IEEE Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS), October 2003

Spatiotemporal Blocks-Based Moving Objects Identification and Tracking

Dragoljub Pokrajac[1], Longin Jan Latecki[2]
[1]*Delaware State University, CIS Dept., Dover,* [2]*Temple University, CIS Dept., Philadelphia*
pokie@ist.temple.edu, latecki@temple.edu

Abstract

In this paper we propose a new representation of videos with spatiotemporal blocks. After a given video is decomposed into the spatiotemporal blocks, a dimensionality reduction technique is applied to obtain a compact vector representation of each block gray level values. The block vectors provide a joint representation of texture and motion patterns in videos. Our results on PETS repository videos show that detection and tracking of moving objects is substantially improved if based on spatiotemporal blocks instead on pixels. Thus, we go away from the standard input of pixel values that are known to be noisy and the main cause of instability of video analysis algorithms.

1. Introduction

The main contribution of this paper is a new representation of videos with 3D blocks. We first decompose a given video into spatiotemporal blocks, e.g., 8x8x3 blocks. We then apply a dimensionality reduction technique to obtain a compact representation of color or gray level values of each block as vector of just a few numbers. The block vectors provide a joint representation of texture and motion patterns in videos.

We propose the use of 3D block vectors as primary input elements to video analysis algorithms. Thus, we go away from the standard input of pixel values that are known to be noisy and the main cause of instability of video analysis algorithms. Our results show that detection of moving object is substantially improved if it is based on spatiotemporal blocks instead on pixels (which are the main input nowadays). We use the principal component analysis [8] to reduce the dimensionality of the 3D blocks. Since each 3D block is represented as vector of a few real numbers, we significantly improve the performance of video analysis algorithms. At the same time we substantially reduce the processing time, thus making the real time processing of high-resolution videos as well as efficient analysis of large-scale video data possible.

We also demonstrate that the proposed representation of videos using spatiotemporal blocks yields improved tracking results.

2. Related work

The research on motion detection belongs to the field of computer vision. A good overview of the existing approaches can be found in the collection of papers edited by Remagnino et al. [14] 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 are based on comparison of color or intensities of pixels in the incoming video frame to a reference image. Jain et al. [7] used 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. For example, this is the case for the segmentation by background subtraction in W4 [6] and for the eigenbackground subtraction [12]. Wren et al. [18] were the first who used a statistical model of the background instead of a reference image.

One of the most successful of these approaches, introduced by Stauffer and Grimson [17], is based on adaptive Gaussian mixture model of the color values distribution over time at a given pixel location. We adopted this approach in our proposal, but with a major difference that our computation is based on the spatiotemporal blocks. This not only has a positive effect on the reduction of the computing requirements but also primarily leads to increased stability. As stated in [9],
"We also note that only pixel level processing is not sufficient for the extraction of foreground from an image sequence. Thus, higher level processing is required to build upon the information obtained from pixel level processing. We use a bottom-to-top hierarchical processing that consists of three different levels, i. Pixel level, ii. Region level, and iii. Frame level."

We completely agree with this statement. The novelty of our approach is based on the fact that we combine the pixel and region levels to a single level texture representation with 3D blocks. This means that we apply Gaussian mixture model to the spatiotemporal blocks, whereas it has been applied on pixel level in [9]. In contrast, other proposed improvements of the approach presented in [17] (e.g., [1]) are all based on motion detection on the pixel level. Furthermore, we introduce several significant improvements, also applicable on pixel

level, to the process of motion detection of Stauffer and Grimson [17].

3. Methodology

3.1. Spatiotemporal blocks

The first task in our approach to video analysis is dimensionality reduction of spatiotemporal blocks. This task is performed on original (non-processed) videos. We treat a given video as three-dimensional (3D) array of gray pixels $p_{i,j,z}$, i=1,…,X; j=1,…,Y; z=1,…,Z with two spatial dimensions X, Y and one temporal dimension Z. The initial step in the proposed approach is creation of block vectors. In general, we propose the use of spatiotemporal (tree-dimensional) blocks represented by N-dimensional vectors $\mathbf{b}_{I,J,t}$, where a block spans (2T+1) frames and contains N_{BLOCK} pixels in each spatial direction per frame. Hence, N=(2T+1)× N_{BLOCK}× N_{BLOCK}.

