Modeling Pixel Process with Scale Invariant Local Patterns for Background Subtraction in Complex Scenes

Shengcai Liao[1], Guoying Zhao[2], Vili Kellokumpu[2], Matti Pietikäinen[2], and Stan Z. Li[*] [1]
[1] Center for Biometrics and Security Research & National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
[2] Machine Vision Group, University of Oulu, Finland
{scliao,szli}@cbsr.ia.ac.cn {gyzhao,kello,mkp}@ee.oulu.fi

Abstract

Background modeling plays an important role in video surveillance, yet in complex scenes it is still a challenging problem. Among many difficulties, problems caused by illumination variations and dynamic backgrounds are the key aspects. In this work, we develop an efficient background subtraction framework to tackle these problems. First, we propose a scale invariant local ternary pattern operator, and show that it is effective for handling illumination variations, especially for moving soft shadows. Second, we propose a pattern kernel density estimation technique to effectively model the probability distribution of local patterns in the pixel process, which utilizes only one single LBP-like pattern instead of histogram as feature. Third, we develop multimodal background models with the above techniques and a multiscale fusion scheme for handling complex dynamic backgrounds. Exhaustive experimental evaluations on complex scenes show that the proposed method is fast and effective, achieving more than 10% improvement in accuracy compared over existing state-of-the-art algorithms.

1. Introduction

Background subtraction is often the first task in stationary video processing applications. Output of the background subtraction is usually an input to a higher level process, such as object categorization, tracking or action recognition. Therefore, its performance can have a huge effect on the performance of higher level tasks.

Background subtraction is a challenging task, especially in complex dynamic scenes that can contain moving vegetation, rippling water, etc. Many approaches have been presented in the literature to deal with such multimodal scenes [1, 2, 4, 8, 10]. The popular idea is to model temporal samples in multimodal distributions, in either parametric or nonparametric way, so that the learned background model is able to tolerate the variations of the background scene. Another major challenge in background subtraction is illumination variation. A number of existing works handle illumination variation such as moving shadows in special color spaces [1, 7], but the learned parameters are not adaptable. In [10], a global illumination change detection was proposed, and a better background model could be selected accordingly. However, there still exists the problem of having suitable and up-to-date models for all cases. A survey on moving shadow detection methods is presented in [7].

In this work, we propose an efficient background subtraction framework that deals with illumination variations on the feature level, in which neither global illumination change detection nor local shadow detection is used. Our work is motivated by that of Heikkilä *et al.*'s [3], who used Local Binary Pattern (LBP) histograms for background modeling. LBP features are tolerable against illumination variations. However, they found that moving shadows could not be handled very well. In this work, we propose to extend LBP to a scale invariant local ternary pattern (SILTP)[1] operator, and show its effectiveness for handling illumination variations. Furthermore, we aim at accurate segmentation of moving objects and improve Heikkilä *et al.*'s region histogram based work by modeling pixel process with a single local pattern instead. However, local patterns are not ordinal numerical values, thus can not be modeled directly into traditional density functions. Therefore we propose a Pattern Kernel Density Estimation (PKDE) technique to effectively model probability distributions of such patterns. In addition, we develop multimodal background models with the above techniques for handling complex dynamic backgrounds, and we also develop a multiscale fusion scheme to consider the spatial scale information.

[*] Stan Z. Li is the corresponding author.

[1] The term "scale" here means gray scale pixel value, not spacial scale.

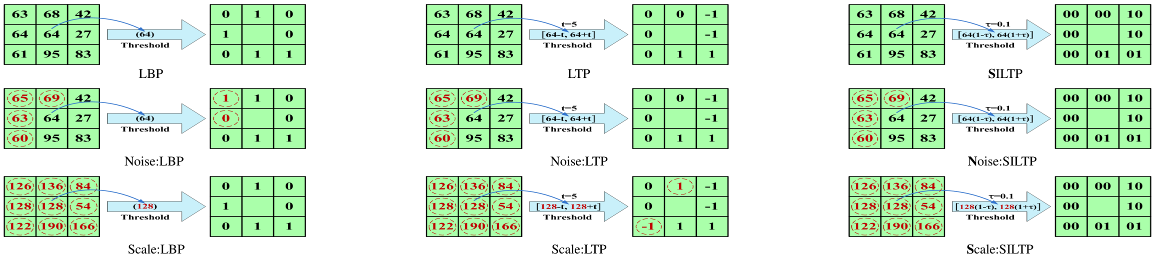

Figure 1. Comparison of LBP, LTP, and SILTP operators. First row: original encodings. Second row: encodings with noises. Third row: encodings with scale transform (all pixel values are doubled). The circled red pixels are changed with noises or by scale transform, and the circled red encodings are affected by those changes correspondingly.

