A Robust SVM for Density based Multifeatured Background Detection/Subtraction

[1]Sireesha K., [2]A.Vishnu Vardhan

[1]Department of Computer science engineering, VVIT, GUNTUR,Ap,India.

2 Assoc. Professor Department of Computer science Engineering,VVIT, Guntur,Ap,India.

***Abstract*— Background modeling and subtraction is a natural technique for object detection in videos captured by a static camera, and also a critical preprocessing step in various high-level computer vision applications. However, there have not been many studies concerning useful features and binary segmentation algorithms for this problem. We propose a pixelwise background modeling and subtraction technique using multiple features, where generative and discriminative techniques are combined for classification. In our algorithm, color, gradient, and Haar-like features are integrated to handle spatio-temporal variations for each pixel. A pixelwise generative background model is obtained for each feature efficiently and effectively by Kernel Density Approximation (KDA). Background subtraction is performed in a discriminative manner using a Support Vector Machine (SVM) over background likelihood vectors for a set of features. The proposed algorithm is robust to shadow, illumination changes, spatial variations of background. We compare the performance of the algorithm with other densitybased methods using several different feature combinations and modelling techniques, both quantitatively and qualitatively.**

Keywords*— *Background modeling and subtraction, Haar-like features, support vector machine, kernel density approximation.

1. Introduction

Both wind energy and ocean energy have been the identification of regions of interest is typically the first step in many computer vision applications, including event detection, visual surveillance, and robotics. A general object detection algorithm may be desirable, but it is extremely difficult to properly handle unknown objects or objects with significant variations in color, shape, and texture. Therefore, many practical computer vision systems assume a fixed camera environment, which makes the object detection process much more straightforward; a background model is trained with data obtained from empty scenes and foreground regions are identified using the dissimilarity between the trained model and new observations. This procedure is called background subtraction. Various background modeling and subtraction algorithms have been proposed [1], [2], [3], [4], [5] which are mostly focused on modeling methodologies, but potential visual features for effective modeling have received relatively little attention. The study of new features for background modeling may overcome or reduce the limitations of typically used features, and the combination of several heterogeneous features can improve performance, especially when they are complementary and uncorrelated. There have been several studies for using texture for background modeling to handle spatial variations in the scenes; they employ filter responses, whose computation is typically very costly. Instead of complex filters, we select efficient Haar-like features [6] and gradient features to alleviate potential errors in background subtraction caused by shadow, illumination changes, and spatial and structural variations. Model-based approaches involving probability density function are common in background modeling and subtraction, and we employ Kernel Density Approximation (KDA) [3], [7], where a density function is represented with a compact weighted sum of Gaussians whose number, weights, means, and covariances are determined automatically based on mean-shift mode-finding algorithm. In our framework, each visual feature is modeled by KDA independently and every density function is 1D. By utilizing the properties of the 1D mean-shift mode-finding procedure, the KDA can be implemented efficiently because we need to compute the convergence locations for only a small subset of data. When the background is modeled with probability density functions, the probabilities of foreground and background pixels should be discriminative, but it is not always true. Specifically, the background probabilities between features may be inconsistent due to illumination changes, shadow, and foreground objects similar in features to the background. Also, some features are highly correlated, i.e., RGB color features. So, we employ a Support Vector Machine (SVM) for nonlinear classification, which mitigates the inconsistency and the correlation problem among features. The final classification between foreground and background is based on the outputs of the SVM.

There are three important aspects of our algorithm—integration of multiple features, efficient 1D density estimation by KDA, and foreground/background classification by SVM. These are coordinated tightly to improve background subtraction performance. An earlier version of this research appeared in [8]; the current paper includes more comprehensive analysis of the feature sets and additional experiments.

