Using spatiotemporal blocks to reduce the uncertainty in detecting and tracking moving objects in video

Longin Jan Latecki

Department of Computer and Information Sciences,
Temple University,
3rd floor Wachman Hall, 1805 N. Broad St.,
Philadelphia PA 19122, USA
Fax: 1 215 204 5082 E-mail: latecki@temple.edu

Vasileios Megalooikonomou*

Data Engineering Laboratory (DEnLab),
Center for Information Science and Technology,
Department of Computer and Information Sciences,
Temple University, 3rd floor Wachman Hall, 1805 N. Broad St.,
Philadelphia PA 19122, USA
Fax: +1 215 204 5082 E-mail: vasilis@temple.edu
*Corresponding author

Roland Miezianko

Department of Computer and Information Sciences,
Temple University,
3rd floor Wachman Hall, 1805 N. Broad St.,
Philadelphia PA 19122, USA
E-mail: rmiezian@temple.edu

Dragoljub Pokrajac

Computer and Information Science Department,
Delaware State University,
1200 N Dupont Hwy, Dover, 19901 DE
E-mail: dragoljub.pokrajac@comcast.net

Abstract: We present a novel method for detecting moving objects in videos. The method represents videos using spatiotemporal blocks instead of pixels. Dimensionality reduction is performed to obtain a compact representation of each block's values. The block vectors provide a joint representation of texture and motion patterns. The motion detection and tracking experiments demonstrate that our method although simpler than a state-of-the-art method based on the Stauffer-Grimson Gaussian mixture model has superior performance. It reduces both the instability and the processing time making real-time processing of high resolution videos and efficient analysis of large scale video data feasible.

Keywords: video analysis; uncertainty; moving object detection; dimensionality reduction; Gaussian mixture model.

Reference to this paper should be made as follows: Latecki, L.J., Megalooikonomou, V., Miezianko, R. and Pokrajac, D. (2006) 'Using spatiotemporal blocks to reduce the uncertainty in detecting and tracking moving objects in video', *Int. J. Intelligent Systems Technologies and Applications*, Vol. 1, Nos. 3/4, pp.376–392.

Biographical notes: Longin Jan Latecki is the winner of the Pattern Recognition Society Award together with Azriel Rosenfeld for the best paper published in the journal Pattern Recognition in 1998. He received the main annual award from the German Society for Pattern Recognition (DAGM), the 2000 Olympus Prize. He chairs the IS&T/SPIE annual conference series on Vision Geometry. He received a PhD in Computer Science from the Hamburg University in 1992. He published and edited over 100 research papers and books. His main research areas are shape representation and similarity, video analysis and mining, robot mapping, and digital geometry and topology.

Vasileios Megalooikonomou received his BS in Computer Engineering and Informatics from the University of Patras, Greece in 1991, and his MS and PhD in Computer Science from the University of Maryland, Baltimore County, in 1995 and 1997, respectively. He is currently an Associate Professor of Computer and Information Sciences and Director of the Data Engineering Laboratory at Temple University. He received a CAREER award from the National Science Foundation in 2003. His main research interests include data mining, data compression, multimedia database systems, pattern recognition, and biomedical informatics.

Roland Miezianko received his BS Degree in Electrical Engineering from Boston University in 1991 and MA Degree from La Salle University in 1997. He is currently a PhD candidate in the Computer and Information Sciences Department of Temple University, Philadelphia, USA. His main research interests are in motion detection, object tracking and video analysis. As a Founder of Terravic Corporation, he has over 14 years of software engineering experience working with machine vision and image processing projects.

Dragoljub Pokrajac received his BS Degree in Electrical Engineering from the University of Nis, Serbia in 1993, his MS Degree in Telecommunication systems from the University of Nis, Serbia in 1997 and his PhD degree in Computer Science from Temple University, Philadelphia, PA in 2002. He is an Assistant Professor in the Department of Computer and Information Science at Delaware State University. His research interests include databases, machine learning, neural networks, statistics and applied mathematics.

1 Introduction

Automatic detection and tracking of moving objects is a necessary pre-processing step of video analysis. The need is especially strong in the case of large scale video analysis and content-based retrieval where moving objects have to be identified prior to generation of video semantics and performance of high-level video analysis such as similarity retrievals. Detection of moving objects in video is inherently prone to uncertainty. Steady objects may appear to be moving because of changing lighting conditions, movement of camera, noise in video acquisition, etc. An overview of existing approaches to motion detection can be found in a collection of papers edited by Remagnino et al. (2002) and in the special section on video surveillance in IEEE PAMI edited by Collins et al. (2000).

