FACTA UNIVERSITATIS (NIŠ)
SER.: ELEC. ENERG. vol. 18, No 1, April 2005, 127-144

A Fast Algorithm for Background Tracking in Video Surveillance, Using Nonparametric Kernel Density Estimation

Codruţ Ianăşi, Vasile Gui, Corneliu I. Toma, and Dan Pescaru

Abstract: Moving object detection and tracking in video surveillance systems is commonly based on background estimation and subtraction. For satisfactory performance in real world applications, robust estimators, tolerating the presence of outliers in the data, are needed. Nonparametric kernel density estimation has been successfully used in modeling the background statistics, due to its capability to perform well without making any assumption about the form of the underlying distributions. However, in real-time applications, the O(N^2) complexity of the method can be a bottleneck, preventing the object tracking and event analysis modules from having the computing time needed. In this paper, we propose a new background subtraction technique, using multiresolution and recursive density estimation with mean shift based mode tracking. An algorithm with complexity independent on N is developed for fast, real-time implementation. Comparative results with known methods are included, in order to attest the effectiveness and quality of the proposed approach.

Keywords: Background subtraction, motion detection, tracking, nonparametric kernel density estimation, video surveillance.

1 Introduction

Much work has been done in the area of visual surveillance in the last years [1, 2, 3]. Applications include car and pedestrian traffic monitoring, human activity surveillance for unusual activity detection, people counting etc. A typical surveillance

Manuscript received January 14, 2005.

C. Ianăşi is with S.C. A & C Blue Sys Technologies S.R.L. Timisoara, Romania (e-mail: codrut@bluesys.ro). V. Gui, C. I. Toma and D. Pescaru are with "Politehnica" University of Timisoara, Department of Communication and Department of Computer Science and Engineering, Bd. Vasile Parvan no. 2, 300223 Timisoara, Romania (e-mails: [vasile.gui, ctoma, dan]@etc.utt.ro).

application consists of three building blocks, responsible of: moving object detection, object tracking and higher level motion analysis. While the last two blocks of the system tend to be the most sophisticated, the overall reliability heavily depends on the accuracy and robustness of the moving object detection step. Because the image is usually captured by a stationary camera, it is easier to detect a still background than moving objects.

Despite the importance of the subject and the intensive research done, background detection remains a challenging problem in applications with difficult circumstances, such as changing illumination, waving trees, water, video displays, rotating fans, moving shadows, inter-reflections, camouflage (foreground objects similar to the background), occasional changes of the true background (for example removing an object from the scene), high traffic etc. Such problems cannot be solved by simplistic, static-background models. Various solutions addressing the mentioned problems exist. Some are very computationally extensive and cannot be used in applications requiring real-time operation. Even in the case when visual analysis is used to evaluate later the detected events, the accumulated data may become too much for off-line processing. Therefore, most systems need to do real-time processing in order to be able to keep pace with the video data flow.

Although multimodal systems [4, 5], using more than just one type of data input (such as stereo or multicamera systems), or feedback from the higher level modules may alleviate some of the problems occurring at background subtraction, in the present work we concentrate on the most common case of systems using single, stationary, color cameras and no feedback from higher level modules. We show that a background tracking approach can be used in the nonparametric kernel density estimation paradigm, and describe an efficient algorithm for fast, real-time implementation. The computation complexity is independent of the dimension N of the frame buffer, in contrast with the basic nonparametric density estimation method, with complexity $O(N^2)$. Experimental results are compared with those obtained through the traditional nonparametric kernel density estimation method.

The structure of the rest of the paper is the following. In Chapter 2, the relevant work related to the background estimation problem is discussed, with emphasis on the dominant trend of parametric and nonparametric density estimation techniques. The proposed background estimation is described in Chapter 3, while the results of our experiments are described and discussed in the last chapter of the paper.

2 Related Work

The goal of video surveillance systems is to monitor the activity in a specified, indoor or outdoor area. Since the cameras used in surveillance are typically sta-

tionary, a straightforward way to detect moving objects is to compare each new frame with a reference frame, representing in the best possible way the scene background. By subtracting the background from the current frame in all regions where the current frame matches the reference frame, a segmentation of the moving objects is readily achieved. The results of this process, called background subtraction, are used by the higher level processing modules for object tracking, event detection and scene understanding purposes. Successful background subtraction plays a key role in obtaining reliable results in the higher level processing tasks. This is why many researchers considered carefully the problem of background modeling.

