Making Background Subtraction Robust to Sudden Illumination Changes

Julien Pilet, Christoph Strecha, and Pascal Fua

École Polytechnique Fédérale de Lausanne, Switzerland
{julien.pilet,christoph.strecha,pascal.fua}@epfl.ch
http://cvlab.epfl.ch/

Abstract. Modern background subtraction techniques can handle gradual illumination changes but can easily be confused by rapid ones. We propose a technique that overcomes this limitation by relying on a statistical model, not of the pixel intensities, but of the illumination effects. Because they tend to affect whole areas of the image as opposed to individual pixels, low-dimensional models are appropriate for this purpose and make our method extremely robust to illumination changes, whether slow or fast.
We will demonstrate its performance by comparing it to two representative implementations of state-of-the-art methods, and by showing its effectiveness for occlusion handling in a real-time Augmented Reality context.

1 Introduction

Background subtraction is a critical component of many applications, ranging from video surveillance to augmented reality. State-of-the-art algorithms can handle progressive illumination changes but, as shown in Fig. 1, remain vulnerable to sudden changes. Shadows cast by moving objects can easily be misinterpreted as additional objects.

This is especially true of approaches [2–4, 1] that rely on statistical background models that are progressively updated as time goes by. They can handle both illumination effects and moving background elements, such as tree leaves or flowing water. This is an obvious strength, but can result in mistakenly integrating foreground elements into the background model. This is a potentially serious problem in surveillance applications: A forgotten luggage could accidentally become part of the background. Furthermore, the model update is usually relatively slow, making it difficult to rapidly adjust to sudden illumination changes and to shadows cast by moving objects.

Here, we propose an approach that overcomes this problem by replacing the statistical background model by a statistical illumination model. More specifically, we model the ratio of intensities between a stored background image and an input image in all three channels as a Gaussian Mixture Model (GMM) that accounts for the fact that different parts of the scene can be affected in different ways. We incorporate this GMM in an efficient probabilistic framework that

Fig. 1. Top row: Three very different input images and a model image of the same scene. The changes are caused by lights being turned on one after the other and the person moving about. Bottom row: Our algorithm successfully segments out the person in all three input images. The rightmost image depicts the completely wrong output of a state-of-the-art approach [1] applied on the third image.

accounts for texture, background illumination, and foreground colour clues. Its parameters are computed by Expectation Maximization (EM) [5].

This approach reflects our key insight that, assuming that the background is static, changes in intensity of non-occluded pixels are mainly caused by illumination effects that are relatively global: They are not the same in all parts of the image but typically affect similarly whole portions of the image as opposed to individual pixels. As a result, they can be modelled using GMMs with only few components—2 in the experiments presented in this paper—which leads to a very robust algorithm.

We will demonstrate that our algorithm outperforms state-of-the-art background subtraction techniques when illumination changes quickly. The key difference between these techniques and ours is that they directly estimate distributions of pixel intensities as opposed to illumination effects as we do. We will also show that our approach performs well in an Augmented Reality context where a moving object is treated as the background from which occluders such as the hands holding it must be segmented out.

2 Related Work

Many background subtraction algorithms try to update on-line a statistical background model. A pixel from a new image is then classified as background if it fits the model. Wren et al. [2] represent the colour of each pixel by a three-dimensional Gaussian, learned from colour observation of consecutive frames. Since a single Gaussian is a poor approximation of the true probability density function, GMMs were proposed instead [3, 4]. These approaches have proved to be effective at handling gradual illumination changes and repetitive dynamic

backgrounds. Many improvements have been published since, such as a recent method that dynamically selects the appropriate number of components for each pixel [1]. We will use it as a benchmark against which we compare our approach because it is representative of this whole class of techniques.

Introducing a GMM is not the only way to model a dynamic background. Elgammal et al. proposed to model both background and foreground pixel intensities by a nonparametric kernel density estimation [6]. In [7], Sheikh and Shah proposed to model the full background with a single distribution, instead of one distribution per pixel, and to include location into the model.

Because these methods do not decouple illumination from other causes of background changes, they are more sensitive to drastic light effects than our approach.

Shadows cast by moving objects cause illumination changes that follow them, thereby hindering the integration of shadowed pixels into the background model. This problem can be alleviated by explicitly detecting the shadows [8]. Most of them consider them as binary [8], with the notable exception of [9] that also considers penumbra by using the ratio between two images of a planar background. Our approach also relies on image ratios, but treats shadows as a particular *illumination effect*, a wider class that also include the possibility of switching lights on.

