Robust Foreground Detection in Video Using Pixel Layers

Kedar A. Patwardhan, *Student Member*, *IEEE*,
Guillermo Sapiro, *Senior Member*, *IEEE*, and
Vassilios Morellas, *Member*, *IEEE*

Abstract—A framework for robust foreground detection that works under difficult conditions such as dynamic background and moderately moving camera is presented in this paper. The proposed method includes two main components: coarse scene representation as the union of pixel layers, and foreground detection in video by propagating these layers using a maximum-likelihood assignment. We first cluster into "layers" those pixels that share similar statistics. The entire scene is then modeled as the union of such nonparametric layer-models. An incoming pixel is detected as foreground if it does not adhere to these adaptive models of the background. A principled way of computing thresholds is used to achieve robust detection performance with a prespecified number of false alarms. Correlation between pixels in the spatial vicinity is exploited to deal with camera motion without precise registration or optical flow. The proposed technique adapts to changes in the scene, and allows to automatically convert persistent foreground objects to background and reconvert them to foreground when they become *interesting*. This simple framework addresses the important problem of robust foreground and unusual region detection, at about 10 frames per second on a standard laptop computer. The presentation of the proposed approach is complemented by results on challenging real data and comparisons with other standard techniques.

Index Terms—Video analysis, scene analysis, foreground detection, surveillance, background subtraction, layer tracking.

1 Introduction and Previous Work

Robust detection of foreground and "interesting" or "unusual" events is an important precursor for many image and video applications, such as tracking, identification, and surveillance. Although, there is often no prior information available about the foreground object to be detected, in many situations the background scene is available in all frames of the video. This allows for detecting the foreground by "subtracting" the background from the scene, instead of explicitly modeling the foreground. Important factors that make detection very challenging are: 1) **Dynamic background** with water ripples, swaying trees, etc., 2) **Camera motion** due to support vibration, wind, etc. (which we will henceforth denote as "nominal" motion), and 3) **Real-time** or quasi-real-time **detection** requirements for most applications. These factors make detection of foreground by simple frame differencing (as in [1]) very difficult. Hence, in general, scene-modeling must be robust enough to accommodate pixel feature variation, pixel position uncertainty, and minimize or completely avoid pixel registration. In our proposed detection framework, the spatio-temporal correlation between pixels is exploited to provide priors on the set of pixel-clusters or layers that any pixel can belong to. This, in turn, allows to handle camera motion, without any explicit registration, while robustly detecting the foreground embedded in dynamic background, as illustrated by our results in Section 4 and in [2].

Prior approaches toward background modeling used single (e.g., [3]) or multiple (e.g., [4], [5], and more recently in [6]) mixture of Gaussians (MoG). We will later show that our proposed approach compares favorably against such models as well as with code-book type of approaches as in [7]. The problem of dynamic background has been recently addressed by Mittal and Paragios [8], where they propose an adaptive Kernel Density Estimation technique (first used in [9]) for modeling individual pixels, working well for a stationary camera but difficult to generalize to a moving camera. Recently, Sheikh and Shah [10] have proposed exploiting spatial correlation among pixels by using the position of a pixel along with the color. They model the entire scene, using a single 5D *pdf*.[1] The authors in [11] describe a three level algorithm where region-level and frame-level information is used to make decisions at the pixel-level. In [12], Zhong and Sclaroff have used a Kalman filter for modeling image regions as an autoregressive moving average (ARMA) process.

The concept of layered representation of a scene was first introduced by Adelson in [13], making a very good case for modeling a video scene as a group of layers instead of single pixels. There has been a large amount of notable work in this direction since, e.g., [14], [15], [16], [17], [18].

In our work, the scene is coarsely modeled as a group of layers in order to robustly detect the foreground under static or dynamic background and in the presence of nominal camera motion. The *training phase* determines the (initial) number of layers and helps to automatically compute bandwidths and thresholds required for the nonparametric modeling of layers and robust foreground detection. Though coarse *layer propagation* is a by-product of the foreground detection step, it is not the main focus of this work, and the reader should refer to the above mentioned works on layer-tracking for the state-of-the-art in that regard. Thus, the main contributions of our work are:

1. a principled way of extracting and automatically computing the number of scene-layers,[2]
2. different detection thresholds for different "layers" using a principled approach for threshold computation that allows for less than a prespecified number of false alarms (NFA),[3]
3. conversion of foreground layers to background and vice-versa based on global layer models, and
4. notion of background memory for each pixel, which helps to reduce false detections when foreground disoccludes the background.

These contributions lead to a simple and fast detection algorithm that can address nominal camera motion as well as dynamic background.

It must be noted that we do not constrain the pixels in one layer to be connected.[4] The main reasons for modeling the scene as a group of layers instead of individual pixels are: 1) to exploit spatio-temporal correlation between pixels, 2) use other similar pixels in the scene to model a pixel x, giving a better nonparametric estimate of the process that generated x, and 3) handling nominal camera motion without explicit registration since we are not constrained to look at instances of x at exactly the same spatial location in every frame (as for example in [8]).