A common feature of existing approaches for moving objects detection is the fact that they are pixel based. Some of the approaches rely on comparison of colour or intensities of pixels in the incoming video frame to a reference image. Jain et al. (1977) 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 colour or grey values over time at each pixel location (e.g., the segmentation by background subtraction in W4 (Haritaoglu et al., 2000), eigenbackground subtraction (Oliver et al., 2000), etc). Wren et al. (1997) were the first that used a statistical model of the background instead of a reference image. Later, Toyama et al. (1999) introduced Wallflower, a system that performs background subtraction and maintains a background model using an appropriate representation of the background and its associated statistics, differentiating background pixels from foreground pixels (that should be processed for identification and tracking of moving objects). One of the most successful approaches for motion detection was introduced by Stauffer and Grimson (2000). It is based on adaptive Gaussian mixture model of the colour values distribution over time at each pixel location. Each Gaussian function in the mixture is defined by its prior probability, mean and covariance matrix.

In this paper, we introduce an approach for video analysis that is based on simple and very efficient dimensionality reduction of videos. We use spatiotemporal blocks as main representation of videos. Each spatiotemporal block represents both texture and motion patterns. To obtain such blocks, we decompose a given 3D video matrix into cuboids. Our results are obtained with cuboids with dimensions $8 \times 8 \times 3$ to be consistent with the 8×8 blocks that MPEG uses for compression. Originally each such block contains 192 colour or grey values, which is reduced to just three values using Principal Components Analysis (PCA). Thus, the obtained video is 64 times smaller in spatial size. Moreover, the obtained block representation handles uncertainty better than the original pixel values, i.e., it is significantly more stable with respect to noise and light changes. An earlier version of the motion detection method was proposed in Pokrajac and Latecki (2003). Here, we have extended the method by replacing the dynamic threshold with dynamic distribution learning and outlier detection significantly improving the performance of the original approach.

The usefulness of dimensionality reduction techniques to compactly represent 3D blocks has already been recognised in video compression. There, 3D discrete cosine and 3D wavelet transforms are employed to reduce the colour or grey level values of a large number of pixels in a given block to a few quantised vector components, e.g., Westwater and Furht (1997). 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) (Jolliffe, 2002) satisfies these requirements. Namely, for a given set of

3D blocks, PCA assigns to each block a vector of the components that maximises the differences among the colocated blocks. Consequently, PCA components are very suitable to detect changes in 3D blocks. As argued in Javed et al. (2002), the application of region level techniques can lead to increased stability when detecting objects in adverse conditions. However, Javed et al. (2002) and related approaches by Buttler et al. (2003), aimed to improve the Stauffer and Grimson algorithm (2000) still perform motion detection on pixel level, and only the post-processing of pixel-based motion detection results is region based.

In the sections that follow, we present a detailed description of the proposed methodology (Section 2). We then describe in detail the datasets used for the evaluation of our proposed approach as well as the experiments performed and the results obtained (Section 3). Finally we present our concluding remarks (Section 4).

2 Methods

We first describe the new representation of video data that use spatiotemporal texture vectors. We then introduce the proposed method for detection of moving objects that is based on the analysis of the distribution of texture vectors.

2.1 Video representation with spatiotemporal texture vectors

We represent videos as three-dimensional (3D) arrays of grey level or monochromatic infrared[1] pixel values $g_{i,j,t}$ at a time instant t and at a pixel location i, j. A video is characterised by temporal dimension Z corresponding to the number of frames and by two spatial dimensions that characterise the number of pixels in horizontal and vertical direction of each frame. Each frame (image) in a video sequence is divided into disjoint $N_{\mathrm{BLOCK}} \times N_{\mathrm{BLOCK}}$ squares (e.g., 8×8 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, 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. Since our goal is to capture change of texture at each spatial location, we used grey level/monochromatic pixel values from each spatiotemporal 3D block as original feature space. Hence, 3D blocks are represented as 192-dimensional vectors of grey level or monochromatic infrared pixel values. We zero mean these vectors and project them to a smaller number of dimensions (e.g., ten dimensions) using principal component analysis (PCA) (Jolliffe, 2002). By performing PCA, we retain the variance of original features (representing spatiotemporal texture at a particular video frame location) while significantly reducing the dimensionality. The obtained low-dimensional vectors therefore form a compact spatiotemporal *texture representation* for each 3D block. The PCA projection matrices are computed separately for each video plane location (a set of disjoint 8×8 squares in our experiments).