Another way to handle illumination changes is by using illumination invariant features, such as edges. Edge information alone is not sufficient, because some part of the background might be uniform. Thus, Jabri et al. presented an approach to detect people fusing colour and edge information [10]. More recently, Heikkilä and Pietikäinen modelled the background using histograms of local binary patterns [11]. The bilayer segmentation of live video presented in [12] fuses colour and motion clues in a probabilistic framework. In particular, they observe in a labeled training set the relation between the image features and their target segmentation. We follow here a similar idea by training beforehand histograms of correlation and amount of texture, allowing us to fuse illumination, colour and texture clues.

3 Method

Our method can serve in two different contexts. For background subtraction, where both the scene and the camera are static. For augmented reality applications, where an object is moving in the camera field and occlusions have to be segmented for realistic augmentation.

Let us assume that we are given an unoccluded *model image* of a background scene or an object. Our goal is to segment the pixels of an *input image* in two parts, those that belong to the same object in both images and those that are occluded. If we are dealing with a moving object, we first need to register the input image and create an image that can be compared to the model image pixelwise. In this work, we restrict ourselves to planar objects and use publicly available software [13] for registration. If we are dealing with a static scene

and camera, that is, if we are performing standard background subtraction, registration is not necessary. It is the only difference between both contexts, and the rest of the method is common. In both cases, the intensity and colour of individual pixels are affected mostly by illumination changes and the presence of occluding objects.

Changes due to illumination effects are highly correlated across large portions of the image and can therefore be represented by a low dimensional model that accounts for variations across the whole image. In this work, we achieve this by representing the ratio of intensities between the stored *background image* and an input image in all three channels as a Gaussian Mixture Model (GMM) that has very few components—2 in all the experiments shown in this paper. This is in stark contrast with more traditional background subtraction methods [2–4, 1] that introduce a model for each pixel and do not explicitly account for the fact that inter-pixel variations are correlated.

Following standard practice [14], we model the pixel colours of occluding objects, such as people walking in front of the camera, as a mixture of Gaussian and uniform distributions.

To fuse these clues, we model the whole image — background, foreground and shadows — with a single mixture of distributions. In our model, each pixel is drawn from one of five distributions: Two Gaussian kernels account for illumination effects, and two more Gaussians, completed by a uniform distribution, represent the foreground. An Expectation Maximization algorithm assigns pixels to one of the five distributions (E-step) and then optimizes the distributions parameters (M-step).

Since illumination changes preserve texture whereas occluding objects radically change it, the correlation between image patches in the model and input images provides a hint as to whether pixels are occluded or not in the latter, especially where there is enough texture.

In order to lower the computational burden, we assume pixel independence. Since this abusive assumption entails the loss of the relation between a pixel and its neighbors, it makes it impossible to model texture. However, to circumvent this issue, we characterize each pixel of the input image by a five dimensional feature vector: The usual red, green, and blue values plus the normalized cross-correlation and texturedness values. Feature vectors are then assumed independent, allowing an efficient maximization of a global image likelihood, by optimizing the parameters of our mixture. In the remainder of this section, we introduce in more details the different components of our model.

3.1 Illumination Likelihood Model

First, we consider the background model, which is responsible for all pixels that have a counterpart in the *model image* $\boldsymbol{m}$. If a pixel u_i of the *input image* $\boldsymbol{u}$ shows the occlusion free target object, the luminance measured by the camera depends on the light reaching the surface (the irradiance e_i) and on its albedo. Irradiance e_i is function of visible light sources and of the surface normal. Under the lambertian assumption, the pixel value u_i is: $u_i = e_i a_i$, where a_i is the

albedo of the target object at the location pointed by u_i. Similarly, we can write: $m_i = e_m a_i$,with e_m assumed constant over the surface. This assumption is correct if the model image $\boldsymbol{m}$ has been taken under uniform illumination, or if a textured model free of illumination effects is available. Combining the above equations yields:

$$l_i = \frac{u_i}{m_i} = \frac{e_i}{e_m} ,$$

which does not depend on the surface albedo. It depends on the surface orientation and on the illumination environment. In the specific case of a planar surface lit by distant light sources and without cast shadows, this ratio can be expected to be constant for all i [9]. In the case of a 3 channel colour camera, we can write the function l_i that computes a colour illumination ratio for each colour band:

$$l_i = \left[\frac{u_{i,r}}{m_{i,r}} \; \frac{u_{i,g}}{m_{i,g}} \; \frac{u_{i,b}}{m_{i,b}} \right]^T ,$$

where the additional indices r, g, b denotes the red, green and blue channel of pixel u_i, recpectively.

