LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-CONF-200706

ROBUST TECHNIQUES FOR BACKGROUND SUBTRACTION IN URBAN TRAFFIC VIDEO

S-C. S. Cheung, C. Kamath

November 3, 2003

IS&T/SPIE's Symposium on Electronic Imaging
San Jose, CA, United States
January 18, 2004 through January 22, 2004

Robust techniques for background subtraction in urban traffic video

Sen-Ching S. Cheung and Chandrika Kamath

{cheung11, kamath2}@llnl.gov
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
7000 East Avenue, Livermore, CA 94550

ABSTRACT

Identifying moving objects from a video sequence is a fundamental and critical task in many computer-vision applications. A common approach is to perform background subtraction, which identifies moving objects from the portion of a video frame that differs significantly from a background model. There are many challenges in developing a good background subtraction algorithm. First, it must be robust against changes in illumination. Second, it should avoid detecting non-stationary background objects such as swinging leaves, rain, snow, and shadow cast by moving objects. Finally, its internal background model should react quickly to changes in background such as starting and stopping of vehicles. In this paper, we compare various background subtraction algorithms for detecting moving vehicles and pedestrians in urban traffic video sequences. We consider approaches varying from simple techniques such as frame differencing and adaptive median filtering, to more sophisticated probabilistic modeling techniques. While complicated techniques often produce superior performance, our experiments show that simple techniques such as adaptive median filtering can produce good results with much lower computational complexity.

Keywords: Background subtraction, urban traffic video

1. INTRODUCTION

Identifying moving objects from a video sequence is a fundamental and critical task in video surveillance, traffic monitoring and analysis, human detection and tracking, and gesture recognition in human-machine interface. A common approach to identifying the moving objects is background subtraction, where each video frame is compared against a reference or background model. Pixels in the current frame that deviate significantly from the background are considered to be moving objects. These "foreground" pixels are further processed for object localization and tracking. Since background subtraction is often the first step in many computer vision applications, it is important that the extracted foreground pixels accurately correspond to the moving objects of interest. Even though many background subtraction algorithms have been proposed in the literature, the problem of identifying moving objects in complex environment is still far from being completely solved.

There are several problems that a good background subtraction algorithm must solve correctly. Consider a video sequence from a stationary camera overlooking a traffic intersection. As it is an outdoor environment, a background subtraction algorithm should adapt to various levels of illumination at different times of the day and handle adverse weather condition such as fog or snow that modifies the background. Changing shadow, cast by moving objects, should be removed so that consistent features can be extracted from the objects in subsequent processing. The complex traffic flow at the intersection also poses challenges to a background subtraction algorithm. The vehicles move at a normal speed when the light is green, but come to a stop when it turns red. The vehicles then remain stationary until the light turns green again. A good background subtraction algorithm must handle the moving objects that first merge into the background and then become foreground at a later time. In addition, to accommodate the real-time needs of many applications, a background subtraction algorithm must be computationally inexpensive and have low memory requirements, while still being able to accurately identify moving objects in the video.

Even though many background subtraction techniques have been proposed, they are typically presented as parts of a larger computer vision application. In this paper, we focus on the problem of background subtraction

and survey many existing schemes in the literature. In order to have a fair comparison between different schemes, we analyze them based on how they differ in four different functional steps: preprocessing, background modeling, foreground detection, and data validation. We have also implemented a number of representative techniques, and we evaluate their performance on urban traffic video sequences under different conditions. The paper is organized as follows: the survey of background subtraction algorithms can be found in Section 2. The experimental results are presented in Section 3. Finally, we conclude our paper and discuss future work in Section 4.

2. BACKGROUND SUBTRACTION ALGORITHMS

Even though there exist a myriad of background subtraction algorithms in the literature, most of them follow a simple flow diagram shown in Figure 1. The four major steps in a background subtraction algorithm are preprocessing, background modeling, foreground detection, and data validation. Preprocessing consists of a collection of simple image processing tasks that change the raw input video into a format that can be processed by subsequent steps. Background modeling uses the new video frame to calculate and update a background model. This background model provides a statistical description of the entire background scene. Foreground detection then identifies pixels in the video frame that cannot be adequately explained by the background model, and outputs them as a binary candidate foreground mask. Finally, data validation examines the candidate mask, eliminates those pixels that do not correspond to actual moving objects, and outputs the final foreground mask. Domain knowledge and computationally-intensive vision algorithms are often used in data validation. Real-time processing is still feasible as these sophisticated algorithms are applied only on the small number of candidate foreground pixels. Many different approaches have been proposed for each of the four processing steps. We review some of the representative ones in the following subsections.

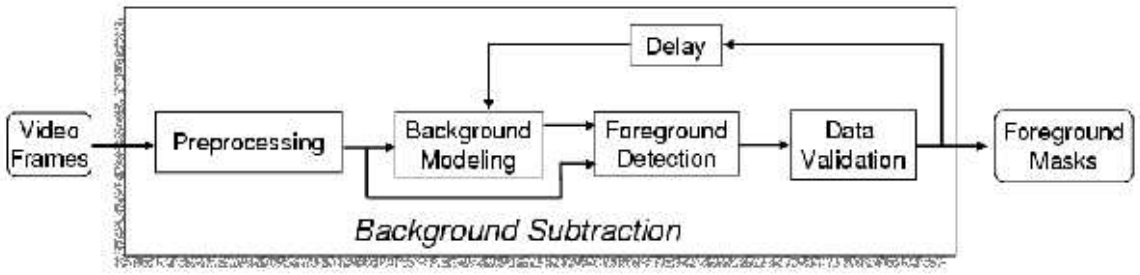


Figure 1. *Flow diagram of a generic background subtraction algorithm.*

2.1. Preprocessing

In most computer vision systems, simple temporal and/or spatial smoothing are used in the early stage of processing to reduce camera noise. Smoothing can also be used to remove transient environmental noise such as rain and snow captured in outdoor camera. For real-time systems, frame-size and frame-rate reduction are commonly used to reduce the data processing rate. If the camera is moving or multiple cameras are used at different locations, image registration between successive frames or among different cameras is needed before background modeling [1, 2].

