Proc. IEEE Int. Conf. on Advanced Video and Signal Based Surveillance, Como, Italy, September 2005

Tracking Motion Objects in Infrared Videos

Longin Jan Latecki[1], Roland Miezianko[1], Dragoljub Pokrajac[2]
[1]Temple University, [2]Delaware State University
[1]{latecki, rmiezian}@temple.edu, [2]dpokraja@desu.edu

Abstract

We propose motion detection and object tracking method that is particularly suitable for infrared videos. Detection of moving objects in infrared videos is based on changing texture in parts of the view field. We estimate the speed of texture change by measuring the spread of texture vectors in the texture space. This method allows us to robustly detect very fast and very slow moving object. Our theoretical and experimental results show that the proposed method significantly outperforms the Stauffer-Grimson approach based on Gaussian mixture model. We observe that the proposed method does not require any post-processing, which is a necessary step for the Stauffer-Grimson approach. Moreover, the object tracking is improved when based on the spatiotemporal texture blocks.

1. Introduction

Motion detection algorithms are the building blocks of various high-level techniques in video analysis that include tracking and classification of trajectories. The most popular motion detection method (Stauffer-Grimson [14]) models the background pixels as multimodal Gaussian distributions of RGB (or other) color values. The Stauffer-Grimson (S&G) algorithm performs adequately on color images, but it does not perform well on infrared (IR) videos. Since IR videos usually provide only one value per pixel a direct adaptation of S&G is forced from multidimensional to one-dimensional Gaussian distribution. However, a single IR value (similar to single grayscale value) does not provide adequate means for pixel classification. An obvious way to improve S&G performance is to provide as the input multi-dimensional texture vectors that characterize pixel neighborhoods. As we will show in this paper S&G algorithm still does not yield a satisfactory performance when applied to IR texture vectors. We propose a new motion detection method that is more suitable to IR videos. It is based on measuring the speed of change of texture vectors. The proposed method can identify moving objects even if their texture is identical to 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.

In comparison to the existing motion detection algorithms (e.g., [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. The proposed technique can use a variety of video sequences as input. In this paper, we demonstrate the usefulness of the proposed method on several monochromatic IR videos obtained from the Ohio State University Thermal Pedestrian Database [19].

2. Motion feature representation

2.1. Spatiotemporal texture vectors

We represent videos as three-dimensional (3D) arrays of monochromatic (infrared or gray level) pixel values $\mathbf{g}_{i,j,t}$ 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 pixels in horizontal and vertical direction of each frame. Each image is divided in a video sequence into disjoint $N_{BLOCK} \times N_{BLOCK}$ squares (e.g., 4x4 squares) that cover the whole image. Spatiotemporal (*sp*) 3D blocks are obtained by combining squares in consecutive frames at the same video plane location. In our experiments reported here, we use 4x4x3 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 obtained 3D blocks are represented as 48-dimensional (4*4*3) vectors of monochromatic infrared pixel values.

In general 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 IR 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 [8] 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 $\mathbf{b}^{*}_{I,J,t} = \mathbf{P}^{K}_{I,J} * \mathbf{b}_{I,J,t}$ provide a joint representation of texture and motion patterns in videos and are used as input of algorithms for detection of motion and objects tracking. We use *K*=10 in our experiments. The obtained 10-dimensional vectors form a compact spatiotemporal texture representation for each block. It is important to notice that a different projection matrix $\mathbf{P}^{K}_{I,J}$ is used for each video plain location. This assures that the obtained texture vectors are able to optimally distinguish different textures that appear in a given block. The initial projection matrix is trained on the first t_0 frames under the assumption that only background is present in all block locations. The projection matrices are then updated during the time periods in which no motion is detected in a given block location.

2.2. Detection of moving features 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, which we call motion orbits. For example, see Fig. 1, where each point represents the first three PCA components of the texture vectors.

