An Integrated Background Model for Video Surveillance Based on Primal Sketch and 3D Scene Geometry

Wenze HU[1,3], Haifeng GONG[1,2], Song-Chun ZHU[1,2] and Yontian WANG [3]
[1]Lotus Hill Institute, Ezhou, China , [2]Department of Statistics, UCLA
[3]School of Computer Science, Beijing Institute of Technology, Beijing, China
wzhu@lotushill.org, {hfgong, sczhu}@stat.ucla.edu, wyt@bit.edu.cn

Abstract

This paper presents a novel integrated background model for video surveillance. Our model uses a primal sketch representation for image appearance and 3D scene geometry to capture the ground plane and major surfaces in the scene. The primal sketch model divides the background image into three types of regions — flat, sketchable and textured. The three types of regions are modeled respectively by mixture of Gaussians, image primitives and LBP histograms. We calibrate the camera and recover important planes such as ground, horizontal surfaces, walls, stairs in the 3D scene, and use geometric information to predict the sizes and locations of foreground blobs to further reduce false alarms. Compared with the state-of-the-art background modeling methods, our approach is more effective, especially for indoor scenes where shadows, highlights and reflections of moving objects and camera exposure adjusting usually cause problems. Experiment results demonstrate that our approach improves the performance of background/foreground separation at pixel level, and the integrated video surveillance system at the object and trajectory level.

1. Introduction

Background modeling is a very important component in video surveillance[27] and remains a bottleneck for system performance, especially for indoor scenes, where shadows, highlights and reflections of moving objects on marble ground and glasses and camera gain adjusting can cause problems[4, 6, 11]. Recent years the literature on background modeling can be roughly classified into two categories, pixel based and block based model.

Pixel based models include raw pixel based and color space transformation based methods. A probability distribution is used to model intensity or color space transformed pixel. The distribution may be Gaussian, Mixture of Gaussians or non-parametric model. Single Gaussian distribution was used in [24] to model each pixel in video sequence. The mean and variance are calculated either by standard maximum likelihood offline estimation or updated recursively by using a simple adaptive filter. The single Gaussian cannot tolerate repetitive motions like trees, water, camera vibration, rain and snow, etc. By using more than one Gaussian distribution per pixel, it improves the performance of backgrounds. Friedman and Russell [3] introduced the mixture of Gaussians approach for a traffic surveillance application. Stauffer and Grimson [19] used an online K-means approximation to update the parameters of the mixture model, which becomes one of the most commonly used approaches and have been improved or extended by many authors[13, 28]. To allow complex distribution of each background pixel, many researchers proposed to use non-parametric models, for example, nonparametric kernel density estimation [1] and quantization/clustering technique [14].

Block based models mainly include features in independent or slightly overlapped blocks, e.g., block-wised edge histogram [17], combination of edge and intensity information[10, 12]. Heikkilä and Pietikäinen [6] proposed an approach that uses the local binary pattern (LBP) operators to capture background statistics. LBP operator can tolerate illumination changes and has shown excellent performance in many applications. Compared with previous approaches, this approach has many advantages and improvements, but it is relatively computation demanding. LBP was also used by Helmut and Horst[7] together with other two types of features, Haar-like features and HOG and combined into an on-line feature selection framework, called On-line Boosting.

Besides modeling the image intensity, there are some other methods considering information other than pixel or block, e.g., inter-frame optical flow [23], segmentation [12] and high level feedback [20].

Researchers in computer and human vision both realized that context provides rich information about an object's identity, location and size. In fact, the structure of many surveillance scenes is governed by strong configurational priors[21]. Therefore, detecting object in 3D geometry context becomes a hot topic in recent literature. Hoiem et al [8] provided a framework for modeling the interdependence of objects, surface orientations, and camera viewpoint by placing local object detection in the context of the overall 3D scene. They used probabilistic object hypotheses to refine geometry and vice-versa. Besides, many authors proposed methods to estimate rough context information from low level features, e.g., rough surface orientation estimation [9], typical 3D scene geometries classification [18], and absolute depth estimation by structure recognition [22], etc.

In this paper, we use primal sketch for 2D image appearance modeling and scene geometry for 3D context. The primal sketch model divides the background into three types of regions — flat, sketchable and textured, according to a primal sketch representation. The three types of regions are modeled respectively by Mixture of Gaissuans, image primitives and LBP histograms. Additionally, we calibrate the camera and recover the major 3D surfaces, such as walls, of the scene, and use the geometry information to further predict blob sizes at different locations and reduce false alarms. Compared with the state-of-the-art background model, our approach has the following contributions: 1) We use a model integrating three types of models, each of which needs different amount of computation resources and targets different areas. 2) To our best knowledge, our work is the first to introduce 3D geometric context to improve background/foreground separation.