Observe that the same matrix is used for all colocated blocks, making possible to maximise the difference among the blocks positioned at the same place of the video plane.

The blocks are represented by N-dimensional vectors $\boldsymbol{b}_{I,J,t}$, specified by spatial indexes (I,J) and time instant t. Vectors $\boldsymbol{b}_{I,J,t}$ contain all values $\boldsymbol{g}_{i,j,t}$ of pixels in the corresponding 3D block. To reduce the dimensionality of $\boldsymbol{b}_{I,J,t}$ while preserving information to the maximal possible extent, we compute a projection of the normalised block vector to a vector of a significantly lower length $K << N$ using a PCA projection matrix $\boldsymbol{P}^{K}_{I,J}$ computed for all $\boldsymbol{b}_{I,J,t}$ at video plane location (I,J). The resulting *sp* texture vectors $\boldsymbol{b}^{*}_{I,J,t} = \boldsymbol{P}^{K}_{I,J} * \boldsymbol{b}_{I,J,t}$ provide a joint representation of texture and motion patterns in videos and are used as input of the algorithms for detection of moving objects (we used $K = 10$ in all of our experiments).

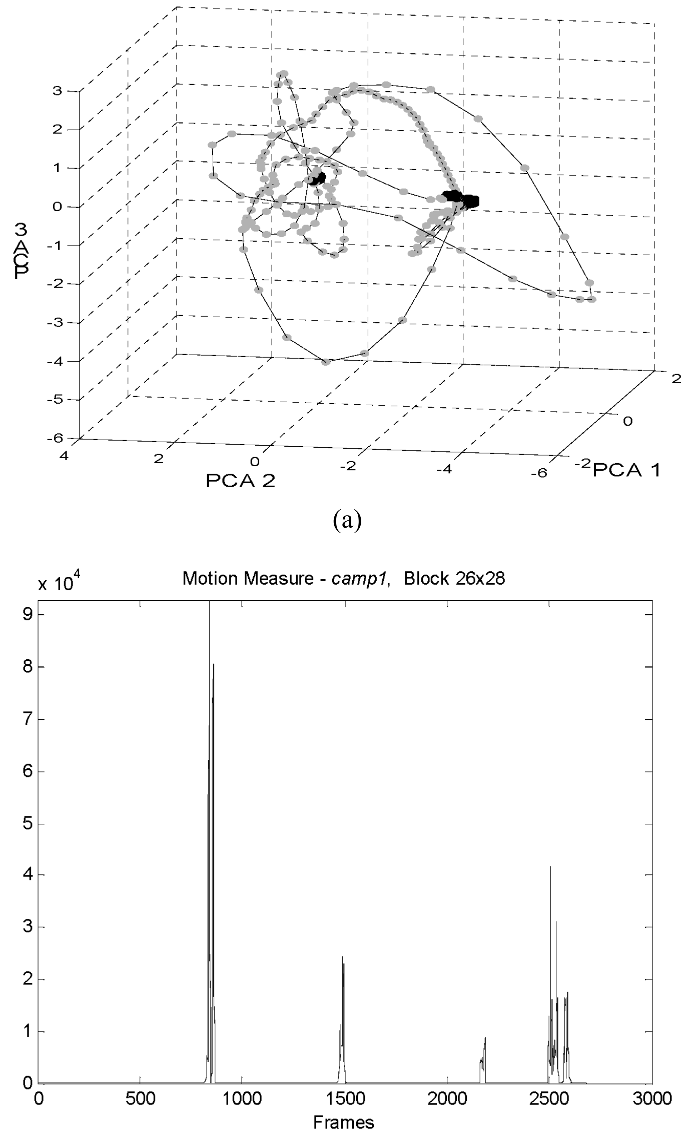
To compute the projection matrix $\boldsymbol{P}^{K}_{I,J}$, we employ the principal values decomposition following Duda et al. (2001) and Flury (1997). A matrix of all normalised block vectors $\boldsymbol{b}_{I,J,t}$ at video plane location (I,J) is used to compute the $N \times N$ dimensional covariance matrix $\boldsymbol{S}_{I,J}$. The PCA projection matrix $\boldsymbol{P}_{I,J}$ for spatial location (I,J) is computed from the $\boldsymbol{S}_{I,J}$ covariance matrix. The projection matrix $\boldsymbol{P}_{I,J}$ of size $N \times N$ represents N principal components. By taking only the principal components that correspond to the K largest eigenvalues, we obtain $\boldsymbol{P}^{K}_{I,J}$.

