Machine Vision and Applications (2014) 25:1163–1174
DOI 10.1007/s00138-013-0569-y

SPECIAL ISSUE PAPER

Background subtraction: separating the modeling and the inference

Manjunath Narayana · Allen Hanson · Erik G. Learned-Miller

Received: 28 January 2013 / Revised: 25 July 2013 / Accepted: 7 October 2013 / Published online: 25 October 2013

Abstract In its early implementations, background modeling was a process of building a model for the background of a video with a stationary camera, and identifying pixels that did not conform well to this model. The pixels that were not well-described by the background model were assumed to be moving objects. Many systems today maintain models for the foreground as well as the background, and these models compete to explain the pixels in a video. If the foreground model explains the pixels better, they are considered foreground. Otherwise they are considered background. In this paper, we argue that the logical endpoint of this evolution is to simply use Bayes' rule to classify pixels. In particular, it is essential to have a background likelihood, a foreground likelihood, and a prior at each pixel. A simple application of Bayes' rule then gives a posterior probability over the label. The only remaining question is the quality of the component models: the background likelihood, the foreground likelihood, and the prior. We describe a model for the likelihoods that is built by using not only the past observations at a given pixel location, but by also including observations in a spatial neighborhood around the location. This enables us to model the influence between neighboring pixels and is an improvement over earlier pixelwise models that do not allow for such influence. Although similar in spirit to the joint domain-range model, we show that our model overcomes certain deficiencies in that model. We use a spatially dependent prior for the background and foreground. The background and foreground labels from the previous frame, after spatial smoothing to account for movement of objects, are used to build the prior for the current frame. These components are, by themselves, not novel aspects in background modeling. As we will show, many existing systems account for these aspects in different ways. We argue that separating these components as suggested in this paper yields a very simple and effective model. Our intuitive description also isolates the model components from the classification or inference step. Improvements to each model component can be carried out without any changes to the inference or other components. The various components can hence be modeled effectively and their impact on the overall system understood more easily.

Keywords Background modeling · Motion segmentation · Surveillance

M. Narayana (✉) · E. G. Learned-Miller · A. Hanson
University of Massachusetts Amherst, Amherst, MA, USA
e-mail: narayana@cs.umass.edu

E. G. Learned-Miller
e-mail: elm@cs.umass.edu

A. Hanson
e-mail: hanson@cs.umass.edu

1 Introduction

Background subtraction for stationary camera videos is a well researched problem. Algorithms have evolved from early approaches modeling the background at each pixel [3,17,19,22] to methods that include an explicit model for the foreground [8,16], and finally to more recent models that incorporate spatial dependence between neighboring pixels [16].

In early algorithms [17,22], a probability distribution $p_{\mathbf{x}}(\mathbf{c}|bg)$ over *background* colors $\mathbf{c}$ is defined and learned for each location $\mathbf{x}$ in the image. These distributions are essentially the background likelihood at each pixel location. Pixels that are well explained by the background likelihood are classified as *background* and the remaining pixels in the image are labeled as *foreground*. Toyama et al. [19] use a Weiner filter to predict the intensities of the background pixels in the

current frame using the observed values from the previous frames and to identify non-conforming pixels as foreground. Wren et al. [22] model the background as a Gaussian distribution at each pixel. To account for the multiple intensities often displayed by background phenomena such as leaves waving in the wind or waves on water surfaces, Stauffer and Grimson [17] learn a parametric mixture of Gaussians (MoG) model at each pixel. The MoG model update procedure as described by Stauffer and Grimson can be unreliable during initialization when not enough data have been observed. To improve the performance during model initialization, Kaewtrakulpong and Bowden [6] suggest a slightly different model update procedure. Porikli and Tuzel [14] obtain the background likelihood by using a Bayesian approach to model the mean and variance values of the Gaussian mixtures. Elgammal et al. [3] avoid the drawbacks of using a parametric MoG model by instead building the background likelihoods with non-parametric kernel density estimation (KDE) using data samples from previous frames in history.

While they are still called "backgrounding" systems, later systems maintain a model for the foreground as well as the background [8,16]. Explicit modeling of the foreground has been shown to improve the accuracy of background subtraction [16]. In these models, pixel labeling is performed in a competitive manner by labeling as foreground the pixels that are better explained by the foreground model. The remaining pixels are labeled as background.

Although it is natural to think about priors along with likelihoods, the use of an explicit prior for the background and foreground is less common. In the object tracking literature, Aeschliman et al. [1] use priors for the background and foreground objects for segmentation of tracked objects. In background modeling algorithms that do not explicitly model the prior, the foreground–background likelihood ratio is used for classification. Pixels that have a likelihood ratio greater than some predefined threshold value are labeled as foreground. This method is equivalent to using an implicit prior that is the same at all pixel locations.

Thus, existing algorithms make use of some subset of the three natural components for background modeling—the background likelihood, the foreground likelihood, and the prior. They make up for the missing components by including effective model-specific procedures at the classification stage. For instance, Elgammal et al. [3] and Stauffer and Grimson [17] use only the background likelihood, but, during classification, consider a likelihood threshold below which pixels are considered as foreground. Zivkovic [24] describes Bayes' rule for computing background posteriors, but since neither the foreground likelihood nor the priors are explicitly modeled, the classification is essentially based on a threshold on background likelihood values. Sheikh and Shah [16] utilize both foreground and background likelihoods, but do not use an explicit prior. Instead, by using a foreground–background likelihood ratio as the classification criterion, they effectively use a uniform prior.

We argue that the logical endpoint of the model evolution for backgrounding is a system where all three components are explicitly modeled and Bayes' rule is applied for classification. Such a system has the advantage of being a simpler model where the modeling of the individual components is isolated from the inference step. This separation allows us to describe the components without any relation to the classification procedure. Our motivation behind this approach is that the components can individually be improved, as we will show in later sections, without affecting each other or the final inference procedure.

In the rest of the paper, we describe the components of our background system and place them in the context of existing algorithms where possible. Section 2 discusses the evolution of the background likelihood models and our improvements to the most successful models. In Sect. 3, we discuss the modifications to the likelihood for modeling the foreground. Modeling of the prior is described in Sect. 4. Computation of posterior probabilities by using the above components is explained in Sect. 5. Results comparing our system to earlier methods on a benchmark data set are given in Sect. 6. Recent improvements to the background likelihood and its impact on the system's accuracy are described in Sects. 7, 8. We conclude with a discussion in Sect. 9.

2 Background likelihood

The background likelihood, which is a distribution over feature values, is a common aspect in many backgrounding systems. Stauffer and Grimson [17] model the background likelihood at each pixel using a MoG approach. The requirement of specifying the number of mixture components in the MoG model is removed in the non-parametric kernel density estimation (KDE) model [3]. In the KDE model, the distributions at each pixel location are estimated by summing up contributions from the observed background data samples at that location from previous frames in history. For each pixel location $\mathbf{x} = [x, y]$, both these models maintain a distribution $p_{\mathbf{x}}(\mathbf{c})$ that is independent of the neighboring pixels. Here, $\mathbf{c} = [r, g, b]$ is a vector that represents color. These neighbor-independent distributions have the drawback of not being able to account for the influence of neighboring pixels on each other's color distributions.

