PIXEL LAYERING AND LAYER PROPAGATION FOR VIDEO MODELLING AND FOREGROUND DETECTION

By

Kedar A. Patwardhan

Guillermo Sapiro

and

Vassilios Morellas

IMA Preprint Series # 2080

(November 2005)

INSTITUTE FOR MATHEMATICS AND ITS APPLICATIONS

UNIVERSITY OF MINNESOTA
400 Lind Hall
207 Church Street S.E.
Minneapolis, Minnesota 55455–0436

Phone: 612/624-6066   Fax: 612/626-7370
URL: http://www.ima.umn.edu

Pixel Layering And Layer Propagation For Video Modelling And Foreground Detection

Kedar A. Patwardhan Guillermo Sapiro
ECE Department, University of Minnesota,
200 Union St. S.E., Minneapolis, MN 55455, USA
kedar,guille@umn.edu

Vassilios Morellas
Automation and Control Solutions Lab, Honeywell
3600 Technology Dr., Minneapolis, MN 55418, USA
vassilios.morellas@honeywell.com

Abstract

Computer vision applications that work with videos often require that the foreground, region of interest, be clearly segmented from the un-interesting background. To address this problem, we present a general framework for scene modelling and robust foreground detection that works under difficult conditions such as moving camera and dynamic background. This is achieved by first representing the scene as a union of pixel layers, and then propagating these layers through the video by a maximum-likelihood (ML) assignment of pixels to the different layers. The possibility of a pixel not belonging to any of the layers in the scene is also one of the hypotheses that are automatically tested during the maximum-likelihood assignment.

The proposed approach has a number of salient virtues. Firstly, the clustering and layering is automatic, while the feature-space can be user defined to suit the application. Secondly, the cluster propagation step implicitly performs layer tracking along with foreground detection. Standard pixel based scene modelling techniques become a particular case of our general framework, when all pixels in the scene are independent and distinct from each other and belong to separate clusters. It is observed that pixels belonging to the same clusters in the feature space usually map to spatially connected layers in the image space, leading us to consider that useful correlation exists between features of pixels in the spatial vicinity. This permits to deal with camera motion with none or nominal registration. We illustrate our ideas with a number of interesting and difficult real-life examples.

1. Introduction And Previous Work

Robustly segmenting foreground is an important requirement for many computer vision algorithms including tracking, identification, and surveillance. Although there is often no prior information available about the foreground object to be segmented, in most situations the background scene is available in all frames of the video, and hence can be learned or modelled. This allows for segmenting the foreground by "subtracting" the background from the scene, instead of explicitly modelling the foreground.

Background subtraction by simple frame differences as in [6] has not been very successful in most real-life situations. Natural scenes often consist of dynamic elements like ripples in water, trees swaying with wind, escalators at airports, and moving crowds, making background modelling in video a very challenging problem. In [4], a statistical modelling of the background is performed using the assumption that the imaging sensor (camera) is completely stationary, to simplify the problem (although the problem still remains challenging). Camera stationarity is a tight constraint, and in many practical scenarios, the assumption of a stationary sensor is violated, as for example in the case of camera shake due to wind, support vibrations, hand instability (in case of hand-held cameras), panning or tilting (in case of a PTZ-camera), and general camera motion induced by embedding cameras on flying drones. Wada *et al*. [15] have proposed a method for handling pan-tilt-zoom camera motion by using an appearance sphere. Planar camera motion can also be handled by precise registration of the frames to create a background image which is then subtracted from the scene to segment the foreground. Such type of registration is very sensitive to noise and in most natural cases is not very precise (and also time consuming).