2. Scale Invariant Local Ternary Pattern

Local Binary Pattern (LBP) is proved to be a powerful local image descriptor [6]. The encoding is shown in the top-left of Fig.1, which is monotonically invariant to gray scale transforms. However, the LBP operator is not robust to local image noises when neighboring pixels are similar, as illustrated in the second row of Fig.1 for an example. To deal with local noises, Tan and Triggs proposed a Local Ternary Pattern (LTP) operator [9] for face recognition. As can be seen from the second row of Fig.1, LTP is more robust by introducing a small tolerative range. However, the descriptor is extended from LBP by simply adding a small offset value for comparison, which is not invariant under scale transform of intensity values by a multiplying constant. An example is shown in the last row of Fig.1, where the LTP descriptor can not keep its invariance against scale transform when all local pixel values are multiplied by 2.

The intensity scale invariant property of a local comparison operator is very important, because illumination variations, either global or local, often cause sudden changes of gray scale intensities of neighboring pixels simultaneously, which would approximately be a scale transform with a constant factor. Therefore, in this work we propose a Scale Invariant Local Ternary Pattern (SILTP) operator, of which the concept is shown in the last column of Fig.1. Given any pixel location (x_c, y_c), SILTP encodes it as

$$SILTP^{\tau}_{N,R}(x_c, y_c) = \bigoplus_{k=0}^{N-1} s_\tau(I_c, I_k), \tag{1}$$

where I_c is the gray intensity value of the center pixel, I_k are that of its N neighborhood pixels equally spaced on a circle of radius R, $\bigoplus$ denotes concatenation operator of binary strings, τ is a scale factor indicating the comparing range, and s_τ is a piecewise function defined as

$$s_\tau(I_c, I_k) = \begin{cases} 01, & if\ I_k > (1+\tau)I_c, \\ 10, & if\ I_k < (1-\tau)I_c, \\ 00, & otherwise. \end{cases} \tag{2}$$

Since each comparison can result in one of three values, SILTP encodes it with two bits (with “11” undefined). The scale invariance of SILTP operator can be easily verified.

The advantage of SILTP operator lies in three folds. First, it is computationally efficient, which causes only one more comparison than LBP for each neighbor. Second, by introducing a tolerative range like LTP, the SILTP operator is robust to local image noises within a range. Especially in the shadowed area, the region is darker and contains more noises, in which SILTP is tolerable while local comparison results of LBP would be affected more (*cf.* Fig.2). Finally, the scale invariance property makes SILTP robust to illumination changes. The SILTP feature is invariant if the illumination is suddenly changed from darker to brighter or vice versa. Besides, SILTP is robust when a soft shadow covers a background region, because the soft cast shadow reserves the background texture information but tends to be darker than the local background region with a scale factor.

Fig.2 shows an example of comparing the three operators on two real video frames for foreground/background segmentation. As can be seen from the histograms, for background with and without shadows, the SILTP operator performs perfectly, with only two patterns being different between the two black blocks, while both LBP and LTP show larger difference. Meanwhile, for the white blocks it can also be seen that SILTP has a better power to distinguish the foreground block from that of the background.

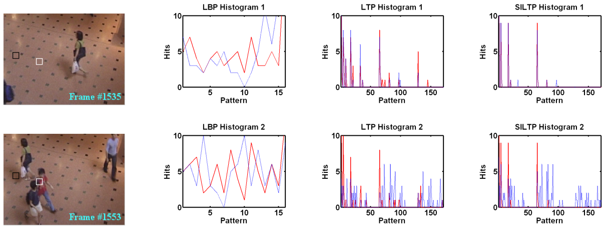

Figure 2. Comparison of $\text{LBP}_{4,1}$, $\text{LTP}^{5}_{4,1}$, and $\text{SILTP}^{0.1}_{4,1}$ operators on real frames. First column: two frames from the ShoppingMall dataset [4], with two 10×10 blocks drawn. Black blocks contain backgrounds with and without shadows. White one is background on the top, and foreground on the bottom. Other columns: histograms of the three local patterns corresponding to black blocks (top row) and white ones (bottom row), and to top frame (solid red) and bottom one (dotted blue).

