Small Moving Targets Detection Using Outlier Detection Algorithms

Natasa Reljin[1], Samantha McDaniel[1], Dragoljub Pokrajac[1], Nebojsa Pejcic[1], Tia Vance[1], Aleksandar Lazarevic[2], Longin Jan Latecki[3]
[1]Delaware State University, 1200 N DuPont Hwy, Dover, DE USA 19901;
[2]United Technologies Research Center, 411 Silver Lane, MS 129-15, East Hartford, CT USA 06108;
[3]Temple University, 1801 N Broad Street, Philadelphia, PA USA 19122

ABSTRACT

Recent research in motion detection has shown that various outlier detection methods could be used for efficient detection of small moving targets. These algorithms detect moving objects as outliers in a properly defined attribute space, where outlier is defined as an object distinct from the objects in its neighborhood. In this paper, we compare the performance of two incremental outlier detection algorithms, namely the incremental connectivity-based outlier factor and the incremental local outlier factor to modified Stauffer-Grimson algorithm. Each video sequence is represented with spatial-temporal blocks extracted from the raw video. Principal component analysis (PCA) is applied on these blocks in order to reduce the dimensionality of extracted data. Extensive experiments performed on several data sets, including infrared sequences from OSU Thermal Pedestrian Database repository, and data collected at Delaware State University from FLIR Systems PTZ cameras have shown promising results in using outlier detection for detection of small moving targets.

Keywords: Incremental outlier detection, connectivity-based outlier factor, local outlier factor, modified Stauffer-Grimson method, motion detection, principal component analysis, outlier, spatial-temporal blocks

1. INTRODUCTION

Motion detection is nowadays a wide spread area of research. There is a high interest in the uses of motion detection in everyday life, such as in traffic monitoring, homeland security, video surveillance, etc. There are several approaches for identifying locations of the moving objects, which are based on the background subtraction [1], multimodal Gaussian distributions [2], the statistical models [3], etc. Some of the popular techniques are pixel-based [1, 2], meaning they are sensitive to the presence of the noise, which can appear due to the capturing device, or sensitive to the slight changes in the illumination in the scene. As a direct consequence of the pixel-based detection, false alarms and artifacts are very frequent.

In this paper, we assume that the feature vectors representing background portions of the video correspond to "normal" behavior, while the vectors corresponding to motion correspond to exceptions (outliers). Such an assumption is valid if moving objects account for a small portion of the frame and if the scene is static for the majority of the time. We compare two incremental outlier detection algorithms, (incremental local outlier factor (LOF) algorithm [4] and the incremental connectivity-based outlier factor (COF) [5]), with the modified Stauffer-Grimson algorithm [6]. All of the proposed outlier detection algorithms are suitable for motion detection in real-video sequences because they handle constant influx of the data into the surveillance system.

Unlike the majority of popular techniques, in the proposed approach we represent videos with spatial-temporal blocks. In the case of the incremental outlier detection algorithms, we label an object as moving if the incrementally computed outlier factor values (*LOF* or *COF*) are above a predefined threshold. In contrast, the modified Stauffer-Grimson algorithm labels a block as moving if the minimal squared Mahalanobis distance from nearest Gaussian cluster is above a predefined threshold value. The proposed algorithms have been applied to IR videos.

2. RELATED WORK

A significant amount of work is dedicated to motion detection in video imagery. Remagnino et al. [7] and Collins et al. [8] provided an overview of various approaches of the motion detection. Some of the approaches rely on comparison of the color or intensities of pixels in the incoming video frame to a reference image. For example, Jain et al. [1] use simple intensity comparison to reference images, so that the values above a given threshold identify the pixels of the moving objects. Numerous approaches are based on the appropriate statistics of color or gray values over time at each pixel location (e.g., the segmentation by background subtraction [9], eigenbackground subtraction [10], etc). Wren et al. [3] were the first who used a statistical model of the background instead of a reference image. Stauffer and Grimson [2] introduced the approach for motion detection based on the adaptive Gaussian mixture model of the color values distribution over time at each pixel location. Each Gaussian component in the mixture is defined by its prior probability, mean vector, and a covariance matrix.