Background modeling is commonly carried out at pixel level. At each pixel, a set of pixel features, collected in a number of frames, is used to build an appropriate model of the local background. Block-based approaches have also been used, mostly in older work, at the expense of resolution. Features used for background modeling can be pixel based, such as intensity or color, local based, such as edges, disparity or depth and region based, such as block correlation. In an ideal situation, the background feature at any pixel is constant in time. In this case, the background feature observed at time index k can be written as:

$$\boldsymbol{x}_k = \boldsymbol{b} + \boldsymbol{n}_k, \tag{1}$$

where $\boldsymbol{b}$ is the unknown background feature vector and $\boldsymbol{n}_k$ is the noise value at observation time k. Suppose a collection of N observation frames are available for the estimation of the background $\boldsymbol{b}$. The estimation problem can be put in the form:

$$\boldsymbol{b} = \arg_{\mathrm{y}} \max\{\boldsymbol{\varepsilon}^2(\boldsymbol{y})\}, \tag{2}$$

with

$$\varepsilon^2(\boldsymbol{y}) = \|\boldsymbol{y}\|^2 = \sum_{k=0}^{N-1} \|\boldsymbol{y} - \boldsymbol{x}_k\|^2 = \sum_{k=0}^{N-1} \|\boldsymbol{n}_k\|^2, \tag{3}$$

which is the well known total least squares estimator, minimizing the sum of squared Euclidean distances from the estimated point to the observed features, or the L^2 norm of noise. By taking the derivative of equation (3), it is straightforward to show that he solution is the mean of the observed feature vectors, or the sample mean:

$$\boldsymbol{y} = \frac{1}{N} \sum_{k=0}^{N-1} \boldsymbol{x}_k. \tag{4}$$

The best way to build such a background model would be to capture the empty scene for a number of frames and take the average frame as the estimated background. Unfortunately, such a scenario is hard to be put in practice in many applications, such as the surveillance at an airport terminal, metro station or on a

highway. Moreover, each time the background changes, the whole scene should be cleared up to repeat the background estimation procedure. Therefore, the background estimation should be carried out during activity in the scene and should be updated to follow background changes occurring in time. As a consequence, the estimation procedure should be designed with a method which is able to tolerate the presence of motion and at the same time to adapt to the true changes of the background.

The presence of moving objects during background estimation involves the presence of pixels in equations (1) - (4) considerably deviating from the real background, called outliers in robust statistics literature [6, 7]. Due to the squaring up operation in equation (3), outliers tend to dominate the sum, causing significant deviation of the estimated feature from the true background. This problem can be alleviated by replacing the sum of squared Euclidean distances in equation (3) with the sum of absolute deviations or L^1 norm. The solution to the new minimization problem is the median sample. Note that finding the median for a large sample set, especially for vector data, is much more computationally expensive than finding the sample mean. Although the median is known to be a robust estimator, its use to background estimation is hindered by the fact that it needs to have at least half of the collected samples taken from the true background. In scenes with heavy traffic, this condition can hardly be met. A more powerful approach to background estimation can be obtained starting from the observation that the true background at a pixel is the most frequently observed feature, consequently the most probable. Even more realistically, we have to suppose that pixel samples collected at instances when the background is not covered by any moving object are affected by noise, due to several factors, like shadows, reflection, camera noise, bulb flickering etc. Therefore, a better way to model the static background is through a random variable or a random vector with an associated probability density function (PDF). In some cases, like trees waving in the background or a rotating fan, more than just one variable should be used for proper background modeling.

The unknown density functions can be represented parametrically, using some specified statistical distributions, and the set of associated parameters minimizing the approximation error with the observed data, can be found through techniques inspired from the statistical literature on parametric estimation methods. A very popular approach is to fit the real data with per-pixel mixtures of Gaussians, as first proposed by Stauffer and Grison [8, 9] and adopted since by many others, like [4, 10, 11, 12]. The strong point of the Gaussian mixture model is that it can work without having to store an important set of input data, as nonparametric methods do. Usually 3 to 5 Gaussians are needed for proper modeling of both background and foreground objects. The Gaussians are weighted by the number of samples

clustered in each of them:

$$\widehat{p}(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k G_k(\boldsymbol{x}; \boldsymbol{\mu}_k, \Sigma_k), \tag{5}$$

where x is the pixel feature vector, π_k are prior probabilities of the Gaussians and μ_k and Σ_k are mean and covariance matrix of the distributions. To reduce the number of parameters that must be estimated, all feature vector components are usually considered independent. A simplified, on-line version of the EM algorithm, introduced by Dempster [13] can be used to obtain fast parameter updating for real-time implementation. Some known problems with this approach are:

- the need for good initializations;
- slow recovery from failures;
- difficult adaptation to fast illumination changes;
- dependence of the results on the true distribution law, which can be non-Gaussian;
- the need to specify the number of Gaussians to be fitted.