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 change of the characteristics of *sp* vectors over time, we are able to detect whether a particular block belongs to a moving object or to the 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* 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 Figure 1(a) we show the trajectory of *sp* vectors corresponding to block location (26,28) in *Campus* 1 video. To make this visualisation possible, we use only the first three PCA components for each *sp* vector. It can be observed that frames where 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 a small spread. In contrary, when moving objects are passing through this block location, the trajectory of *sp* vectors (blue-grey dots) is typically elongated and the variance is relatively large.

Figure 1 (a) Orbits of block vectors with blue-grey dots corresponding to the frames where the block (26, 28) of the *Campus* 1 video was identified as moving by the proposed method and (b) the graph of local variance *mm* over time for the same block



(a)

(b)

A simple way to determine the speed of *sp* vectors' 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 a 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 time instant *t*. For these vectors, we compute the covariance matrix $\boldsymbol{C}_{x,y,t}$. We assign the largest eigenvalue of $\boldsymbol{C}_{x,y,t}$, 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*)

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

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 Figure 1(b). The large values (spikes) correspond to time intervals when moving objects were observed at this particular video location.

As the graph shown in Figure 1(b) 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 labelled as moving if the motion measure (*mm*) 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 distribution-learning algorithm is described in detail below.

2.3 Dynamic distribution learning and outlier detection

Consider labelling each video position as moving or stationary (background) based on whether the motion measure *mm* is larger or smaller than a suitably defined threshold. The uncertainty in deciding about motion is handled as follows: 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 Figure 1(b)), 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*(t_0) and variance $\sigma^2(t_0)$ of all values *f*(*t*) in some initial interval $t = 1, \ldots, t_0$. 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 standardised feature value is sufficiently large, i.e., when

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

where C_1 is a constant and $std(f(t)) = \sqrt{\sigma^2(f(t))}$ .Once an outlier is detected at time t_1, value $f(t_1)$ is labelled as an outlier. We update the nominal state at time t, if the standardised feature value drops below a threshold $C_2 < C_1$, i.e.,

$$\frac{f(t) - mean\ (f(t-1))}{std\ (f(t-1))} > C_2,$$

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 equation 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):

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

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

$$std\ (f(t)) = \sqrt{\sigma^2(f(t))}.$$

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 prominent right tail. This assumption clearly applies to the Gaussian distribution, but is significantly more general.

3 Experimental evaluation

3.1 Ground truth dataset

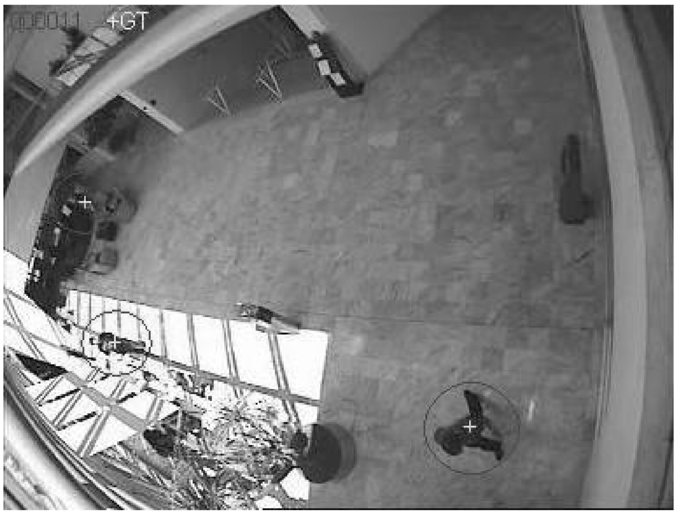
The video clips and corresponding ground truth data used in our evaluation were created by the CAVIAR project team (http://homepages.inf.ed.ac.uk/rbf/CAVIAR/). The video was captured using a wide-angle lens at a resolution of 384 × 288 pixels and 25 fps and then compressed using MPEG2 codec. Each video clip shows different scenarios, such as people walking, meeting, fighting and leaving objects behind. Ground truth data were established for each video sequence by manually labelling each activity and region of interest, and stored in an XML file. The XML file conforms to Computer Vision Markup Language (CVML) format.