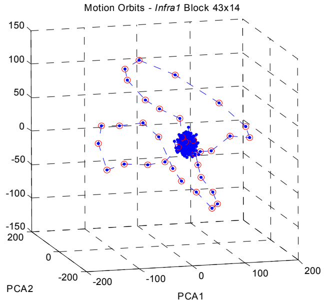


Fig 1. Motion orbits for block location 43x14 of *Infra1* video

If during an observed time interval there is no moving object in the block, i.e., a stationary background is only present in the block, the *sp* vectors will be close to each other. The background texture is represented by the large cluster of points as seen in Fig. 1. 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.

To summarize, it can be observed that frames with only stationary objects are visible in the observed block location correspond to regions where *sp* vectors are clustered into fairly spherical shapes with small spread. In contrary, when moving objects are passing through this block location, the trajectory of *sp* vectors 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.

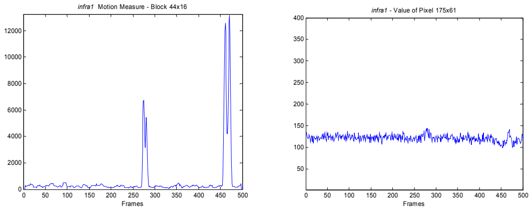
We propose to assess the *sp* vector spread in the direction of maximal variance. To measure the variance of *sp* vectors, we compute the covariance matrix of *sp* vectors corresponding to the same block location for a pre-specified number of consecutive frames. We use the maximal eigenvalue as the measure of trajectory elongation. More formally, for each location (*x,y*), and temporal instant *t*, we consider vectors of the form

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

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 sp video position to define a local variance measure, which we will also refer to as *motion measure*

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

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. 2(a), which measures the spread of the motion orbits depicted in Fig. 1. The large values (spikes) correspond to time intervals when moving objects where observed at this video location. The large values exactly correspond to two elongated motion orbits in Fig. 1, while the small values correspond to the texture vectors within the background cluster.



(a) (b)

Fig. 2. (a) Motion measure value for block 44x16 showing motion around frames 275 and 475; (b) IR values of pixel 175x61 inside block 44x16 not showing any significant change.

For comparison, we show IR values of a pixel within block location 43x16 in Fig. 2(b). Due to a significant amount of noise, detection of moving objects seems to be a very challenging if not impossible task when base on pixel IR values. We can see a distinct advantage to *sp* block processing here, where motion is detected in block 44x16, yet the pixels inside that block show no relevant texture changes.

As the graph in Fig. 2(a) suggests, 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. To perform the outlier detection, we first learn the nominal distribution of *mm*(*x,y,t*) values over some initial time period (*t*=1, …, t_1). This requires that the amount of unusual activity is relatively small in the initial time period, i.e., the part of the scene we mostly view at this location in the initial time period is stationary (background) .Then we use running average to update the mean and standard deviation of this distribution. The update is not performed if the position is classified as moving. A particular *mm*(*x,y,t*) is classified as outlier if it is further away from the mean than a certain number of standard deviations. Our improvements to the distribution learning algorithm are described in Section 2.3.

2.3. Dynamic distribution learning and outlier detection

Consider labeling each video position as moving or stationary (background) 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. 2(a)), the task reduces to dynamic estimation of the mean and standard deviation of *mm*. The only assumption that we make about the distribution of values of function *f* is that it has a prominent right tail (general Gaussian distribution).

Given a function *f* of one variable, we compute initial values of *mean*(t_0) and variance $\sigma^2(t_0)$ of all values *f*(*t*) in some initial interval *t*=1, …, t_0. For $t>t_0$, we update the estimates using the technique described in the next paragraph. An outlier is detected at time $t>t_0$ if the standardized feature value is sufficiently large, i.e., when

$$\frac{f(t) - mean(t-1)}{std(t-1)} > C_1, \qquad (2.1)$$

where C_1 is a constant and $std(t) = \sqrt{\sigma^2(t)}$

Once an outlier is detected at time t_1, value $f(t_1)$ is labeled as an outlier. We update the nominal state at time *t*, if the standardized feature value drops below a threshold $C_2 < C_1$, i.e.,

$$\frac{f(t) - mean(t-1)}{std(t-1)} < C_2, \qquad (2.2)$$

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 (2.2) holds. Then, *mean*(*t*) and *std*(*t*) are updated using running average :