Alternatively, the density function modeling the background at each pixel can be obtained through nonparametric kernel density estimation methods [14, 15]. They are known to be able to produce smooth, continuous, differentiable and accurate estimates without having to assume any particular underlying distribution. The number of modes does not have to be known in advance and adaptation to new data is automatic. Although falling in the class of nonparametric methods, kernel density estimation methods still require one scale parameter to be specified or computed from the data. Given a sample of N data points, $\boldsymbol{x}_i$, drawn from a distribution with multivariate probability density function $p(\boldsymbol{x})$, an estimate of this density at $\boldsymbol{x}$ can be written as:

$$\widehat{p}(\boldsymbol{x}) = \frac{1}{N} \sum_{i=1}^{N} K_h(\boldsymbol{x} - \boldsymbol{x}_i), \tag{6}$$

where K_h is the kernel function with scale h. Equation (6) can be seen as a superposition sum of effects of all samples at the currently estimated point, $\boldsymbol{x}$. Common choices for the kernel function are the hypercube and Gaussian shapes. The last one has been successfully used in background and foreground modeling for visual surveillance by Elgamal et al. in [16] and more recently in [17]. Nonparametric methods are less frequently used in visual surveillance applications than the parametric ones, mainly because of their heavier computational load. If the PDF is evaluated at each input point, $\mathrm{O}(N^2)$ operations are needed for a direct computation from the equation (6). One way to reduce the computational load to $\mathrm{O}(2N)$ it

is based on the use fast Gauss transform, data clustering and clever data structures [17].

3 Nonparametric Background Detection and Tracking

Starting from a careful analysis of the use of kernel density estimation technique in the specific case of background modeling, we develop a new algorithm which is at the same time accurate and fast.

3.1 Adopting the proper size of frame buffer

An important simplifying assumption made in the present work is that, in the set of N frames, used for background estimation, the background feature vector is observed at least once within the desired measurement error margin. We consider that in practice, the frame buffer size, should be selected big enough to fulfill such a requirement. As a consequence, the background feature value at any spatial location can be found by selecting one of the N pixels collected at that location in the frame buffer.

When deciding for a big buffer size, N, two other limiting aspects should be considered. The most important one is adaptivity. A big N results in slow adaptation to illumination changes. The second aspect may be computational complexity. For one dimensional data, the dependence of the computational load on N can be avoided by storing and processing a histogram of the input data instead of the real data. If the number of bins, M, is smaller than N, histogram processing is faster. On the other hand, for multichannel data, such as color video input, a histogram with M^3 cells results, making the histogram approach less likely to be faster. In our experiments, we found that values of N in the range of 128-256 worked well in a large variety of surveillance scenes. As M cannot be taken smaller than 64 without introducing significant quantization errors, we ruled out the histogram approach for probability density estimation. Instead, we developed a recursive method to track background changes within the framework of nonparametric kernel density estimation methods and used a roughly quantized 3D histogram to speed up the computations.

3.2 Initial background estimation

We divided our background subtraction task into two stages. The first one is the initial background estimation, while the second one is background tracking. Initial background estimation is carried out only once, when the system is started. A set

of N input frames are accumulated and stored in a frame buffer. No assumption about the absence of foreground object is needed. The input data consists of color vectors of the form:

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix}.$$

At each spatial location, the PDF is estimated by evaluating equation (6) for all N data points from the frame buffer:

$$\widehat{p}(\boldsymbol{x}_k) = \frac{1}{N}\sum_{i=1}^{N} K_h(\boldsymbol{x}_k - \boldsymbol{x}_i), \qquad \text{for} \quad k = 1, 2, ..., N \tag{7}$$

with

$$K_h(\boldsymbol{x}_k - \boldsymbol{x}_i) = \prod_{c=1}^{3} \text{rect}\Big(\frac{x_{kc} - x_{ic}}{h_c}\Big) \tag{8}$$

and

$$\text{rect}(u) = \begin{cases} 1 & \text{if} \quad |u| \leq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

For simplicity, the spatial coordinates of data vectors are omitted. At each pixel, the kernel bandwidth or scale parameters are computed from the equation:

$$h_c = \alpha_{\,i} \text{median}|x_{ic} - x_{(i-1)c}|, \tag{10}$$

as in [16]. The median of absolute differences between intensities from consecutive frames is a robust estimate of the intraclass data variance, as is relatively unaffected by a few high-amplitude jumps expected to occur when edges separating moving objects are projected on a certain pixel. In order to obtain a fast estimation of the scale, an on-line approximation of the median is used. The updating equation for a new sample is:

$$\text{median}(i) = \text{median}(i-1) + \eta\, \text{sign}(x_{ic} - x_{(i-1)c}), \tag{11}$$

with η a small learning rate.

Foreground detection (**F**) can be obtained by thresholding the density functions:

$$\boldsymbol{x}_k \in \boldsymbol{F} \quad \Leftrightarrow] \quad \hat{p}(\boldsymbol{x}_k) < Th, \tag{12}$$

while the maximum likelihood background estimate is

$$\boldsymbol{b} = \arg_{x_k} \max\{\hat{p}(\boldsymbol{x}_k)\}. \tag{13}$$

3.3 Foreground/background segmentation

Although the foreground/background segmentation can be done from equation (12) without explicit extraction of the background distribution mode, **b** may be needed for example in shadow detection and/or as an alternative solution to the segmentation problem. We decided to adopt this second approach, as it separates the segmentation from the density estimation stage, allowing the estimation to be done at a much lower rate than the frame rate needed for motion tracking. Consequently, a pixel **x** can be classified as belonging to the foreground if

