Detection of change in video based on local pattern and photometric features

K L Chan

Department of Electronic Engineering, City University of Hong Kong
83 Tat Chee Avenue
Hong Kong, China

itklchan@cityu.edu.hk

ABSTRACT

The segmentation of moving objects in video can be formulated as a background subtraction problem – the detection of change in each image frame. The background scene is learned and modeled. A pixelwise process is employed to determine whether the current pixel is similar or not to the background model. The detection of change in video is challenging due to the non-stationary background such as illumination change, background motions, etc. We propose new features for background modeling. Perception-based local ternary patterns are generated from the same color channels as well as from different color channels. Features computed from the local patterns are stored in the background model as samples. If the current pixel is classified as background, the background model is updated. Finally, we propose a probabilistic refinement to improve each change region by taking into account the spatially consistency of image features. We compare our method with various background subtraction algorithms on some video datasets. Our method can achieve 13% better performance than other methods.

Keywords

Background modeling, Change detection, Local ternary pattern, Cross-channel pattern, Probabilistic foreground refinement

1. INTRODUCTION

Moving objects such as humans or vehicles, are often the focus of image sequence analysis. The segmentation of moving objects can be formulated as change detection. One common approach is to perform background subtraction. In that sense, the background scene is modeled. Change region is detected when it is found to be different from the background model. Sobral and Vacavant [Sob14a] presented a review and evaluation of 29 background subtraction methods. Background subtraction techniques can be categorized based on the features being used to model the background scene.

Statistical – Stauffer and Grimson [Sta00a] proposed modeling of background colors using mixture of Gaussian distributions (MoG). In contrast with a fixed number of Gaussians in the original MoG model, Zivkovic [Ziv04a] proposed an algorithm for selecting the number of Gaussian distributions using the Dirichlet prior. Bouwmans [Bou11a] presented a survey on statistical background modeling.

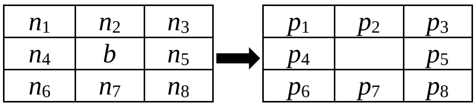
Bag of visual words – Intensities or colors are sampled over a short image sequence. Elgammal *et al.* [Elg02a] proposed an algorithm for estimating the pdf directly from previous pixels using kernel estimator. Kim *et al.* [Kim05a] proposed to represent the background by codebooks which contain quantized background colors. Barnich and Van Droogenbroeck [Bar09a] proposed a sample-based background subtraction algorithm called ViBe. Background model is initialized by randomly sampling of pixels on the first image frame. Hofmann *et al.* [Hof12a] proposed a similar non-parametric sample-based background subtraction method with 9 tunable parameters.

Pattern – Recent research showed that modeling background by local patterns can achieve higher accuracy. Heikkilä and Pietikäinen [Hei06a] proposed to model the background of a pixel by local binary pattern (LBP) histograms estimated around that pixel. Liao *et al.* [Lia10a] proposed the scale invariant local ternary pattern (SILTP) which can tackle illumination variations. St-Charles *et al.* [Stc15a] proposed a pixelwise background modeling method using local binary similarity pattern (LBSP) estimated in the spatio-temporal domain. Ma and Sang [Ma12a] proposed the multi-channel SILTP (MC-SILTP), which is an improvement of SILTP, with pattern computed from RGB color channels. In [Cha16a], we proposed to model the background by perception-based local binary pattern.

2. SAME-CHANNEL AND CROSS-CHANNEL LOCAL TERNARY PATTERNS

We propose novel perception-based local ternary patterns which can be used effectively to characterize various dynamic circumstances in the scene. At each image pixel, patterns can be generated from the same color channels and different color channels. Figure 1 shows a block of 3 x 3 pixels. Each pixel of the block, n1 to n8, (except the center pixel) is compared with the confidence interval (*CI*) of the center pixel *b*. *CI*(*b*) is defined by (CI_l, CI_u) where CI_l and CI_u are the lower bound and upper bound of *CI* respectively. The pattern value *p* is set according to the following equation

$$p_k = \begin{cases} 00, CI_l \leq n_k \leq CI_u \\ 01, \quad n_k > CI_u \\ 10, \quad n_k < CI_l \end{cases}, 1 \leq k \leq 8 \qquad (1)$$



n_1	n_2	n_3
n_4	b	n_5
n_6	n_7	n_8

p_1	p_2	p_3
p_4		p_5
p_6	p_7	p_8

Figure 1. A block of pixels with center pixel *b* and neighbors n_1 to n_8 is transformed into ternary pattern.