To allow neighboring pixels to influence the background likelihood at a given pixel location, we model the likelihood at a particular pixel location to be a weighted sum of distributions from its spatial neighbors. Our *smoothed* background likelihood $P_{\mathbf{x}}(\mathbf{c})$ for each pixel location $\mathbf{x}$ is a weighted sum of distributions from a spatial neighborhood $\mathcal{N}_{\mathcal{B}}$ around $\mathbf{x}$. Each neighboring likelihood is weighted by its spatial dis-

tance (i.e., distance in the image coordinates) from $\mathbf{x}$:

$$P_{\mathbf{x}}(\mathbf{c}|\mathrm{bg};\, \Sigma_{\mathbf{S}}^{B}) = \frac{1}{Z} \sum_{\Delta \in \mathcal{N}_{\mathcal{B}}} p_{\mathbf{x}+\Delta}(\mathbf{c}|\mathrm{bg}) \times G(\Delta;\, \mathbf{0},\, \Sigma_{\mathbf{S}}^{B}). \tag{1}$$

Here Δ is a spatial displacement that defines a spatial neighborhood $\mathcal{N}_{\mathcal{B}}$ around the pixel location $\mathbf{x}$ at which the likelihood is being computed. $G(\cdot;\, \mathbf{0},\, \Sigma_{\mathbf{S}}^{B})$ is a zero-mean multivariate Gaussian with covariance $\Sigma_{\mathbf{S}}^{B}$. B indicates that the covariance is for the background model and $\mathbf{S}$ denotes the spatial dimensions. The normalization constant Z is

$$Z = \sum_{\Delta \in \mathcal{N}_{\mathcal{B}}} G(\Delta;\, \mathbf{0},\, \Sigma_{\mathbf{S}}^{B}). \tag{2}$$

The weighted sum results in a spatial smoothing of the distributions as shown in Fig. 1. This spreading of information is useful in modeling spatial uncertainty of background pixels. $\Sigma_{\mathbf{S}}^{B}$ controls the amount of smoothing and spreading of information in the spatial dimensions.

Explicitly maintaining a distribution at each pixel location is impractical for color features which can take one of 256^3 values if each of the three color channels have a range between 0 and 255. Instead, we compute likelihoods with KDE using the data samples from the previous frame. Let $\mathbf{b}_{\mathbf{x}}^{t-1}$ be the observed background color at pixel location $\mathbf{x}$ in the previous frame. Using a Gaussian kernel with covariance $\Sigma_{\mathbf{C}}^{B}$ in the color dimensions, our KDE background likelihood in the video frame numbered t is given by

$$P_{\mathbf{x}}^{t}(\mathbf{c}|\mathrm{bg};\, \Sigma_{\mathbf{C}}^{B},\, \Sigma_{\mathbf{S}}^{B}) = \frac{1}{Z} \sum_{\Delta \in \mathcal{N}_{\mathcal{B}}} G(\mathbf{c} - \mathbf{b}_{\mathbf{x}+\Delta}^{t-1};\, \mathbf{0},\, \Sigma_{\mathbf{C}}^{B}) \times G(\Delta;\, \mathbf{0},\, \Sigma_{\mathbf{S}}^{B}). \tag{3}$$

Figure 2 illustrates the process of computing the background likelihood using the observed background colors in one image. It may be noted that the covariance matrix $\Sigma_{\mathbf{S}}^{B}$ controls the amount of spatial influence from neighboring pixels. The covariance matrix $\Sigma_{\mathbf{C}}^{B}$ controls the amount of variation allowed in the color values of the background pixels.

Finally, we consider background data samples not just from the previous frame, but from the previous T frames in order to obtain a more accurate likelihood. We also allow probabilistic contribution from the previous frames' pixels by weighting each pixel according to its probability of belonging to the background:

$$P_{\mathbf{x}}^{t}(\mathbf{c}|\mathrm{bg};\, \Sigma^{B}) = \frac{1}{K_{\mathrm{bg}}} \sum_{i \in 1:T} \sum_{\Delta \in \mathcal{N}_{\mathcal{B}}} G(\mathbf{c} - \mathbf{b}_{\mathbf{x}+\Delta}^{t-i};\, \mathbf{0}, \Sigma_{\mathbf{C}}^{B}) \times G(\Delta;\, \mathbf{0}, \Sigma_{\mathbf{S}}^{B}) \times P_{\mathbf{x}}^{t-i}(\mathrm{bg}|\mathbf{b}_{\mathbf{x}+\Delta}^{t-i}). \tag{4}$$

Σ^{B} represents the covariance matrices for the background model and consists of the color dimensions covariance matrix $\Sigma_{\mathbf{C}}^{B}$ and the spatial dimensions covariance matrix $\Sigma_{\mathbf{S}}^{B}$. $P_{\mathbf{x}}^{t}(\mathrm{bg}|\mathbf{b}_{\mathbf{x}}^{t})$ is the probability that pixel at location $\mathbf{x}$ in the frame t is background. K_{bg} is the appropriate normalization factor:

$$K_{bg} = \sum_{i \in 1:T} \sum_{\Delta \in \mathcal{N}_{\mathcal{B}}} G(\Delta;\, \mathbf{0},\, \Sigma_{\mathbf{S}}^{B}) \times P_{\mathbf{x}}^{t-i}(\mathrm{bg}|\mathbf{b}_{\mathbf{x}+\Delta}^{t-i}). \tag{5}$$

For efficiency, we restrict the covariance matrices to be diagonal and hence parameterize them by their diagonal elements.

2.1 Existing work on spatial smoothing of distributions

The use of spatial smoothing of distributions is not entirely new. Sheikh and Shah [16] use a joint domain-range model

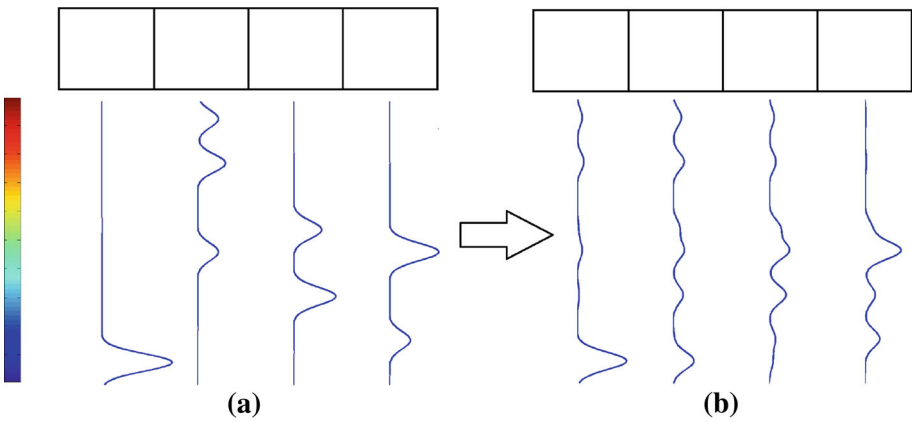

Fig. 1 Influence of neighboring pixels on each other is modeled by spreading information spatially. **a** Some example likelihoods for each pixel in a single-dimensional (row) image. The distributions shown below each pixel are the estimated background likelihoods. The vertical axis corresponds to color values which are visualized in the color map on the left side of the image. The horizontal axis corresponds to the probability of the corresponding color. **b** The smoothed likelihood at each pixel, which is a weighted sum of the likelihoods in the pixel's neighborhood. The effect of smoothing is clearly visible in the first pixel. The distribution in the first pixel clearly influences the distributions at the second and third pixels. The distance-dependent nature of the weights results in the first pixel influencing the third pixel less than it does the second pixel

