Proc. of CVPR Workshop Object Tracking and Classification Beyond the Visible Spectrum,
Washington, July 2004

Motion Detection Based on Local Variation of Spatiotemporal Texture

Longin Jan Latecki[1], Roland Miezianko[1], Dragoljub Pokrajac[2]
[1]Temple University, CIS Dept., Philadelphia, PA, latecki@temple.edu, rmiezian@temple.edu
[2]Delaware State University, CIS Dept., Dover, DE, dpokraja@desu.edu

Abstract

In this paper we propose to use local variation of spatiotemporal texture vectors for motion detection. The local variation is defined as the largest eigenvalue component of spatiotemporal (sp) texture vectors in certain time window at each location in a video plane.

Sp texture vectors are computed using a dimensionality reduction technique applied to spatiotemporal (3D) blocks. They provide a compact vector representation of texture and motion patterns for each block. The fact that we go away from the standard input of pixel values and instead base the motion detection on sp texture of 3D blocks, significantly improves the quality of motion detection. This is particularly relevant for infrared videos, where pixel values have smaller range than in daylight color or gray level videos.

Keywords. Video analysis, video mining, surveillance videos, distribution learning, motion detection.

1. Introduction

Let us focus on a fixed position in a video image and observe the sequence of visual vectors (e.g., gray level, infrared, color, or texture vectors) at this location in a video plane over time. We assume a stationary camera. If we observe a part of the scene background at this location, then clearly we will have only very small variation of visual vectors over time at this location due to slight illumination changes and errors of the video capture device. On the other hand, if a moving object is passing through this location, then due to this motion we will see different parts of the object, which are very likely to have different texture. Therefore, the texture at a given location is very likely to highly vary.

If at the observed position in the video plane the values of only a single pixel are considered, then the variation alone might not be sufficient for proper identification of movement. For example, consider a white object moving through an observed pixel location. At the edge of this object, the variation will be high, but it may be low in the interior of the object. Thus, by detecting high variation we will identify the edge but not the inner parts of a moving object. The solution to this problem proposed by Stauffer and Grimson [14] was to, in addition to covariance matrix, also consider the mean of the color values vector, which leads to the Gaussian mixture model.

In this paper, instead of color or infrared values at pixel locations, we consider the values of all pixels in spatiotemporal regions represented as 3D blocks. To compactly represent these values and to reduce the influence of noise, we introduce a spatiotemporal texture representation of the 3D blocks. This texture representation is the input to the proposed motion detection technique based on local variation. Thus, we go away from the standard input of pixel values for motion detection that are known to be noisy and the main cause of instability of video analysis algorithms.

We decompose a given video into 3D spatiotemporal (sp) blocks (e.g., 8x8x3 blocks) and apply a dimensionality reduction technique to obtain a compact representation of color, infrared, or gray level values at each block (as vector of just a few components). The obtained sp texture vectors provide a joint representation of texture and motion patterns in videos and are used as primary input elements to video analysis algorithms.

The power of dimensionality reduction techniques to compactly represent 3D blocks has already been recognized in video compression. There, 3D discrete cosine and 3D wavelet transforms are employed to reduce the color or gray level values of a large number of pixels in a given block to a few quantized vector components, e.g., [15]. However, these techniques are not particularly suitable for detecting moving objects, since the obtained components do not necessarily provide good means to differentiate the texture of the blocks. Namely, these transformations are context free and intrinsic in that their output depends only on a given input 3D block. In contrast, we propose to use a technique that allows us to obtain an optimal differentiation for a given set of 3D blocks. To reach this goal, we need an extrinsic and context sensitive transformation such that a representation of the given block depends on its context—the set of other 3D blocks in a given video. The applied Principal Component Analysis (PCA) [8] satisfies these requirements. Namely, for a given set of 3D blocks PCA assigns to each block a vector of the components that maximize the differences among the blocks. Consequently, PCA components are very suitable to detect changes in 3D blocks.

