Implementation of Spatio-Temporal Background Subtraction Model

Waribam Jotin Singh
IV Sem , M.tech DCN branch
T. John Institute of Technology
Bangalore, India

Ms Anni Susan Basu
Assistant professor, Dept. of ECE
T. John Institute of Technology
Bangalore, India

***Abstract* - Background subtraction continues to be in open drawback in the context of advanced situations like dynamic backgrounds, illumination variations, and bleary foreground objects. To overcome these challenges, we tend to propose a good background subtraction method by mean of spatial-temporal representations. Within the experiments, I validate the projected methodology in many advanced situations, and show superior performances over different progressive approaches of background subtraction. There area a several challenges in developing a decent background subtraction algorithmic rule. First, it should be strong against changes in illumination. Second, it ought to avoid background objects like swinging leaves, rain, snow, and shadow forged by moving objects. Finally, its internal background model ought to react quickly to changes in background like beginning and stopping of vehicles. During this project, I tend to compare varied background sub-traction algorithms for police investigating moving vehicles and pedestrians in urban trace video sequences. I tend to contemplate variable approaches from easy techniques like frame defacing and adaptive median filtering, to more sophisticated probabilistic modelling techniques.**

Keywords - Spatio-temporal representation, video bricks, background subtraction.

I. INTRODUCTION

Background subtraction or foreground extraction has been extensively studied for years, yet it still remains an open challenge due to the real time applications. Background subtraction is a common method for detecting moving objects, and it has been widely used in many surveillance systems. Moving object detection by mean of background subtraction, compares an input image with a background model previously prepared, and picks up regions in an input image, which do not match a background mode. Moving object detection with background subtraction

has the advantage of not requiring previous knowledge of moving objects. On the other hand, background subtraction has a problem in that it

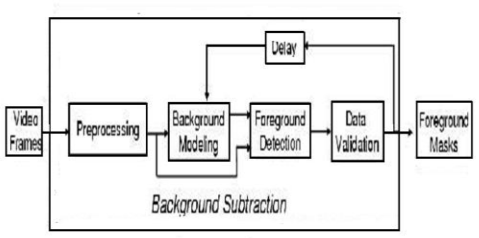

Fig1: Flow diagram for a background subtraction algorithm

Cannot, discriminate moving objects from backgrounds when these backgrounds change significantly. Video surveillance systems tend to automatically identify events of interest in a variety of circumstances. Example applications include intrusion detection, activity monitoring, and pedestrian counting. The capability of extracting moving objects from a video sequence is a fundamental and crucial problem of these vision systems. For systems using static cameras, background subtraction is the method typically used to segment moving regions in the image sequences, by comparing each new frame to a model of the scene background.

In our proposed method, we tend to learn and maintain the dynamic models within spatio-temporal video patches (i.e. video bricks), accounting surveillance scenarios for real challenges. The algorithm can process 15∼ 20 frames per second in the resolution 352 × 288 (pixels) on average. We briefly overview the proposed framework of background modelling in following aspects:

1) Spatio-Temporal Representations: We represent the observed scene by video bricks, i.e. video volumes spanning over both

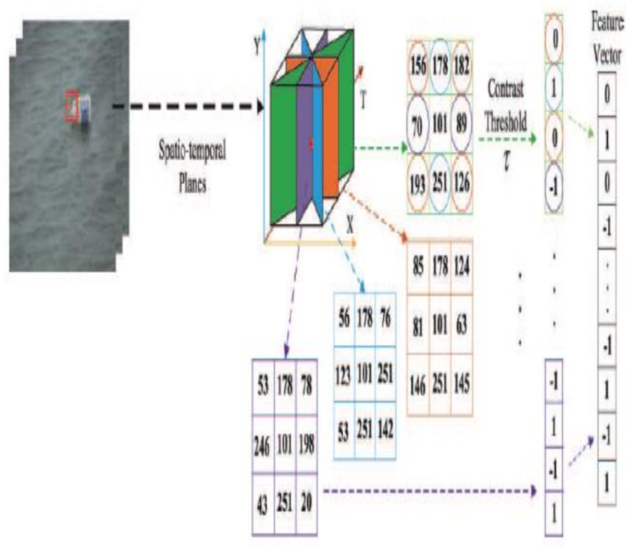

Fig. 2. An example of computing the CS-STLTP feature. For one pixel in the video brick, we construct four spatio-temporal planes. The center-symmetric local ternary patterns for each plane is calculated, which compares the intensities in a center-symmetric direction with a contrasting threshold τ. The CS-STLTP feature is concatenated by the vectors of the four planes.

