Background modeling using adaptive pixelwise kernel variances in a hybrid feature space DRAFT COPY

Manjunath Narayana
narayana@cs.umass.edu

Allen Hanson
hanson@cs.umass.edu

Erik Learned-Miller
elm@cs.umass.edu

University of Massachusetts, Amherst

Abstract

Recent work on background subtraction has shown developments on two major fronts. In one, there has been increasing sophistication of probabilistic models, from mixtures of Gaussians at each pixel [8], to kernel density estimates at each pixel [1], and more recently to joint domain-range density estimates that incorporate spatial information [7]. Another line of work has shown the benefits of increasingly complex feature representations, including the use of texture information, local binary patterns, and recently scale-invariant local ternary patterns [5]. In this work, we use joint domain-range based estimates for background and foreground label scores and show that dynamically choosing kernel sizes in our kernel estimates at each individual pixel can significantly improve results. We give a heuristic method for selectively applying the adaptive kernel calculations which is nearly as accurate as the full procedure but runs 5 to 10 times as fast. We combine these modeling improvements with recently developed complex features [5] and show significant improvements on a standard backgrounding benchmark.

1. Introduction

Background modeling is often an important step in detecting moving objects in video sequences [8, 4, 1]. A common approach to background modeling is to define and learn a background distribution over feature values at each pixel location and then classify each image pixel as belonging to the background process or not. The distributions at each pixel may be modeled in a parametric manner using a mixture of Gaussians [8] (MoG) or using non-parametric kernel density estimation [1] (KDE). More recently, models that allow a pixel's spatial neighbors to influence its distribution have been developed by joint domain-range density estimation [7]. These models that allow spatial influence from neighboring pixels have been shown to perform better than earlier neighbor-independent models.

Sheikh and Shah [7] also show that the use of an explicit foreground model along with a background model can be useful. In a manner similar to [7], we use a kernel estimate to obtain scores for the background and foreground labels at each pixel location using data samples from a spatial neighborhood around that location from previous frames. The score for the background label is computed as a kernel estimate depending on the distance in the joint domain-range space between the estimation point and the samples in the background model. A similar estimate is obtained for the foreground score. Each pixel is then assigned a (soft) label based on the ratio of the background and foreground scores.

The variance used in the estimation kernel reflects the spatial and appearance uncertainties in the scene. On applying our method to a data set with wide variations across the videos, we found that choosing suitable kernel variances during the estimation process is very important. With various experiments, we establish that the best kernel variance could vary for different videos and more importantly, even within a single video, different regions in the image should be treated with different variance values. For example, in a scene with a steady tree trunk and leaves that are waving in the wind, the trunk region can be explained with a small amount of spatial variance. The leaf regions may be better explained by a process with a large variance. Interestingly, when there is no wind, the leaf regions may also be explained with a low variance. The optimal variance hence changes for each region in the video and also across time. This phenomenon is captured reasonably in MoG [8] by use of different parameters for each pixel which adapt dynamically to the scene statistics, but the pixel-wise model does not allow a pixel's neighbors to affect its distribution. In [7], the process of updating the model with data samples from the most recent frame addresses the phenomenon partly. However, we show that using location-specific variances greatly improves background modeling. Our approach with pixel-wise variances, which we call the variable kernel score (VKS) method results in significant improvement over uniform variance models and state of the art backgrounding systems.

The idea of using a pixel-wise variance for background modeling is not new. Although [7] uses a uniform variance, it discusses the use of variances that change as a function of the data samples or as a function of the point at which the estimation is made (called *sample-point estimator* and *balloon estimator* in the literature respectively [3, 6]). Variance selection for KDE is a well studied problem [10] with common solutions including MISE, AMISE, and leave-one-out-estimator. In the background subtraction context, [6] and [9] use a different covariance at each pixel. While [6] require that the uncertainties in the feature values can be calculated in closed form, [9] learns the covariances for each pixel from a training set of frames and keeps the learned covariances fixed for the entire classification phase. We use a maximum-likelihood approach to select the best variance at each pixel location. For every frame of the video, at each pixel location, the best variance is picked from a set of variance values by maximizing the likelihood of the pixel's observation under different variances. This makes our method a *balloon estimator*. By explicitly selecting the best variance from a range of variance values, we do not require the covariances to be calculable in closed-form and also allow for more flexibility at the classification stage.

