Department of Electrical
and
Computer Systems Engineering

Technical Report
MECSE-5-2006

Smooth Background Maintenance with Markov Random Fields

K. Schindler and H. Wang

MONASH
UNIVERSITY

Smooth Background Maintenance with Markov Random Fields

Konrad Schindler * and Hanzi Wang

Electrical and Computer Systems Engineering
Monash University, Clayton, 3800 VIC, Australia

Abstract

Background maintenance is a prerequisite for many video processing tasks. The mixture-of-Gaussian (MOG) model is an elegant way to formulate an adaptive statistical description of the background. In this work we incorporate several improvements developed for other background maintenance methods into the MOG model and show that, when properly implemented, the model is competitive with more recent methods. Secondly, most background maintenance algorithms regard the pixels in an image as independent and disregard the fundamental concept of smoothness. We propose to use a Markov random field to cleanly model smoothness of the foreground and background. Experimental results on the *Wallflower* benchmark show that our algorithm outperforms other background maintenance methods by more than 50%.

Key words: background maintenance, mixture of Gaussians, Markov random field

1 Introduction

A basic requirement for video processing tasks with static cameras, such as surveillance and object tracking, is to segment the interesting objects in the observed scene from the permanently present background. To this end, a model is estimated which describes the background, and those parts of a frame which do not fit the model within a certain tolerance are labeled as foreground. What may seem a trivial task at first glance turned out to be difficult, because the background dynamically changes over time. The background model must be able to adapt to these changes. Toyama et al. have termed the task "background maintenance" to point out the dynamic aspect

* Corresponding author
Email address: konrad.schindler@eng.monash.edu.au (Konrad Schindler).

of keeping the model up to date, and have presented a taxonomy of possible difficulties to be encountered [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 section 4 [1].

1.1 Related work

The last decade has produced a wealth of literature about background maintenance, which can be broadly classified into two main approaches. The first class, which we will call *non-predictive* methods, recovers a probability density function (*pdf*) of the observations at each pixel, and classifies pixels as foreground, which do not match the density function. Nakai has approximated the *pdf* by the histogram [4]. Wren et al. use a single Gaussian distribution [5], Stauffer and Grimson a mixture of Gaussians [6]. Elgammal et al. estimate non-parametric probability distributions from the data with kernel density methods [7]. They have also introduced normalized chromaticity instead of (R, G, B)-colorspace to remove shadows. Mittal and Paragios also use kernel density estimation, but base their background model not only on image intensities, but also optical flow [8]. While this certainly adds valuable information, it depends on optical flow estimation, which in itself is a difficult problem.

A few methods do not work on single pixels: Kottow et al. compress the background model to a set of codebook vectors [2], while Matsuyama et al. use a simple mean image as background model, but work on windows rather than pixels, and uses normalized cross-correlation instead of the intensity difference to measure how well two regions match [9].

A second class of methods uses *prediction* rather than estimation of the density to predict the pixel value, and classifies pixels as foreground, which do not match the prediction. Linear prediction is the basis of the method by Toyama et al. [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 portion of the image labeled as foreground does not become too

[1] The *Wallflower* benchmark results have served as a reference in background modeling, and other authors have used them for comparisons [2, 3] because they provide a cleverly composed set of problems as well as the most comprehensive comparison of different algorithms. However, the numerical values in the original publication are incorrect. The experiments section also reproduces the corrected results obtained for other background modeling methods on the same data, in order to provide correct figures for future reference.

large. Prediction has also been performed with a Kalman filter [10], through projection onto a PCA-basis [11], and with an autoregressive model [12].

The methods mentioned so far learn a background model and assign pixels which do not fit the model to the foreground. Some authors have proposed to also learn a distribution of the foreground, which is useful for settings, where the type of foreground objects is restricted, because then the statistics of the foreground over time also conveys useful information. Both Friedmann and Russell [13] and Ritscher et al. [14] consider the case of detecting and tracking cars, and model cars, road, and shadows as states of a hidden Markov model (HMM).

1.2 In defense of the MOG algorithm

An simple, yet powerful, statistical model, which is able to deal with many of the mentioned difficulties, is the mixture-of-Gaussian (MOG) model introduced by Stauffer and Grimson [6]. It describes the values of each background pixel throughout the sequence statistically 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. On the other hand, the use of simple parametric distributions makes the model computationally efficient and easy to configure.

A straight-forward implementation of the MOG method has been shown to fail on several of the potential difficulties a video processing system could meet [1]. One goal of this paper is to show that most of the encountered problems 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 those obtained with other state-of-the-art methods.