Fig. 1 gives a brief overview of our algorithm, which is realized in two main steps: the *training or layer extraction step* and the

• *K.A. Patwardhan is with the Visualization and Computer Vision Lab, GE Global Research, One Research Circle, KW-C210, Niskayuna, NY 12309. E-mail: kedar.patwardhan@ge.com.*
• *G. Sapiro is with the Electrical and Computer Engineering Department, University of Minnesota, 200 Union St. S.E., Minneapolis, MN 55455. E-mail: guille@umn.edu.*
• *V. Morellas is with the Computer Sciences and Engineering Department, University of Minnesota, 200 Union St. S.E., Minneapolis, MN 55455. E-mail: morellas@cs.umn.edu.*

Manuscript received 12 June 2007; revised 20 Oct. 2007; accepted 27 Nov. 2007; published online 19 Dec. 2007.
Recommended for acceptance by H. Sawhney.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2007-06-0352.
Digital Object Identifier no. 10.1109/TPAMI.2007.70843.

1. Our proposed approach generates comparable results using a 3D feature space (see Fig. 6).
2. The model is sufficiently simple and accurate for the task of foreground detection.
3. It must be emphasized that in the proposed framework, **detection** is not the same as **segmentation**, i.e., a few pixels inside the object of interest with minimal false alarms are usually sufficient for detecting a foreground object.
4. For example, some part of the sky is visible through tree branches, in such cases, it would not be reasonable to put connectivity constraints which might lead to layers containing inconsistent pixels.

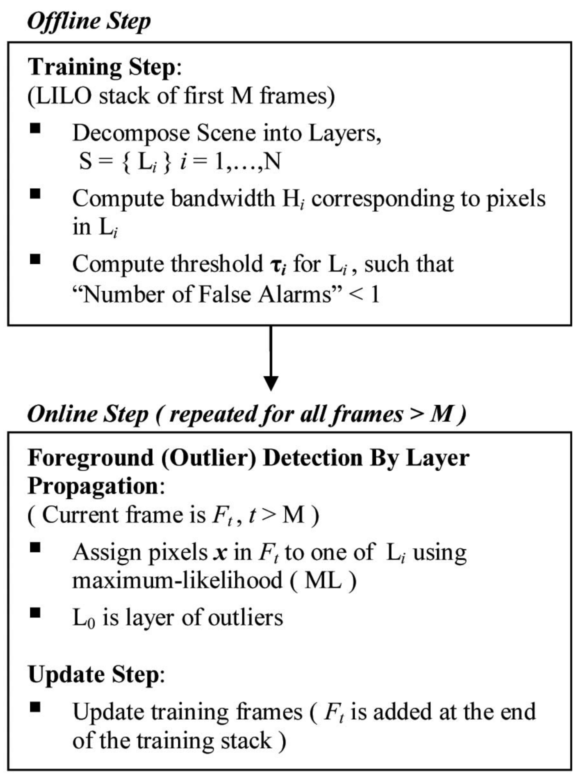

Offline Step
Training Step:
(LILO stack of first M frames)
Decompose Scene into Layers, S = { L_i } i = 1,…,N
Compute bandwidth H_i corresponding to pixels in L_i
Compute threshold τ_i for L_i, such that "Number of False Alarms" < 1
Online Step (repeated for all frames > M)
Foreground (Outlier) Detection By Layer Propagation:
(Current frame is F_t, t > M)
Assign pixels x in F_t to one of L_i using maximum-likelihood (ML)
L_0 is layer of outliers
Update Step:
Update training frames (F_t is added at the end of the training stack)

Fig. 1. Overview of the proposed framework.

foreground detection step. The following sections describe the proposed approach in more detail.

2 Automatic Image Layering

This section describes the method that we use to automatically cluster the pixels, which is the initial off-line (learning) step in Fig. 1. It should be noted that our layering approach has been developed to coarsely aggregate pixels that share similar statistical properties with the goal of detection and not for accurate image segmentation.

2.1 Extracting a Layer: Initial Guess

We first generate an initial guess of a layer that we wish to extract from the scene. We compute a local maximum of the image histogram (h_{max} at the gray value g_{max}[5]), and a radius (ρ) which is the square root of the trace of the global covariance matrix. These computations are similar to those in [19]. Thus, all pixels with gray-values between $g_{max} - \rho$ and $g_{max} + \rho$ form our initial guess or "layer-candidate" (L_C) of the layer to be extracted in the image domain.[6]

2.2 Extracting a Layer: Refinement Step

The refinement of this initial candidate is achieved by extending a very simple and elegant approach called Sampling-Expectation (SE) described in [21].[7] There are three main steps in this refining process:

5. We have assumed a gray-valued histogram in this discussion only for the sake of simplicity. Please note that we use the color of pixels as our feature-space.

6. We have used the "$L*u*v$" color-space for computing the initial-guess, because of the approximately isometric property of this color-space [20], which allows us to use a sphere of radius ρ around g_{max} when computing the initial-guess in the feature-space. For the rest of the work, we revert back to the "$r*g*S$" color-space used for example in [8], due to its robustness to illumination changes.

7. In [21], the SE algorithm is used only for figure-background segmentation with the assumption that the figure is situated roughly at the center of the image.

