Available online at www.sciencedirect.com
SCIENCE DIRECT®
Pattern Recognition Letters 27 (2006) 773–780
Pattern Recognition Letters
www.elsevier.com/locate/patrec

Efficient adaptive density estimation per image pixel for the task of background subtraction

Zoran Zivkovic [a,*], Ferdinand van der Heijden [b]

[a] Faculty of Science, University of Amsterdam, Kruislaan 403, 1098SJ Amsterdam, The Netherlands
[b] University of Twente, P.O. Box 217, 7500AE Enschede, The Netherlands

Received 5 July 2004; received in revised form 17 August 2005
Available online 6 January 2006

Communicated by Prof. M. Lindenbaum

Abstract

We analyze the computer vision task of pixel-level background subtraction. We present recursive equations that are used to constantly update the parameters of a Gaussian mixture model and to simultaneously select the appropriate number of components for each pixel. We also present a simple non-parametric adaptive density estimation method. The two methods are compared with each other and with some previously proposed algorithms.

Keywords: Background subtraction; On-line density estimation; Gaussian mixture model; Non-parametric density estimation

1. Introduction

A static camera observing a scene is a common case of a surveillance system. Detecting intruding objects is an essential step in analyzing the scene. An usually applicable assumption is that the images of the scene without the intruding objects exhibit some regular behavior that can be well described by a statistical model. If we have a statistical model of the scene, an intruding object can be detected by spotting the parts of the image that do not fit the model. This process is usually known as "background subtraction".

In the case of common pixel-level background subtraction the scene model has a probability density function for each pixel separately. A pixel from a new image is considered to be a background pixel if its new value is well described by its density function. For a static scene the simplest model could be just an image of the scene without the intruding objects. The next step would be, for example, to estimate appropriate values for the variances of the pixel intensity levels from the image since the variances can vary from pixel to pixel. This single Gaussian model was used in (Wren et al., 1997). However, pixel values often have complex distributions and more elaborate models are needed.

A Gaussian mixture model (GMM) was proposed for the background subtraction in (Friedman and Russell, 1997) and efficient update equations are given in (Stauffer and Grimson, 1999). In (Power and Schoonees, 2002) the GMM is extended with a hysteresis threshold. In (Hayman and Eklundh, 2003) the GMM approach was applied to pan-tilt cameras. The standard GMM update equations are extended in (KaewTraKulPong and Bowden, 2001; Lee, 2005) to improve the speed of adaptation of the model. All these GMMs use a fixed number of components. In (Stenger et al., 2001) the topology and the number of components of a hidden Markov model was selected in an off-line training procedure. The first contribution of this paper is an improved GMM algorithm based on the recent results from Zivkovic and van der Heijden (2004). We show, from a Bayesian perspective, how to use a model

* Corresponding author. Tel.: +31 20 525 7564; fax: +31 20 525 7490.
E-mail address: zivkovic@science.uva.nl (Z. Zivkovic).

0167-8655/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2005.11.005

selection criterion to choose the right number of components for each pixel on-line and in this way automatically fully adapt to the scene.

The non-parametric density estimates also lead to flexible models. The kernel density estimate was proposed for background-subtraction in (Elgammal et al., 2000). A problem with the kernel estimates is the choice of the fixed kernel size. This problem can be addressed using the variable-size kernels (Wand and Jones, 1995). Two simple approaches are: the "balloon estimator" adapts the kernel size at each estimation point; and the "sample-point estimator" adapts the kernel size for each data point. In (Mittal and Paragios, 2004) an elaborate hybrid scheme is used. As the second contribution of the paper, we use here the balloon variable-size kernel approach. We use uniform kernels for simplicity. The balloon approach leads to a very efficient implementation that is equivalent to using a fixed uniform kernel (see Section 4). Finally, as the third contribution, we analyze and compare the standard algorithms (Stauffer and Grimson, 1999; Elgammal et al., 2000) and the newly proposed algorithms.

The paper is organized as follows. In the next section, we state the problem of the pixel-based background subtraction. In Section 3, we review the GMM approach from Stauffer and Grimson (1999) and present how the number of components can be selected on-line to improve the algorithm. In Section 4, we review the non-parametric kernel-based approach from Elgammal et al. (2000) and propose a simplification that leads to better experimental results. In Section 5, we give the experimental results and analyze them.

2. Problem definition

The value of a pixel at time t in RGB is denoted by $\vec{x}^{(t)}$. Some other color space or some local features could also be used. For example, in (Mittal and Paragios, 2004) normalized colors and optical flow estimates were used. The pixel-based background subtraction involves decision if the pixel belongs to the background (BG) or some foreground object (FG). The pixel is more likely to belong to the background if

$$\frac{p(\mathrm{BG}|\vec{x}^{(t)})}{p(\mathrm{FG}|\vec{x}^{(t)})} = \frac{p(\vec{x}^{(t)}|\mathrm{BG})p(\mathrm{BG})}{p(\vec{x}^{(t)}|\mathrm{FG})p(\mathrm{FG})}, \tag{1}$$

is larger then 1 and vice versa. The results from the background subtraction are usually propagated to some higher level modules, for example, the detected objects are often tracked. While tracking an object we could obtain some knowledge about the appearance of the tracked object and this knowledge could be used to improve the background subtraction. This is discussed, for example, in (Harville, 2002; Withagen et al., 2002). In the general case we do not know anything about the foreground objects that can be seen nor when and how often they will be present. Therefore we assume a uniform distribution for the appearance of the foreground objects $p(\vec{x}^{(t)}|\mathrm{FG})$. The decision that a pixel belongs to the background is made if