Firstly, we show that errors due to shadows and highlights can be avoided by using chromaticity coordinates also in the MOG method. The second problem 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 delay is introduced into the learning process, which explicitly states how long a static object should be remain in the foreground. Thirdly, we show that if information can only be detected at frame-level, such as sudden changes in global illumination, it can easily be fed back into the MOG model via the learning rate.

In section 2, the mixture-of-Gaussian model is presented in a more formal way, and the proposed modifications are discussed in more detail. In the

experiments section, results obtained with the modified MOG method are presented, and the results are compared with those of other methods, showing that the enhanced MOG model is competitive with all other background maintenance methods we are aware of.

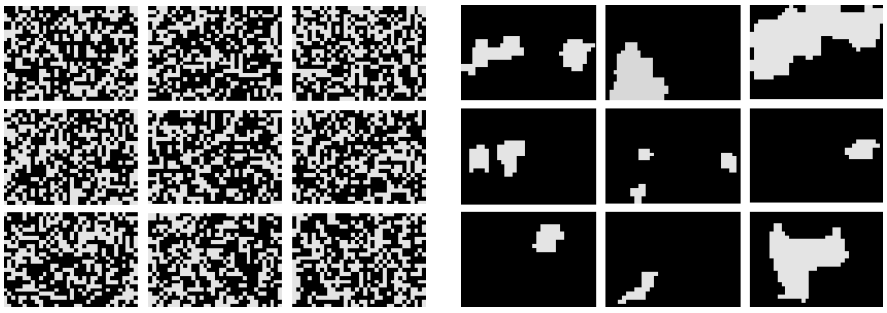
1.3 Spatially smooth foreground segmentation

The second 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. We present a Markov random field formulation of the labeling task, which is expressive enough to capture the smoothness of the visual world, but simple enough to be globally optimized in real-time. In a probabilistic background model such as MOG, the normalized residual of each pixel with respect to the background distribution is a continuous measure of the likelihood that the pixel belongs to the background. Commonly, the likelihood values are simply thresholded to obtain a binary labeling as foreground or background. A notable exception is the work of Cristani et al. [15]. In their approach, background maintenance is coupled with a semantic segmentation of the scene in order to treat different semantic regions of the background separately. However, this makes the low-level task of background maintenance dependent on the high-level problem of scene understanding.

In contrast, we argue that even at a low level, the spatial distribution within the field of background probabilities contains 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 [16]. This does not require semantic interpretation of the image – even if the objects are unknown, the world is *a priori* more likely to generate a smooth foreground/background pattern, rather than a random dot pattern (see Figure 1). To make full use of the estimated likelihoods and add a smoothness prior, we cast the foreground/background segmentation problem as a labeling problem on a first-order Markov random field (MRF), and show how the optimal configuration of the field can be efficiently found.

The approach of Paragios and Ramesh [17] is probably the most similar in spirit to the one presented here. They also enforce smoothness with a Markov random field, and combine intensity and normalized color (as advocated in this paper), conventional (R, G, B) color, and the output of an edge detector, to obtain a complicated energy functional. The method uses a large amount of information, but has the drawback that the resulting optimization problem is very complex. Only a local minimum of undetermined goodness is found. The method has been developed for a specific environment (subway monitoring), and a relatively large number of parameters has to be adjusted empirically to the application.

Section 3 describes in detail, how the output of the MOG-method at pixel



(a) (b)

Fig. 1. Smoothness of foreground segmentation as prior belief. Random samples from the posterior distribution of segmentations (a) without smoothness prior, and (b) with smoothness prior. The foreground/background probabilities of the pixels are uniformly distributed, there *is* no underlying semantics. Still the examples in (b) are visually more realistic foreground patterns.

level is combined with a Markov random field to model spatial coherence, and a method for efficient global optimization of the posterior probability is given. In section 4, results on several video sequences are presented, including the *Wallflower* benchmark. The results show that the proposed combination of the MOG model with a smoothness prior outperforms all other tested methods by more than a factor of 2.

2 The Mixture-of-Gaussian Model

2.1 Principle

The intuition behind the MOG model is the following: the intensities $\mathbf{x}$ 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 $\mathbf{x}$ at a certain time t is then given by [6]

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

where $\mathbf{m}_i$ 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 $\mathtt{S}_k = \text{diag}(\mathbf{s}_k^2)$. To determine how many of the K Gaussians are needed for a pixel, the Gaussians are sorted by $\frac{w_k}{\text{mean}(\mathbf{s}_k)}$, 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 location $\mathbf{x}$, 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. Our implementation uses $T = 0.9$ – this retains several (typically 3-5) Gaussians for unstable pixels and allows for multi-modal background distributions, so that the algorithm is able to deal with periodic changes such as flickering lights or waving trees in the background.