In our background illumination model we suppose that the whole scene can be described by K different illumination ratios, that correspond to areas in u_i with different orientations and/or possible cast shadows. Each area is modelled by a Gaussian distribution around the illumination ratio μ_k and with full covariance Σ_k. Furthermore we introduce a set of binary latent variables $x_{i,k}$ that take the value 1 iff pixel i belongs to Gaussian k and 0 otherwise. Then, the probability of the ratio l_i is given by:

$$p(l_i \,|\, x_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{k=1}^{K} \pi_k^{x_{i,k}} \mathcal{N}\left(l_i; \mu_k, \Sigma_k\right)^{x_{i,k}} \quad , \tag{1}$$

where $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ denote all parameters of the K Gaussians. π_k weights the relative importance of the different mixture components. Even though the ratios l_i are not directly observed, this model has much in common with a generative model for illumination ratios.

So far we described the background model. The foreground model is responsible for all pixels that do not correspond to the model image $\boldsymbol{m}$. These pixels are assumed to be generated by sampling the foreground distribution, which we model as a mixture of $\bar{K}$ Gaussians and a uniform distribution. By this choice, we implicitely assume that the foreground object is composed of $\bar{K}$ colours μ_k, handled by the normal distributions $\mathcal{N}(u_i; \mu_k, \Sigma_k)$, and some suspicious pixels that occur with probability $1/256^3$. Again, as in the background model, the latent variables are used to select a specific Gaussian or the uniform distribution. The probability of observing a pixel value u_i given the state of the latent variable x_i and the parameters $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ is given by:

$$p(u_i \,|\, x_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \left(\frac{\pi_{K+\bar{K}+1}}{256^3}\right)^{x_{i,K+\bar{K}+1}} \prod_{k=K+1}^{K+\bar{K}} \pi_k^{x_{i,k}} \mathcal{N}\left(u_i; \mu_k, \Sigma_k\right)^{x_{i,k}} \quad . \tag{2}$$

The overall model consist of the background (Eq. 1) and the foreground (Eq. 2) model. Our latent variables x_i select the one distribution among the total $K+\bar{K}+1$ components which is active for pixel i. Consider figures 2(a) and 2(b) for example: The background pixels could be explained by $K = 2$ illumination ratios, one for the cast shadow and one for all other background pixels. The hand in the foreground could be modelled by the skin colour and the black colour of the shirt ($\bar{K} = 2$). The example in Fig. 2 shows clearly that the importance of the latent variable components is not equal. In practice, there is often one Gausssian which models a global illimination change, *i.e.* most pixels are assigned to this model by the latent variable component $x_{i,k}$. To account for the possibly changing importance, we have introduced π_k that globally weight the contribution of all Gaussian mixtures $k = 1 \ldots \bar{K}$ and the uniform distribution $k = \bar{K} + 1$.

A formal expression of our model requires combining the background pdf of Eq. 1 and the foreground pdf of Eq. 2. However, one is defined over illumination, whereas the other over pixel colour, making direct probabilities incompatible. We therefore express the background model as a function of pixel colour instead of illumination:

$$p(u_i \,|\, x_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\mid J_i \mid} p(l_i \,|\, x_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \;, \tag{3}$$

where $\mid J_i \mid$ is the determinant of the Jacobian of function $l_i(u_i)$. Multiplying this equation with Eq. 2 composes the complete colour pdf.

Some formulations define an appropriate prior model on the latent variables $\boldsymbol{x}$. Such a prior model would incorporate the prior belief that the model selection $\boldsymbol{x}$ shows spatial [14] and spatio-temporal [12] correlations. These priors on the latent variable $\boldsymbol{x}$ have shown to improve the performance of many vision algorithms [15]. However, they increase the complexity and slow down the computation substantially. To circumvent this, we propose in the next section a spatial likelihood model, which can be seen as a model to capture the spatial nature of pixels and which allows real-time performance.

3.2 Spatial Likelihood Model

In this section, we present an image feature and a way to learn off-line its relationship with our target segmentation. Consider an extended image patch around pixel i for which we extract a low dimensional features vector $f_i = [f_i^1, f_i^2]$. The basic idea behind our spatial likelihood model is to capture texture while keeping a pixel independence assumption. To achieve real-time performance we use two features that can be computed very fast and model their distribution independently for the background and for the foreground, by histograms of the discretized feature values. We use the normalized cross-correlation (NCC) between input and model image as one feature and a measure of the amount of texture as the other feature. f_i^1 is given by:

$$f_i^1 = \frac{\sum_{j \in w_i} (u_j - \bar{u}_i)\,(m_j - \bar{m}_i)}{\sqrt{\sum_{j \in w_i} (u_j - \bar{u}_i)^2 \sum_{j \in w_i} (m_j - \bar{m}_i)^2}} \;,$$