For a given block location specified by spatial indexes (I,J) and time instant t, the corresponding block vector contains pixel values from spatial locations bounded by coordinates (N_{BLOCK}-1)×(I-1)+1, N_{BLOCK}×I, (N_{BLOCK}-1)×(J-1)+1, N_{BLOCK}×J and from frames t-T, t-T+1,…,t+T. Hence the block vectors can be defined formally as:

$$\mathbf{b}_{I,J,t} = [p_{i,j,z}]_{\substack{i=(N_{BLOCK}-1)\times(I-1)+1,\ldots,N_{BLOCK}\times I \\ j=(N_{BLOCK}-1)\times(J-1)+1,\ldots,N_{BLOCK}\times J \\ z=t-T,\ldots,t+T}}.$$

Observe that the length of the block vector is proportional to the square of linear block size N_{BLOCK}. To reduce dimensionality of $\mathbf{b}_{I,J,t}$ while preserving information to the maximal possible extent, we compute a projection of the original block vector to a vector of significantly lower length $N'<<N$ using principal component analysis of the sample of the block vectors [8]. For example, we may project the vectors of 8×8×3 blocks to vectors of tree components. The resulting transformed block vectors $\mathbf{b}^*_{I,J,t}$ provide a joint representation of texture and motion patterns in videos. More precisely, using a representative sample of block vectors corresponding to the considered types of movies, we first compute N-dimensional mean vector $\mathbf{m}$ and N×N dimensional covariance matrix $\mathbf{S}$. Next, eigenvalues and eigenvectors of the covariance matrix $\mathbf{S}$ are computed and sorted with respect to decreasing eigenvalues. The N×N' projection matrix $\mathbf{P}$ is created to contain N' eigenvectors $\mathbf{e}_1,\ldots,\mathbf{e}_{N'}$ corresponding to the largest eigenvalues $\lambda_1,\ldots,\lambda_{N'}$ such that $\mathbf{P}=[\mathbf{e}_1\ldots\mathbf{e}_{N'}]$. To compute the transformed block vector $\mathbf{b}^*_{I,J,t}$, following the procedure described in [4], we subtract the mean vector $\mathbf{m}$ from block vector $\mathbf{b}_{I,J,t}$ and multiply the difference by the projection matrix $\mathbf{P}$, so that the coordinates of the block vector become decorrelated after the transformation. Observe that the algorithm for moving block detection (proposed in the next section) implicitly assumes equal standard deviations of the coordinates. That is why we further divide each component of the transformed block vector by the square root of the corresponding eigenvalue, so that each coordinate of the resulting N' dimensional vectors $\mathbf{b}^*_{I,J,t}$ has unit standard deviation. Using the matrix notation, the computation of the transformed block vector $\mathbf{b}^*_{I,J,t}$ can be described as:

$$\mathbf{b}^*_{I,J,t} = \left(\mathbf{b}_{I,J,t} - \mathbf{m}\right)\times \mathbf{P} \times \begin{bmatrix} 1/\sqrt{\lambda_1} & 0 & \ldots & 0 \\ 0 & 1/\sqrt{\lambda_2} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 1/\sqrt{\lambda_{N'}} \end{bmatrix}$$

3.2. Detection of moving blocks

The main step of our technique for detection of moving objects is an incremental algorithm for learning parameters of data distribution associated with each spatial location of a block, specified by indices I, J.

The proposed algorithm is a variant of the incremental EM algorithm for estimating the Gaussian mixtures in Stauffer und Grimson [17] extended by additional mechanism for detecting blocks corresponding to moving objects. The mixture consists of K components, and each component is specified by its estimated mean vector $\boldsymbol{\mu}_k$, a diagonal N'×N' covariance matrix $diag\left(\sigma_k^2,\sigma_k^2,\ldots,\sigma_k^2\right)$, and a distributional prior w_k, k=1,…,K.

