

# Complex Background Modeling and Motion Detection based on Texture Pattern Flow

Baochang Zhang<sup>1</sup>, Yongsheng Gao<sup>2</sup>, Bineng Zhong<sup>3</sup>

<sup>1</sup>*School of Automation Science and Electrical Engineering, Beihang University, Beijing, China*

<sup>2</sup>*Institute for Integrated and Intelligent System, Griffith University, Australia*

<sup>3</sup>*Computer College, Harbin Institute of Technology, China*

*b.zhang@griffith.edu.au; yongsheng.gao@griffith.edu.au; bnzhong@vilab.hit.edu.cn*

## Abstract

*This paper proposes a novel Texture Pattern Flow (TPF) for complex background modeling and motion detection. The Pattern Flow is proposed to encode the binary pattern changes among the neighborhoods in the space-time domain. To model the distribution of the TPF, the TPF integral histograms are used to extract the discriminative features to represent the input video. Experimental results on the public videos testify the effectiveness of the proposed method in comparison to LBP and GMM based background modeling methods.*

## 1 Introduction

Background modeling and moving objects detection in video or image sequence is one of the main tasks in many computer vision applications, such as industrial automation, transportation, security & surveillance. To achieve the goal of capturing moving objects, one has to obtain a representation of the background, which should be robust to the changes of the illumination condition, swaying trees, rippling water, and shadows of the moving and scene objects.

Recently, many methods have been proposed to model the background. A very popular technique is to model each pixel in a video frame with a Gaussian distribution [1]. The model does not work well in the case of dynamic natural environments, where the scene background is not completely static. In [2], the Mixture of Gaussians Models (GMM) is used to model complex background, and applied to the traffic monitoring problem. To deal with the limitations of parametric methods, a nonparametric approach to background modeling is proposed in [3]. The method utilizes a general nonparametric kernel density

estimation technique for building a statistical representation of the scene background. These methods do not consider the relationship between two neighborhood pixels. [4] reports a promising background method to model the dynamic background based on the discriminative Local Binary Pattern (LBP) histograms. The LBP operator is originally designed for the texture description [5]. The operator encodes the relationship between the central point and its neighbors. LBP has been successfully applied to facial expression analysis [6], background modeling [4] and face recognition [7]. In background modeling, it achieves a better performance than GMM in terms of false negative and positive. However, the LBP method does not consider the variation over time in its pattern, which contains the inherent motion information derived from moving objects in videos.

In this paper, a novel Texture Pattern Flow (TPF) is proposed to encode the local pattern information in space and time domains. In the space domain, TPF is a directional texture descriptor, which is important because the foreground object always moves in some direction. In the time space, the TPF operator is proposed to capture motion information over time in the image sequence. Furthermore, the integral histogram is combined with TPF, named Background Modeling Texture Pattern Flow (BM-TPF), to statistically model the input video. The reason for this lies in that complexity of integral histogram is  $O(N)$  [11].

The reminder of the paper is organized as follows. In Section 2, the Texture Pattern Flow is proposed for background modeling and motion detection. In Section 3, we discuss the construction and maintenance of the background model using integral histogram. Experimental results and discussion are presented in Section 4 and Section 5.

## 2 Modeling by Texture Pattern Flow

LBP is generated by concatenating the binary result of the gradient direction, which is widely used in the field of image processing and object representation [5]. In this section, we propose a novel operator, Texture Pattern Flow (TPF), for background modeling and motion detection. TPF is a method to encode local pattern flow in a local region in both space and time aspects.

### 2.1 Local Binary Pattern

The original LBP operator labels the pixels of an image by coding the  $3 \times 3$  neighborhood relationship of each pixel with the value of the central pixel and concatenating the results as a binary string. The threshold function  $f$  for LBP can be written as:

$$f(I_{x,y}, I_{x',y'}) = \begin{cases} 0, & \text{if } I_{x,y} - I_{x',y'} \leq \text{threshold} \\ 1, & \text{if } I_{x,y} - I_{x',y'} > \text{threshold} \end{cases} \quad (1)$$

where  $I_{x,y}$  is the neighborhood around  $I_{x',y'}$ , as shown in Fig. 1. The histograms of these micro-patterns contain information of the distribution of the edges, spots, and other local features in an image [5].

$I_{x-1,y-1}$	$I_{x,y-1}$	$I_{x+1,y-1}$
$I_{x-1,y}$	$I_{x,y}$	$I_{x+1,y}$
$I_{x-1,y+1}$	$I_{x,y+1}$	$I_{x+1,y+1}$

Fig. 1. An example of 8-neighborhood around  $I_{x,y}$ .