As mentioned in the previous section, the majority of the existing approaches are pixel based. However, the application of region level techniques, e.g. such as proposed by Javed et al. [11], can lead to increased stability when detecting objects in adverse conditions, such as substantial presence of noise. Nevertheless, Javed et al. [11] and related approaches (e.g., Buttler et al. [12]) aimed to improve the Stauffer-Grimson algorithm to still perform motion detection on a pixel level (i.e., only the postprocessing of pixel-based motion detection results is region based). Pokrajac et al. proposed the use of a variation of the Stauffer and Grimson approach, based on the computation of spatial-temporal blocks and the application of principal component analysis (PCA) to reduce dimensionality of feature vectors [6]. The used method is purely region-based. Latecki et al. [13] proposed the method of detecting the moving objects based on local variance of PCA projections of feature vectors obtained from spatial-temporal blocks. If the variance measure (so-called motion measure) is large, then it is more likely that the motion occurred.

Outliers can be identified as data different from its neighbors in certain feature space, with respect to some geometric properties: neighborhood density, single-link distance, other approaches [14, 15]. Incremental outlier detection algorithms, as a special class of outlier detection algorithms can detect if data is an outlier immediately after the data arrives in the database [5], which are especially suitable techniques for processing real-time videos. In our previous work [16], motion detection based on the incremental connectivity-based outlier factor (incCOF) algorithm is used on RGB and IR video sequences. An incremental version of LOF is proposed by Pokrajac et al. [4].

3. PROPOSED METHODOLOGY

In this research, we use video sequences from stationary cameras. Each video is represented as a three-dimensional (3D) array of gray-level or monochromatic infrared pixel values $g_{i,j,t}$ at time instance t and at a pixel location i,j [13]. In addition, every video is characterized by the total number of frames, N_T and by two spatial dimensions N_X and N_Y that characterize the number of pixels for each frame in the horizontal and vertical directions, respectively. We divide each frame in a video sequence into N_{BLOCK}xN_{BLOCK} disjoint squares that cover the entire image. Spatiotemporal (3D) blocks are obtained by combining squares in F consecutive frames at the same video plane location. The obtained 3D blocks are represented as N_{BLOCK}xN_{BLOCK}xF dimensional vectors of gray level or monochromatic infrared pixel values. These vectors are representing the feature vectors. Since the dimension (length) of such feature vector is very large, we use principal component analysis (PCA) [17] to reduce it in the following way. First, we compute a projection of the original block vector to a vector of significantly lower length d. Then, the obtained d dimensional vectors form a compact spatiotemporal texture representation for each block. The PCA projection matrices are computed separately for all video plane locations (a set of disjoint N_{BLOCK}xN_{BLOCK} squares in our experiments). Finally, each PCA projection matrix is created to contain d eigenvectors corresponding to the largest eigenvalues of vectors representing spatial-temporal blocks.

The main premise of this work is that moving objects can be considered as outliers in the feature space. An outlier is defined as a data record very different from the majority of data based on a particular measure [18]. Hence, to identify moving objects, we employ techniques for outlier detection. In the case of video sequences, when new frames continuously arrive into the system, *static* outlier detection algorithms (that need all the data prior to determining outliers) cannot be applied. Thus, we use incremental outlier detection algorithms, incLOF [4] and incCOF [5]. Large values of *LOF* (*COF*) factors will correspond to the unusual feature vectors at the location, which we presume belong to the objects in motion. Namely, the proposed incremental LOF (COF) algorithms compute *outlierfactors* (*LOF*, *COF* value) for each feature vector inserted into the data set. The corresponding block is marked as moving if *outlierfactor>Th* where *Th* is an appropriately chosen threshold. Simultaneously, *LOF* (*COF*) values for previously inserted feature vectors are updated if needed. In contrast, the modified Stauffer-Grimson [6] algorithm incrementally learns a mixture of Gaussian distributions and detect moving objects to correspond to points far enough from their closest distributional component. Here, we provide an overview of incremental LOF (IncLOF), incremental COF (IncCOF), and the modified Stauffer-Grimson algorithms. For details, please see [4, 5, 6].