PREVIOUS WORK

The main objective of background subtraction is to obtain an effective and efficient background model for foreground object detection. In the early years, simple statistics, such as frame differences and median filtering, were used to detect foreground objects. Some techniques utilized a combination

of local statistics [9] or vector quantization [10] based on intensity or color at each pixel. More advanced background modeling methods are densitybased, where the background model for each pixel is defined by a probability density function based on the visual features observed at the pixel during a training period. Wren et al. [11] modeled the background in YUV color space using a Gaussian distribution for each pixel. However, this method cannot handle multimodal density functions, so it is not robust in dynamic environments. A mixture of Gaussians is another popular density-based method which is designed for dealing with multiple backgrounds. Three Gaussian components representing the road, shadow, and vehicle are employed to model the background in traffic scenes in [12]. An adaptive Gaussian mixture model is proposed in [2] and [13], where a maximum of K Gaussian components for each pixel are allowed but the number of Gaussians is determined dynamically. Also, variants of incremental EM algorithms have been employed to deal with real-time adaptation constraints of background modeling [14], [15]. However, a major challenge in the mixture model is the absence or weakness of strategies to determine the number of components; it is also generally difficult to add or remove components to/from the mixture [16]. Recently, more elaborate and recursive update techniques are discussed in [4] and [5]. However, none of the Gaussian mixture models have any principled way to determine the number of Gaussians. Therefore, most real-time applications rely on models with a fixed number of components or apply ad hoc strategies to adapt the number of mixtures over time [2], [4], [5], [13], [17]. Kernel density estimation is a nonparametric density estimation technique that has been successfully applied to background subtraction [1], [18]. Although it is a powerful representation for general density functions, it requires many samples for accurate estimation of the underlying density functions and is computationally expensive, so it is not appropriate for real-time applications, especially when high-dimensional features are involved. Most background subtraction algorithms are based on pixelwise processing, but multilayer approaches are also introduced in [19] and [20], where background models are constructed at the pixel, region, and frame levels and information from each layer is combined for discriminating foreground and background. In [21], several modalities based on different features and algorithms are integrated and a Markov Random Field (MRF) is employed for the inference procedure. The co-occurrence of visual features within neighborhood pixels is used for robust background subtraction in dynamic scenes in [22].

Some research on background subtraction has focused more on features than the algorithm itself. Various visual features may be used to model backgrounds, including intensity, color, gradient, motion, texture, and other general filter responses. Color and intensity are probably the most popular features for background modeling, but several attempts have been made to integrate other features to overcome their limitations. There are a few algorithms based on motion cue [18], [23]. Texture and gradients have also been successfully integrated for background modeling [24], [25] since they are relatively invariant to local variations and illumination changes. In [26], spectral, spatial, and temporal features are combined, and background/foreground classification is performed based on the statistics of the most significant and frequent features. Recently, a feature selection technique was proposed for background subtraction [27].

3 BACKGROUND MODELING AND SUBTRACTION ALGORITHM

This section describes our background modeling and subtraction method based on the 1D KDA using multiple features. KDA is a flexible and compact density estimation technique [7], and we present a faster method to implement KDA for 1D data. For background subtraction, we employ the SVM, which takes a vector of probabilities obtained from multiple density functions as an input.

3.1 Multiple Feature Combination

The most popular features for background modeling and subtraction are probably pixelwise color (or intensity) since they are directly available from images and reasonably discriminative. Although it is natural to monitor color variations at each pixel for background modeling, they have several significant limitations as follows:

- They are not invariant to illumination changes and shadows.
- Multidimensional color features are typically correlated and joint probability modeling may not be advantageous in practice.
- They rely on local information only and cannot handle structural variations in neighborhoods.

We integrate color, gradient, and Haar-like features together to alleviate the disadvantages of pixelwise color modeling. The gradient features are more robust to illumination variations than color or intensity features and are able to model local statistics effectively. So, gradient features are occasionally used in background modeling problems [26], [27]. The strength of Haar-like features lies in their simplicity and the ability to capture neighborhood information. Each Haar-like feature is weak by itself, but the collection of weak features has significant classification power [6], [28]. The integration of these features is expected to improve the accuracy of background subtraction. We have 11 features altogether, RGB color, two gradient features (horizontal and vertical), and six Haar-like features. The Haar-like features employed in our implementation are illustrated in Fig. 1. The Haar-like features are extracted from 9 X 9 rectangular regions at each location in the image, while the gradient features are extracted with 3 X 3 Sobel operators. The fourth and fifth Haarlike features are similar to the gradient features, but differ in filter design, especially scale.