### 2.2 Texture Pattern Flow

Different from LBP encoding differences between the central point and its neighbors, Texture Pattern Flow (TPF) is proposed to encode the texture pattern flow in a local region.

Given an image sequence  $I_{x,y,t}$  with  $x$ ,  $y$  and  $t$  representing space and time indexes respectively. The gradients in  $x$ ,  $y$  directions can be calculated as

$$\frac{\partial I_{x,y,t}}{\partial x} = I_{x,y,t} - I_{x+1,y,t}, \quad (2)$$

$$\frac{\partial I_{x,y,t}}{\partial y} = I_{x,y,t} - I_{x,y+1,t}, \quad (3)$$

The threshold functions,  $f_x(\frac{\partial I_{x,y,t}}{\partial x})$  and  $f_y(\frac{\partial I_{x,y,t}}{\partial y})$ , are used to encode the binary results in  $x$  and  $y$  directions defined as

$$f_x\left(\frac{\partial I_{x,y,t}}{\partial x}\right) = \begin{cases} 1, & \text{if } \frac{\partial I_{x,y,t}}{\partial x} \geq 0 \\ 0, & \text{if } \frac{\partial I_{x,y,t}}{\partial x} < 0 \end{cases}, \quad (4)$$

$$f_y\left(\frac{\partial I_{x,y,t}}{\partial y}\right) = \begin{cases} 1, & \text{if } \frac{\partial I_{x,y,t}}{\partial y} \geq 0 \\ 0, & \text{if } \frac{\partial I_{x,y,t}}{\partial y} < 0 \end{cases}, \quad (5)$$

Different from the space variations in  $x, y$  direction encoding by simple threshold functions, we define derivative over time as

$$\frac{\partial I_{x,y,t}}{\partial t} = I_{x,y,t} - \mu_{x,y,t-1}, \quad (6)$$

where  $\mu_{x,y,t-1}$  is one of the mean values in GMM, which is calculated by using a mixture of Gaussian models [2]. It is exploited to model the distribution of recent history pixel values as

$$P(I_{x,y,t}) = \sum_{i=1}^K \omega_{i,t} * \eta(I_{x,y,t}, \mu_{x,y,t,i}, \Sigma_{t,i}), \quad (7)$$

where  $K$  is the number of Gaussian models,  $\omega_{i,t}$  is an estimate of the weight of the  $i^{\text{th}}$  model,  $\mu_{x,y,t,i}$  and  $\Sigma_{t,i}$  are the mean value and co-variance matrix of the Gaussian probability density function  $\eta$ .  $\eta$  is defined as

$$\eta(I_{x,y,t}, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{1}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(I_{x,y,t} - \mu_{x,y,t,i})^T \Sigma^{-1} (I_{x,y,t} - \mu_{x,y,t,i})}, \quad (8)$$

In [2], the on-line K-means approximation is exploited to implement GMM. Every pixel is checked against the existing  $K$  Gaussian distributions ( $t-1$ ), until a match is found. A match is defined as a pixel value within 2.5 standard deviation of a distribution, which means that  $\text{GMM}(I_{x,y,t})$  is true.

$$f_t(I_{x,y,t}) = \begin{cases} 1, & \text{if } \text{GMM}(I_{x,y,t}) \text{ is true} \\ 0, & \text{otherwise} \end{cases}, \quad (9)$$

where  $\text{GMM}(I_{x,y,t})$  is used to decide whether the pixel at the position  $(x, y)$  is part of background or not. If it is a part of background, the threshold function  $f_t(I_{x,y,t})$  gets 1, otherwise, it is labeled as 0. It is actually a kind of threshold function encoding the binary result of  $I_{x,y,t} - \mu_{x,y,t-1}$  as shown in Eq. 8 and Eq. 9. The Texture Pattern Flow (TPF) is defined as

$$\text{DRP}_t(I_{x,y,t}) = \left\{ f_x\left(\frac{\partial I_{x-1,y,t}}{\partial^2 x}\right), f_x\left(\frac{\partial I_{x+1,y,t}}{\partial^2 x}\right), f_y\left(\frac{\partial I_{x,y-1,t}}{\partial^2 y}\right), f_y\left(\frac{\partial I_{x,y+1,t}}{\partial^2 y}\right), f_t(I_{x,y,t}) \right\}. \quad (10)$$

From the definition of TPF, we can know that TPF is proposed to reveal the relationship between two derivative directions in both space and time domains. Similar to LBP, we also add a perturbation to the first order derivative in the space domain, i.e.,  $I_{x,y,t} - I_{x+1,y,t} + 3$ , which makes TPFs more robust against the negligible changes in pixel values.

