Static and Moving Object Detection Using Flux Tensor with Split Gaussian Models

Rui Wang[1], Filiz Bunyak[1], Guna Seetharaman[2], Kannappan Palaniappan[1]
[1]Department of Computer Science, University of Missouri, Columbia MO 65211, USA
[2] Air Force Research Laboratory, Rome NY 13441, USA
{rwc33,bunyak,palaniappank}@missouri.edu, guna@ieee.org

Abstract

In this paper, we present a moving object detection system named Flux Tensor with Split Gaussian models (FTSG) that exploits the benefits of fusing a motion computation method based on spatio-temporal tensor formulation, a novel foreground and background modeling scheme, and a multi-cue appearance comparison. This hybrid system can handle challenges such as shadows, illumination changes, dynamic background, stopped and removed objects. Extensive testing performed on the CVPR 2014 Change Detection benchmark dataset shows that FTSG outperforms state-of-the-art methods.

1. Introduction

In real world monitoring applications, moving object detection remains to be a challenging task due to factors such as background complexity, illumination variations, noise, and occlusions. As a fundamental first step in many computer vision applications such as object tracking, behavior understanding, object or event recognition, and automated video surveillance, various motion detection algorithms have been developed ranging from simple approaches to more sophisticated ones [11].

In this paper, we present a novel hybrid moving object detection system that uses motion, change, and appearance information for more reliable detections. The main contributions of this paper are: (i) A motion computation method based on spatio-temporal tensor formulation named flux tensor; (ii) A novel split Gaussian method to separately model foreground and background; (iii) A robust multi-cue appearance comparison module to remove false detections due to illumination changes, shadows etc. and to differentiate stopped objects from revealed background by removed objects. Our method can handle shadow, illumination changes, ghosts, stopped or removed objects, some dynamic background and camera jitter while still maintaining a fast boot-strapping. Our method outperforms most well known techniques on moving object detection. As of submission date of this paper, our results outrank submissions to CVPR 2014 change detection challenge [7] in overall ranking that combines eleven categories.

2. System Overview

Figure 1 shows our system flow diagram. Flux Tensor with Split Gaussian models (FTSG) consists of three main modules described below:

a) Pixel level motion detection module: two complementary methods, flux tensor based motion detection and split Gaussian models based background subtraction, run separately on input images and produce foreground detection results.
b) Fusion module: flux tensor based and split Gaussian based detection results are fused using a rule-based system to produce improved results that reduce errors due to noise, illumination changes, and halo effects.
c) Object level classification module: removed and stopped objects are handled. Edges of the static objects in foreground detection mask are compared to the edges of the corresponding object in current image and background model using chamfer matching.

Detailed descriptions of each component are given in the following sections.

3. Flux Tensor based Motion Detection

Motion blob detection is performed using multichannel version of flux tensor method [3] which is an extension to 3D grayscale structure tensor. Using flux tensor, motion information can be directly computed without expensive eigenvalue decompositions. Flux tensor represents the temporal variation of the optical flow field within the local 3D spatiotemporal volume. In expanded matrix form, flux