3. Kernel Density Estimation of Local Patterns

To segment the foreground/background precisely, we consider to model the texture background in a pixelwise manner. Given a monocular grayscale video sequence, the *pixel process* [8] with the local pattern observations over time $1, 2, \ldots, t$ at a pixel location (x_0, y_0) is defined as

$$\{p_1, p_2, \ldots, p_t\} = \{F(x_0, y_0, i) : 1 \leq i \leq t\}, \quad (3)$$

where F is the texture image sequence. Fig.3 shows an example of six pixel processes with LBP, LTP, and SILTP descriptors at two locations respectively. It can be seen that all background patterns within a pixel process, either unitary or dynamic, are distributed at just several possible bins. Especially for SILTP at the entrance, the pattern is invariant (20 or 00010100) among all the 500 frames counted. Therefore, it is promising to model the complex unitary/dynamic background pixelwisely with local texture descriptors.

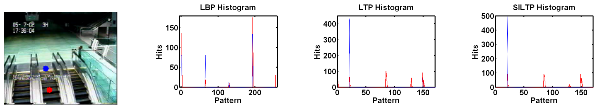

Figure 3. Histograms of six pixel processes with $\text{LBP}_{8,1}$, $\text{LTP}^{5}_{4,1}$, and $\text{SILTP}^{0.1}_{4,1}$ operators at two locations respectively. Red point is dynamic background on a moving escalator, and blue one is unitary background at the entrance, with solid red line and blue dotted one showing the corresponding histograms (500 frames counted).

To model the background we need to estimate the probability distribution of the pixel process with local texture descriptors. It is never been done before because local texture patterns are not numerical values that have partial ordering relationships. As a result, traditional numerical value based methods, either parametric one like Gaussian mixture models [8], or nonparametric one like kernel density estimation (KDE) [1], can not be used directly for modeling local patterns into background. Instead most existing researches calculate region histograms of local patterns as numerical features. The background subtraction using block-based LBP histograms is of this kind [3], which loses some localization information, and generates a huge feature set.

In this work, we develop a pattern kernel density estimation (PKDE) technique with a particular local pattern kernel that is suitable for descriptors like LBP, LTP, and SILTP. First we define a distance function $d(p, q)$ as the number of different bits between two local patterns p and q, which can be computed fast via XOR operation. For SILTP defined in Equ.(1), such a distance is also well defined, because for each comparison there are 2-bits differences between 01 and 10 encodings, and both of them have 1-bit difference with 00. Then we derive our local pattern kernel as $\Phi(p, q) = g(d(p, q))$, where g is a weighting function that can typically be a Gaussian. Afterwards, given a pixel process of local patterns (see Equ.(3)), the probability density function can be estimated smoothly as

$$\hat{f}(q) = \sum_{i=1}^{t} c_i \Phi(p_i, q), \quad (4)$$

where c_i are weighting coefficients. For example, in Fig.3 the distribution of LTP pixel process with unitary background is: 20(00010100)/432 hits, 21(00010101)/4 hits, 84(01010100)/6 hits, and 148(10010100)/58 hits. If no kernel technique is used, the patterns 21 and 84 might be regarded as foregrounds and the pattern 148 might be considered to be another modal. But in our pattern kernel, the latter three patterns are all neighbors of the first one, thus they all will be considered in the same distribution.

For background subtraction in real applications, online adaptation is important. So we also develop an online version of PKDE for adaptation. Given an estimated density function at time t-1 and a new coming background pattern p_t, we update the new density function as

$$\hat{f}_t(q) = (1 - \alpha)\hat{f}_{t-1}(q) + \alpha\Phi(p_t, q), \quad (5)$$

where α is a learning rate. Note that the online PKDE needs no storage for the recent pattern frames.