$$mean(t) = u \cdot mean(t-1) + (1-u) \cdot f(t)$$

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

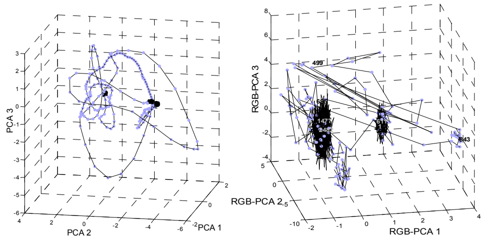
In our experiments, we use C_1=9, C_2=3, and *u*=0.99 in the case of the detection of moving blocks for *f*=*mm*.

3. Objective performance evaluation

In this section we introduce an objective method of performance evaluation and use such a method to compare the proposed use of spread measure of texture vectors to the Gaussian mixture model based technique 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. First we also show in Section 3.1 that the Gaussian mixture model on texture vectors significantly outperforms the original representation used in [14] (RGB color values on a pixel level). To do this we need to consider RGB color videos in Section 3.1.

3.1. Motion orbits in texture space

Recall that we use texture vectors composed of the first *N* PCA components of each spatiotemporal block vector. If *N=3*, 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. 3(a), we see the orbit for the block (*24,28*) of a RGB color video. 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 demonstrate how noisy RGB color values of a single pixel can be in Fig. 3 (b), 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.



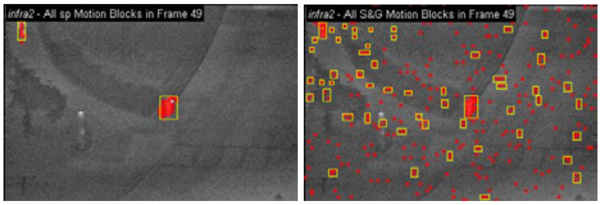
(a) (b)

Fig. 3. Orbits of block vectors with blue-gray dots corresponding to the frames where the block was identified as moving by the proposed method (a) *Campus 1* video: block *I*=24, *J*=28; (b) Standardized PCA components of RGB pixel values at pixel location (*185,217*) that is inside of block (*24,28*).

For better visualization, we show the linearly transformed space of PCA projections of the original RGB color values. We can also see two distribution components corresponding to the background. To allow us a proper comparison to the results in Fig. 3(a) (computed by our local variance technique), we carried over the moving (blue-gray) dot labels from Fig. 3(a). Notice that the moving dots incorrectly became parts of two background components. Since the background variance in the sp block-based approach is much smaller, the usage of sp texture vectors results in effective noise reduction in comparison to using "raw" pixels. Hence, any technique to detect moving objects as outliers will perform much better on sp blocks than on raw pixel values.

3.2. Decreased sensitivity to noise

It is well-known that noise to signal ratio in IR videos is higher than in visible light videos. The IR noise can be viewed as jitters in IR values. Fig. 4(b) illustrates the performance of the S&G method [14] on IR pixels values on video from [19]. We see a large number of false-positives, some of which has the size of moving objects. The usage of the proposed *sp* texture vectors eliminates very effectively the IR jitter noise as we can see in Fig. 4(a).


infra2 - All sp Motion Blocks in Frame 49
infra2 - All S&G Motion Blocks in Frame 49

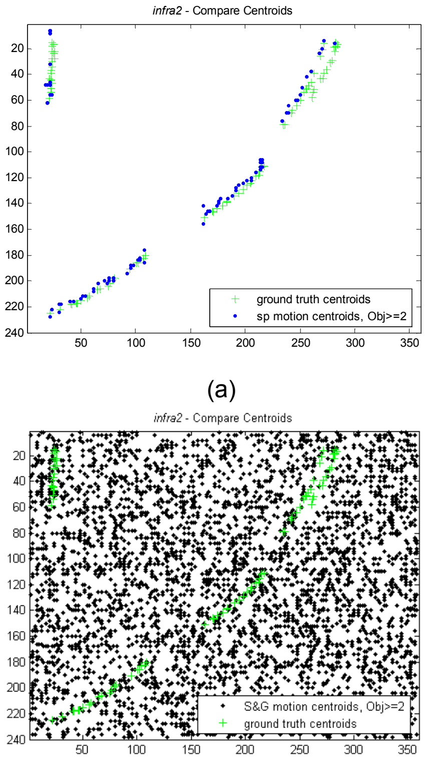
(a) (b)

Fig. 4. *Infra2* video frame 49 with detected motion: (a) the proposed outlier detection based on sp blocks; (b) S&G Gaussian mixture model [14].