Selecting the best of many kernel variances for each pixel means increased computation. One possible trade-off between accuracy and speed can be achieved by a caching scheme where the best kernel variances from the previous frame are used to calculate the scores for the current frame pixels. If the resulting classification is overwhelmingly in favor of either label, there is no need to perform a search for the best kernel variance for that pixel. The expensive variance selection procedure can be applied only to pixels where there is some contention between the two labels. We present a heuristic that achieves significant reduction in computation compared to our full implementation while maintaining the benefits of adaptive variance.

Development and improvement of the probabilistic models is one of the two main themes in background modeling research in recent years. The other theme is the development of complex features like local binary [2] and ternary patterns [5] that are more robust than color features for the task of background modeling. Scale-invariant local ternary patterns [5] (SILTP) are recently developed features that have been shown to be very robust to lighting changes and shadows in the scene. By combining color features with SILTP features in our adaptive variance kernel model, we bring together the best ideas from both themes in the field and achieve state of the art results on a benchmark data set.

The main contributions of this paper are:
(1) A practical scheme for pixel-wise variance selection for background modeling.
(2) A heuristic for selectively updating variances to improve speed further.
(3) Incorporation of complex SILTP features into the joint domain-range kernel framework to achieve state of the art results.

The paper is organized as follows. Section 2 discusses our background and foreground models. Dynamic adaptation of kernel variances is discussed in section 3. Results and comparisons are in section 4. An efficient algorithm is discussed in section 5. We end with a discussion in section 6.

2. Background and foreground models

In a video captured by a static camera, the pixel values are influenced by the background phenomenon, and new or existing foreground objects. Any phenomenon that can affect image pixel values is called a process. Following the same principle as [7], we model the background and foreground processes using data samples from previous frames. A score for the background and foreground labels at each pixel location is calculated using contributions from the data samples in each model. One major difference between [7] and our model is that we allow the data samples to contribute probabilistically to the score of a label depending on the samples' probability of belonging to the label.

Let a pixel sample $a = [a_x a_y a_r a_g a_b]$, where (a_x, a_y) are the location of the pixel and (a_r, a_g, a_b) are the red, green, and blue values of the pixel. In each frame of the video, we compute background and foreground scores using pixel samples from the previous frames. The background model consists of the samples are $B = \{b_i : i \in [1\colon n_B]\}$ and foreground samples are $F = \{f_i : i \in [1 : n_F]\}$, with n_B and n_F being the number of background and foreground samples respectively, and b_i and f_i being pixel samples obtained from previous frames in the video. Under a KDE model [7], the likelihood of the sample under the background model is

$$P(a|bg; \sigma^B) = \frac{1}{n_B} \sum_{i=1:N_B} G(a - b_i; \sigma^B), \qquad (1)$$

where $G(x; \sigma)$ is a multivariate Gaussian with zero mean and covariance σ.

$$G(x; \sigma) = (2\pi)^{\frac{-D}{2}} |\sigma|^{\frac{-1}{2}} \exp(\frac{-1}{2} x^T \sigma^{-1} x), \qquad (2)$$

where D is the dimensionality of the vector x.

In our model, we approximate the score of the background process at sample a as

$$S_B(a; \sigma_d^B, \sigma_{rgb}^B) = \frac{1}{N_B} \sum_{i=1}^{n_B} \{G([a_r a_g a_b] - [b_{ir} b_{ig} b_{ib}]; \sigma_{rgb}^B) \times G([a_x a_y] - [b_{ix} b_{iy}]; \sigma_d^B) \times P(bg|b_i)\}. \qquad (3)$$

N_B is the number of frames from which the background samples have been collected, σ_d^B and σ_{rgb}^B are two and three dimensional background covariance matrices in spatial and color dimensions respectively. A large spatial covariance allows neighboring pixels to contribute more to the score at a given pixel location. Color covariance allows for some color appearance changes at a given pixel location.

The above equation basically sums the contribution from each background sample based on its distance in color space, weighted by its distance in spatial dimensions and the probability of the sample belonging to the background class.

The use of $P(bg|b_i)$ in equation 3 and normalization by the number of frames as opposed to the number of samples means that the score does not sum to 1 over all possible values of a. Thus, the score, although similar to the likelihood in equation 1, is not a probability distribution.

A similar equation holds for the foreground process:

$$S_F(a;\sigma_d^F,\sigma_{rgb}^F) = \frac{1}{N_F}\sum_{i=1}^{n_F}\{G([a_r a_g a_b]-[f_{ir} f_{ig} f_{ib}];\sigma_{rgb}^F) \times G([a_x a_y]-[f_{ix} f_{iy}];\sigma_d^F) \times P(fg|f_i)\}. \tag{4}$$

N_F is the number of frames from which the foreground samples have been collected, σ_d^F and σ_{rgb}^F are the covariances associated with the foreground process.