Another key issue in preprocessing is the data format used by the particular background subtraction algorithm. Most of the algorithms handle luminance intensity, which is one scalar value per each pixel. However, color image, in either RGB or HSV color space, is becoming more popular in the background subtraction literature [3–10]. These papers argue that color is better than luminance at identifying objects in low-contrast areas and suppressing shadow cast by moving objects. In addition to color, pixel-based image features such as spatial and temporal derivatives are sometimes used to incorporate edges and motion information. For example, intensity values and spatial derivatives can be combined to form a single state space for background tracking with the Kalman filter [11]. Pless et al. combine both spatial and temporal derivatives to form a constant velocity background model for detecting speeding vehicles [12]. The main drawback of adding color or derived features in background modeling is the extra complexity for model parameter estimation. The increase in complexity is often significant as most background modeling techniques maintain an independent model for each pixel.

2.2. Background Modeling

Background modeling is at the heart of any background subtraction algorithm. Much research has been devoted to developing a background model that is robust against environmental changes in the background, but sensitive enough to identify all moving objects of interest. We classify background modeling techniques into two broad categories – non-recursive and recursive. They are described in the following subsections. We focus only on highly-adaptive techniques, and exclude those that require significant resource for initialization. These include schemes described in [13] and [14], which store tens of seconds of video to construct initial background models that are characterized by eigen-images [13] or temporal maximum, minimum, and maximum inter-frame differences of all identified background pixels [14]. For the remainder of this paper, $I_t(x, y)$ and $B_t(x, y)$ are used to denote the luminance pixel intensity and its background estimate at spatial location (x, y) and time t. The spatial coordinate (x, y) may be dropped if it is not relevant in the description.

2.2.1. Non-recursive Techniques

A non-recursive technique uses a sliding-window approach for background estimation. It stores a buffer of the previous L video frames, and estimates the background image based on the temporal variation of each pixel within the buffer. Non-recursive techniques are highly adaptive as they do not depend on the history beyond those frames stored in the buffer. On the other hand, the storage requirement can be significant if a large buffer is needed to cope with slow-moving traffic. Given a fixed-size buffer, this problem can be partially alleviated by storing the video frames at a lower frame-rate r. Some of the commonly-used non-recursive techniques are described below:

Frame differencing Arguably the simplest background modeling technique, frame differencing uses the video frame at time $t - 1$ as the background model for the frame at time t. Since it uses only a single previous frame, frame differencing may not be able to identify the interior pixels of a large, uniformly-colored moving object. This is commonly known as the aperture problem.

Median filter Median filtering is one of the most commonly-used background modeling techniques [4,10,15–17]. The background estimate is defined to be the median at each pixel location of all the frames in the buffer. The assumption is that the pixel stays in the background for more than half of the frames in the buffer. Median filtering has been extended to color by replacing the median with the medoid [10]. The complexity of computing the median is $O(L \log L)$ for each pixel.

Linear predictive filter Toyama et al. compute the current background estimate by applying a linear predictive filter on the pixels in the buffer [18]. The filter coefficients are estimated at each frame time based on the sample covariances, making this technique difficult to apply in real-time.

Non-parametric model Unlike previous techniques that use a single background estimate at each pixel location, Elgammal et al. [5] use the entire history $I_{t-L}, I_{t-L+1}, \ldots, I_{t-1}$ to form a non-parametric estimate of the pixel density function $f(I_t = u)$:

$$f(I_t = u) = \frac{1}{L} \sum_{i=t-L}^{t-1} K(u - I_i) \tag{1}$$

$K(\cdot)$ is the kernel estimator which was chosen to be Gaussian. The current pixel I_t is declared as foreground if it is unlikely to come from this distribution, i.e. $f(I_t)$ is smaller than some predefined threshold. The advantage of using the full density function over a single estimate is the ability to handle multi-modal background distribution. Examples of multi-modal background include pixels from a swinging tree or near high-contrast edges where they flicker under small camera movement. The implementation in [5] uses the median of the absolute differences between successive frames as the width of the kernel. Thus, the complexity of building the model is the same as median filtering. On the other hand, the foreground detection is more complex as it needs to compute Equation (1) for each pixel.

2.2.2. Recursive Techniques

Recursive techniques do not maintain a buffer for background estimation. Instead, they recursively update a single background model based on each input frame. As a result, input frames from distant past could have an effect on the current background model. Compared with non-recursive techniques, recursive techniques require less storage, but any error in the background model can linger for a much longer period of time. Most schemes include exponential weighting to discount the past, and incorporate positive decision feedback to use only background pixels for updating. Some of the representative recursive techniques are described below:

Approximated median filter Due to the success of non-recursive median filtering, McFarlane and Schofield propose a simple recursive filter to estimate the median [19]. This technique has also been used in background modeling for urban traffic monitoring [20]. In this scheme, the running estimate of the median is incremented by one if the input pixel is larger than the estimate, and decreased by one if smaller. This estimate eventually converges to a value for which half of the input pixels are larger than and half are smaller than this value, that is, the median.