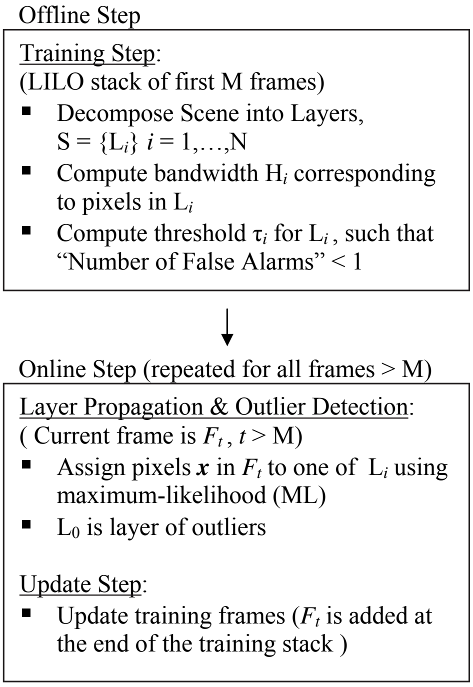
Thus, scene-modelling must be robust enough to accommodate pixel position uncertainty, and minimize or completely avoid pixel registration. In our proposed framework, the spatio-temporal correlation between pixels is exploited to provide priors on the set of pixel-clusters or layers that any pixel can belong to. This, in turn, allows to handle camera motion, without any explicit registration, while robustly detecting the foreground as illustrated by our results in Section 4.

Background modelling techniques are categorized mainly as *pixel* based and *region* based methods. The majority of background modelling methods in current literature are of the first type. The initial approaches in this regard (most notably [16]) assumed that the *pdf* of a pixel at location (x, y) can be modelled by a single 3-D Gaussian distribution $N(\mu(x, y), \Sigma(x, y))$. The mean ($\mu$) and variance ($\Sigma$) are estimated at each pixel position over a set of images, and then the likelihood of a pixel belonging to the background can be computed, in order to assign it to the background or foreground. Finding a single Gaussian unsuitable to model a color pixel, in [5, 12], and more recently in [8], each pixel is modelled as a mixture of a pre-determined number of Gaussians. This helps to address the different "modes" in the behavior of each pixel. The problem of dynamic background has been recently addressed by Mittal and Paragios [7], where the authors propose an adaptive Kernel Density Estimation technique that uses optical flow as well as color to generate a 5-D *pdf* for each pixel. This technique works well with a stationary camera constraint, but is difficult to generalize to a moving camera scenario. We provide a more general framework which exploits spatial correlation among pixels to handle camera motion. Critical thresholds in [7] are automatically handled in our work using an *a-contrario* model.

In region based methods, the background is modelled as a group of regions. The authors in [13] describe a three level algorithm where region-level and frame-level information is used to make decisions at the pixel-level. In [19], Zhong *et al.* have used a Kalman filter for modelling image regions as an autoregressive moving average (ARMA) process. Recently, Sheikh and Shah [10], have proposed exploiting spatial correlation among pixels by using the position of a pixel along with the color to generate a single 5-D *pdf* that models the entire image.

In this paper, we present a more generalized and simultaneously simpler method for image layering or pixel cluster based background modelling. The background is first segmented into "easy to model" clusters of pixels. Subsequently, pixels of incoming frames are either assigned to one of the existing clusters (layers) based on maximum-likelihood, or categorized as "outliers" via automatic threshold computation. This classification helps to propagate clusters in downstream image frames while achieving a finer and more correct definition of layers. The following sections describe our framework in more detail.

1.1. Algorithm Overview


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)

Layer Propagation & Outlier Detection:
(Current frame is F_t, t > M)
- Assign pixels $\boldsymbol{x}$ 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)

Figure 1. *Overview of the proposed modelling and foreground detection framework*

Figure 1 gives a brief overview of our algorithm. The first few frames (say M) of a video sequence form the "Last-In-Last-Out" (LILO) stack of training frames. Frames in this stack are layered into N layers,[1] $\{L_i\}_{i=1}^{N}$, using a Sampling-Expectation (SE) algorithm proposed in [18]. We simultaneously and automatically compute the bandwidths (H_i's), and thresholds (τ_i's) corresponding to each layer (L_i), that are required for the ML-based pixel assignment (refer to Section 3).

In the layer propagation step, each pixel is assigned to one of the layers using maximum-likelihood. The possible layers that each pixel can belong to are short-listed by observing the layer labels from the training stack in the spatio-temporal vicinity of the pixel. The layer L_0 (initially empty) is the layer of outliers. Outliers are detected using the thresholds (τ_i) automatically computed during the training step, see Section 3.2. Once all the pixels are assigned to a layer, the frame is added ("pushed") at the end of the LILO stack and the oldest frame in the stack is released ("popped").