The confidence interval *CI*(*b*) can be defined as (*b* – d_1, *b* + d_2). According to Weber's law [Gon10a], d_1 and d_2 depend on the perceptual characteristics of *b*. That is, they should be small for darker color and large for brighter color. Haque and Murshed [Haq13a] derived the linear relationship $d_1 = d_2 = c * b$, where *c* is a constant. We adopt the human visual perception characteristics in transforming pixel colors into local ternary pattern. *CI*(*b*) is defined as (*b* – c_1b, *b* + c_2b). Using peak signal-to-noise ratio (PSNR) measure, *b* and *b* – c_1b are just perceptually different from each other if

$$20\log_{10}\frac{I_{\max}}{b - c_1 b} - 20\log_{10}\frac{I_{\max}}{b} = T_p \qquad (2)$$

where $I_{\max}$ is the maximum intensity and T_p is the perceptual threshold. Similarly, *b* and *b* + c_2b are just perceptually different from each other if

$$20\log_{10}\frac{I_{\max}}{b} - 20\log_{10}\frac{I_{\max}}{b + c_2 b} = T_p \qquad (3)$$

To determine c_1 and c_2, the equations are simplified.

$$c_1 = \frac{10^{\frac{T_p}{20}} - 1}{10^{\frac{T_p}{20}}} \qquad (4)$$

$$c_2 = 10^{\frac{T_p}{20}} - 1 \qquad (5)$$

Assume T_p is 0.5 dB, c_1 = 0.0559 and c_2 = 0.0593.

If the center pixel b and neighbors n_1 to n_8 are from the same color channel, the Same-Channel Pattern (*SCP*) is generated as follows:

$$SCP_k = \oplus\{p_k^c\}, c = \{R, G, B\}, 1 \leq k \leq 8 \qquad (6)$$

where p_k^c is the binary pattern value for color channel *c* at position *k*, and ⊕ is the concatenation of the corresponding binary pattern values for all color channels. We choose the RGB color model instead of other color model such as YIQ because that avoids more computation in transforming the pixel values. If the center pixel b and neighbors n_1 to n_8 are from different color channels, the Cross-Channel Pattern (*CCP*) is generated as follows:

$$CCP_k = \oplus\{p_k^{c_b:c_n}\},$$
$$c_b: c_n = \{R{:}B, G{:}R, B{:}G\}, 1 \leq k \leq 8 \qquad (7)$$

where $p_k^{c_b:c_n}$ is the binary pattern value at position *k* estimated with center pixel *b* from color channel c_b and neighbor n_k from color channel c_n, and ⊕ is the concatenation of the corresponding cross-channel binary pattern values. One advantages of our local ternary patterns is that they are estimated from all color channels which can provide a more informative characterization of local image texture. Also, in flat image regions, the features derived from gray values or the same color channel may not be distinctive. This limitation can be alleviated with the use of *CCP*.

3. BACKGROUND MODELING AND CHANGE DETECTION

A number of image frames in each video are allocated for background model initialization. With this short image sequence, the local patterns are transformed into concise representation and stored as background samples. Also, the original color values are saved as photometric features in the background model.

First, *SCP* and *CCP* are combined into a 12-bit string:

$$CP_k = SCP_k \oplus CCP_k, 1 \leq k \leq 8 \qquad (8)$$

For convenient of storage, *CP*$_k$ is transformed into a single value:

$$F_k = \sum_{p=1}^{12} CP_k(p) \cdot 2^p \,, 1 \leq k \leq 8 \qquad (9)$$

where $CP_k(p)$ is bit *p* of CP_k. Therefore, at each image pixel location, the local ternary patterns are transformed into an 8-dimensional feature vector $\{F_k\}$ and saved in the background model as one sample.