$$p(\vec{x}^{(t)}|\mathrm{BG}) > c_{\mathrm{thr}} (= p(\vec{x}^{(t)}|\mathrm{FG})p(\mathrm{FG})/p(\mathrm{BG})), \tag{2}$$

where c_{thr} is a threshold value. We will refer to $p(\vec{x}|\mathrm{BG})$ as the background model. The background model is estimated from a training set $\mathcal{X}$. The estimated model is denoted by $\hat{p}(\vec{x}|\mathcal{X}, \mathrm{BG})$ and depends on the training set as denoted explicitly. In practice, the illumination in the scene could change gradually (daytime or weather conditions in an outdoor scene) or suddenly (switching the light off or on in an indoor scene). A new object could be brought into the scene or a present object removed from it. In order to adapt to these changes we can update the training set by adding new samples and discarding the old ones. We assume that the samples are independent and the main problem is how to efficiently estimate the density function on-line. There are models in the literature that consider the time aspect of an image sequence and then the decision depends also on the previous pixel values from the sequence. For example, in (Toyama et al., 1999; Monnet et al., 2003) the pixel value distribution over time is modelled as an autoregressive process. In (Stenger et al., 2001; Kato et al., 2002) hidden Markov models are used. However, these methods are usually much slower and adaptation to changes of the scene is difficult.

The pixel-wise approaches assume that the adjacent pixels are uncorrelated. Markov random field can be used to model the correlation between the adjacent pixel values (Kato et al., 2002) but leads to slow and complex algorithms. Some additional filtering of the segmented images often improves the results since it imposes some correlation (Elgammal et al., 2000; Cemgil et al., 2005). Another related subject is the shadow detection. The intruding object can cast shadows on the background. Usually, we are interested only in the object and the pixels corresponding to the shadow should be detected (Prati et al., 2003). In this paper, we analyze the pure pixel-based background subtraction. For the various applications some of the mentioned additional aspects and maybe some postprocessing steps might be important and could lead to improvements but this is out of the scope of this paper.

3. Gaussian mixture model

In order to adapt to possible changes the training set should be updated. We choose a reasonable time adaptation period T. At time t we have $\mathcal{X}_T = \{x^{(t)}, \ldots, x^{(t-T)}\}$. For each new sample we update the training data set $\mathcal{X}_T$ and reestimate the density. These samples might contain values that belong to the foreground objects. Therefore, we should denote the estimated density as $\hat{p}(\vec{x}^{(t)}|\mathcal{X}_T, \mathrm{BG}+\mathrm{FG})$. We use a GMM with M components:

$$\hat{p}(\vec{x}|\mathcal{X}_T, \mathrm{BG}+\mathrm{FG}) = \sum_{m=1}^{M} \hat{\pi}_m \mathcal{N}(\vec{x}; \hat{\vec{\mu}}_m, \hat{\sigma}_m^2 I), \tag{3}$$

where $\hat{\vec{\mu}}_1, \ldots, \hat{\vec{\mu}}_M$ are the estimates of the means and $\hat{\sigma}_1^2, \ldots, \hat{\sigma}_M^2$ are the estimates of the variances that describe the Gaussian components. For computational reasons the covariance matrices are kept isotropic. The identity matrix I has proper dimensions. The estimated mixing weights denoted by $\hat{\pi}_m$ are non-negative and add up to one.

3.1. Update equations

Given a new data sample $\vec{x}^{(t)}$ at time t the recursive update equations are (Titterington, 1984):

$$\hat{\pi}_m \leftarrow \hat{\pi}_m + \alpha(o_m^{(t)} - \hat{\pi}_m), \tag{4}$$

$$\hat{\vec{\mu}}_m \leftarrow \hat{\vec{\mu}}_m + o_m^{(t)}(\alpha/\hat{\pi}_m)\vec{\delta}_m, \tag{5}$$

$$\hat{\sigma}_m^2 \leftarrow \hat{\sigma}_m^2 + o_m^{(t)}(\alpha/\hat{\pi}_m)(\vec{\delta}_m^T\vec{\delta}_m - \hat{\sigma}_m^2), \tag{6}$$

where $\vec{\delta}_m = \vec{x}^{(t)} - \hat{\vec{\mu}}_m$. Instead of the time interval T that was mentioned above, here the constant α defines an exponentially decaying envelope that is used to limit the influence of the old data. We keep the same notation having in mind that effectively $\alpha = 1/T$. For a new sample the ownership $o_m^{(t)}$ is set to 1 for the "close" component with largest $\hat{\pi}_m$ and the others are set to zero. We define that a sample is "close" to a component if the Mahalanobis distance from the component is, for example, less than three. The squared distance from the mth component is calculated as: $D_m^2(\vec{x}^{(t)}) = \vec{\delta}_m^T\vec{\delta}_m/\hat{\sigma}_m^2$. If there are no "close" components a new component is generated with $\hat{\pi}_{M+1} = \alpha$, $\hat{\vec{\mu}}_{M+1} = \vec{x}^{(t)}$ and $\hat{\sigma}_{M+1} = \sigma_0$ where σ_0 is some appropriate initial variance. If the maximum number of components is reached we discard the component with smallest $\hat{\pi}_m$.