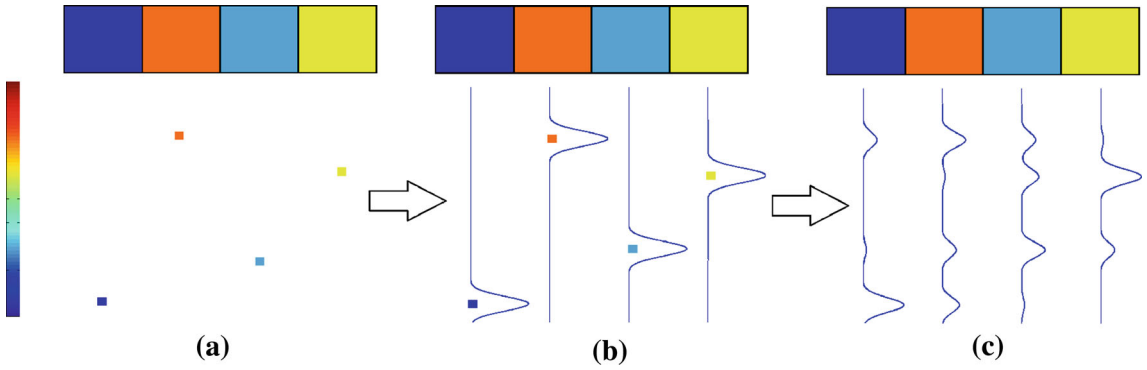

Fig. 2 Modeling the likelihoods using pixel data samples and KDE. **a** The colors at each pixel. The corresponding color and its location with respect to the vertical color axis is shown under each pixel. **b** The likelihood at each pixel estimated using KDE with a Gaussian kernel. **c** The effect of spatial smoothing of the KDE-based likelihoods. Again, the illustration uses a one-dimensional row image in which a pixel's color is also represented in one dimension. It is straightforward to extend the example to two-dimensional spatial coordinates and three-dimensional color space

that combines the pixels' position values and color observations into a joint five-dimensional space. By modeling the likelihoods in the joint space, they allow pixels in one location to influence the distribution in another location. Their background likelihood is defined as:[1]

$$P^t(\mathbf{c}, \mathbf{x}|\mathrm{bg}; \Sigma^B) = \frac{1}{K} \sum_{i \in 1:T} \sum_{\Delta \in \mathcal{N}_\mathcal{B}} G(\mathbf{c} - \mathbf{b}^{t-i}_{\mathbf{x}+\Delta}; \mathbf{0}, \Sigma^B_\mathbf{C}) \times G(\Delta; \mathbf{0}, \Sigma^B_\mathbf{S}) \times P^{t-i}_\mathbf{x}(\mathrm{bg}|\mathbf{b}^{t-i}_{\mathbf{x}+\Delta}). \quad (6)$$

The normalization constant, K, is given by

$$K = \sum_{i \in 1:T} \sum_{\Delta \in \mathcal{N}_\mathcal{B}} P^{t-i}_\mathbf{x}(\mathrm{bg}|\mathbf{b}^{t-i}_{\mathbf{x}+\Delta}). \quad (7)$$

The key difference between their model and ours is that theirs is, for the entire image, a *single* distribution in the joint domain-range space whereas ours consists of a different location-dependent distribution at each pixel. This difference has a big effect on the classification stage. As we will see later, their classification criterion, based on the ratio of foreground and background likelihoods in this five-dimensional space, has an undesirable dependence on the size of the image. By replacing the single joint distribution with a *field* of distributions dependent on image location, we avoid the dependence on image size and achieve better results.

The joint domain-range model has been used earlier in the object tracking literature. Elgammal et al. [2] use a joint domain-range model that is almost identical to the background model of Sheikh and Shah [16]. A scheme very similar to our Eq. 1 was used in a tracking system by Han and Davis [5] to interpolate the pixelwise appearance distributions for an object whose size has changed during the tracking process. The close resemblance between these models suggests that tracking and background modeling share similar fundamental principles and can be achieved under the same framework. One such framework that integrates segmentation and tracking has been described by Aeschliman et al. [1].

Ko et al. [7] use a histogram-based variant of the Sheikh and Shah [16] background model which is built from observations in a spatial neighborhood around each pixel from previous frames in history. However, they do not consider the spatial distance between a pixel and its neighbor when summing up the contributions. In addition, they build another distribution, which can be interpreted as the "texture" at each pixel, by using only the current frame observations in each pixel's spatial neighborhood. Their classification criterion for foreground pixels is to threshold the Bhattacharya distance between the background distribution and the "texture" distribution. Our model is different because of our classification criterion that uses foreground likelihoods and explicit priors for the background and foreground which we discuss in subsequent sections.

3 Foreground likelihood

Explicit modeling of the foreground likelihood has been shown to result in more accurate systems [8,16]. Our foreground likelihood is very similar to our background likelihood. However, it is important to consider in the foreground likelihood, the possibility of hitherto unseen color values appearing as foreground. This may happen because a new foreground object enters the scene or an existing foreground object either changes color or, by moving, exposes a previ-

[1] We have modified their equation to allow probabilistic contributions from the pixels and changed the notation to make it easily comparable to ours.

ously unseen part of it. We find it useful to separate the foreground process into two different sub-processes: previously seen foreground, which we shall refer to as *seen* foreground, and previously unseen foreground, which we shall refer to as *unseen* foreground. The likelihood for the seen foreground process is computed using a KDE procedure similar to the background likelihood estimation:

$$P^t_{\mathbf{x}}(\mathbf{c}|\mathtt{fg};\,\Sigma^F) = \frac{1}{K_{\mathtt{fg}}} \sum_{i\in 1:T} \sum_{\Delta\in\mathcal{N}_{\mathcal{F}}} G(\mathbf{c}-\mathbf{f}^{t-i}_{\mathbf{x}+\Delta};\,\mathbf{0},\,\Sigma^F_{\mathbf{C}}) \times G(\Delta;\,\mathbf{0},\,\Sigma^F_{\mathbf{S}}) \times P^{t-i}_{\mathbf{x}}(\mathtt{fg}|\mathbf{f}^{t-i}_{\mathbf{x}+\Delta}). \quad (8)$$

Similar to Eq. 4, $\mathbf{f}^t_{\mathbf{x}}$ is the observed foreground color at pixel location $\mathbf{x}$ in frame t. Σ^F is the covariance matrix for the foreground model, and $K_{\mathtt{fg}}$ is the normalization factor, analogous to $K_{\mathtt{bg}}$. $P^t_{\mathbf{x}}(\mathtt{fg}|\mathbf{f}^t_{\mathbf{x}})$ is the probability that pixel at location $\mathbf{x}$ in the frame t is foreground.