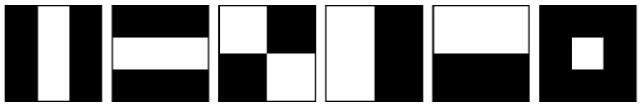


Fig. 1. Haar-like features for our background modeling.

3.2 Background Modeling by KDA

The background probability of each pixel for each feature is modeled with a Gaussian mixture density function.1 There are various methods to implement this idea, and we adopt KDA, where the density function for each pixel is represented with a compact and flexible mixture of Gaussians. The KDA is a density approximation technique based on mixture models, where mode locations (local maxima) are detected automatically by the meanshift algorithm and a single Gaussian component is assigned to each detected mode. The covariance for each Gaussian is computed by curvature fitting around the associated mode. More details about KDA are found in [7], and we summarize how to build a density function by KDA for 1D case.

Suppose that training data for a sequence are composed of n frames and $x_{F,i}$ (i=1,2,3…) is the ith value for feature F at a certain location. Note that the index for pixel location is omitted for simplicity. For each feature, we first construct a 1D density function at each pixel by kernel density estimation based on Gaussian kernel as follows:

$$\hat{f}_F(x) = \frac{1}{\sqrt{2\pi}} \sum_{i=1}^{n} \frac{\kappa_{F,i}}{\sigma_{F,i}} \exp\left(-\frac{(x - x_{F,i})^2}{2\sigma_{F,i}^2} \right), \qquad (1)$$

As described above, the KDA finds local maxima in the underlying density function (1), and a mode-based representation of the density is obtained by estimating all the parameters for a compact Gaussian mixture. The original density function is simplified by KDA as

$$\tilde{f}_F(x) = \frac{1}{\sqrt{2\pi}} \sum_{i=1}^{m_F} \frac{\tilde{\kappa}_{F,i}}{\tilde{\sigma}_{F,i}} \exp\left(-\frac{(x - \tilde{x}_{F,i})^2}{2\tilde{\sigma}_{F,i}^2} \right), \qquad (2)$$

3.3 Optimization in One Dimension

We are only interested in 1D density functions, and the convergence of each sample point can be obtained much faster than the general case. Given 1D samples, x_i and x_j (i; j ¼ 1; . . . ; n), the density function created by kernel density estimation has the following properties:

$$(x_i \le x_j \le \hat{x}_i) \vee (x_i \ge x_j \ge \hat{x}_i) \Rightarrow \hat{x}_i = \hat{x}_j,$$

We sort the sample points in ascending order, and start performing mean-shift mode finding from the smallest sample. When the current sample moves in the gradient ascent direction by the mean-shift algorithm in the underlying density function and passes another sample's location during the iterative procedure, we note that the convergence point of the current sample must be the same as the convergence location of the sample just passed, terminate the current sample's mean-shift process, and move on to the next smallest sample, where we begin the mean-shift process again.[3] If a mode is found during the mean-shift iterations, its location is stored and the next sample is considered. After

2. Strictly speaking, it may be sometimes necessary to compute the convergences at more locations due to a cascaded merges.

This can be implemented using a reference variable. If a sample is moving backward, its convergence location is the same as the previous one. If a sample is moving forward, its convergence location can be set when the next mode is found.

3.4 Foreground and Background Classification

After background modeling, each pixel is associated with k 1D Gaussian mixtures, where k is the number of features integrated. Background/foreground classification for a new frame is per-formed using these distributions. The background probability of a feature value is computed by (2), and k probability values are obtained from each pixel, which are represented by a k-dimen-sional vector. Such k-dimensional vectors are collected from annotated foreground and background pixels, and we denote them by y_j (j ¼ 1; . . . ; N), where N is the number of data points.

In most density-based background subtraction algorithms, the probabilities associated with each pixel are combined in a straightforward way, either by computing the average probability or by voting for the classification. However, such simple methods may not work well under many real-world situations due to feature dependency and nonlinearity. For example, pixels in shadow may have a low-background probability in color modeling unless shadows are explicitly modeled as transformations of color variables, but high-background probability in texture modeling. Also, the foreground color of a pixel can look similar to the corresponding background model, which makes the background probability high although the texture probability is probably low. Such inconsistency among features is aggravated when many features are integrated and data are high dimensional, so we train a classifier over the background probability vectors for the feature set, $fy_jg_{1:N}$. Another advantage to integrating the classifier for foreground/background segmentation is to select discriminative features and reduce the feature dependency problem; otherwise, highly correlated nondiscriminative features may dominate the classification process regardless of the states of other features.