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 $\mathbf{x}_t$ belongs to the i^{th} distribution, the parameters are updated to

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

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

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

If a value does not match any of the distributions, the weakest Gaussian is discarded and a new one is instantiated with low weight and high standard deviation. 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.

2.2 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 section 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 [18]. To suppress the influence of the lightness, several background modeling methods use normalized chromaticity coordinates,e.g. [7, 8, 17]. The normalized chromaticity values (r, g, b) are

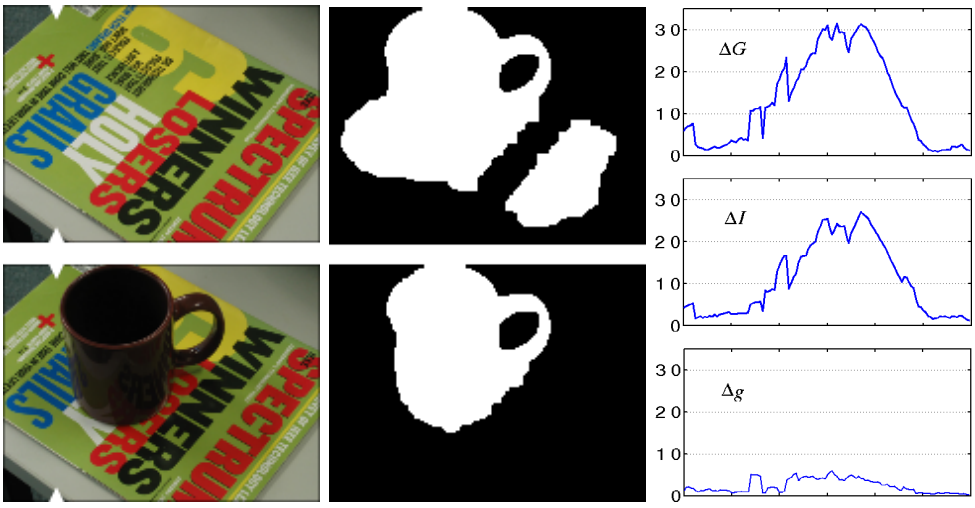
defined by

$$\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \frac{1}{R+G+B} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{5}$$

where two of the three values are sufficient, because the vector is of unit length. As a third coordinate, the intensity $I = (R+G+B)$ is used, which otherwise would be lost, and all three color coordinates are rescaled to $[0 \dots 255]$. In the new colorspace (r, g, I) color and intensity have been separated, and a shadow or highlight is expected to leave the r- and g-components unchanged and alter only the intensity. In any environment with a diffuse lighting component or multiple light sources, a shadow will only be able to occlude a certain portion of the light (and similarly a highlight can only add a certain amount of light), so the change in intensity is expected to stay within a certain range. If we call the previous intensity of a pixel I_b, and the current intensity I_t, then we may formulate this observation as $\beta \leq I_t/I_b \leq \gamma$. 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 we use $\beta = 0.6, \gamma = 1.5$. The effect of using (r, g, I) instead of (R, G, B) colorspace is illustrated in Figure 2.

Another issue when using the MOG model is that the distribution of the gray-values is at best approximately Gaussian, so that the standard deviations $\mathbf{s}$ 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 represents only one mode of the distribution, so $\mathbf{s}$ should only account for the variation within that mode. It is a matter of good engineering to bound $\mathbf{s}$ 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 and solve the problem is to stop learning the pixel process, when a pixel 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 the pixel will fade into the background with the speed given by the learning rate, if it remains static.



(a) (b) (c)

Fig. 2. Colorspace transformation for shadow removal. (a) Image without and with foreground object. (b) Change detection – **top:** using (R, G, B), **bottom:** using (r, g, I). In the top image one can clearly see the cast shadows from the two principal light sources.(c) Difference profile for the marked image column – **top:** green channel G, **center:** intensity channel I, **bottom:** normalized green channel g. In (r, g, I) the intensity and the chromaticity have been separated.

3 Adding Smoothness

In the standard MOG model, 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, foreground/background segmentation.

3.1 Markov Random Fields

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 [19], and have been applied to a wide variety of problems such as image restoration [20], stereo matching [21] and optical flow estimation [22]. A Markov random field 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 $P(U) \propto \exp(-\sum V_{ij}(U))$. A standard reference for MRFs in computer vision is [23].

If only cliques of 1 or 2 sites are used, the field is called a first-order MRF, and

$$P(U) \propto \exp\left(-\sum_{p_i} \sum_{p_j \in N_i} V_{ij}(u_i, u_j)\right) \tag{6}$$

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

$$E(U) = \sum_{x_i} \sum_{x_j \in N_i} V_{ij}(u_i, u_j) + \sum_{x_i} W_i(u_i) \tag{7}$$

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* [21]

$$V_{ij} = \begin{cases} d_{ij} & \text{if } u_i \neq u_j \\ 0 & \text{else} \end{cases} \tag{8}$$

If two neighboring sites have the same label, the incurred cost is 0, if they have different labels, the cost is some value d_{ij}, independent of what the labels u_i and u_j are. The d_{ij} can be constant, or they can be some function of the sites x_i and x_j.