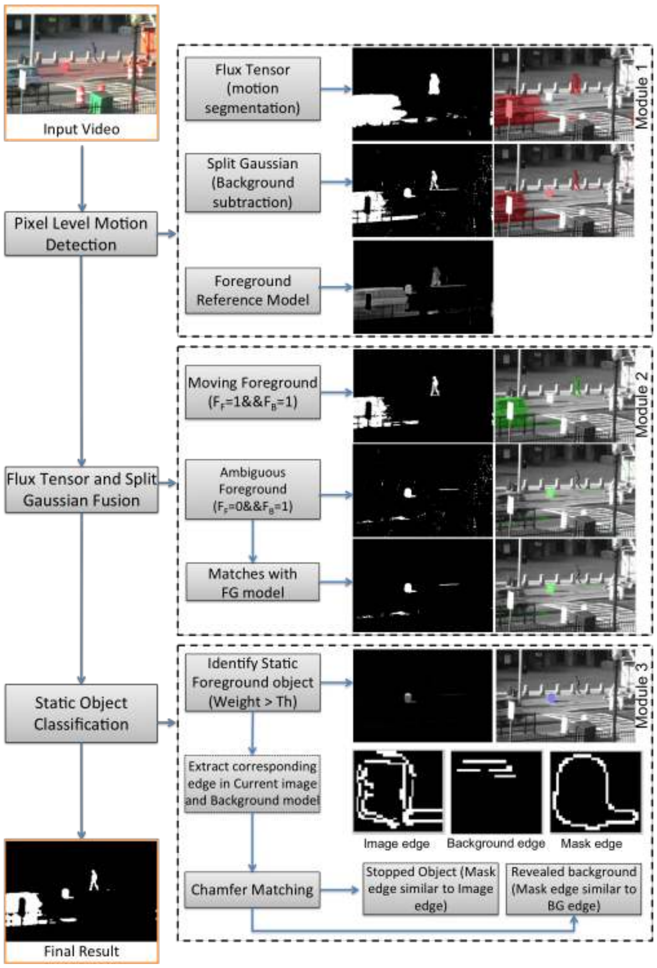

Figure 1. System flow diagram for static and moving object detection using flux tensor with split Gaussian models. The system is composed of three major modules.

tensor is written as:

$$\mathbf{J_F} = \begin{bmatrix} \int_{\mathbf{\Omega}} \{\frac{\partial^2 \mathbf{I}}{\partial x \partial t}\}^2 d\mathbf{y} & \int_{\mathbf{\Omega}} \frac{\partial^2 \mathbf{I}}{\partial x \partial t}\frac{\partial^2 \mathbf{I}}{\partial y \partial t} d\mathbf{y} & \int_{\mathbf{\Omega}} \frac{\partial^2 \mathbf{I}}{\partial x \partial t}\frac{\partial^2 \mathbf{I}}{\partial t^2} d\mathbf{y} \\ \int_{\mathbf{\Omega}} \frac{\partial^2 \mathbf{I}}{\partial y \partial t}\frac{\partial^2 \mathbf{I}}{\partial x \partial t} d\mathbf{y} & \int_{\mathbf{\Omega}} \{\frac{\partial^2 \mathbf{I}}{\partial y \partial t}\}^2 d\mathbf{y} & \int_{\mathbf{\Omega}} \frac{\partial^2 \mathbf{I}}{\partial y \partial t}\frac{\partial^2 \mathbf{I}}{\partial t^2} d\mathbf{y} \\ \int_{\mathbf{\Omega}} \frac{\partial^2 \mathbf{I}}{\partial t^2}\frac{\partial^2 \mathbf{I}}{\partial x \partial t} d\mathbf{y} & \int_{\mathbf{\Omega}} \frac{\partial^2 \mathbf{I}}{\partial t^2}\frac{\partial^2 \mathbf{I}}{\partial y \partial t} d\mathbf{y} & \int_{\mathbf{\Omega}} \{\frac{\partial^2 \mathbf{I}}{\partial t^2}\}^2 d\mathbf{y} \end{bmatrix} \tag{1}$$

The elements of the flux tensor incorporate information about temporal gradient changes which leads to efficient discrimination between stationary and moving image features. Thus the trace of the flux tensor matrix which can be compactly written and computed as,

$$\mathbf{trace}(\mathbf{J_F}) = \int_{\mathbf{\Omega}} ||\frac{\partial}{\partial t}\nabla \mathbf{I}||^2 d\mathbf{y} \tag{2}$$

can be directly used to classify moving and non-moving regions without eigenvalue decompositions. Flux tensor based moving object detection has been successfully used in both surveillance [4, 10] and biomedical video analysis applications [9, 8].

4. Split Gaussian Models

Gaussian models have been widely used in background subtraction methods. Mixture of Gaussians can efficiently represent multimodal signals, which makes them suitable for background modeling and subtraction. We adopt mixture of Gaussians as our background model. However, unlike MoG in [12] where background and foreground are blended together into a single model with fixed number of Gaussians, we model foreground and background separately, and use adaptively changing number of Gaussians for the background model. This simplifies the background/foreground classification step, prevents background model from being corrupted by foreground pixels, and also provides better adaptation for different background types (static vs. dynamic backgrounds). This approach has fast boot-strapping, adaptive updating and complex background environment modeling capabilities.