The presented algorithm presents an on-line clustering algorithm. Usually, the intruding foreground objects will be represented by some additional clusters with small weights $\hat{\pi}_m$. Therefore, we can approximate the background model by the first B largest clusters:

$$\hat{p}(\vec{x}|\mathscr{X}_T, \mathrm{BG}) \sim \sum_{m=1}^{B} \hat{\pi}_m \mathscr{N}(\vec{x}; \hat{\vec{\mu}}_m, \sigma_m^2 I). \tag{7}$$

If the components are sorted to have descending weights (slightly different ordering is originally used in (Stauffer and Grimson, 1999)) $\hat{\pi}_m$, we have

$$B = \arg\min_b \left(\sum_{m=1}^{b} \hat{\pi}_m > (1 - c_\mathrm{f}) \right), \tag{8}$$

where c_f is a measure of the maximum portion of the data that can belong to foreground objects without influencing the background model. For example, if a new object comes into a scene and remains static for some time it will be temporally presented as an additional cluster. Since the old background is occluded the weight π_{B+1} of the new cluster will be constantly increasing. If the object remains static long enough, its weight becomes larger than c_f and it can be considered to be part of the background. If we look at (4), we can conclude that the object should be static for approximately $\log(1 - c_\mathrm{f})/\log(1 - \alpha)$ frames. For example, for $c_\mathrm{f} = 0.1$ and $\alpha = 0.001$ we get 105 frames.

3.2. Selecting the number of components

The weight π_m is the fraction of the data that belongs to the mth component of the GMM. It can be regarded as the probability that a sample comes from the mth component and in this way the π_m-s define an underlying multinomial distribution. Let us assume that we have t data samples and each of them belongs to one of the components of the GMM. Let us also assume that the number of samples that belong to the mth component is $n_m = \sum_{i=1}^{t} o_m^{(i)}$ where $o_m^{(i)}$-s are defined in the previous section. The assumed multinomial distribution for n_m-s gives a likelihood function $\mathscr{L} = \prod_{m=1}^{M} \pi_m^{n_m}$. The mixing weights are constrained to sum up to one. We take this into account by introducing the Lagrange multiplier λ. The Maximum Likelihood (ML) estimate follows from: $\frac{\partial}{\partial \hat{\pi}_m}\left(\log \mathscr{L} + \lambda(\sum_{m=1}^{M} \hat{\pi}_m - 1)\right) = 0$. After getting rid of λ, we get

$$\hat{\pi}_m^{(t)} = \frac{n_m}{t} = \frac{1}{t}\sum_{i=1}^{t} o_m^{(i)}. \tag{9}$$

The estimate from t samples is denoted as $\hat{\pi}_m^{(t)}$ and it can be rewritten in a recursive form as a function of the estimate $\hat{\pi}_m^{(t-1)}$ for $t-1$ samples and the ownership $o_m^{(t)}$ of the last sample:

$$\hat{\pi}_m^{(t)} = \hat{\pi}_m^{(t-1)} + \frac{1}{t}(o_m^{(t)} - \hat{\pi}_m^{(t-1)}). \tag{10}$$

If we now fix the influence of the new samples by fixing $1/t$ to $\alpha = 1/T$ we get the update Eq. (4). This fixed influence of the new samples means that we rely more on the new samples and the contribution from the old samples is down-weighted in an exponentially decaying manner as mentioned before.

Prior knowledge for multinomial distribution can be introduced by using its conjugate prior, the Dirichlet prior $\mathscr{P} = \prod_{m=1}^{M} \pi_m^{c_m}$. The coefficients c_m have a meaningful interpretation. For the multinomial distribution, the c_m presents the prior evidence (in the maximum a posteriori (MAP) sense) for the class m—the number of samples that belong to that class a priori. As in (Zivkovic and van der Heijden, 2004), we use negative coefficients $c_m = -c$. Negative prior evidence means that we will accept that the class m exists only if there is enough evidence from the data for the existence of this class. This type of prior is also related to the Minimum Message Length criterion that is used for selecting proper models for given data (Zivkovic and van der Heijden, 2004). The MAP solution that includes the mentioned prior follows from $\frac{\partial}{\partial \hat{\pi}_m}(\log \mathscr{L} + \log \mathscr{P} + \lambda(\sum_{m=1}^{M} \hat{\pi}_m - 1)) = 0$, where $\mathscr{P} = \sum_{m=1}^{M} \pi_m^{-c}$. We get

$$\hat{\pi}_m^{(t)} = \frac{1}{K}\left(\sum_{i=1}^{t} o_m^{(i)} - c\right), \tag{11}$$

where $K = \sum_{m=1}^{M}(\sum_{i=1}^{t} o_m^{(t)} - c) = t - Mc$. We rewrite (11) as

$$\hat{\pi}_m^{(t)} = \frac{\hat{\Pi}_m - c/t}{1 - Mc/t}, \tag{12}$$

where $\hat{\Pi}_m = \frac{1}{t}\sum_{i=1}^{t} o_m^{(t)}$ is the ML estimate from (9) and the bias from the prior is introduced through c/t. The bias decreases for larger data sets (larger t). However, if a small bias is acceptable we can keep it constant by fixing c/t to $c_T = c/T$ with some large T. This means that the bias will always be the same as if it would have been for a data set with T samples. It is easy to show that the recursive version of (11) with fixed $c/t = c_T$ is given by

$$\hat{\pi}_m^{(t)} = \hat{\pi}_m^{(t-1)} + 1/t\left(\frac{o_m^{(t)}}{1 - Mc_T} - \hat{\pi}_m^{(t-1)}\right) - 1/t\frac{c_T}{1 - Mc_T}. \tag{13}$$