3.2 Application to Background Modeling

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

$\mathbf{x} = [x_\mathbf{r}, x_\mathbf{g}, x_\mathbf{I}]^\mathsf{T}$ in the current frame is labeled as foreground, if it is too far away from all Gaussians of the background, or, according to our discussion of shadows in section 2.2, if the intensity difference is too large in all modes.

$$\mathbf{x} \rightarrow \mathcal{F} \text{ if } \begin{cases} \frac{(x_{\mathbf{r}i}-m_{\mathbf{r}i})^2}{s^2_{\mathbf{r}i}} + \frac{(x_{\mathbf{g}i}-m_{\mathbf{g}i})^2}{s^2_{\mathbf{g}i}} > \theta^2 & \forall\, i \in \{1..K\} \\ \text{or} & \\ \frac{x_\mathbf{I}}{m_{\mathbf{I}i}} < \beta \text{ or } \frac{x_\mathbf{I}}{m_{\mathbf{I}i}} > \gamma & \forall\, i \in \{1..K\} \end{cases} \tag{9}$$

In other words: $\mathbf{x}$ 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 $\mathbf{x}$ 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 (9) does not hold. The negative log-likelihood (the cost) of $\mathbf{x}$ in the i^{th} Gaussian is

$$W_i(\mathbf{x}) = \begin{cases} \frac{(x_\mathbf{r}-m_{\mathbf{r}i})^2}{s^2_{\mathbf{r}i}} + \frac{(x_\mathbf{g}-m_{\mathbf{g}i})^2}{s^2_{\mathbf{g}i}} & \text{if } \beta \leq \frac{x_\mathbf{I}}{m_{\mathbf{I}i}} \leq \gamma \\ a\theta^2 & \text{else} \end{cases} \tag{10}$$

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 $\mathbf{x}$ 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, hence the cost of assigning $\mathbf{x}$ to the background/foreground is

$$\begin{aligned} W(\mathbf{x} \in \mathcal{B}) &= \arg\min_i (W_i(\mathbf{x})) \\ W(\mathbf{x} \in \mathcal{F}) &= \theta^2 \end{aligned} \tag{11}$$

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 (10) and the clique potential are on the same scale. Useful values are $1 \leq b \leq 4$.

The presented case is a particularly simple MRF: there are only two labels (the so-called *Ising model*), and the connectivity graph is planar. For this special configuration, the global maximum of the posterior can be found in an efficient way. Maximizing the posterior is equivalent to minimizing the energy functional (7) over the space of realizations of the MRF. In general, this is a combinatorial problem, which is NP-hard for >2 labels, but it can be exactly solved in low polynomial time for only 2 labels and planar connectivity with

the min-cut/max-flow algorithm [24]: 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}. Furthermore the graph is augmented with two *terminal nodes* for the two labels, which are connected to every node of the graph with an arc representing the corresponding likelihood W_i (plus a constant which is larger than the maximum possible 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 terminal (label). The method is illustrated in Figure 3.

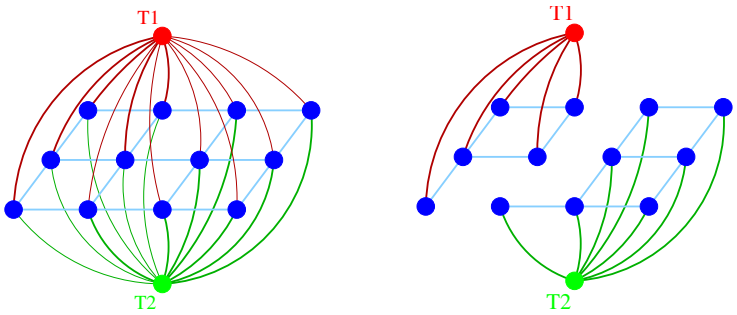


Fig. 3. Labeling through graph cuts. (a) Graph representing a MRF before segmentation. Line width denotes edge weight. (b) The minimum cut to obtain two unconnected subgraphs corresponds to the optimal labeling.

The min-cut algorithm is very efficient: we have tested it on several video sequences with image size 160×120 pixels (see section 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, including their own algorithm *Wallflower*. It has also been used by Kottow et al. to assess their method [2]. 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. These difficulties are

Moved Object (MO): A person enters a room, makes a phone call, and leaves. The telephone and chair are left in a different position.