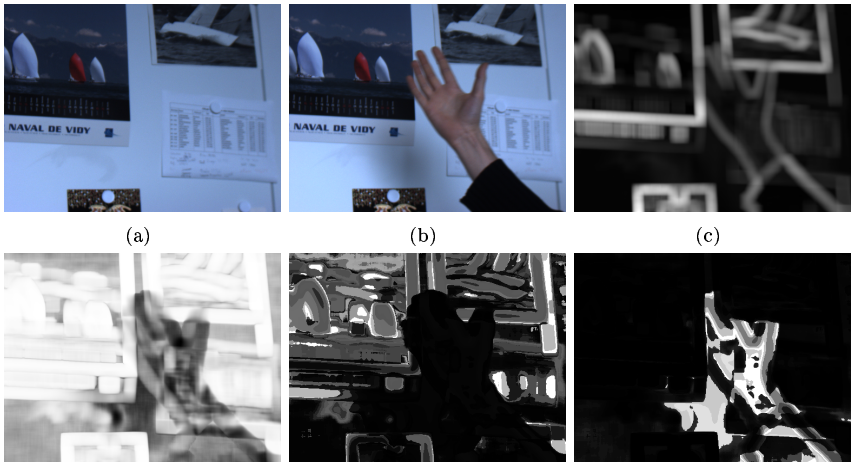

Fig. 2. Elements of the approach. (a) Background image $\boldsymbol{m}$. (b) Input image $\boldsymbol{u}$. (c) Textureness image $\boldsymbol{f}^2$. (d) Correlation image $\boldsymbol{f}^1$. (e) Probability of observing $\boldsymbol{f}$ on the background, according to the histogram $h(f_i \mid v_i)$ (f) Probability of observing $\boldsymbol{f}$ on the foreground, according to the histogram $\bar{h}(f_i \mid \bar{v}_i)$.

where w_i denotes a window around pixel i, and $\bar{u}_i = \frac{1}{|w_i|}\sum_{j\in w_i} u_j$ is the average over w_i. The correlation is meaningful only in windows containing texture. Thus, the texturedness of window i is quantified by:

$$f_i^2 = \sqrt{\sum_{j\in w_i} (u_j - \bar{u}_i)^2} + \sqrt{\sum_{j\in w_i} (m_j - \bar{m}_i)^2}\,.$$

We denote the background and foreground distributions by $h(f_i \mid v_i)$ and $\bar{h}(f_i \mid \bar{v}_i)$, respectively. They are trained from a set of manually segmented image pairs. Since joint correlation and amount of texture is modelled, the histograms remain valid for new illumination conditions and for new backgrounds. Therefore, the training is done only once, off-line. Once normalized, these histograms model the probability of observing a feature f_i on the backgound or on the foreground. Fig. 3 depicts both distributions. One can see that both distributions are dissociate, especially in highly textured areas.

Figure 2 shows a pair of model and input images, the corresponding texture and correlation images f_i^2 and f_i^1, and the results of applying the histograms to $\boldsymbol{f}$. It is obvious that the correlation measure is only meaningful in textured areas. In uniform areas, because NCC is invariant to illumination, it can not make the difference between a background with some uniform illumination or a uniform foreground.

Both histograms are learnt in the two cases of background and foreground which are related to the latent variable x_i designing one of the distributions of our model. Therefore, h can be used together with all background distributions

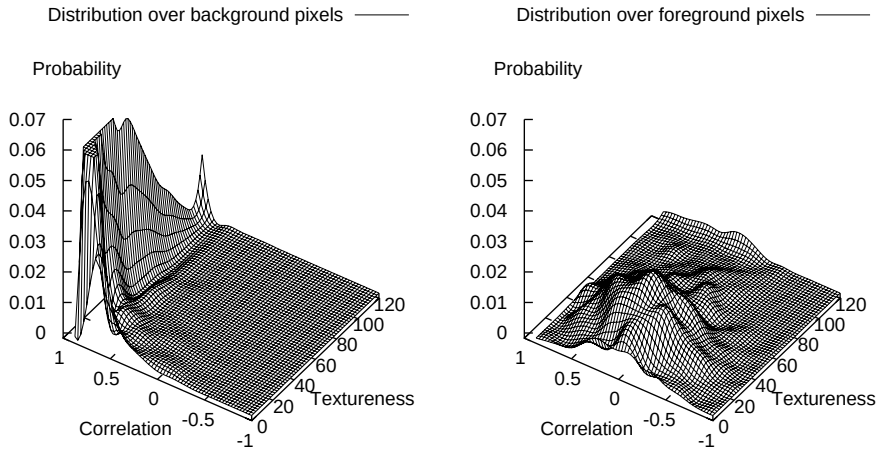

Fig. 3. Joint correlation and texturedness distributions over background and foreground pixels.

corresponding to $\{x_{i,1}, ..., x_{i,K}\}$ and $\bar{h}$ with all foreground ones, corresponding to $\{x_{i,K+1}, ..., x_{i,K+\bar{K}+1}\}$.