Change regions are detected by comparing each pixel of the image frame with the background model. It is a background/foreground segregation process. If features of the pixel match with the background model, it is a background pixel. Otherwise, it is a foreground (change) pixel. We adopted the sample-based approach. At each pixel of the current image frame, *SCP* and *CCP* are computed using the method as described in section 2. If the local ternary patterns and photometric features match with one sample of the background model, the current pixel is classified as background pixel. The similarity between patterns of the current pixel and the background model can be computed by measuring the Hamming distance between two bit strings

$$d_p = \sum_{k=1}^{8} \left| CP_k^i \otimes CP_k^B \right| \qquad (10)$$

where CP_k^i is CP_k of the current pixel, CP_k^B is CP_k of a background sample, ⊗ is the XOR operator, |·| is the cardinality. The two sets of local ternary patterns are considered as similar if $d_p < \varepsilon_p$.

The similarity between the current pixel color and the color of the background sample is computed by

$$s_c = \frac{C^i \cdot C^B}{\|C^i\|_2 \cdot \|C^B\|_2} \qquad (11)$$

where C^i is color the current pixel, C^B is color of a background sample, $\|\cdot\|_2$ denotes the Euclidean length of a vector. The two colors are considered as similar if $s_c > \varepsilon_c$.

The background model is updated alongside with the change detection. If the current pixel is classified as background, the features of the matched background sample will be replaced by the features of the current pixel.

4. FOREGROUND REFINEMENT

The background/foreground segregation result may contain false positive and false negative errors. For instance, isolated scene pixels may have features deviate from the background model due to illumination change or background motion. As they are not connected to form a region, they can be discarded without affecting the detection of real moving objects. Therefore, foreground regions less than 15 pixels are eliminated. The remaining foreground regions may have holes. The silhouette of the change region may be distorted. These false negative errors are usually caused by the similarity of the image features in the change region to the background model. We analyze the spatially consistency of image features and refine the change region probabilistically. Let *x* be a foreground (*FG*) pixel. Its neighboring background (*BG*) pixels *y* are defined by

$$y \mid dist(x, y) < D,\ x = FG,\ y = BG \qquad (12)$$

where *dist*(·) is the city-block distance and *D* is fixed as 1. *y* are changed to *FG* when they have image features more similar to neighboring *FG* pixels than neighboring *BG* pixels. To analyze the local ternary pattern feature

$$y_i = FG \quad \text{if } log\frac{P(y_i = FG)}{P(y_i = BG)} > T_{f1} \qquad (13)$$

$$P(y_i = FG) = exp\left(-\sum_j \left| d_p^{y_j} - d_p^{y_i} \right|\right), dist(y_i, y_j) < D, y_j = FG \qquad (14)$$

$$P(y_i = BG) = exp\left(-\sum_j \left| \mu(d_p^{y_j}) - d_p^{y_i} \right|\right), dist(y_i, y_j) < D, y_j = FG \qquad (15)$$

where *μ*(·) is the mean of the local ternary pattern features in the background model. To analyze the photometric feature

$$y_i = FG \quad \text{if } log\frac{P(y_i = BG)}{P(y_i = FG)} > T_{f2} \qquad (16)$$

$$P(y_i = FG) = exp\left(-\sum_j \left| s_c^{y_j} - s_c^{y_i} \right|\right), dist(y_i, y_j) < D, y_j = FG \qquad (17)$$

$$P(y_i = BG) = exp\left(-\sum_j \left| \mu(s_c^{y_j}) - s_c^{y_i} \right|\right), dist(y_i, y_j) < D, y_j = FG \qquad (18)$$

where *μ*(·) is the mean of the photometric features in the background model. A false negative pixel will be corrected when both Equations (13) and (16) are satisfied.