In our previous paper [11] we have shown that the use of sp texture vectors of spatiotemporal blocks in the framework of Stauffer and Grimson [14] can improve the detection of moving objects while potentially cutting back the processing time due to the reduction of the number of input vectors per frame. In this paper, we propose a novel motion detection technique that can lead to further performance improvements.

As we mentioned above, texture at a given location in video plane is very likely to highly vary when a moving object is passing through this location. Therefore, we propose to base motion detection on local variation of sp texture vectors. The question arises how to robustly measure this variation. We definitely need to measure it in a limited and as short as possible window of time, since at a given position a moving object can quickly appear or disappear. We propose to define the *local variation* as the largest eigenvalue of sp texture vectors in a small time window. It is computed by applying PCA to the covariance matrix of the sp texture vectors within a small temporal window. Thus, in the proposed approach we use PCA twice, first time to compute the sp texture vectors, and the second time to compute the variation of a set of texture vectors in a given time window. The decision whether a moving object or a stationary background is identified at a given spatiotemporal location is then made by dynamic thresholding of the obtained eigenvalues.

2. Related work

A good overview of the existing approaches to motion detection can be found in the collection of papers edited by Remagnino et al. [13] and in the special section on video surveillance in IEEE PAMI edited by Collins et al. [2]. A common feature of the existing approaches for moving objects detection is the fact that they are pixel based. Some of the approaches rely on comparison of color or intensities of pixels in the incoming video frame to a reference image. Jain et al. [7] use simple intensity comparison to reference images so that the values above a given threshold identify the pixels of moving objects. A large class of approaches is based on appropriate statistics of color or gray values over time at each pixel location. (e.g., the segmentation by background subtraction in W4 [6], eigenbackground subtraction [10], etc). Wren et al. [16] were the first who used a statistical model of the background instead of a reference image.

One of the most successful approaches for motion detection was introduced by Stauffer and Grimson [14]. It is based on adaptive Gaussian mixture model of the color values distribution over time at each pixel location. Each Gaussian function in the mixture is defined by its prior probability, mean and a covariance matrix. In this paper we show that the proposed local variation is not only a much simpler but also a more adequate model for motion detection for infrared videos. It can significantly reduce the processing time in comparison to the Gaussian mixture model, due to smaller complexity of the local variation computation, thus making the real time processing of high-resolution videos as well as efficient analysis of large-scale video data viable. Moreover, the local-variation based algorithm remains stable with higher dimensions of input data, which is not necessarily the case for an EM type algorithm (used for Gaussian model estimation). This makes the proposed technique potentially appealing for moving detection in higher dimensional domains, such as multispectral remote sensing imagery.

As argued in [9], the application of region level techniques can lead to increased stability when detecting objects in adverse conditions. However, [9] and related approaches (e.g., [1]) aimed to improving the Stauffer-Grimson algorithm [14] still perform motion detection on pixel level (i.e., only the postprocessing of pixel-based motion detection results is region based). In contrast, the motion detection in the proposed approach is solely region-based.

3. Proposed methodology

3.1. Video representation with spatiotemporal (sp) texture vectors

We represent videos as three-dimensional (3D) arrays of gray level or monochromatic infrared pixel values $g_{i,j,t}$ at a time instant t and a pixel location i, j. A video is characterized by temporal dimension Z corresponding to the number of frames, and by two spatial dimensions, characterizing number of vectors in horizontal and vertical direction of each frame. We divide each image in a video sequence into disjoint $N_{BLOCK} \times N_{BLOCK}$ squares (e.g., 8x8 squares) that cover the whole image. Spatiotemporal (3D) blocks are obtained by combining squares in consecutive frames at the same video plane location. In our experiments, we used 8x8x3 blocks that are disjoint in space but overlap in time, i.e., two blocks at the same spatial location at times t and $t+1$ have one square in common. The fact that the 3D blocks overlap in time allows us to perform successful motion detection in videos with very low time frequency, e.g., in our experimental results [12] videos with 1 fps (frame per second) are included. The obtained 3D blocks are represented as 192 dimensional vectors of gray level or monochromatic infrared pixel values. We then zero mean these vectors and project them to 3 dimensions using PCA. The obtained 3 dimensional vectors form a compact spatiotemporal texture representation for each block. The PCA projection matrices are computed separately for all video plane locations (a set of disjoint 8x8 squares in our experiments).