2. Primal Sketch for Image Appearance

2.1. Information Scaling and Primal Sketch

Objects and image structures can appear at a wide range of distances or scales, and the same structure appearing at different distances or scales produces different images with different statistical properties. Wu et al [25] showed that the entropy rate of the image data and the perceptual uncertainty changes over the viewing distance (as well as the camera resolution). They called these changes information scaling. Information scaling triggers transitions of statistical models as Fig. 1 illustrated. Based on the information scaling principle, they proposed a full-zoom primal sketch model[5] that integrates both sparse coding and Markov random fields. In this model, local image intensity patterns are classified into ”sketchable regime” and ”non-sketchable regime” by a sketchability criterion.

According to the above analysis, image patches with different scale have different properties. The original work of Wu et al[25] decomposed the image lattice into 2 two parts. But we prefer to dividing it into 3 parts. The cartoon region is mainly composed of bars, edges, L-junction and other primitives. The textured region can be best represented by texture descriptors such as LBP[6]. The flat region can be best represented by the region color. Fig. 2 shows an indoor scene as well as partitions of it divided image by primal sketch. For efficiency consideration, we used a reduced set of primitives instead of the full collection, which are shown in Fig 3.

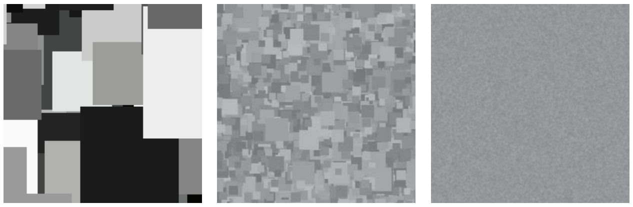
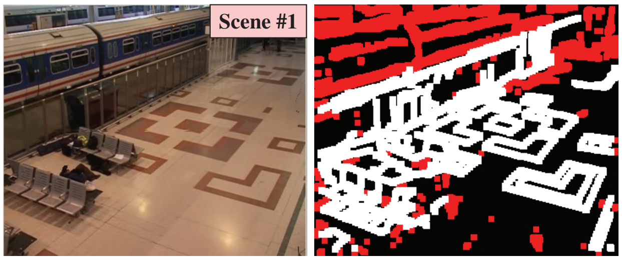

scale 0, cartoon scale 3, texture scale 6, flat

Figure 1. Images of simulated ivy wall taken at 8 scales, each scale is obtained by 2×2 pixel average of the previous scale, three of them are shown. (a) At scale 0, the image consists of a number of large squares of uniform intensity and can be deterministically represented by a relatively small number of local geometric shapes such as edges, corners, etc. (b) At scale 3, the effective representation has to be some sort of simple histograms coding the image intensities statistically. (c) At scale 6, the image is approaching the flat area and can be best described by a single Gaussian.



(a) Background Image (b) Partitioned Background

Figure 2. Background image and primal sketch partition. (a) The background images of two running examples. (b) The primal sketch partition of the image. Flat region are shown in black, sketchable in white and textured in red. Top row, scene #1; bottom row, scene #2.

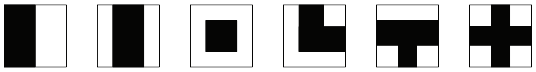


Figure 3. Some image primitives for sketchable areas. Image primitives are composed of $+1$'s and -1's and allowed to rotate in 8 directions.

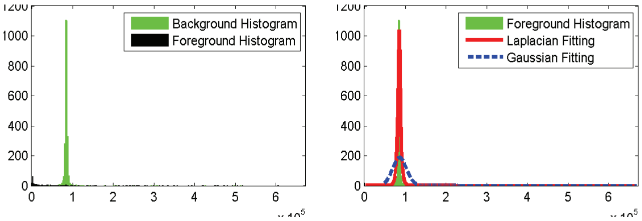
Let Λ be the area of image I, we divide it into 3 parts $\Lambda = \Lambda_{\text{fl}} \cup \Lambda_{\text{sk}} \cup \Lambda_{\text{txt}}$ where Λ_{fl}, Λ_{sk} and Λ_{txt} are lattices occupied by flat, sketchable and textured regions respectively. Each region is further decomposed into primitive patches

S_{sk}, textured patches S_{txt} and flat patches S_{fl}. We solve all the patches $S = (S_{\text{fl}}, S_{\text{sk}}, S_{\text{txt}})$ from background image I by maximizing the following posterior probability [5] $p(S|I) = p(S_{\text{sk}}|I)p(S_{\text{fl}}|S_{\text{sk}}, I)p(S_{\text{txt}}|S_{\text{sk}}, I)$.

2.2. The Sketch Algorithm

To achieve real-time performance, we simplify the original sketching algorithm[5]. Our algorithm consists of two steps, initialization and refinement. For initialization, we first apply canny edge detection and edge linking to place primitives such as step edges, bars and junctions, then we get an initial sketchable part. After that we divide the remaining part of the lattice into flat region and textured region by a criterion of intensity variance, if the variance is above a threshold, the patch is textured, otherwise, it is flat. This is a pursuit process, we find the textured patch one by one with variance drop down to a threshold. During the initialization, the primitives with relatively low edge intensity, the flat patches with relatively high variance and the textured patches with relatively low variance are marked as borderline patches. The borderline patches in initial solution are refined by maximize posterior probability, $p(S|I)$ [5] to enforce local smoothness in the partition.