Since foreground objects typically move more than background objects and also exhibit more variation in their color appearance, we typically use higher covariance values for the foreground than for the background.

The likelihood for the unseen foreground process is simply a uniform distribution over the color space.

$$P^t_{\mathbf{x}}(\mathbf{c}|\mathtt{fu}) = \frac{1}{R\times G\times B} \quad (9)$$

for all locations $\mathbf{x}$ in the image, where R, G, and B, are the number of possible intensity values for red, green, and blue colors respectively.

The unseen foreground process constantly tries to account as foreground any colors not reasonably explained by both the background and the seen foreground likelihood.

The concept of using a uniform likelihood is not new. For instance, Sheikh and Shah [16] mix a uniform distribution (in five-dimensional space) to their foreground likelihoods to explain the appearance of new foreground colors in the scene. Separation of the foreground process into two sub-processes, as we have done, is equivalent to the mixing of the likelihoods into one combined likelihood. The advantage of considering them as separate sub-processes is that when combined with a separate prior for each, greater modeling flexibility can be achieved. For instance, at image boundaries where new objects tend to enter the scene, a higher prior can be used for the unseen foreground process.

4 Priors

In addition to modeling the likelihoods, we explicitly model spatially varying priors for the background and foreground processes. Such spatial priors have recently been used for segmentation of objects being followed in a tracking algorithm [1]. Background modeling systems that use a likelihood ratio as the classification criterion are implicitly assuming a uniform prior for the entire image. In such systems, if the foreground–background likelihood ratio at a given pixel is greater than some predefined threshold L, then the pixel is labeled as foreground. Using a value of 1 for L means that the background and foreground processes have a uniform and equal prior value at every pixel location. Other values of L imply using a uniform but unequal prior for the background and foreground.

We generalize the notion of the prior by considering a spatially varying prior. The uniform prior is simply a special case of our model. We define pixelwise priors for the three processes involved— background, previously seen foreground, and unseen foreground. The classified pixel labels from the previous frame are used as a starting point for building the priors for the current frame. We assume that a pixel that is classified as background in the previous frame has a 95 % probability of belonging to the background in the current frame as well. The pixel has a 2.5 % probability of belonging to a seen foreground object, and a 2.5 % probability of coming from a previously unseen foreground object. For a foreground pixel in the previous frame, we assume that due to object motion, there is a 50 % probability of this pixel becoming background, a 25 % probability of this pixel belonging to the same foreground object as in the previous frame, and a 25 % probability that it becomes a new unseen object. Experimental validation and the justification for the choice of these values is provided later.

There are hence essentially two settings for the prior at each pixel depending on whether the pixel was labeled background or foreground in the previous frame. Instead of using the hard thresholds described above, we use the pixel's background label probability from the previous frame when computing the prior. For instance, a pixel that has probability p of being background in the previous frame will have a background prior equal to $p \times 0.95 + (1-p)\times .5$. Also, since objects typically move by a few pixels from the previous frame to the current frame, we apply a smoothing (7×7 Gaussian filter with a standard deviation value of 1.75) to the classification results from the previous frame before computing the priors for the current frame. Let $\tilde{P}^{t-1}_{\mathbf{x}}(\mathtt{bg})$ be the smoothed background posterior image from the previous frame. The priors for the current frame are

$$\begin{aligned} P^t_{\mathbf{x}}(\mathtt{bg}) &= \tilde{P}^{t-1}_{\mathbf{x}}(\mathtt{bg})\times 0.950 + (1-\tilde{P}^{t-1}_{\mathbf{x}}(\mathtt{bg})\times 0.500,\\ P^t_{\mathbf{x}}(\mathtt{fg}) &= \tilde{P}^{t-1}_{\mathbf{x}}(\mathtt{bg})\times 0.025 + (1-\tilde{P}^{t-1}_{\mathbf{x}}(\mathtt{bg})\times 0.250,\\ P^t_{\mathbf{x}}(\mathtt{fu}) &= \tilde{P}^{t-1}_{\mathbf{x}}(\mathtt{bg})\times 0.025 + (1-\tilde{P}^{t-1}_{\mathbf{x}}(\mathtt{bg})\times 0.250. \end{aligned} \quad (10)$$

Figure 3 is an illustration of the prior computation process. Figure 3**a** shows the previous frame for which the background label probabilities at each pixel have been computed in **b**. The background probabilities are smoothed with a Gaussian filter

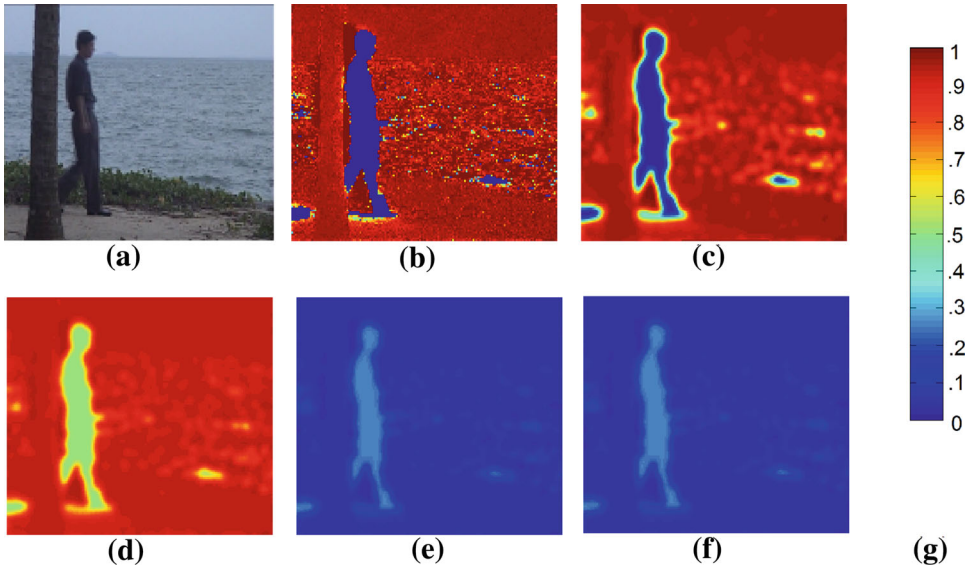

Fig. 3 Illustration of computation of the spatially dependent prior. The image from the previous frame is shown in **a**. The background probabilities in **b** are first smoothed with a Gaussian filter to allow for some amount of object motion in the scene. The smoothed probabilities are shown in **c**, from which the background prior (**d**), the foreground prior (**e**), and the unseen foreground prior (**f**) are computed. The mapping from color to probability values is given in **g**. We use equivalent equations for the foreground and unseen foreground priors which result in **e** and **f** being identical

in **c**. Using Eq. 10, the background prior **d**, the foreground prior **e**, the unseen foreground prior **f** are computed. These priors are then used for computing the posterior probabilities in the current frame, as we explain in the next section.

In our implementation, although the likelihoods for the foreground and unseen foreground processes are different, the priors for the two processes are equal at every pixel. It is not necessary that the priors for the seen foreground and the unseen foreground be the same in all background modeling systems. For instance, at image boundaries, using a higher prior value for the unseen foreground could result in better detection of new objects that enter the scene in these regions.