3.3 Maximum likelihood estimation

Having defined the illumination and the spatial likelihood model we are now in the position to describe the Maximum Likelihood (ML) estimation of the combined model. Let $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}\}$ denote the vector of all unknowns. The ML estimate $\tilde{\boldsymbol{\theta}}$ is given by:

$$\tilde{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \left\{ \log \sum_{\boldsymbol{x}} p(\boldsymbol{u}, \boldsymbol{f}, \boldsymbol{x} \,|\, \boldsymbol{\theta}) \right\} \tag{4}$$

where $p(\boldsymbol{u}, \boldsymbol{f}, \boldsymbol{x} \,|\, \boldsymbol{\theta}) = p(\boldsymbol{u}, \boldsymbol{x} \,|\, \boldsymbol{\theta}) p(\boldsymbol{f}, \boldsymbol{x} \,|\, \boldsymbol{\theta})$ represents the combined pdf of the illumination and the spatial likelihood models given by the product of eqs. 3, 2 and the histrogram distributions $h(f_i \,|\, v_i)$, $\bar{h}(f_i \,|\, \bar{v})$. Since the histogram distributions are computed over an image patch, the pixel contributions are not independent. However, in order to reach the real-time constraints, we assume the factorisation over all pixels i in Eq. 4 to be approximately true. We see this problem as a trade-off between *(i)* a prior model on $\boldsymbol{x}$, that models spatial interactions [12, 15] with a higher computational complexity and *(ii)* a more simple, real time model for which the independence assumption is violated, in the hope that the spatially dependent feature description $\boldsymbol{f}$ account for pixel dependence.

The pixel independence assumption simplifies the ML estimate to:

$$\tilde{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \left\{ \log \prod_{i} \sum_{x_i} p(u_i, l_i, f_i, x_i \,|\, \boldsymbol{\theta}) \right\} \tag{5}$$

The expectation-maximization (EM) algorithm can maximize equation 5. It alternates the computation between an expectation step (E-step), and a maximization step (M-step).

E-Step: On the $(t+1)^{th}$ iteration the conditional expectation $\boldsymbol{b}^{t+1}$ of the log-likelihood *w.r.t.* the posterior $p(\boldsymbol{x} \,|\, \boldsymbol{u}, \boldsymbol{\theta})$ is computed in the E-step. By construction, *i.e.* by the pixel independence, this leads to a closed-form solution for the latent variable expectations b_i, which are often called beliefs. Note, that in other formulations, where the spatial correlation is modelled explicitly, the E-step requires graph-cut optimisation [14] or other iterative approximations like mean field [15]. The update equations for the expected values $b_{i,k}$ of $x_{i,k}$ are given by:

$$b_{i,k=1\ldots K}^{t+1} = \frac{1}{N}\pi_k \frac{1}{\mid J_i \mid}\mathcal{N}(l_i; \mu_k^t, \Sigma_k^t) h(f_i \,|\, v_i) \tag{6}$$

$$b_{i,k=K+1\ldots \bar{K}}^{t+1} = \frac{1}{N}\pi_k \mathcal{N}(u_i; \mu_k^t, \Sigma_k^t) \bar{h}(f_i \,|\, \bar{v}_i) \tag{7}$$

$$b_{i,\bar{K}+1}^{t+1} = \frac{1}{N}\pi_{K+\bar{K}+1} \frac{1}{256^3} \bar{h}(f_i \,|\, \bar{v}_i) \;,$$

where $N = \sum_k b_{i,k}^{t+1}$ normalises the beliefs $b_{i,k}^{t+1}$ to one. The first line in Eq. 6 corresponds to the beliefs that the k^{th} normal distribution of the illumination background model is active for pixel i. Similarly, the other two lines (Eq. 7) correspond to the beliefs *w.r.t.* for the foreground illumination model.