$$d(\boldsymbol{x},\boldsymbol{b}) > Th, \tag{14}$$

where $d(\boldsymbol{x},\boldsymbol{b})$ is an appropriate measure of color similarity and Th a threshold. Several solutions to evaluate color similarity exist. The most straightforward is the use of Euclidean norm of the difference vector in the RGB space. Better correspondence with subjective ratings can be obtained using Yuv or Lab color spaces. An additional advantage of these spaces is the direct access to the luminance information, which can be better used in shadow detection. The same is true for the HSV space or the simpler, linear solution:

$$\begin{aligned} s &= (R+G+B)/3, \\ r &= R/s, \\ g &= G/s. \end{aligned} \tag{15}$$

In order to facilitate shadow detection, apparently luminance information should be totally discarded. For example, chroma coefficients r and g could be used without the "sum" term, s. However, such a solution would make gray objects on gray, black or white background undetectable. More, the color of the very dark objects is ill defined. The same is true with very bright objects, leading sometimes to camera saturation. While the shadow detection is a problem on its own and subject of recent research [18], we obtained satisfactory results with equation [15] and scaled L^1 norm:

$$d(\boldsymbol{c}_1,\boldsymbol{c}_2) = |s_1 - s_2| + M|r_1 - r_2| + M|g_1 - g_2|, \tag{16}$$

where M is the maximum value of the R,G,B signals, that is 256 in the present work. Scaling is needed to compensate for the very different range of the r, g and s variables.

Foreground object segmentation masks obtained after thresholding the difference between the current frame and the estimated background, using equation (14) are affected by several sources of errors. Some, such as those produced by the presence of the shadows or camouflage may need special attention and perhaps multimodal sensing. Others consist of scattered groups of pixels forming small regions

representing false objects or holes in the true objects. Such errors can be effectively corrected by spatial filtering. While morphological filtering of the binary segmentation mask is a common approach, we decided to exploit the additional information contained in the difference image. Instead of directly thresholding the difference image and filtering the result, we first filter the difference image and then threshold the result in a processing step called thresholded convolution. To obtain fast convolution, we used a separable 9×9 binomial filter. The horizontal filtering kernel is:

$$\boldsymbol{B}_{\mathrm{H}} = [1, 8, 28, 56, 70, 56, 28, 8, 1]/256. \tag{17}$$

3.4 Fast background tracking

We propose a fast background tracking technique that combines strong points from parametric, nonparametric and histogram based density estimation methods. A straightforward implementation of nonparametric density estimation from a set of N data points involves the evaluation of equations (7) and (13) at each data point, leading to a number of N^2 operations. A careful analysis reveals that, after doing so for the first N frames, a background tracking approach can be used, based on recursive data processing and simple heuristics.

When a new frame is received, a new data point replaces the oldest data point, at each pixel in a buffer of length N. For the unchanged $N-1$ data points, the new densities can be obtained from the old ones recursively, by simply adding the contributions of the new pixel and subtracting the contributions of the old, outgoing pixel:

$$\hat{p}_{new}(\boldsymbol{x}) = \hat{p}_{old}(\boldsymbol{x}) + \frac{1}{N}K_h(\boldsymbol{x}-\boldsymbol{x}_{new}) - \frac{1}{N}K_h(\boldsymbol{x}-\boldsymbol{x}_{old}). \tag{18}$$

This means only two operations per data point, that is $2(N-1)$ operations. For the new pixel, there is no previous estimate of the PDF and apparently the N operations from equation (7) are needed. While recursive evaluation of densities can reduce the processing load from $\mathrm{O}(N^2)$ to $\mathrm{O}(3N)$, more can be done. First, we notice that for most of the old data points not belonging to the background distribution, the density is much lower than for the background. The chances of such points to win the density competition in equation (13), even with the possible contribution of the new data point, are zero. Therefore, an accurate evaluation of the PDF at such points would be a waist of time, once they were identified. A cheap solution to identify low density points is to keep a low resolution histogram for each spatial location of the image frame. Histogram updating can be done with only one increment and one decrement operation per new estimation frame. Low resolution is beneficial for both dealing with data sparseness and memory considerations. In

this work, we used a $16\times16\times16$ color histogram to fast discard low density points from accurate evaluation.

A concise pseudo-code description of the background tracking algorithm for a given spatial location is given in Figure 1.

```
if ( K_h(x_new − b)! = 0 )
    update ( b and p̂(b) );
    else if ( Hist(x_new) > threshold )
        if ( p̂(x_new) > p̂(b) )
                b =: x_new;
```

Fig. 1. Pseudo code description of the fast background tracking algorithm.

If the newly entered data point is within the domain of the kernel function centered at the currently estimated background, the density and the location of the background mode are updated in the next code line. Otherwise, the histogram based density estimate at the new data point is checked against a threshold. Only if the density threshold is passed, the new data point is submitted to accurate density evaluation by equation (7) and the result is compared to the current density maxima of the background for possible replacement.