Our choice of the values 0.95 and 0.50 for the background prior for pixels that have been labeled as background and foreground in the previous frame respectively is guided by the intuition that background pixels change their label from one frame to the next very rarely and foreground objects that are moving have a moderate chance of revealing the background in the next frame. That these values are set by hand is a weakness of our current system.[2] The advantage of our approach is that these values can easily be learned automatically by accumulating statistics from the scene over a long period of time. Although the effect of using different priors for the background and foreground is equivalent to using a decision threshold on the foreground–background likelihood ratio, the priors are easier to understand and update. For example, the priors at each pixel can be updated using the statistics of pixel labels from long term scene history. The statistics could reveal a higher foreground prior near doors in the scene and at image borders. A similar scheme to update a decision threshold at these locations is far less natural.

We use a Gaussian filter of size 7 because the foreground objects in these videos typically move by 5–10 pixels. The size of the filter can potentially be learned by tracking the foreground objects. If there is a significant depth variation in different parts of the scene, a different parameter can be learned for the corresponding image regions by using tracking information [11].

5 Computing the posteriors: putting the components together during inference

Given the likelihoods and the priors as described in the previous sections, the only thing left to do is to compute the posterior probability of background and foreground, conditioned on the observed pixel values using Bayes' rule.

Given an observed color vector $\mathbf{c}$ at pixel location $\mathbf{x}$ in frame t, the probability of background and foreground are

$$P^t_{\mathbf{x}}(\mathtt{bg}|\mathbf{c}) = \frac{P^t_{\mathbf{x}}(\mathbf{c}|\mathtt{bg};\,\Sigma^B) \times P^t_{\mathbf{x}}(\mathtt{bg})}{\sum_{l=\mathtt{bg},\,\mathtt{fg}} P^t_{\mathbf{x}}(\mathbf{c}|l;\,\Sigma^l) \times P^t_{\mathbf{x}}(l) + P^t_{\mathbf{x}}(\mathbf{c}|\mathtt{fu}) \times P^t_{\mathbf{x}}(\mathtt{fu})},$$
$$P^t_{\mathbf{x}}(\mathtt{fg}) = 1 - P^t_{\mathbf{x}}(\mathtt{bg}|\mathbf{c}). \quad (11)$$

When the ideal likelihoods and priors are known, classification based on Bayes' rule gives the minimum possible error. A common alternative classification criterion is the ratio of the foreground likelihood to the background likelihood. The likelihood ratio classification in the joint domain-range model deserves special consideration because it implicitly includes

[2] Observations from the ground truth labels from videos in the change detection data set [4] show that between 95 and 100 % of all pixels labeled as background in each frame retain their background label in the next frame. We believe the use of the value 0.95 for background prior is justified in light of this observation. The use of 0.50 for the background prior in pixel locations that were labeled as foreground in the previous frame essentially allows the likelihood to decide the labels of these pixels in the current frame.

a notion of a prior. However, as we show in the next section, the implicit prior involved causes a peculiar dependence on the image size. Our model does not exhibit this undesired consequence.

5.1 Likelihood ratio-based classification in the joint domain-range model

In the Sheikh and Shah joint domain-range model [16], the classification of pixels is done based on the likelihood ratios of the background and foreground processes. The decision criterion based on the ratios of the five-dimensional background and foreground likelihoods can be represented as

$$P^t(\mathbf{c}, \mathbf{x}|\mathtt{bg}) \gtrless^{?} P^t(\mathbf{c}, \mathbf{x}|\mathtt{fg})$$
$$P^t(\mathbf{c}|\mathbf{x}, \mathtt{bg}) \times P^t(\mathbf{x}|\mathtt{bg}) \gtrless^{?} P^t(\mathbf{c}|\mathbf{x}, \mathtt{fg}) \times P^t(\mathbf{x}|\mathtt{fg}). \quad (12)$$

The classification decision hence depends on the factors $P^t(\mathbf{x}|\mathtt{bg})$ and $P^t(\mathbf{x}|\mathtt{fg})$. These factors are the prior probability of a particular pixel location given the background or foreground process. For any pixel location $\mathbf{x}$, these factors can depend upon parts of the image that are arbitrarily far away. This is because the prior likelihood of a given pixel location being foreground will be smaller if more pixels from another part of the image are detected as foreground, and larger if fewer pixels elsewhere are detected as foreground (since $P^t(\mathbf{x}|\mathtt{fg})$ must integrate to 1). Furthermore, these factors will change when the image size is changed, hence affecting the classification [13]. By separating the system components and bringing them together during the posterior computation, we avoid this arbitrary dependence on the size of the image.

6 Comparison to earlier systems

In this section, we compare our system to the various earlier systems described in the paper so far. We use the I2R benchmark data set [8] with nine videos taken in different settings. The videos have several challenging features like moving leaves and waves, strong object shadows, and moving objects becoming stationary for a long duration. The videos are between 500 and 3,000 frames in length and typically 128 $\times$ 160 pixels in size. Each video has 20 frames for which the ground truth has been marked. We use the F-measure to judge accuracy [9]; the higher the F-measure, the better the system:

$$F = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}}. \quad (13)$$

We use a Markov random field to post-process the labels as is done by Sheikh and Shah. Further, to be consistent with the experimental set up of earlier systems [9,12], we discard any foreground detections smaller than 15 pixels in size. The various systems compared are the MoG model of Stauffer and Grimson [17], the KDE model of Elgammal et al. [3], the complex background–foreground model of Li et al. (ACMMM03) [8], the joint domain-range model of Sheikh and Shah (jKDE) [16],[3] and our model, which we call the distribution field background (DFB) model. The naming reflects the fact that our model is a *field* of distributions with one distribution at each pixel location and was inspired by the description of such models in the tracking literature by Sevilla-Lara and Learned-Miller [15].

Results in Table 1 show that systems that model the spatial influence of pixels, namely the jKDE model and our DFB model yield significantly higher accuracy. The table shows that the jKDE system is most accurate for our chosen parameter setting. Although this is not true for other parameter settings,[4] the table makes an important point that very effective systems can be built even if the underlying model has certain deficiencies (as we showed in Sect. 5.1 for the jKDE). Mere separation of the model components as we have done and computing posterior probabilities for the labels does not guarantee better results. The usefulness of our system description is in the clear understanding of the different components and allowing for better modeling of the components without having to tweak the inference procedure. To illustrate this aspect of our system, we next describe one specific example of improving the background likelihood model by identifying a shortcoming in the model and developing a strategy to fix it.

7 Adaptive kernel variances for the background likelihood

In this section we discuss recent improvements to our KDE likelihood model. Although KDE is a non-parametric approach to estimate probability densities, the choice of the kernel variance or the bandwidth is an important one. Using large bandwidth values can result in a very smooth density function while low bandwidth values result in insufficient smoothing of the density function.

In the context of background modeling, different parts of a dynamic scene may exhibit different statistics over the feature values and hence may need to be explained by different kernel variance values. Consider the result from a slightly different KDE model [12] shown in Fig. 4. The figure shows background classification results when the background likelihoods were computed with increasing values of

[3] The KDE and jKDE models are our own implementations and include spatially-dependent priors and Bayes' classification criterion in order to make a fair comparison.