A more detailed explanation follows now. The blocks are represented by N-dimensional vectors $\mathbf{b}_{I,J,t}$, specified by spatial indexes (I,J) and time instant t. Vectors $\mathbf{b}_{I,J,t}$ contain all values $g_{i,j,t}$ of pixels in the corresponding 3D block. Thus, for a given block location specified by spatial indexes (I,J) and time instant t, the corresponding block

vector $\mathbf{b}_{I,J,t}$ contains pixel values $g_{i,j,t}$ from spatial locations with coordinates
$(N_{BLOCK}-1)\times(I-1)+1, \ldots, N_{BLOCK}\times I$;
$(N_{BLOCK}-1)\times(J-1)+1, \ldots, N_{BLOCK}\times J$
and from frames $t-T,\ldots, t+T$.
Observe that the length N of the block vector $\mathbf{b}_{I,J,t}$ is equal to $N_{BLOCK}\times N_{BLOCK}\times (2T+1)$..

To reduce dimensionality of $\mathbf{b}_{I,J,t}$ while preserving information to the maximal possible extent, we compute a projection of the normalized block vector to a vector of significantly lower length $K<<N$ using a PCA projection matrix $\boldsymbol{P}^{K}_{I,J}$ computed for all $\mathbf{b}_{I,J,t}$ at video plane location (I,J). The resulting sp texture vectors $\mathbf{b}^{*}_{I,J,t} = \boldsymbol{P}^{K}_{I,J} * \mathbf{b}_{I,J,t}$ provide a joint representation of texture and motion patterns in videos and are used as input of algorithms for detection of moving objects.

To compute $\boldsymbol{P}^{K}_{I,J}$ we employ the principal values decomposition following [4,5]. A matrix of all normalized block vectors $\mathbf{b}_{I,J,t}$ at video plane location (I,J) is used to compute the $N\times N$ dimensional covariance matrix $\boldsymbol{S}_{I,J}$. The PCA projection matrix $\boldsymbol{P}_{I,J}$ for spatial location (I,J) is computed from the $\boldsymbol{S}_{I,J}$ covariance matrix. The projection matrix $\boldsymbol{P}_{I,J}$ of size $N\times N$ represents N principal components. By taking only the principal components that corresponds to the K largest eigenvalues, we obtain $\boldsymbol{P}^{K}_{I,J}$.

3.2. Moving objects detection based on local variation

The assumption of the proposed technique is that the variation of location vectors—corresponding to the same location within a small number of consecutive frames—will increase if the vectors correspond to a moving object. In practice, for each location (x,y), we consider vectors $\mathbf{v}_{x,y,t-W}, \mathbf{v}_{x,y,t-W+1}, \ldots, \mathbf{v}_{x,y,t}, \ldots, \mathbf{v}_{x,y,t+W}$ corresponding to a symmetric window of size $2W+1$ around the temporal instant t, where $\mathbf{v}_{I,J,t} = \mathbf{b}^{*}_{I,J,t}$ are the sp texture vectors. For these vectors, we compute the covariance matrix $\mathbf{C}_{x,y,t}$. We assign to a given spatiotemporal video position a local *variance measure,* which we will also refer to as *motion measure*,

$$mm(x,y,t) = \Lambda_{x,y,t}$$

where $\Lambda_{x,y,t}$ is the largest eigenvalue of $\mathbf{C}_{x,y,t}$. The larger the variance measure $mm(x,y,t)$, the more likely is the presence of a moving object at position (x,y,t).