As a generalization of the distance criterion proposed in [17], at each time instant t (corresponding to a frame number) we compute the squared Mahalanobis distances [4] $d_k\left(\mathbf{b}^*_{I,J,t}\right)$ of the block vector $\mathbf{b}^*_{I,J,t}$ with respect to the distribution components k=1,…,K of the mixture estimated for all blocks that appeared at the same position I, J at previous time instants 1,…,t-1. If the minimal squared distance is above a pre-specified threshold Th, the block is considered as outlier and labeled as 'moving'. Subsequently, the k_r-th distribution component at that moment having the smallest estimated prior probability w_{k_r} is replaced by a new Gaussian distribution centered around the block vector $\mathbf{b}^*_{I,J,t}$ and having a large initial standard deviation (set to σ_0) but a relatively small prior (set to w_0). We call this mechanism *reset*, since we reset the parameters of one of the distributions.

In [17] the threshold Th was specified ad-hoc. In contrast, to determine the threshold, we propose the probabilistic approach, based on controlling the probability of the Type I error [3]. Specifically, when the Mahalanobis distance is employed, in [13] this probability (of falsely rejecting the hypothesis that a new block vector actually belongs to the estimated distribution) is shown equal to

$$1-\gamma\left(N'/2, Th/2\right)/\Gamma\left(N'/2\right),$$

where γ and Γ are the incomplete and complete Gamma functions, and N' is the dimensionality of a transformed block vector.

If the minimal squared Mahalanobis distance to one of distribution components is below the threshold *Th*, the block is not considered as outlier using the reset mechanism. However it still may belong to a moving object. Therefore, we employ the second criterion to detect moving blocks, which we refer to as a *hold* mechanism. First, we check whether an outlier has been detected (using the reset criterion defined above) within *H* frames preceding the current frame at the considered block position. If there were no outliers within the *H* previous frames, the block at the current frame is labeled as background. This criterion is a significant modification of the original algorithm [17] aimed to further prevent false labeling of blocks as moving (in addition, parameter *H* is related to minimal speed of an object that can be detected as moving). If there is at least one outlier within *H* previous frames at a given block position, we identify whether the distribution component closest to the current frame value is labeled as 'background' or 'moving'. This label carries over to the block vector $\mathbf{b}^*_{I,J,t}$ (as it is the case in [17])—for example, $\mathbf{b}^*_{I,J,t}$ is labeled as 'moving' if its closest distribution component has been labeled as 'moving' too. To assign each distributional component to foreground ('moving') or background, we proceed similar as [17]. The inductive bias of the assignment procedure is that the distributional components having large priors and small variances correspond to the background. First, we sort distribution components in decreasing order of w_k/σ_k quotients, where σ_k is the standard deviation of the mixture component *k*. Then, the smallest possible set of *L* components (having the highest quotients) is determined such that the sum of corresponding priors is at least T_L. These distribution components from the set are labeled as 'background'. The remaining *K*–*L* components are subsequently assigned to the foreground.

Final steps of the algorithm that include parameters update and priors renormalization are analog to those in [17].

3.3. Tracking

The fact that we base video processing on spatiotemporal blocks leads also to an improved performance of tracking algorithms (see [2,14] for good overviews of tracking algorithms). Recall that the goal of tracking is to establish the correspondence (so called motion correspondence) between detected moving objects across video frames. In order to uniquely identify a given moving object in a video, usually the proximity of its images across frames, or direction and speed of its motion are used. Once the object is identified, motion trajectory can be computed. A comprehensive overview of the problems related to tracking can be found in [10].

A simple rule-based tracking algorithm that establishes the correspondence among objects and frame *t*-1 and frame *t* using topological relations of objects already yields good results for PETS sequences as demonstrated in Section 4.3. We consider objects as connected components of moving blocks. The topological relations of neighborhood and continuity are defined using corresponding concepts of digital topology, 8-neighborhood and digital continuous functions (Rosenfeld [15,16]). They are applied to binary images composed of blocks in which ones denote moving blocks. Observe that although the same definitions make sense for binary images composed of original pixels, they would lead to very complicated rules for tracking moving objects when applied on the pixel level. The object displacements measured in pixels are significantly higher than measured in blocks and the number of connected components is significantly larger.

4. Results

We have demonstrated the performance of the proposed approach on sequences from the Performance Evaluation of Tracking and Surveillance (PETS) repository[1]. Processed video-sequences that illustrate the performance of our algorithm are available on our web site: http://divac.ist.temple.edu/~pokie/longin/.