Initialization. It is assumed that the pixels in the candidate layer and the rest of the image are realizations of two separate random processes. To initialize the process, we start with an initial (spatial) probability distribution P_{Lc} on the image pixels, where, pixels belonging to L_C have high P_{Lc} values and pixels not in L_C get low values. This distribution P_{Lc} indicates our (initial) confidence about the chance that a particular image pixel belongs to L_C. Similarly, P_{bg} forms the competing background process, i.e., $P_{bg} = 1 - P_{Lc}$.

S-step. The image is uniformly sampled to get a set of samples $S = \{x^i\}_{i=1}^m$. Generally a sample size of about 10 to 20 percent of the pixels in the image has been found to be satisfactory. Each of these samples has two probability values associated with it, i.e., $P_{Lc}(x^i)$ and $P_{bg}(x^i)$ (from the first step above) which are used in (1) and (2). For more information see the description below.

E-Step. The distributions P_{Lc} and P_{bg} are re-estimated using (3) and (4) respectively. The pixels are then assigned to or removed from L_C based on maximum-likelihood.

The S and E steps are iterated until the composition of L_C becomes stable. In the above algorithm, the likelihood of a pixel belonging to one of the two processes is refined using a weighted Kernel Density Estimation (KDE). Kernel Density Estimation is a fairly common technique for estimating the probability density of data in a nonparametric fashion. For details about the concept of KDE, the reader is referred to [22]. Given a pixel y belonging to the image,[8] we estimate the probabilities $P_{Lc}(y)$ and $P_{bg}(y)$ by first computing the following weights (as in [21]):

$$W_{Lc}(y) = \sum_{i=1}^{m} P_{Lc}(x_i) \prod_{j=1}^{d} K\left(\frac{y_j - x_{ij}}{h_j}\right), \tag{1}$$

$$W_{bg}(y) = \sum_{i=1}^{m} P_{bg}(x_i) \prod_{j=1}^{d} K\left(\frac{y_j - x_{ij}}{h_j}\right), \tag{2}$$

where, the subscript j indicates the jth element of the d-dimensional vector, K is the kernel or smoothing function (we use a Gaussian kernel), d is the dimension of the feature-space (three in our case, five if we incorporate optical flow for example), and h_js are the kernel bandwidths, which we estimate using $h_j \approx 1.06\hat{\sigma}_j m^{\frac{1}{5}}$ as in [23]. Here $\hat{\sigma}_j$ is the standard deviation estimated over the sample S (S-step above), in dimension j, and m is the number of samples. The pixel probabilities are then re-estimated as

$$P_{Lc}(y) = W_{Lc}/(W_{Lc} + W_{bg}), \tag{3}$$

$$P_{bg}(y) = W_{bg}/(W_{Lc} + W_{bg}). \tag{4}$$

It should be noted that we do not put explicit constraints of spatial smoothness in the layer extraction process. The initialization of P_{Lc} and P_{bg} can be spatially smooth, if we want to "encourage" spatial smoothness in the layers.

2.3 Extracting a Layer: Multiple Layers and Validation

This layer extraction (initial guess and refinement) is performed iteratively to extract all layers in an image. The previously considered histogram maxima are suppressed during this process. In order to ascertain that the extracted layer is both meaningful and of significant size (based on number of member pixels), we propose the use of a Kullback-Leibler (KL) divergence. First (before beginning the layer extraction process), assume that the entire image is the candidate layer (i.e., $\rho \rightarrow \infty$). After a few refinement steps, compute the KL divergence between the foreground and background distributions (let us call this KL_{before}) as follows:

8. The variables y and x indicate the feature vectors of different pixels, not their position.

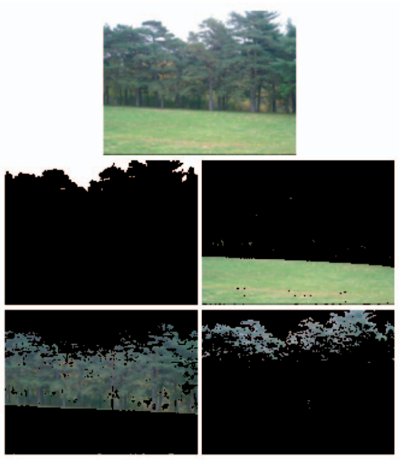


Fig. 2. Original (top) image is automatically decomposed into four layers (bottom two rows). Observe that the refinement step ensures consistency in the layers and more accurately defines their boundaries.

$$KL(P_{Lc} \| P_{bg}) = -\sum_{i=0}^{m} P_{Lc}(x^i) log\left(\frac{P_{bg}(x^i)}{P_{Lc}(x^i)}\right), \quad (5)$$

where, $x^i \in S$ is the sample collected in the S-step and m is the total number of samples. Now, find the real candidate layer (finite ρ), perform the refinement and after this layer is stable, compute the KL divergence (let us call this KL_{after}), using again (5). As long as the data (image) supports the initial guess and the refined layer is meaningful (has some significant number of pixels), the condition $KL_{after} > KL_{before}$ will hold. We consider the layer to be valid if this condition is satisfied. The layering is stopped when there are no more valid layers possible. Thus, the number of extracted layers is determined automatically.[9]

Fig. 2 shows the complete layering of a natural scene. These layers are very similar to how a human observer would segregate the scene. Pixels in these layers belong to the same cluster in the feature space and are also spatially connected as seen in the image. These and more examples can be seen at http://www.tc.umn.edu/~patw0007/videolayers/.