[1] The words "layering" and "clustering" have been used interchangeably as it is often observed that clusters in the feature-space usually correspond to connected layers in the image domain.

Section 2 and Section 3 describe in detail the main components of the algorithm. We then proceed by presenting some interesting and challenging results in Section 4. Discussion of limitations of the proposed framework and future work is given in Section 5.

2. Automatic Layering

In our framework, the scene is modelled as a group of pixel clusters or layers. This is done in order to exploit redundancy in the features of pixels belonging to the same cluster. Here, we have considered only the colors of a pixel as the features, but the feature space can be completely user defined (e.g., can include optical flow, leading to a $5D$ vector, see discussion in Section 5). This section describes the method that we use to automatically cluster the pixels. It is observed that such clusters in-fact correspond to connected layers in the image domain.[2] For a review of color image layering techniques in the literature please refer to [2].

2.1. Initial Guess

We first compute the color C corresponding to a local maximum (h_{max}) of the histogram of the image. All pixels with colors lying inside a particular radius (ρ) of C form our *candidate layer* (L_C). The histogram maximum and radius are computed as in [3].

2.2. Refinement Step

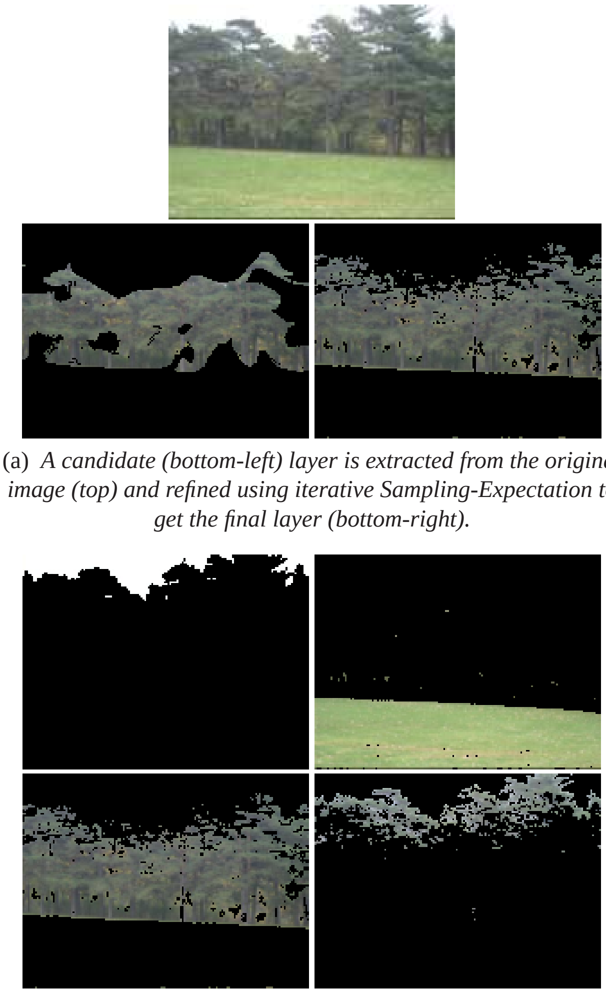
Once an initial guess for a layer is obtained, it is important to add or remove pixels from the layer depending on consistency of features, in order to improve the homogeneity and integrity of the layer. This is done by a refinement step that uses the Sampling-Expectation (SE) technique proposed in [18]. There are 3 main steps in this refining process:

- It is assumed that the pixels in the candidate layer and the rest of the image come from two separate processes. To initialize these processes, start with an initial distribution P_{Lc} on the image pixels, with pixels belonging to L_C having high values and pixels not in L_C getting low values (with a gradual spatial decay, for example a Gaussian distribution). This probability P_{Lc} indicates our confidence about the chance that the pixel belongs to L_C. P_{bg} forms the complementary background process.
- 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.
- E-step: Pixels are assigned-to or removed from L_C based on maximum-likelihood, i.e., if $P_{Lc} > P_{bg}$, the pixel is assigned to L_C, else is removed.

