$ belongs to the i^{th} distribution, the parameters are updated to

$$\begin{aligned} \boldsymbol{m}_{i,t} &= (1-\alpha)\boldsymbol{m}_{i,t-1} + \alpha \boldsymbol{x}_t \\ \boldsymbol{s}_{i,t}^2 &= (1-\alpha)\boldsymbol{s}_{i,t-1}^2 + \alpha(\boldsymbol{x}_t - \boldsymbol{m}_{i,t})^{\mathsf{T}}(\boldsymbol{x}_t - \boldsymbol{m}_{i,t}) \end{aligned} \tag{3}$$

Here, α is the learning rate, which determines, how fast the parameters are allowed to change. The weights are updated to

$$w_{k,t} = (1-\alpha)w_{k,t-1} + \alpha U_{k,t} \quad , \quad U_{k,t} = \begin{cases} 1 \ldots \text{if } i = k \\ 0 \ldots \text{else} \end{cases} \tag{4}$$

Since new data gradually replaces older data in the background model, the algorithm can deal with gradual changes of the background, such as the ones typically encountered with natural light.

Implementation Issues. After its appearance in the literature, the MOG-model has been criticized by proponents of other background models, based on failure in a number of experiments. In this section we will argue that the MOG-model performs at least as well as other state-of-the-art methods, if it is carefully implemented. A quantitative comparison is presented in Sect. 4.

A frequent problem of background modeling methods is that cast shadows and moving highlights are incorrectly labeled as foreground, because they induce a sudden change of brightness. The common assumption to deal with these situations is that a change in illumination intensity alters only the lightness, but not the color of the region [15]. To suppress the influence of the lightness, several background modeling methods use normalized chromaticity coordinates, e.g. [4, 5]. The normalized chromaticity values are defined by $(r, g, b) = \frac{1}{R+G+B}(R, G, B)$, where two of the three values are sufficient. As a third coordinate, the intensity $I = (R + G + B)$ is used, which otherwise would be lost. In the new colorspace (r, g, I), color and intensity have been separated, and a shadow or highlight is expected to alter only the intensity. In any environment with a diffuse lighting component or multiple light sources, a shadow will only occlude a certain portion of the light, and a similar argument can be made for a highlight. Hence, the change in intensity is expected to stay within a certain range, $\beta \leq I_t/I_{t-1} \leq \gamma$. Within that range, the distribution is *not* Gaussian. Translated to the MOG-model, where we have to deal with multiple modes, and the expectation of the previous intensity is the mean $m_{\mathbf{I}i}$, we get the condition $\beta \leq I_t/m_{\mathbf{I}i} \leq \gamma$. Empirically, the intensity change due to shadows and highlights is at most 50%, so $\beta = 0.6, \gamma = 1.5$. In [11], a more exact procedure is derived based on statistical hypothesis-testing. However, we found that our simple approach gives good results and thus avoid the hypothesis test, which may be particularly vulnerable to the simplifying assumption of independence between the color channels.

Another issue when using the MOG-model is that the gray-value distribution is at best approximately Gaussian, so that the standard deviations $\boldsymbol{s}$ may be estimated incorrectly. On one hand, the sensor accuracy is limited, so extremely small standard deviations do not make sense. On the other hand, each Gaussian accounts only for one mode of the distribution, so $\boldsymbol{s}$ should only account for the variation within that mode. It is a matter of good engineering to bound $\boldsymbol{s}$ to reasonable values. In our implementation, we use $2 < s_{\mathbf{r},\mathbf{g}} < 15$ (for 8-bit images).

Thirdly, there is a dilemma how to set the correct learning rate. If a low α is chosen, the background model will take too long to adapt to illumination changes, while a high α will quickly merge the objects of interest into the background when they stop or move slowly. The reason is that a single learning rate is used to cover two different phenomena, namely the smooth variation of the background process over time, and the transition from foreground to background. This transition is a discrete process depending on the user's requirements ("after how many frames shall a static foreground object become background?"). A straight-forward way to separate the two phenomena is to stop learning a pixel's process, when it becomes foreground. After the pixel has continuously remained in the foreground for a given number of frames, background learning with equations (3) and (4) continues, and it will fade into the background with the speed given by the learning rate, if it remains static.