3.1 Incremental LOF algorithm

LOF algorithm [14] assigns to each data record a degree of being an outlier – *local outlier factor* (*LOF*). Data records with high *LOF* have local densities smaller than their neighborhood and typically represent stronger outliers [14]. The algorithm for each data record q computes the k-distance(q) as the Euclidean distance to the k-th nearest neighbor of q, as well as the *reachability distance* (*reach-dist*) defined as:

$$reach-dist(q,p) = \max(d(q,p), k-\text{distance}(p)) \tag{1}$$

where $d(q,p)$ is the Euclidean distance. Subsequently, the *local reachability density* (*lrd*) of a data record is computed as an inverse of the average reachability distance of k-nearest neighbors of the data record q:

$$lrd(q) = \frac{1}{\sum_{p \in kNN(q)} reach-dist_k(q,p)/k} \tag{2}$$

Finally, *LOF* is computed as the ratio of the average local reachability density of k-nearest neighbors of a data record and the local reachability density at the data record itself:

$$LOF(q) = \frac{\frac{1}{k}\sum_{p \in kNN(q)} lrd(p)}{lrd(q)} \tag{3}$$

Incremental LOF (incLOF) algorithm performs two steps: insertion of the new data record, when it computes reachability distance, local reachability density and *LOF* values of a new point; and maintenance, when it updates k-distances, *reach-dist*, *lrd* and *LOF* values for affected existing points. When a new data record n is inserted, only a limited number of neighboring data records must be updated. We demonstrated that the number of updates of LOF and its parameters (*lrd, reach-dist, k-distances*) using the algorithm from Figure 1 is restricted to a neighborhood of the inserted point [4].

```
Incremental LOF_insertion(Dataset S)
• Given: Set S {p1, … ,pN} pi∈R^D, where D corresponds
to the dimensionality of data records.
• For each data point pc in data set S
  - insert(pc)
  - Compute kNN(pc)
  - (∀pj∈ kNN(pc))
        compute reach-distk(pc,pj) using Eq.(1);
  //Update_neighbors of pc
  - S_update_k_distance = kRNN(pc);
  - (∀pj ∈ S_update_k_distance)
        update k-distance(pj)
  - S_update_lrd = S_update_k_distance;
     - (∀pj ∈ S_update_k_distance), (∀pi∈kNN(pj)\{pc})
        reach-distk(pi,pj) = k-distance(pj);
        if pj ∈ kNN(pi)
            S_update_lrd = S_update_lrd ∪{pi};
  - S_update_LOF = S_update_lrd;
  -(∀pm ∈ S_update_lrd)
        update lrd(pm) using Eq. (2);
        S_update_LOF = S_update_LOF ∪ kRNN(pm);
   - (∀pl ∈ S_update_LOF)
        update LOF(pl) using Eq.(3);
   - compute lrd(pc) using Eq.(2);
   - compute LOF(pc) using Eq.(3);
• End //for
```

Figure 1. The general framework for the insertion of new data record in incremental LOF algorithm.

3.2 Incremental COF algorithm

COF algorithm [15] identifies outliers based on neighborhood density and neighborhood connectivity. This algorithm computes the average chaining distance (*ac-dist*) and identifies data points as outliers whose *COF* factor is large enough. The *COF* factor is defined as the ratio of the *ac-dist* at a point and the *ac-dist* at its *k*-nearest neighborhood [15]:

$$COF(p) \equiv \frac{ac-dist_{N_k(p)\cup p}(p)}{\frac{1}{k}\sum_{o\in N_k(p)} ac-dist_{N_k(o)\cup o}(o)}. \tag{4}$$