An SVM is employed for the classification of background and foreground. We used a radial basis function kernel to handle nonlinear input data, and used approximately 40 K data points (10 K for foreground and 30 K for background) for training. Note that a universal SVM is used in our method for all sequences—not a separate SVM for each pixel nor for each sequence; this is possible because we learn the classifier based on probability vectors

rather than feature vectors directly. A similar approach is introduced in [29], where the magnitude of optical flow and interframe image difference are used as features for classification. However, the 2D features have limitation in handling dynamic and multimodal scenes properly—e.g., stationary foreground and moving background. Other background subtraction methods using SVM require per-pixel (or per-block) classifiers and separate

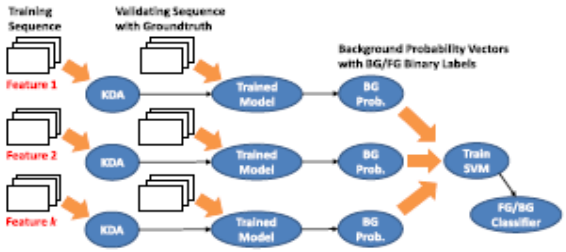


Fig. 3. Diagram for training procedure. Note that we use pixelwise density functions by KDA but train a single common SVM for all pixels and all sequences.

	R	G	B	H1	H2	H3	H4	H5	H6	dx	dy
R	1.00	0.97	0.94	0.79	0.77	0.01	0.00	0.00	0.88	0.00	0.01
G	0.84	1.00	0.98	0.80	0.78	0.00	0.01	0.00	0.90	0.00	0.01
B	0.74	0.73	1.00	0.79	0.76	0.00	0.00	0.02	0.88	0.01	0.01
H1	0.67	0.67	0.63	1.00	0.88	0.01	0.01	0.00	0.96	0.01	0.00
H2	0.66	0.66	0.63	0.84	1.00	0.00	0.00	0.01	0.95	0.01	0.00
H3	0.42	0.42	0.41	0.56	0.55	1.00	0.01	0.00	0.00	0.00	0.00
H4	0.53	0.52	0.51	0.70	0.67	0.69	1.00	0.00	0.00	0.57	0.00
H5	0.50	0.50	0.49	0.64	0.65	0.71	0.69	1.00	0.01	0.00	0.55
H6	0.61	0.61	0.59	0.82	0.83	0.53	0.65	0.61	1.00	0.01	0.00
dx	0.48	0.48	0.47	0.55	0.53	0.56	0.71	0.57	0.50	1.00	0.02
dy	0.46	0.46	0.46	0.51	0.52	0.53	0.56	0.65	0.48	0.61	1.00

Fig. 4. Correlations between features. The upper triangular matrix illustrates the correlation coefficients between feature values while the lower triangular matrix represents the correlation coefficients between background likelihoods of features.

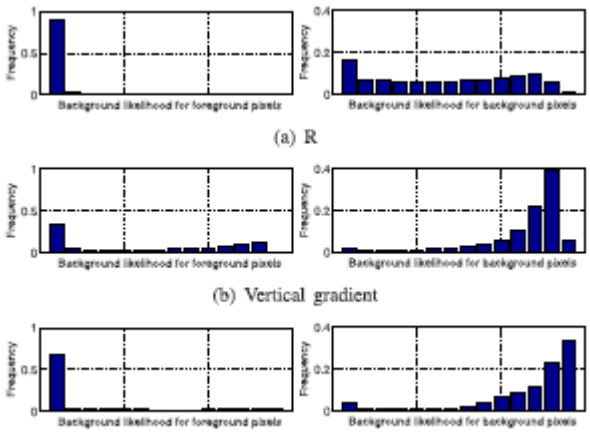


Fig. 5. Feature performance for classification. The histograms of background probabilities for foreground and background pixels are presented for each feature. Note that, in the ideal case, foreground should have low-background probabilities only and background pixels should have high-background probability only.