[4] For a detailed comparison of our model and the joint domain-range model, the reader is referred to our earlier paper [13].

Table 1 *F*-measure comparison between various existing algorithms on I2R data

Video	MoG	KDE	ACMMM03	jKDE	DFB
Airport hall	57.86	62.46	50.18	70.13	67.95
Bootstrap	54.07	61.15	60.46	71.77	69.17
Curtain	50.53	61.83	56.08	87.34	85.66
Escalator	36.64	40.84	32.95	53.70	54.01
Fountain	77.85	52.76	56.49	57.35	77.11
Shopping mall	66.95	63.05	67.84	74.12	70.95
Lobby	68.42	22.78	20.35	27.88	21.64
Trees	55.37	64.01	75.40	85.80	82.61
Water surface	63.52	51.16	63.66	78.16	75.80
Average	59.02	53.34	53.71	67.36	67.21

Modeling the spatial influence of pixels (jKDE and DFB) significantly improves accuracy. MoG and ACMMM03 results are as reported by Li et al. [9]. For KDE, jKDE, and DFB, we use color dimension covariance value of 45/4 for both the background and foreground models. For jKDE and DFB, we use spatial dimension covariance values of 3/4 and 12/4 for the background and foreground models respectively

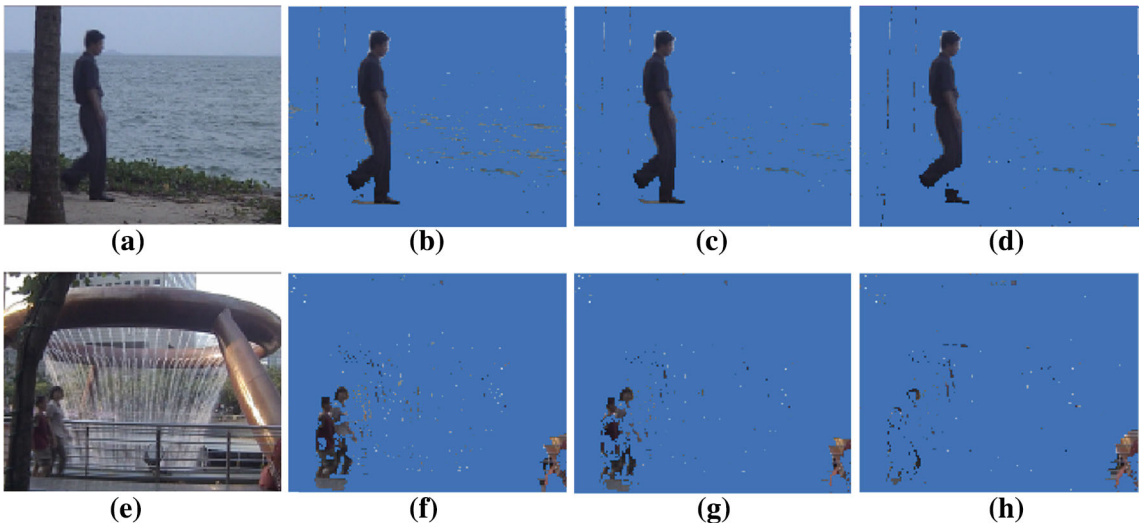

Fig. 4 Two video sequences classified using increasing values of spatial kernel variance. Column 1, original image; column 2, low variance; column 3, medium variance; column 4 high variance

spatial dimension variance for two different videos. Recall from Sect. 2 that the spatial variance controls the amount of influence that neighboring pixels have on a given pixel's background likelihood. Figure 4a–d show that having a high spatial dimension kernel variance helps in accurate classification of the water surface pixels, but doing so causes some pixels on the person's leg to become part of the background. Ideally, we would have different kernel variances for the water surface pixels and the rest of the pixels. Similarly in the second video (Fig. 4e–h), having a high kernel variance causes incorrect classification of many foreground pixels.

Kernel variance selection for KDE is a well studied problem [20], which can be addressed with variable-sized kernels [21]. The kernel size or variance can be adapted at the estimation point (*balloon estimator*) or at each data sample point (*sample-point estimator*). Zivkovic and Heijden [25] use a balloon estimator to adapt the kernel variance. Mittal and Paragios [10] use a hybrid approach but require that the uncertainty in the features be known.

Using a different parameter for each pixel location can be useful in accounting for the varied nature of the background phenomenon at each pixel. For the MoG model, Zivkovic [24] describes a method to find the optimal number of Gaussians to use at each pixel. For KDE models, Tavakkoli et al. [18] learn the variance for each pixel from a training set of frames, but do not adapt the learned values during the classification stage.

To address this problem, in earlier work [12,13], we proposed a location-specific variance and an adaptive method to select the best variance at each location. For each pixel

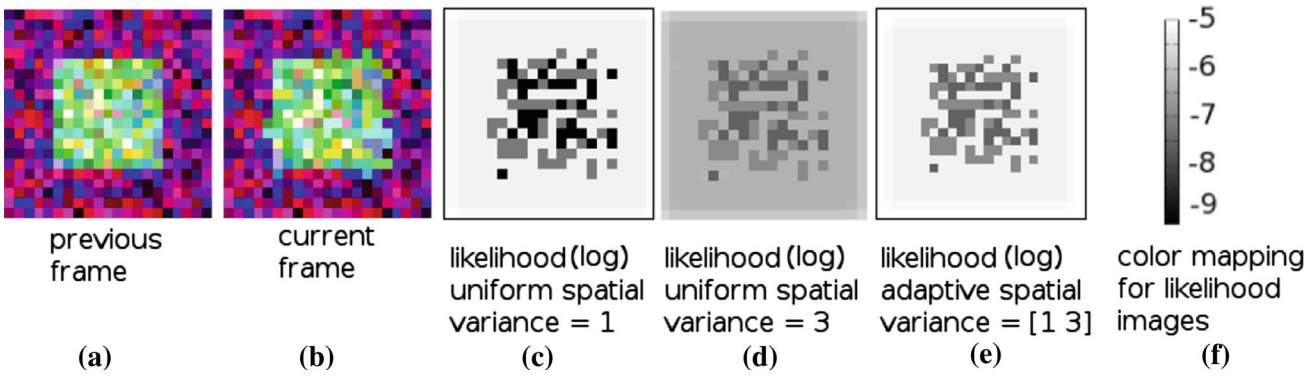

Fig. 5 **a, b** Spatial uncertainty in the central part of the background. **c** Small uniform variance results in low likelihoods for pixels that have moved. **d** Large uniform variance results in higher likelihoods of the moved pixels at the expense of lowering the likelihoods of stationary pixels. **e** Adaptive variance results in high likelihoods for both the moved and stationary pixels