2.3. Probability Model of Background



(a) (b)

Figure 4. Histogram of primitive responses in sketchable region Λ_{sk} for background and foreground. The foreground distribution shown in dark approximately follows uniform distribution and the background distribution approximately follows Laplacian distribution.

We further decomposed the sketchable areas into primitive patches $\Lambda_{\text{sk}} = \bigcup_{i}^{K_{\text{sk}}} \Lambda_{\text{sk},i}$, where each patch $\Lambda_{\text{sk},i}$ at position $\vec{x}$ and time t is represented by a primitive $B_{\vec{x}}$.To tolerate the disturbance of camera jitter, we allow the primitve to shift in 2 pixels, i.e., we gather the response with largest background probability in a 10×10 patch as effective response. The response of a primitive at location $\vec{x}$ and time t is

$$r_{\vec{x},t} = \max_{\vec{x}' \in \partial \vec{x}} B_{\vec{x}'} * I_t. \tag{1}$$

where $\partial \vec{x}$ denote the neighborhood of $\vec{x}$.

In Fig. 4, we randomly select one patch on sketchable region from scene in Fig. 2 and plot the histograms of model responses $r_{\vec{x},t}$ on foreground and background. From the histogram, one can see that the background distribution is more peak than Gaussian and can be roughly approximated by Laplacian distribution. Therefore, for sketchable region, we model the responses as a Laplacian distribution.

$$p(r) \quad = \quad \frac{1}{2b} \exp\left\{ -\frac{|r - \mu|}{b} \right\} \tag{2}$$

where μ and b are two model parameters. The initial parameters are estimated from the first N images of the sequence

$$\hat{\mu} = \frac{1}{N} \sum_i r_i, \; \hat{b} = \frac{1}{N} \sum_i |r_i - \hat{\mu}|. \tag{3}$$

where N is the number of samples and r_i the i-th sample. After initialization, we update the parameters frame by frame, and the on-line updating rules are

$$\hat{\mu}^{t+1} \quad = \quad (1 - \alpha)\hat{\mu}^t + \alpha r_{t+1} \tag{4}$$

$$\hat{b}^{t+1} \quad = \quad (1 - \alpha)\hat{b}^t + \alpha |r_{t+1} - \hat{\mu}^{t+1}|. \tag{5}$$

where α is the updating ratio and r_t the response of the primitive at frame t.

For flat region, the intensity follows Mixtures of Gaussian (GMM) distribution, and the model parameters are estimated as GMM standard. GMM can also be substituted by some shadow suppress variants [13].

For textured region, a modified LBP is used to reduce the computation. For each pixel in textured region, a LBP descriptor is the histogram of several LBP operators over a neighborhood. LBP operator $v(\vec{x})$ is defined as a binary vector of $v_i(\vec{x}) = \text{sign}\,[I(\Delta\vec{x}_i + \vec{x}) - I(\vec{x})]$, where $\Delta\vec{x}_i$ is the offset of the i-th neighbor pixel. And the LBP descriptor h is defined as the histogram of v over a neighborhood of $\vec{x}$, for the j-th bin, $h_j(x) = \sum_{\vec{x}' \in \partial \vec{x}} \delta\left[v(\vec{x}') = j\right]$. The similarity between two histogram h and h' is defined as $d(h, h') = \sum_i \min(h_i, h'_i)$. During the modeling, a running mean and running variance of the model is maintained. The mean m_t is updated as $m_t = (1 - \alpha_m)m_{t-1} + \alpha_m h_t$ where h_t is the current histogram and α_m is the updating rate. Variance σ_t is maintained like $\sigma_t = (1 - \alpha_\sigma)\sigma_{t-1} + \alpha_\sigma d(h_t, m_t)$. where α_σ is the updating rate of variance. Our final decision is made by applying threshold on $d(h_t, m_t)/\sigma_t$, which is a simplified verion of the original.

3. 3D Scene Geometric Context

In this paper, geometric context refers to two aspects, 1) camera calibration and ground plane estimation, 2) 3D estimation of major surfaces in the scene. As mentioned in Section 1, previous work mostly gave a rough estimation of the camera geometry and the scene, e.g., Hoiem et al [8] assumes that camera tilt angle is under 15 degree and the rotation angle is zero, but in surveillance scenario, this does not always hold. Instead, we designed an interactive system for camera parameter estimation and important surface annotation, which calculates camera parameter matrix and surface locations in 3D space. Though the results are not as accurate as cube calibration and direct measuring of surfaces, it is quite convenient and accurate enough for predicting object scale and location in the scene.