When the new data point falls within the domain of the kernel function, the background density is updated from equation (18), while the background color is updated using the following rule:

$$\boldsymbol{b}_{new} = (1-\alpha)\boldsymbol{b}_{old} + \alpha\boldsymbol{x}_{new} = \boldsymbol{b}_{old} + \alpha(\boldsymbol{x}_{new} - \boldsymbol{b}_{old}). \tag{19}$$

Such a rule has been successfully used for mean value (mode) updating for the Gaussian mixture model in parametric background estimation. However, in the framework of nonparametric density estimation techniques, our theoretical motivation behind this option is related to the mean shift paradigm [19, 20]. The mean shift is a gradient ascent algorithm used to detect local maxima of a nonparametrically estimated PDF. It can be shown that for the Epanechnikov kernel function, the estimated gradient of the estimated PDF at a point **x** points in the direction of the mean shift, that is the difference between the arithmetic mean of the points lying within the domain of the kernel function centered at **x** and **x** itself. This can be shown in a straightforward manner by differencing equation (7), where the Epanechnikov kernel:

$$K_{Eh}(\boldsymbol{x}) = \begin{cases} \text{ct.}\left(1 - \frac{\boldsymbol{x}^T\boldsymbol{x}}{h^2}\right) & \text{if} \quad \boldsymbol{x}^T\boldsymbol{x} < 1 \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

is used. At each step, the algorithm shifts from the current point to the mean of the points lying within the domain of the kernel function. As the move is in the direction of the increase of the estimated PDF and the function is bounded, convergence is granted. A proof for the discrete case is given in [19]. Note that the mean shift mode estimate is not confined to any discretization and potentially overperforms evaluations restricted at the N target data points as in the work of Elgamal [16, 17]. This supposition is also supported by our experimental findings, although the improvements are barely noticeable.

Suppose we have an estimate of the background, $\boldsymbol{b}_{old}$ and a new pixel, $\boldsymbol{x}_{new}$ enters the data buffer and it is within the active range of the kernel function. If $\boldsymbol{b}_{old}$ was the old mode, it also was the old mean. Therefore, the new mean, with the contribution of the new pixel included, can be computed from equation (18) if we put

$$\alpha = \frac{1}{1 + N\hat{p}(\boldsymbol{b}_{old})}. \tag{21}$$

In practice, we used a constant and presumably higher value, $\alpha = 1/8$, with very stable results. Only one mean shift iteration is done for a new data point. Usually the mean shift algorithm converges in no more than two or three steps. Since we start from a point supposed to be fairly close to the convergence point, no further iterations are done. If integer arithmetic is used for background representation, the learning rate should be related to the scale of the kernel by the constraint $\alpha h_c > 1/4$. Otherwise, the background newer gets changed by equation (19), due to quantization the effects of integer representation.

Normally, most of the pixels observed at a location belong to the background, except for periods with very heavy traffic. New pixels failing to meet the TRUE condition of the if() statement in the first pseudo code line are checked on the third line whether or not they represent possible candidates to change the current background estimate. Most of these pixels belong to the moving foreground and have very low PDF, therefore they are quickly discarded from further evaluation by the histogram density test. The quantization cells of the histogram have to be large in comparison with the kernel scale, h_c, in order to avoid false negative decisions. When significant background changes accumulate during a number of frames, for example as a result of a new static object being included in the background or an object being carried out, the number of counts eventually raise above a threshold level in one or more of the histogram bins and will be considered for more precise nonparametric probability density evaluation, through equation (7). If the density at the new pixel exceeds the density of the currently estimated background, the background estimate is replaced by the new pixel. The histogram threshold level is set as a fraction β of the number of kernel hits at the estimated background, $N\hat{p}(\boldsymbol{b})$.

In our experiments, $\beta = 0.5$ was most frequently used but the exact value proved to be noncritical.

Scale parameter updating, based on equations (10) and (11), is done each time a new image frame is acquired from the video stream. The background information is also updated every frame, but for only one of pixel from a 4×4 block at a time. This way, 16 frames are needed for a complete background update and the computational load of the background estimation process is correspondingly diminished. Processing in turn different pixels from 4×4 blocks instead of processing all pixels simultaneously in every 16-th frame is twofold beneficial. First, errors resulting from biased background values induced by moving foreground objects are spread in space and therefore can be easier removed by filtering the segmented images. Second, the processing load is more evenly distributed in time.

4 Results and Discussion

To assess the performance of the proposed background estimation and tracking technique, we run comparative tests with the traditional background estimation based on the evaluation of the PDF at each of the N data points stored for each pixel. We performed both qualitative and quantitative tests.