Time of Day (TOD): The light in a room gradually changes from dark to bright. Then a person enters the room and sits down.

Light Switch (LS): A room scene begins with light on. A person enter the

room and turns off the light for a longer period. Later, a person walks into the room, switches on the light, and moves the chair.
Waving Trees (WT): A tree is swaying in the background, and a person walks in front of it.
Camouflage (C): A person enters the scene and occludes a monitor with rolling interference bars. The bars include colors similar to the person's clothing.
Bootstrapping (B): An image sequence from a busy cafeteria, all frames contain foreground objects.
Foreground Aperture (FA): A person with uniformly colored shirt wakes up and begins to move in the foreground.

For the last used frame of each sequence, manually segmented ground truth is available to enable a quantitative comparison. Table 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 $\frac{FN+FP}{160\times120}$. Furthermore, the total number and percentage of wrongly labeled pixels over all 7 difficulties is given. In their algorithm, Toyama et al. have used a long-term memory to maintain multiple background models and switch between them to cope with the sudden switching on of the light in the **Light Switch** sequence. 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, we increase the learning rate to boost the adaptation to the new global conditions. However, the authors of *Wallflower* do not seem to have included the information at frame-level in their implementations of other tested algorithms. This distorts the comparison, hence we also display the total results without the **Light Switch** sequence (column TOT*).

Here, we have presented two improvements to background modeling. First, we have shown that the original MOG-method is a valid and competitive algorithm, if implemented in the right colorspace and with the same care as other background modeling methods, and secondly we have applied the MRF concept as a sound way to incorporate spatial smoothness in low-level image processing, and have shown that in the special case of background modeling MRFs need not be computationally expensive. To isolate the contribution of each of these two parts, we present the results of our MOG algorithm using the conventional connected component method for cleaning up the segmentation, and the improved results using smoothing. We did not tune our methods towards the single sequences. In the MOG part, the only parameter change was the (automatic) increase of the learning rate from $\alpha = 0.001$ to $\alpha = 0.1$ in case of a sudden illumination change, as explained above. In the MRF part, the threshold $\theta = 2.81$ was used (covering 99.5% of the inliers to a Gaussian), and the two required parameters $a = 3.5$ and $b = 2.5$ were also kept constant.

For some practical applications it may be possible to exclude certain scenarios and empirically find better parameter settings. We have found that the "all-purpose" values given above are quite robust, and that the overall performance only increases by ≈600 pixels (15%), even if the optimal parameter values are chosen for each sequence separately (which of course is improper tuning towards a specific data set).

Figure 4 depicts the segmentation results for the algorithms, which have been most successful on the *Wallflower* sequence. A quantitative comparison is given in Table 1. The comparison shows *corrected* results: in the original paper [1], the column for the total error is wrong (the **Foreground Aperture** results were accidentally not added, although the correct results are displayed in a chart). We reproduce the corrected results for all algorithms as a reference for future publications.

The comparison should be taken with a grain of salt: choosing an algorithm will depend on which difficulties are expected in a given application. Note however that our method yields the best result for all sequences. Furthermore, the actual implementation must take into account the nature of the application. For example, in a high-security setting, one will seek to minimize the number of false negatives and rather accept more false alarms. Any of the given algorithms in some form contains a parameter, which governs its sensitivity (up to which distance from the expected value a pixel is assigned to the background), and can be tuned accordingly.

Two more results are shown in Figure 5 and 6. The first sequence shows a car passing in front of a background with swaying trees, recorded under natural outdoor lighting. The sequence was processed with the described algorithm. Both contributions of this paper were tested separately: first, the improved MOG algorithm was applied with the conventional connected components filtering, then it was applied with MRF smoothing. The results show that the MOG algorithm already achieves a quite good segmentation of the foreground object, and that MRF smoothing further improves the result and achieves a nearly perfect segmentation.

The second sequence is more difficult. It shows a person walking in front of a fountain, again filmed outdoors. Parts of the person's clothing are similar in color to the water and to the stone base of the fountain. Again, the results for the MOG algorithm alone as well as for the MOG algorithm with MRF smoothing are displayed. Some errors occur in regions, where the foreground color is similar to the background, but a large portion of the errors is repaired by smoothing.