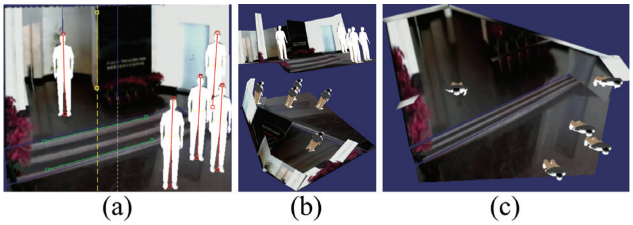


Figure 5. Calibration and surface projection. (a) the calibration interface, user drags the vertical structure, horizontal structures to revise the camera parameters. (b) Surface annotation. (c) Surface projection, from top view, invisible regions are shown in gray.

Our interactive calibration system use the calibration algorithm proposed in [15, 16], with revision to allow user to interactively modifying the parameters by dragging horizon line, vertical line, vanishing point etc (see Fig. 5 (a)). With the interaction, we can ensure that the parameters are obtained as accurate as possible and avoid the problems caused by computational stability such as overflow failure. After the calibration, the user is requested to annotate the important surfaces in the scene which can be automated in future work(see Fig. 5 (b)). Because the homography between parallel plane and the image plane is also known, we can easily determine the location of all the major surfaces in 3D space (see Fig. 5 (c)).

In surveillance applications, there are typically four types of interested targets – human, vehicle, bicycle, and baggage. For each type of targets, we have very strong prior about their physical sizes and possible positions. For example, an adult pedestrian cannot present above the ground in a substantial time period, without help of other support surfaces. With calibration and ground assumption, we can estimate the expected physical size of each foreground blob in the image. To make the blob size well defined in the image plane, we model each type of object with a cuboid and make the following assumptions:

Human can touch the ground, parallel and stair surface with each side, but can only touch vertical surface with front or left side. Vehicle and bicyle can touch the ground, parallel and stair surface with bottom side.

Under the above assumptions, with prior dimension of each type of targets, positions of surfaces and calibration information given, we can easily infer the possible blob size and orientation at each image pixel by projecting the cuboids attached to the surface to the image plane and count the area of the convex hull. For pedestrians, the size threshold is selected as $1m \times 0.2m$ to ensure low miss rate and allow other postures. Though it may not be compatible with some extreme postures, it nearly misses no object in the experiments. Fig. 6 gives an illustration of the location-scale constraints. For example, we can predict the size of vertical human as following. Let B and C be the head and feet of the human in image, A be the intersection of human and the horizon in the image plane, D be the intersection of human and the baseline of the wall, and V_Y be the vertical vanishing point. Besides, let h denote the human height and H_c the camera height (See Fig. 6). If wall is not present, the human height BC is constrained by the following equation (Fig. 6 (c), simply following the cross ratio theorem)

$$\frac{BC}{BA} / \frac{V_Y C}{V_Y A} = \frac{h}{h - H_c} \tag{6}$$

On the other hand, if a wall is present in the middle, a real person will not be occluded by the wall, but might be climbing the wall, then the equation becomes (Fig. 6 (d), simply applying the cross ratio theorem twice)

$$\frac{BC}{BA} / \frac{V_Y C}{V_Y A} = \frac{h}{h - H_f} \tag{7}$$

$$\frac{AD}{AC} / \frac{V_Y D}{V_Y C} = \frac{H_c}{H_f} \tag{8}$$

where H_f is the camera height calculated from feet of the human and can be eliminated from above equations.

In implementation, the minimal size of blob at each location and each orientation can be computed and stored at initialization stage (as shown in Fig. 6), so nearly no burden is added for on-line tracking.

4. Implementation

During the initialization, we average the first several frames to obtain a background image. Then we apply primal sketch algorithm on it to get three types of regions and estimate parameters of each patch according to Section 2. Additionally, the minimal blob size at each location for each orientation is computed according to previous section and stored for further use.

For each type of region, we use the corresponding criterion for judgement of foreground. After the patch level processing, the size constraints are applied to remove blobs whose size are below the minimal acceptable size at the location.

To allow long staying objects turn into background and allow background objects move away, we update the primal sketch model periodically. The parameters of LBP are updated according to [6], the GMM parameters are updated