Since we usually expect only a few components M and c_T is small we assume $1 - Mc_T \approx 1$. As mentioned we set $1/t$ to α and get the final modified adaptive update equation

$$\hat{\pi}_m \leftarrow \hat{\pi}_m + \alpha(o_m^{(t)} - \hat{\pi}_m) - \alpha c_T. \tag{14}$$

This equation is used instead of (4). After each update we need to normalize π_m-s so that they add up to one. We start with a GMM with one component centered on the first sample and new components are added as mentioned in the previous section. The Dirichlet prior with negative weights will suppress the components that are not supported by the data and we discard the component m when its weight π_m becomes negative. This also ensures that the mixing weights stay non-negative. For a chosen $\alpha = 1/T$, we could require that at least $c = 0.01 * T$ samples support a component and we get $c_T = 0.01$.

Note that the direct recursive version of (11) given by $\hat{\pi}_m^{(t)} = \hat{\pi}_m^{(t-1)} + (t - Mc)^{-1}(o_m^{(t)}(\vec{x}^{(t)}) - \hat{\pi}_m^{(t-1)})$ is not very useful. We could start with a larger value for t to avoid negative update for small t but then we cancel out the influence of the prior. This motivates the important choice we made to fix the influence of the prior.

4. Non-parametric methods

4.1. Kernel density estimation

Density estimation using a uniform kernel starts by counting the number of samples k from the data set $\mathcal{X}_T$ that lie within the volume V of the kernel. The volume V is a hypersphere with diameter D. The density estimate is given by

$$\hat{p}_{\text{non-parametric}}(\vec{x}|\mathcal{X}_T, \text{BG} + \text{FG}) = \frac{1}{TV}\sum_{m=t-T}^{t} \mathcal{K}\left(\frac{\|\vec{x}^{(m)} - \vec{x}\|}{D}\right) = \frac{k}{TV}, \tag{15}$$

where the kernel function $\mathcal{K}(u) = 1$ if $u < 1/2$ and 0 otherwise. The volume V of the kernel is proportional to D^d where d is the dimensionality of the data. Other smoother kernel functions $\mathcal{K}$ are often used. For example, a Gaussian profile is used in (Elgammal et al., 2000). In practice the kernel form $\mathcal{K}$ has little influence but the choice of D is critical (Wand and Jones, 1995). In (Elgammal et al., 2000) the median med is calculated for the absolute differences $\|\vec{x}^{(t)} - \vec{x}^{(t-1)}\|$ of the samples from $\mathcal{X}_T$ and a simple robust estimate of the standard deviation is used $D = \text{med}/(0.68\sqrt{2})$.

4.2. Simple balloon variable kernel density estimation

The kernel estimation is using one fixed kernel size D for the whole density function which might not be the best choice (Wand and Jones, 1995). The so called "balloon estimator" adapts the kernel size at each estimation point $\vec{x}$. Instead of trying to find the globally optimal D, we could increase the width D of the kernel for each new point $\vec{x}$ until a fixed amount of data k is covered. In this way we get large kernels in the areas with a small number of samples and smaller kernels in the densely populated areas. This estimate is not a proper density estimate since the integral of the estimate is not equal to 1. There are many other more elaborate approaches (Hall et al., 1995). Still the balloon estimate is often used for classification problems since it is related to the k-NN classification (see Bishop, 1995, p. 56). One nearest neighbor is common but to be more robust to outliers we use $k = [0.1T]$ where $[\cdot]$ is the "round-to-integer" operator.

The balloon approach leads to an efficient implementation that is equivalent to using a fixed uniform kernel. Only the choice for the threshold c_{thr} from (2) is different. For both the fixed kernel and the balloon estimate the decision that a new sample $\vec{x}$ fits the model is made if there are more than k points within the volume V (15). The kernel based approach has V fixed and k is the variable parameter that can be used as the threshold $c_{\text{thr}} \sim k$ from (2). For the uniform kernel k is discrete and we get discontinuous estimates. The balloon variable kernel approach in this paper has the k fixed and the volume V is the variable parameter $c_{\text{thr}} \sim 1/V \sim 1/D^d$. The problems with the discontinuities do not occur. An additional advantage is that we do not estimate the sensitive kernel size parameter as in (Elgammal et al., 2000).

4.3. Practical issues

In practice T is large and keeping all the samples in $\mathcal{X}_T$ would require too much memory and calculating (15) would be too slow. It is reasonable to choose a fixed number of samples $K \ll T$ and randomly select a sample from each subinterval T/K. This might give too sparse sampling of the interval T. In (Elgammal et al., 2000) the model is split into a "short-term" model that has K_{short} samples form T_{short} period and a "long-term" model with K_{long} samples form T_{long}. The "short-term" model contains a denser sampling of the recent history. We use a similar "short-term–long-term" strategy as in (Elgammal et al.,

2000). We would like to compare the non-parametric approaches to the GMM approach. Without a proof of optimality we select $K_{\text{short}} = K_{\text{long}} = K/2$. The exponentially decaying envelope defined by the parameter $\alpha = 1/T$ is used to limit the influence of the old data in the GMM approach. The 'short-term' and ''long-term'' models can be seen as a step-like approximation of the envelope. We choose the ''short-term'' model to approximate the first 30% of the information under the envelope and we get $T_{\text{short}} = [\log(0.7)/\log(1-\alpha)]$.