5 Conclusions

An improved version of the mixture-of-Gaussian method for background maintenance has been presented, which overcomes a number of problems of

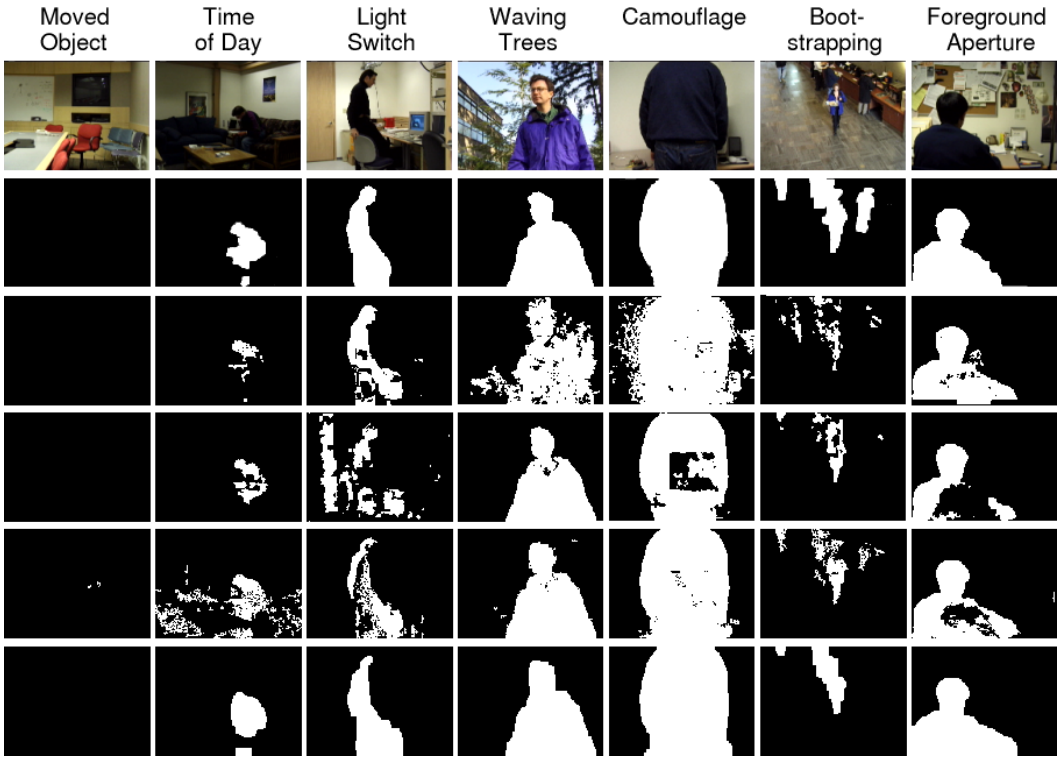


Fig. 4. Foreground/background segmentation for Wallflower benchmark. **Top row:** the image at which the processing was stopped and the results were evaluated. **2nd row:** manually segmented ground truth (from [1]). **3rd row:** Wallflower (from [1]). **4th row:** Tracey LAB LP (from [2]). **5th row:** our MOG-implementation. **Bottom row:** Our MOG algorithm with MRF smoothing.

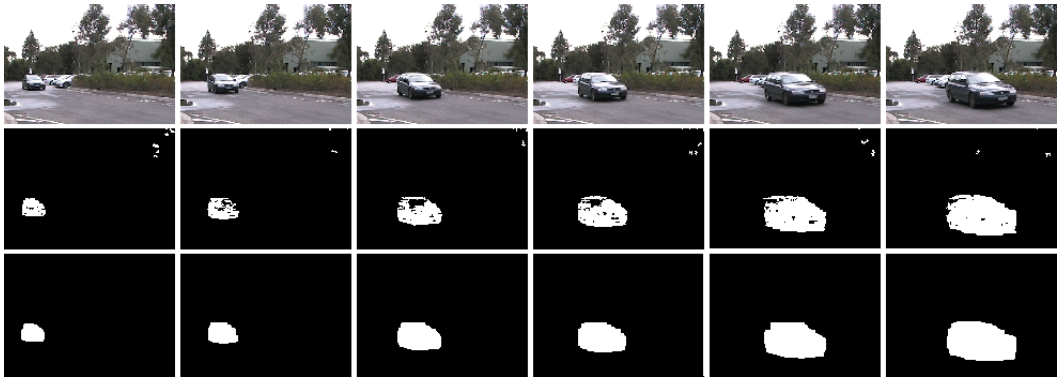


Fig. 5. Foreground/background segmentation for "car" video. **Top row:** 6 frames of the sequence. **2nd row:** Foreground segmentation with improved MOG algorithm. **Bottom row:** Foreground segmentation obtained with MOG algorithm and MRF smoothing.

the original algorithm. (r, g, I)-colorspace is used to cope with shadows and highlights, a frame-level component has been added to detect global illumination changes which cannot be dealt with at pixel level, and a short-term memory has been added to separate the adaptation to lighting changes from