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 computed using a weighted Kernel Density Estimation. For details about the concept of Kernel Density Estimation the reader is referred to [11]. Given a pixel y belonging to the image, we estimate the probabilities $P_{Lc}(y)$ and $P_{bg}(y)$ by first computing the following parameters (as in [18]):

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

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



(a) *A candidate (bottom-left) layer is extracted from the original image (top) and refined using iterative Sampling-Expectation to get the final layer (bottom-right).*

(b) *Layers extracted from the original image in* (a).

Figure 2. *The automatic layering process.*

[2] Connectivity can be explicitly forced, e.g., by adding more dimensions to the feature vector that indicates (say) the median color of the neighborhood.

where, K is the kernel or smoothing function (we use a Gaussian kernel), d is the dimension of the feature-space (3 in our case, 5 if we incorporate optical flow) and h_j's are the kernel bandwidths, which we estimate using $h_j \approx 1.06\hat{\sigma_j} m^{\frac{1}{5}}$ [9], where $\hat{\sigma_j}$ is the standard deviation estimated over the sample S, in dimension j. The pixel probabilities are then computed as:

$$\begin{aligned} P_{Lc}(y) &= W_{Lc}/(W_{Lc}+W_{bg}) && (3)\\ P_{bg}(y) &= W_{bg}/(W_{Lc}+W_{bg}) && (4)\end{aligned}$$

The "initial guess" (previously extracted layers are excluded for computing h_{max}), and "refinement" steps are performed repetitively, generating layers in the image domain, until the initial guess L_C has fewer than 1% of pixels in the entire image. It is also possible that some pixels are classified as belonging to multiple layers (as the SE refinement is carried out over the entire image). These pixels along with the residual un-assigned pixels are assigned to one of the layers using maximum-likelihood. This process allows us to describe the scene $\mathbb{S}$ as $\mathbb{S} = \bigcup_{i=1}^{N} L_i$, where L_i are the extracted N layers. Note that both the layers and their number N are automatically computed.

Figure 2(a) shows the initial guess and the refined final layer. Observe that the refinement step ensures consistency in the layer and more accurately defines its boundaries. All layers extracted from the original frame are shown in Figure 2(b). 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.

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.

3. Layer Propagation

Once the initial training stack T is layered, the rest of the background modelling process is to assign all incoming pixels to one of the predetermined layers in the scene, or identify it as an outlier/foreground (assign to layer L_0).

3.1. Density Estimation

Similar to the work in [10], we believe that there exits 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 user-defined parameter gives us an idea of the size ($w \times w \times M$, where M is the number of frames in the stack T), of spatio-temporal-neighborhood of pixels in the training stack T, that may be correlated to the current pixel. All pixels ($\mathbf{x_i}$) from the stack T, assigned to layer L_i, which lie in the "w-vicinity" of the current examined pixel ($\mathbf{y}$), form the sample set $\mathbf{S}_i$. To compute the probability of the pixel $\mathbf{y}$ to belong to any layer L_i, we use a Non-Parametric Kernel Density Estimator with a Gaussian kernel:

$$\hat{f}_{Li}(\mathbf{y}) = \frac{1}{n_i}\sum_{k=0}^{n_i}\frac{1}{\|\mathbf{H}(L_i)\|^{1/2}}K(\mathbf{H}(L_i)^{-1/2}(\mathbf{y}-\mathbf{x_i})) \quad (5)$$

where n_i is the number of samples ($\mathbf{x_i}$'s) belonging to L_i. The bandwidth matrix $\mathbf{H}$ is assumed to be diagonal, $\mathbf{H}(L_i) = h(L_i)\mathbf{I}$, where the argument L_i is used to indicate all samples belonging to L_i, i.e., we use the same bandwidth for samples from one layer, when computing the density estimate. For the results shown in this paper, we have approximated the diagonal values ($h(L_i)$'s) with the standard deviation of all the training samples ($\mathbf{S}_i$), as is done in the layer refinement step (refer to Section 2.2). The layer of outliers (L_0) also contributes samples to the likelihood-computation, when there is a previously detected outlier in the w-vicinity. Thus, there is a competitive classification between outliers and background, at the same time, propagating the background layers throughout the video.