Here, we present results on a video sequence from PETS2001[2] (here referred to as the *Outdoor video* sequence) and on an indoor sequence from PETS2002[3] (referred to as the *Indoor video* sequence).

Since the original sequences contained RGB colors, prior to applying our technique, we converted RGB to grayscale (PAL luminance). In addition, we reduced the size of the videos twice such that the frame size for the *Outdoor video* sequence is *X*=288, *Y*=384 (in contrast to the 576×784 pixel frames of the original video) and for

[1]Available at ftp://pets.rdg.ac.uk/.
[2]ftp://pets.rdg.ac.uk/PETS2001/DATASET1/TESTING/CAMERA1_JPEGS/
[3]ftp://pets.rdg.ac.uk/PETS2002/PEOPLE/TESTING/DATASET2/

the *Indoor video* sequence X=120, Y=320 (as compared to the original 240×640).

In our experiments we use T=1 and N_{BLOCK} = 8, thus the length of a block vector $\mathbf{b}_{I,J,t}$ is N = 192 = 8×8×3. To compute the projection matrix **P**, we take the block vectors from each 50th frame of the movie (the blocks from these frames are assumed to adequately represent the texture from the whole movie). We use the transformed block vectors $\mathbf{b}^*_{I,J,t}$ with N' = 3 components such that the performed PCA projection preserves more than 99.5% of the block vectors variance. Prior to processing, the components of transformed block vectors are rescaled into [0,255] range.

Using the proposed algorithm, we detect moving blocks by estimating the mixture of K=5 Gaussian components. The remaining parameter values of the algorithm are:

Th=2.5^2=6.25; T_L=1-1/K-0.01=0.79; α=0.075;

w_0 = 0.02; H=25; σ_0=12 ; $\boldsymbol{\mu}_0$=[0 0 0].

4.1 Moving objects identification

The result of the proposed approach on the *Outdoor video* sequence is illustrated in Fig. 1, where we see four sample frames with the blocks labeled using our algorithm. In Fig. 1, the background blocks are shown red, while moving blocks detected by our technique are colored green and blue, depending whether they are detected by reset or hold mechanisms, respectively. While analyzing these frames we focus our attention on the block position (24, 28) framed yellow in Fig. 1. The frame 499 captures the moment when a person walks through the center of image while at the same time a car appears on the right lower corner of scene. On the block position (24,28), we see a green block that identifies the person as moving. At the frame 624, the car goes towards the center of the scene while the pedestrian leaves the scene. In the yellow box, we see a blue block that identifies the car moving. At the frame 863, a white van drives (left to right) through the position of the yellow box while on frame 1477 two people walk from right to the left. As we can see, our method was able to detect moving blocks with good accuracy.

To illustrate how reset and hold mechanisms contribute and intervene in moving block detection process, in Fig. 2 we show frames detected by these two methods at the block position (24, 28) (the position is marked with the yellow-framed boxes in Fig. 1). As we can see, the 'reset' mechanism typically triggers the sequence of moving blocks being identified by 'hold' mechanism. From this figure, we can also see that there are four major groups of non-stationarities, and they actually correspond to the four moving objects that appeared in the 'yellow' frame in this video, as illustrated in the frames shown in Fig. 1. Resets are relatively infrequent for slow-moving objects and the major mechanism to detect blocks corresponding to the moving objects is hold (e.g. frames 1477–1500).

Therefore, by adjusting the "persistence" parameter H, we could regulate the sensitivity of the system to slow objects and limit the minimal speed of such identified objects. In contrast, for fast objects, the predominant detection mechanism is reset (e.g. frames 826–866) and the maximal speed of identified objects is not explicitly limited. The result of the proposed approach on the *Indoor video* sequence is illustrated in Fig. 3, where we see five characteristic frames with the blocks labeled using our algorithm (the blocks coloring scheme is the same as in Fig. 1). While analyzing these frames we focus our attention on the block position (7, 25) framed yellow in Fig. 3. The frame 145 captures the moment when a person walks left to right, while the frame 265 corresponds to the person walking right to left through the observed block position. At the frame 745, two persons are walking right to left, while at the frame 997 a group of persons moves in the same direction. At the frame 1675, a person is moving in background while another person is moving very slowly in front of the window. As we can see, in case of an indoor video, the ability of our method to detect moving blocks is also very good.