To classify a particular sample as background or foreground, we can use a Bayes-like formula:

$$P(bg|a) = \frac{S_B(a;\sigma_d^B,\sigma_{rgb}^B)}{S_B(a;\sigma_d^B,\sigma_{rgb}^B + S_F(a;\sigma_d^F),\sigma_{rgb}^F)} \tag{5}$$
$$P(fg|a) = 1 - P(bg|a).$$

2.1. Poor estimates for low-scoring pixel samples

For pixel samples that are very far away from both the background and foreground samples, the above equation results in instability in the classification. In practice, if a color that is very different from the existing colors in the scene is observed at a location (a_x, a_y), it would result in very low scores (or likelihoods in a proper KDE) for both the background and foreground labels. The resulting classification using equation 5 could return arbitrary results. To account for such new colors as belonging to the foreground, we add a constant term ϵ to the denominator resulting in a modified equation :

$$P(bg|a) = \frac{S_B(a;\sigma_d^B,\sigma_{rgb}^B)}{S_B(a;\sigma_d^B,\sigma_{rgb}^B + S_F(a;\sigma_d^F,\sigma_{rgb}^F)) + \epsilon} \tag{6}$$
$$P(fg|a) = 1 - P(bg|a).$$

The modification has some very interesting properties. When either or both of the background and foreground scores is much larger than ϵ, ϵ has an insignificant effect in the equation. However, if both the background and foreground scores are comparable to ϵ, then $P(bg|a)$ is significantly reduced. ϵ acts as a natural threshold that applies the proper Bayes-like rule when either of the scores is much larger than ϵ and favors the foreground label when both scores are comparable to ϵ. This is significant for the background-foreground modeling problem because if a pixel color is not explainable by both models, it is natural to assume that the pixel is a result of a new object in the scene. We use $\epsilon = 10^{-6}$. We believe that this technique for biasing the decision in favor of a particular class in the absence of evidence for either class can find applications in other classification scenarios as well.

2.2. Model initialization and update

To initialize the models, it is assumed that the first few frames (typically 50) are all background pixels. The background model is populated using pixel samples from these frames. In order to improve efficiency, we sample 5 frames at equal time intervals from these 50 frames in order to compute the score in equation 4 ($N_B = 5$). The foreground model is initialized to have no samples. The modified equation 6 classifies colors that are not well explained by the background model to be classified as foreground, thus bootstrapping the foreground model. Samples from the previous 5 frames are used for the foreground model ($N_F = 5$). Once the pixel at location (a_x, a_y) from a new frame is classified using equation 5, the background and foreground models at the location (a_x, a_y) can then be updated with the new sample a. Background and foreground samples at location (a_x, a_y) from the oldest frame in the models are replaced by a. The label probabilities of the background/foreground from equation 6 are also saved along with the sample values subsequent for use in the score equations 3 and 4.

One consequence of the update procedure described above is that when a large foreground object occludes a background pixel at (a_x, a_y) for more than N_B number of frames, all the background samples in the spatial neighborhood of (a_x, a_y) are replaced by these foreground samples that have very low $P(bg|b_i)$ values. This causes the pixel at (a_x, a_y) to be misclassified as foreground even when the occluding foreground object has moved away (because the background score will be extremely low due to the influence of $P(bg|b_i)$ in equation 3). To avoid this problem, we replace the background sample from location (a_x, a_y) in the oldest frame in the background model with the new sample a from the current frame only if $P(bg|a)$ estimated from equation 6 is greater than 0.5.

In our chosen evaluation data set, there are several videos with moving objects in the first 50 frames. The assumption that all these pixels are background is not severely limiting even in these videos. The model update procedure allows us to recover from any errors that are caused by the presence of foreground objects in the initialization frames.

2.3. Using MRF to clean the classification

Following the procedure as described in [7], we use a Markov random field (MRF) defined over the posterior label probabilities of the 4-neighbors of each pixel and perform the min-cut procedure to post-process the labels. The λ interaction factor between the nodes was set to 1 for all our experiments.

3. Pixel-wise adaptive kernel variance selection

Background and foreground kernels - Sheikh and Shah [7] use the same kernel parameters for background and foreground models. Given the different nature of the two processes, it is reasonable to use different kernel parameters. For instance, foreground objects typically move between 5 and 10 pixels per frame in the I2R data set, whereas background pixels are either stationary or move very little. Hence, it useful to have a larger spatial variance for the foreground model than for the background model.

Optimal kernel variance for all videos - In the results section, we show that for a data set with large variations like I2R [4], a single value for kernel variance for all videos (as done in [7]) is not sufficient to capture the variability in all the videos.