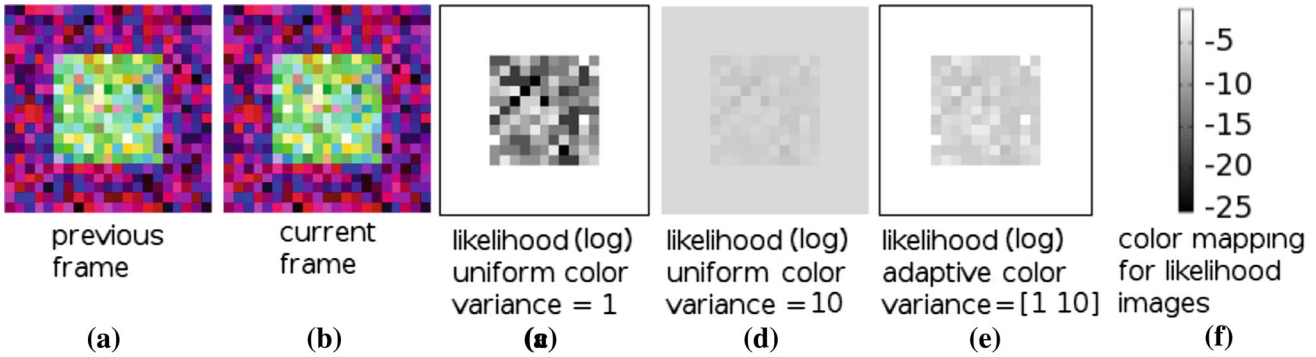

Fig. 6 Color uncertainty in the central part of the background is best modeled by using adaptive kernel variances. **c** Small uniform variance results in low likelihoods for pixels that have changed color. **d** Large uniform variance results in higher likelihoods of the altered pixels at the expense of lowering the likelihoods of other pixels. **e** Adaptive variance results in high likelihoods for both kinds of pixels

location, for the background model, a set of variance values for both spatial and color dimensions is tried and the configuration that results in the highest likelihood is chosen for that particular pixel.

The effect of the adaptive kernel variance method can be interpreted easily in Figs. 5, 6 (figures are originally from [13]). Consider a synthetic scene with no foreground objects, but in which the colors in the central greenish part of the background have been displaced at random by one or two pixel locations to simulate spatial uncertainty. As shown in Fig. 5, the adaptive kernel variance method models the scene better by applying a high spatial variance for pixels that have moved and a low spatial variance for pixels that have not moved. Similarly, for color variance, Fig. 6 shows the resulting likelihoods when uniformly sampled noise is added to the color values in the central part of the image. A small color variance value results in low likelihoods for pixels whose colors have changed. A large color variance results in low likelihoods for pixels that have not changed. The adaptive kernel variance method performs well in both kinds of pixels.

This improved background likelihood can be plugged into our system without any changes to the rest of the system. The following section discusses the increased accuracy that results from the substitution.

8 Comparison

Table 2 shows the results after using the adaptive kernel variance likelihood for the background. We compare our system to a very successful background system that uses recently developed complex texture features called scale invariant local ternary patterns (SILTP) [9] in a MoG model. These features are specifically designed to be robust to lighting changes and soft shadows in the scene and represent the state of the art accuracy on this benchmark. Results from the joint domain-range model with the use of the adaptive variance likelihood (abbreviated as jKDE-A) show a decrease in accuracy compared to the earlier likelihood (jKDE). Using the adaptive procedure in our system (DFB-A) results in a remarkable increase in accuracy. Using simple color features, our system is able to achieve accuracy comparable to SILTP on many videos.

Using a combination of color and texture features has been shown to be useful for background modeling [12,23]. Texture

Table 2 *F*-measure on I2R data

Video features	SILTP [9] siltp	jKDE rgb	jKDE-A rgb	DFB rgb	DFB-A rgb	DFB-A lab+siltp
Airport Hall	68.02	70.13	65.52	67.95	68.28	**70.75**
Bootstrap	72.90	71.77	71.38	69.17	71.86	**77.64**
Curtain	92.40	87.34	79.76	85.66	93.57	**94.07**
Escalator	**68.66**	53.70	54.02	54.01	66.37	49.99
Fountain	85.04	57.35	49.89	77.11	77.43	**85.88**
Shopping Mall	79.65	74.12	74.43	70.95	76.46	**82.64**
Lobby	**79.21**	27.88	33.34	21.64	13.24	62.60
Trees	67.83	85.80	85.57	82.61	83.88	**87.64**
Water surface	83.15	78.16	64.03	75.80	**93.81**	93.79

The highest accuracy for each video is marked in bold letters. Using the adaptive kernel variance method with LAB color features and SILTP texture features results in the highest accuracy. Compared to uniform kernel variance DFB model, the adaptive variance method DFB-A is more accurate

features are robust to lighting changes but not effective on large texture-less objects. Color features are effective on large objects, but not very robust to varying illumination. Including the SILTP feature representation along with LAB color features, which are more robust to lighting changes, and performing background modeling in this hybrid color-texture space returns the best results on a majority of videos. The parameters used for the adaptive kernel variance method and explanations of the improvement in the results are detailed in our earlier work [13].

Our results are poor on two videos in the set—escalator and lobby. The escalator video is from an indoor mall scene with a fast moving escalator. The escalator pixels exhibit a large amount of motion causing them to be incorrectly classified as foreground in many frames. The lobby video is from an office scene where a light switch is turned on and off at various times during the video. Our likelihood model fails during the light switching and our use of an explicit foreground model causes the background model to take a very long time to recover. Use of LAB color features and SILTP features helps in the drastic illumination change scenario of the lobby video.

8.1 Processing times

Our unoptimized Matlab code for distribution field background modeling with adaptive variance for each pixel (DFB-A) takes 10 s per frame for videos of size 128 × 160 pixels. In comparison, our implementation of the Sheikh and Shah model and our DFB model without the adaptive variance selection takes 5 s per frame. In earlier work [12], we describe a scheme to reduce computation time with the adaptive kernel method by recording the best variance values for each pixel from the previous frame. These cached variance values are first used to classify pixels in the current frame. The expensive variance adaptation is performed only for pixels where a confident classification is not achieved using the cached variance values. The caching method reduces the processing time to about 6 s per frame.

9 Discussion

We argue that the view of background modeling described in this paper is, from a probabilistic perspective, clean and complete for the purpose of background modeling. By separating the various aspects of a background modeling system, namely the background likelihood, the foreground likelihood, and a prior, into distinct components, we have presented a simple view of background modeling. For inference, these separate components are brought together to compute the posterior probability for background.

Previous backgrounding systems have also modeled the components that we have described, but have often combined them with each other or caused dependence between the components and the inference. The separation of the components from each other and their isolation from the inference step makes the system easy to understand and extend. The individual components can be improved without having to consider their interdependence and effect on the inference. We have shown one example of improving the background likelihood model and its positive impact on the system's accuracy.

We use a spatially varying prior that depends on the labels from the previous frame. The model can further be improved by using a different prior at the image boundaries where new foreground objects are more likely. The modeling of the prior can also be improved by the explicit use of object tracking information.

We also believe that isolation of the model components can help in the development of effective learning methods

for each of them. For example, the prior can be learned simply by counting the number of times each pixel is labeled as background or foreground. Maintaining a record of the number of times a pixel changes its label from background to foreground and vice-versa is one possible scheme to learn the prior values described in Sect. 4. Such a learning scheme can help build a dynamic model for the priors at different regions in the image.