Finally, Toyama et al. have used a long-term memory to maintain multiple background models and switch between them to cope with sudden changes, such as switching on the light in a room. We agree with their reasoning that information at the frame level, rather than pixel level, is required to detect this type of change. The MOG-model provides an elegant way to deal with such situations: if a global change occurs, and almost the entire image is labeled as foreground, increasing the learning rate will automatically boost the adaptation to the new global conditions.

3 Adding Smoothness

In most probabilistic background models, each pixel is considered independent of the others, and a binary decision is taken: if the pixel does not match any of the background distributions, it is labeled as foreground. This contradicts the well-known fact that the world consists of spatially consistent entities, often called the *smoothness* assumption. In fact, standard background modeling algorithms such as the original MOG-method or *Wallflower* use an ad-hoc version of the smoothness assumption: they clean the foreground/background segmentation by deleting small foreground clusters using connected components.

We propose a more principled way to incorporate a smoothness prior: rather than simple thresholding, a continuous background probability value is retained for each pixel, and the foreground segmentation is treated as a labeling problem on a first-order Markov random field. Maximizing the posterior probability then results in a smooth, and more correct, segmentation.

Markov Random Fields (MRF) are a probabilistic way of expressing spatially varying priors, in particular smoothness. They were introduced into computer vision by Geman and Geman [16]. A MRF consists of a set of sites $\{x_1 \dots x_n\}$ and a neighborhood system $\{N_1 \dots N_n\}$, so that N_i is the set of sites, which are neighbors of site x_i. Each site contains a random variable U_i, which can take different values u_i from a set of labels $\{l_1 \dots l_k\}$. Any labeling $U = \{U_1 = u_1 \dots U_n = u_n\}$ is a realization of the field. The field is a MRF, if and only if each random variable U_i depends only on the site x_i and its neighbors $x_j \in N_i$. Each combination of neighbors in a neighborhood system is called a *clique* C_{ij}, and the prior probability of a certain realization of a clique is $e^{-V_{ij}}$, where V_{ij} is called the *clique potential.* The basis of practical MRF modeling is the Hammersley-Clifford Theorem, which states that the probability of a realization of the field is related to the sum over all clique potentials via $P(U) \propto \exp(-\sum V_{ij}(U))$.

If only cliques of 1 or 2 sites are used, the field is called a first-order MRF. The 1-site clique for each x_i is just the site itself, with likelihood $e^{-W_i(u_i)}$. Each 2-pixel clique consists of x_i and one of its neighbors, and has the likelihood $e^{-V_{ij}(u_i,u_j)}$. Following Bayes' theorem, the most likely configuration of the field is the one which minimizes the posterior energy function

$$E(U) = \sum_{x_i} \sum_{x_j \in N_i} V_{ij}(u_i, u_j) + \sum_{x_i} W_i(u_i) \tag{5}$$

It remains to define the clique potentials V_{ij}. If the goal is smoothness, and the set of labels does not have an inherent ordering, a natural and simple definition is the *Potts model* [17]

$$V_{ij} = \begin{cases} d_{ij} & \text{if } u_i \neq u_j \\ 0 & \text{else} \end{cases} \tag{6}$$

If two neighboring sites have the same label, the incurred cost is 0, else the cost is some value d_{ij}, independent of what the labels u_i and u_j are.

Application to Background. In the following, we will convert the background modeling problem into an MRF and show how to efficiently solve it. First, we have to define a background likelihood for each pixel. In the conventional MOG-method, a pixel $\boldsymbol{x} = [x_\mathbf{r}, x_\mathbf{g}, x_\mathbf{I}]^\mathsf{T}$ in the current frame is labeled as foreground, if it is far away from all modes of the background in terms of color or intensity.

$$
\boldsymbol{x} \rightarrow \mathcal{F} \text{ if } \begin{cases} \frac{(x_{\mathbf{r}i}-m_{\mathbf{r}i})^2}{s_{\mathbf{r}i}^2} + \frac{(x_{\mathbf{g}i}-m_{\mathbf{g}i})^2}{s_{\mathbf{g}i}^2} > \theta^2 & \forall i \in \{1..K\} \quad \textbf{or} \\ \frac{x_{\mathbf{I}}}{m_{\mathbf{I}i}} < \beta \textbf{ or } \frac{x_{\mathbf{I}}}{m_{\mathbf{I}i}} > \gamma & \forall i \in \{1..K\} \end{cases} \tag{7}
$$