COF algorithm considers the Set Based Nearest path (SBN), which is basically a minimal spanning tree with k nodes, starting from the record in question. The average chaining distance from a point to its *k-nearest* neighbor, $ac\text{-}dist_{Nk(p)\cup p}(p)$, is defined as the weighted average of lengths of edges on the SBN where the larger weights are assigned to edges appearing earlier on the trail:

$$ac-dist_{N_k(p)\cup p}(p) \equiv \sum_{i=1}^{k} \frac{2(k+1-i)}{k(k+1)} dist(e_i) . \tag{5}$$

Similarly as in incLOF, when the new data record arrives, incremental COF (incCOF) performs the following two steps: insertion of the new record into the database and computation of its *ac-dist* and *COF*; maintenance, when *ac-dist* and *COF* values need to be updated for a limited number of the affected records already in the database. The pseudocode for incCOF is given in Figure 2.

```
IncCOF_insertion(Dataset S)
Given: Set S {p_1, … ,p_N} p_i∈R^D to be inserted into
the database
For ∀p_c∈S
                              insert(p_c);
        compute N_k(p_c)=kNN(p_c)
        compute ac-dist_Nk(pc)∪pc(p_c) using Eq. (5)
        S_update_ac_dist =reverse k-NN(p_c);
        //candidates for update of ac_dist;
        (∀p_j ∈ S_update_ac_dist)
                update ac-dist_Nk(pi)∪pi (p_j); //Eq.
(5)
        S_update_COF = S_update_ac_dist;
        (∀p_m ∈ S_update_ac-dist)
                S_update_COF = S_update_COF∪reverse k-NN(p_m);
      (∀p_l ∈ S_update_COF)
                update COF(p_l) using Eq. (4);
      compute COF(p_c), Eq.(5);
End //for
```

Figure 2. The general framework of the insertion of new data record in incremental COF algorithm.

3.3 Modified Stauffer-Grimson algorithm

This algorithm is a variation of the incremental EM algorithm used for estimation of Gaussian mixtures, extended by an additional mechanism for the detection of blocks corresponding to moving objects [6]. The mixture consists of K components, where each component is specified by its estimated mean vector $\boldsymbol{\mu}_l$, the diagonal convariance matrix $diag\left(\sigma_l^2, \sigma_l^2, \ldots, \sigma_l^2\right)$, and distributional priors w_l, l=1,…,K.

At each time instant t the squared Mahalanobis distances of the spatiotemporal block with respect to the distribution components of the mixture are estimated for all blocks that appeared at the same position at the previous time instants [6]. The block is considered an outlier, if the minimal squared distance is above a predefined threshold value Th. The l_r-th distribution component that has the smallest estimated prior probability w_{lr} is replaced by the new Gaussian distribution, which is centered around that block vector. This mechanism is called *reset* [6]. On the other hand, if the minimal squared Mahalanobis distance to one of the distribution components is below the threshold, then the block is not considered as an outlier. However, it is possible that the block belongs to the moving object. In order to determine if the block is part of the moving object, there is a mechanism called *hold* [6]. That block will be labeled as 'moving' only in the following two cases: 1) if the closest distributional component has a relatively large variance but small prior probability; 2) if an outlier has been detected within G frames that precede the current frame at the same block position [19].

4. RESULTS

We tested density-based incremental LOF motion detection and connectivity-based incremental COF approaches on a video sequence from publicly accessible repository OSU Thermal Pedestrian Database - OTCBVS (referred to as *roofCam*) [20]. In the experiments, we projected spatiotemporal vectors to vectors of 3 principal components (d=3). We used spatiotemporal block sizes of 8x8x3 and 4x4x3. In addition, connectivity-based incremental COF and modified Stauffer-Grimson algorithms were tested on LWIR video sequence *1_11_2009nfov2_seg3*, taken at Delaware State University using FLIR Systems 35*140mm PTZ camera [21]. For this experiment, only a block size of 4x4x3 was used, and vectors were projected to 3 principal components.