Algorithm	**ERR**	**MO**	**TOD**	**LS**	**WT**	**C**	**B**	**FA**	**TOT**	**TOT***
Frame difference†	FN	0	1165	2479	3509	9900	1881	3884		
	FP	0	193	86	3280	170	294	470	27311	24746
	%	*0.0*	*7.1*	*13.4*	*35.4*	*52.5*	*11.3*	*22.7*	*20.3*	*21.5*
Mean+ threshold†	FN	0	873	1116	17	194	415	2210		
	FP	0	1720	15116	3268	1638	2821	608	29996	13764
	%	*0.0*	*13.5*	*84.5*	*17.1*	*9.5*	*16.9*	*14.7*	*22.3*	*12.0*
Mean+ covariance†	FN	0	949	1857	3110	4101	2215	3464		
	FP	0	535	15123	357	2040	92	1290	35133	18153
	%	*0.0*	*7.7*	*88.4*	*18.1*	*32.0*	*12.0*	*24.8*	*26.1*	*15.8*
MOG (original)†	FN	0	1008	1633	1323	398	1874	2442		
	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*
Block correlation†	FN	0	1030	883	3323	6103	2638	1172		
	FP	1200	135	2919	448	567	35	1230	21683	17881
	%	*6.3*	*6.1*	*19.8*	*19.6*	*34.7*	*13.9*	*12.5*	*16.1*	*15.5*
Temporal derivative†	FN	0	1151	752	2483	1965	2428	2049		
	FP	1563	11842	15331	259	3266	217	2861	46167	30084
	%	*8.1*	*67.7*	*83.8*	*14.3*	*27.2*	*13.8*	*25.6*	*34.4*	*26.1*
Bayesian decision†	FN	0	1018	2380	629	1538	2143	2511		
	FP	0	562	13439	334	2130	2764	1974	31422	15603
	%	*0.0*	*8.2*	*82.4*	*5.0*	*19.1*	*25.6*	*23.4*	*23.4*	*13.5*
Eigen-background†	FN	0	879	962	1027	350	304	2441		
	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*
Linear Prediction†	FN	0	961	1585	931	1119	2025	2419		
	FP	0	25	13576	933	2439	365	649	27027	11866
	%	*0.0*	*5.1*	*79.0*	*9.7*	*18.5*	*12.4*	*16.0*	*20.1*	*10.3*
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 Lab LP‡	FN	0	772	1965	191	1998	1974	2403	12035	8046
	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 (only MOG)	FN	0	203	1148	43	110	1159	1023	7340	5628
	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 (smoothed)	FN	0	47	204	15	16	1060	34	3808	3058
	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
Experimental results on Wallflower benchmark. † were reported in [1], ‡ were reported in [2]. See text for explanation.

the merging of static foreground objects into the background.

The main contribution of the paper is that the smoothness assumption for foreground/background segmentation has been treated in a principled, but computationally tractable way, and it has been demonstrated that a combination of the mixture-of-Gaussian algorithm with Markov random field modeling is efficient and outperforms other methods, which neglect smoothness or incorporate it in an ad-hoc way. 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 [1]. Rather, we claim that spatial smoothness is a guiding principle already at a low level, before semantic interpretation.

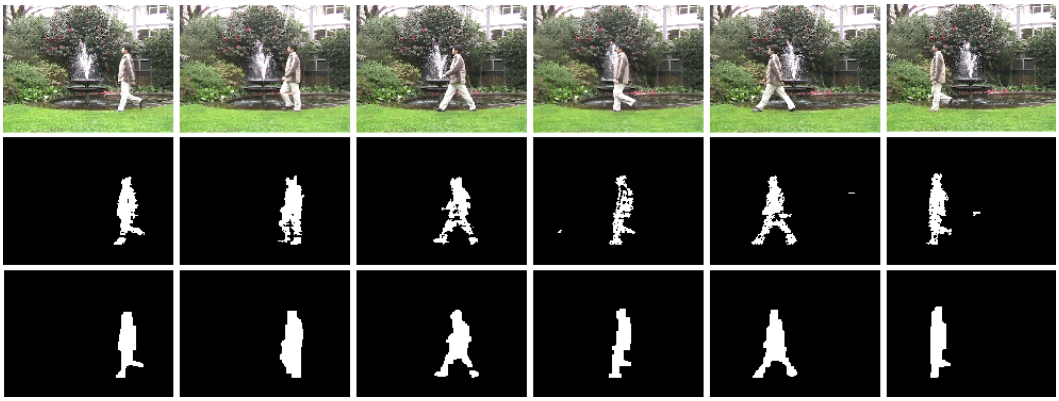


Fig. 6. Foreground/background segmentation for "fountain" video. **Top row:** 6 frames of the sequence. **2nd row:** Foreground segmentation with improved MOG algorithm. **Bottom row:** Foreground segmentation obtained with MOG algorithm and MRF smoothing.