4. Modeling Background with Local Patterns

A popular technique dealing with dynamic background is multimodal background modeling [3, 8]. Fig.3 also shows multimodality of local pattern based representation. For example, in LBP pixel process on the moving escalator, the patterns 0(00000000)/137 hits, 193(11000001)/176 hits, and 255(11111111)/30 hits constitute three main dynamic background modalities. To handle such case, we estimate K density functions for each pixel process via PKDE technique, with $w_{k,t}, k = 1, 2, .., K$ being the corresponding weights and normalized to 1. The K pdfs are sorted with the corresponding weights in descending order. Then in current state first we determine the number of backgrounds as

$$M = \operatorname*{argmin}_{r} \left(\sum_{k=1}^{r} w_{k,t-1} > T_b \right), \quad (6)$$

where T_b is a threshold indicating how many data should be considered as background. Afterwards, we estimate the probability of a new pattern p_t being background as

$$\hat{P}(p_t) = \frac{1}{\sum_{k=1}^{M} w_{k,t-1}} \sum_{k=1}^{M} w_{k,t-1} \hat{f}_{k,t-1}(p_t). \quad (7)$$

Therefore, foreground/background decision can be made by thresholding $\hat{P}(p_t)$ with a predefined parameter T_s.

In the update procedure, the new pattern p_t is matched to the sorted pdfs in turn, and a match is found when

$f_{k,t-1}(p_t) > T_m$, where T_m is a threshold parameter controlling the matching. Once a match is found, the matched pdf is updated as Equ.(5), while other pdfs remain the same. In addition, the K weights are updated as $w_{k,t} = (1-\alpha)w_{k,t-1} + \alpha x_{k,t}$, where $x_{k,t}$ is an indicator variable being 1 for the matched model and 0 otherwise. If none of the K distributions matches the current pattern, the one with the lowest weight is replaced with a new distribution of $f(q) = \Phi(p_t, q)$, and a low initial weight.

Furthermore, we propose to fuse multiscale spatial information to achieve better performance. Inspired by [5], in this work we further develop a Multiscale Block-based SILTP (MB-SILTP) operator to consider the spatial scale. MB-SILTP encodes in a way similar with Equ.(1), where I_c and I_k are replaced with mean values of corresponding blocks, of which the size $\omega \times \omega$ indicates the scale. Note that MB-SILTP with $\omega = 1$ is actually the SILTP operator.

To implement the MB-SILTP based background subtraction efficiently, first the original video frame is downsampled by ω, with mean values in non-overlapping $\omega \times \omega$ blocks being new pixels. Afterwards, MB-SILTP can be calculated, and the proposed background model can be applied on the reduced space. Finally, the resulted probabilities are upsampled bilinearly to the original space, and thresholded by T_s to generate the foreground/background segmentation result. In this way the speed is much faster with larger scale, whereas the precision is generally lower.

The background probabilities at each scale can be fused on the original space, so that complementary information at each scale can be explored. We simply adopt the geometric average of probabilities at each scale as the fusion score, which are defined as $G(p_t) = \sqrt[n]{\hat{P}_{\omega_0}(p_t)\hat{P}_{\omega_1}(p_t)\cdots\hat{P}_{\omega_{n-1}}(p_t)}$, and the threshold T_s can be applied on this new score for the final decision.

5. Experimental Results

We ran exhaustive experiments on nine datasets containing complex backgrounds, which are publicly available made by [4]. The test databases are difficult videos including busy human flows, moving cast shadows, etc. Most of the datasets contain several thousand video frames, with 20 frames manually labeled for each dataset as the groundtruth.

The proposed approach was compared with existing state-of-the-art online background subtraction algorithms, including mixture of Gaussian ("MoG") [8], "ACMMM03" for complex scene [4], blockwise LBP histogram based approach ("LBP-B") [3], and pixelwise LBP histogram based one ("LBP-P") [2]. The same PKDE based background subtraction framework with LTP (PKDE$_{ltp}$) and SILTP (PKDE$_{siltp}$) operators were both tested. For MB-SILTP we ran other two scales ($\omega = 2, 3$) on each dataset, denoted by "PKDE$^{\omega=2}_{mb-siltp}$" and "PKDE$^{\omega=3}_{mb-siltp}$" respectively. We also applied the proposed multiscale fusion technique in the experiments using the three scales of MB-SILTP, named after "PKDE$^{\omega=1+2+3}_{mb-siltp}$".