4.2 Comparison to pixel-based approaches

Recall that we performed the detection of moving objects using the first three PCA components of each 3D block vector. Hence, we can observe each particular block in time through visualizing its trajectory in the feature space of the PCA components. In Fig. 4a, we show the trajectory for the block (24, 28) of the *Outdoor video* as well as frames identified as moving using both mechanisms (reset and hold). From Fig. 4a we can see that the distribution corresponding to the blocks is multimodal globally.

We can observe at least two 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. We can see that our technique is able to identify the "distributional outliers" that correspond to the moving objects (marked in figure with green and blue dots depending whether they are identified by reset or hold mechanisms). This is clearly visible for frames 826–866. After these frames, we enter into the second mode, and after a brief"excursion", we return to this mode. However, although we can observe several distributional components while looking at all frames, when observing only a window of frames (e.g. 200 frames) the data typically belong to only one component. Consequently, in frames when the object corresponding to the observed block location is stationary, the estimated prior corresponding to one

specific distributional component is typically much larger than the sum of other priors.

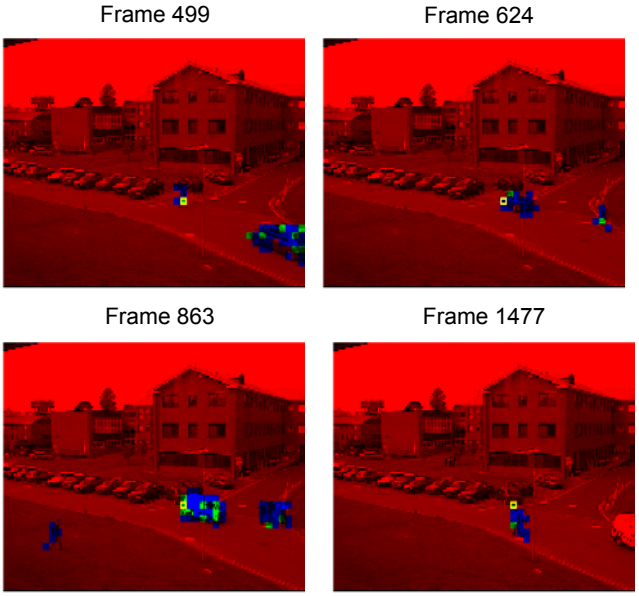

Figure 1. Moving blocks (green identified by reset, blue identified by hold mechanism) as detected in four characteristic frames of the *Outdoor video* sequence using the proposed technique with 5-component EM and 8×8×3 blocks projected onto 3 principal components. The block *(24,28)* is marked by a yellow-bordered box.

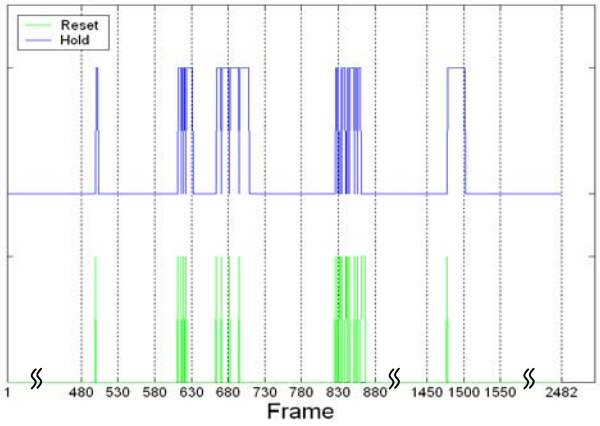

Figure 2. Frames identified as moving at block *I*=24, *J*=28 of the *Outdoor video* sequence using reset and hold mechanisms.