### 3. Background Modeling based on Integral TPF Histogram

In this section, we will exploit the integral Texture Pattern Flow histogram to more efficiently model the background.

#### 3.1 Integral Texture Pattern Flow Histogram

The integral image was first proposed by Viola and Jones [10], the advantage of which is that the sum of all pixel intensity of any rectangle region in an image can be calculated at a low cost. In order to efficiently calculate the histogram feature, the integral histogram is proposed to extract the statistic feature for each local region [11, 12]. The integral histogram feature is actually extracted from a local region for the pixel  $I(i, j)$  as shown in Fig. 4, details about which can refer to [11]. The mean grey-level value for a local region is used to represent each pixel, which is used on GMM, named L-GMM. As for the BM-TPF method, we calculate a TPF pattern based on the mean grey-level image, and then the histogram features are extracted as the final representation. In this paper, we model each pixel of the background identically, which allows for a high-speed parallel implementation if needed.

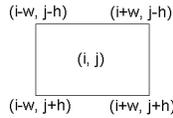


Fig. 4. A sample of the local region for the pixel  $I(i, j)$

#### 3.2 Background Detection by Integral TPF histogram

The background model for the pixel consists of a group of TPF histograms with an adaptive threshold. If the similarity is below the threshold  $Th$  for all model histograms, we create a new model histogram with a low initial weight to replace the model histogram with the lowest weight. We select the best match as the model histogram with the highest proximity value. The matched model histogram is adapted with the new data

by updating its bins using Eq. 12, the bigger the weight, the higher the probability of being a background histogram. As a last stage of the updating procedure, we sort the model histograms in decreasing order according to their weights.

The propagation speed of the background model is controlled by the learning rate parameter  $\alpha$ . In our experiments, a value of  $\alpha = 0.01$  is used. The bigger the learning rate, the faster the propagation is, and weights are updated as:

$$w_{k,t} = (1 - \alpha)w_{k,t-1} + \alpha(M_{k,t}), \quad (11)$$

where  $M_{k,t}$  is 1 for the best matched model and 0 for the remaining models. Furthermore, the best matched model histogram is updated as

$$\mu_{k,t} = (1 - \alpha)\mu_{k,t-1} + \alpha H, \quad (12)$$

where  $H$  is the histogram feature extracted from the current local region around the given pixel.

In the classification phase, the first  $n$  model histograms are chosen as the background model according to Eq. 13, where the weights of model histograms are sorted in descending order.

$$w_{0,t} + w_{1,t} + \dots + w_{n-1,t} > T_w, \quad (13)$$

where  $T_w = 0.7$ . If the similarity is higher than the threshold for at least one background model, the pixel is classified as background. Otherwise, the pixel is labeled as foreground.

In this paper, the similarity between two histograms is calculated by the following similarity rule,

$$HI(x, y) = \sum_{i=1}^B \min(x_i, y_i), \quad (14)$$

where  $HI(x, y)$  is histogram intersection. This measure has an intuitive motivation in that it calculates the common parts of two histograms. Its advantage is that the computation is very fast as it requires very simple operations only.

## 4 Experiments

Two public videos were used to validate the proposed method, and numerical evaluation is usually done in terms of the number of false negatives (the number of foreground pixels that were missed) and false positives (the number of background pixels that were marked as foreground). The ground truth is provided in [13]. In the experiment, video3 (898 frames) and video4 (818 frames) have sudden changes and swaying trees.

In the experiments, all the test phases begin from the 50<sup>th</sup> frame, and both visual and numerical methods are used to evaluate our approach. We compared L-GMM, BM-TPF with GMM [3] and the LBP-based

methods. According to the test results, the overall performances of our methods are better than the performances of the comparison methods for the test sequences used. The sequences used in the experiment focus on the outdoor scenes with the provided ground truth sequences. The results shown in Table 1 and Table 2 illustrate that the numbers of error classifications which are calculated by summing the errors from the frames corresponding to the ground truth frames. For the two sequences, the proposed BM-TPF method gives less false negatives and false positives than the LBP method.