3.2. Threshold Computation And Outlier Detection

Depending upon the homogeneity and integrity of the pixels belonging to the 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 (τ_i's), we use the *a-contrario* framework [1, 14]. Using samples ($\mathbf{S}_i$) from T that belong to (say) layer L_i, we compute the layer probability ($P_{Li}(\mathbf{x}_i)$) of all pixels $\mathbf{x}_i$ in T which are already labeled as belonging to L_i. This allows us to compute the probability that a pixel $\mathbf{y}$ belongs to layer L_i, such that $P_{Li}(\mathbf{y})$ is less than a certain threshold (say) μ:

$$\mathbb{P}(L_i,\mu) := Pr\big(P_{L_i}(\mathbf{y}) < \mu \mid \mathbf{y} \in L_i\big) \quad (6)$$

This allows us to say that a pixel $\mathbf{z}$ is an ϵ-*meaningful outlier* from the layer L_i, if $\mathbb{P}(L_i, P_{Li}(\mathbf{z})) < \frac{\epsilon}{n_i}$, where n_i is the number of queried pixels in T belonging to L_i. The *a-contrario* model assumes that the such outliers are uniformly distributed, hence setting $\epsilon = 1$, like we do, allows us to ensure that the average number of false detections over the layer L_i is less than one (for more details refer to, e.g., [1, 14]). Thus all thresholds are computed as :

$$\tau_i = \min \mu \; s.t. \{\mathbb{P}(L_i,\mu) < \frac{1}{n_i}\} \quad (7)$$

Figure 3 illustrates the inverted winning maximum-likelihood probabilities (top-right) for all the pixels. The

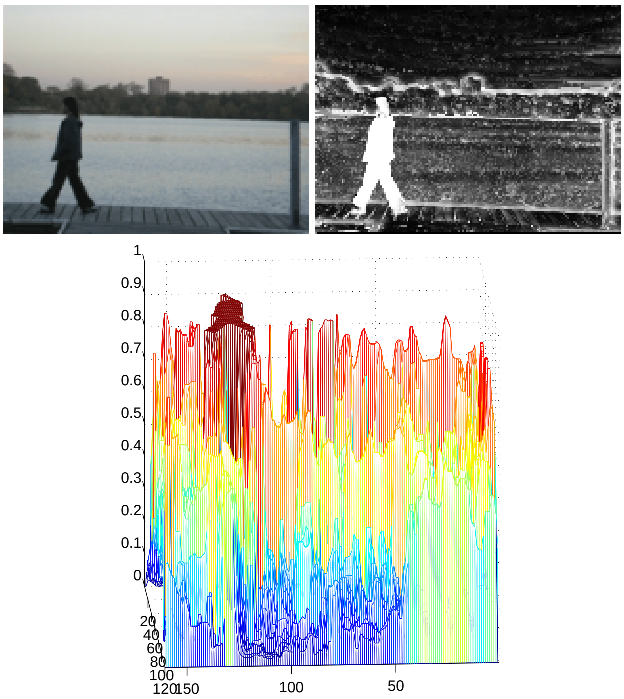


Figure 3. *Image on the top-right shows the inverted (i.e, subtracted from* 1*) maximum-likelihood probabilities, corresponding to the original frame (top-left). When these probabilities are plotted in a* 3D *plot (bottom), the moving person (dark-brown), is easily distinguishable.*

$3D$ perspective view shows that the moving person (indicated by dark-brown color in the bottom plot) can be easily detected. The boundaries of the layers look slightly brighter (have low ML probabilities) because of color-blending at the edges. Further grouping of individual outliers as indicated in [1] can lead to even more robust detections than presented here.

The bandwidths and thresholds can be left unchanged or updated every few frames. The color space that we use (same as in [7]) is robust to illumination changes, and thereby adaptation is not necessary if these are the only type of changes expected.