spatial and temporal domain, for to model spatial and temporal information jointly. Specifically, for every location in the scene, sequence of video bricks are extracted, using which we can learn and update the background models. Moreover, to compactly encode the video bricks against illumination variations, we design a brick- based descriptor, call Center Symmetric Spatio-Temporal Local Ternary Pattern (CS-STLTP), which is inspired by the 2D scale invariant local pattern operator. Its effectiveness is also validated in the experiments.

2) Pursuing Dynamic Subspaces: We treat each video bricks sequence at a certain location as a consecutive signal, and generate the subspace within these video bricks. The linear dynamic system (i.e. Auto Regressive Moving Average, ARMA model) is adopted to characterize the spatio-temporal statistics of the subspace. Specifically, with the observed video bricks, we can express them by a data matrix, in which each column contains the feature of a video brick. The basis vectors (i.e. eigenvectors) of the matrix can be then analytically estimated, representing the appearance parameters of the subspace, and the parameters of dynamical variations are further computed based on the fixed appearance parameters. It is worth mentioning that our background model jointly captures the information of appearance and motion as the data (i.e. features of the video bricks) are extracted over both spatial and temporal domains.

3) Maintaining Dynamic Subspaces Online: Given the newly appearing video bricks with our model, moving fore-ground objects are segmented by estimating the residuals within the related subspaces of the scene, while the back-ground models are maintained simultaneously to account for the scene changes. The raising problem is to update parameters of the subspaces incrementally against disturbance of foreground objects and background noise. The new observation may include noise pixels (i.e. outliers), resulting in degeneration of model updating. Furthermore, one video brick could be partially occluded by foreground objects in our representation, i.e. only some of pixels in the brick are true positives. To overcome this problem, we present a good approach to compensate observations (i.e. the observed video bricks) by generating data from the current models. Specifically, we replace the pixels labelled as non-background by the generated pixels to synthesize the new observations. The algorithm for online model updating includes two steps: (i) update appearance parameters using the incremental sub-space learning technique, and (ii) update dynamical variation parameters by analytically solving the linear reconstruction. The experiments show that the proposed method effectively improves the robustness during the online processing.

II. DYNAMIC SPATIO-TEMPORAL MODEL

In this section, we introduce the background model, and then discuss the video brick representation and our model definition, respectively.

A. Background

In general, a complex surveillance background may include diverse appearances that sometimes move and change randomly and dynamically over time. There is a bunch of works on time-varying texture modeling in computer vision. They often treated the scene as a whole, and pursued a global subspace by utilizing the linear dynamic system (LDS). These models worked well on some natural scenes such as some homogeneous textures, as with a set of linearly combined components the LDS characterizes the subspace. However, under real surveillance challenges, it could be intractable to pursue the global subspace. Here , we represent the observed scene by an array of small and independent subspaces, each of which is defined by the linear system, so be able to handle better challenging scene variations. Our background model can be viewed as a mixed compositional model consisting of the linear subspaces. In particular, we conduct the background subtraction based on the following observations

Assumption 1: The local scene variants (i.e. appearance and motion changing over time) can be captured by the low- dimensional subspace.

Assumption 2: It is feasible to separate foreground moving objects from the scene background by fully exploiting spatio- temporal statistics