The initial training stack (T) used for training the background model consists of the first M frames in the sequence. The first frame is layered using the above technique and the remaining frames in T are layered using the layer labels in the previous frame as a starting point for the refinement step.[10]

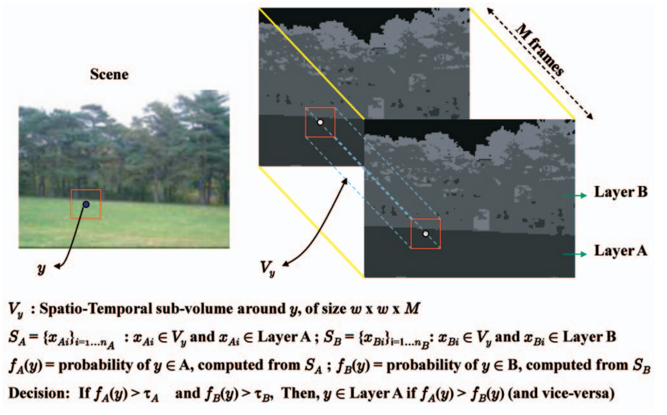
3 Foreground Detection in Video

Once the initial training stack T is layered, what remains to be done is to continually assign all incoming pixels to one of the existent layers in the scene, **or** identify them as outliers/foreground (assign to layer L_0), **or** identify them as part of a temporally persistent (or *uninteresting*) foreground object and assign them to a completely new background layer. The following sections describe this assignment process in detail.

Density estimation. We consider that there exists meaningful correlation between pixels in the spatial vicinity. To imbibe this correlation into our framework, we use a parameter w which indicates the *registration uncertainty* or the *spatial variance* of a pixel. This parameter, which may be user-defined, defines the size $w \times w \times M$, where M is the number of frames in the stack T, of the spatio-temporal-past of pixels in the training stack T that may be

9. Unlike in [19], extracted layers are not constrained to be connected or of a particular minimum size, which avoids inconsistent assignment of pixels by force.

10. Even though we use labels from previous frames as initial guess, in the end we also check for appearance of completely new layers which may not have been previously seen.



V_y : Spatio-Temporal sub-volume around y, of size w x w x M

$S_A = \{x_{Ai}\}_{i=1...n_A}$: $x_{Ai} \in V_y$ and $x_{Ai} \in$ Layer A ; $S_B = \{x_{Bi}\}_{i=1...n_B}$: $x_{Bi} \in V_y$ and $x_{Bi} \in$ Layer B

$f_A(y)$ = probability of $y \in$ A, computed from S_A ; $f_B(y)$ = probability of $y \in$ B, computed from S_B

Decision: If $f_A(y) > \tau_A$ and $f_B(y) > \tau_B$, Then, $y \in$ Layer A if $f_A(y) > f_B(y)$ (and vice-versa)

Fig. 3. Assignment of a pixel $\mathbf{y}$ to foreground or background. Gray-scales indicate the different layer labels in the scene.

correlated to the current pixel. Now consider a pixel $\mathbf{y}$ in the current frame to be analyzed, refer to Fig. 3. The spatio-temporal subvolume $\mathbf{V_y}$ $(w \times w \times M)$ in the training stack tells us that $\mathbf{y}$ can belong only to layers A, B in the background (see Fig. 3), or the layer of outliers (foreground). We now need to compute the probability of $\mathbf{y}$ belonging to either of these layers. Let us first find this probability in the case of layer A. All pixels in the training stack, that belong to the subvolume $\mathbf{V_y}$ and are labeled as belonging to layer A, form the sample $\mathbf{S_A} = \{\mathbf{x}_{\mathbf{A}_i}\}_{i=0}^{n_A}$. Now, the probability $f_A(\mathbf{y})$ is computed using a Nonparametric Kernel Density Estimator with a Gaussian kernel $\mathbf{K}$,

$$f_A(\mathbf{y}) = \frac{1}{n_A}\sum_{i=0}^{n_A} \frac{1}{\|\mathbf{H_A}\|^{1/2}} \mathbf{K}\left(\mathbf{H_A}^{-1/2}(\mathbf{y} - \mathbf{x_{Ai}})\right), \quad (6)$$

where $\mathbf{H_A}$ is the bandwidth matrix for layer A. Similarly, we compute the value $f_{(.)}(y)$ corresponding to other layers in $\mathbf{V_y}$. The bandwidth matrix is assumed to be diagonal, i.e., $\mathbf{H_A} = h_A^2\mathbf{I}$, which is computed during the training step. The Kernel Density Estimates are computed using the *Improved Fast Gauss Transform* (IFGT) described in [24].[11] The layer of outliers (L_0) also contributes samples to the likelihood-computation, when there is a previously detected outlier in $\mathbf{V_y}$. Thus, there is a competitive classification between outliers and background, at the same time, coarsely propagating the background layers throughout the video. The pixel $\mathbf{y}$ is assigned to that layer-model (say A), which has the maximum probability of generating $\mathbf{y}$, as long as this probability is greater than the threshold for that layer (i.e., $f_A(\mathbf{y}) > \tau_A$, see below for comments on the automatic computation of this threshold). If no layer-model satisfies this criterion, then $\mathbf{y}$ is classified as an outlier by default. Thus, a pixel can be detected as a foreground pixel (outlier) in two cases, either when the pixel does not belong to any other background layer, or when the pixel belongs to the layer of outliers (L_0) by maximum-likelihood assignment.