Kalman filter Kalman filter is a widely-used recursive technique for tracking linear dynamical systems under Gaussian noise. Many different versions have been proposed for background modeling, differing mainly in the state spaces used for tracking. The simplest version uses only the luminance intensity [3, 21–23]. Karmann and von Brandt use both the intensity and its temporal derivative [24], while Koller, Weber, and Malik use the intensity and its spatial derivatives [11]. We provide a brief description of the popular scheme used in [24]. The internal state of the system is described by the background intensity B_t and its temporal derivative B'_t, which are recursively updated as follows:

$$\begin{bmatrix} B_t \\ B'_t \end{bmatrix} = \mathbf{A} \cdot \begin{bmatrix} B_{t-1} \\ B'_{t-1} \end{bmatrix} + \mathbf{K_t} \cdot \left(I_t - \mathbf{H} \cdot \mathbf{A} \cdot \begin{bmatrix} B_{t-1} \\ B'_{t-1} \end{bmatrix} \right) \tag{2}$$

Matrix $\mathbf{A}$ describes the background dynamics and $\mathbf{H}$ is the measurement matrix. Their particular values used in [24] are as follows:

$$\mathbf{A} = \begin{bmatrix} 1 & 0.7 \\ 0 & 0.7 \end{bmatrix}, \quad \mathbf{H} = [1 \;\; 0] \tag{3}$$

The Kalman gain matrix $\mathbf{K_t}$ switches between a slow adaptation rate α_1 and a fast adaptation rate $\alpha_2 > \alpha_1$ based on whether I_{t-1} is a foreground pixel:

$$\mathbf{K_t} = \begin{bmatrix} \alpha_1 \\ \alpha_1 \end{bmatrix} \text{ if } I_{t-1} \text{ is foreground, and } \begin{bmatrix} \alpha_2 \\ \alpha_2 \end{bmatrix} \text{ otherwise.} \tag{4}$$

Mixture of Gaussians (MoG) Unlike Kalman filter which tracks the evolution of a single Gaussian, the MoG method tracks multiple Gaussian distributions simultaneously. MoG has enjoyed tremendous popularity since it was first proposed for background modeling in [25]. Similar to the non-parametric model described in Section 2.2.1, MoG maintains a density function for each pixel. Thus, it is capable of handling multi-modal background distributions. On the other hand, since MoG is parametric, the model parameters can be adaptively updated without keeping a large buffer of video frames. Our description of MoG is based on the scheme described in [6]. The pixel distribution $f(I_t = u)$ is modeled as a mixture of K Gaussians:

$$f(I_t = u) = \sum_{i=1}^{K} \omega_{i,t} \cdot \eta(u; \mu_{i,t}, \sigma_{i,t}) \tag{5}$$

where $\eta(u; \mu_{i,t}, \sigma_{i,t})$ is the i-th Gaussian component with intensity mean $\mu_{i,t}$ and standard deviation $\sigma_{i,t}$. $\omega_{i,t}$ is the portion of the data accounted for by the i-th component. Typically, K ranges from three to five, depending on the available storage. For each input pixel I_t, the first step is to identify the component $\hat{i}$ whose mean is closest to I_t. Component $\hat{i}$ is declared as the *matched component* if $|I_t - \mu_{\hat{i},t-1}| \leq D \cdot \sigma_{\hat{i},t-1}$,

where D defines a small positive deviation threshold. The parameters of the matched component are then updated as follows:

$$\begin{aligned} \omega_{\hat{i},t} &= (1-\alpha)\omega_{\hat{i},t-1} + \alpha \\ \mu_{\hat{i},t} &= (1-\rho)\mu_{\hat{i},t-1} + \rho I_t \\ \sigma^2_{\hat{i},t} &= (1-\rho)\sigma^2_{\hat{i},t-1} + \rho(I_t - \mu_{\hat{i},t})^2, \end{aligned} \tag{6}$$

where α is a user-defined learning rate with $0 \leq \alpha \leq 1$. ρ is the learning rate for the parameters and can be approximated as follows*:

$$\rho \approx \frac{\alpha}{\omega_{\hat{i},t}} \tag{7}$$

If no matched component can be found, the component with the least weight is replaced by a new component with mean I_t, a large initial variance σ_o and a small weight ω_o. The rest of the components maintain the same means and variances, but lower their weights to achieve exponential decay:

$$\omega_{i,t} = (1-\alpha)\omega_{i,t-1} \tag{8}$$

Finally, all the weights are renormalized to sum up to one. To determine whether I_t is a foreground pixel, we first rank all components by their values of $\omega_{i,t}/\sigma_{i,t}$. Higher-rank components thus have low variances and high probabilities, which are typical characteristics of background. If $i_1, i_2, \ldots, i_K$ is the component order after sorting, the first M components that satisfy the following criterion are declared to be the *background components*:

$$\sum_{k=i_1}^{i_M} \omega_{k,t} \geq \Gamma, \tag{9}$$

where Γ is the weight threshold. I_t is declared as a foreground pixel if I_t is within D times the standard deviation from the mean of any one of the background components. Note that the above formulation can be easily extended to handle color data. The computational complexity and storage requirement of MoG is linear in terms of the number of components K.

Recent development in MoG technologies include a sensitivity analysis of parameters [27], improvements in complexity and adaptation [7, 26, 28], and an extension to construct a panoramic background [1].

2.3. Foreground Detection

Foreground detection compares the input video frame with the background model, and identifies candidate foreground pixels from the input frame. Except for the non-parametric model and the MoG model, all the techniques introduced in Section 2.2 use a single image as their background models. The most commonly-used approach for foreground detection is to check whether the input pixel is significantly different from the corresponding background estimate:

$$|I_t(x,y) - B_t(x,y)| > T \tag{10}$$

Another popular foreground detection scheme is to threshold based on the normalized statistics:

$$\frac{|I_t(x,y) - B_t(x,y) - \mu_d|}{\sigma_d} > T_s, \tag{11}$$

where μ_d and σ_d are the mean and the standard deviation of $I_t(x,y) - B_t(x,y)$ for all spatial locations (x,y). Most schemes determine the foreground threshold T or T_s experimentally.