M-Step: Given the beliefs $b_{i,k}^{t+1}$, the M-step maximises the log-likelihood by replacing the binary latent variables $x_{i,k}$ by their expected value $b_{i,k}^{t+1}$.

$$\mu_k^{t+1} = \begin{cases} \frac{1}{N_k}\sum_{i=1}^N b_{i,k}^{t+1} l_i & \text{if } k \leq K \\ \frac{1}{N_k}\sum_{i=1}^N b_{i,k}^{t+1} u_i & \text{otherwise} \end{cases} \;, \tag{8}$$

where $N_k = \sum_{i=1}^N b_{ik}^{t+1}$. Similarly, we obtain:

$$\Sigma_k^{t+1} = \begin{cases} \frac{1}{N_k}\sum_{i=1}^N b_{i,k}^{t+1} (l_i - \mu_k)(l_i - \mu_k)^{\mathrm{T}} & \text{if } k \leq K \\ \frac{1}{N_k}\sum_{i=1}^N b_{i,k}^{t+1} (u_i - \mu_k)(u_i - \mu_k)^{\mathrm{T}} & \text{otherwise} \end{cases} \tag{9}$$

$$\pi_k^{t+1} = \frac{N_k}{\sum_k N_k} \tag{10}$$

Alternating E and M steps ensure convergence to a local minimum. After convergence, we can compute the segmentation by summing the beliefs corresponding to the foreground and the background model. The probability of a pixel beeing described by the background model is therefore given by:

$$p(v_i \,|\, \tilde{\boldsymbol{\theta}}, \boldsymbol{u}) = \sum_{k=1}^{K} b_{i,k} \,. \tag{11}$$

In the next section, we discuss implementation and performance issues.

3.4 Implementation details

Our algorithm can be used in two different manners. First, it can run on-line, with a single E-M iteration at each frame, which allows fast computation. On very abrupt illumination changes, convergence is reached after a few frames (rarely more than 6). Second, the algorithm can run offline, with only two images as input instead of a video history. In this case, several iterations, typically 5 to 10, are necessary before convergence.

Local NCC can be computed efficiently with integral images, with a complexity linear with respect to the number of pixels and constant with respect to the window size. Thus, the complexity of the complete algorithm is also linear with the number of pixels, and the full process of acquiring, segmenting, and displaying images is achieved at a rate of about 2.3×10^6 pixels per second, using a single core of a 2.0GHz CPU. This is about 18 fps for half PAL (360x288), 12 FPS for 512x384, and 5-6 FPS for 720x576 images.

Correlation and texturedness images, as presented in section 3.2, are computed from single channel images. We use the green channel only, because it is more represented on a Bayer pattern. The correlation window is a square of 25×25 pixels, cropped at image borders.

For all experiments presented in the paper, $K = 2$ and $\bar{K} = 2$. The histograms h and $\bar{h}$ have been computed only once, from 9 pairs of images (about 2×10^6 training pixels). Training images do not contain any pattern or background used in test experiments.

The function l_i as presented in previous section is sensitive to limited dynamic range and to limited precision in low intensity values. Both following functions assume the same role with more robustness and give good result:

$$l_i^a(u_i) = \left[\arctan\left(\frac{u_{i,r}}{m_{i,r}}\right) \arctan\left(\frac{u_{i,g}}{m_{i,g}}\right) \arctan\left(\frac{u_{i,b}}{m_{i,b}}\right)\right]^T$$

$$l_i^c(u_i) = \left[\frac{u_{i,r}+c}{m_{i,r}+c} \frac{u_{i,g}+c}{m_{i,g}+c} \frac{u_{i,b}+c}{m_{i,b}+c}\right]^T$$

where c is an arbitrary positive constant. In our experiments, we use $c = 64$.

4 Results

In this section, we show results on individual frames of video sequences that feature both sudden illumination changes and shadows cast by occluding objects. We also compare those results to those produced by state-of-the-art techniques [1, 11]. We supply the original videos and corresponding results as supplementary material.

4.1 Robustness of Illumination Changes and Shadows

We begin by the sequence of Fig. 5 in which an arm is waved in front of a cluttered wall. The arm casts a shadow, which affects the scene's radiosity and

