Activity and Motion Detection Based on Measuring Texture Change

Longin Jan Latecki[1], Roland Miezianko[1], and Dragoljub Pokrajac[2]

[1] Temple University, CIS Dept., Philadelphia, PA 19122, USA
{latecki, rmiezian}@temple.edu
[2] Delaware State University, CIS Dept, Dover, DE 19901, USA
pokie@ist.temple.edu

Abstract. We estimate the speed of texture change by measuring the spread of texture vectors in their feature space. This method allows us to robustly detect even very slow moving objects. By learning a normal amount of texture change over time, we are also able to detect increased activities in videos. We illustrate the performance of the proposed techniques on videos from PETS repository and the Temple University Police department.

1 Introduction

Motion detection algorithms are important research area of computer vision and comprise building blocks of various high-level techniques in video analysis that include tracking and classification of trajectories. It is an obvious and biologically motivated observation that the main clue for detection of moving objects is the changing texture in parts of the view field. All optical flow computation algorithms use derivative computation to estimate the speed of texture change. However, derivative computation may be very unstable in finite domains of images. Therefore, in this paper we introduce a method that does not require any derivative computation. We propose an approach to motion and activity detection based on statistical properties of texture vectors.

Let us focus on a fixed position in a video plane and observe the sequence of texture vectors representing a patch around this position over time. Each texture vector describes the texture of the patch in a single video frame. We assume a stationary camera. If we observe the patch that corresponds to part of the background image, the texture vectors will not be constant due to various factors (e.g., illumination changes, errors of the video capture device), but combined effect is merely a small spread of texture vectors over time. Also a repetitive background motion like tree branches waving in the wind yields a relatively small spread of texture vectors. Since similar texture repeats frequently, the texture vectors in this case are highly correlated.

On the other hand, if a moving object is passing through the observed location, it is very likely that object will have a different texture from the background patch. Therefore, the texture vectors are very likely to have a large spread. Even if different parts of the moving object have the same texture that is the same as the background texture,

P. Perner and A. Imiya (Eds.): MLDM 2005, LNAI 3587, pp. 476 – 486, 2005.

the texture vectors will have large spread at the observed location, since different texture parts will appear in the patch. This holds under the assumption that the texture is not completely uniform, since then different texture parts have different texture vectors. To summarize, the proposed approach can identify moving objects even if their texture is identical with the background texture, due to the fact that our classification is based on measuring the amount of texture change and texture structure is extremely unlikely to be perfectly uniform.

Observe that we measure the spread of texture vectors in the texture space. Because of this, we are not able to compute the optical flow directly, i.e., to estimate the directions and speed of moving objects. However, we are able to perform robust detection of moving objects. In comparison to the existing motion detection algorithms [6,7,14], we do not compute any model of the background. We measure the amount of texture change and classify it into two categories: moving and stationary objects. The aforementioned situation in which the background texture and the texture of moving object are similar illustrates a typical situation in which the proposed approach outperforms any background modeling method. In such cases, in the background modeling approaches the texture of a moving object can be easily misclassified as background texture. A detailed explanation follows in Section 3.

Instead of color, gray level, or infrared values at pixel locations, we consider the values of all pixels in spatiotemporal regions represented as 3D blocks. These 3D blocks are represented through compact spatiotemporal texture vectors to reduce the influence of noise and decrease computational demands. In [11] we have shown that the use of such texture vectors 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. 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 stress that the proposed motion detection technique is independent of any particular texture representation used.

To represent texture, we consider the values of all pixels within spatiotemporal regions represented as 3D blocks. A 3D block (e.g., 8x8x3 block) consists of a few successive frames (e.g., 3) at the same quadratic patch (8x8) of a scene. To compactly represent these values and to reduce the influence of noise, we apply a dimensionality reduction technique by using principal components projection (PCA). As the result, texture is represented by a vector containing only the most significant projected components of texture, while less significant components and noise are filtered out through the process of feature extraction. The most significant projected components represent a small subset of all the projections. The obtained texture vectors provide a compact low-dimensional joint representation of texture and motion patterns in videos and are used as primary inputs to a motion detection technique. As we mentioned above, texture at a given location in video plane is very likely to considerably vary while a moving object is passing through this location. To measure this variance, we estimate covariance matrix of the texture vectors from the same location within a window of a small number of successive frames, and determine the texture spread as the largest eigenvalue of the covariance matrix. This way, we indirectly determine the magnitude of texture variability in the direction of its maximal change. Finally, the decision whether a moving object or a stationary background is identified at a given spatiotemporal location is made by dynamic distribution learning of the obtained largest eigenvalue.