Ideally, the threshold should be a function of the spatial location (x,y). For example, the threshold should be smaller for regions with low contrast. One possible modification is proposed by Fuentes and Velastin [29].

*Stauffer and Grimson use $\alpha \cdot \eta(I_t; \mu_{\hat{i},t}, \sigma_{\hat{i},t})$ to compute ρ in [6], which is incorrect as pointed out in [26]. ρ should be the product between $\alpha/\omega_{\hat{i},t}$ and the posterior probability of I_t belonging to the matched component $\hat{i}$. Here, we assume the posterior probability to be one as I_t is much closer to the matched component than any other component.

They use the relative difference rather than absolute difference to emphasize the contrast in dark areas such as shadow:

$$\frac{|I_t(x,y) - B_t(x,y)|}{B_t(x,y)} > T_c \tag{12}$$

Nevertheless, this technique cannot be used to enhance contrast in bright images such as an outdoor scene under heavy fog.

Another approach to introduce spatial variability is to use two thresholds with hysteresis [10, 23]. The basic idea is to first identify "strong" foreground pixels whose absolute differences with the background estimates exceeded a large threshold. Then, foreground regions are grown from strong foreground pixels by including neighboring pixels with absolute differences larger than a smaller threshold. The region growing can be performed by using a two-pass, connected-component grouping algorithm [30].

2.4. Data Validation

We define data validation as the process of improving the candidate foreground mask based on information obtained from outside the background model. All the background models in Section 2.2 have three main limitations: first, they ignore any correlation between neighboring pixels; second, the rate of adaption may not match the moving speed of the foreground objects; and third, non-stationary pixels from moving leaves or shadow cast by moving objects are easily mistaken as true foreground objects.

The first problem typically results in small false-positive or false-negative regions distributed randomly across the candidate mask. The most common approach is to combine morphological filtering and connected component grouping to eliminate these regions [6, 18, 21]. Applying morphological filtering on foreground masks eliminates isolated foreground pixels and merges nearby disconnected foreground regions. Many applications assume that all moving objects of interest must be larger than a certain size. Connected-component grouping can then be used to identify all connected foreground regions, and eliminates those that are too small to correspond to real moving objects.

When the background model adapts at a slower rate than the foreground scene, large areas of false foreground, commonly known as "ghosts", often occur [6, 14]. If the background model adapts too fast, it will fail to identify the portion of a foreground object that has corrupted the background model. A simple approach to alleviate these problems is to use multiple background models running at different adaptation rates, and periodically cross-validate between different models to improve performance [5, 8, 23]. Sophisticated vision techniques can also be used to validate foreground detection. Computing optical flow for candidate foreground regions can eliminate ghost objects as they have no motion [10, 31]. Color segmentation can be used to grow foreground regions by assuming similar color composition throughout the entire object [18]. If multiple cameras are available to capture the same scene at different angles, disparity information between cameras can be used to estimate depth. Depth information is useful as foreground objects are closer to the camera than background [8, 32].

The moving-leaves problem can be addressed by using sophisticated background modeling techniques like MoG and applying morphological filtering for cleanup. On the other hand, suppressing moving shadow is much more problematic, especially for luminance-only video. A recent survey and comparison of many shadow suppression algorithms can be found in [33].

3. EXPERIMENTAL RESULTS

In this section, we compare the performance of a number of popular background modeling techniques. Table 1 lists all the techniques being tested, in the increasing order of complexity. We fix the buffer size for the median filter and the number of components for MoG so that they have comparable storage requirements and computational complexity. In the performance evaluation, we will vary the test parameters to show the performance of each algorithm at different operation points.

In this paper, we apply the background models to luminance sequences only. For preprocessing, we first apply a three-frame temporal erosion to the test sequence, that is we replace I_t with the minimum of I_{t-1}, I_t, and I_{t+1}. This step can reduce temporal camera noise and mitigate the effect of snowfall present in one of our test sequences. Then, a 3×3 spatial Gaussian filter is used to reduce spatial camera noise. Simple thresholding with

normalized statistics is used for foreground detection, except for MoG which has a separate foreground detection process as described in Section 2.2.2. No data validation is performed to postprocess the output foreground masks.

Schemes	**Fixed parameters**	**Test parameters**
Frame differencing (FD)	None	Foreground threshold T_s
Approximated median filter (AMF)	None	Foreground threshold T_s
Kalman filter (KF)	None	Adaptation rates α_1, α_2
		Foreground threshold T_s
Median filter (MF)	Buffer size $L = 9$	Buffer sampling rate r
		Foreground threshold T_s
Mixture of Gaussian (MoG)	Number of components $K = 3$	Adaptation rate α
	Initial variance $\sigma_o^2 = 36$	Weight threshold Γ
	Initial weight $\omega_o = 0.1$	Deviation threshold D

Table 1. *Background modeling schemes tested and their parameters.*

3.1. Test Sequences

We have selected four publicly-available urban traffic video sequences from the website maintained by KOGS/-IAKS Universitaet Karlsruhe[†] A sample frame from each sequence is shown in the first row of Figure 2. The first sequence is called "Bright", which is 1500 frames long showing a traffic intersection in bright daylight. This sequence contains some "stop-and-go" traffic – vehicles come to a stop in front of a red-light and start moving once the light turns green. The second sequence is called "Fog", which is 300 frames long showing the same traffic intersection in heavy fog. The third sequence "Snow" is also 300 frames long and shows the intersection while snowing. Fog and Snow were originally in color; we have first converted them into luminance and discarded the chroma channels. The first three sequences all have low to moderate traffic. They are selected to demonstrate the performance of background subtraction algorithms under different weather conditions. The last sequence "Busy" is 300 frames long. It shows a busy intersection with the majority of the vehicle traffic flowing from the top left corner to the right side. A quarter of the intersection is under a shadow of a building. A number of pedestrians are walking on the sidewalk on the left. The camera appears to be behind a window and the base of the window is partially reflected at the lower right corner of the video frames. This sequence is selected because of the large variation in the sizes of the moving objects and the presence of the shadow of a large building.