B. Spatio-Temporal Video Brick

The surveillance video of one scene is given , it is then decomposed with a batch of small brick-like volumes. The video brick of small size (e.g.,4×4×5 pixels) is considered which includes relative simple content, which can be thus generated by few bases (components). And the brick volume integrates both spatial and temporal information, that we can better capture complex appearance and motion variations compared with the traditional image patch representations. Each frame I_i, (i = 1,2,...,n) is divided into a set of image patches with the width w and height h. A number t of patches at the same location across the frames are combined together to form a brick. In this way, a sequence of video bricks $V = \{v_1, v_2,..., v_n\}$ is extracted at every location for the scene. Moreover, descriptor is design to describe the video brick instead of using RGB values. For any video brick v_i, the CS-STLTP operator is applied on each pixel and sum all the feature values into a histogram. For a pixel x_c, a few 2D spatio-temporal planes is constructed centred at it, and compute the local ternary patterns (LTP) operator on each plane. The CS-STLTP then encodes x_c by combining the LTP operators of all planes. Note that the way of splitting spatio-temporal planes little affects the operator's performance. To simplify the implementation, we make the planes parallel to the Y axis, as Fig. 2 shown. To measure the operator response, we transform the binary vector of CS-STLTP into a uniform value that is defined as the number of spatial transitions (bitwise changes). For example, the pattern (i.e. the vector of 16 bins) 0000000000000000 has a value of 0 and 1000000000000000 of 1. In our implementation, we further quantize all possible values into 48 levels. To further improve the capability, we can generate histograms in each color channel and concatenate them together. The proposed descriptor is computationally efficient and compact to describe the video brick. In addition, by intro-ducing a tolerative comparing range in the LTP operator computation, it is robust to local spatio-temporal noise within a range.

C. Model Definition

Let m be the descriptor length for each brick, and $V = \{v_1,v_2,...,v_n\}$, $v_i \in R_m$ be a sequence of video bricks at a certain location of the observed background. We can use a set of bases (components) $C = [C_1,C_2,...,C_d]$ to represent the subspace where V lies in. Each video brick v_i in V can be represented as

$$v = \sum_{j=1}^{d} z_{i,j} C_j + \omega_i$$

Where, C_j is the j^{th} basis (j^{th} column of matrix C) of the subspace, $z_{i,j}$ the coefficient for C_j, and ω_i the appearance residual. We denote C to represent appearance consistency of the sequence of video bricks. In some traditional background models by subspace learning, zi,j can be solved and kept as a constant, with the underlying assumption that the appear- ance of background would be stable within the observations. In contrast, we treat $z_{i,j}$ as the variable term that can be further phrased as the time-varying state, accounting for temporally coherent variations (i.e. the motions).

The proposed model is time-varying, and the parameters C_n, A_n, B_n can be updated incrementally along with the processing of new observations, in order to adapt our model with scene changes.

III. LEARNING ALGORITHM

In this section, the learning for spatio-temporal background models is discussed, including initial subspace generation and online maintenance. The initial learning is performed at the beginning of system deployment, when only a few foreground objects move in the scene. Afterwards, the system switches to the mode of online maintenance.

A. Initial Model Learning

In the initial stage, the model defined in the above equation can be degenerated as a non-dynamic linear system, as the n observations are extracted and fixed. When a brick sequence $V = \{v_1, v_2,...,v_n\}$ is given, an algorithm to identify the model parameters C_n, A_n, B_n, is presented.

B. Online Model Maintenance

Then we discuss the online processing with our model that segments foreground moving objects and keeps the model updated.

1) Foreground Segmentation:

Given one newly appearing video brick v_{n+1}, we can determine whether pixels in v_{n+1} belong to the background or not by thresholding their appearance residual and state residual. We first estimate the state of v_{n+1} with the existing C_n, and further the appearance residual of $v_{n+1.}$

As the state z_n and the temporal coherence A_n have been solved, we can then estimate the state residual according to system equation.

With the state residual z_{n+1} and the appearance residual ω_{n+1} for the new video brick v_{n+1}, we conduct the following criteria for foreground segmentation, in which two thresholds are introduced.

1) V_{n+1} is classified into background, only if all dimensions of _n are less than a threshold $T_{\omega.}$

2) If v_{n+1} has been labeled as non-background, perform the pixel-wise segmentation by comparing ω_{n+1} with a threshold T_ω: the pixel is segmented as foreground if its corresponding dimension in ω_{n+1} is greater than T_ω.

2) Model Updating:

During the online processing, to deal with foreground disturbance is the key problem for model updating, *i.e.* to avoid absorbing pixels from foreground objects or noise. In this work, we develop an effective approach to update the model with the synthesized data. We first generate a video brick from the current model, namely noise-free brick, v_{n+1}, then we extract pixels from ˆ vn+1 to compensate occluded (*i.e.* foreground) pixels in the newly appearing brick. Concretely, the pixels labeled as non-background are replaced by the pixels from the noise-free video brick at the same place. We can thus obtain a synthesized video brick. v_{n+1} for model updating.

Given the brick, v_{n+1}, the data matrix W_n composed of observed video bricks is extended to W_{n+1}. The, the model $\mathbf{C}_{n+1}$ is update.