The left side image from Figure 2 was obtained by a direct implementation of the kernel density estimation technique using the separable rectangular window from equation (8) with kernel bandwidth scaling factor in equation (10) $\alpha = 1$, while the right side image was obtained with the recursive mean shift tracking method proposed and the same kernel. Despite of being used from 128 frames with very heavy traffic, severe shadow and lighting source reflection, both images have a good quality and can be successfully used for the purpose of background subtraction. The images are virtually identical, with a very slight and favorable edge preserving smoothing effect in the case of the proposed approach. The higher granularity of the background obtained by kernel density estimation is a result the discrete nature of the method, the density being evaluated only at data points in the RGB feature space. By contrast, the mean shift based mode tracking theoretically corresponds to a continuous estimate of the density and benefits from the edge preserving smoothing effect of the conditional averaging resulting from equation (19).

Results for other two frames from the same indoor people counting sequence are illustrated in Figure 3. In Figure 3a), the scaling factor α from equation (10) for the kernel bandwidth was 1, while in Figure 3b) it was 3. The extracted background is practically unaffected by the scale factor change, demonstrating the robustness of the nonparametric estimation approach to the selection of the estimator's scale. The

Fig. 2. Left: background obtained by direct implementation of the kernel density estimation method. Right: background from the same image sequence obtained by the mean shift based mode tracking.

estimated background is on the bottom-right position. In the bottom-left position, the results of foreground segmentation are illustrated. The top-left images show the current frame, while the top-right images contain the same image with the enclosing rectangle of the valid countable moving objects included.

In order to obtain quantitative assessment of the proposed background subtraction method, we compared the results of the mean shift tracking estimator with the results of the kernel density estimator on synthetic data, with available background truth. Namely, we generated a constant background corrupted with zero mean white noise uniformly distributed between -0.5 and 0.5. We evaluated standard deviation of the estimation error obtained with the standard kernel density estimation and the mean shift tracking estimator for 200 samples from a sequence of random samples, using different kernel bandwidths.

A sliding data window of 40 samples was used for the kernel density estimator. The results of the performed simulations are shown in Figure 4. Note that the mean shift tracking estimator was started from the true zero background level. While this may be considered an idealized start, its much lower variance undoubtedly confirm the quality of smooth estimator of the mean shift mode tracker noticeable from Figure 2 too.

To obtain a better insight on the properties of the estimators under test, the instant estimation error sequence recorded for kernel scale 0.6, which is the best for the kernel density estimator and the worst for the mean shift tracking estimator, is illustrated in Figure 5.

The complexity of the method was reduced from O(N^2) to O($2N$) in the worst case, the same as for the more elaborated Elgamal solution, based on Fast Gauss Transform [21]. However, we argue and our experiments confirm that such worst

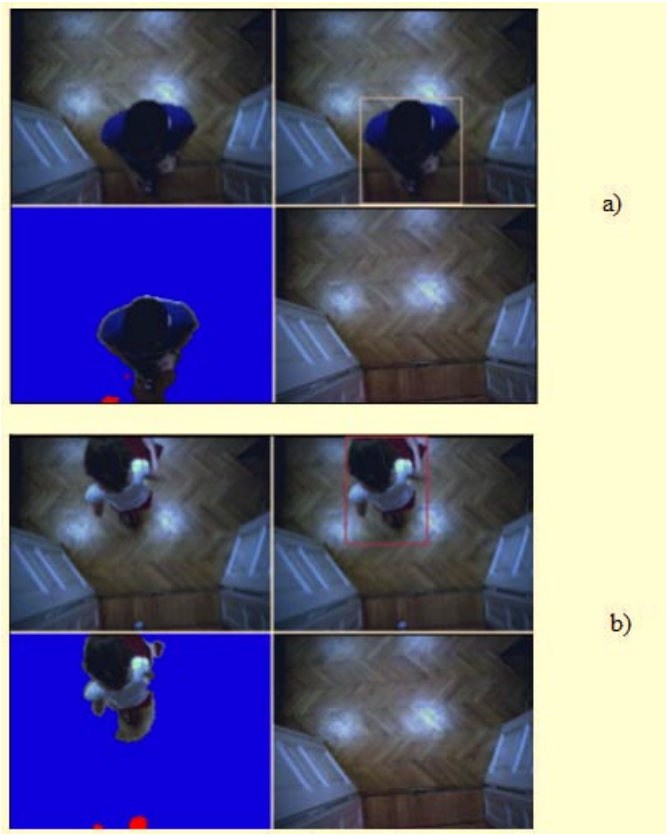

Fig. 3. Groups of pictures a) and b) show two frames from the same sequence obtained wit kernel scale factors 1 and 3. The top-left images show the current frame. Bottom-left images show the results of foreground segmentation. Bottom-right images show the estimated backgrounds. Top-right images contain the current frame images with the enclosing rectangle marking the valid, countable, moving objects.