The proposed technique can use a variety of video sequences as input, ranging from monochromatic gray scale or infra-red (IR) videos to multispectral videos in visible or IR spectral domain. In this paper, we demonstrate the usefulness of the proposed method on several benchmark videos from PETS workshop. The robust performance of the proposed motion detection method, allows us to base our increased activity detection on it. We define motion amount as a sum of motion activates of all blocks in a given frame (Section 4). By applying a simple statistical learning of the motion amount we are able to detect increased activities. We learn the distribution of the total motion amount in all previous frames, under the assumption that mostly normal activities are present. An increased activity is detected as outlier of the learned distribution.

Our approach to increased activity detection does not include any specific domain knowledge about the monitored objects. Such knowledge can be incorporated in our framework, e.g., we can focus on monitoring only human or vehicle activities. By adding a classifier that is able to label moving object categories, we can restrict our attention to particular object categories, e.g., see [18].

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.

The usefulness 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 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.

2 Proposed Methodology

2.1 Video Representation with Spatiotemporal (*sp*) Texture Vectors

We represent videos as three-dimensional (3D) arrays of gray level or monochromatic infrared pixel values $\mathbf{g}_{i,j,t}$ at a time instant t and a pixel location i,j. 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 two squares in common. The fact that the 3D blocks overlap in time allows us to perform successful motion detection in videos with very low frame rate, e.g., in our experimental results, videos with 2 fps (frames per second) are included.

The blocks are represented by N-dimensional vectors $\mathbf{b}_{I,J,t}$, (e.g., $N=8x8x3$) specified by spatial indexes (I,J) and time instant t. Vectors $\mathbf{b}_{I,J,t}$ contain all values $\mathbf{g}_{i,j,t}$ of pixels in the corresponding 3D block. 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 a significantly lower length $K<<N$ using a PCA projection matrix $\mathbf{P}^K_{I,J}$ computed for all $\mathbf{b}_{I,J,t}$ at video plane location (I,J). The resulting *sp* texture vectors $\mathbf{b}^*_{I,J,t} = \mathbf{P}^K_{I,J} \cdot \mathbf{b}_{I,J,t}$ provide a joint representation of texture and motion patterns in videos and are used as input of algorithms for detection of moving objects. We used K=10 in our experiments. To compute $\mathbf{P}^K_{I,J}$ 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 $\mathbf{S}_{I,J}$. The PCA projection matrix $\mathbf{P}_{I,J}$ for spatial location (I,J) is computed from the $\mathbf{S}_{I,J}$ covariance matrix. The projection matrix $\mathbf{P}_{I,J}$ of size $N \times N$ represents N principal components. By taking only the principal components that corresponds to the K largest eigenvalues, we obtain $\mathbf{P}^K_{I,J}$.

2.2 Detection of Moving Objects by Measuring Texture Spread

The spread of texture vectors over time indicates whether the corresponding object texture is stationary or moving. Recall that each *sp* vector represents texture of the corresponding block. Hence, by observing the characteristics of *sp* vectors change over time, we are able to detect whether a particular block belongs to a moving object or to a background. Consider a single block position in a video plane. We can observe the trajectory of its *sp* vectors, i.e., the loci of *sp* vectors in successive time frames. If during an observed time interval there is no moving object in the block, the *sp* vectors will be close to each other. Hence the variance of *sp* vectors during the time interval will be small. In contrast, if there is a moving object passing through this block, the *sp* texture vectors will change fast, i.e., the *sp* vectors will be spread in the space of their coordinates. Therefore, the variance of *sp* vectors within an observation time window will be fairly large. In Fig. 2(a), we show the trajectory of *sp* vectors corresponding to block location (*24,28*) in Campus 1 video. To make this visualization possible, we use only first three PCA components for each *sp* vector. It can be observed that frames when only stationary objects are visible in the observed block location correspond to regions where *sp* vectors are clustered into fairly spherical shapes (black dots) with small spread. In contrary, when moving objects are passing through this block location, the trajectory of *sp* vectors (blue-gray dots) is typically elongated and the variance is relatively large.