5. RESULTS AND DISCUSSION

We evaluated the performance quantitatively in terms of F-Measure (F1). We compared our method with other background subtraction algorithms on two publicly available datasets. We selected sample-based method ViBe [Bar09a], pattern-based methods SILTP [Lia10a] and MC-SILTP [Ma12a] for comparison. Based on sample consensus, ViBe can achieve very good results with very few tunable parameters. ViBe uses RGB color model and a fixed spherical distance of 30 in matching new pixel with background samples. It keeps 20 background samples and the new pixel is identified as background with 2 matches. SILTP employs scale invariant local patterns. MC-SILTP is one latest pattern-based method and can perform better than SILTP. We implemented SILTP with the same set of parameters as reported in [Lia10a]. The only parameter value which was not mentioned is the number of training frames. Through experimentation, we find that the number of training frames is best fixed as 150. Similarly, we implemented MC-SILTP with the same setting as reported in [Ma12a]. As for our method, the first 50 image frames of the video are used for background model initialization. Other parameters are: $\varepsilon_p = 16$, $\varepsilon_c = 0.9$, $T_{f1}= 2.0$, $T_{f2}= 0.1$.

The Wallflower dataset [Toy99a] contains 6 videos. Each video comes with 1 manually labeled ground truth. The image frame size is 160 x 120 pixels. The dataset contains videos exhibiting gradual illumination change (TimeOfDay), sudden illumination change (LightSwitch), similar background and object color (Camouflage), moving background elements (Waving Trees), etc. Table 1 shows the F1 results of our method, ViBe, SILTP and MC-SILTP. The best result in a given row is highlighted. No method can achieve the highest F1 on all videos. Our method can achieve highest F1 on 5 videos. Overall, texture-based methods perform better than ViBe. Our method achieves the highest average F1 which is 13% higher than the second best method MC-SILTP. Also, our method can achieve consistently high F1 as indicated by the lowest variance.

Figure 2 shows the visual results. In “Bootstrap”, humans already exist in the initialization image sequence. ViBe produces more false negative errors. Our method and SILTP relatively have lesser false negative errors. Our method also has lesser false positive errors than MC-SILTP. In “Camouflage”, the difficulty is that the monitor and the clothing have similar color. Therefore, ViBe, SILTP and MC-SILTP produce many false negative errors. With probabilistic refinement, our method can drastically reduce false negative error. In “ForegroundAperture”, the human remains stationary and stooped over the desk for some time. Features of the human are included in the background model. When the human rises, all methods produce false negative errors. In “LightSwitch”, ViBe cannot adapt to the sudden change of light. Other methods can quickly respond. In “TimeOfDay”, the room is very dark at the beginning. The light is turned on gradually and a human enters the room. SILTP and MC-SILTP cannot adapt to the change and result in large amount of false positive errors. ViBe performs better but the detected human is small. Benefit by the local ternary pattern features, our method can detect a larger human. In “WavingTrees”, ViBe and SILTP produce many false positive errors in the trees behind the human. MC-SILTP still produce moderate amount of false positive error. Our method is quite effective in identifying the waving trees as background. In summary, our method can achieve a consistent and accurate performance under various kinds of complication in the background scene.

Sequence	**Our method**	**ViBe**	**SILTP**	**MC-SILTP**
Bootstrap	**0.846**	0.478	0.766	0.740
Camouflage	**0.966**	0.931	0.927	0.896
ForegroundAperture	0.768	0.644	**0.849**	0.665
LightSwitch	**0.759**	0.159	0.730	0.745
TimeOfDay	**0.678**	0.394	0.175	0.181
WavingTrees	**0.965**	0.933	0.712	0.946
Average	**0.830**	0.590	0.693	0.695
Variance	**0.014**	0.095	0.071	0.075

Table 1. F1 results on the Wallflower dataset

The Star dataset [Li04a] contains more challenging videos. Each video comes with 20 manually labeled frames as ground truths. The videos have different image frame size, from 160 x 120 pixels to 320 x 256 pixels. We selected ViBe [Bar09a], MC-SILTP [Ma12a], statistical method MoG [Sta00a], and pixelwise LBP (LBP-P) [Hei06a] for comparison. Table 2 shows the F1 results. The numeric results of MoG and LBP-P are from [Lia10a]. Our method can achieve highest F1 on 6 videos and second best on 2 videos. Overall, our method achieves the highest average F1 than all comparing methods which is 5% higher than the second best method LBP-P. Also, our method has the lowest variance.