In other words: $\boldsymbol{x}$ matches the i^{th} Gaussian, if its normalized distance from the mean is below a threshold θ (to cover 99.5% of the inliers to a Gaussian, $\theta = 2.81$). The evidence that $\boldsymbol{x}$ belongs to the background $\mathcal{B}$ is the probability that it belongs to the Gaussian, which it fits best, and only those Gaussians are valid, for which the intensity difference is not too large.

It is easy to convert this condition into a likelihood. The cost for labeling a pixel as foreground is constant, and shall be lower than the cost for labeling it as background only if condition (7) does not hold. The negative log-likelihood (the cost) of $\boldsymbol{x}$ in the i^{th} Gaussian is

$$
W_i(\boldsymbol{x}) = \begin{cases} \frac{(x_{\mathbf{r}}-m_{\mathbf{r}i})^2}{s_{\mathbf{r}i}^2} + \frac{(x_{\mathbf{g}}-m_{\mathbf{g}i})^2}{s_{\mathbf{g}i}^2} & \text{if } \beta \leq \frac{x_{\mathbf{I}}}{m_{\mathbf{I}i}} \leq \gamma \\ a\theta^2 & \text{else} \end{cases} \tag{8}
$$

where a is a constant >1, stating that the background cost is higher than the foreground cost, if the intensity difference is large Empirically, $a = 2.5$ performs satisfactory for all image sequences we have tested. Among the K Gaussians, the strongest evidence that $\boldsymbol{x}$ belongs to the background is the one with the lowest cost. If the modes are well separated, the likelihood of belonging to any other Gaussian is small, so the cost of assigning $\boldsymbol{x}$ to the background/foreground is

$$
\begin{aligned} W(\boldsymbol{x} \in \mathcal{B}) &= \arg\min_i (W_i(\boldsymbol{x})) \\ W(\boldsymbol{x} \in \mathcal{F}) &= \theta^2 \end{aligned} \tag{9}
$$

To model the neighborhood, we use the simplest possible definition: a pixel is connected to each neighbor in its 4-neighborhood, and the clique potential is a constant, which determines the amount of smoothing. We write the constant $V_{ij} = b\theta^2$, so that the cost for large intensity differences in equation (8) and the clique potential are on the same scale. Useful values are $1 \leq b \leq 4$.

Maximizing the posterior likelihood of the MRF is equivalent to minimizing the energy functional (5) over the space of realizations of the MRF. Since our special case has only 2 labels (background and foreground), the *global minimum* can be found in low polynomial time with the min-cut algorithm [18]: the MRF is converted into a graph, where the sites x_i are the nodes, and the cliques C_{ij} are the arcs joining the nodes x_i and x_j, with cost V_{ij}. The graph is augmented with two extra nodes for the two labels, which are connected to every site by an arc representing the corresponding likelihood W_i (plus a constant larger than the maximum clique potential for one node). The minimum cut on this graph partitions it into two sub-graphs, such that each node is only connected to one label. Min-cut is very efficient: we have tested it with the *Wallflower* benchmark with image size 160×120 pixels (see Sect. 4 for results). On a 2 GHz desktop PC, constructing the graph, solving the optimization, and clearing the memory takes on average 14 milliseconds, and thus does not impair the real-time capabilities of the MOG method.

4 Experimental Results

The algorithm has been tested with the *Wallflower* benchmark. This data set has been used by Toyama et al. to assess a large number of background maintenance methods. It has also been used by Kottow et al. to assess their method [6]. The data set consists of 7 video sequences of resolution 160×120 pixels, each representing a different type of difficulty that a background modeling system may meet in practice. For the last used frame of each sequence, manually segmented ground truth is available to enable a quantitative comparison. Tab. 1 shows the number of foreground pixels labeled as background (false negatives - FN), the number of background pixels labeled as foreground (false positives - FP), and the total percentage of wrongly labeled pixels $\frac{FN+FP}{160\times 120}$. Furthermore, the total number and percentage of wrongly labeled pixels over all 7 difficulties is given. As explained above, the authors of *Wallflower* have noted that information at the frame level is needed to deal with sudden illumination changes. However, they do not seem to have included this information in their implementations of other tested algorithms. This distorts the comparison, hence we also display the total results without the **Light Switch** sequence (column TOTAL*).