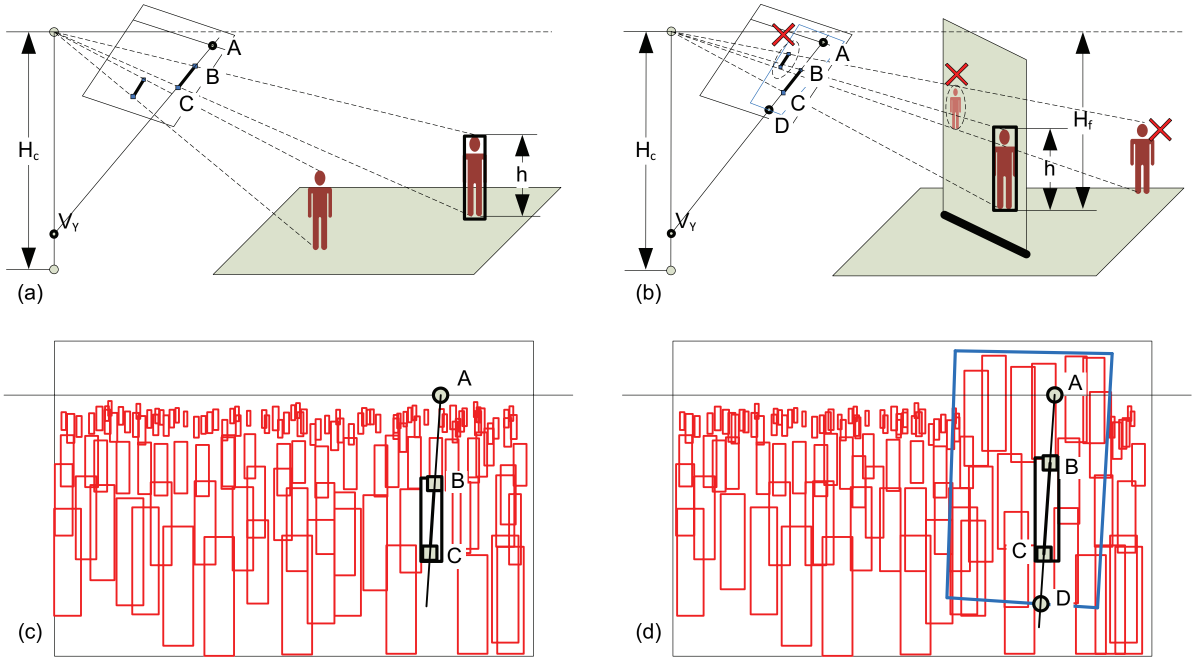
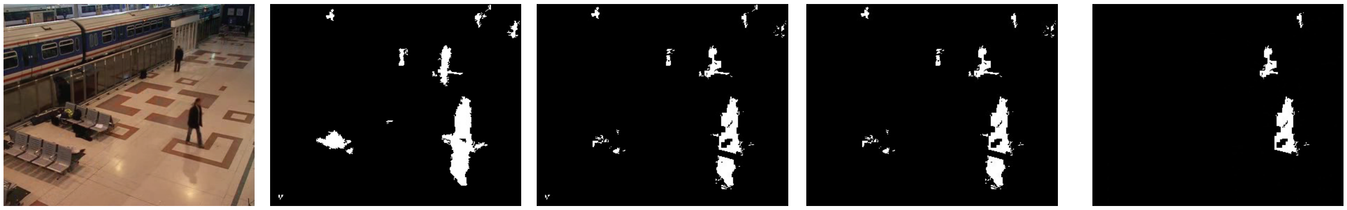


Figure 6. Blob scale constraints imposed by surfaces and calibration. (a) If only ground plane is present, the size constraints can be easily estimated by homography between imaging plane and ground plane. (b) If vertical surfaces present, using only ground plane and imaging plane homograph will result in seeing object behind the wall, which should be removed as false alarm. (c) Predicted blobs without vertical surface, BC is a pedestrian blob corresponds to BC in (a). (d) On vertical plane, homograph between vertical plane and imaging plane is used to suppress false alarms. Blob BC is a pedestrian corresponds to BC in (a). The expected pedestrians in the wall area are larger within the rectangular window in (d).



(a) Original image (b) GMM (c) GMM+Primitive (d) GMM+Primitive+LBP (e) GMM+Primitve+LBP+3D

Figure 7. Comparison of mask generated by different combinations. (a) Original image, (b) Pure GMM produces lots of false alarms, (c) GMM+Primitive, the false alarms in the sketchable region are removed by primitives, (d) GMM+Primitive+LBP, the false alarms in the textured region are removed by LBP, (e) GMM+Primitve+LBP+3D, the remain difficult false alarms are further suppressed by 3D location and size constraints.

in the standard way, and the primitves' parameters are updated according to Equation 5. Besides, we also change the overall configuration of the primal sketch, i.e., re-align each types of patches. To reduce computation, the configuration is updated every 1 minute, not per frame. The updating method is as following: We maintain a long-term running mean of the sequence, $\bar{m}^{t+1} = (1-\beta)\bar{m}^t + \beta I_t,$, where I_t is the image at time t, β is the updating rate. Besides, we maintain another pool of recent frames $\{I_j\}$, each one in which is extracted from each second in recent 1 minute. Using the recent pool and running mean as an image collection, we can run primal sketch just like initialization, with only difference that the running mean is more important than each single frame in recent pool and thus receives a weight. To avoid the time-consuming edge linking, we initialize current configuration by recent configuration with changed patch revision. The changed patches are detected by input $I_{\text{mean},t}$ to previous model. During the refinement, we only use changed patches and their neighbors as proposal, which can highly reduce computation.

We have developed a surveillance system based on the proposed background modeling method. In the system, the background model is followed by a foreground segmentation module and tracking module. Further details cannot be covered in this paper.