Figure 3 shows the visual results of our method, ViBe and MC-SILTP. Some videos contain busy human flows (AirportHall, Bootstrap, Escalator, ShoppingMall). “Curtain” has a slowly moving curtain in the background. “Fountain” and

"WaterSurface" contain moving water. In "Lobby", the light is dimmed and turned on later. "Trees" has waving trees and banner in the background. In "AirportHall", all methods produce false negative errors. ViBe and MC-SILTP have more false positive errors than our method. In "Bootstrap" and "Curtain", ViBe produces more false negative errors while MC-SILTP produces more false positive errors. Our method can detect a fairly good shape of the humans. "Escalator" is a difficult video. All methods many false positive and negative errors. In "Fountain", ViBe cannot detect the humans completely. MC-SILTP produces many false positive errors. Our method can effectively model the fountain as background and detect the humans. In "ShoppingMall", the main difficulty is the shadow. That causes more false positive errors in ViBe. Pattern-based methods can tackle shadow much better as can be seen in the shape of the humans detected by our method. In "Lobby", ViBe and MC-SILTP cannot adapt to the dimming of light and produce many false positive errors. Our local ternary pattern features have no problem in characterizing the illumination change. ViBe and MC-SILTP produce large amount of false positive errors in "Trees". ViBe also cannot detect the bus completely. Our method can effectively treats the waving trees as background and window of the bus as change region. In "WaterSurface", ViBe fails to detect the legs. Pattern-based method can model the water surface. However, due to similarity between clothing and water, they produce many false negative errors (holes) within the change region.

Sequence	**Our method**	**ViBe**	**MC-SILTP**	**MoG**	**LBP-P**
AirportHall	0.653	0.496	**0.659**	0.579	0.503
Bootstrap	**0.725**	0.514	0.649	0.541	0.520
Curtain	**0.794**	0.775	0.707	0.505	0.714
Escalator	**0.566**	0.445	0.439	0.366	0.539
Fountain	**0.801**	0.425	0.504	0.779	0.753
ShoppingMall	0.648	0.522	0.513	**0.670**	0.629
Lobby	**0.708**	0.029	0.690	0.684	0.523
Trees	**0.611**	0.345	0.222	0.554	0.606
WaterSurface	0.612	0.801	0.570	0.635	**0.822**
Average	**0.680**	0.483	0.550	0.590	0.623
Variance	**0.007**	0.052	0.024	0.014	0.013

Table 2. F1 results on the Star dataset

6. CONCLUSION

We propose a method for change detection in video. The background model is represented by samples of local ternary pattern and photometric features. We propose new local ternary patterns which are generated from the same color channels as well as from different color channels. The local ternary patterns make full use of all color channels. The features derived from the patterns are more informative and distinctive than other pattern-based methods that use gray values or the same color channel. In the change detection process, a current pixel is classified as background only when both pattern and photometric features match with one background sample. Otherwise, that pixel is classified as foreground (change region). Features of background pixel will be used to update the background model. Finally, we propose a probabilistic refinement to improve each change region by taking into account the spatially consistency of image features. We compare our method with various background modeling algorithms on two video datasets. Our method can achieve better and more consistent performance than all other methods.

7. REFERENCES

[Bar09a] Barnich, O., and Van Droogenbroeck, M. ViBe: a powerful random technique to estimate the background in video sequences. Proceedings of International Conference on Acoustics, Speech and Signal Processing pp.945-948, 2009.

[Bou11a] Bouwmans, T. Recent advanced statistical background modeling for foreground detection: a systematic survey. Recent Patents on Computer Science Vol. 4, No. 3, pp.147-176, 2011.

[Cha16a] Chan, K.L. Background modeling using perception-based local pattern. Proceedings of 24th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision WSCG 2016, pp.253-260, 2016.

[Elg02a] Elgammal, A., Duraiswami, R., Harwood, D., and Davis, L.S. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. Proceedings of IEEE Vol. 90, No. 7, pp.1151-1163, 2002.

[Gon10a] Gonzalez, R.C., and Woods, R.E. Digital Image Processing. Pearson/Prentice Hall 2010.

[Haq13a] Haque, M., and Murshed, M. Perception-inspired background subtraction. IEEE Transactions on Circuits and Systems for Video Technology Vol. 23, No. 12, pp.2127-2140, 2013.