A simple way to determine the speed of *sp* vector change would be to compute the norms of their first derivatives. However, computing finite differences of consecutive *sp* vectors may be unreliable. In order to determine whether the consecutive vectors belong to elongated trajectories, we need to observe whether they are making a consistent progress in one particular direction within a certain time interval. We propose to assess the *sp* vector spread in the direction of maximal variance. To measure the variance of *sp* vectors, we compute the covariance matrix of *sp* vectors corresponding to the same block location for a pre-specified number of consecutive frames. We use the maximal eigenvalue as the measure of trajectory elongation.

More formally, for each location (*x,y*), and temporal instant *t*, we consider vectors

$$b^*_{x,y,t-W}, b^*_{x,y,t-W+1}, \ldots, b^*_{x,y,t}, \ldots, b^*_{x,y,t+W} \quad . \tag{1}$$

corresponding to a symmetric window of size *2W+1* around the instant *t*. For these vectors, we compute the covariance matrix $\mathbf{C}_{x,y,t}$. We assign the largest eigenvalue of $\mathbf{C}_{x,y,t}$, denoted as $\Lambda_{x,y,t}$, to a given spatiotemporal video position to define a local variance measure, which we will also refer to as *motion measure*

$$mm(x,y,t) = \Lambda_{x,y,t} \tag{2}$$

The larger the motion measure *mm*(*x,y,t*), the more likely is the presence of a moving object at position (*x,y,t*). An example graph of *mm* is shown in Fig. 1.

The large values (spikes) correspond to time intervals when moving objects where observed at this particular video location. As this graph suggest, we can label video position (*x,y,t*) based on the history of *mm*(*x,y,t*) values over time (frames 1, …, *t*-1) as moving by applying an outlier detection method to *mm* values, i.e., a position is labeled as moving if motion measure value at a given time is classified as outlier.

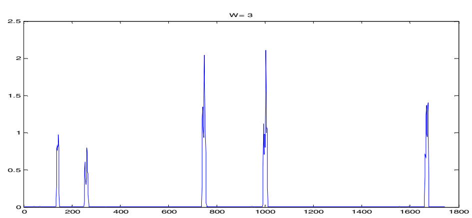


Fig. 1. The graph of local variance *mm* over time for the block (*24,28*) of the *Campus 1* video

2.3 Dynamic Distribution Learning and Outlier Detection

In the proposed approach for activity detection we apply outlier detection algorithms two times: for labeling of moving blocks and to detect increased activities. Now we describe outlier detection in more detail. Consider labeling each video position as moving or stationary based on whether the motion measure *mm* is larger or smaller than a suitably defined threshold. We use a dynamic distribution learning to determine the threshold value at position (x,y,t) based on the history of $mm(x,y,t)$ values over time (at frames 1, …, t-1). Since $mm(x,y,t)$ is a function of one variable t for a fixed position (x,y) (see Fig. 1), the task reduces to dynamic estimation of the mean and standard deviation of *mm*. Given a function f of one variable, we compute initial values of mean $mean(t_0)$ and variance $\sigma^2(t_0)$ of all values $f(t)$ in some initial interval t=1..t_0. An outlier is detected at time $t>t_0$ if the standardized feature value is sufficiently large, i.e., when

$$\frac{f(t) - mean(t-1)}{std(t-1)} > C_1 \text{ , where } C_1 \text{ is a constant} \tag{3}$$

Once an outlier is detected at time t_1, all values $f(t)$ are labeled as outliers for $t_1<t$ until we switch to a nominal state. We switch to the nominal state at time t, $t_1<t$, if the standardized feature value drops below a threshold $C_2 < C_1$, i.e.,

$$\frac{f(t) - mean(t-1)}{std(t-1)} < C_2 \tag{4}$$

We update the estimates of mean and standard deviation only when the outliers are not detected (nominal state), i.e., at the beginning of the execution of the algorithm and when (4) holds, *mean* and *std* are updated using running average (an algorithm for incremental estimation of parameters of distributions, that is commonly applied in the case of Gaussian distribution):

$$mean(t) = u \cdot mean(t-1) + (1-u) \cdot f(t) \text{ and } std(t) = \sqrt{\sigma^2(t)} \tag{5}$$

$$\sigma^2(t) = u \cdot \sigma^2(t-1) + (1-u) \cdot (f(t) - mean(t-1))^2 \tag{6}$$

For example, we use C_1=9, C_2=3, and u=0.99 in the case of the detection of moving blocks for f=mm. The only assumption that we make about the distribution of values of function f is that it has a significant right tail. This assumption clearly applies to the Gaussian distribution, but is significantly more general.