In the Figure 3, we show results of applying the incremental COF algorithm on the 107th frame of the *roofCam* video. It is obvious that in this case the choice of the three parameters of the algorithm, critical to the quality of motion detection, is not optimal. E.g., the block size used in the particular case is too big. The comparison between Figures 3 and 4 indicate that the slight improvement in the quality of detected motion is possible with the decrease in the block size. A comparison to the results shown in [16] leads to the same conclusion. However, a too small block size could lead to the artifacts (false positives) due to a lack of noise suppression.

Figure 3. incCOF algorithm on 107th frame of *roofCam* video, with k=20, block size 8x8x3 and threshold 2.4.

Figure 4. incCOF algorithm on 107th frame of *roomCam* video, with k=20, block size 4x4x3 and threshold 2.4.

Block size is not the only parameter of which its setting can affect the quality of motion detection. The following Figure 5 demonstrates that significant improvements can be achieved by the optimal choice of the number of nearest neighbors k and the threshold value. Please note the difference between Figures 4 and 5. In Figure 5, both persons were completely detected, which lead us to the conclusion that with the optimal choice of the parameter k better results could be achieved.

Figure 5. incCOF algorithm on 107th frame of *roomCam* video, with k=40, block size 4x4x3 and threshold 2.4.

Figure 6. incLOF algorithm on 107th frame of *roomCam* video, with k=20, block size 4x4x3 and threshold 3.7.

Figure 6 shows results of the incremental LOF algorithm on the same frame (107th), with the same block size as in Figure 5, but for a smaller number of nearest neighbors, k=20. As the results show, incLOF was able to detect the small object entering the scene (bottom of the frame) for which incCOF failed to detect.

Incremental LOF and incremental COF algorithms showed good results on the *roofCam* video. In addition, we compared the extended Stauffer-Grimson method and incremental COF on a LWIR video sequence *1_11_2009nfov2_seg3*, taken at Delaware State University using FLIR Systems 35*140mm PTZ camera [21] (see Figure 7.). In this video, the resolution was about 1inch/pixel.

Figure 7. FLIR Systems 35*140mm PTZ camera at Delaware State University.

Here, the spatiotemporal vectors were projected to vectors of 3 principal components. According to the aforementioned results on the *roofCam* video, for this video sequence, we used spatiotemporal blocks of size 4x4x3 only. Also, as for the incremental COF method, we used k=40, which was demonstrated to be optimal number of nearest neighbors for the *roofCam* video. The results for incCOF algorithm applied on the 404th frame of the *1_11_2009nfov2_seg3* video are shown in Figure 8. As a threshold value, we used the same value 2.4 as the case of the *roofCam* sequence.

Figure 8. incCOF algorithm on 404th frame of *1_11_2009nfov2_seg3*, with k=40, block size 4x4x3 and threshold 2.4.

We varied the threshold value for the incremental COF algorithm. In Figure 9 we show the same 404th frame of the same video sequence, but using the threshold value 1.8. When comparing Figures 8 and 9, the difference in detection of moving objects is visible. For the lower value of the threshold, the moving objects are better detected. However, the motion detection is more prone to noise, i.e., the number of false positives seems to increase (Figure 9).

For both values of the threshold, the algorithm is able to detect all moving objects. In Figures 10 and 11 results are shown for the incCOF algorithm which is applied to the 935th frame of the video sequence, with thresholds 1.8 and 2.4, respectively. We can observe that, besides moving humans, the algorithm was able to detect the moving vehicle (top part of the image).

Figure 9. incCOF algorithm on 404th frame of *1_11_2009nfov2_seg3*, with *k*=40, block size 4x4x3 and threshold 1.8.

Figure 10. incCOF algorithm on 935th frame of *1_11_2009nfov2_seg3*, with *k*=40, block size 4x4x3 and threshold 1.8.

Figure 11. incCOF algorithm on 935th frame of *1_11_2009nfov2_seg3*, with *k*=40, block size 4x4x3 and threshold 2.4.