Automatic threshold computation. In most of the previous work mentioned in Section 1, the thresholds for detecting an outlier are quite arbitrary and require manual tuning. In our case, depending upon the homogeneity and integrity of the pixels belonging to a layer, each layer will need to have a different threshold to achieve the same "Number of False Alarms" (NFA) rate. In order to avoid any arbitrariness in automatically computing these thresholds (τs), we use a principled way with the help of the *a-contrario* framework [25], [26]. More details can be found in the longer version of this paper [2].

Background memory. In our implementation of the proposed framework, we employ a strategy of memorizing the models of the

11. Our unique representation of the scene using homogeneous layers allows to adapt the IFGT to the introduction of new samples, for more details see [2].

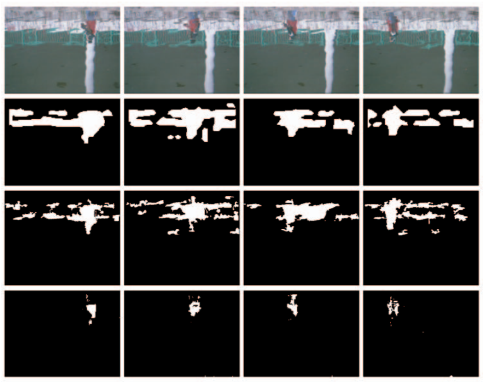


Fig. 4. Qualitative comparison of the proposed algorithm (last row), with Gaussian Mixture Modeling (second row) and Code-book Modeling (third row) type of approaches. The top row shows the reflections of a person (foreground) on dynamic water ripples.

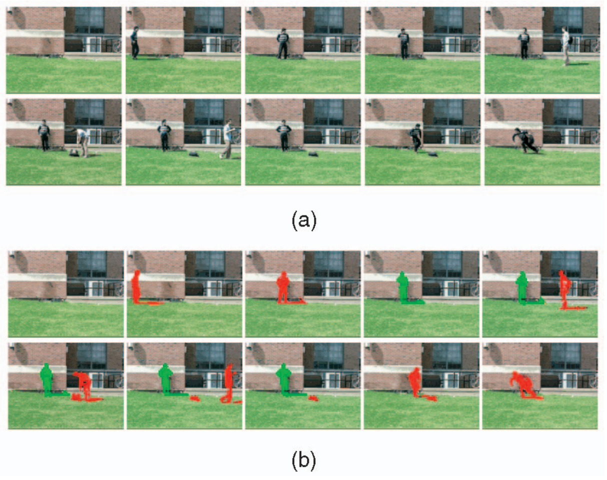


Fig. 5. Example of a video where detections are based on temporal-persistence. Temporally persistent (uninteresting) foreground is converted to background (which is tracked nonetheless as a layer, shown in green) and then converted back to foreground when it becomes interesting again. Please see complete video at http://www.tc.umn.edu/~patw0007/videolayers/. (a) Original frames of a video sequence with multiple foreground objects. (b) Detection results, the foreground is colored in red. The first person is detected as foreground initially but converted to a background layer (indicated by green) after he becomes temporally persistent (uninteresting), fourth figure. The same person is converted to foreground again when he becomes interesting, last two figures.

layers previously seen at a particular pixel location. Let pixel $\mathbf{y}$ at spatial location (x, y) be assigned to layers $\mathcal{L} = \{A, B, \ldots\}$ in the past.[12] Later, if $\mathbf{y}$ is assigned to a persistent outlier which occludes the background layers, we keep testing the pixel $\mathbf{y}$ against the layer-models $\mathcal{L}$, so that when the persistent outlier moves away and disoccludes the background, $\mathbf{y}$ can again be assigned back to one of the background layers instead of being falsely detected as an outlier. An example can be seen in Fig. 4.

Temporal Persistence. When objects in the foreground are temporally persistent (uninteresting, e.g., a car parked at a parking lot), we can convert them into separate background layers, and re-convert them into foreground when they are interesting again (e.g., car moving out of parking lot). This idea has been previously explored at pixel-level in [27], but the use of layers allows an elegant way of applying such temporal constraints in a global fashion. Fig. 5 shows an example of such a situation. More details can be found in [2].

12. This needs to be done, because the surrounding region $\mathbf{V}_{\mathbf{y}}$, see Fig. 3, may not contain some layers from the whole considered past $\mathcal{L}$.