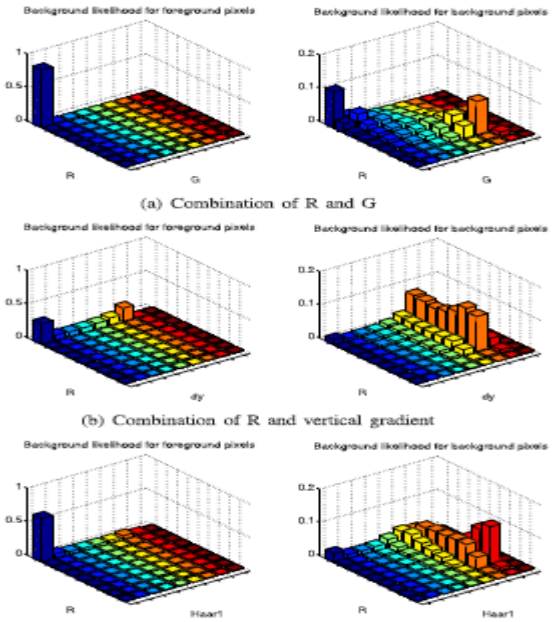


Fig. 6. Improvement of discriminativeness by adding heterogeneous features— gradient or Haar-like features. Note that the combination of R and G may not improve classification performance compared with R only. (See also Fig. 5a.) However, when the vertical gradient or the first Haar-like feature are combined with R feature, the characteristics of foreground pixels become quite different from those of background pixels.

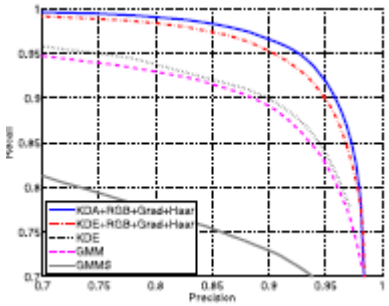


Fig. 7. PR curves for different density estimations.

4 EXPERIMENTS

We present the performance of our background modeling and subtraction algorithm using real videos. Each sequence involves challenges such as significant pixelwise noises (subway), dynamic background of a water fountain (fountain), and reflections and shadow in wide area (caviar [32]).

4.1 Feature Analysis

We describe the characteristics of individual features and the performance of multiple feature integration. Fig. 4 illustrates the correlation between every pair of features. RGB colors and three Harr-like features are significantly correlated, and the fourth and

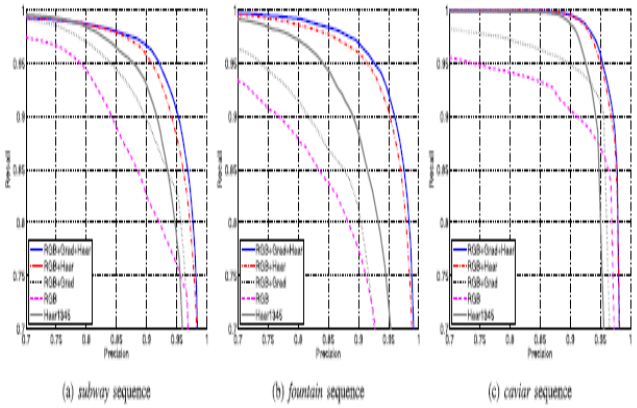
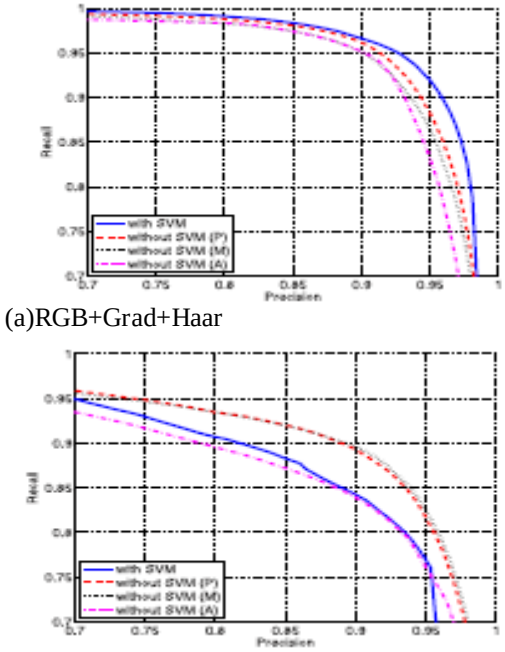