Finally, we label each video position as moving or stationary (background) depending whether the motion measure is larger or smaller than a suitably defined threshold. We use a dynamic thresholding algorithm to determine the threshold value at position (x,y,t) based on the history of $mm(x,y,s)$ values over time ($s=1, \ldots, t-1$). Since $mm(x,y,t)$ is a 1D function of t for fixed (x,y) (see Fig. 1), the tasks reduces to simple analysis of the graph of this function. First we compute mean *meanl* and standard deviation *stdl* (using a running average) of all previous $mm(x,y,s)$ for $s=1, \ldots, t-1$ that were labeled as stationary. A moving object is detected if

$meanrw - meanl > C1*stdl$, where $C1$ is a constant and

$$meanrw = \frac{1}{w}\sum_{\tau=1}^{w} mm(x,y,t+\tau) \cdot$$

Once motion is detected, we switch to a stationary state if

$$meanrw - meanl < C2*stdl,$$

where $C2 < C1$ is a second constant.

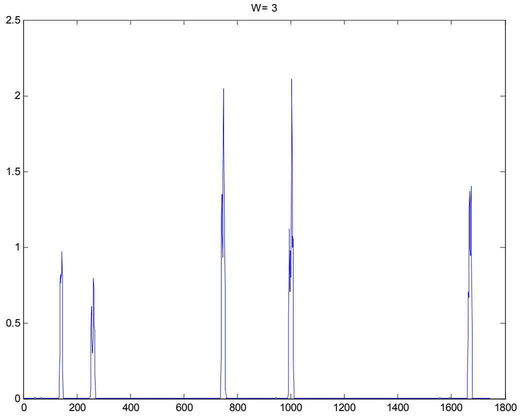


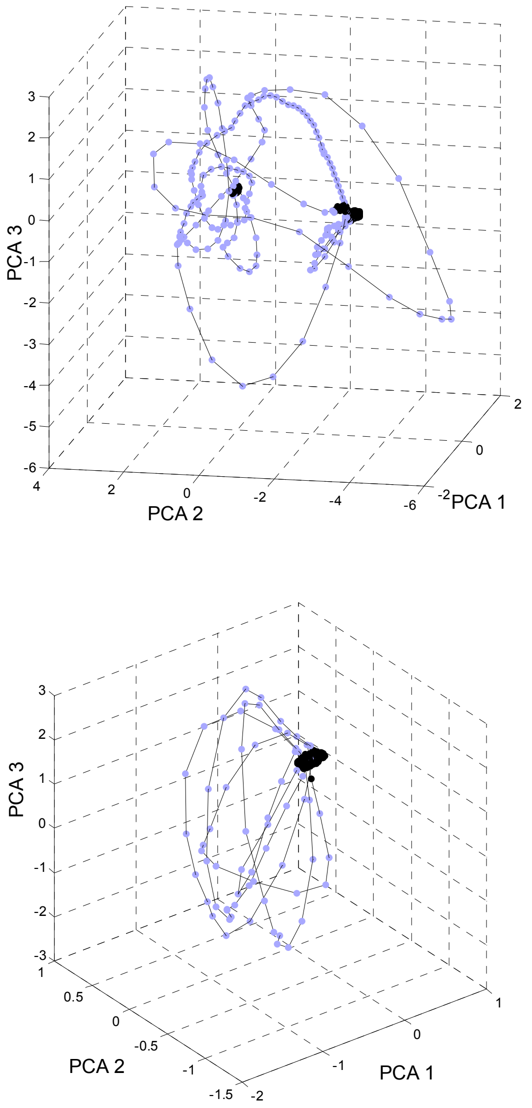
Figure 1. The graph of local variance mm over time.

4. Performance evaluation with motion orbits

The most common method to evaluate the performance of motion detection is simply to view the videos with moving objects marked by the applied algorithm as we discuss in Section 5. However, in our framework a more objective method of performance evaluation is also possible. In this section we introduce and use such a method to compare the proposed local-variation technique to the improved version of the incremental EM algorithm in [11]. Both compared techniques are based on the same spatiotemporal blocks that represent texture and motion patterns.