3.3. Ground truth data evaluation

The video clips and corresponding ground truth data used in our evaluation were obtained from Ohio State University Thermal Pedestrian Database [19]. Video was captured using a Raytheon 300D thermal sensor core with 75 mm lens. Camera was mounted on an 8-story building overlooking a pedestrian intersection on the Ohio State University (OSU) campus. Ground truth data gives us number of objects and their centroids in each video frame. In order to compare the two methods to the ground truth data, we must detect motion, find objects from motion data, and compute their centroids. Process each video sequence to identify motion on block level and establish motion/no motion binary image as described in Section 2. The output from motion detection is fed into object labeling algorithm to measure the object's region of interest and centroid location. Connected components are used to establish motion regions of interests with a minimum of 2 blocks per region. We evaluate motion block components as 8-connected objects.

Ground truth centroids for *Infra2* video are shown in Fig. 5. All ground truth centroid are shown in green to visualize all motion paths simultaneously.



(a)

(b)

Fig. 5. Projection of all ground truth data for *Infra2* video with objects size >= 2; (a) Ground truth data and sp motion centroids; and (b) Ground truth data and Stauffer-Grimson Gaussian mixture model centroids.

Observe that the motion centroids (in blue) coincide very well with the ground truth for the proposed method (Fig. 5(a)). On average, our *sp* motion tracking centroid distance from ground truth data was 4.62 pixels with standard deviation of 2.54 pixels for *Infra2* video. The IR jitter noise on the pixel level makes the detected moving objects by S&G method [14] (without post processing) to form a dense set in the video plane (Fig. 5(b)).

4. Object tracking

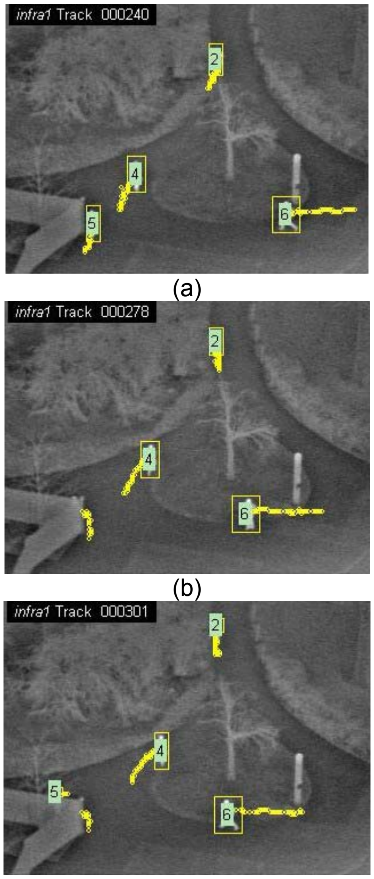
Robust detection of motion regions in IR videos introduced in Section 2 is the basis for tracking moving object. We have modified and simplified the minimum cost computation introduced by [22]. Each new detected motion region $\boldsymbol{i}$ in frame $\boldsymbol{t}$ has a know bounding box $\boldsymbol{B}_i$, centroid location $\boldsymbol{X}_i$ and initial zero velocity $\boldsymbol{V}_i$. Known motion region $\boldsymbol{L}$ in a frame $\boldsymbol{t}$-1 has centroid $\boldsymbol{X}_L$ and velocity $\boldsymbol{V}_L$ and a predicted centroid $\boldsymbol{X}_{LP}$ in frame $\boldsymbol{t}$. Minimum cost $\boldsymbol{C}_{Li}$ between $\boldsymbol{X}_L$ and $\boldsymbol{X}_i$ is computed based on the predicted location of known track labeled regions and new detected motion regions.

$$X_{LP}^t = X_L^{t-1} + V_L^{t-1}$$

$$C_{Li} = \left\| X_{LP}^t - X_i^t \right\|$$

$$M_i = \sum_j^L m_{ji}$$

where $m_{Li} = 1$ if $X_{LP}^t \cap B_i^t$, and 0 otherwise.