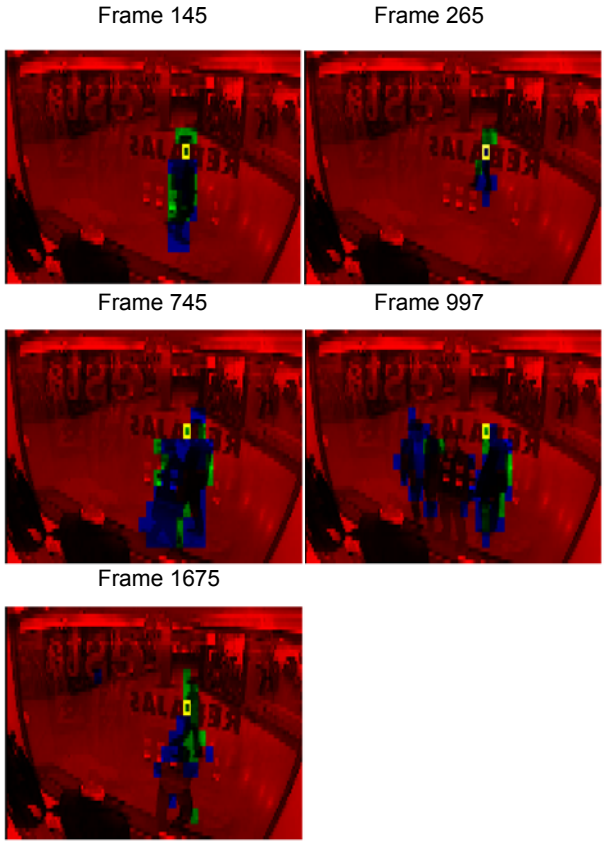

Figure 3. Moving blocks (green identified by reset, blue identified by hold mechanism) as detected in four characteristic frames of the Indoor video sequence using the proposed technique with 5-component EM and 8×8×3 blocks projected onto three principal components. The block *I*=7, *J*=25 is specially denoted by a yellow-bordered box.

In comparison to any pixel-based approach (e.g., Stauffer and Grimson [17]), our technique 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). To demonstrate this, in Fig. 4b we plot trajectory over time of RGB color values that occur at the pixel (185, 217), which is one of the pixels in the block (24, 28). For better visualization, in Fig. 4b 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). In Fig. 4b, we superimpose green and blue dots computed by our algorithm for block (24,28), that correctly correspond to moving objects at this position.

By comparison of Figs. 4a and 4b one can conclude that in both cases there are two distributional components corresponding to the background. However, using the proposed technique, the background variance is much

smaller (since using block vectors that contain texture information effectively results in effective noise reduction in comparison to using "raw" pixels). Hence, a distribution-based technique to detect moving objects as outliers will perform much better using spatiotemporal blocks than when using raw pixels (either original or linearly transformed).

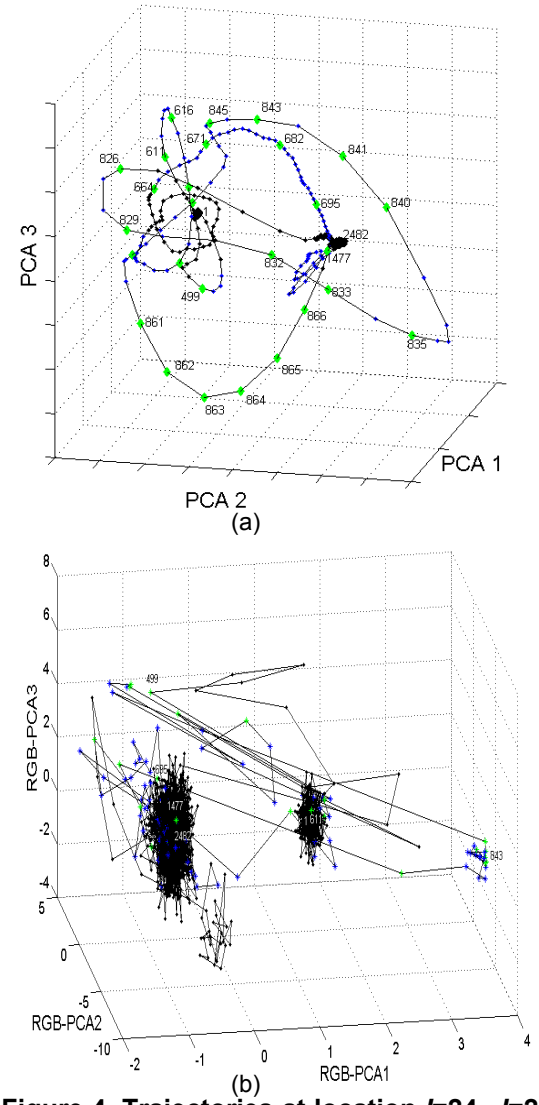

Figure 4. Trajectories at location *I*=24, *J*=28 of the *Outdoor video* in feature space of a) first three (standardized) PCA components of block vectors; b) standardized PCA components of RGB pixel coordinates at pixel location *(185, 217)* (inside block *I*=24, *J*=28). Black, blue and green dots corresponding to the frames where the block (24, 28) was identified as background, and moving (using 'reset' and 'hold' mechanisms) by the proposed algorithm.