cases consist of the relatively infrequent situation of radical background change, for example as a result of adding or removing static objects from the scene background. In order to evaluate the computational complexity of the proposed background estimation method, we set up a radical background changing experiment. To this end, a black square of 32×32 pixels was inserted in a central position of the images from the incoming video stream, for the first $N/2+16=80$ frames. This signal was initially detected as the true background at the corresponding pixels. After being removed, the black rectangle was gradually replaced with the real background, having much different values. To do so, the algorithm had to go through the long estimation loop with complexity O($2N$) operations per pixel. We recorded the number of times this long estimation was used each frame and illustrated it in Figure

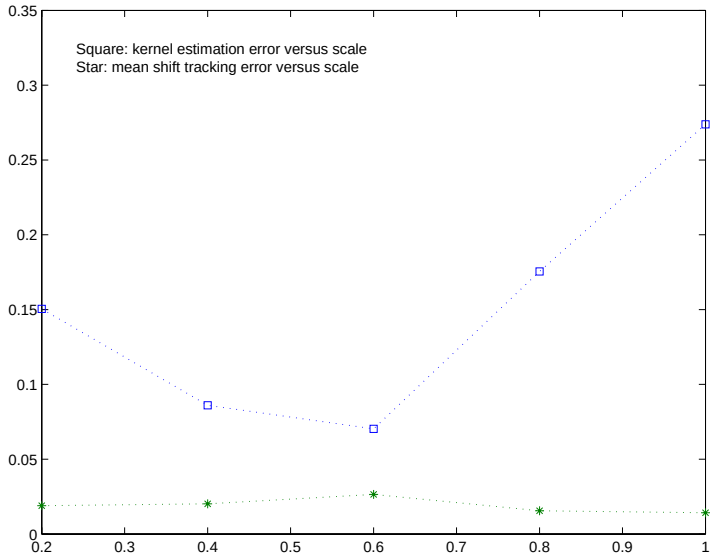

Fig. 4. Results of kernel and mean shift tracking estimators for different kernel scales.

6, as a percentage of the changed background pixels from the black square. Notice in Figure 6 that, except for the transient time of drastic background change, the long loop was unused for the histogram threshold $\beta N \widehat{p}(\mathbf{b})$ As a result, we can conclude that the complexity of the proposed background estimation method is not dependent on N, except for transient periods, when static objects are removed or introduced in the background and the complexity it is still only O($2N$).

For the rest of the time, very few operations per pixel are needed: an increment and a decrement for histogram updating, two operations for background tracking in equation (18), two subtractions, comparisons and increment/decrement operations for PDF update with equation (21), a subtraction, comparison and addition for median updating in equation (11) and a multiplication for scale estimation in equation (10). All these operations are done for each color channel. A few data shuffling operations, not seen in this analysis, have to be added for a more realistic evaluation. The computation time for the background tracking step obviously depends linearly on the number of pixels in the frame. In our experiments, with a 700 MHz Pentium III processor based PC, the computing time was of $1.5\mu s$ per pixel. For a 352×240 image resolution and 1/16 of the background image pixels updated each frame, a computing time of nearly 8ms per frame results, which allows comfortable real-time implementation and leaves a lot of processing time for object tracking and scene understanding purposes. On the same computer, we run a version of the Improved Fast Gauss Transform (IFGT) implemented and kindly provided by Yang et

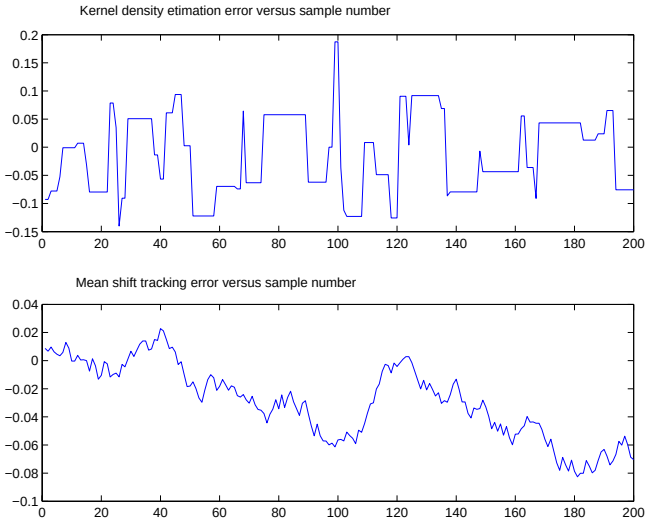

Fig. 5. Instant estimation errors versus sample number for the kernel density and mean shift mode tracking estimators.