Background model: We use a mixture of K Gaussians to model the background where K is a spatially and temporally adaptive variable. Every new pixel value, $I_t(x, y)$, is checked against the existing K Gaussian distributions. A match to a Gaussian is defined as pixel values within T_b standard deviations of the mean :

$$D_{min}(x, y) = \min_{i \in K} \max_{j \in C}((\mathbf{I}_t(x, y) - \mu_{i,j})^2 - T_b \cdot \sigma^2) \tag{3}$$

A pixel is labeled as foreground if it does not match any of the Gaussians in the background model:

$$\mathbf{F}_B(x, y) = \begin{cases} 1, & if\ \mathbf{D}_{min}(x, y) > 0 \\ 0, & otherwise \end{cases} \tag{4}$$

T_b is a fixed threshold and stands for number of standard deviations, and $\sigma = \sum_i^k \omega_i \sigma_i$. For each pixel, there will be $K \times C$ Gaussian models where C is the number of channels, e.g. 3 for RGB. For simplicity, all the channels share the same variance σ and weight ω.

Foreground appearance model: We use a single Gaussian to model the foreground. Foreground appearance model (shown in Figure 1, module 1) is used to distinguish static foreground (stopped object and revealed background) from spurious detections due to illumination changes and noise within ambiguous regions, $\mathbf{F}_{amb}(x, y)$ where $\mathbf{F}_F = 0$ and $\mathbf{F}_B = 1$ (detected as background by flux but as foreground by background subtraction shown as *ambiguous foreground* in Figure 1 module 2). Static foreground regions $\mathbf{F}_S$ are identified within ambiguous detections $\mathbf{F}_{amb}$ using foreground model:

$$\mathbf{F}_S(x, y) = \begin{cases} 1, & if \quad \mathbf{F}_{amb}(x, y) = 1\ and \\ & \quad \mathbf{I}_t(x, y) - \mu_f(x, y) < T_f \\ 0, & otherwise \end{cases} \tag{5}$$

Model initialization: Flux tensor provides motion information, and the fusion and classification modules greatly reduce false positives. Therefore, the background model can be directly initialized using the first few frames and the foreground appearance model can be initialized to be empty.
Background model update: Common background model update schemes can be classified as blind update or conservative update [1]. Blind update, such as in MoG [12], incorporates all sample values into the background model, while conservative update only incorporates sample values that are previously classified as background. We use the conservative update policy for both our background and foreground models. Fusion (Section 5) and object classification (Section 6) modules considerably reduce potential deadlock problems in conservative update where temporary detection errors may become permanent ghosts. Static background and illumination changes are updated into background model as:

$$\mu_t = (1-\alpha)M\mu_{t-1} + \alpha M I_t \tag{6}$$

$$\sigma_t^2 = (1-\alpha)M\sigma_{t-1}^2 + M\alpha(I_t-\mu)^T\alpha(I_t-\mu) \tag{7}$$

$$\omega_{i,t} = (1-\alpha)\omega_{i,t-1} + \alpha M \tag{8}$$

$$M = (1-\mathbf{F}_B) \cup (\mathbf{F}_{amb} - \mathbf{F}_S) \tag{9}$$

where α is a fixed learning rate set to 0.004 and M stands for update mask. Background revealed by removed objects and dynamic background are incorporated to background model as new Gaussian distributions. A new Gaussian is initialized with a high variance and low weight, and its mean is set to the current pixel value.

If there is a large persistent change, a new model will be added to each pixel (i.e. in PTZ scenario [7], camera field of view change triggers large persistent change). Existing Gaussian models with weights less then a threshold T_l are discarded.
Foreground model update: As in the case of the background model, a conservative update strategy is used for the foreground model. Foreground model is only updated with the foreground regions indicated by the inverse of the background model update mask. In order to accommodate fast changing foreground, a high learning rate is used for foreground update.