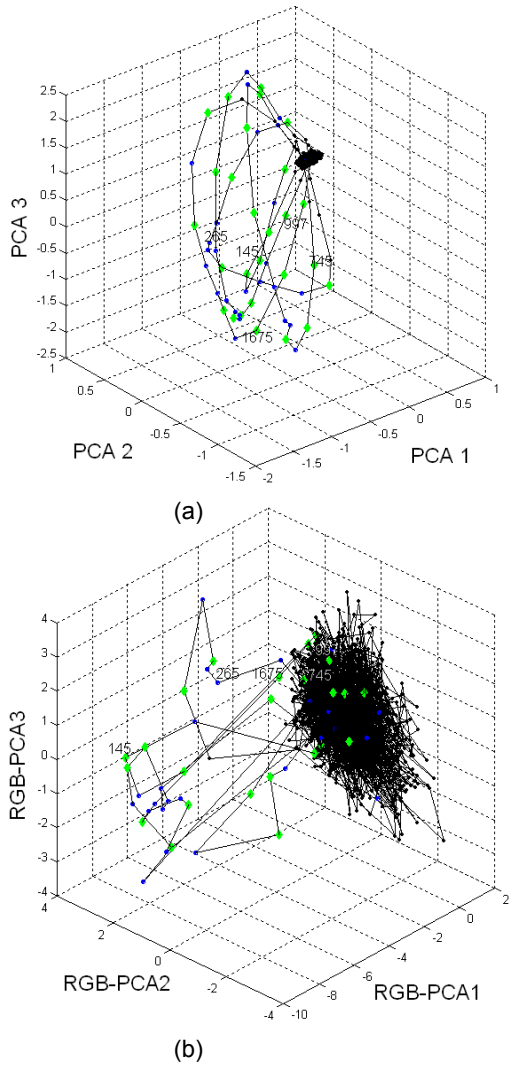

Figure 5. Trajectories at location *I*=7, *J*=25 of the *Indoor video* in feature space of a) first three (standardized) PCA components of block vectors; b) standardized PCA components of RGB pixel coordinates at pixel location *(49, 193)* (inside block *I*=7, *J*=25). Black, blue and green dots corresponding to the frames where the block (7, 25) was identified as background, and moving (using 'reset' and 'hold' mechanisms) by the proposed algorithm.

Recall that the original method [17] was proposed for RGB coordinates without the principal component projection. However, as illustrated in Fig. 4b, the principal components themselves also cannot significantly decrease the noise and thus make outliers detection easier.

Now consider whether the algorithm [17] applied to pixels can identify different moving objects that appear at the location of the observed pixel. At the time interval 826–866 (corresponding to the white van passing through the observed pixel location) the pixel values in RGB coordinates are close to the maximum (255), and form a separate cluster, as can be seen in Fig 4b. Hence, the algorithm [17] (since it is based on color of pixels) will not have difficulties to identify this time interval. However, when the color of the moving object is close to the color of background, that algorithm can be inappropriate for detection of moving objects. As it can be seen in Fig. 4b, the method from [17] (either used on the original or transformed RGB coordinates) will have difficulties in properly detecting frames 611, 695, 1477 belonging to the second and fourth moving objects that appear at the observed pixel. The reason is that many green and blue dots incorrectly become parts of the two background components in Fig. 4b, which means that a pixel-based method would classify the corresponding colors as the background.