[Hei06a] Heikkilä, M., and Pietikäinen, M. A texture-based method for modeling the background and detecting moving objects. IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 28, No. 4, pp.657-662, 2006.

[Hof12a] Hofmann, M., Tiefenbacher, P., and Rigoll, G. Background segmentation with feedback: the Pixel-Based Adaptive Segmenter. Proceedings of IEEE Workshop on Change Detection at IEEE Conference on Computer Vision and Pattern Recognition pp.38-43, 2012.

[Kim05a] Kim, K., Chalidabhongse, T.H., Harwood, D., and Davis, L.S. Real-time foreground-background segmentation using codebook model. Real-Time Imaging Vol. 11, pp.172-185, 2005.

[Li04a] Li, L., Huang, W., Gu, I.Y.-H., and Tian, Q. Statistical modelling of complex backgrounds for foreground object detection. IEEE Transactions on Image Processing Vol. 13, No. 11, pp.1459-1472, 2004.

[Lia10a] Liao, S., Zhao, G., Kellokumpu, V., Pietikäinen, M., and Li, S.Z. Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition pp.1301-1306, 2010.

[Ma12a] Ma, F., and Sang, N. Background subtraction based on multi-channel SILTP. Proceedings of Asian Conference on Computer Vision pp.73-84, 2012.

[Sob14a] Sobral, A., and Vacavant, A. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. Computer Vision and Image Understanding Vol. 122, pp.4-21, 2014.

[Sta00a] Stauffer, C., and Grimson, W.E.L. Learning patterns of activity using real-time tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 22, No. 8, pp.747-757, 2000.

[Stc15a] St-Charles, P.-L., Bilodeau, G.-A., and Bergevin, R. SuBSENSE: a universal change detection method with local adaptive sensitivity. IEEE Transactions on Image Processing Vol. 24, No. 1, pp.359-373, 2015.

[Toy99a] Toyama, K., Krumm, J., Brumitt, B., and Meyers, B. Wallflower: principles and practice of background maintenance. Proceedings of International Conference on Computer Vision pp.255-261, 1999.

[Ziv04a] Zivkovic, Z. Improved adaptive Gaussian mixture model for background subtraction. Proceedings of International Conference on Pattern Recognition pp.28-31, 2004.