The modified Stauffer-Grimson method was applied to the same *1_11_2009nfov2_seg3* video sequence, with the same block size, 4x4x3. The threshold value was set to 2.5. Figures 12 and 13 show the results obtained by this algorithm, on the same frames used for the incCOF, 404th and 935th. We can observe that all the moving objects are detected. Visual comparisons of the incCOF and the extended Stauffer-Grison methods suggests that better results are obtained with the extended Stauffer-Grimson method.

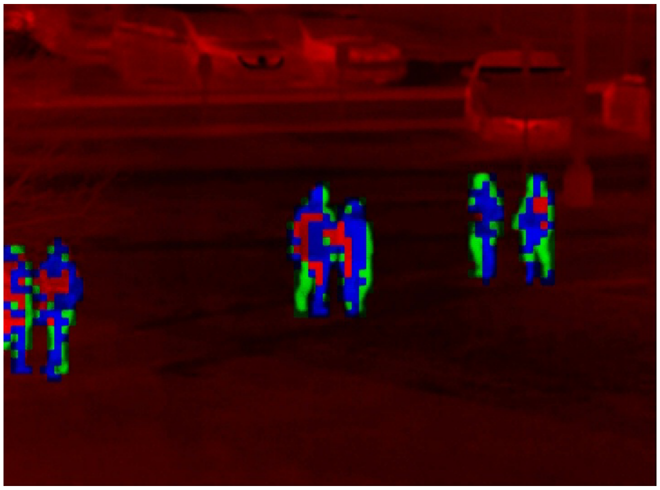

Figure 12. Extended Stauffer-Grimson method on 404th frame of *1_11_2009nfov2_seg3* video, with block size 4x4x3 and threshold 2.5.

Figure 13. Extended Stauffer-Grimson method on 935th frame of *1_11_2009nfov2_seg3* video, with block size 4x4x3 and threshold 2.5.

5. CONCLUSION

In this paper, we first demonstrated viability of the incremental outlier detection methods incLOF and incCOF for the motion detection, and then compared the incremental outlier detection algorithms with a modified Stauffer-Grimson method. All the methods are performed on spatial-temporal blocks projected into a lower dimensional space using principal component analysis. We experimented with various values of block size, number k of nearest neighbors, as well as threshold values. By proper parameter settings, we were able to detect practically any motion in the presence of noise (presence of random artifacts in the detected video).

Our results indicate that k – the size of the neighborhood — is the most important parameter for the tested incremental outlier detection algorithms. Namely, a data record is labeled as an outlier only relative to its neighbors. Very low values of k mean that some moving objects cannot be detected due to the insufficiently large neighborhoods being considered.

Also, such choices of *k* can lead to unacceptable levels of false alarms, especially in the case of incLOF. In addition, the optimal choice of the threshold value can, to some degree, offset for a too low choice of *k*. In order to correctly detect moving objects, high threshold values are required for low *k* and vice versa.

Incremental LOF can produce better results on the used dataset when smaller size *k* of the nearest neighborhood is used compared to the COF algorithm applied to the same dataset. The experimental results on the *roofCam* video suggest that incLOF performed better with *k*=20 than did incCOF with double the size of the nearest neighborhood (*k*=40). On the other hand, incCOF is highly dependent on the optimal choice of *k*. Our results suggest that for a sufficiently high parameter *k*, the threshold value can belong to a broader range without significantly compromising the quality of detection. Furthermore, overly big block size can mask small movements.

The results suggest that the incLOF algorithm is more sensitive on moving objects in the case of the *roomCam* video in comparison to the incCOF. However, in the case of the *1_11_2009nfov2_seg3* video, incCOF detected all moving objects, for the various threshold values. The extended Stauffer-Grimson method showed slightly better results than those of the incCOF. It remains to be determined why incLOF and incCOF do not provide better results. The hypothesis is that the current versions of the incremental LOF and COF algorithms have infinite capacity of memory, i.e. they do not 'forget' the past. Our work in progress involves the developement of algorithms which have only finite memory capacity and hence have the ability to accentuate more recent data.