The information stored in the ground truth file contains more data than was necessary to perform our evaluation. Therefore, to simplify the computation and data analysis, a new simple text file was created from the extracted XML data. The following individual object data were extracted for each frame: frame number, object id and (x,y) centroid location of the region of interest. An example of extracted data is shown in Table 1, where three distinct objects were identified in frame 11 of the *Split*1 video, as shown in Figure 2.

Table 1 Sample of *Split*1 video ground truth data

Frame	*Object ID*	*cx*	*cy*
10	0	266	239
10	1	65	193
10	2	47	112
11	0	268	241
11	1	65	193
11	2	48	113
12	0	269	244
12	1	65	193
12	2	48	113

Figure 2 Ground truth data centroids for three objects in frame 11 of *Split*1 video



Evaluation of two videos out of 28 available from the CAVIAR project is presented in Section 3.2. The first video is labelled *Walk*1 and is identified as ‘One person walking – straight line’ with ground truth data file wk1gt.xml. The second video is labelled *Split*1 and is identified as ‘Two people meet, walk together and split’ with ground truth data file mws1gt.xml. Each video exhibits slight periodic frame jumps owing to transmission and compression artifacts. Both motion detection methods must be able to distinguish true motion in a noisy video stream. The videos can be viewed on http://knight.cis.temple.edu/~video/VA/.

3.2 Ground truth data evaluation

Ground truth data give us the number of objects and their centroids in each video frame. In order to compare the proposed method and the previous state-of-the-art method to the ground truth data, we must detect motion, find objects from motion data and compute their ROI and centroids. The processing of each video sequence using our method to identify motion on block level and establish the motion/no motion binary image is performed as described in Section 2. An initial comparison of the Stauffer-Grimson Gaussian mixture model (S&G) (Stauffer and Grimson, 2000) with our method has been presented in Pokrajac and Latecki (2004), showing global variation of spatiotemporal blocks and stop-and-hold thresholding algorithm.

The output from motion detection is fed into the object-labelling algorithm to measure the object's region of interest and centroid location. Connected components are used to establish motion regions of interest with a minimum of ten blocks per region. We evaluate motion blocks as 8-connected objects. The connected component general procedure is outlined in Haralick and Shapiro (1992). We compute the region of interest for each labelled component taken from the minimum and maximum block location of the labelled object, and finally compute the block's centroid location.

3.3 Walk1 video evaluation

Ground truth for Walk1 video is shown in Figure 3. All ground truth centroids are projected onto one frame to visualise all motion paths. All motion centroids for objects with more than ten *sp* motion blocks are shown in Figure 4(a). The same is shown for all detected motion blocks in Figure 4(b) using the S&G method. A substantially large number of moving objects with more than ten blocks per object is evident in Figure 4(b), as the noise of the video on the pixel level contributed to the detection of false-positives. Figure 5 presents frame 390 of Walk1 video showing ground truth data and *sp* motion centroids.