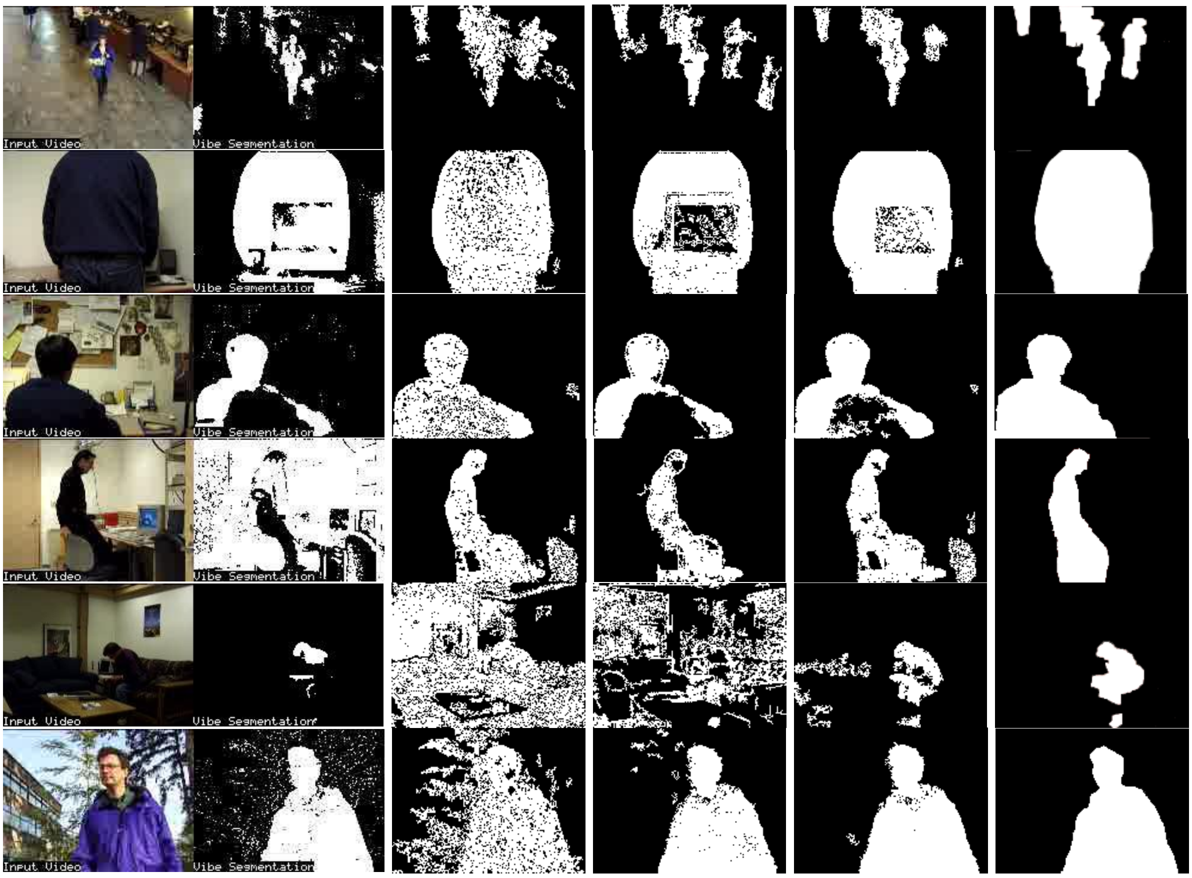


Figure 2. Background subtraction results on the Wallflower dataset (6 image sequences) – original image frames (first column), results obtained by ViBe (second column), results obtained by SILTP (third column), results obtained by MC-SILTP (fourth column), results obtained by our method (fifth column), ground truths (last column).