ACKNOWLEDGEMENTS

This work has been partially supported by NSF (CREOSA grant #HRD-0630388), NIH (grant #2 P20 RR016472-04) and DoD/DoA (45395-MA-ISP, #54412-CI-ISP).

REFERENCES

[1] Jain, R., Militzer, D. and Nagel, H., "Separating nonstationary from stationary scene components in a sequence of real world TV images," Proc. IJCAI, 612-618 (1977).
[2] Stauffer, C. and Grimson, W. E. L., "Learning patterns of activity using real-time tracking," IEEE Transactions on PAMI 22(8), 747-757 (2000).
[3] Wren, C., Azarbayejani, A., Darrell, T. and Pentland, A. P., "Pfinder: Real-time Tracking of the Human Body," IEEE Transactions on PAMI 19(7), 780-785 (1997).
[4] Pokrajac, D., Lazarevic, A. and Latecki, L. J., "Incremental local outlier detection for data streams," IEEE Symp. Computational Intelligence and Data Mining, 504-515 (2007).
[5] Pokrajac, D., Reljin, N., Pejcic, N., and Lazarevic, A., "Incremental Connectivity-Based Outlier Factor Algorithm," Proc. BCS, 211-224 (2008).
[6] Pokrajac, D. and Latecki, L. J., "Spatiotemporal Blocks-Based Moving Objects Identification and Tracking," Proc. IEEE VSPETS, 70-77 (2003).
[7] Remagnino, P., Jones, G. A., Paragios, N. and Regazzoni, C. S., [Video-Based Surveillance Systems], Kluwer Academic Publishers, (2002).
[8] Collins, R. T., Lipton, A. J. and Kanade, T., "Introduction to the Special Section on Video Surveillance," IEEE Transactions on PAMI 22(8), 745-746 (2000).
[9] Haritaoglu, I., Harwood, D. and Davis, L., "W4: Real-Time Surveillance of People and Their Activities," IEEE Transactions on PAMI 22(8), 809-830 (2000).
[10] Oliver, N. M., Rosario, B. and Pentland, A. P., "A Bayesian Computer Vision System for Modeling Human Interactions," IEEE Transactions on PAMI 22(8), 831-843 (2000).
[11] Javed, O., Shafique, K. and Shah, M. A., "Hierarchical approach to robust background subtraction using color and gradient information," IEEE Workshop on Motion and Video Computing, 22-27 (2002).
[12] Buttler, D., Sridharan, S. and Bove, V. M., "Real-time adaptive background segmentation," Proc. IEEE Int. Conf. Multimedia and Expo, 341-344 (2003).
[13] Latecki, L. J., Miezianko, R. and Pokrajac, D., "Motion Detection based on Local Variation of Spatiotemporal Texture," Proc. Conf. OTCBVS, 135-141 (2004).

[14] Breunig, M. M., Kriegel, H. P., Ng, R. T. and Sander, J., "LOF: Identifying density based local outliers," Proc. ACM SIGMOD Int. Conf., 93-104 (2000).
[15] Tang, J., Chen, Z., Fu, A. W. and Cheung, D., "A robust outlier detection scheme for large data sets," Proc. Pacific-Asia Conf. Knowledge Discovery and Data Mining, (2002).
[16] Pejcic, N., Reljin, N., McDaniel, S., Pokrajac, D. and Lazarevic, A., "Detection of moving objects using incremental connectivity outlier factor algorithm," Conf. ACM SE, (2009).
[17] Jolliffe, I. T., [Principal Component Analysis. 2nd edn.], Springer Verlag (2002).
[18] Barnett, V. and Lewis, T., [Outliers in Statistical Data], John Wiley and Sons, New York, NY (1994).
[19] Pokrajac, D., Zeljkovic, V. and Latecki, L. J., "Spatial-temporal algorithm for moving objects detection in infrared video sequences," Int. Conf. TELSIKS, (2005).
[20] http://www.cse.ohio-state.edu/otcbvs-bench/
[21] http://tesla.cis.desu.edu/Data/LONGIN/VIPMobile/1_11_2010videos/1_11_2009nfov2_seg3/