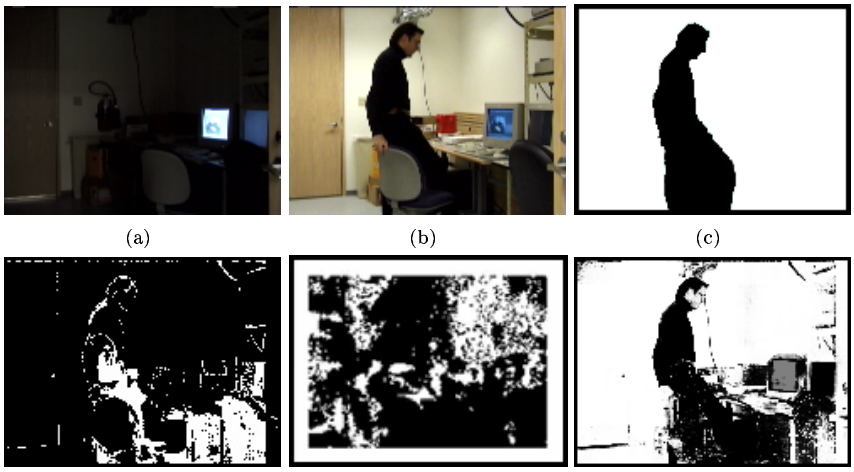

Fig. 4. Segmenting the *light switch* test images from [16]. (a) Background model. (b) Test image. (c) Manually segmented ground truth. (d) The output of Zivkovic's method [1]. (e) Result published in [11], using an approach based on local binary patterns. (f) Our result, obtained solely by comparing (a) and (b). Unlike the other two methods, we used no additional video frames.

causes the camera to automatically adapt its luminosity settings. With default parameters, the algorithm of [1] reacts to this by slowly adapting its background model. However, this adaptation cannot cope with the rapidly moving shadow and produces the poor result of Fig. 5(a). This can be prevented by increasing the rate at which the background adapts, but, as shown in Fig. 5(b), it results in the sleeve being lost. By contrast, by explicitly reevaluating the illumination parameters at every frame, our algorithm copes much better with this situation, as shown in Fig. 5(c). To compare these two methods independently of specific parameter choices, we computed the ROC curve of Fig. 5(d). We take precision to be the number of pixels correctly tagged as foreground divided by the total number of pixels marked as foreground and recall to be the number of pixels tagged as foreground divided by the number of foreground pixels in the ground truth. The curve is obtained by binarizing using different thresholds for the probability of Eq. 11. We also represent different runs of [1] by crosses corresponding to different choices of its learning rate and the decision threshold. As expected, our method exhibits much better robustness towards illumination effects.

Fig. 1 depicts a sequence with even more drastic illumination changes that occur when the subject turns on one light after the other. The GMM based-method [1] immediately reacts by classifying most of the image as foreground. By contrast, our algorithm correctly compares the new images with the background image, taken to be the average of the first 25 frames of the sequence.

Fig. 4 shows the *light switch* benchmark of [16]. We again built the background representation by averaging 25 consecutive frames showing the room

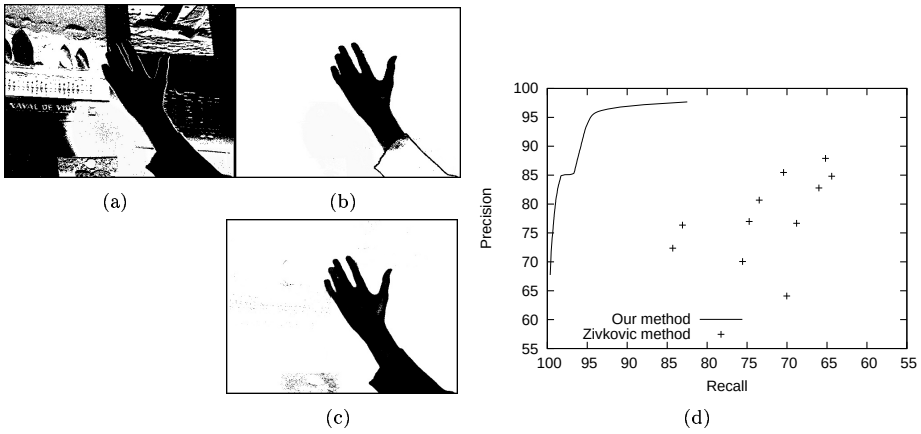

Fig. 5. Segmenting the hand of Fig. 2(b). (a) Result of [1] when the background model adjusts too slowly to handle a quick illumination change. (b) When the background model adjusts faster. (d) ROC curve for our method obtained by varying a threshold on the probability of Eq. 11. The crosses represent results obtained by [1] for different choices of learning rate and decision threshold.

with the light switched off. We obtain good results when comparing it to an image where the light is turned on even though, unlike the other algorithms [1, 11], we use a single frame instead of looking at the whole video. To foreground recall of 82% that appears in [11] entails a precision of only 25%, whereas our method achieves 49% for the same recall. With default parameters, the algorithm of [1] cannot handle this abrupt light change and yields a precision of 13% for a recall of 70%.

Finally, as shown in Fig. 6, we ran our algorithm on one of the PETS 2006 video sequences that features an abandoned luggage to demonstrate that our technique is indeed appropriate for surveillance applications because it does not lose objects by unduly merging them in the background.