Figure 3 Projected GT centroids, *Walk*1 video

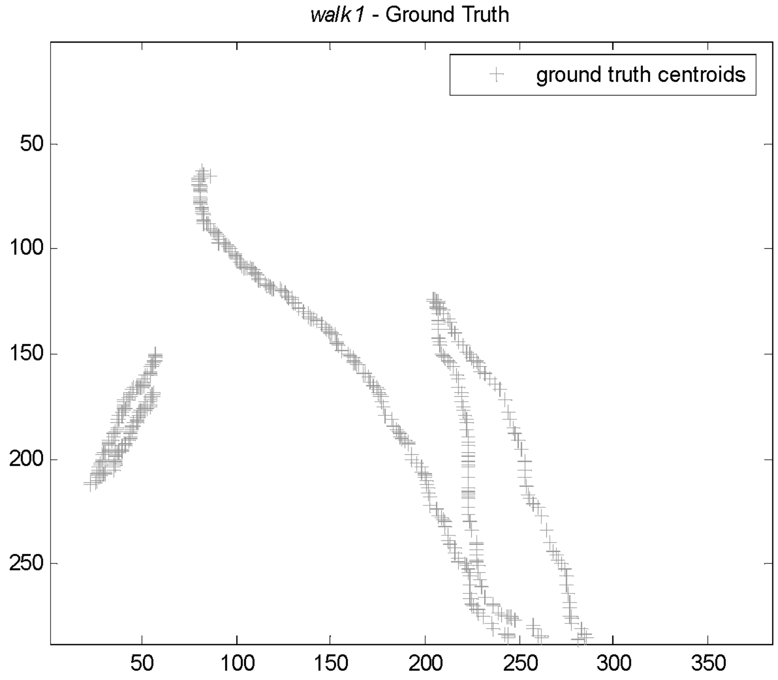
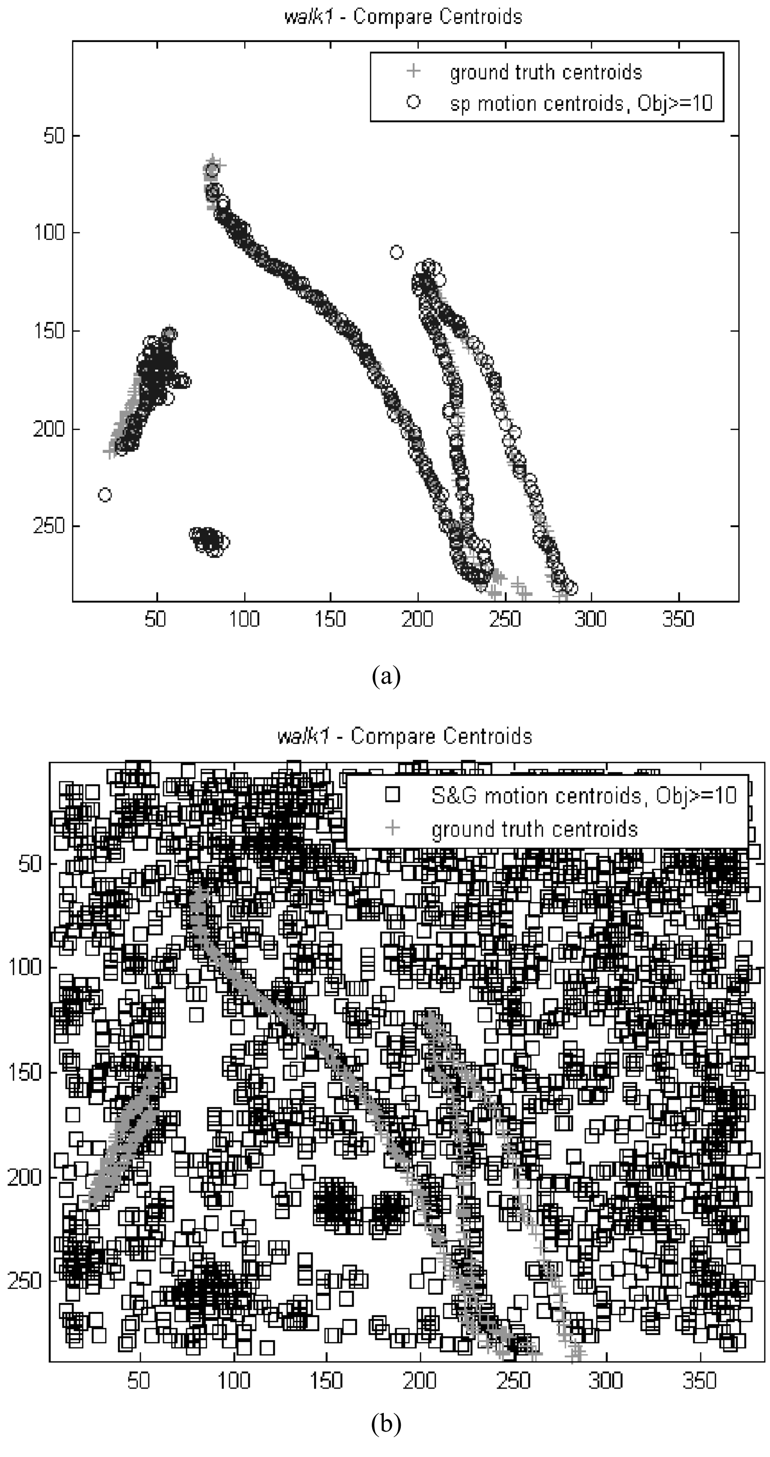