5. Fusion of Flux Tensor and Split Gaussian Models

The goal of this decision fusion module is to exploit complementary information from two inherently different approaches to boost overall detection accuracy. Flux tensor based motion segmentation produces spatially coherent results due to spatio-temporal integration. These results are also robust to illumination changes and soft shadows due to use of gradient based information. But since the method relies on motion, it fails to detect stopped foreground objects

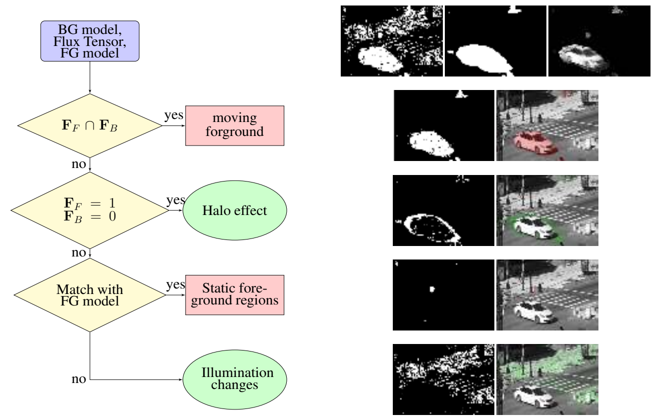

Figure 2. Fusion of flux tensor and split Gaussian models. Images on the right hand side are corresponding to those elements in the flowchart on the left hand side. $\mathbf{F}_F$, $\mathbf{F}_B$ stand for flux tensor motion segmentation mask and split Gaussian background subtraction mask respectively.

and tends to produce masks larger than the objects. Background subtraction on the other hand can detect stopped objects, but is sensitive to noise, illumination changes and shadows. Here we extend flux tensor based motion segmentation with split Gaussian foreground and background models to generate a more complete and accurate foreground object detection method. Figure 2 shows fusion flow chart and some examples of flux tensor and split Gaussian model fusion results. Pixels that are detected as foreground by both flux tensor and split Gaussian background subtraction are classified as moving foreground objects. Pixels that are detected as foreground by background subtraction only and have a match in foreground model correspond to static foreground objects.

6. Stopped and Removed Object Classification

Fusion procedure classifies both stopped objects (true positives) and revealed background by removed objects (false positives) as static foreground. Distinguishing these two types of static foreground can effectively reduce the false positive rate and tackle deadlock problem. The method

(a) Stopped object

(b) Revealed background by removed object

Figure 3. Classification of stopped objects vs. background revealed by removed objects. Images on the first row from left to right are current image, background model and foreground mask. Images on the second row are edge maps corresponding to the regions of interest marked by red rectangle in the images of the first row.

used for removed and stopped objects classification is based on [6], which basically has three steps: 1. Identify pixels corresponding to static regions; 2. Perform edge detection on static regions in current image, background generated by background subtraction and foreground detection mask; 3. Perform classification based on edge matching. Figure 3 a, b show classification examples for stopped object (an abandoned bag) and revealed background by removed object (ghost effect due to background model initialization) respectively. Stopped object has higher edge similarity between current image and foreground mask, while revealed background by removed object has higher edge similarity between background model and foreground mask.

7. Results and Analysis

The proposed flux tensor with split Gaussian models system is evaluated using the dataset and evaluation metrics in CVPR 2014 Change Detection challenge [7]. One fixed set of parameters is used for all the sequences. The learning rate α is 0.004 for background model and 0.5 for foreground model. The matching threshold T_b in Eq. 3 is 3 and the similarity matching threshold T_f in Eq. 5 is 20. The threshold for flux tensor to segment moving foreground object from non-moving background is dynamically changing according to the number of Gaussians distributions at each pixel location. This avoids the use of a fixed global threshold unlike most other temporal differencing methods.