Variable kernel variance for a single video- As explained in the introduction, different parts of the scene may have different statistics and hence need different kernel variance values. For example, in the Figure 1a to 1d, 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 (Figure 1e to 1h), having a high kernel variance allows accurate classification of some of the fountain pixels as background at the cost of misclassifying many foreground pixels. The figure also shows that while the medium kernel variance may be the best choice for the first video, the low kernel variance may be best for the second video.

Optimal kernel variance for classification- Having different variances for the background and foreground models reflects the differences between the expected uncertainty in the two processes. However, having different variances for the two processes could cause erroneous classification of pixels. Figure 2 shows a 1-dimensional example where using a very wide kernel (high variance) or very narrow kernel for the background process causes misclassification. Assuming that the red point (square) is a background sample and the blue point (triangle) is a foreground sample, having a very low variance kernel (dashed red line) or a very high variance (solid red line) for the background process makes the background likelihood of the center point 'x' lower than the foreground likelihood. Thus, it is important to pick the optimal kernel variance for each process while classifying between the two processes.

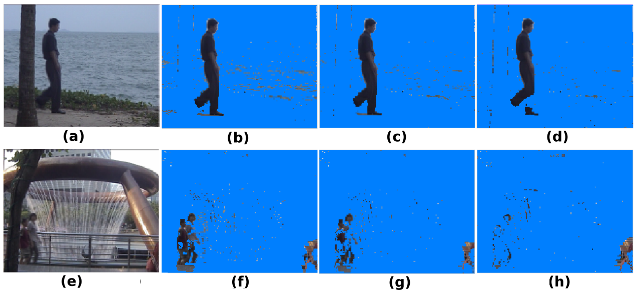

Figure 1. 2 video sequences classified using increasing values of spatial kernel variance. Column 1 - original image, Column 2 - $\sigma_d^B = 1/4$, Column 3 - $\sigma_d^B = 3/4$, Column 4 - $\sigma_d^B = 12/4$

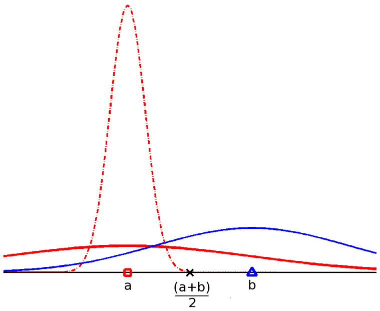

Figure 2. 1-dimensional example showing role of the kernel variance in classification

In order to address all four issues discussed above, we propose the use of location-specific variances. For each location in the image, a range of kernel variances is tried and the variance which results in the highest score is chosen for the background and the foreground models separately.

The background score with location-dependent variances is

$$S_B(a;\sigma^B_{d,a_x,a_y},\sigma^B_{rgb,a_x,a_y}) = \frac{1}{N_B}\sum_{i=1}^{n_B}\{ \quad G([a_r a_g a_b] - [b_{ir} b_{ig} b_{ib}]; \sigma^B_{rgb,a_x,a_y}) \times G([a_x a_y] - [b_{1x} b_{1y}]; \sigma^B_{d,a_x,a_y}) \times P(bg|b_i)\}, \quad (7)$$

where $\sigma^B_{d,x,y}$ and $\sigma^B_{rgb,x,y}$ represent the location-specific spatial and color dimension variances at location (x, y).

For each pixel location (a_x, a_y), the optimal variance for the background process is selected by maximizing the score of the background label at sample a under different variance values:

$$\{\sigma^{B*}_{d,a_x,a_y},\sigma^{B*}_{rgb,a_x,a_y}\} = \operatorname*{argmax}_{\sigma^{B}_{d,a_x,a_y},\sigma^{B}_{rgb,a_x,a_y}} S_B(a;\sigma^{B}_{d,a_x,a_y},\sigma^{B}_{rgb,a_x,a_y}). \qquad (8)$$

Here, $\sigma^{B}_{d,a_x,a_y} \in R^{B}_{d}$ and $\sigma^{B}_{rgb,a_x,a_y} \in R^{B}_{rgb}$. R^{B}_{d} and R^{B}_{rgb} represent the set of spatial and color dimension variances to choose the optimal variance from.

A similar procedure may be followed for the foreground score. However, in practice, it was found that the variance selection procedure yielded large improvements when applied to the background model and little improvement in the foreground model. Hence, our final implementation uses an adaptive kernel variance procedure for the background model and a fixed kernel variance for the foreground model.