Fig. 8. PR curves of our algorithms with several different feature sets.

fifth Haar-like features have nontrivial correlation with vertical and horizontal gradient features, respectively, as one would expect. However, note that the correlations between background likelihoods of even highly correlated features are not so strong, which explains why there are still benefits of integrating highly correlated features. Fig. 5 illustrates the discriminativeness of features for foreground/background classification. The histograms of background probabilities for foreground and background pixels are presented for three different features—a representative feature for color, gradient, and Haar-like feature. According to Fig. 5, color features have substantially poor quality, compared with the other two features; background probabilities for background pixels vary significantly, which makes it difficult to classify them correctly. Also, background probabilities of gradient features for foreground pixels are widespread.

The combination of heterogeneous features improves back-ground/foreground classification performance. Recall that the R feature has mediocre quality for background subtraction. The inclusion of the G feature does not help much (Fig. 6a) because



(a)RGB+Grad+Haar

(b)RGB

Fig. 9. PR curves for testing the contribution of SVM.

background probabilities of pixels in these two feature bands are highly correlated. However, this problem is alleviated when gradient or Haar-like feature are integrated; foreground and background pixels are now more separable so that classification is more straightforward (Figs. 6b and 6c).

Evaluation of Background Subtraction

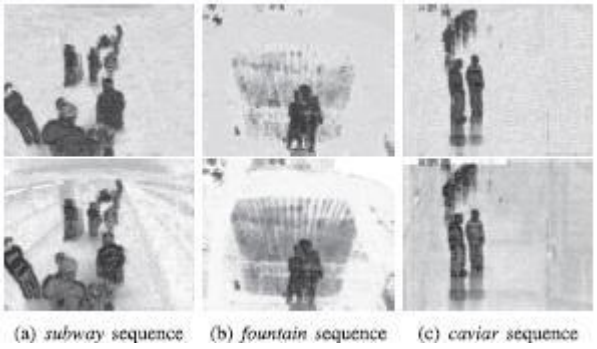
The performance of our background subtraction algorithm is evaluated in various ways. For quantitative evaluation, Precision- Recall (PR) curves are employed, where the precision and recall are defined as

$$precision = \frac{n(\mathcal{B}_T \cap \mathcal{B})}{n(\mathcal{B}_T)}, \text{ and } recall = \frac{n(\mathcal{F}_T \cap \mathcal{F})}{n(\mathcal{F}_T)},$$

We first compared our algorithm with other density-based techniques such as Gaussian Mixture Model (GMM) [5] and Kernel Density Estimation (KDE) [1] based on RGB color feature only. For the GMM method, we downloaded the code for [5] and tested two versions—with and without shadow detection. In our experiments, our algorithm with SVM (KDA+RGB+Grad+Haar) is noticeably better than KDE and GMM with/without shadow detection as shown in Fig. 7. Additionally, we implemented KDE with all 11 features and SVM (KDE+RGB+Grad+Haar) and obtained comparable accuracy to our technique, which suggests the performance of KDA with respect to KDE.[4] Note that the range of 0:9 to 0:95 in both precision and recall is critical to actual performance in background subtraction.

The performance of several different feature sets are compared and illustrated in Fig. 8. Our algorithm with all

three features (RGB+Grad+Haar) has the best performance for all three sequences. The RGBþGrad shows better results than RGB only, and the performance of RGB+Haar is comparable to RGB+Grad+ Haar; the integration of gradient feature is not very useful due to the correlation and similarity with related Haar-like features. However, the combination of the correlated features still improves the classification performance nontrivially in the subway and the fountain sequence since the SVM plays a role to select proper features implicitly. We also tested our algorithm using four independent Haar-like features based on the result in Fig. 4, where RGB features are excluded since they are significantly correlated with some of the selected Haar-like features. The accuracy based on the independent Haar-like features is signifi-cantly lower than full feature combination.