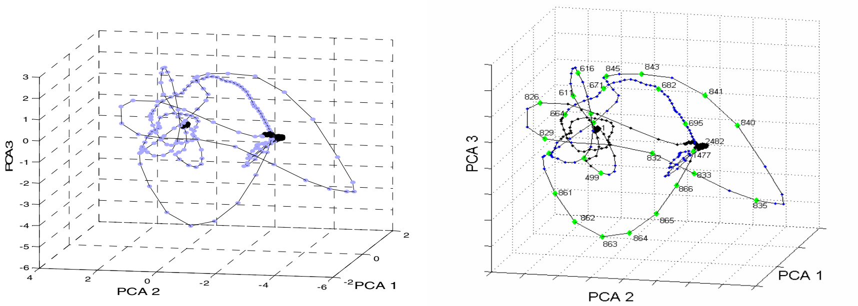
3 Motion Orbits in Texture Space

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 2. 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 spread measure of texture vectors to the Gaussian mixture model introduced in [14]. To make the comparison more realistic, we apply the Gaussian mixture model to texture vectors. Hence, both compared techniques are based on the same spatiotemporal blocks that represent texture and motion patterns. We also show that the Gaussian mixture model on texture vectors significantly outperforms the original representation used in [14] (RGB color values on a pixel level).

We define a motion orbit as path that the texture representation at the fixed video plane location traverses over time. Recall that we use texture vectors composed of the first 3 PCA components of each spatiotemporal block vector. Hence, 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. 2(a), we see the orbit for the block (*24,28*) of the *Campus 1* PETS video [19]. 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. 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*): a part of parking lot and a parked car. Around these blobs we see 1D orbits marked with blue-gray dots corresponding to moving objects. We can view the proposed local variance method as orbit classification algorithm. The reason is that elongated 1D orbits that identify motion have higher spread than the stationary background objects.

We stress that the dot labeling as shown was computed by the proposed method for detection moving objects. 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 *Campus 1* video the incremental EM method [14] failed to identify the motion orbit containing frames 633—663.In comparison to any pixel-based approaches (e.g., as originally proposed in [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 spatiotem-



(a) (b)

Fig. 2. (a) Orbits of block (*24,28*) vectors with blue-gray dots corresponding to the frames in *Campus 1* where the block was identified as moving by the proposed method; (b) Orbits of block (*24,28*) vectors marked with dots: black as background, blue and green as moving—using 'reset' and 'hold' mechanisms, correspondingly, identified by the EM algorithm

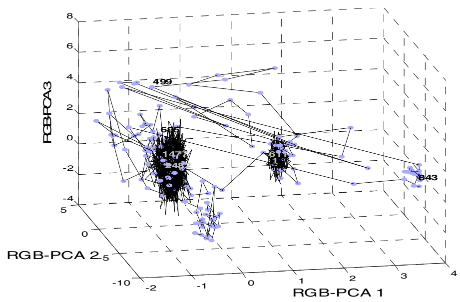


Fig. 3. Standardized PCA components of RGB pixel values for *Campus 1* at pixel location (*185,217*) that is inside block (*24,28*); allowS a direct comparison to Fig. 2(a)

poral texture representation of 3D blocks instead of pixels). We demonstrate how noisy RGB color values of a single pixel can be in Fig. 3, 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 *Campus 1* video. For better visualization, in Fig. 3 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. 2(a) (computed by our local variance technique), we carried over the dot labels from Fig. 2(a).

By comparison of Fig. 3 to Fig. 2(a), 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. As it can be seen in Fig. 3, 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. The blue-gray dots incorrectly become parts of two background components, which imply that a pixel-based method [14] would classify the corresponding blue-gray dots as belonging to a background distribution. The proposed local variation based technique can also be applied on pixel level. However, due to problems with large uniform texture regions as well as noise inherent to pixel values (shown above), our preferred technique is to apply local variance method on *sp* block texture vectors.