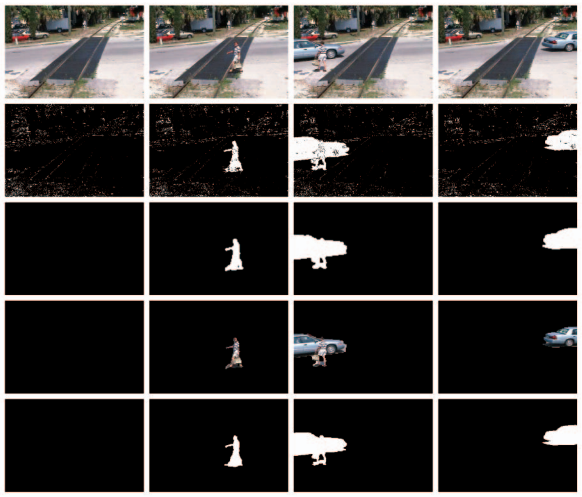


Fig. 6. Foreground detection with average camera motion of about 14.66 pixels in arbitrary directions. Top row shows the original frames. The second row shows foreground detection results using a Mixture-of-Gaussians (MoG) model. Third row shows results from the proposed approach. The fourth row, shows the exact image pixels that are detected as foreground with our approach, while the last row shows the ground-truth segmentation from [10]. (This is a color figure.)

4 Implementation Details and Experimental Results

The videos up to Fig. 5 (and others available at http://www.tc.umn.edu/~patw0007/videolayers/) are outdoor scenes of resolution 160×120. The algorithm was implemented using C++, on a machine with Intel-Pentium IV 1.8 GHz processor. In the offline training step, we used an initial training stack of approximately 30 frames for all the results,[13] achieving a running speed of 10 frame/second with our experimental code (for the "online" detection and update process in Fig. 1). The initial layering and training steps usually require about 5 minutes (for layering all the frames in the training stack using the technique described in Section 2).

Fig. 4 shows a qualitative comparison of our approach with those in [6] and [7], especially in the presence of highly dynamic background provided by ripples in the water. The scene model using layers and detection thresholds are robust to such dynamic background.

Fig. 5 shows an example of a temporally persistent object in the scene being converted to a new background layer (which is still continuously tracked as a layer) and then reconverted to a foreground object when it becomes interesting again.

In order to provide a quantitative perspective about the quality of foreground detection with our approach, we have used a test sequence and the corresponding ground-truth segmentation from [10].[14] Fig. 6 first shows a qualitative illustration of the results as compared to the ground truth and a Mixture of Gaussians (MoG) model, and the corresponding quantitative comparison is shown in Fig. 7. It can be seen that the proposed approach has almost no false detections in spite of the camera motion. The sequence contains average nominal motion of about 14.66 pixels. Fig. 7a shows a comparative plot of the total number of pixels detected as foreground. It can be seen that the MoG model produces a large number of false alarms even when we use a mixture of five Gaussians.

13. This is the number of frames that are "layered" using the algorithm mentioned in Section 2. This training stack (or moving temporal window) can be later extended to 100-200 frames during the online process.

14. Note again that this comparison is for completeness mainly, since our proposed framework attempts only at detection and not full segmentation.

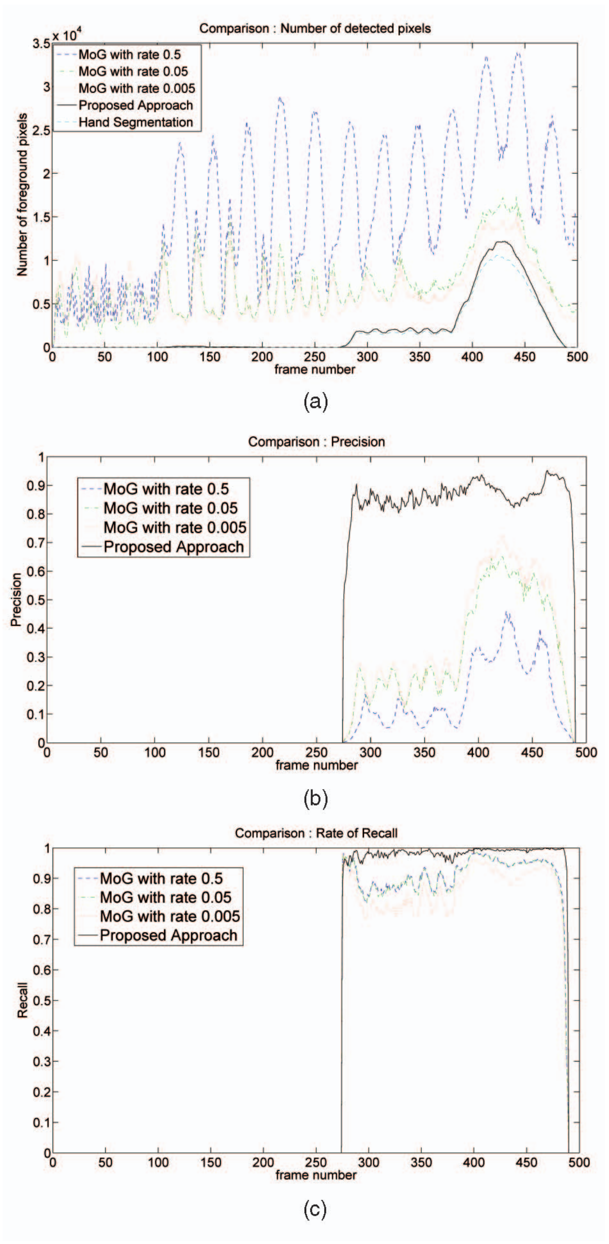


Fig. 7. Quantitative comparison. (a) indicates the number of pixels detected as foreground using various learning rates of the MoG model, as well as the proposed approach, against the ground-truth manually segmented video. (b) compares the *Precision* and (c) compares the *Recall* rates with various learning rates of the MoG model against our approach.