(a)

(b)

(c)

Fig. 6. *Infra1* video frame sequence 240-301 showing object ID, bounding box and tracking trail. (a) Object 5 walking along the fence (b) Object 5 turning corner and hidden behind the fence; and (c) Head of Object 5 reapers behind the fence, tracking continues.

If $\boldsymbol{M_i} = 0$ then there is no known region association with any labeled region $\boldsymbol{L}$. If the best $\boldsymbol{C_{Li}}$ is less than the minimum cost threshold $\boldsymbol{T_C}$, then L is selected as the best match. Otherwise new tracking motion region is created with initial velocity set to 0. If $\boldsymbol{M_i} = 1$ then there is exactly one tracking $\boldsymbol{L}$ region association (Fig. 6). If however $\boldsymbol{M_i} > 1$, then there is more than one tracking centroid within $\boldsymbol{B_i}$ (merge or crossover of motion regions). In this case $\boldsymbol{X_i}$ is updated using only the predicted location and the velocity remains constant.

Each labeled object L has a time to stop tracking factor associated with it, $\boldsymbol{T_L}$. For each selected associated pair (L,i), the $\boldsymbol{T_L}$ is set to the maximum allowed time to track value $\boldsymbol{T_{Lmax}}$. All labeled objects $\boldsymbol{L}$ not associated with any current detected motion regions $\boldsymbol{i}$ has its $\boldsymbol{T_L}$ decremented by 1. Once $\boldsymbol{T_L}$ reaches 0 the labeled object $\boldsymbol{L}$ is no longer used in computing the minimum cost association between pairs (L,i).

The minimum cost computation as proposed by [22] is also based on the size of the bounding box and predicted size computation. In our experiments the size component of the minimum cost computation is negligible and therefore not used.

5. Conclusion

In this paper we show a much simpler but also a more adequate model for motion detection for thermal infrared surveillance videos. It can significantly reduce the processing time in comparison to the Gaussian mixture model, due to smaller complexity of the local variation computation. 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. The minimum cost tracking based on *sp* motion regions is the foundation for more sophisticated object classification algorithm.

6. References

[1] D. Buttler, S. Sridharan, and V.M. Bove, “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 ed., Int. Thomson Publishing Company, Belmont, 2000.

[4] Duda, R., P. Hart, and D. Stork, *Pattern Classification*, 2nd ed., 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] R. Jain, D. Militzer, and H. Nagel, “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] O. Javed, K. Shafique, and M.A. Shah,. “Hierarchical approach to robust background subtraction using color and gradient information”. *In Proc. IEEE Workshop 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] L.J. Latecki, R. Miezianko, and D. Pokrajac. “Motion Detection Based on Local Variation of Spatiotemporal Texture”. *CVPR Workshop on OTCBVS*, Washington, July 2004.

[12] Temple University ViVi Lab video results and data URL: 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, and W. E. L. Grimson, “Learning patterns of activity using real-time tracking”, *IEEE PAMI* 22(8) (2000), pp. 747–757.

[15] Westwater, R., and B. Furht, *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, and 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 ICME*, 2004.

[19] J. Davis, and M. Keck, “A two-stage approach to person detection in thermal imagery” *In Proc. Workshop on Applications of Computer Vision*, January 2005.

[20] Haralick, R.M., and L.G. Shapiro, *Computer and Robot Vision*, Volume I, Addison-Wesley, 1992.

[21] 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.

[22] J. Omar, and M. Shah, "Tracking and Object Classification for Automated Surveillance", *The seventh European Conference on Computer Vision*, Copenhagen, May 2002.

[23] L.J. Latecki, R. Miezianko, and D. Pokrajac. “Activity and Motion Detection Based on Measuring Texture Change”. *International Conference on Machine Learning and Data Mining MLDM’2005*, Leipzig, June 2005.