3.2. Evaluation

In order to have a quantitative evaluation of the performance, we have selected ten frames at regular intervals from each test sequence, and manually highlighted all the moving objects in them. These "ground-truth" frames are selected from the latter part of each of the test sequences[‡] to minimize the effect of the initial adaptation of the algorithms. In the manual annotation, we highlight only the pixels belonging to vehicles and pedestrians that are actually moving at that frame. Since we do not use any shadow suppression scheme in our comparison, we also include those shadow pixels cast by moving objects. The ground-truth frames showing only the moving objects are shown in the second row of Figure 2.

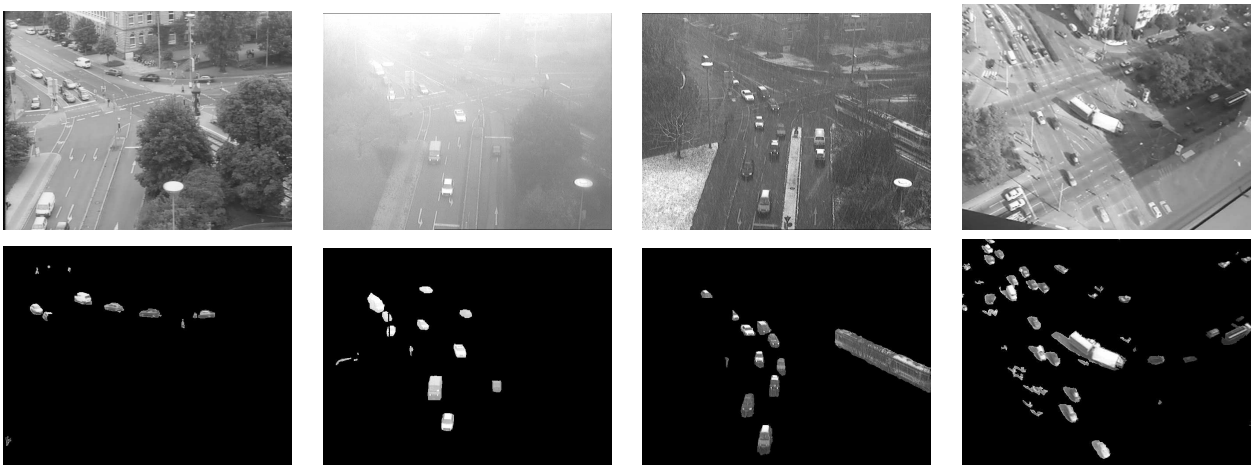
We use two information retrieval measurements, recall and precision, to quantify how well each algorithm matches the ground-truth [34]. They are defined in our context as follows:

$$\text{Recall} = \frac{\text{Number of foreground pixels correctly identified by the algorithm}}{\text{Number of foreground pixels in ground-truth}} \tag{13}$$

$$\text{Precision} = \frac{\text{Number of foreground pixels correctly identified by the algorithm}}{\text{Number of foreground pixels detected by the algorithm}} \tag{14}$$

[†]The URL is http://i21www.ira.uka.de/image_sequences. All sequences are copyrighted by H.-H. Nagel of KOGS/IAKS Universitaet Karlsruhe.

[‡]The ground-truth frames are selected from the last 1000 frames in the Bright sequence, and the last 200 frames in the remaining three sequences.



(a) Bright (b) Fog (c) Snow (d) Busy

Figure 2. *Sample frames and the corresponding ground-truth frames from the four test sequences: Bright, Fog, Snow, and Busy.*

Recall and precision values are both within the range of 0 and 1. When applied to the entire sequence, the recall and precision reported are averages over all the measured frames. Typically, there is a trade-off between recall and precision – recall usually increases with the number of foreground pixels detected, which in turn may lead to a decrease in precision. A good background algorithm should attain as high a recall value as possible without sacrificing precision.

In our experiments, we vary the parameters in each algorithm to obtain different recall-precision operating points. The resulting graphs for the four test sequences are shown in Figures 3(a) to (d). There are four plots for each sequence. The first plot corresponds to the two simplest algorithms, FD and AMF. The curves are generated by varying the foreground threshold T_s. The second plot corresponds to MF at buffer sampling rates of 1, 5, and 10 frames per second. The curves are also generated by varying T_s. The third plot corresponds to MoG at different combinations of α and Γ. The curves are generated by varying the deviation threshold D. Compared with the previous two schemes, there are far fewer actual data points on the MoG curves. The reason is that D directly affects the future states of MoG. To generate a single data point, one needs to run the algorithm through the entire video sequence at a particular value of D. On the other hand, T_s has no effect on the internal states of FD, AMF, or MF. To generate the results at different values of T_s, it is sufficient to run the algorithm once, save the raw difference frames, and then threshold them with different values of T_s. The final plot contains results from KF at different values of α_1 and α_2. The curves are also generated by varying T_s. Note that in the cases when α_1 and α_2 are equal, the feedback information is not used. The update equation in (2) reduces to a leaky moving average with exponential decay on past values.