Figs. 7b and 7c also compares the proposed approach with a 5-component MoG model with respect to the precision and recall rates computed as

$$\text{Precision} = \frac{\text{Number of true positives detected}}{\text{Total number of positives detected}};$$

$$\text{Recall} = \frac{\text{Number of true positives detected}}{\text{Total number of true positives}}.$$

The comparison is highly favorable to our approach and as shown in Fig. 8, our algorithm is also able to detect a person in the distant background, which is not detected even in the ground-truth segmentation in [10] probably because it is very difficult to discriminate visually.

Fig. 9 illustrates the importance of robust pixel level detections. If we perform a spatial consistency operation (like 3×3 median filtering), then the detection results from our approach and a MoG model look alike (Fig. 9d), and this hides the intrinsic robustness of the detections in our approach. In Fig. 9b, it can be seen that the proposed approach can correctly detect a person in the distant background, whereas a spatial consistency operation on the MoG result simply produces noisy detections. Hence, it is imperative that, in spite of the use of spatial filtering to refine the detection, the original detections must be highly robust. Additional examples and details can be found in [2].

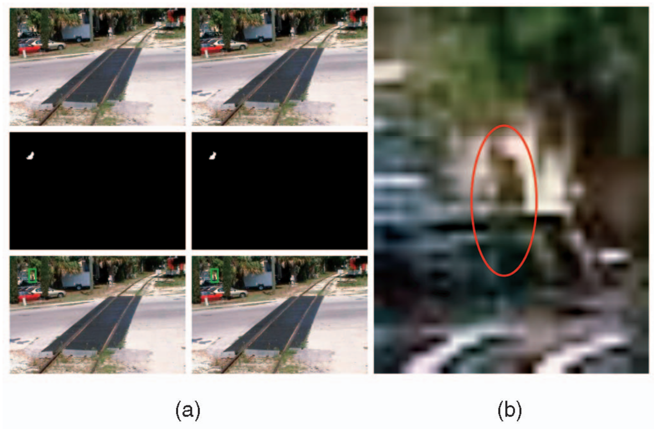


Fig. 8. (a) Proposed approach can detect a person of very small size in the distant background (indicated by green rectangle in bottom row). Detection masks are shown in the middle row. (b) A blow-up of the detected person is shown here highlighted in a red ellipse.

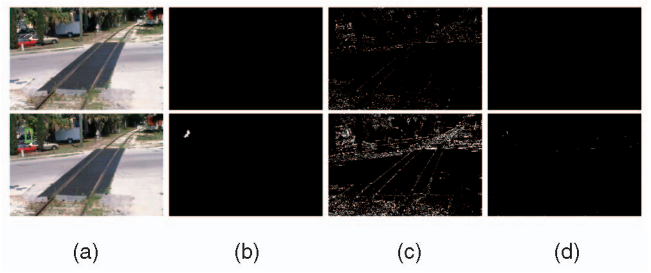


Fig. 9. The quality of original detections is very important. (a) shows original frame, (b) shows the detection results with the proposed approach, (c) column shows detections with a MoG model, and (d) shows the same MoG result after using a 3×3 median filter as an example of spatial filtering.

5 Discussion and Future Scope

In this work, we have proposed a simple framework for coarse scene modeling and foreground detection using pixel-clusters (layers), which allows for integrated analysis and detection in a video scene. The scene is modeled coarsely as a union of pixel clusters or layers. The task at hand then is to assign any incoming pixel to one of these layers or to the foreground. Thresholds are chosen in a nonarbitrary fashion to give robust outlier detections. The results presented show very satisfactory performance in difficult environments. The proposed approach also addresses most of the challenging situations mentioned in [11] like:

1. gradual movement of background layers or objects (refer Section 3),
2. gradual changes in illumination, by using a sufficiently illumination insensitive color space,
3. waving trees or reflections in water ripples (see Fig. 4),
4. foreground aperture (this is avoided partly by not using optical flow),
5. sleeping person (temporal persistence), and
6. waking person (background memory).

The layer-based scene representation is a flexible framework that allows for various possible future applications (apart from robust foreground detection): 1) *Anomaly detections*: where the layer-models learned from one scene (expected data) can be used to find nonconformist outliers in a completely different scene, 2) *layer transfer*: where layer-models learned from one scene are used to index/search for similar layers in another scene, and 3) *multicamera person matching*: Using layered representation of a foreground person to generate an appearance model that detects across cameras. Preliminary results in these directions can be seen at http://www.tc.umn.edu/~patw0007/videolayers/. In the future, we would like to adapt the framework described here to multicamera scenarios where the different cameras may or may not overlap and also may be of different modalities. The foreground models of moving persons should be made more robust, for example by adding shape information to the global feature-set, toward their use in person identification and tagging throughout the area of surveillance. In order to be able to perform even higher level (than illustrated in this work) detections of anomalous behavior (outliers), it is important to include "interregion" (or interlayer) relationships into the background model. Results in these directions will be reported elsewhere.