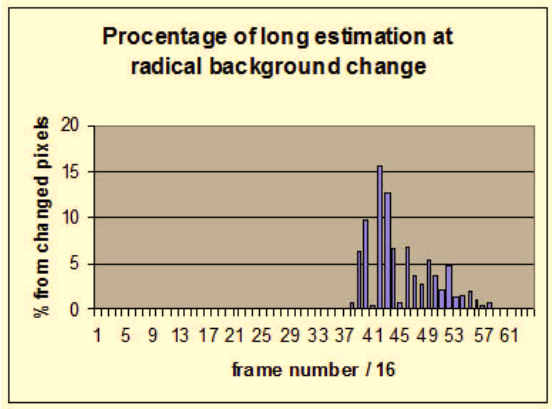

Fig. 6. Percentage of the long, O(2*N*), computations for background estimation, in an experiment with a radical background change, produced by removing a static object. Each bin corresponds to a group of 16 frames, needed for a complete background update in the current work.

al. [21]. The tests confirmed the theoretical O(*N*+*M*) complexity, with *N* source points and M target points and lead on our computer to an average computing time of 0.387 ms per data sample. This comparison is only made to evaluate the time saved by using the proposed method instead of the IFGT. We have to underline that IFGT is a much wider use general purpose density estimation algorithm using

Gaussian kernel. Its efficiency mainly applies for low dimensional data and big number (tenths of thousands) of source and target data points, while the mean shift mode tracker was designed to exploit in the best possible way the particularities of the background estimation problem in surveillance applications with multidimensional feature spaces and moderate size (several hundreds of samples) data buffer.

5 Conclusions

In this work, we introduced a fast algorithm for background estimation, based on recursive data processing and a rough histogram based density test used to avoid useless computations. The algorithm recursively computes a nonparametric kernel based probability density estimate by means of mean shift mode tracking. Qualitative and quantitative tests performed asses the accuracy of the proposed approach, while the computation time is not dependent on the length of the data buffer used for background estimation. That is, the computational complexity is $O(N^0)$. This compares favorably with the fast nonparametric background estimation techniques benefiting of the existence of the Improved Fast Gauss Transform algorithm and, we believe, even with parametric estimation, where up to five Gaussian distributions from a mixture need to be estimated at each step.

References

[1] C. R. Wren, A. Azarbayejani, T. Darel, and A. Pentland, "Pfinder: Real-time tracking of human body," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 7, pp. 780–785, July 1997.

[2] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: principles and practice of background maintenance," in *IEEE Conference on Computer Vision, Kerkyra, Greece*, 1999, pp. 255–261.

[3] D. Harwood, I. Haritaogly, and L. S. Davis, "W^4: Real-time surveillance of people and their activities," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 809–830, Aug. 2000.

[4] M. Harville, G. Gordon, and J. Woodfill, "Adaptive video background modeling using color and depth," in *International Conference on Image Processing ICIP 2001, Tessaloniki, Greece*, Oct. 2001.

[5] G. Gordon, T. Darrell, M. Harville, and J. Woodfill, "Background estimation and removal based on range and color," in *Proc. CVPR'98*, 1999, pp. 459–464.

[6] P. J. Rouseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*. Wiley, 1987.

[7] P. Meer, "Robust techniques for computer vision," in *Emerging Topics in Computer Vision*, G. Medioni and S. B. Kang, Eds. Prentice Hall, 2004, pp. 107–190.

[8] W. E. L. Grimson, C. Stauffer, R. Romano, and L. Lee, "Using adaptive tracking to classify and monitor activities in a site," in *IIII Conference on Computer Vision and Pattern Recognition, Santa Barbara, CA*, 1998, pp. 22–29.

[9] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models foe real-time tracking," in *IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, CO*, vol. 2, 1999, pp. 246–252.

[10] P. Kaew, T. K. Pong, and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems*, Sept. 2001.

[11] P. Withagen, K. Schutte, and F. Groen, "Object detection and tracking using a likelihood based approach," in *Proc. ASCI 2002 Conference, Lochem, The Netherlands*, June 2002, pp. 248–253.

[12] M. Pic, L. Berthouze, and T. Kurita, "Adaptive background estimation: Computing a pixel-wise learning rate from local confidence and global correlation values," in *IEICE Trans. Inf & Syst.*, vol. E87-D, No.1, Jan. 2004.

[13] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39(1), Series B, pp. 1–38, 1977.

[14] M. P. Wand and M. C. Jones, *Kernel Smoothing*. Chapman & Hall, 1995.

[15] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley, 2000.

[16] A. Elgamal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1151–1162, July 2002.

[17] A. Elgamal, R. Duraiswami, and L. S. Davis, "Efficient kernel density estimation using the Fast Gauss Transform with applications to color modeling and tracking," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, no. 11, pp. 1499–1504, Nov. 2003.

[18] R. Cucciara, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts and shadows in video streams," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, no. 10, pp. 1337–1342, Oct. 2003.

[19] D. Comaniciu and P. Meer, "Mean shift analysis and applications," in *International Conference on Computer Vision, Kerkyra, Greece*, 1999, pp. 1197–1203.

[20] ——, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Machine Intell*, vol. 24, pp. 603–619, 2002.

[21] J. Yang, R. Duraiswami, N. Gumerov, and L. Davis, "Improved Fast Gauss Transform and Efficient Kernel Density Estimation," in *IEEE Intl. Conference on Computer Vision ICCV 2003*, 2003, pp. 464–471.