Recall that with the local-variation based technique we perform the detection of moving objects using the first 3 PCA components of each spatiotemporal block vector. We define a *motion orbit* as path that the vector of the PCA components, corresponding to a particular location in the video plane, traverses over time. In other words, the motion orbit at video plane location (x,y) is a sequence of points in the 3D Euclidean space $\mathbf{v}_{x,y,1}, \mathbf{v}_{x,y,2}, \ldots, \mathbf{v}_{x,y,T}$, where $\mathbf{v}_{I,J,t} = \mathbf{b}^{*}_{I,J,t}$ and T is the total number of frames.

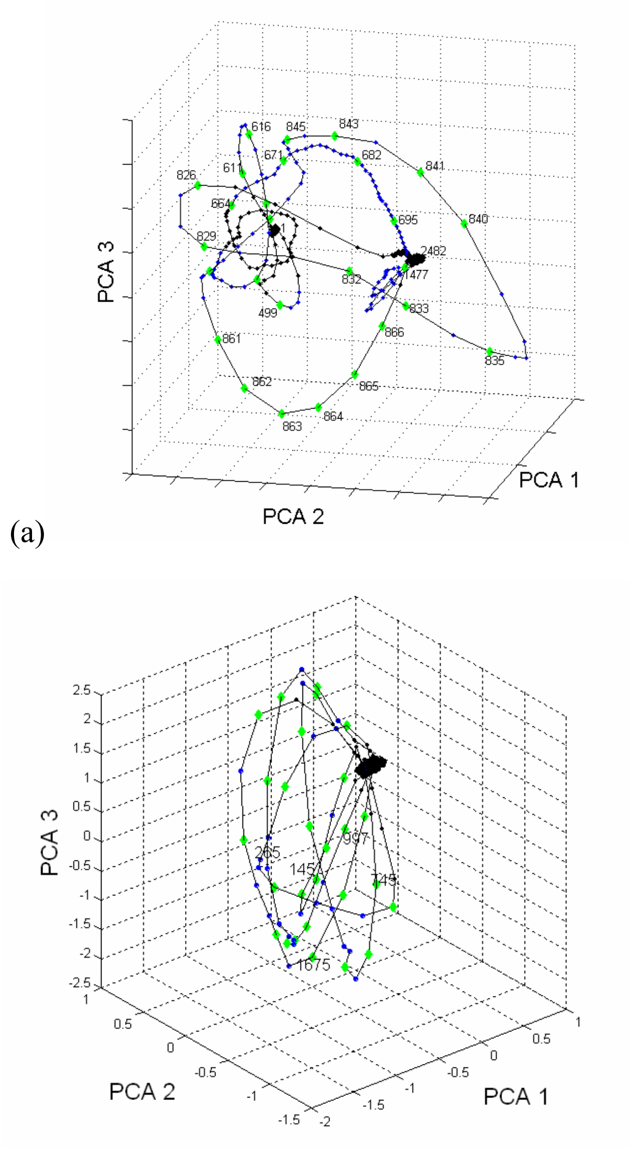
For instance, in Fig. 2a, we see the orbit for the block (*24,28*) of the *Outdoor video* (described in Section 5). Frames identified as moving using our local variation method are marked with blue-gray dots while stationary frames are marked with black dots. The distribution of black dots is multimodal globally. We observe two main modes that represent the background blocks (marked with black dots): one corresponding to the frames at the beginning, and another to the frames at the end of movie. They are identified as two 3D blobs that correspond to two different background textures that appeared in the course of this video at block position (*24,28*). Around these blobs we see 1D orbits marked with blue-gray dots. Therefore, we can view the proposed local variance method as orbit classification algorithm. The reason is that the elongated 1D orbits that identify motion have higher variation than the stationary background objects.