According to the overall results, the proposed method outperforms the comparison methods for the used test sequences. It should be noted that most of false positives occur on the moving objects' contours. The BM-TPF method is the best in terms of the sum of false negatives and positives as shown in Table 1 and 2. These experimental results show that TPF is effective in extracting the discriminant features from the input videos. L-GMM can also significantly enhance the original GMM method. Note that most of the false positives occur near the contour part of the moving objects.

**Table.1 Comparative results for BM-TPF, LBP, GMM and L-GMM in terms of false positive and negative ( $\times 10^5$ ) on video3**

	False Positive	False Negative	Sum
BM-TPF	19.3	1.64	20.94
LBP	21.76	2.26	24.02
GMM	66.84	2.06	68.91
L-GMM	23.13	1.85	24.99

**Table.2 Comparative results for BM-TPF, LBP, GMM and L-GMM in terms of False positive and negative ( $\times 10^5$ ) on video4**

	False Positive	False Negative	Sum
BM-TPF	29.7	1.95	31.65
LBP	40.6	2.98	43.58
GMM	68.80	3.01	71.81
L-GMM	45.4	2.53	47.93

In the experiment, the threshold value ( $Th$ ) for the TPF and LBP methods is set to 0.7, and  $T_\alpha = 0.01$ . 3 Gaussian models are used for GMM. For integral histogram and L-GMM, the sizes of the local region is  $10 \times 10$  and  $8 \times 8$ , respectively. The bin numbers for LBP are 32 by using the equalization method. Therefore, the feature lengths for the LBP and TPF integral histogram are same.

## 5 Conclusion and future work

This paper proposes a new background modeling method, which is a combination of a new Texture Pattern Flow feature and integral histogram. The proposed method is successfully tested on two public videos, showing that TPF is much more robust to the background variations. TPF achieved a better performance than LBP and GMM, because it identifies more discriminant information in the space-time domain. In future work, we will focus on the efficiency of proposed method through the parallel implementation.

## 6. References

- [1] C.R. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland, Pfunder: Real-Time Tracking of the Human Body, IEEE TPAMI, vol. 19, no. 7, pp. 780-785, 1997.
- [2] C. Stauffer and W.E.L. Grimson, Adaptive Background Mixture Models for Real-Time Tracking, in proc. of IEEE CVPR, vol. 2, pp. 246-252, 1999.
- [3] A. Elgammal, R. Duraiswami, D. Harwood, and L.S. Davis, Background and Foreground Modelling Using Nonparametric Kernel Density Estimation for Visual Surveillance, in proc. of IEEE, vol. 90, no. 7, pp. 1151-1163, 2002.
- [4] M. Heikkilä and M. Pietikäinen, A Texture-Based Method for Modelling the Background and Detecting Moving Objects, IEEE TPAMI, vol. 28, no. 4, pp. 657-662, 2006.
- [5] T. Ojala, M. Pietikäinen, and T. Mäenpää, Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns, IEEE TPAMI, vol. 24, no. 7, pp. 971-987, 2002.
- [6] G. Zhao and M. Pietikäinen, Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions, IEEE TPAMI, vol. 29, no. 6, pp. 915-928, 2007.
- [7] T. Ahonen, A. Hadid, and M. Pietikäinen, Face Description with Local Binary Patterns: Application to Face Recognition, IEEE TPAMI, vol. 28, no. 12, pp. 2037-2041, 2006.
- [8] Y. Gao and M. Leung, Face Recognition Using Line Edge Map, IEEE TPAMI, vol. 24, no.6, pp.764-779, 2002.
- [9] D. Gabor, Theory of Communication, Journal of the Institution of Electrical Engineers, vol. 93, part III, no. 26, pp. 429-457, 1946.
- [10] P. Viola, M. Jones, Robust Real-time Face Detection. International Journal of Computer Vision, vol. 57, no. 2, pp. 137-154, 2004.
- [11] H. Wang, P. Li, T. Zhang, Proposal of Novel Histogram Features for Face Detection. In proc. of the 3rd International Conference on Advances in Pattern Recognition, vol. 3687, pp. 334-343, 2005.
- [12] F. Porikli, Integral Histogram, A Fast Way to Extract Histograms in Cartesian Spaces, in proc. of IEEE CVPR, vol. 1, pp. 829-836, 2005.
- [13] [http://mmc36.informatik.uni-augsburg.de/VSSN06\\_OSA\\_C/#ref](http://mmc36.informatik.uni-augsburg.de/VSSN06_OSA_C/#ref)