4. Results

For comparisons, we use the I2R data set [4] which consists of 9 videos taken using a static camera in various environments. The data set offers various challenges including dynamic background like trees and waves, gradual and sudden illumination changes, and the presence of multiple moving objects. Ground truth for 20 frames in each video is provided with the data set. The F-measure as defined in [5] is used to measure accuracy.

The effect of choosing various kernel widths for the background and foreground models is shown in Table 1. The table shows the F-measure for each of the videos in the data set for various choices of the kernel variances. The first 5 columns correspond to using a constant variance for each process at all pixel locations in the video. Having identical kernel variances for the background and foreground models (columns 1, 2) is not as effective as having different variances (all other columns). Comparing columns 2 and 3 shows that using a larger spatial variance for the foreground model than for the background model is beneficial. Changing the spatial variance from 3 (column 3) to 1 (column 4) helps the overall acccuracy in one video (Fountain). Using a selection procedure where the best kernel variance is chosen from a set of values gives the best results for most videos (column 6) and frames.

Comparison of our selection procedure to a baseline method of using a standard algorithm for variance selection in KDE (AMISE criterion[1]) shows that the standard algorithm is not as accurate as our method (column 7). Our choice for the variance values for spatial dimension reflects no motion ($\sigma^{B}_{d}= 1/4$) and very little motion ($\sigma^{B}_{d}= 3/4$) for the background, and moderate amount of motion ($\sigma^{F}_{d}= 12/4$) for the foreground. For the color dimension, the choice is between little variation ($\sigma^{B}_{rgb}= 5/4$), moderate variation ($\sigma^{B}_{rgb}= 15/4$), and high variation ($\sigma^{B}_{rgb}= 45/4$) for the background, and moderate variation ($\sigma^{F}_{rgb}= 15/4$) for the foreground. These choices are based on our intuition about the processes involved. The baseline method may well be more successful if this information is not known beforehand. The baseline method however takes 6 times as long to execute.

We would like to point out that ideally the variance value sets should be learned automatically from a separate training data set. In absence of suitable training data for these videos in particular and for background subtraction research in general, we resort to manually choosing these values. This also appears to be the common practice among researchers in this area.

Benchmark comparisons are provided for selected existing methods - MOG [8], the complex foreground model [4] (ACMMM03), and SILTP [5]. To evaluate our results, the posterior probability of the background label is thresholded at a value of 0.5 to get the foreground pixels. Following the same procedure as [5], any foreground 4-connected components smaller than a size threshold of 15 pixels are ignored.

Figure 3 shows qualitative results for the same frames that were reported in [5]. We present results for our kernel method with uniform variances and adaptive variances with RGB features (Uniform-rgb and VKS-rgb respectively), and adaptive variances with a hybrid feature space of LAB color and SILTP features (VKS-lab+siltp). Except for the Lobby video, the VKS results are better than other methods. The Lobby video is an instance where there is a sudden change in illumination in the scene (turning a light switch on and off). Due to use of an explicit foreground model, our kernel methods misclassify most of the pixels as foreground and take a long time to recover from this error. A possible solution for this case is presented later. Compared to the uniform variance kernel estimates, we see that VKS-rgb has fewer false positive foreground pixels.

Quantitative results in Table 2 compare the F-measure scores for our method against MoG, ACMMM03, and SILTP results as reported in [5]. The table shows that methods that share spatial information (uniform kernel and VKS) with RGB features give significantly better results than methods that use RGB features without spatial sharing. Comparing the variable kernel method to a uniform kernel method in the same feature space (RGB), we see a significant improvement in performance for most videos. Scale-invariant local ternary pattern (SILTP) [5] is a recent texture feature that is robust to soft shadows and lighting changes. We believe SILTP represents the state of the art in background modeling and hence compare our results to this method. Scale-invariant local states [12] is a slight variation in the representation of the SILTP feature. For comparison, we use SILTP results from [5] because in [12], hu-

[1] We use the publicly available implementation from http://www.ics.uci.edu/ ihler/code/kde.html.