We have presented two improvements. First, we have shown that the original MOG-method is a valid and competitive algorithm, if implemented with the same care as other methods, and secondly we have applied the MRF-concept as a sound way to incorporate spatial smoothness. To separate the two parts' contributions, we present the results of our MOG algorithm cleaned up with the conventional connected component method, and the improved results using MRF smoothing. We did not tune towards the single sequences. All parameters were kept constant, except for the (automatic) increase of the learning rate in case of a sudden illumination change, as explained above. For some practical applications it may be possible to exclude certain scenarios and empirically find better parameter settings. However, we have found that this is not critical. The overall performance only increases by ≈600 pixels (15%), even if the optimal values are chosen for each sequence separately (which of course is improper tuning towards a specific data set).

Figure 2 shows the segmentation results for the most successful algorithms on the *Wallflower* data. A quantitative comparison is given in Tab. 1. The comparison should be taken with a grain of salt: choosing an algorithm will depend on the expected difficulties in a given application. Note however that our method yields the best result for all sequences. Also, an actual implementation must take into account the nature of the application. For example, in a high-security setting, one will seek to minimize false negatives and rather accept more false alarms. Any of the given algorithms has a parameter, which governs its sensitivity (up to which distance from the model a pixel is assigned to the

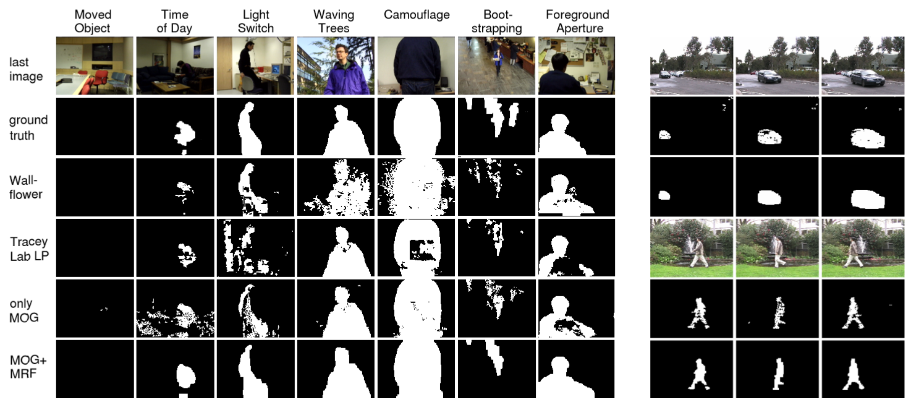

Fig. 2. ***Left:*** Wallflower benchmark. ***Right:*** "Car" and "fountain" videos. 3 frames of each sequence, results of improved MOG, results of MOG with MRF smoothing.

background), and can be tuned accordingly. Two more results of our method are shown in Fig. 2: a car moving in front of waving trees, and a person walking past a fountain, which is similar in color to the person's clothing.

Algorithm	**errors**	**MO**	**TOD**	**LS**	**WT**	**C**	**B**	**FA**	**TOTAL**	**TOTAL***
Eigen-	FN	0	879	962	1027	350	304	2441		
background†	FP	1065	16	362	2057	1548	6129	537	17677	16353
	%	*5.6*	*4.7*	*6.9*	*16.1*	*9.9*	*33.5*	*15.5*	*13.2*	*14.2*
MOG	FN	0	1008	1633	1323	398	1874	2442		
(original)†	FP	0	20	14169	341	3098	217	530	27053	11251
	%	*0.0*	*5.4*	*82.3*	*8.7*	*18.2*	*10.9*	*15.5*	*20.1*	*9.8*
Wallflower†	FN	0	961	947	877	229	2025	320		
	FP	0	25	375	1999	2706	365	649	11478	10156
	%	*0.0*	*5.1*	*6.9*	*15.0*	*15.3*	*12.5*	*5.1*	*8.5*	*8.8*
Tracey	FN	0	772	1965	191	1998	1974	2403	12035	8046
Lab LP‡	FP	1	54	2024	136	69	92	356		
	%	*0.0*	*4.3*	*20.8*	*1.7*	*10.8*	*10.8*	*14.4*	*9.0*	*7.0*
this paper	FN	0	203	1148	43	110	1159	1023	7340	5628
(only MOG)	FP	19	1648	564	278	468	143	534		
	%	*0.1*	*9.6*	*8.9*	*1.7*	*3.0*	*6.8*	*8.1*	*5.5*	*4.9*
this paper	FN	0	47	204	15	16	1060	34	3808	3058
(MRF smoothed)	FP	0	402	546	311	467	102	604		
	%	*0.0*	*2.3*	*3.9*	*1.7*	*2.5*	*6.1*	*3.3*	*2.8*	*2.7*