4. Results

The videos described in this section (and available at `www.tc.umn.edu/~patw0007/videolayers`) are outdoor scenes of resolution 160x120. The algorithm was implemented using C++, on a machine with Intel-Pentium IV 1.8GHz processor. We used a training stack of 30 frames for all the results, achieving a running speed of 1 frame/second (using a completely un-optimized experimental code). Figure 4 illustrates the performance of our algorithm in presence of moderately dynamic background along with a lot of camera shake (please observe uploaded original videos). The outliers detected are very robust to the dynamics of the background and the camera motion. Figure 5, shows a very challenging situation, produced by highly perturbed water in the background. Our technique performs very well to distinguish the moving person (looks inverted due to reflection) from the ripples. For the results in figures 4 and 5, we have used a w parameter value of 3. Hence the spatio-temporal training window for each pixel is of size $3 \times 3 \times 30$. Our algorithm does not need very precise registration. If the uncertainty in the registration computation is known, it can be figured into the w-parameter. As indicated in Figure 6, in spite of the significant camera panning we have not used any registration, which is adjusted for by using a w value of 11, indicating the increased position uncertainty. Figure 6 also indicates how well the layers in the scene are propagated through the video sequence in spite of severe camera panning.

5. Discussion And Future Scope

In this work we have proposed a general framework for scene modelling and foreground detection using pixel-clusters (layers). Redundancy in the feature-space and spatial correlation in the image-domain are exploited by clustering pixels into finite number of layers and modelling the scene as a union of these layers rather than individual pixels. The task at hand then is to assign any incoming pixel to one of these layers or as an outlier by using a maximum-likelihood assignment, which allows for competitive classification between the scene layers and the layer of outliers. Thresholds are chosen in a non-arbitrary fashion to give robust outlier detections. The results presented show very satisfactory performance in very difficult environments.

It should be noted that we have used only the color information of the pixels, and results can be further improved by adding additional information like optical flow, though it should be noted that in case of severe camera motion (most result videos shown here), using optical flow may in-fact misguide the background model and generate false alarms. In circumstances where positional uncertainty is large, using very high w-values is not efficient and also degrades accuracy. Approximate (rough) registration of frames within a certain error bound can be utilized to optimize the performance of our algorithm on videos with severe panning or camera motion. The run-time can be improved to reach real-time by optimization and using the *Improved Fast Gauss Transform* (IFGT), as shown in [17], for fast density computation. In some detection results a few noisy false alarms are observed which can be further removed by a “meaningful” grouping of outliers as proposed in [1]. In the future we would also like to address the appearance of novel layers (not seen previously), or appearance of foreground that remains static in the scene (e.g. a car arriving in a parking lot and parked for a long time). This can be achieved by simply considering a "temporally persistent" group of outliers as a completely new *depth-ordered* background layer. Further work in this direction will be reported elsewhere.

Acknowledgements

We would like to thank Diego Rother for interesting discussions about Density Estimation. This work is partially supported by grants from NSF, ONR, NGA and DARPA. The first author performed part of this work at Honeywell Technology Solutions Lab, Minneapolis, during the summer of 2005.

References

[1] F. Cao, Y. Gousseau, P. Muse, F. Sur, and J.-M. Morel. Accurate estimates of false alarm number in shape recognition. Technical report, Cachan, France (http://www.cmla.ens-cachan.fr/Cmla/), 2004. 4, 5

[2] H. Cheng, X. Jiang, Y. Sun, and J. Wang. Color image segmentation: advances and prospects. *Pattern Recognition*, 34(12):2259–2281, December 2001. 3

[3] D. Comaniciu and P. Meer. Robust analysis of feature spaces: color image segmentation. In *CVPR*, pages 750–, 1997. 3

[4] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis. Background and foreground modeling using nonparametric kernel density for visual surveillance. In *Proceedings of the IEEE*, volume 90, pages 1151–1163, July 2002. 1

[5] N. Friedman and S. J. Russell. Image segmentation in video sequences: A probabilistic approach. In *UAI*, pages 175–181, 1997. 2

[6] R. C. Jain and H. H. Nagel. On the analysis of accumulative difference pictures from image sequences of real world scenes. 1(2):206–213, 1979. 1