4 Detection of Increased Activities

Due to the fact that we robustly compute the motion measure *mm*, we can also reliably estimate the motion amount in each video frame. Motion amount can be defined as the sum of motion measures of all blocks:

$$ma(t) = \sum_{x,y} mm(x,y,t) \tag{7}$$

The proposed method of detecting increased activities is again based on outlier detection (see Section 2.3) but this time of the motion amount over time. Thus, we first learn the distribution of motion amount over time when the recorded video activity was considered usual/nominal. Then time intervals with increased activity are detected as outliers of the learned distribution. The proposed approach works under the assumption that there exists an upper bound on the size of moving objects whose motion we want to detect (measured in the number of moving blocks), and that the genuine moving objects do not appear rapidly in the frame. These assumptions hold for most surveillance videos. Let us consider an example video, called *Temple 1*, that satisfies the assumptions. Indeed, this video is recorded by a roof mounted, stationary camera, so that a certain minimal distance to moving objects is guaranteed. Typical moving objects there, humans and vehicles, cannot get arbitrarily large. Hence, the fraction of the scene occupied by a moving object is limited. Observe that the actual value of the upper bound on the size of moving objects needs not to be known, since our algorithm learns it automatically. Similarly, the number of humans and vehicles cannot rapidly increase, since the regions of entry into the camera view field are limited in size.

In Fig. 4(a), we see the graphs of function *ma* for *Temple 1* video and correctly detect alarm situations as shown in Fig. 4(b). For example, a significant increase in the

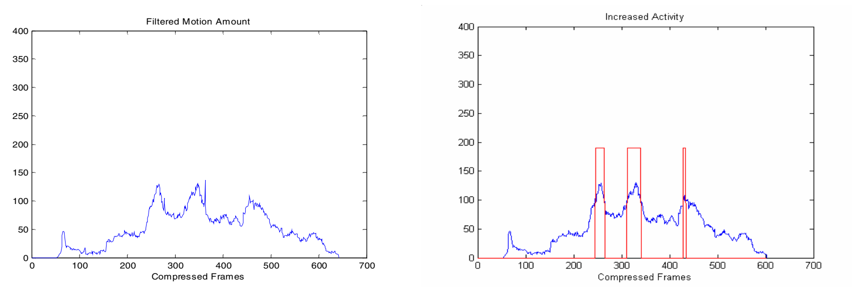


Fig. 4. Activity Detection. (a) Motion amount of *Temple 1* video; (b) Increased activity blocks marked with red boundaries

(a) (b)

Fig. 5. *Temple 1* video (a) showing no activity and (b) showing increased activity due to street fight (ACTIVITY label is shown next to the frame number)

number of motion blocks around frame 300 indicates an alarm situation. This is a correct prediction, since a street fight is recorded on the video around frame 300, see Fig. 5 and the *Temple 1* video [12].

5 Performance Evaluation on Test Videos

A set of several test videos showing our motion detection results and our results on detecting increased activity can be viewed on [12]. Our test set of videos includes several videos from the Performance Evaluation of Tracking and Surveillance (PETS) repository. In particular, the results include the above discussed *Campus 1* video from PETS2001, videos obtained from the Police Dept. of Temple Univ., Philadelphia, and infrared videos, for which the same settings of parameters as for visual light videos were used.

6 Conclusions

In this paper we propose a local variation based method for motion detection. Our preliminary results on surveillance 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.

We show that the proposed local variation algorithm 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 Gaussian model estimation algorithm. This makes the proposed technique potentially appealing for moving detection in higher dimensional domains, such as multispectral remote sensing imagery.

Our approach to increased activity detection does not include any specific domain knowledge about the monitored objects. Such knowledge can be incorporated in our framework, e.g., we can focus on monitoring only human or vehicle activities if we can restrict our attention to particular object categories.

Acknowledgements

D. Pokrajac has been partially supported by NIH-funded Delaware IDeA Network of Biomedical Research Excellence (INBRE) Grant, DoD HBCU/MI Infrastructure Support Program (45395-MA-ISP Department of Army), National Science Foundation (NSF) Infrastructure Grant (award # 0320991) and NSF grant “Seeds of Success: A Comprehensive Program for the Retention, Quality Training, and Advancement of STEM Student” (award #HRD-0310163).

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 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. Link to Temple ViVi Lab video 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.
17. S. Glisic, Z. Nikolic, D. Pokrajac, P. Leppanen, "Performance Enhancement of DS Spread Spectrum systems: Two Dimensional Interference Suppression," IEEE Trans. Communication, Vol. 47, No. 10, pp.1549-1560, 1999.
18. W. Niu, J. Long, D. Han, and Y.-F. Wang. Human Activity Detection and Recognition for Video Surveillance. In Proc. IEEE Int. Conf. on Multimedia and Expo (ICME), 2004.
19. Performance Evaluation of Tracking and Surveillance (PETS) repository videos Campus 1 and 3: ftp://pets.rdg.ac.uk/PETS2002//DATASET1/TESTING/***/