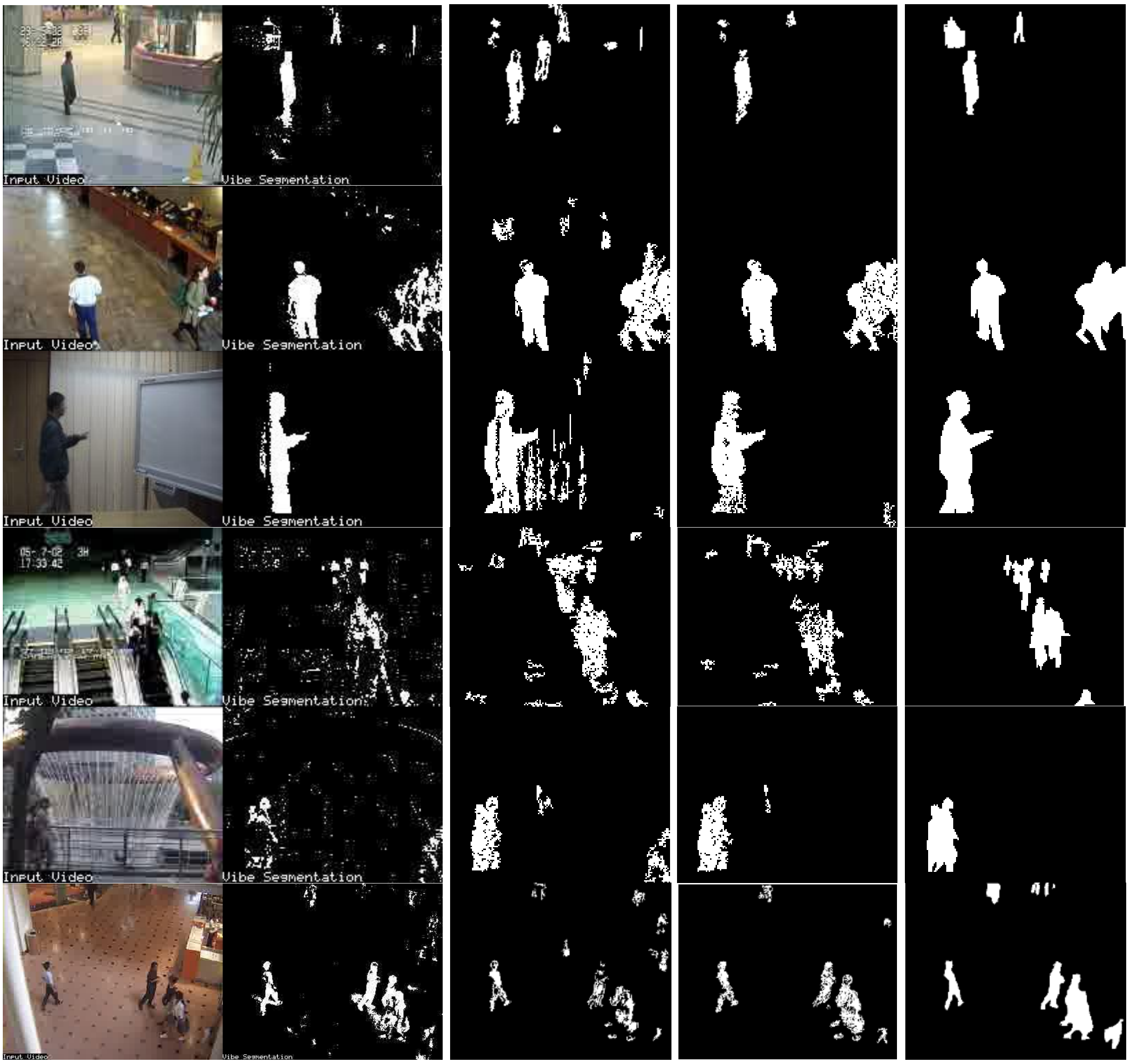


Figure 3. Background subtraction results on the Star dataset (9 image sequences) – original image frames (first column), results obtained by ViBe (second column), results obtained by MC-SILTP (third column), results obtained by our method (fourth column), ground truths (last column).

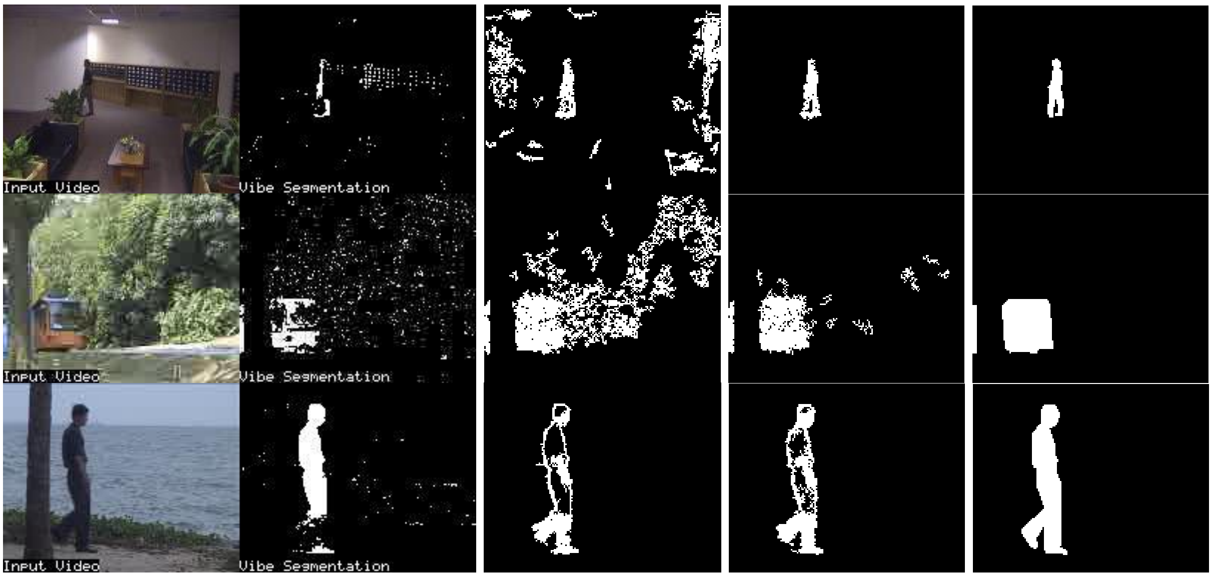


Figure 3. continue.