[7] A. Mittal and N. Paragios. Motion-based background subtraction using adaptive kernel density estimation. In *CVPR (2)*, pages 302–309, 2004. 2, 5

[8] V. Morellas, I. Pavlidis, and P. Tsiamyrtzis. Deter: detection of events for threat evaluation and recognition. *Mach. Vision Appl.*, 15(1):29–45, 2003. 2

[9] D. W. Scott. Multivariate Density Estimation. Wiley Inter-Science, 1992. 4

[10] Y. Sheikh and M. Shah. Bayesian modeling of dynamic scenes for object detection. *PAMI*, 27(11):1778–1792, November 2005. 2, 4

[11] B. W. Silverman. Density Estimation for Statistics and Data Analysis. Chapman and Hall, London, UK, 1986. 3

[12] C. Stauffer and W. E. L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000. 2

[13] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *ICCV*, pages 255–261, 1999. 2

[14] 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). 4

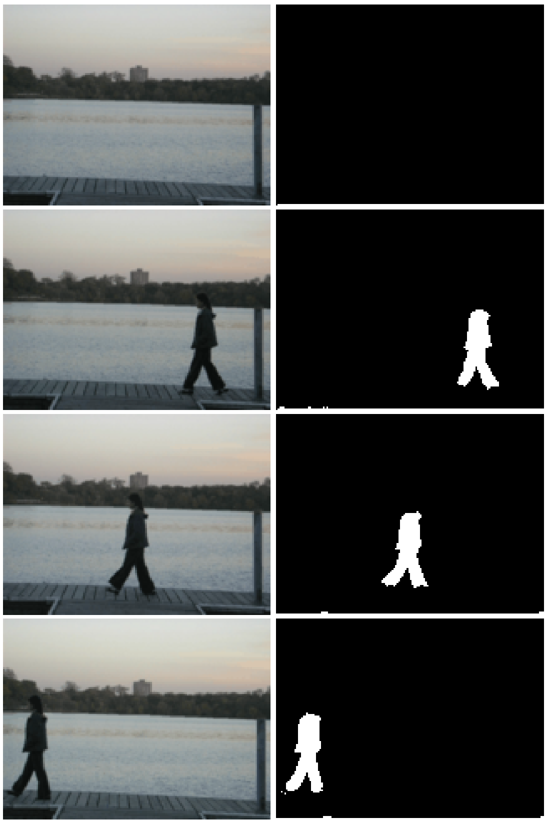


Figure 4. *The dynamic background contains motion of water with wind. Considerable camera shake can be observed (please see uploaded videos). Original frames (#* $0, 66, 82, 107$*) are shown on the left, while corresponding outliers detected are shown on the right.*

[15] T. Wada and T. Matsuyama. Appearence sphere: Background model for pan-tilt-zoom camera. In *ICPR '96: Proceedings of the 1996 International Conference on Pattern Recognition (ICPR '96) Volume I*, page 718, Washington, DC, USA, 1996. IEEE Computer Society. 1

[16] C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997. 2

[17] C. Yang, R. Duraiswami, N. Gumerov, and L. Davis. Improved fast gauss transform and efficient kernel density estimation. In *International Conference on Computer Vision, Nice, France, 2003.*, pages 464–471, 2003. 5

[18] L. Zhao and L. S. Davis. Iterative figure-ground discrimination. In *ICPR (1)*, pages 67–70, 2004. 2, 3

[19] J. Zhong and S. Sclaroff. Segmenting foreground objects from a dynamic textured background via a robust kalman filter. In *ICCV*, pages 44–50, 2003. 2