Based on the measurements shown in Figure 3 and visual examination on the resulting foreground masks, we make the following observations regarding the background algorithms tested in this paper:

1. With the appropriate parameters, MoG achieves the best precision and recall in Figure 3. MF is a very close second, followed by AMF and KF. FD is significantly worse than all the other schemes.

2. Even though AMF is not as good as MoG and MF, it produces good performance with an extremely simple implementation. Since the amount of background update (+1 or -1) is independent of the foreground pixels, it is very robust against moving traffic. The only drawback is that it adapts slowly toward a large change in background – for example, as shown in the bottom image of Figure 4(c), AMF needs many frames to learn the new dark background revealed by a white car that moves away after being stationary for a long time.

3. Visually, KF produces the worst foreground masks among all the schemes. Even with a large foreground threshold and slow adapting rates, the background model in KF is easily affected by the foreground pixels. As a result, it typically leaves a long trail after a moving object.

4. All background algorithms tested are sensitive to environmental noise, as evident in the comparatively lower recall and precision numbers for the Fog and Snow sequences.

5. For our test sequences, algorithms that adapt more slowly often have better performance than those that adapt quickly. For example, MoG or KF running at a smaller value of α produces better results than the same algorithm running at a larger α. The same applies to MF with a larger value of r. This can be easily explained: slow adaptation prevents transient foreground from corrupting the background model. A visual example is shown in the top row of Figure 4. These images are the results of applying different algorithms to detecting a fast moving vehicle. There is a "tail" trailing behind the moving car in the leftmost image, which is produced by a fast-adapting MF. The tail is completely removed in the second image when a slow-adapting MF is used. Nonetheless, slow adaptation can be a double-edged sword – it also makes the algorithm less responsive to changes in background. The bottom row shows the same algorithms being applied to a scene in which a car begins to move after the light has turned green. The fast-adapting MF in the leftmost image quickly detects the change and updates its background model. In the second image, the slow-adapting MF still remembers the old location of the car, and produces a "ghost" object behind the car. The third column shows the results by AMF, which represents a compromise between the slow and fast-adapting MF. The best results come from MoG which are shown in the last column. This can be attributed to the multi-modal characteristics of MoG. Even though the old location of the car is still part of the background model in MoG, the model also captures the newly-revealed background as an alternative explanation of the input pixels.

6. The MoG method also has its own drawbacks. First, it is computationally intensive and its parameters require careful tuning. Second, it is very sensitive to sudden changes in global illumination. If a scene remains stationary for a long period of time, the variances of the background components may become very small. A sudden change in global illumination can then turn the entire frame into foreground.

4. CONCLUSIONS

In this paper, we survey a number of background subtraction algorithms in the literature. We analyze them based on how they differ in preprocessing, background modeling, foreground detection, and data validation. Five specific algorithms are tested on urban traffic video sequences: frame differencing, adaptive median filtering, median filtering, mixture of Gaussians, and Kalman filtering. Mixture of Gaussians produces the best results, while adaptive median filtering offers a simple alternative with competitive performance. More research, however, is needed to improve robustness against environment noise, sudden change of illumination, and to provide a balance between fast adaptation and robust modeling.