For all algorithms, no morphological operation was applied, instead a standard OpenCV postprocessing was used which eliminates small pieces less than 15 pixels. We ran both MoG and ACMMM03 algorithms implemented in OpenCV2.0 with default parameters. Parameters for LBP-B and LBP-P were: 9×9 block size, $LBP_{6,2}$ operator, $K = 3$, $\alpha_b = \alpha_w = 0.01$, $T_D = 0.65$, and $T_B = 0.6$, which were suggested by the authors in their experiments [2]. For the proposed method we used a set of consistent parameters for all experiments: LTP$^{5}_{4,1}$/SILTP$^{0.05}_{4,1}$ operator, $K = 3$, $T_b = 0.7$, $T_s = 0.01$, $T_m = 0.01$, and $\alpha = 0.005$. The weighting function was simply defined as $g(x) = aI_{[x=0]} + bI_{[x=1]}$, with which the kernel function affected first-order neighboring patterns. We set $a = 0.36$, and $b = 0.08$, which worked quite well in the experiments.

We summarize background scenes of all the nine datasets with four categories: indoor busy scenes with moving cast shadows (AirportHall, Bootstrap, and ShoppingMall), with light switching (Lobby), and with dynamic background (Curtain and Escalator); and outdoor scenes with dynamic background (Fountain, Trees, and WaterSurface). With these four categories we want to evaluate the performances of all compared algorithms on two aspects: illumination variation and dynamic background. Due to the paper length, we select four typical datasets among them to analyze the results, which are shown in Fig.4. The foreground/background segmentation results of the compared algorithms (except LBP-B and PKDE$^{\omega=2}_{mb-siltp}$) are demonstrated with one frame for each dataset, of which the classified backgrounds are masked with green color.

For the indoor busy scene "AirportHall", one can see that both MoG and LBP-P fail to suppress the moving shadows. Besides, due to region histogram based method, the detected objects of LBP-P are not accurate in boundary. Besides, ACMMM03 successfully removes the shadows, but many small objects are missed in the detection results. On the other hand, the PKDE framework with local ternary patterns perform much better, where most moving shadows are removed, and the segmentation results are more accurate thanks to the pixel process modeling. Note that the proposed method only uses the grayscale information on the feature level to remove the soft cast shadows from foreground objects, in contrast many shadow removal algorithms design particular shadow detection techniques on color space [7]. Also notice that some small body parts are not detected with both PKDE$_{ltp}$ and PKDE$_{siltp}$, and it is fixed by larger scale of MB-SILTP descriptor, since it can capture larger image structures. And so the fusion of three scales in PKDE$^{\omega=1+2+3}_{mb-siltp}$ takes complementary information between them and gets more accurate segmentation result.

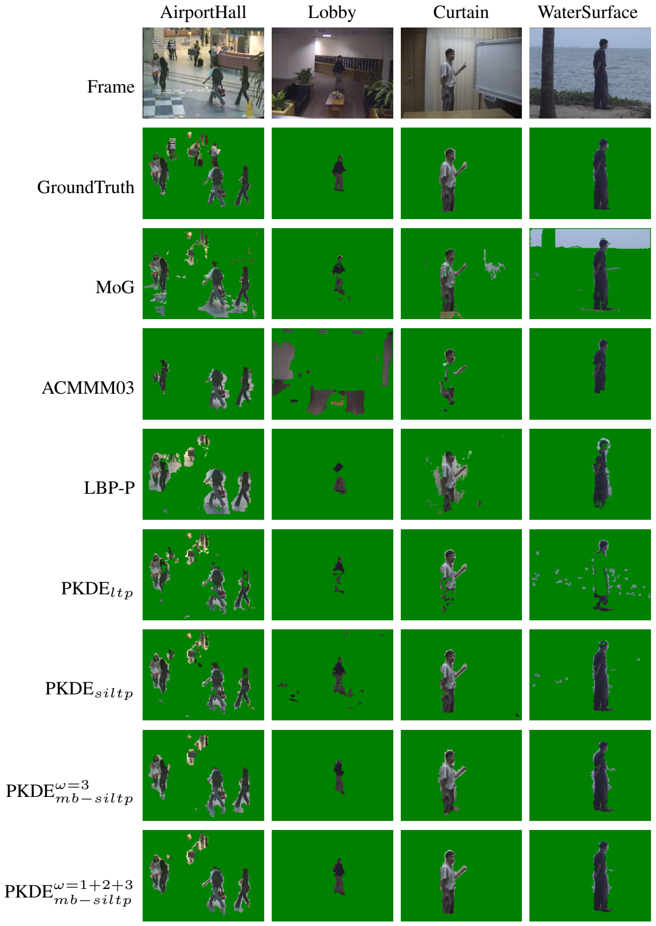

Figure 4. Segmentation results on four datasets.