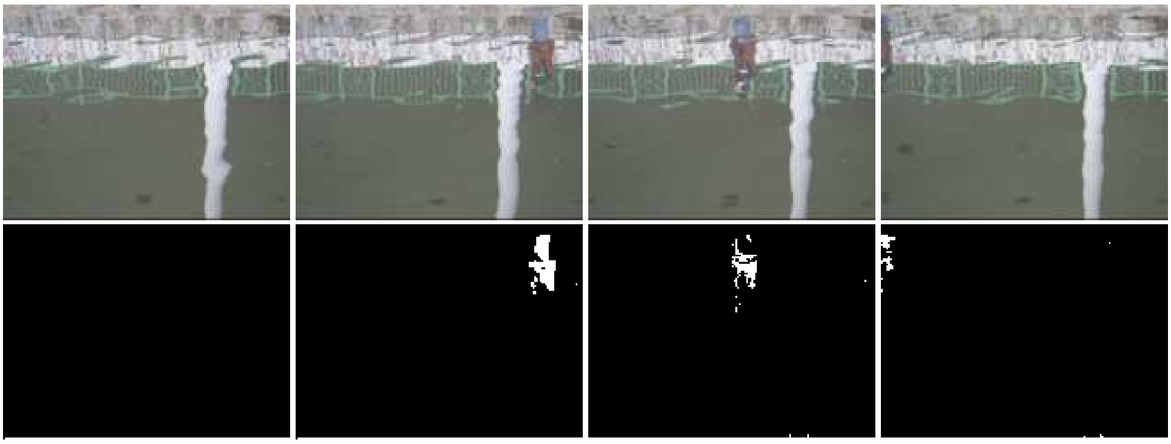


Figure 5. *Large amount of water ripples present a challenging situation for foreground, reflection, detection (see uploaded video). Original frames are shown on the top (# $0, 47, 62, 89$) along with outliers on the bottom.*

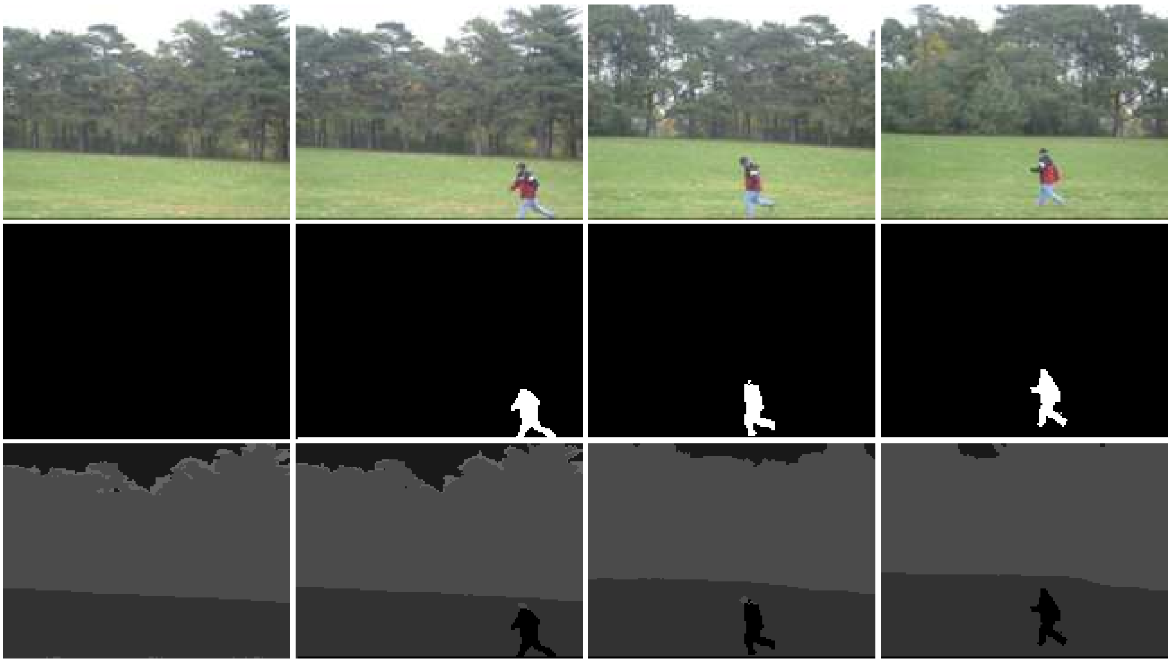


Figure 6. *Trees swaying with wind along with camera tilt and panning are very difficult scenarios for background modelling (see uploaded video). Outliers are shown in the center row, while the propagation of various layers (indicated by different scales of gray) is shown in the bottom row.*