The $\mathcal{X}_T$ contains also samples from the foreground. Therefore, for automatic learning we keep also a set of corresponding indicators $b^{(1)}, \ldots, b^{(T)}$. The indicator $b^{(m)}$ has a value 0 if the sample is assigned to the foreground. The background model considers only the samples with $b^{(m)} = 1$ that were classified to belong to the background:

$$\hat{p}_{\text{non-parametric}}(\vec{x}|\mathcal{X}_T, \text{BG}) \approx \frac{1}{TV} \sum_{m=t-T}^{t} b^{(m)} \mathcal{K}\left(\frac{\|\vec{x}^{(m)} - \vec{x}\|}{D}\right). \tag{16}$$

If this value is greater than the threshold c_{thr} the pixel is classified as background. Eq. (15), which considers all the samples regardless of the $b^{(m)} - s$, is used to determine $b^{(m)}$ for the new sample. If the object remains static than the new samples are expected to be close to each other. For $T = 1000$ ($\alpha = 0.001$), we could expect that (15) becomes greater than the c_{thr} (we use the same threshold) after approximately $k = [0.1T] = 100$ video frames. The samples from the object then start being included into the background model ($b^{(m)}$-s set to 1). This is similar to the automatic learning for the GMM method. Note that this is slightly different from the strategy from (Elgammal et al., 2000) where they proposed to use only the ''long-term'' model to decide when the samples regarded as foreground can be considered as background samples.

5. Experiments

A brief summary of the two algorithms is given in Table 1. To analyze the performance of the algorithms we used three dynamic scenes (Fig. 1). The ''Traffic'' scene sequence has 1000 frames and it was taken with a high-quality camera but under poor light conditions. The ''Lab'' sequence has 845 frames and it is more dynamic. It has a monitor with rolling interference bars in the scene. The plant from the scene was swaying because of the wind. This sequence is taken by a low-quality web-camera. The highly dynamic sequence ''Trees'' is taken from (Elgammal et al., 2000). This sequence has 857 frames. We will analyze only the steady state performance and the performance with slow gradual changes. Therefore, the first 500 frames of the sequences were not used for evaluation and the rest of the frames were manually segmented to generate the ground truth. Some experiments considering adaptation to the sudden changes and the initialization problems can be found in (Toyama et al., 1999, 2001, 2005). For both algorithms and for different threshold values (c_{thr} from (2)), we measured the true positives—percentage of the pixels that belong to the intruding objects that are correctly assigned to the foreground and the false positives—percentage of the background pixels that are incorrectly classified as the foreground. These are results are plotted as the receiver operating characteristic (ROC) curves (Egan, 1975) that are used for evaluation and comparison (Zhang, 1996). For both algorithms, we use $\alpha = 0.001$.

5.1. Improved GMM

We compare the improved GMM algorithm with the original algorithm (Stauffer and Grimson, 1999) with a fixed number of components $M = 4$. In Fig. 1, we demonstrate the improvement in the segmentation results (the ROC curves) and in the processing time. The reported processing time is for 320×240 images and measured on a 2 GHz PC. In the second column of Fig. 1, we also illustrate how the new algorithm adapts to the scene. The gray values in the images indicate the selected number of components per pixel. Black stands for one Gaussian per pixel and a pixel is white if a maximum of 4 components is used. For example, the scene from the ''Lab'' sequence has a monitor with rolling interference bars and the waving plant. We see that the dynamic areas are modelled using more components. Consequently, the processing time also depends on the complexity of the scene. For the highly dynamic ''Trees'' sequence the processing time is close to that of the original algorithm (Stauffer and Grimson, 1999). Intruding objects introduce generation of new components that are removed after some time (see the ''Traffic'' sequence). This also influences the processing speed. For simple scenes like the ''Traffic'' often a single Gaussian

Table 1
A brief summary of the GMM and the non-parametric background subtraction algorithms

General steps	GMM	Non-parametric
Classify the new sample $\vec{x}^{(t)}$ $p(\vec{x}^{(t)}\|\mathcal{X}_T, \text{BG}) > c_{\text{thr}}$	Use (7)	Use (16)
Update $p(\vec{x}\|\mathcal{X}_T, \text{BG} + \text{FG})$	Use (14), (5) and (6), see Section 3 for some practical issues	Add the new sample to $\mathcal{X}_T$ and remove the oldest one, see Section 4.3 for some practical issues
Update $p(\vec{x}\|\mathcal{X}_T, \text{BG})$	Use (8) to select the components of the GMM that belong to the background	If (15) $> c_{\text{thr}}$ use the new sample for $p(\vec{x}\|\mathcal{X}_T, \text{BG})$ (set $b_m = 1$ for the sample)

These steps are repeated for each new video frame.