(a)

(b)

Figure 2. Orbits of block vectors with blue-gray dots corresponding to the frames where the block was identified as moving by the proposed method;
a) *Outdoor Video*: block *I*=24, *J*=28;
b) *Indoor Video*: block *I*=7, *J*=25.



(b)

Figure 3. Orbits of block vectors marked with dots: black as background, blue and green as moving—using 'reset' and 'hold' mechanisms, correspondingly, identified by the incremental EM algorithm [11];
a) *Outdoor Video* block *I*=24, *J*=28;
b) *Indoor Video:* block *I*=7, *J*=25.

We stress that the dot labeling as shown was computed by the proposed method. Observe that the blue-gray dots perfectly correspond to the 1D motion orbits that identify moving blocks. Thus, our algorithm correctly detected moving objects. In contrast, for the same *Outdoor video* the incremental EM method [11] failed to identify the motion orbit containing frames 633—663. In Fig. 3a this orbit is labeled with black dots that correspond to falsely identified stationary (background) blocks. For the rest of the orbits the incremental EM is generally able to identify the "distributional outliers" that correspond to the moving objects (marked with green and blue-gray dots depending on whether the reset or hold mechanisms applied [11]).

In Fig. 2b we see the orbit for block location (7,*25*) in *Indoor Video* (described in Section 5). Here we observe only one category of background texture represented by a single cluster of black dots. Again the proposed method was more successful in identifying moving blocks that EM method. As can be seen in Fig. 3b, the EM-based technique had difficulties in correctly labeling moving blocks belonging to orbits close to the bulk of background distribution.

In comparison to any pixel-based approach (e.g., [14]), motion detection based on 3D blocks performs better since it reduces noise in background and can extract information about temporal change of texture (since it is based on spatiotemporal texture representation of 3D blocks instead of pixels). This fact is particularly important for infrared videos, where the range of infrared values of pixels is much smaller than in a day light gray level or color videos. Therefore, the usage of spatiotemporal texture information is even more beneficial here.

We demonstrate how noisy RGB color values of a single pixel can be in Fig. 4, where we plot an orbit over time of RGB color values that occur at the pixel (*185,217*) which is one of the pixels in the block *(24,28)* of *Outdoor video*. For better visualization, in Fig. 4 we show the linearly transformed space of PCA projections of the original RGB color values (the trajectory in the space of original RGB colors is similar). To allow us a proper comparison to the results in Fig. 2a (computed by our local variance technique), we carried over the dot labels from Fig. 2a (where blue-gray dots identify moving blocks). By comparison of Fig. 4 to Fig. 2a, one can conclude that in both representations there are two distribution components corresponding to the background. However, using the block-based approach, the background variance is much smaller, since using block vectors that contain texture information results in effective noise reduction in comparison to using "raw" pixels. Hence, any technique to detect moving objects as outliers will perform much better using spatiotemporal blocks than when using the raw pixels.

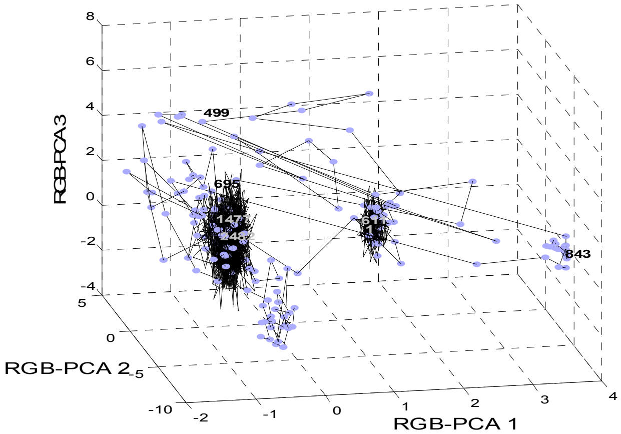


Figure 4. Standardized PCA components of RGB pixel values for *Outdoor Video* at pixel location *(185,217)* that is inside of block *(24,28)*. To allow a direct comparison to Fig. 2a, we carried over the colors of dots; black is background and blue-gray is moving.