Fig. 6. PETS 2006 Dataset. (a) Initial frame of the video, used as background model. (b) Frame number 2800. (c) The background subtraction of [1]: The abandoned bag in the middle of the scene has mistakenly been integrated into the background. (d) Our method correctly segment the bag, the person who left after sitting on the bottom left corner, and the chair that has been removed on the right.

(a) (b) (c) (d)

Fig. 7. Occlusion segmentation on a moving object. (a) Input frame in which the card is tracked. (b) Traditional background subtraction provide unsatisfying results because of the shadow cast by the hand, and because it learned the fingers hiding the bottom left corner as part of the background. (c): Our method is far more robust and produces a better segmentation. (d) We use its output as an alpha channel to convincingly draw the virtual text and account for the occluding hand.

4.2 Augmented Reality

Because our approach is very robust to abrupt illumination changes, it is a perfect candidate for occlusion segmentation in augmented reality. The task is the following: A user holds an object that is detected and augmented. If the detected pattern is occluded by a real object, the virtual object should also be occluded. In order to augment only the pixels actually showing the pattern, a visibility mask is required. Technically, any background subtraction technique could produce it, by unwarping the input images in a reference frame, and by rewarping the resulting segmentation back to the input frame.

The drastic illumination changes produced by quick rotation of the pattern might hinder a background subtraction algorithm that has not been designed for such conditions. That is why the Gaussian mixture based background subtraction method of [1] has difficulties to handle our test sequence illustrated by figure 7. On the other hand, the illumination modeling of our approach is able to handle this situation well and, unsurprisingly, shows superior results. The quality of the resulting segmentation we obtain allows convincing occluded augmented reality, as illustrated by figure 7(d).

5 Conclusion

We presented a fast background subtraction algorithm that handles heavy illumination changes by relying on a statistical model, not of the pixel intensities, but of the illumination effects. The optimized likelihood also fuses texture correlation clues by exploiting histograms trained off-line.

We demonstrated the performance of our approach under drastic light changes that state-of-the-art technique have trouble to handle.

Moreover, our technique can be used to segment the occluded parts of a moving planar object and therefore allows occlusion handling for augmented reality applications.

Although we do not explicitely model spatial consistency, the learnt histograms of correlation captures texture. Similarly, we could easily extend our method by integrating temporal dependence using temporal features.

References

1. Zivkovic, Z., van der Heijden, F.: Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern Recognition Letters **27**(7) (2006) 773–780
2. Wren, C., Azarbayejani, A., Darrell, T., Pentland, A.: Pfinder: Real-time tracking of the human body. In: Photonics East, SPIE. Volume 2615. (1995)
3. Friedman, N., Russell, S.: Image segmentation in video sequences: A probabilistic approach. In: Annual Conference on Uncertainty in Artificial Intelligence. (1997) 175–181
4. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: CVPR. (1999) 246–252
5. Bishop, C.: Pattern Recognition and Machine Learning. Springer (2006)
6. Elgammal, A., Duraiswami, R., Harwood, D., Davis, L.: Background and foreground modeling using nonparametric kernel density for visual surveillance. In: Proceedings of the IEEE. Volume 90. (July 2002) 1151–1163
7. Sheikh, Y., Shah, M.: Bayesian modeling of dynamic scenes for object detection. PAMI **27** (2005) 1778–1792
8. Prati, A., Mikic, I., Trivedi, M., Cucchiara, R.: Detecting moving shadows: Algorithms and evaluation. PAMI **25** (2003) 918–923
9. Stauder, J., Mech, R., Ostermann, J.: Detection of moving cast shadows for object segmentation. IEEE Transactions on Multimedia **1**(1) (1999) 65–76
10. Jabri, S., Duric, Z., Wechsler, H., Rosenfeld, A.: Detection and location of people in video images using adaptive fusion of color and edge information. In: International Conference on Pattern Recognition. (2000) 627–630 vol.4
11. Heikkila, M., Pietikainen, M.: A texture-based method for modeling the background and detecting moving objects. PAMI **28**(4) (April 2006) 657–662
12. Criminisi, A., Cross, G., Blake, A., Kolmogorov, V.: Bilayer segmentation of live video. In: CVPR. (2006) 53–60
13. : Bazar http://cvlab.epfl.ch/software/bazar.
14. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. In: ACM SIGGRAPH. (2004)
15. Fransens, R., Strecha, C., Van Gool, L.: A mean field EM-algorithm for coherent occlusion handling in map-estimation problems. In: CVPR. (2006)
16. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: principles and practice of background maintenance. In: International Conference on Computer Vision. (1999) 255–261 vol.1