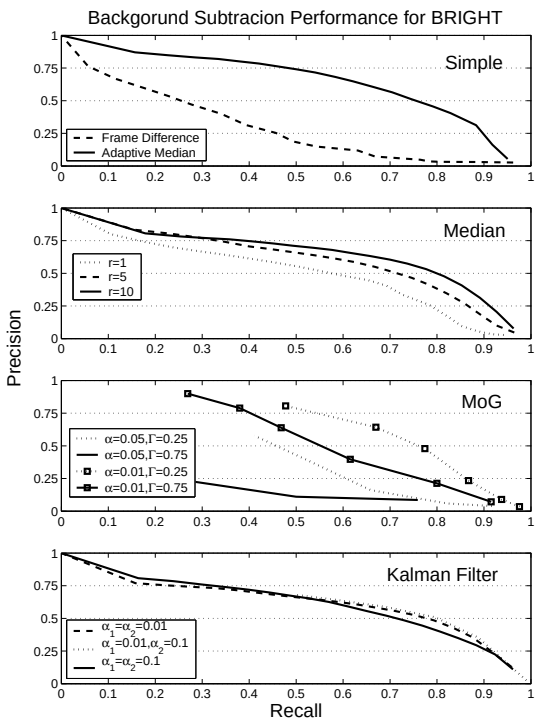
ACKNOWLEDGMENTS

UCRL-CONF-200706: This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

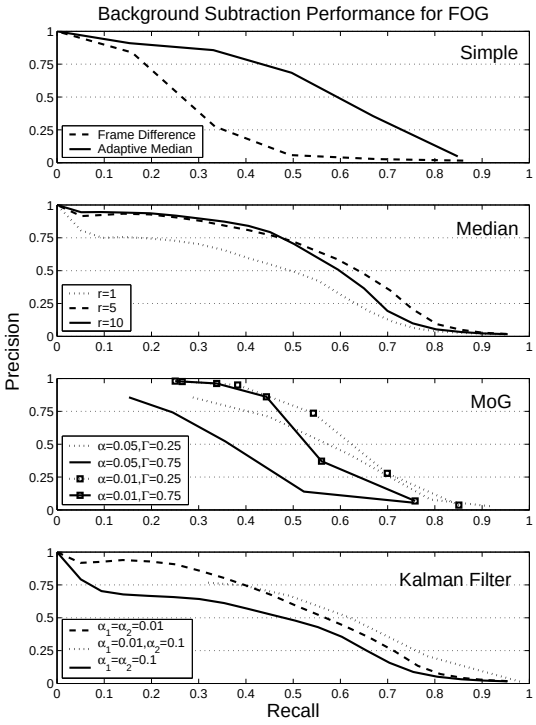
REFERENCES

1. A. Mittal and D. Huttenlocher, "Scene modeling for wide area surveillancd and image synthesis," in *Proceedings IEEE conference on computer vision and pattern recognition*, **2**, pp. 160–167, (Hilton Head Isand, SC), June 2000.
2. J. Kang, I. Cohen, and G. Medioni, "Continuous tracking within and across camera streams," in *Proceedings IEEE conference on computer vision and pattern recognition*, **1**, pp. 267–272, (Madison, WI), June 2003.
3. C. Wren, A. Azabayejani, T. Darrel, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**, pp. 780–785, July 1997.
4. R. Cutler and L. Davis, "View-based detection," in *Proceedings Fourteenth International Conference on Pattern Recognition*, **1**, pp. 495–500, (Brisbane, Australia), Aug 1998.
5. A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proceedings of IEEE ICCV'99 Frame-rate workshop*, Sept 1999.

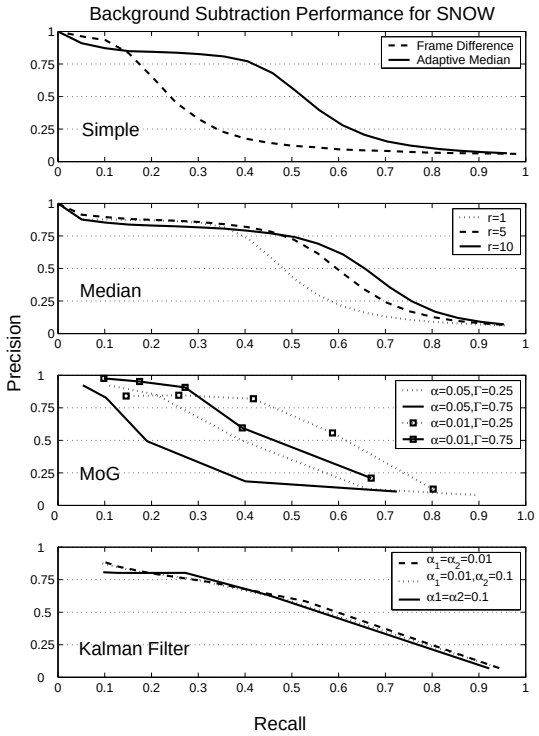
6. C. Stauffer and W. Grimson, "Learning patterns of activity using real-time tracking," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **22**, pp. 747–57, Aug 2000.
7. P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Proceedings of the 2nd European Workshop on Advanced Video-Based Surveillance Systems*, Sept. 2001.
8. M. Harville, "A framework for high-level feedback to adaptive, per-pixel, mixture-of-Gaussian background models," in *Proceedings of the Seventh European Conference on Computer Vision, Part III*, pp. 543–60, (Copenhagen, Denmark), May 2002.
9. D. R. Magee, "Tracking multiple vehicles using foreground, background, and motion models," in *Proceedings of the Statistical Methods in Video Processing Workshop*, pp. 7–12, (Copenhagen, Denmark), June 2002.
10. R. Cucchiara, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**, pp. 1337–1342, Oct 2003.
11. D. Koller, J. Weber, and J. Malik, "Robust multiple car tracking with occlusion reasoning," Tech. Rep. UCB/CSD-93-780, EECS Department, University of California, Berkeley, Oct 1993.
12. R. Pless, J. Larson, S. Siebers, and B. Westover, "Evaluation of local models of dynamic background," in *Proceedings IEEE conference on computer vision and pattern recognition*, **2**, pp. 73–78, (Madison, WI), June 2003.
13. N. Oliver, B. Rosario, and A. Pentland, "A Bayesian computer vision system for modeling human interactions," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**, pp. 831–843, Aug 2000.
14. I. Haritaoglu, D. Harwood, and L. Davis, "W^4: real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**, pp. 809–830, Aug 2000.
15. B. Gloyer, H. Aghajan, K.-Y. Siu, and T. Kailath, "Video-based freeway monitoring system using recursive vehicle tracking," in *Proceedings of SPIE*, **2421**, pp. 173–180, Feb 1995.
16. B. Lo and S. Velastin, "Automatic congestion detection system for underground platforms," in *Proceedings of 2001 International symposium on intelligent multimedia, video, and speech processing*, pp. 158–161, (Hong Kong), May 2001.
17. Q. Zhou and J. Aggarwal, "Tracking and classifying moving objects from videos," in *Proceedings of IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, 2001.
18. K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *ICCV (1)*, pp. 255–261, 1999.
19. N. McFarlane and C. Schofield, "Segmentation and tracking of piglets in images," *Machine Vision and Applications* **8**(3), pp. 187–193, 1995.
20. P. Remagnino *et al.*, "An integrated traffic and pedestrian model-based vision system," in *Proceedings of the Eighth British Machine Vision Conference*, pp. 380–389, 1997.
21. J. Heikkila and O. Silven, "A real-time system for monitoring of cyclists and pedestrians," in *Second IEEE Workshop on Visual Surveillance*, pp. 246–252, (Fort Collins, Colorado), Jun 1999.
22. G. Halevy and D. Weinshall, "Motion of disturbances: detection and tracking of multi-body non-rigid motion," *Maching Vision and Applications* **11**, pp. 122–137, 1999.
23. T. Boult *et al.*, "Frame-rate omnidirectional surveillance and tracking of camuflaged and occluded targets," in *Proceedings Second IEEE Workshop on Visual Surveillance*, pp. 48–55, (Fort Collins, CO), June 1999.
24. K.-P. Karmann and A. Brandt, "Moving object recognition using and adaptive background memory," in *Time-Varying Image Processing and Moving Object Recognition*, V. Cappellini, ed., **2**, pp. 289–307, Elsevier Science Publishers B.V., 1990.
25. N. Friedman and S. Russell, "Image segmentation in video sequences: A probabilistic approach," in *Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–97)*, pp. 175–181, Morgan Kaufmann Publishers, Inc., (San Francisco, CA), 1997.
26. P. W. Power and J. A. Schoonees, "Understanding background mixture models for foreground segmentation," in *Proceedings Image and Vision Computing New Zealand*, pp. 267–271, (Auckland, New Zealand), Nov 2002.
27. X. Gao, T. Boult, F. Coetzee, and V. Ramesh, "Error analysis of background adaption," in *Proceedings IEEE conference on computer vision and pattern recognition*, **1**, pp. 503–510, (Hilton Head Isand, SC), June 2000.
28. D.-S. Lee, J. Hull, and B. Erol, "A Bayesian framework for gaussian mixture background modeling," in *Proceedings of IEEE International Confererence on Image Processing*, (Barcelona, Spain), Sept 2003.
29. L. Fuentes and S. Velastin, "From tracking to advanced surveillance," in *Proceedings of IEEE International Conference on Image Processing*, (Barcelona, Spain), Sept 2003.
30. B. Horn, *Robot Vision*, The MIT Press, 1986.
31. D. Gutchess *et al.*, "A background model initialization algorithm for video surveillance," in *Proceedings Eighth IEEE International Conference on Computer Vision*, **1**, pp. 744–740, (Vancouver, BC), July 2001.
32. Y. Ivanov, A. Bobick, and J. Liu, "Fast lighting independent background," *International Journal of Computer Vision* **37**(2), pp. 199–207, 2000.
33. A. Prati, I. Mikic, M. Trivedi, and R. Cucchiara, "Detecting moving shadows: algorithms and evaluation," *IEEE Transactions on Pattern Analysis and Maching Intelligence* **25**, pp. 918–923, July 2003.
34. C. J. van Rijsbergen, *Information retrieval*, Butterworth & Co (Publishers) Ltd, second ed., 1979.