As it can be seen in Fig. 4, the method from [14] have difficulties in properly detecting frames 611, 695, 1477 belonging to the second and fourth moving objects that appear at the observed pixel. There the blue-gray dots incorrectly become parts of two background components, which means that a pixel-based method would classify the corresponding blue-gray dots as background.

The proposed local variation based technique can also provide satisfactory results on pixel level, thus providing a viable alternative to the much more complex approach [14]. However, due to problems with large uniform texture regions as well as noise inherent to pixel values (shown above), local variance method based on sp block texture is our preferred technique.

5. Performance evaluation on test videos

A set of several videos showing our motion detection results can be viewed on [12]. This set includes infrared videos, for which the same settings of parameters as for visual light videos were used. Here we focus on our results on two video sequences from the Performance Evaluation of Tracking and Surveillance (PETS) repository: a sequence from PETS2001[1] here referred to as the *Outdoor Video* sequence and a sequence from PETS2002[2] here referred to as the *Indoor Video* sequence.

The parameter settings are described now. For the spatial-temporal blocks, we used T=1 and N_{BLOCK} = 8, such that the length of a block vectors $\mathbf{b}_{I,J,t}$ was N = 192 =

[1] ftp://pets.rdg.ac.uk/PETS2001/DATASET1/TESTING/CAMERA1_JPEGS/

[2] ftp://pets.rdg.ac.uk/PETS2002/PEOPLE/TESTING/DATASET2/

8×8×3. When applying local variation-based technique, it is not necessary to *precisely* estimate the covariance matrix $\mathbf{C}_{x,y,t}$, since the method is based primarily on the value of its largest eigenvalue $\Lambda_{x,y,t}$. Hence, we can employ the estimation window of a relatively small size. In this study, we set W=3 that led to the window size of 2W+1=7. We used the transformed block vectors $\mathbf{b}^{*}_{I,J,t}$ with K = 3 components such that the performed PCA projection preserved more than 89% of the block vectors variance.

To further justify that the proposed method based on the variation in a local window is not only a much simpler but at the same time more robust, let us consider Fig. 5. It shows moving blocks detected by the local variation method and by reset and hold mechanisms of the incremental EM algorithm [11] at the block position (*24,28*) of *Outdoor video*. As we can see, the 'reset' mechanism typically triggers the sequence of moving blocks being identified by 'hold' mechanism. Resets are relatively infrequent for slow-moving objects and the major mechanism to detect blocks corresponding to such moving objects is hold (e.g. frames 1477–1500). However, for fast moving objects (such is the car at frames 608—708 and a van at frames 821—874) the reset mechanism frequently interweaves with the hold. When the moving objects consist of large uniform-textured areas, neither mechanism might be capable of identifying movement, which leads to false detection of background. In contrast, the local variance technique is more robust and in this case correctly identifies motion for the whole duration of the frame intervals 492—512, 608—708, and 821—874.

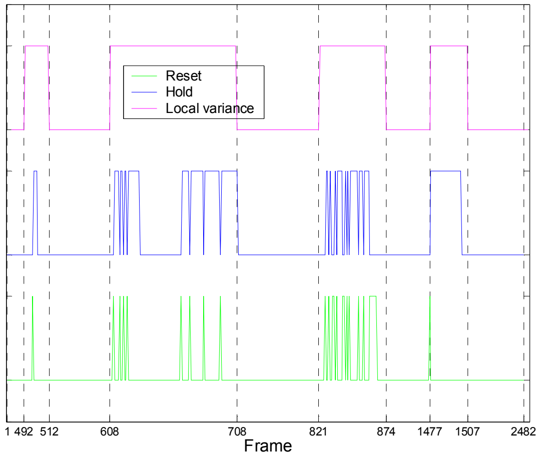


Figure 5. Frames identified as moving at block *I*=24, *J*=28 of the *Outdoor Video* sequence using the proposed local variance technique in comparison to the reset and hold mechanisms of the incremental EM algorithm.