Table 1 shows the comparison result of FTSG with state-of-the-art change detection methods. Evaluation scores of those methods are obtained from `http://www.changedetection.net`. Best result of each metric is highlighted and in all the measures listed in Table 1. It can be seen that FTSG outperforms all the listed methods in five out of seven measures and has the second best score in the remaining two measures, specificity and FPR. Table 2 shows results of the proposed approach on all eleven scenarios. On seven out of eleven scenarios and on the overall evaluation FTSG outperforms not only the listed state-of-the-art methods but also the new change detection challenge submissions in terms of average ranking.

Figure 4 shows moving object detection results for various algorithms including proposed Flux Tensor with Split Gaussian models (FTSG) on CVPR 2014 Change Detection dataset [7] with some typical frames selected from the 11 categories. The proposed FTSG is robust to illumination changes (col 1), it can detect long term static objects (col 3), and it also handles dynamic background (col 2). Image in col 4 demonstrates that FTSG can correctly identify revealed background by removed object, and image in col 5 shows that FTSG can adapt to scene changes quickly (sudden change of camera focus).

A prototype of the proposed system implemented in Matlab runs at 10 fps for a 320×240 video. Matlab implementation of Flux tensor only detection runs at 50 fps. Flux tensor computation can be easily parallelized for different architectures as in [10] because of the fine grain parallelism of the filter operations.

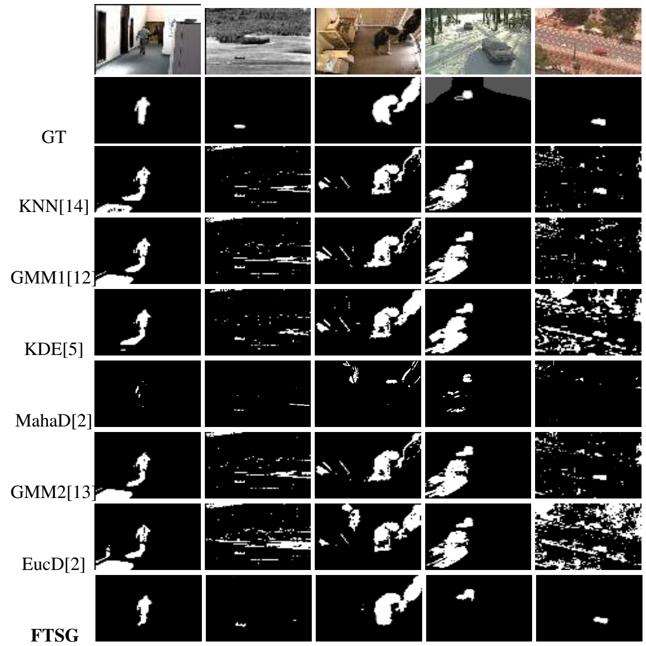

Figure 4. Selected foreground detection results from six state-of-the-art change detection algorithms and our FTSG method on CVPR 2014 Change Detection dataset [7]. See Table 1 for quantitative results.

8. Conclusion

We described a moving object detection system that combines spatio-temporal tensor-based motion estimation with a novel background modeling scheme. Use of tensor-based motion segmentation results in coherent detections robust to noise and illumination artifacts, while the proposed background subtraction process handles detection of static objects. The final multi-cue object level classification distinguishes stopped objects from background revealed by removed objects and thus reduces false positives. We experimentally show that the proposed system outperforms most state-of-the-art methods on the CVPR2014 challenge dataset[7].

References

[1] O. Barnich and M. Van Droogenbroeck. Vibe: A universal background subtraction algorithm for video sequences. *IEEE Trans. Image Proc.*, 20(6):1709–1724, 2011.

[2] Y. Benezeth, P.-M. Jodoin, B. Emile, C. Rosenberger, and H. Laurent. Comparative study of background

	Recall	Spec	FPR	FNR	PWC	F	Prec
KNN[14]	0.6650	0.9802	0.0198	0.3350	3.3200	0.5937	0.6788
GMM1[12]	0.6846	0.9750	0.0250	0.3154	3.7667	0.5707	0.6025
KDE[5]	0.7375	0.9519	0.0481	0.2625	5.6262	0.5688	0.5811
MahaD[2]	0.1644	**0.9931**	**0.0069**	0.8356	3.4750	0.2267	0.7403
GMM2[13]	0.6604	0.9725	0.0275	0.3396	3.9953	0.5566	0.5973
EucD[2]	0.6803	0.9449	0.0551	0.3197	6.5423	0.5161	0.5480
FTSG	**0.7657**	0.9922	0.0078	**0.2343**	**1.3763**	**0.7283**	**0.7696**

Table 1. Quantitative comparison of the proposed FTSG system to several state-of-the-art methods.