5. Experiments

We use the four scenes of PETS 2006 ([2]) and two scenes of LHI data set[26] (see Fig. 8). PET2006 has 7 sequences for each scene. We use one of them for each scene. The sequences of PETS 2006 are classified into 5 subjective difficulty levels for left-luggage detection. For background modeling, we believe their difficulties are similar, so we choose the sequences with the highest subjective difficulties for our test.

5.1. Pixel level validation

To validate the effectiveness of the three types of image models on the corresponding regions, we gather some of the foreground masks generated by different algorithms (shown in Fig. 7). From the foreground, one can see that when camera adjusting the gain, or heavy reflection and shadow present, Mixture of Gaussians (GMM) produces many false alarms. After we add primitives at sketchable areas, most of the false alarms in sketchable areas are removed, though there are meanwhile a few missing patches, which can be easily compensated by tracking. After we add LBP at textured areas, more false alarms are removed. If 3D information is further introduced, nearly all false alarms are removed.

To show the pixel level performance of the model in quantity, we give the ROCs of the foreground detection results of each type of model. We compare GMM and the proposed method on sketchable area Λ_{sk} and textured area $\Lambda_{\mathtt{txt}}$ separately. We simplify the cross comparisons of different models on different type of regions, because of the following reasons: 1) Heikkilä and Pietikäinen [6] et al admitted that LBP may not work well on flat area, 2) we realized that LBP and primitive are similar for edge area, but primitive can be computed much faster, 3) it is obvious that primitives will work poor on textured area.

The results are shown in Fig. 9. From the comparison of GMM and primitives on sketchable region, one can see that at given hit rate, say 0.8, primitives can reduce false alarm in great magnitude. From the comparison of GMM and LBP on textured region, one can see that LBP outperforms GMM more than primitives on sketchable, but the absolutely number is less than primitives, because areas of textured region are relatively small and less objects presented there.

5.2. Blob level comparison

In the surveillance system, the foreground masks are usually passed to subsequent modules in terms of blobs. The subsequent modules split, merge or discard blobs. So the performance counted in blobs is one of the most important measures for background modeling. We compare the numbers of false alarm blobs and hit blobs of our algorithm and GMM, optionally using 3D location and size constraints. False alarms is defined as a detected foreground blob not covering any of the true foreground blob or there intersection below 70% of the detected area. True positive is defined as 70% area of a ground truth blob covered by detected foreground blob. Experiments are conducted on 6 scenes, besides the 2 running examples, others are shown in Fig. 8. The results are shown in Fig. 10. Results show that our algorithm outperforms GMM and 3D provide very useful information for false alarm supression. In Fig. 10 (a), (c), (d) and (f), the proposed method outmatched GMM in large magnitue, just because the scenes are more structural. In Fig. 10 (e), all algorithms are similar, because the scene is quite crowded. In crowded area, objects occupy longer than foreground, while for other regions, relatively less objects appear. In Fig. 10 (a) and (f), walls occupied large area and many shadows are there, so 3D helps to emerge large improvement there.

5.3. Trajectory level performance evaluation

Because in a surveillance system, tracking and recognition algorithms can further suppress false positives and negatives, we integrate three of the combination into our surveillance system - IntMon, to test the overall performance. For the surveillance system, we evaluate the performance of detection and tracking. The false alarm is defined as a reported trajectory in which 20% frames are false alarm blobs or 20% frames are not in correct correspondences. The true positive is defined as a ground truth tracjectory in which 80% frames are covered by a reported trajectory. Results (Fig. 11) show that the proposed method improves the system in term of FN+FP in all selected experiments. The improvements lie in three levels. 1) For Scene #2 and #6, the improvements are very remarkable, which agrees with blob level results. 2) Scene #1, #3 and #4 have improvements in certain degree, because tracking algorithms can further remove false alarms, the improvements may not as obvious as blob level. 3) The proposed method obtains minor improvement on Scene #5, which agrees with blob level results.

6. Conclusion

We present a primal sketch model for representation of background images. The background is divided into three types of regions — flat, sketchable and textured region according to a primal sketch representation. The three types of regions are modeled respectively by Mixture of Gaussians, image primitives and LBP histograms. For sketchable and textured region, primitives and LBP can obtain better results than GMM, and for flat and sketchable region, GMM and primitives can reduce computation than LBP. Additionally,

Figure 8. Snaps of additional scenes in the experiments.