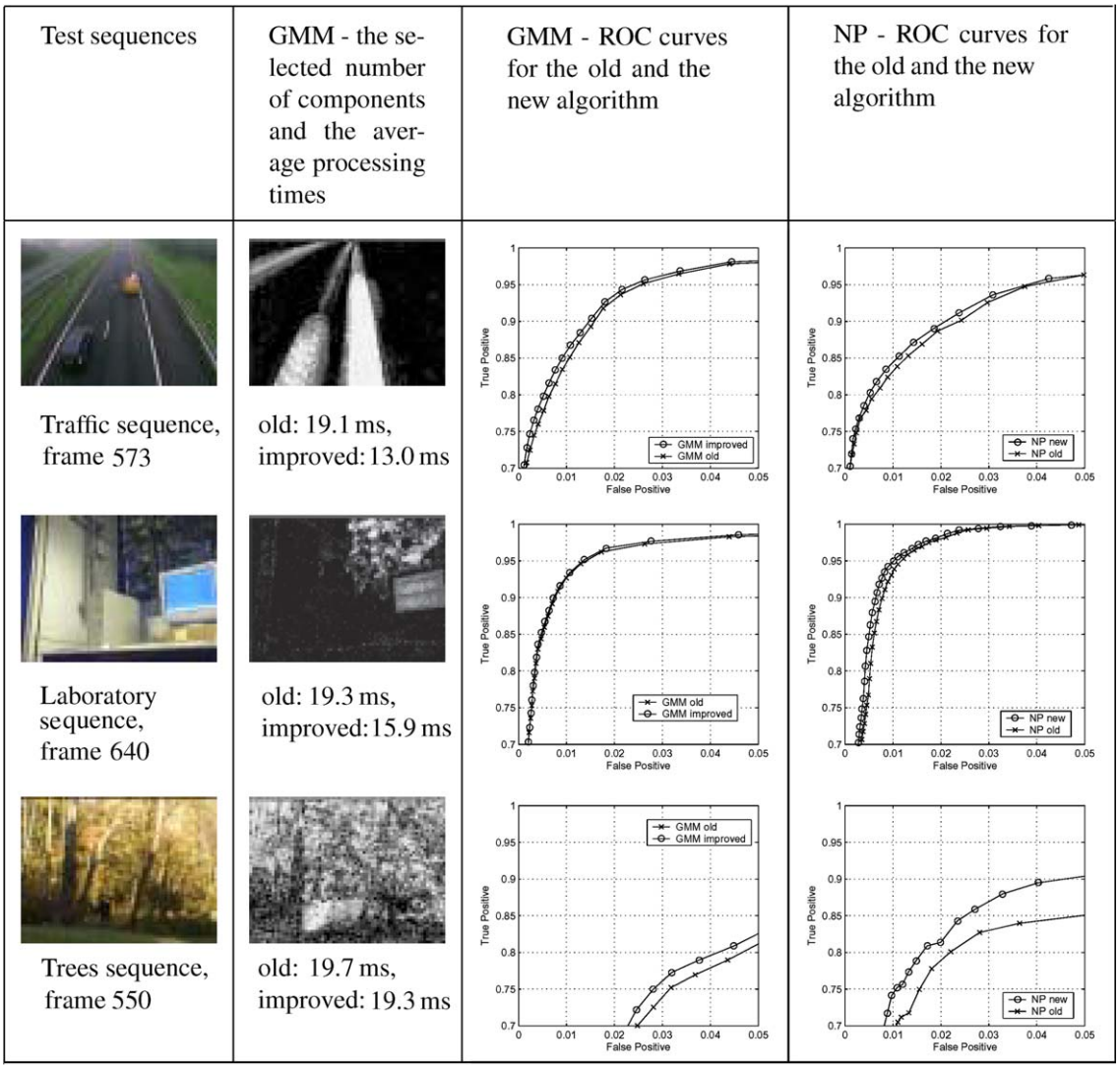

Fig. 1. Comparison of the new proposed methods to the previous methods. The ROC curves are presented for the GMMs and the non-parametric (NP) models. For the new GMM model we also present the selected number of mixture components using the new algorithm. We also report the average processing times in the second column of the table.

per pixel is enough and the processing time is greatly reduced. For complex scenes like "Trees" the segmentation improvement seems to be the greatest. This could be explained by the fact that because of the prior the new GMM is more aggressive in discarding the unimportant components and therefore better in finding the important modes of the GMM.

5.2. Improved non-parametric method

We compare the balloon variable kernel method to the kernel-based method from Elgammal et al. (2000). The ROC curves are reported in the last column of Fig. 1. We used Gaussian kernels for the kernel-based method and uniform for the balloon method. The much simpler new method was constantly giving better segmentation. Again we observe the largest improvements for the complex scene "Trees". This is due to the fact that the small number of samples is widely spread in such cases and the choice of the kernel width D becomes more important. For both methods we used $K = 20$ samples. This choice was made here because the average processing time for this number of samples was similar to the average processing time of the GMM method. Also for other reasonable choices of the number of samples the new method was always performing better. Implementation of the new method is straightforward but for implementing (Elgammal et al., 2000) there are still many choices to be made. Therefore we do not compare the average processing times. However, even without estimating the kernel width in (Elgammal et al., 2000), the new method is much faster since we can use the uniform kernels and still get smooth results for different thresholds (see Section 4.2). Using uniform kernels in (Elgammal et al., 2000) would lead to a very coarse estimate of the density especially in the areas with a small number of samples.