Both parts have been separately evaluated on the *Wallflower* benchmark data set and have obtained lower error rates than other state-of-the-art algorithms.

Acknowledgments

This work was carried out within the Monash University *Institute for Vision Systems Engineering*. We would like to thank David Suter for valuable comments on the manuscript, and Kentaro Toyama for providing the *Wallflower* sequences. The code for solving the minimum cut problem has been made publicly available by Vladimir Kolmogorov on his web-page.

References

[1] K. Toyama, J. Krumm, B. Brumitt, B. Meyers, Wallflower: principles and practice of background maintenance, in: Proc. 7th International Conference on Computer Vision, Kerkyra, Greece, 1999, pp. 255–261.

[2] D. Kottow, M. Koppen, J. Ruiz del Solar, A background maintenance model in the spatial-range domain, in: Proc. 2nd Workshop on Statistical Methods in Video Processing, Prague, Czech Republic, 2004.

[3] H. Wang, D. Suter, A re-evaluation of mixture-of-Gaussian background modeling, in: Proc. International Conference on Acoustics, Speech, and Signal Processing, Philadelphia, USA, 2005, to appear.

[4] H. Nakai, Non-parameterized Bayes decision method for moving object detection, in: Proc. 2nd Asian Conference on Computer Vision, Singapore, 1995.

[5] C. R. Wren, A. Azarbayejani, T. Darrell, A. P. Pentland, Pfinder: Real-time tracking of the human body, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 780–785.

[6] C. Stauffer, W. E. L. Grimson, Adaptive background mixture models for real-time tracking, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, Colorado, 1999, pp. 246–252.

[7] A. Elgammal, D. Harwood, L. S. Davis, Non-parametric model for background subtraction, in: Proc. 6th European Conference on Computer Vision, Dublin, Ireland, 2000, pp. 751–767.

[8] A. Mittal, N. Paragios, Motion-based background subtraction using adaptive kernel density estimation, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Washington, D.C., 2004, pp. 302–309.

[9] T. Matsuyama, T. Ohya, H. Habe, Background subtraction for non-stationary scenes, in: 4th Asian Conference on Computer Vision, Taipei, Taiwan, 2000, pp. 662–667.

[10] D. Koller, J. Weber, J. Malik, Robust multiple car tracking with occlusion reasoning, in: Proc. 3rd European Conference on Computer Vision, Stockolm, Sweden, 1994, pp. 189–196.

[11] N. Oliver, B. Rosario, A. Pentland, A Bayesian computer vision system for modeling human interactions, in: Proc. International Conference on Vision Systems, Gran Canaria, Spain, 1999.

[12] A. Monnet, A. Mittal, N. Paragios, V. Ramesh, Background modeling and subtraction of dynamic scenes, in: Proc. 9th International Conference on Computer Vision, Nice, France, 2003, pp. 1305–1312.

[13] N. Friedman, S. Russell, Image segmentation in video sequences: A probabilistic approach, in: Annual Conference on Uncertainty in Artificial Intelligence, 1997, pp. 175–181.

[14] J. Ritscher, J. Kato, S. Joga, A. Blake, A probabilistic background model for tracking, in: Proc. 6th European Conference on Computer Vision, Dublin, Ireland, 2000.

[15] M. Cristani, M. Bicego, V. Murino, Multi-level background initialization using hidden Markov models, in: 1st International ACM Workshop on Video Surveillance, Berkeley, California, 2003, pp. 11–20.

[16] T. Poggio, V. Torre, C. Koch, Computational vision and regularization theory, Nature 317 (26) (1985) 314–319.

[17] N. Paragios, V. Ramesh, A MRF-based approach for real time subway monitoring, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, 2001.

[18] M. D. Levine, Vision in Man and Machine, McGraw-Hill, 1985.

[19] S. Geman, D. Geman, Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images, IEEE Transactions on Pattern Analysis and Machine Intelligence 6 (6) (1984) 721–741.

[20] J. Besag, On the statistical analysis of dirty pictures, Journal of the Royal Statistical Society, Series B 48 (1986) 259–302.

[21] Y. Boykov, O. Veksler, R. Zabih, Markov random fields with efficient approximations, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara, California, 1998, pp. 648–655.

[22] V. Kolmogorov, R. Zabih, Multi-camera scene reconstruction via graph cuts, in: Proc. 7th European Conference on Computer Vision, Copenhagen, Denmark, 2002.

[23] S. Z. Li, Markov Random Field Modeling in Image Analysis, 2nd Edition, Springer Verlag, 2001.

[24] Y. Boykov, V. Kolmogorov, An experimental comparison of min-cut/max-flow algorithms for energy minimization in computer vision, in: Proc. 3rd International Workshop on Energy Minimization Methods in Computer Vision, 2001.