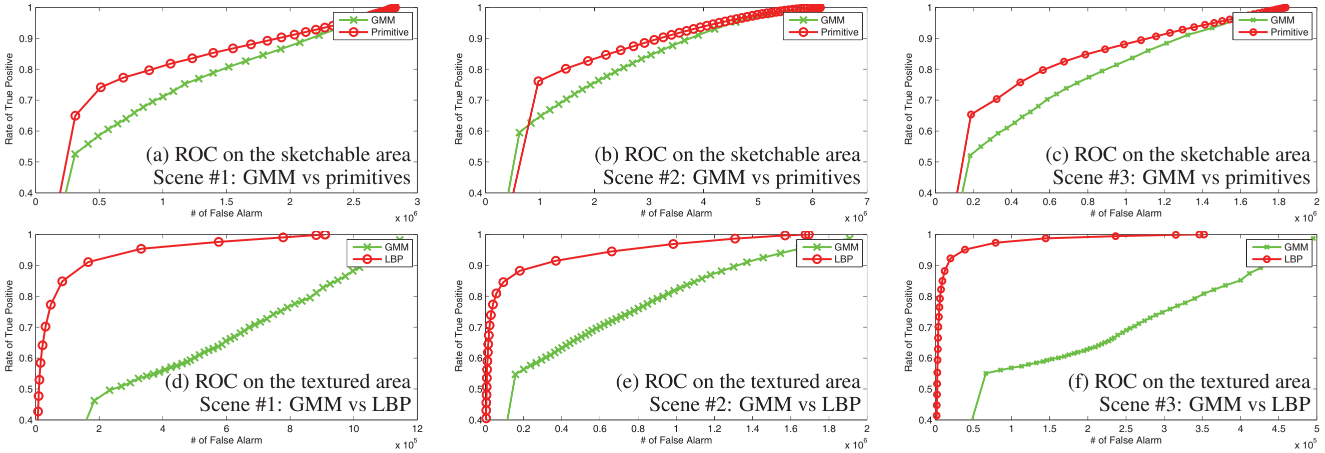


Figure 9. Pixel level comparison. Because sketchable area are larger than textured, the quantitive level of the number of false alarms is larger. Textured is more difficult for GMM, so on textured are, LBP sees large improvement.

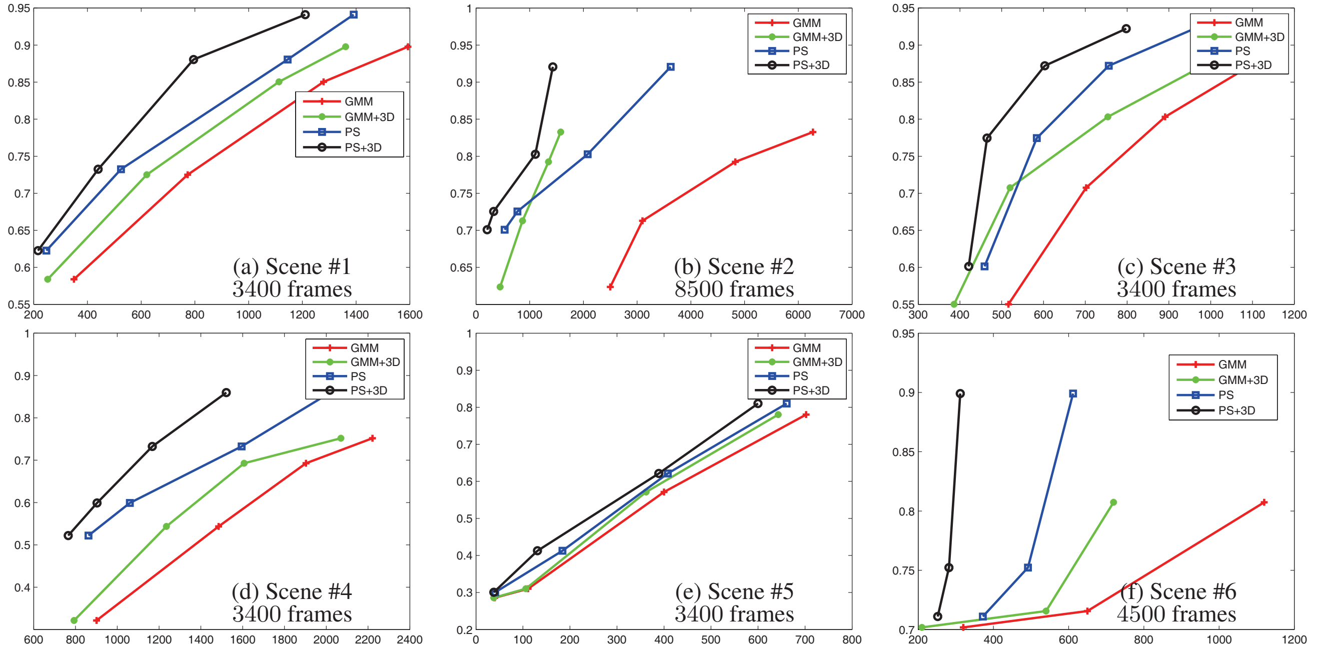


Figure 10. Blob level performance comparison. Horizontal axis — number of false alarm blobs, vertical axis — true positive rate.

we introduced 3D geometry context to improve background model. We calibrate the camera and recover the 3D vertical surfaces, such as walls, of the scene, and use the geometry information to further reduce false alarms.