(a) *subway* sequence (b) *fountain* sequence (c) *caviar* sequence

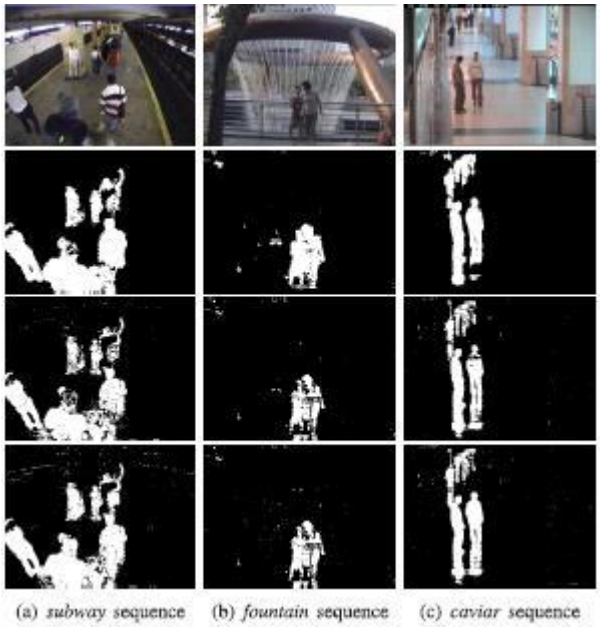
Fig. 10. Background likelihood (top) with and (bottom) without SVM. Bright pixels have higher BG likelihoods.

The importance of the SVM to foreground/background segmentation was tested, and Fig. 9 illustrates the SVM improves classification. Three other simple classifiers are employed for comparison; the product (P), minimum (M), and sum (S) of probabilities for multiple features are computed and predefined thresholds are applied for classification. SVM is particularly useful for high-dimensional data; it is probably because the high-dimensional data have more nonlinear characteristics and it is more difficult to classify such data accurately with simple methods. Note that the performance of the SVM for correlated RGB is not good as presented in Fig. 9b. Fig. 10 presents the background likelihood at each pixel, and provides a clear idea about the reason for better classification; the SVM tends to handle dynamic backgrounds and shadow better. Another advantage observed in Fig. 10 is that background likelihoods of foreground and background pixels are separated better by the SVM, and the classification is not very sensitive to threshold value. We also tested how the performance of SVM changes with different kernels, but did not observe any noticeable differences.

Qualitative results for comparison of background subtraction algorithms are presented in Fig. 11. The threshold for each

(a) *caviar2* sequence (b) *outdoor* sequence (c) *indoor* sequence

Fig. 11. Background subtraction results of three algorithms. From top to bottom, the original images and the results of our method, KDE, and GMM are presented. Note that there is a moving car on the left side of the child in the fountain sequence, which is clearly visible in our algorithm but not detected by KDE or GMM.



(a) *subway* sequence (b) *fountain* sequence (c) *caviar* sequence

Fig. 12. Background subtraction results for three different feature sets. From top to bottom, results with

RGB+Grad+Haar, RGB+Grad, and RGB are illustrated. Note that various challenges such as reflection, shadow, dynamic background, etc., are handled effectively by our algorithm with the new feature set.

algorithm is set to have 95 percent precision in classification. Fig. 11 shows our algorithm also has good qualitative performance compared with the other two. The performance of feature combinations are evaluated qualitatively and presented in Fig. 12. The combination of color, gradient, and Haar-like features outper-forms other feature sets in segmentation, especially when shadows, reflections, and dynamic backgrounds—tree branches and leaves in the rightmost area in the outdoor sequence—are involved.

CONCLUSION

We have introduced a multiple feature integration algorithm for background modeling and subtraction, where the background is modeled with a generative method and background and foreground are classified by a discriminative technique. KDA is used to represent a probability density function of the background for RGB, gradient, and Haar-like features in each pixel, where 1D independent density functions are used for simplicity. For classification, an SVM based on the probability vectors for the given feature set is employed. Our algorithm demonstrates better performance than other density-based techniques such as GMM and KDE, and the performance is tested quantitatively and qualitatively using a variety of indoor and outdoor videos.

REFERENCES

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

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

[3] B. Han, D. Comaniciu, and L. Davis, "Sequential Kernel Density Approximation through Mode Propagation: Applications to Background Modeling," Proc. Asian Conf. Computer Vision, 2004.