6. Conclusions and work in progress

In this paper we propose a local variation based method for motion detection. Our preliminary results on infrared videos and on PETS repository videos show that the proposed method applied to spatiotemporal blocks results in better detection of moving objects in comparison to standard pixel-based techniques and to the incremental EM algorithm technique.

The proposed texture representation of spatiotemporal blocks can potentially be very useful in object tracking. While tracking an object, we can simultaneously learn the distribution of its blocks. Subsequently, while performing the proposed moving object detection, we can improve the tracking performance, since we perform unsupervised object segmentation. Observe that we would profit here from the fact that our underlying representation is based on texture of 3D blocks as compared to the existing approaches solely based on pixel values.

7. Acknowledgements

We would like to thank the reviewers for their helpful comments. D. Pokrajac has been partially supported by NIH-funded Delaware Biomedical Research Infrastructure Network (BRIN) Grant (P20 RR16472), and DoD HBCU/MI Infrastructure Support Program (45395-MA-ISP Department of Army).

8. References

[1] Buttler, D., Sridharan, S., and Bove, V. M. Real-time adaptive background segmentation. *In Proc. IEEE Int. Conf. on Multimedia and Expo (ICME)*, Baltimore 2003.

[2] R.T. Collins, A.J. Lipton, and T. Kanade, "Introduction to the Special Section on Video Surveillance", *IEEE PAMI* 22(8) (2000), pp. 745–746.

[3] Devore, J. L., *Probability and Statistics for Engineering and the Sciences*, 5th edn., Int. Thomson Publishing Company, Belmont, 2000.

[4] Duda, R., P. Hart, and D. Stork, *Pattern Classification,* 2nd edn., John Wiley & Sons, 2001.

[5] Flury, B. A *First Course in Multivariate Statistics*, Springer Verlag, 1997.

[6] I. Haritaoglu, D. Harwood, and L. Davis, "W4: Real-Time Surveillance of People and Their Activities", *IEEE PAMI* 22(8) (2000), pp. 809–830.

[7] Jain, R., Militzer, D., and Nagel, H. Separating nonstationary from stationary scene components in a sequence of real world TV images. In Proc. IJCAI, 612–618, Cambridge, MA, 1977

[8] Jolliffe, I. T, *Principal Component Analysis*, 2nd edn., Springer Verlag, 2002.

[9] Javed, O., Shafique, K., and Shah, M. A. Hierarchical approach to robust background subtraction using color and gradient information. In Proc. IEEE Workshop on Motion and Video Computing (MOTION), 22-27, Orlando, 2002,.

[10] N. M. Oliver, B. Rosario, and A. P. Pentland, "A Bayesian Computer Vision System for Modeling Human Interactions", *IEEE PAMI* 22(8) (2000), pp. 831–843.

[11] D. Pokrajac and L. J. Latecki: Spatiotemporal Blocks-Based Moving Objects Identification and Tracking, *IEEE Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, October 2003.

[12] R. Miezianko, L. J. Latecki, D. Pokrajac. Link to test results. *http://knight.cis.temple.edu/~video/VA*

[13] Remagnino, P., G. A. Jones, N. Paragios, and C. S. Regazzoni, eds., *Video-Based Surveillance Systems*, Kluwer Academic Publishers, 2002.

[14] C. Stauffer, W. E. L. Grimson, "Learning patterns of activity using real-time tracking", *IEEE PAMI* 22(8) (2000), pp. 747–757.

[15] Westwater, R., Furht, B., *Real-Time Video Compression: Techniques and Algorithms*, Kluwer Academic Publishers, 1997.

[16] C. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland, "Pfinder: Real-time Tracking of the Human Body", *IEEE PAMI* 19(7) (1997), pp. 780–785.