5.3. Comparison

In order to better understand the performance of the algorithms we show the estimated decision boundary for the background models for a pixel in Fig. 2a. The pixel comes from the image area where there was a plant waving because of the wind. This leads to a complex distribution. The GMM tries to cover the data with two isotropic Gaussians. The non-parametric model is more flexible and captures the presented complex distribution more closely. Therefore the nonparametric method usually outperforms the GMM method in complex situations as we can clearly observe in Fig. 2b where we compare the ROC curves of the two new algorithms. However, for a simple scene as the "Traffic" scene the GMM presents also a good model. An advantage of the new GMM is that gives a compact model which might be useful for some further postprocessing like shadow-detection, etc. Isotropic Gaussians leads to crude models as mentioned. Updating full covariance matrices (Zivkovic and van der Heijden, 2004) might give some improvements but this is computationally expensive.

An important parameter that has influence on the performance of the non-parametric method is the number of samples K we use. With more samples we should get better results but the processing time will be increased. For each number of samples (K from 3 to 60) and for each threshold we measure the true positives, false positives and the average processing time. We interpolate this results to get a surface in this 3D fitness-cost space. This surface is presents the best achievable results in the terms of the true positives, false positives and the processing time and it can be regarded as a generalization of the ROC curves. In literature this is a standard way to perform a parameter-free comparison. This surface is called "Pareto front" (Pareto,

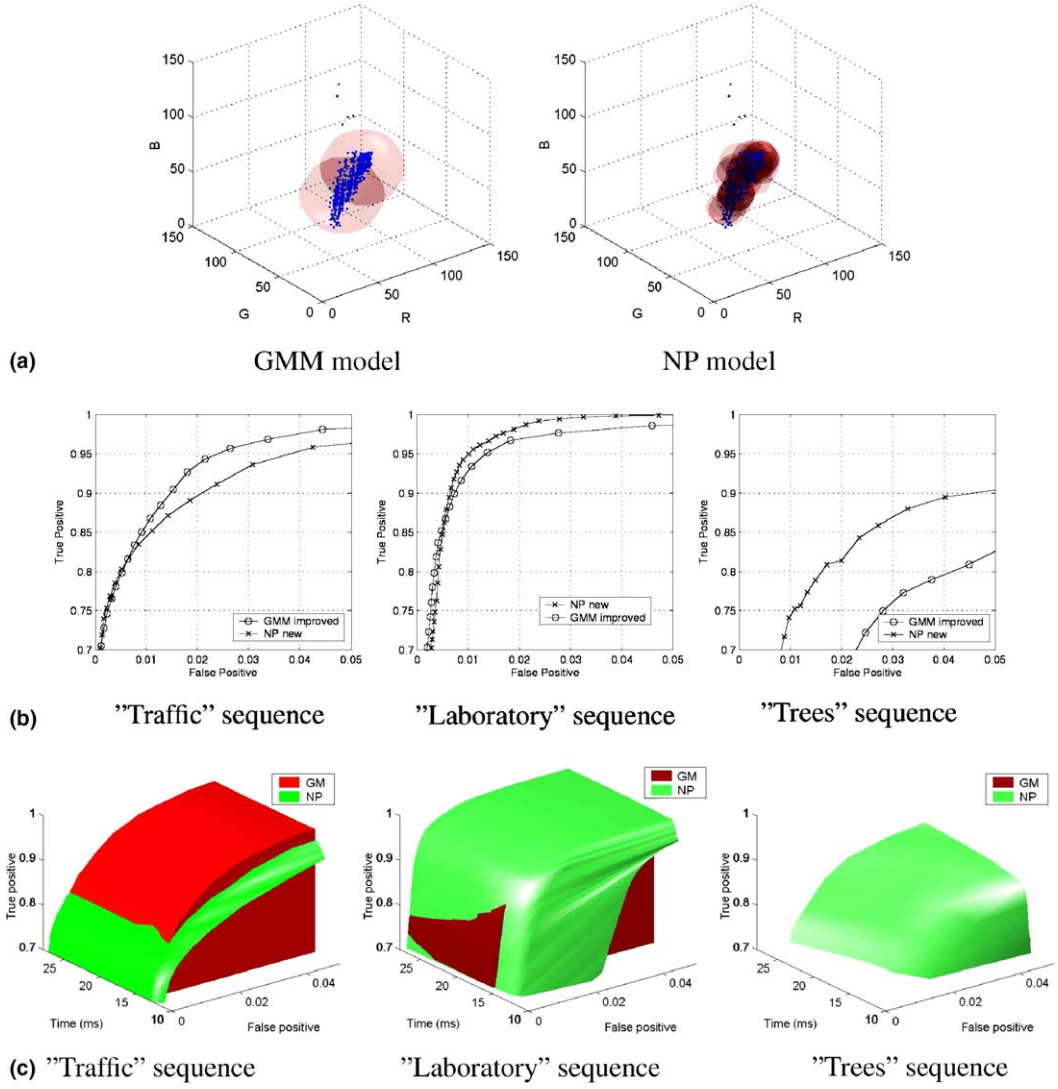

Fig. 2. Comparison of the new GMM algorithm and the new non-parametric (NP) method: (a) an illustration of how the models fit the data. The estimated models are presented for a certain threshold for the frame 840 of the "Laboratory" sequence and for the pixel (283, 53) (the pixel is in the area of the waving plant above the monitor, see Fig. 1), (b) ROC curves for comparison, (c) the convex hull surfaces (Pareto front) that represent the best possible performance of the algorithms for different parameter choices.