Table 1. Wallflower benchmark. † were reported in [1], ‡ were reported in [6].

5 Conclusions

A framework for smooth foreground/background segmentation in video streams has been presented, which can be applied with any probabilistic background model. In the present work, an improved MOG method is used, which overcomes a number of problems of the original method. The assumption of a smooth foreground/background pattern is treated in a principled, but computationally tractable way: segmentation is cast as a labeling problem on a particularly simple Markov random field, and solved with a classical algorithm.

It has been demonstrated that the method is fast enough for video-processing, and that it outperforms methods, which neglect smoothness or incorporate it in an ad-hoc manner. We do not challenge the principle formulated by Toyama et al., that semantic segmentation should not be handled by a low-level module like background maintenance. Rather, we claim that spatial smoothness already is a guiding principle before semantic interpretation.

References

1. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: principles and practice of background maintenance. In: Proc. 7th ICCV. (1999) 255–261
2. Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A.P.: Pfinder: Real-time tracking of the human body. IEEE TPAMI **19** (1997) 780–785
3. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: Proc. IEEE CVPR. (1999) 246–252
4. Elgammal, A., Harwood, D., Davis, L.S.: Non-parametric model for background subtraction. In: Proc. 6th ECCV. (2000) 751–767
5. Mittal, A., Paragios, N.: Motion-based background subtraction using adaptive kernel density estimation. In: Proc. IEEE CVPR. (2004) 302–309
6. Kottow, D., Koppen, M., Ruiz del Solar, J.: A background maintenance model in the spatial-range domain. In: Proc. 2nd SMVP. (2004)
7. Matsuyama, T., Ohya, T., Habe, H.: Background subtraction for non-stationary scenes. In: Proc. 4th ACCV. (2000) 662–667
8. Koller, D., Weber, J., Malik, J.: Robust multiple car tracking with occlusion reasoning. In: Proc. 3rd ECCV. (1994) 189–196
9. Oliver, N., Rosario, B., Pentland, A.: A Bayesian computer vision system for modeling human interactions. In: Proc. ICVS. (1999)
10. Monnet, A., Mittal, A., Paragios, N., Ramesh, V.: Background modeling and subtraction of dynamic scenes. In: Proc. 9th ICCV. (2003) 1305–1312
11. Greiffenhagen, M., Ramesh, V., Niemann, H.: The systematic design and analysis cycle of a vision system: a case study in video surveillance. In: Proc. IEEE CVPR. Volume 2. (2001) 704–711
12. Poggio, T., Torre, V., Koch, C.: Computational vision and regularization theory. Nature **317** (1985) 314–319
13. Paragios, N., Ramesh, V.: A MRF-based approach for real time subway monitoring. In: Proc. IEEE CVPR. (2001)
14. Sheikh, Y., Shah, M.: Bayesian object detection in dynamic scenes. In: Proc. IEEE CVPR. (2005) 74–81
15. Levine, M.D.: Vision in Man and Machine. McGraw-Hill (1985)
16. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. IEEE TPAMI **6** (1984) 721–741
17. Boykov, Y., Veksler, O., Zabih, R.: Markov random fields with efficient approximations. In: Proc. IEEE CVPR. (1998) 648–655
18. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in computer vision. In: Proc. 3rd EMMCVPR. (2001)