IV. EXPERIMENTS

For, to perform simulation, we collected a number of challenging videos to validate our approach. Our algorithm had shown a good result in adapting the complex background changes. The total evaluation of our work will be presented after the full implementation.

The Algorithm for the proposed model is given below

Algorithm 1: The Sketch of the Proposed Algorithm

Input: Video brick sequence V={} for every location for the scene.

Output: Maintained Background models and foreground regions.

```
For all the locations for the scene do
    Given the observed video bricks V, extract the CS-STLTP descriptor
    Initialize the subspace by estimating Cn, An, Bn;
    for the newly appearing video brick $v_{n+1}$ do
        (1) Extract the CS-STLTP descriptor for $v_{n+1}$;
        (2) Calculate its state residual and appearances residual;
        (3) For each pixel of $v_{n+1}$, classify it into foreground or background by thresholding the two residuals with ;
        (4) Generate the noise-free brick $\grave{\upsilon}_{n+1}$ from the current model;
        (5) Synthesize video brick $\bar{\upsilon}_{n+1}$ for model updating;
        (6) Update $\bar{\upsilon}_{n+1}$ into $\tilde{v}_{n+1}$ by introducing a robustness function;.
        (7) Update the new appearance parameter $C_{n+1}$ by calculating the covariance matrix $Cov_{n+1}$ with the learning rate $\alpha$;
        (8) Update the state variation parameters An+1,Bn+1;
    end
end
```

V. OUTPUT

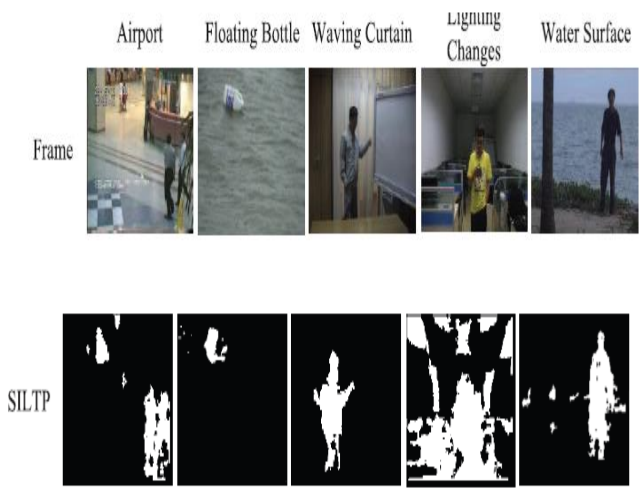

Fig 3: Sampled results of background subtraction generated by our approach

VI. CONCLUSION

By using the above algorithm we have obtain the fig 3 results. The work is still in progress and further work will be done using some edge detection and smoothing filter along with spatio-temporal method to get a more noise free foreground object detection.

REFERENCES

[1] C. Stauffer and W. Grimson , "Adaptive background mixture models for real-time tracking", in Proc. IEEE Conf. CVPR, Jun. 1999.
[2] T. Bouwmans, F. E. Baf, and B. Vachon, " Background modeling using mixture of Gaussians for foreground detection-a survey ", Recent Patents Comput. Sci.,vol. 1, no. 3, pp. 219–237, 2008.
[3] L. Maddalena and A. Petrosino, "A self-organizing approach to back-ground subtraction for visual surveillance applications," IEEE Trans. Image Process., vol. 17, no. 7, pp. 1168–1177, Jul. 2008.
[4] D. M. Tsai and S.-C. Lai, "Independent component analysis-based background subtraction for indoor surveillance", IEEE Trans. Image Process., vol. 18, no. 1, pp. 158–167, Jan. 2009.
[5] H. Chang, H. Jeong, and J. Choi, "Active attentional sampling for speed- up of background subtraction", in Proc. IEEE Conf. CVPR, Jun. 2012, pp. 2088–2095.
[6] X. Liu, L. Lin, S. Yan, H. Jin, and W. Tao, "Integrating spatio-temporal context with multiview representation for object recognition in visual surveillance", IEEE Trans. Circuits Syst. Video Technol., vol. 21, no. 4, pp. 393–407, Apr. 2011.
[7] L. Lin, Y. Lu, Y. Pan, and X. Chen, "Integrating graph partitioning and matching for trajectory analysis in video surveillance," IEEE Trans. Image Process., vol. 21, no. 12, pp. 4844–4857, Apr. 2012.