1971; Everingham et al., 2002). Since the new GMM has no such parameters we present the best achievable results of the GMM as a cylindrical surface constructed from the ROC curve and cut it off at the average processing time. The Pareto front comparison is presented in Fig. 2c. We observe that if we use more samples the processing time of the non-parametric method is increased and the segmentation is improved as mentioned. However, for the simple "Traffic" sequence the generalization properties of the GMM are still better even when a larger number of samples is used for the non-parametric model. Another conclusion is that we could get a very fast nonparametric algorithm with slightly worse performance if we reduce the number of samples K. This will also reduce the memory requirements for the non-parametric approach.

6. Conclusions

We improved the two common background subtraction schemes presented in (Stauffer and Grimson, 1999; Elgammal et al., 2000). The new GMM algorithm can automatically select the needed number of components per pixel. In this way it can fully adapt to the observed scene. The new kernel method is much simpler than the previously used kernel-based approach. In both cases the processing time is reduced and the segmentation is improved. We also compared the new algorithms. The GMM gives a compact representation which is suitable for further processing. It also seems to be a better model for simple static scenes. The non-parametric approach is very simple to implement and it is a better model for complex dynamic scenes.

References

Bishop, C., 1995. Neural Networks for Pattern Recognition. Oxford University Press.

Cemgil, A.T., Zajdel, W., Krose, B., 2005. A hybrid graphical model for robust feature extraction from video. In: Proc. of the Conf. on Computer Vision and Pattern Recognition.

Egan, J., 1975. Signal Detection Theory and ROC Analysis. Academic Press, New York.

Elgammal, A., Harwood, D., Davis, LS., 2000. Non-parametric background model for background subtraction. In: Proc. of the European Conf. of Computer Vision.

Everingham, M.R., Muller, H., Thomas, B.T., 2002. Evaluating image segmentation algorithms using the Pareto front. In: Proc. of the 7th European Conf. on Computer Vision. pp. 34–48.

Friedman, N., Russell, S., 1997. Image segmentation in video sequences: a probabilistic approach. In: Proc. 13th Conf. on Uncertainty in Artificial Intelligence.

Hall, P., Hui, T.C., Marron, J.S., 1995. Improved variable window kernel estimates of probability densities. Ann. Statist. 23 (1), 1–10.

Harville, M., 2002. A framework for high-level feedback to adaptive, per-pixel, mixture-of-Gaussian background models. In: Proc. of the European Conf. on Computer Vision.

Hayman, E., Eklundh, J.-O., 2003. Statistical background subtraction for a mobile observer. In: Proc. of the Internat. Conf. on Computer Vision. pp. 67–74.

KaewTraKulPong, P., Bowden, R., 2001. An improved adaptive background mixture model for real-time tracking with shadow detection. In: Proc. of 2nd European Workshop on Advanced Video Based Surveillance Systems.

Kato, J., Joga, S., Rittscher, J., Blake, A., 2002. An HMM-based segmentation method for traffic monitoring movies. IEEE Trans. Pattern Anal. Mach. Intell. 24 (9), 1291–1296.

Lee, D.-S., 2005. Effective Gaussian mixture learning for video background subtraction. IEEE Trans. Pattern Anal. Mach. Intell. 27 (5), 827–832.

Mittal, A., Paragios, N., 2004. Motion-based background subtraction using adaptive kernel density estimation. In: Proc. of the Conf. on Computer Vision and Pattern Recognition.

Monnet, A., Mittal, A., Paragios, N., Ramesh, V., 2003. Background modeling and subtraction of dynamic scenes. In: Proc. of the Internat. Conf. on Computer Vision. pp. 1305–1312.

Pareto, V., 1971. Manual of political economy, A.M. Kelley, New York (Original in French 1906).

Power, P.W., Schoonees, J.A., 2002. Understanding background mixture models for foreground segmentation. In: Proc. of the Image and Vision Computing New Zealand.

Prati, A., Mikic, I., Trivedi, M., Cucchiara, R., 2003. Detecting moving shadows: Formulation, algorithms and evaluation. IEEE Trans. Pattern Anal. Mach. Intell. 25 (7), 918–924.

Stauffer, C., Grimson, W., 1999. Adaptive background mixture models for real-time tracking. In: Proc. of the Conf. on Computer Vision and Pattern Recognition. pp. 246–252.

Stenger, B., Ramesh, V., Paragios, N., Coetzec, F., Buhmann, J.M., 2001. Topology free hidden Markov models: application to background modeling. In: Proc. of the Internat. Conf. on Computer Vision.

Toyama, K., Krumm, J., Brumitt, B., Meyers, B., 1999. Wallflower: principles and practice of background maintenance. In: Proc. of the Internat. Conf. on Computer Vision.

Titterington, D., 1984. Recursive parameter estimation using incomplete data. J. Roy. Statist. Soc., Ser. B (Methodological) 2 (46), 257–267.

Wand, M., Jones, M., 1995. Kernel Smoothing. Chapman and Hall, London.

Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A., 1997. Pfinder: Real-time tracking of the human body. IEEE Trans. Pattern Anal. Mach. Intell. 19 (7), 780–785.

Withagen, P.J., Schutte, K., Groen, F., 2002. Likelihood-based object tracking using color histograms and EM. In: Proc. of the Internat. Conf. on Image Processing. pp. 589–592.

Zhang, Y., 1996. A survey on evaluation methods for image segmentation. Pattern Recognition 29, 1335–1346.

Zivkovic, Z., van der Heijden, F., 2004. Recursive unsupervised learning of finite mixture models. IEEE Trans. Pattern Anal. Mach. Intell. 26 (5), 651–656.