Column num	(1)	(2)	(3)	(4)	(5)	(6)	(7)
$4{*}\sigma_d^B \longrightarrow$	3	3	3	1	3	[1 3]	AMISE
$4{*}\sigma_{rgb}^B \longrightarrow$	15	45	45	45	15	[5 15 45]	AMISE
$4{*}\sigma_d^F \longrightarrow$	3	3	12	12	12	[12]	[12]
$4{*}\sigma_{rgb}^F \longrightarrow$	15	45	45	45	15	[15]	[15]
AirportHall	40.72	59.53	67.07	63.53	47.21	**70.44**	53.01
Bootstrap	49.01	57.90	63.04	58.39	51.49	**71.25**	63.38
Curtain	66.26	83.33	91.91	89.52	81.54	**94.11**	52.00
Escalator	20.92	30.24	34.69	28.58	22.65	**48.61**	32.02
Fountain	41.87	51.89	73.24	74.58	67.60	**75.84**	28.50
ShoppingMall	55.19	60.17	64.95	62.18	63.85	**76.48**	70.14
Lobby	22.18	23.81	25.79	25.69	25.06	18.00	**36.77**
Trees	30.14	58.41	73.53	47.03	67.80	**82.09**	64.30
WaterSurface	85.82	94.04	**94.93**	92.91	94.64	94.83	30.29
Average	45.79	57.70	65.46	60.27	52.98	**70.18**	47.82

Table 1. *F-measure* on different kernel variances. Using a selection procedure results in better performance than using fixed variances

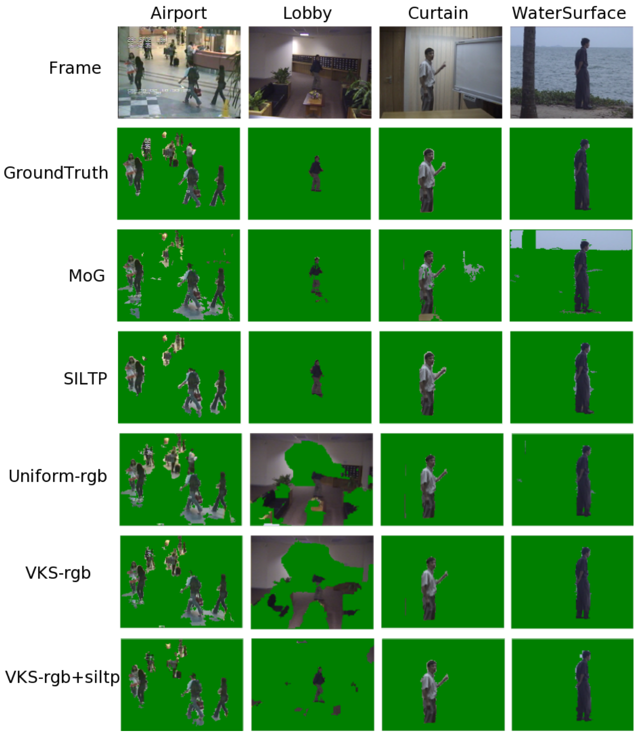

Figure 3. Qualitative comparison of algorithms on image results reported in [5]

man judgement[2] was used to vary a size threshold parameter for each video. We believe results from [12] fall under a different category of human-assisted backgrounding and hence do not compare to our method where no video-specific hand-tuning of parameters was done. Table 2 shows that SILTP is very robust to lighting changes and works well across the entire data set. Blue entries in the table correspond to videos where our method performs better than SILTP. VKS with RGB features (VKS-rgb) performs well in videos that have few shadows and lighting changes. Use of color features that are more robust to illumination change, like LAB features in place of RGB helps in successful classification of the shadow regions as background. Texture 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. By combining texture features with LAB color features, we expect to benefit from the strengths of both feature spaces. Such a combination has proved useful in earlier work [11]. Augmenting the LAB features with SILTP features (computed at 3 resolutions) in the VKS framework (VKS-lab+siltp) results in an improvement in 7 out of 9 videos (last column). The variance values used for VKS were:
([ ] indicates selection set)
For both models: $4{*}\sigma_d^B = [1, 3], 4{*}\sigma_d^F = 12$
rgb model : $4{*}\sigma_{rgb}^B = [5, 15, 45], 4{*}\sigma_{rgb}^F = 15$
lab+siltp model :
$4{*}\sigma_l^B = [5, 10, 20], 4{*}\sigma_{ab}^B = [4, 6], 4{*}\sigma_{siltp}^B = 3,$
$4{*}\sigma_l^F = 15, 4{*}\sigma_{ab}^F = 4, 4{*}\sigma_{siltp}^F = 3$
For SILTP space, the XOR operation between the binary representations of two SILTP features gives the distance between them [5].

We also compare our results (VKS-lab+siltp) to the 5 videos that were submitted as supplementaray material along with [5]. Figure 4 highlights some key frames that highlight the strengths and weaknesses of our system versus the SILTP results. The common problems with our algorithm are shadows being classified as foreground (row e) and initialization errors (row e shows a scene where the desk was occluded by people when the background model

[2] This was learned via personal communication with the authors.

was initialized. Due to the explicit foreground model, VKS takes some time to recover from the erroneous initialization). A common drawback with SILTP is that large textureless objects have ”holes” in them (row a). Use of color features helps avoid these errors. The SILTP system also loses objects that stop moving (rows b, c, d, f). Due to the explicit modeling of the foreground, VKS is able to detect objects that stop moving.