Analog results are obtained for the *Indoor video* sequence. As it can be seen from Fig. 5a, the 3D block representation can provide clear separation between background blocks and moving blocks (that comprise distinguishably separated "orbits"). Interestingly, in this case (block (7,25)) each orbit corresponds to a separate moving object, which potentially opens possibility to use an orbit identification algorithm for detection and classification of moving objects. On the other hand, as demonstrated in Fig. 5b, the algorithm [17] directly applied to pixels will have difficulties in properly identifying movement in some frames that are easily identified by the proposed technique (e.g. frames 745, 997). In such frames, the patterns representing the frames are close to the distributional components representing background in the pixel space, and hence cannot be easily identified as distributional outliers. In contrast, the proposed technique uses 3D blocks that, in addition to spatial dimensions, also contain the information about the pixel values in time. Since we perform principal component analysis of the 3D block vectors, we are capable of extracting the information about temporal change in location values. Using this information we are able to significantly reduce problems in identifying moving objects.

4.3. Tracking results

As we stated in Section 3.3, we used a simple rule-based tracking algorithm, since our main goal is to demonstrate that the proposed technique provides a significantly improved input to tracking algorithms in general. As the result we obtain robust trajectories for isolated moving objects. For example, the trajectories of three objects in the *Outdoor video* obtained by our simple tracking algorithm are shown in Fig. 6. However, our tracking fails in the presence of occlusions. Clearly, more sophisticated tracking algorithms, e.g., as proposed in Javed and Shah [10], are needed to solve this problem.

Figure 6. Trajectories of thee objects in the *Outdoor video* sequence obtained by our simple tracking algorithm.

5. Conclusions and work in progress

In this paper we demonstrated that using spatiotemporal blocks and linear variance-preserving dimensionality reduction can result with better detection and tracking of moving objects in comparison to standard pixel-based techniques.

The proposed technique implicitly assumes the feasibility of computing projection matrix from blocks that adequately represent the texture from the movies to be processed by our system. While this approach can provide good results when applied on videos with comparatively high stationarity in background (e.g., indoor surveillance videos with artificial illumination), further improvements are possible if the projection matrix is computed dynamically. However, the techniques for adaptive estimation of projection coefficients in time are out of scope of this study.

Our future contribution to tracking will be based on object recognition. While tracking a given object, we simultaneously learn the distribution of its blocks, which forms a model of this object. Subsequently, we can improve the tracking performance, since we perform unsupervised object recognition (by comparing the distributions) in addition to tracking. This significantly improves the tracking performance in the presence of occlusion (with other moving objects as well as with stationary objects) and shadows. Again, we profit here from the fact that our underlying representation is based

on texture of 3D blocks as compared to the existing approaches that are based on color or gray level values of pixels. Experimental results to justify this claim will be presented in a forthcoming paper.

6. Acknowledgements

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).

7. References

[1] Buttler, D., Sridharan, S., and Bove, V. M. Real-time adaptive background segmentation. In Proc. 4th IEEE Int. Conf. on Multimedia and Expo ICME 2003 (Baltimore, MD, July 2003).

[2] R.T. Collins,A.J. Lipton, and T. Kanade, "Introduction to the Special Section on Video Surveillance", *IEEE Trans. Pattern Analysis and Machine Intelligence (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 Trans. Pattern Analysis and Machine Intelligence (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. International Joint Conference on Artificial Intelligence IJCAI 77 (Cambridge, MA, 1977), 612–618.

[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. Proc. IEEE Workshop on Motion and Video Computing MOTION '02 (Orlando, FL, Dec 5-6 2002), 22-27.

[10] Javed, O., and Shah, M. Tracking and object classification for automated surveillance. In Proc. 7th European Conf. on Computer Vision ECCV 2002 (Copenhagen, Denmark, May 2002), Springer, Vol. 4, 343-357.

[11] Lee, I., and Guan, L. Centralized peer-to-peer streaming with layered video. In Proc. IEEE Int. Conf. on Multimedia and Expo, (Baltimore, MD, July 2003), in press.

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

[13] Pokrajac, D., Milutinovich, J., and Jankovic, D., 2003, On hypothesis testing in multidimensional outlier detection. In Proc. 6th Int. Conf. on Telecommunications in Modern Satellite, Cable and Broadcasting Services—TELSIKS (Nis, Serbia, October 2003), in press.

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

[15] Rosenfeld, A. Digital Topology. *Amer. Math. Monthly*, 621-630, 1979.

[16] Rosenfeld, A. (1986): ´Continuous´ functions on digital pictures. *Pattern Recognition Letters* 4, 177-184, 1986.

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

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