Next, a frame of "Lobby" dataset is shown in Fig.4 when the indoor light is suddenly turned off and the illumination is much lower. As illustrated, ACMMM03 performs unsatisfactorily in this case, where no object is detected except false foregrounds. In contrast, other methods adapt quickly with the illumination change. Yet some foreground parts are still not detected by MoG, LBP-P, and PKDE$_{ltp}$, while algorithms with SILTP based operators in this case give the most accurate segmentation results.

The indoor dynamic background scenario "Curtain" shown in Fig.4 contains a set of strongly waving curtains due to the wind. As can be seen, the MoG result contains false detections caused by the reflection of the white board. LBP-P detects some waving curtains wrongly as foreground. ACMMM03 successfully learns the dynamic background, but it also falsely suppresses many foreground pixels as background. In contrast, the proposed algorithms all model the dynamic background perfectly and generate accurate results compared to the groundtruth, except that PKDE$_{ltp}$ loses some human body parts.

As for the case of outdoor dynamic scene, the last column of Fig.4 shows a dataset containing a large area of water rippling and a camera jitter. As demonstrated, all algorithms output good segmentation results, except that MoG suddenly segments the blue sky as foreground, and PKDE$_{ltp}$ absorbs the main human body parts as backgrounds. The reason why PKDE$_{ltp}$ performs worse in this case is that the intensities of the foreground object are much lower, and the constant offset value of LTP can not adapt to fit all cases. In contrast, SILTP is scale invariant and can be applied on the full range of grayscale intensities.

A quantitative evaluation is also done on all of the nine datasets, as shown in Tab.1 with the F-score and running framerate in average for all compared algorithms. The F-score measures the segmentation accuracy by considering both the recall and the precision, which is defined as

$$F = \frac{2 \cdot recall \cdot precision}{recall + precision} = \frac{2TP}{2TP + FN + FP}, \quad (8)$$

where TP, FP, and FN are true positives (true foreground pixels), false positives, and false negatives (false background pixels) respectively, computed in pixels of all 20 labeled frames for each dataset. The total F-score is calculated with the TP, FP, and FN summing over all the sequences for an average measure.

From the table it can be inferred that the proposed methods outperforms all of the compared state-of-the-art algorithms on all sequences except the Trees dataset. Furthermore, in average the improvements made by PKDE with SILTP based operators are more than 10% which is impressive. One can also see that the proposed scale invariant LTP operator performs better than LTP, particularly for the AirportHall, Curtain, and the WaterSurface dataset. In general, PKDE with smaller scale MB-SILTP operator is slightly more accurate than that of larger scale, especially for moving shadows and other illumination changes. But in the scenes containing large background motion or large objects, the larger scale is more robust (Note that PKDE$^{\omega=3}_{mb-siltp}$ performs the best on the Watersurface dataset compared to all other algorithms). Finally, by fusing several scales of MB-SILTP, the result is more accurate in average.

All tested algorithms were implemented in c++ and ran on a standard PC with 2.40GHz CPU, 2.0G memory, and Windows XP operation system. The running framerate was tested on the ShoppingMall dataset, with the frame size of 320×256. From the results recorded one can infer that the proposed method can run in real time with a scale larger than one. Balancing between speed and accuracy, we select PKDE$^{\omega=3}_{mb-siltp}$ as the best algorithm among all, which is about two times faster than MoG, and outperforms existing algorithms by more than 10% in accuracy. Also notice that PKDE with a single pattern in pixel process runs faster than LBP-P which needs comparing region-based histograms.