Experiments are carried out at pixel, blob and trajectory levels. The results demonstrate that our approach improves

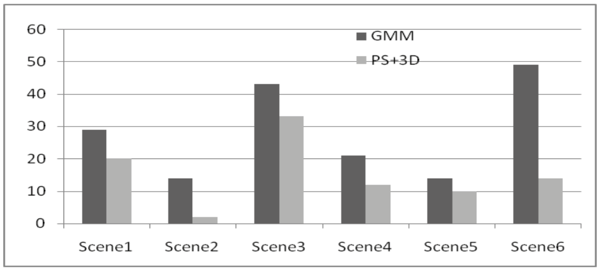


Figure 11. Trajectory level performance comparison, vertical bars show total number of FN and FP. The GMM performance produces more errors than primal sketch(PS) and 3D geometric context (3D).

the performance of both the background modeling component and the integrated video surveillance system in object detection and classification.

Acknowledge

This work is supported by Hi-Tech Research And Development Program Of China (2006AA01Z121, 2006AA01Z339, 2007AA01Z340).

References

[1] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proc. IEEE*, 90(7):1151–1163, 2002. 1

[2] J. M. Ferryman, editor. *Proc. Ninth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2006)*, 2006. 6

[3] N. Friedman and S. Russell. Image segmentation in video sequences: A probabilistic approach. In *UAI*, pages 175–181, 1997. 1

[4] S. Greenhill, S. Venkatesh, and G. West. Adaptive model for foreground extraction in adverse lighting conditions. In *PRICAI: Trends in Artificial Intelligence*, pages 805–811, 2004. 1

[5] C.-E. Guo, S.-C. Zhu, and Y. N. Wu. Primal sketch: Integrating texture and structure. *CVIU*, 106(1):5–19, 2007. 2, 3

[6] M. Heikkilä and M. Pietikäinen. A texture-based method for modeling the background and detecting moving objects. *PAMI*, 28(4):657–662, 2006. 1, 2, 4, 6

[7] G. Helmut and B. Horst. On-line boosting and vision. In *CVPR*, pages 260–267, 2006. 1

[8] D. Hoiem, A. A. Efros, and M. Hebert. Putting objects in perspective. In *CVPR*, pages 2137 – 2144, June 2006. 2, 3

[9] D. Hoiem, A. A. Efros, and M. Hebert. Recovering surface layout from an image. *IJCV*, 75(1), October 2007. 2

[10] S. Jabri, Z. Duric, H. Wechsler, and A. Rosenfeld. Detection and location of people in video images using adaptive fusion of color and edge information. In *ICPR*, 2000. 1

[11] O. Javed, Z. Rasheed, O. Alatas, and M. Shah. KnightM: A real time surveillance system for multiple overlapping and non-overlapping cameras. In *ICME*, 2003. 1

[12] O. Javed, K. Shafique, and M. Shah. A hierarchical approach to robust background subtraction using color and gradient information. In *IEEE Workshop on Motion and Computing*, 2002. 1

[13] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *Proc. European Workshop Advanced Video Based Surveillance Systems*, 2001. 1, 3

[14] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis. Background modeling and subtraction by codebook construction. In *ICIP*, 2004. 1

[15] N. Krahnstoever and PauloR.S.Mendonca. Bayesian autocalibration for surveillance. In *ICCV*, 2005. 4

[16] F. Lv, T. Zhao, and R. Nevatia. Camera calibration from video of a walking human. *PAMI*, 28(9):1513–1518, 2006. 4

[17] M. Mason and Z. Duric. Using histograms to detect and track objects in color video. In *Proc. Applied Imagery Pattern Recognition Workshop*, pages 154–159, 2001. 1

[18] V. Nedović, A. W. Smeulders, A. Redert, and J.-M. Geusebroek. Depth information by stage classification. In *ICCV*, 2007. 2

[19] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, pages 246–252, 1999. 1

[20] L. Taycher, J. W. F. III, and T. Darrell. Incorporating object tracking feedback into background maintenance framework. In *IEEE Workshop on Motion and Video Computing*, pages 120–125, 2005. 1

[21] A. Torralba. Contextual priming for object detection. *IJCV*, 53(2):169–191, 2003. 2

[22] A. Torralba and A. Oliva. Depth estimation from image structure. *PAMI*, 24(9):1–13, 2002. 2

[23] L. Wixson. Detecting salient motion by accumulating directionally consistent flow. *PAMI*, 22(8):774–780, 2000. 1

[24] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):780–785, July 1997. 5. 1

[25] Y. N. Wu, S.-C. Zhu, and C. en Guo. From information scaling of natural images to regimes of statistical models. *Quarterly of Applied Mathematics*, 2007 (To appear). 2

[26] Z. Yao, X. Yang, and S. C. Zhu. Introduction to a large scale general purpose groundtruth database: methodology, annotation tools, and benchmarks. In *6th International Conference on EMMCVPR*, 2007. 6

[27] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4), 2006. 1

[28] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *ICPR*, 2004. 1