	Recall	Spec	FPR	FNR	PWC	F	Prec
Bad Weather	0.7457	0.9991	0.0009	0.2543	0.5109	0.8228	0.9231
Low Framerate	0.7517	0.9963	0.0037	0.2483	1.1823	0.6259	0.6550
Night Videos	0.6107	0.9759	0.0241	0.3893	4.0052	0.5130	0.4904
PTZ	0.6730	0.9770	0.0230	0.3270	2.5519	0.3241	0.2861
Turbulence	0.6109	0.9998	0.0002	0.3891	0.1987	0.7127	0.9035
Baseline	0.9513	0.9975	0.0025	0.0487	0.4766	0.9330	0.9170
Dynamic Background	0.8691	0.9993	0.0007	0.1309	0.1887	0.8792	0.9129
Camera Jitter	0.7717	0.9866	0.0134	0.2283	2.0787	0.7513	0.7645
Intermittent Object	0.7813	0.9950	0.0050	0.2187	1.6329	0.7891	0.8512
Shadow	0.9214	0.9918	0.0082	0.0786	1.1305	0.8832	0.8535
Thermal	0.7357	0.9960	0.0040	0.2643	1.1823	0.7768	0.9088
Overall	0.7657	0.9922	0.0078	0.2343	1.3763	0.7283	0.7696

Table 2. Comparison of the proposed FTSG system on all eleven scenarios using all seven measures.

subtraction algorithms. *Journal of Electronic Imaging*, 19(3):033003–033003–12, 2010.

[3] F. Bunyak, K. Palaniappan, S. K. Nath, and G. Seetharaman. Flux tensor constrained geodesic active contours with sensor fusion for persistent object tracking. *Journal of Multimedia*, 2(4):20, 2007.

[4] F. Bunyak, K. Palaniappan, S. K. Nath, and G. Seetharaman. Geodesic active contour based fusion of visible and infrared video for persistent object tracking. In *IEEE Workshop Applications of Computer Vision (WACV)*, 2007.

[5] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *European Conf. Comp. Vision*. Springer, 2000.

[6] R. H. Evangelio and T. Sikora. Complementary background models for the detection of static and moving objects in crowded environments. In *IEEE Int. Conf. Advanced Video Signal-Based Surveillance*, 2011.

[7] N. Goyette, P. Jodoin, F. Porikli, J. Konrad, and P. Ishwar. Changedetection.net: A new change detection benchmark dataset. In *IEEE Conf. Comp. Vision Pattern Recog. Workshops*, pages 1–8, 2012.

[8] K. Palaniappan, F. Bunyak, S. Nath, and J. Goffeney. Parallel processing strategies for cell motility and shape analysis. In *High-Throughput Image Reconstruction and Analysis*, pages 39–87. Artech House Publishers, 2009.

[9] K. Palaniappan, I. Ersoy, and S. K. Nath. Moving object segmentation using the flux tensor for biological video microscopy. *Lecture Notes in Computer Science (PCM)*, 4810:483–493, 2007.

[10] K. Palaniappan, I. Ersoy, G. Seetharaman, S. Davis, P. Kumar, R. M. Rao, and R. Linderman. Parallel flux tensor analysis for efficient moving object detection. In *14th Int. Conf. Information Fusion*, 2011.

[11] R. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: a systematic survey. *IEEE Trans. Image Proc.*, 14(3):294–307, 2005.

[12] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Conf. Comp. Vision Pattern Recog.*, volume 2, 1999.

[13] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Int. Conf. Pattern Recog.*, volume 2, 2004.

[14] Z. Zivkovic and F. van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recogn. Lett.*, 27(7):773–780, May 2006.