Sequences	MoG	ACMMM03	LBP-B	LBP-P	PKDE$_{ltp}$	PKDE$_{siltp}$	PKDE$^{\omega=2}_{mb-siltp}$	PKDE$^{\omega=3}_{mb-siltp}$	PKDE$^{\omega=1+2+3}_{mb-siltp}$
AirportHall	57.86	50.18	47.73	50.29	62.13	**68.14**	65.87	63.60	68.02
Bootstrap	54.07	60.46	52.81	52.00	73.86	**75.35**	69.45	64.87	72.90
Curtain	50.53	56.08	66.08	71.42	74.19	91.16	89.37	87.97	**92.40**
Escalator	36.64	32.95	59.08	53.93	67.71	63.90	64.37	60.18	**68.66**
Fountain	77.85	56.49	70.52	75.33	81.05	83.45	81.17	77.60	**85.04**
ShoppingMall	66.95	67.84	54.67	62.92	73.91	79.62	77.75	74.49	**79.65**
Lobby	68.42	20.35	50.29	52.34	77.85	78.80	73.82	67.16	**79.21**
Trees	55.37	**75.40**	62.85	60.57	42.98	42.54	51.88	61.53	67.83
WaterSurface	63.52	63.66	76.80	82.21	41.46	74.30	81.08	**83.51**	83.15
Total	60.14	59.21	58.69	63.46	67.59	75.35	75.24	73.59	**78.69**
Framerate	33.5	23.3	10.5	1.5	8.9	8.7	31.8	**63.9**	6.3

Table 1. Performance of F-score (%) on the nine complex video sequences and the running framerate (fps) in average.

6. Conclusion

In this paper we have demonstrated a novel pixelwise background subtraction algorithm with local patterns on monocular grayscale video sequences. We have proposed an improved local image descriptor called SILTP, and have demonstrated its power for background subtraction. We have also proposed a multiscale block-based SILTP operator for considering the spatial scale information. For modeling the pixel process with local texture descriptors, we have proposed a Pattern Kernel Density Estimation technique and based on it we have developed a multimodal background modeling framework. We have run extensive experiments on complex scenes to prove that one single local texture pattern instead of region histogram is really enough for the background subtraction task. We have achieved more than 10% improvement in accuracy compared to existing algorithms, with a speed of about two times faster than the standard mixture of Gaussian approach.

Future works will be investigating color information into the proposed background subtraction framework. Furthermore, we believe that the proposed SILTP operator and the PKDE technique for estimating the distribution of local patterns are not limited to the field of background modeling. It is expected that they will also be effective on other research areas such as object detection and object recognition.

Acknowledgements.

The authors would like to thank Dr. Marko Heikkilä and Machine Vision Group for providing LBP based results. This work was supported by the following funding resources: the Academy of Finland, the Chinese National Hi-Tech (863) Program Projects #2008AA01Z124, and AuthenMetric R&D Funds.

References

[1] A. Elgammal, R. Duraiswami, D. Harwood, L. S. Davis, R. Duraiswami, and D. Harwood. “Background and foreground modeling using nonparametric kernel density for visual surveillance”. In *Proceedings of the IEEE*, pages 1151–1163, 2002.

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

[3] M. Heikkilä, M. Pietikäinen, and J. Heikkilä. “A texture-based method for detecting moving objects”. In *British Machine Vision Conference*, pages 187–196, 2004.

[4] L. Li, W. Huang, I. Gu, and Q. Tian. “Foreground object detection from videos containing complex background”. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 2–10, New York, USA, 2003. http://perception.i2r.a-star.edu.sg.

[5] S. Liao, X. Zhu, Z. Lei, L. Zhang, and S. Z. Li. “Learning multi-scale block local binary patterns for face recognition”. In *Proceedings of the 2nd IAPR/IEEE International Conference on Biometrics*, volume LNCS-4642, pages 828–837, Seoul, Korea, 2007.

[6] T. Ojala, M. Pietikäinen, and M. Mäenpää. “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[7] A. Prati, I. Mikic, M. M. Trivedi, and R. Cucchiara. “Detecting moving shadows: Algorithms and evaluation”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):918–923, 2003.

[8] C. Stauffer and W. Grimson. “Adaptive background mixture models for real-time tracking”. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999.

[9] X. Tan and B. Triggs. “Enhanced local texture feature sets for face recognition under difficult lighting conditions”. In *Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, 2007.

[10] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. “Wallflower: principles and practice of background maintenance”. In *Proceedings of IEEE International Conference on Computer Vision*, volume 1, pages 255–261, 1999.