[4] D.S. Lee, "Effective Gaussian Mixture Learning for Video Background Subtraction," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 27, no. 5, pp. 827-832, May 2005.

[5] Z. Zivkovic and F. van der Heijden, "Efficient Adaptive Density Estimation Per Image Pixel for Task of Background Subtraction," Pattern Recognition Letters, vol. 27, no. 7, pp. 773-780, 2006.

[6] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp.511-518, 2001.

[7] B. Han, D. Comaniciu, Y. Zhu, and L.S. Davis, "Sequential Kernel Density Approximation and Its Application to Real-Time Visual Tracking," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 30, no. 7, pp. 1186-1197, July 2008.

[8] B. Han and L.S. Davis, "Adaptive Background Modeling and Subtraction: A Density-Based Approach with Multiple Features" Intelligent Video Surveillance Systems and Technology, Y. Ma and G. Qian eds., ch. 4, pp. 79-103, CRC Press, 2010.

[9] I. Haritaoglu, D. Harwood, and L.S. Davis, "W4: Real-Time Surveillance of People and Their Activities," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 809-830, Aug. 2000.

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

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

[12] N. Friedman and S. Russell, "Image Segmenation in Video Sequences: A Probabilistic Approach," Proc. 13th Conf. Uncertainty in Artificial Intelligence, 1997.

[13] A. Mittal and D. Huttenlocher, "Scene Modeling for Wide Area Surveillance and Image Synthesis," Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2000.

[14] R.M. Neal and G.E. Hinton, "A View of the EM Algorithm that Justifies Incremental, Sparse, and Other Variants," Learning in Graphical Models, M.I. Jordan, ed., pp. 355-368, Kluwer Academic, 1998.

[15] A.D. Jepson, D.J. Fleet, and T.F. El-Maraghi, "Robust Online Appearance Models for Visual Tracking," Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. I, pp. 415-422, 2001.

[16] C.E. Priebe and D.J. Marchette, "Adaptive Mixture Density Estimation," Pattern Recognition, vol. 26, no. 5, pp. 771-785, 1993.

[17] S.J. McKenna, Y. Raja, and S. Gong, "Tracking Colour Objects Using Adaptive Mixture Models," Image and Vision Computing J., vol. 17, pp. 223-229, 1999.

[18] A. Mittal and N. Paragios, "Motion-based Background Subtraction Using Adaptive Kernel Density Estimation," Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2004.

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

[20] O. Javed and M. Shah, "Tracking and Object Classification for Automated Surveillance," Proc. European Conf. Computer Vision, pp. 343-357, 2002.

[21] N. Paragios and V. Ramesh, "A MRF-Based Approach for Real-Time Subway Monitoring," Proc. IEEE CS

Conf. Computer Vision and Pattern Recognition, vol. I, pp. 1034-1040, 2001.

[22] M. Seki, T. Wada, H. Fujiwara, and K. Sumi, "Background Subtraction Based on Coocurence of Image Variations," Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2003.

[23] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh, "Background Modeling and Subtraction of Dynamic Scenes," Proc. Ninth IEEE Int'l Conf. Computer Vision, 2003.

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

[25] M. Heikkila¨ and M. Pietika¨inen, "A Texture-Based Method for Modeling the Background and Detecting Moving Objects," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 28, no. 4, pp. 657-662, Apr. 2006.

[26] L. Li, W. Huang, I. Gu, and Q. Tian, "Statistical Modeling of Complex Backgrounds for Foreground Object Detection," IEEE Trans. Image Proces-sing, vol. 13, no. 11, pp. 1459-1472, Nov. 2004.

[27] T. Parag, A. Elgammal, and A. Mittal, "A Framework for Feature Selection for Background Subtraction," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 1916-1923, 2006.

[28] R.E. Schapire, "The Strength of Weak Learnability," Machine Learning, vol. 5, no. 2, pp. 197-227, 1990.

[29] H.-H. Lin, T.-L. Liu, and J.-H. Chuang, "Learning a Scene Background Model via Classification," IEEE Trans. Signal Processing, vol. 57, no. 5, pp.1641-1654, May 2009.