Acknowledgments This work was supported in part by the National Science Foundation under CAREER award IIS-0546666 and grant CNS-0619337. Any opinions, findings, conclusions, or recommendations expressed here are the authors' and do not necessarily reflect those of the sponsors.

References

1. Aeschliman, C., Park, J., Kak, A.: A probabilistic framework for joint segmentation and tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1371–1378 (2010)
2. Elgammal, A., Duraiswami, R., Davis, L.S.: Probabilistic tracking in joint feature-spatial spaces. In: IEEE Conference on, CVPR'03 Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, Washington, DC, USA , pp. 781–788. (2003). http://dl.acm.org/citation.cfm?id=1965841.1965943
3. Elgammal, A.M., Harwood, D., Davis, L.S.: Non-parametric model for background subtraction. In: European Conference on Computer Vision, pp. 751–767 (2000)
4. Goyette, N., Jodoin, P.M., Porikli, F., Konrad, J., Ishwar, P.: Changedetection.net: a new change detection benchmark dataset. In: IEEE Workshop on Change Detection (CDW 12) at CVPR (2012)
5. Han, B., Davis, L.: On-line density-based appearance modeling for object tracking. In: Proceedings of the Tenth IEEE International Conference on Computer Vision, ICCV 05, IEEE Computer Society, vol. 2 , Washington, DC, USA, pp. 1492–1499.(2005). doi:10.1109/ICCV.2005.181. http://dx.doi.org/10.1109/ICCV.2005.181
6. Kaewtrakulpong, P., Bowden, R.: An improved adaptive background mixture model for real-time tracking with shadow detection. In: Proceedings of 2nd European Workshop on Advanced Video Based Surveillance Systems, vol. 5308 (2001)
7. Ko, T., Soatto, S., Estrin, D.: Background subtraction on distributions. European Conference on Computer Vision, ECCV '08, pp. 276–289. Springer, Berlin (2008)
8. Li, L., Huang, W., Gu, I.Y.H., Tian, Q.: Foreground object detection from videos containing complex background. In: ACM International Conference on Multimedia, pp. 2–10 (2003)
9. Liao, S., Zhao, G., Kellokumpu, V., Pietikäinen, M., Li, S.Z.: Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , pp. 1301–1306 (2010)
10. Mittal, A., Paragios, N.: Motion-based background subtraction using adaptive kernel density estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. II-302–II-309 (2004)
11. Narayana, M.: Automatic segmentation and tracking of moving objects in video for surveillance applications. Master's thesis, University of Kansas, Lawrence, Kansas, USA (2007)
12. Narayana, M., Hanson, A., Learned-Miller, E.: Background modeling using adaptive pixelwise kernel variances in a hybrid feature space. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
13. Narayana, M., Hanson, A., Learned-Miller, E.: Improvements in joint domain-range modeling for background subtraction. In: Proceedings of the British Machine Vision Conference. BMVA Press, pp. 115.1–115.11 (2012). http://dx.doi.org/10.5244/C.26.115
14. Porikli, F., Tuzel, O.: Bayesian background modeling for foreground detection. In: Proceedings of the third ACM international workshop on Video surveillance & sensor networks, VSSN 05. ACM, New York, NY, USA, pp. 55–58 (2005). doi:10.1145/1099396.1099407. http://doi.acm.org/10.1145/1099396.1099407
15. Sevilla-Lara, L., Learned-Miller, E.: Distribution fields for tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
16. Sheikh, Y., Shah, M.: Bayesian modeling of dynamic scenes for object detection. IEEE Trans. Pattern Anal. Mach. Intell. **27**, 1778–1792 (2005)
17. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 246–252 (1999)
18. Tavakkoli, A., Nicolescu, M., Bebis, G., Nicolescu, M.: Non-parametric statistical background modeling for efficient foreground region detection. Mach. Vis. Appl. **7**, 1–15 (2009)
19. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: principles and practice of background maintenance. In: IEEE International Conference on Computer Vision, vol. 1, pp. 255–261. doi:10.1109/ICCV.1999.791228. http://dx.doi.org/10.1109/ICCV.1999.791228
20. Turlach, B.A.: Bandwidth selection in kernel density estimation: a review. In: CORE and Institut de Statistique (1993)
21. Wand, M.P., Jones, M.C.: Kernel smoothing. Chapman and Hall, London (1995)
22. Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A.: Pfinder: real-time tracking of the human body. IEEE Trans.Pattern Anal. Mach. Intell. **19**, 780–785 (1997). doi:10.1109/34.598236
23. Yao, J., Odobez, J.M.: Multi-layer background subtraction based on color and texture. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2007)
24. Zivkovic, Z.: Improved adaptive gaussian mixture model for background subtraction. In: International Conference on Pattern Recognition (ICPR), vol. 2, pp. 28–31 (2004)
25. Zivkovic, Z., van der Heijden, F.: Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern Recognit. Lett. 27(7), 773–780 (2006). doi:10.1016/j.patrec.2005.11.005. http://dx.doi.org/10.1016/j.patrec.2005.11.005

Author Biographies

Manjunath Narayana is a Ph.D. candidate at the University of Massachusetts, Amherst. He obtained his M.S. degree in Computer Engineering from the University of Kansas in 2007 and his B.E. degree in Electronics and Communications Engineering from B. M. S. College of Engineering, Bangalore, India, in 2004. He is currently a research engineer at Metaio, Inc. working on computer vision algorithms for augmented reality applications. He interned in the Com-

puter Vision Systems Toolbox group in Mathworks, Inc. in 2012 developing object detection algorithms. He worked as a computer vision engineer in 2008 at Pixblitz Studios, Inc., a start-up company developing object detection systems for live video broadcasts. His research interests include machine learning, motion segmentation and tracking, face detection, object recognition, and augmented reality.

Allen R. Hanson received the BS degree from Clarkson College of Technology in 1964 and the MS and PhD degrees in electrical engineering from Cornell University in 1966 and 1969, respectively. He is Professor Emeritus in the Computer Science Department at the University of Massachusetts, Amherst and director of the Computer Vision Lab. His main research interests are in computer vision, particularly vision systems that are capable of functioning flexibly and robustly in complex changing environments, artificial intelligence, pattern recognition, and learning. He is the author of numerous technical papers in these areas, has been on the organizing committees of most of the major vision conferences, and founded two technology oriented companies. He is a member of the IEEE, ACM, and AAAI.

Erik G. Learned-Miller is an Associate Professor of Computer Science at the University of Massachusetts, Amherst, where he joined the faculty in 2004. He spent two years as a postdoctoral researcher at the University of California, Berkeley, in the Computer Science Division. Learned-Miller received a B.A. in Psychology from Yale University in 1988. In 1989, he co-founded CORITechs, Inc., where he co-developed the second FDA cleared system for image-guided neurosurgery. He worked for Nomos Corporation, Pittsburgh, PA, for two years as the manager of neurosurgical product engineering. He obtained M.S. (1997) and Ph. D. (2002) degrees in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology. In 2006, he received an NSF CAREER award for his work in computer vision and machine learning. He is a Program Chair for the 2015 IEEE Conference on Computer Vision and Pattern Recognition.