The two videos in the dataset where our algorithm performs worse than SILTP are the Escalator video (rows g, h) and the Lobby video (rows i, j). In the Escalator video, our algorithm fails at the escalator steps due to large variation in color in the region.

In the Lobby video, at the time of sudden illumination change, many pixels in the image get classified as foreground. Due to the foreground model, these pixels continue to be misclassified for a long duration (row j). The problem is more serious for RGB features (Figure 3 column 2). One method to address the situation is to observe the illumination change from one frame to the next. If more than half the pixels in the image change in illumination by a threshold value of T_I or more, we throw away all the background samples at that instance and begin learning a new model from the subsequent 50 frames. This method allows us to address the poor performance in the Lobby video with resulting F-measure values of 86.77 for uniform-rgb, 78.46 for VKS-rgb, and 77.76 for VKS-lab+siltp. T_I of 10 and 2.5 were used for RGB and LAB spaces respectively. The illumination change procedure does not affect the performance of VKS on any other video in the data set.

5. Caching optimal kernel variances from previous frame

A major drawback with trying multiple variance values at each pixel to select the best variance is that the amount of computation per pixel increases significantly. In order to reduce the complexity the algorithm, we use a scheme where the current frame’s optimal variance values for each pixel location for both the background and foreground processes is stored $(\sigma_{x,y}^{*B_{cache}}, \sigma_{x,y}^{*F_{cache}})$ for each location (x, y) in the image. When classifying pixels in the next frame, these cached variance values are first tried. If the resulting scores are very far apart, then it is very likely that the pixel has not changed its label from the previous frame. The expensive variance selection procedure is performed only at pixels where the resulting scores are close to each other. Algorithm 1 for efficient computation results in a reduction in computation in about 80% of the pixels in the I2R videos when τ_{BF} is set to 2, with a slight reduction in the F-measure by about 1 to 2% on most videos when compared to the full implementation. The efficient variance selection procedure however still performs significantly better than the uniform variance model by 2 to 10% on most videos.

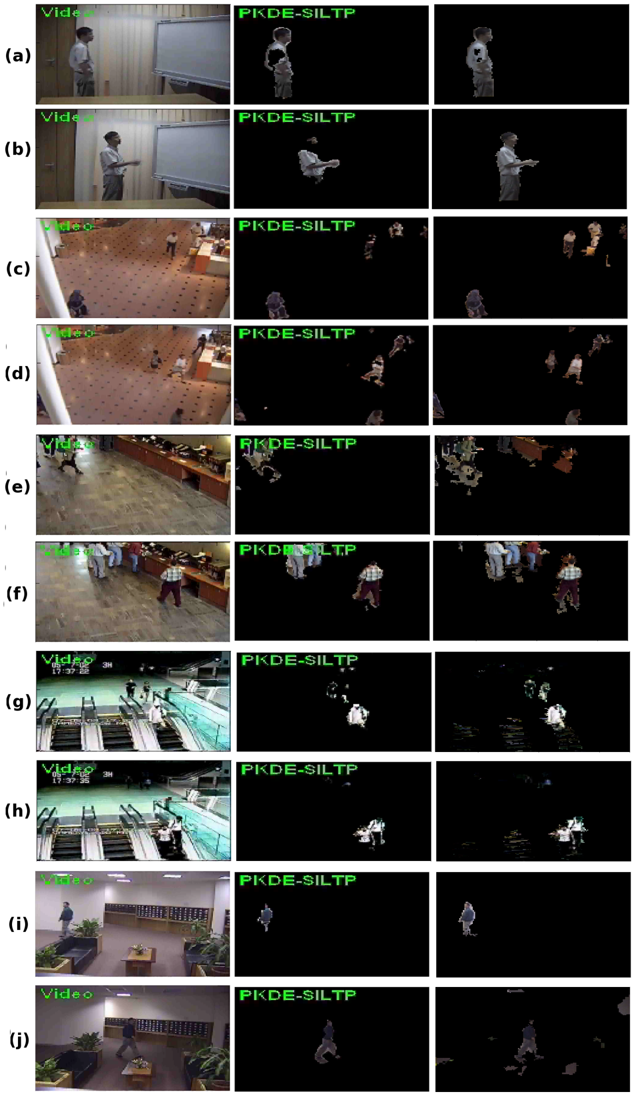

Figure 4. Comparing VKS results to SILTP video results from [5]. Column 1 - original video frame, column 2 - SILTP results [5], column 3 - VKS-lab+siltp results

6. Discussion

By applying kernel estimate method to a large data set, we have established, as do [7], that the use of spatial information is extremely helpful. Some of the important issues pertaining to the choice of kernel parameters for data set with wide variations have been addressed. Having a uniform kernel variance for the entire data set and for all pixels in the image results in a poor overall system. Dynamically adapting the variance for each pixel results in a significant increase in accuracy.