ACKNOWLEDGMENTS

This work is partially supported by the US Office of Naval Research, the US National Science Foundation, NGA, ARO, and the US Defense Advanced Research Projects Agency. The authors would like to acknowledge the help of Honeywell Labs and Professor Larry Davis for providing comparative results shown in Fig. 4. The authors are also grateful to Professor Mubarak Shah and Dr. Yaser Sheikh for providing them with the original video and segmented ground-truth shown in Fig. 6. The AE Dr. Harpreet Sawhney and the reviewers selected by him have helped significantly in improving the presentation of key ideas and contributions of this work.

REFERENCES

[1] R.C. Jain and H.H. Nagel, "On the Analysis of Accumulative Difference Pictures from Image Sequences of Real World Scenes," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 1, no. 2, pp. 206-213, Apr. 1979.

[2] Detailed version of this paper available at http://www.tc.umn.edu/~patw0007/videolayers/, June 2006.

[3] C.R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-Time Tracking of the Human Body," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 7, pp. 780-785, July 1997.

[4] N. Friedman and S.J. Russell, "Image Segmentation in Video Sequences: A Probabilistic Approach," *Proc. 13th Conf. Uncertainity in Airtificial Intelligence,* pp. 175-181, 1997.

[5] C. Stauffer and W.E.L. Grimson, "Learning Patterns of Activity Using Real-Time Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 747-757, Aug. 2000.

[6] V. Morellas, I. Pavlidis, and P. Tsiamyrtzis, "Deter: Detection of Events for Threat Evaluation and Recognition," *Machine Vision Application,* vol. 15, no. 1, pp. 29-45, 2003.

[7] K. Kim, T.H. Chalidabhongse, D. Harwood, and L.S. Davis, "Real-Time Foreground-Background Segmentation Using Codebook Model," *Real-Time Imaging,* vol. 11, no. 3, pp. 172-185, 2005.

[8] A. Mittal and N. Paragios, "Motion-Based Background Subtraction Using Adaptive Kernel Density Estimation," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* pp. 302-309, 2004.

[9] A.M. Elgammal, D. Harwood, and L.S. Davis, "Non-Parametric Model for Background Subtraction," *Proc. European Conf. Computer Vision,* pp. 751-767, 2000.

[10] Y. Sheikh and M. Shah, "Bayesian Modeling of Dynamic Scenes for Object Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 11, pp. 1778-1792, Nov. 2005.

[11] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and Practice of Background Maintenance," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 255-261, 1999.

[12] J. Zhong and S. Sclaroff, "Segmenting Foreground Objects from a Dynamic Textured Background via a Robust Kalman Filter," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 44-50, 2003.

[13] E.H. Adelson, "Layered Representation for Vision and Video," *Representation of Visual Scenes, in conjunction with Proc. IEEE Int'l Conf. Computer Vision,* p. 3, 1995.

[14] H. Tao, H.S. Sawhney, and R. Kumar, "Object Tracking with Bayesian Estimation of Dynamic Layer Representations," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 1, pp. 75-89, Jan. 2002.

[15] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov, "Bilayer Segmentation of Live Video," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* pp. 53-60, 2006.

[16] Y. Weiss, "Smoothness in Layers: Motion Segmentation Using Nonparametric Mixture Estimation," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* p. 520, 1997.

[17] S. Khan and M. Shah, "Object Based Segmentation of Video Using Color Motion and Spatial Information," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 746-751, 2001, citeseer.ist.psu.edu/khan01object.html,

[18] Y. Zhou and H. Tao, "A Background Layer Model for Object Tracking through Occlusion," *Proc. IEEE Int'l Conf. Computer Vision,* p. 1079, 2003.

[19] D. Comaniciu and P. Meer, "Robust Analysis of Feature Spaces: Color Image Segmentation," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* p. 750, 1997.

[20] G. Wyszecki and W.S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae.* Wiley, 1982.

[21] L. Zhao and L.S. Davis, "Iterative Figure-Ground Discrimination," *Proc. Int'l Conf. Pattern Recognition,* pp. 67-70, 2004.

[22] B.W. Silverman, *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, 1986.

[23] D.W. Scott, *Multivariate Density Estimation.* Wiley Inter-Science, 1992.

[24] C. Yang, R. Duraiswami, N. Gumerov, and L. Davis, "Improved Fast Gauss Transform and Efficient Kernel Density Estimation," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 464-471, 2003, citeseer.ist.psu.edu/yang03improved.html,

[25] F. Cao, Y. Gousseau, P. Muse, F. Sur, and J.-M. Morel, "Accurate Estimates of False Alarm Number in Shape Recognition," technical report, 2004, http://www.cmla.ens-cachan.fr/Cmla/, citeseer.ist.psu.edu/muse04accurate.html.

[26] T. Veit, F. Cao, and P. Bouthemy, "An a contrario Framework for Motion Detection," Technical Report 5313, INRIA, 2004, http://www.irisa.fr/vista/Publis/Auteur/Frederic.Cao.english.html,

[27] D.H. Kyungnam Kim and L.S. Davis, "Background Updating for Visual Surveillance," *Proc. Int'l Symp. Visual Computing,* 2005.