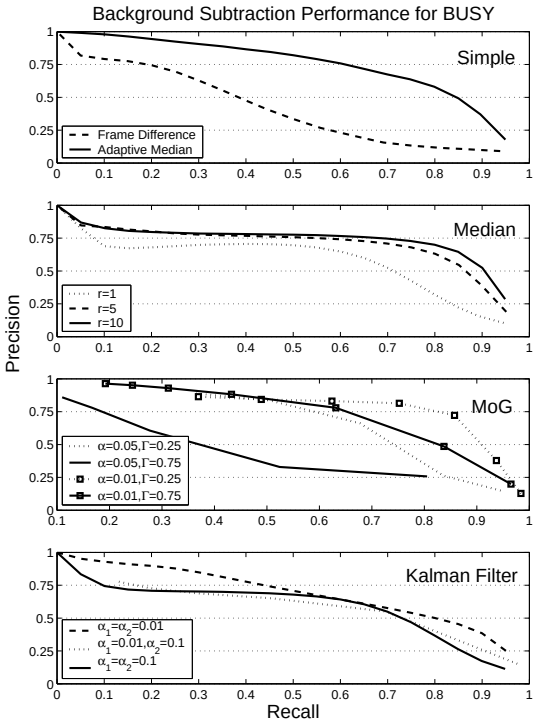
(a) Bright



(b) Fog

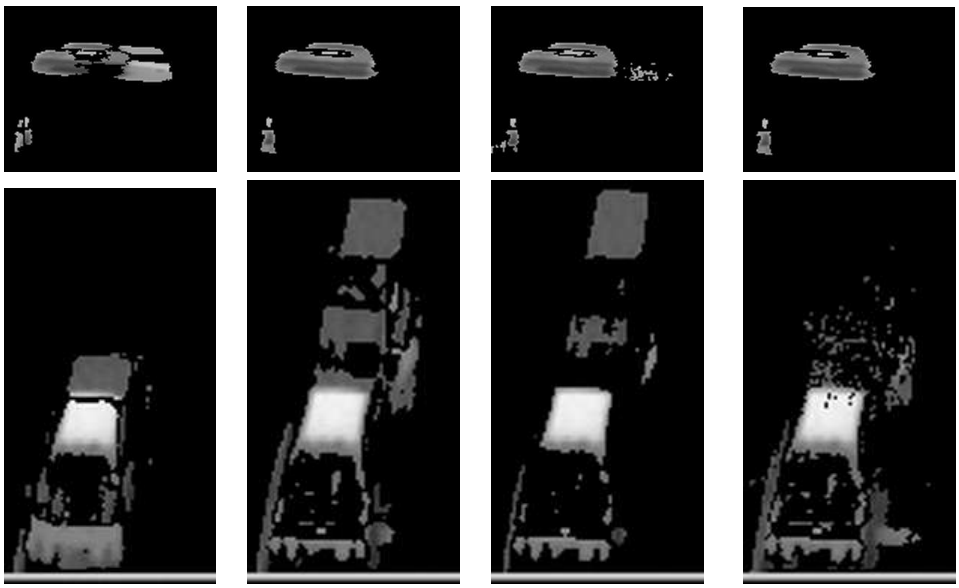


(c) Snow



(d) Busy

Figure 3. *Precision-recall plots for (a) Bright, (b) Fog, (c) Snow, and (d) Busy.*



(a) MF ($r = 1$) (b) MF ($r = 10$) (c) AMF (d) MoG ($\alpha = 0.01$, $\Gamma = 0.25$)

Figure 4. *Trailing tails and ghosts observed in the results of various background subtraction algorithms.*