Using color features in the joint domain-range kernel estimation approach can complement recent complex background model features in settings the where the later are

Video	ACMMM03	MoG	SILTP [5]	uniform	VKS	VKS
				rgb	rgb	lab+siltp
AirportHall	50.18	57.86	68.02	67.07	70.44	**71.28**
Bootstrap	60.46	54.07	72.90	63.04	71.25	**76.89**
Curtain	56.08	50.53	92.40	91.91	**94.11**	94.07
Escalator	32.95	36.64	**68.66**	34.69	48.61	49.43
Fountain	56.49	77.85	85.04	73.24	75.84	**85.97**
ShoppingMall	67.84	66.95	79.65	64.95	76.48	**83.03**
Lobby	20.35	68.42	**79.21**	25.79	18.00	60.82
Trees	75.40	55.37	67.83	73.53	82.09	**87.85**
WaterSurface	63.66	63.52	83.15	**94.93**	94.83	92.61

Table 2. *F-measure* on I2R data. VKS significantly outperforms other color feature-based methods and improves on SILTP texture features on most videos. Blue color indicates performance better than SILTP

Algorithm 1 Efficient variance selection

```
for each pixel sample $a = (a_x, a_y, a_r, a_g, a_b)$ in the current frame do
  if $\frac{S_B(a;\sigma^{*Bcache}_{a_x,a_y})}{S_B(a;\sigma^{*Bcache}_{a_x,a_y})} > \tau_{BF}$ then
    Compute the label scores resulting from use of the cached variance values
  else
    Search over the values in the variance sets to pick the optimal variances
    Compute the label scores using the optimal variances
  end if
end for
```

known to be inaccurate. Combining robust color features like LAB with texture features like SILTP in a VKS framework yields a highly accurate background classification system.

For future work, we believe our method could be explained more elegantly in a probabilistic framework where the scores are replaced by the true likelihoods and informative priors are incorporated. Equation 6 provides sound classification decisions in the absence of supporting evidence for either class. It could be useful in other classification problems to bias the decision in favor of a particular class when the likelihoods are inconclusive.

References

[1] A. M. Elgammal, D. Harwood, and L. S. Davis. Non-parametric model for background subtraction. In *Proceedings of the 6th European Conference on Computer Vision*, pages 751–767, 2000. 1

[2] M. Heikkila and M. Pietikainen. A texture-based method for modeling the background and detecting moving objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(4):657 –662, 2006. 2

[3] M. Jones. Variable kernel density estimates and variable kernel density estimates. *Australian Journal of Statistics*, 32(3):361–371, 1990. 2

[4] L. Li, W. Huang, I. Y. H. Gu, and Q. Tian. Foreground object detection from videos containing complex background. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 2–10, 2003. 1, 4, 5

[5] S. Liao, G. Zhao, V. Kellokumpu, M. Pietikainen, and S. Li. Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 1301 –1306, 2010. 1, 2, 5, 6, 7, 8

[6] A. Mittal and N. Paragios. Motion-based background subtraction using adaptive kernel density estimation. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, pages II–302 – II–309 Vol.2, 2004. 2

[7] Y. Sheikh and M. Shah. Bayesian modeling of dynamic scenes for object detection. *PAMI*, 27:1778–1792, 2005. 1, 2, 4, 7

[8] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, volume 2, pages 246–252, 1999. 1, 5

[9] A. Tavakkoli, M. Nicolescu, G. Bebis, and M. Nicolescu. Non-parametric statistical background modeling for efficient foreground region detection. *Mach. Vision Appl.*, 7:1–15, 2009. 2

[10] B. A. Turlach. Bandwidth selection in kernel density estimation: A review. In *CORE and Institut de Statistique*, 1993. 2

[11] J. Yao and J.-M. Odobez. Multi-layer background subtraction based on color and texture. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1 –8, 2007. 6

[12] J.-C. Yuk and K.-Y. Wong. An efficient pattern-less background modeling based on scale invariant local states. In *Advanced Video and Signal-Based Surveillance (AVSS), 8th IEEE International Conference on*, pages 285 –290, 2011. 5, 6