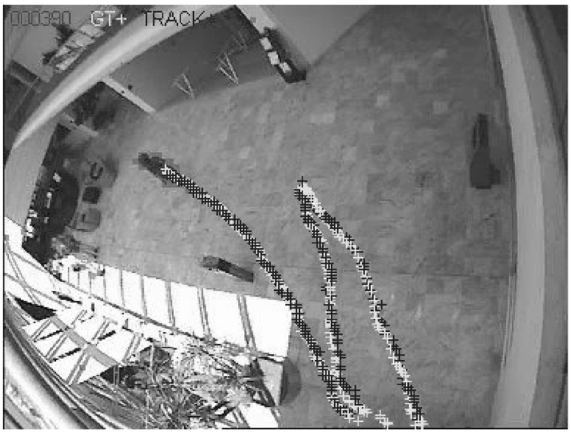
Figure 4 Projection of all centroids for *Walk*1 video; (a) ground truth data and spatiotemporal motion centroids and (b) ground truth data and Stauffer-Grimson Gaussian mixture model centroids



(a)

(b)

Figure 5 Frame 390 of *Walk*1 video showing ground truth data and *sp* motion centroids

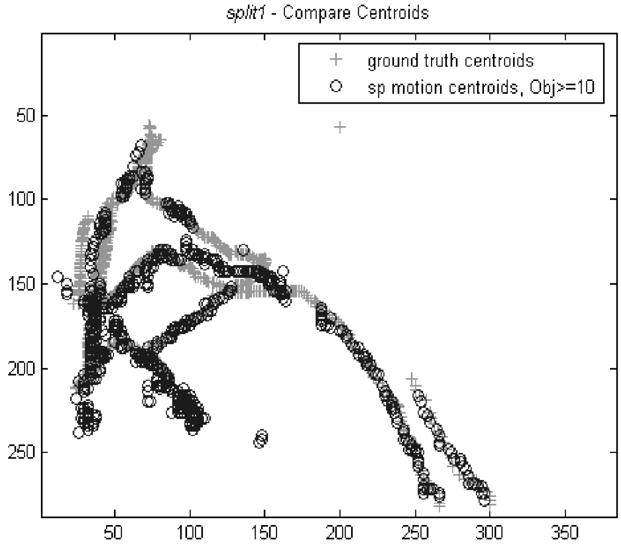


The original S&G requires morphological post-processing. The difficulty is in selecting the size of structuring element, which adds one more parameter. If it is too large, small moving objects disappear, if it is too small, many motion artifacts appear. Our *sp* method does not require any post-processing.

3.4 Split1 video evaluation

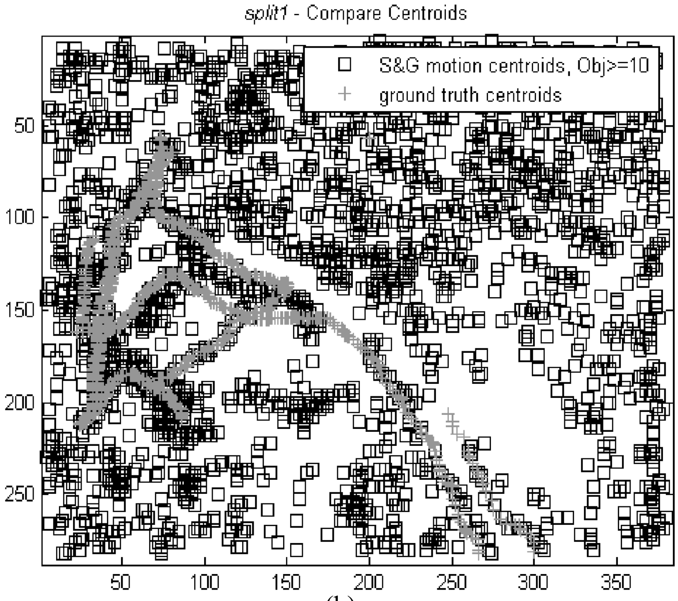
Ground truth for *Split*1 video along with *sp* motion centroids and S&G motion centroids is shown in Figure 6. All centroids are projected onto one frame to visualise all motion paths for objects with at least ten motion blocks.

Figure 6 Projection of all centroids for *Split*1 video; (a) ground truth data and spatiotemporal motion centroids and (b) ground truth data and Stauffer-Grimson Gaussian mixture model centroids



(a)

Figure 6 Projection of all centroids for *Split*1 video; (a) ground truth data and spatiotemporal motion centroids and (b) ground truth data and Stauffer-Grimson Gaussian mixture model centroids (continued)



(b)

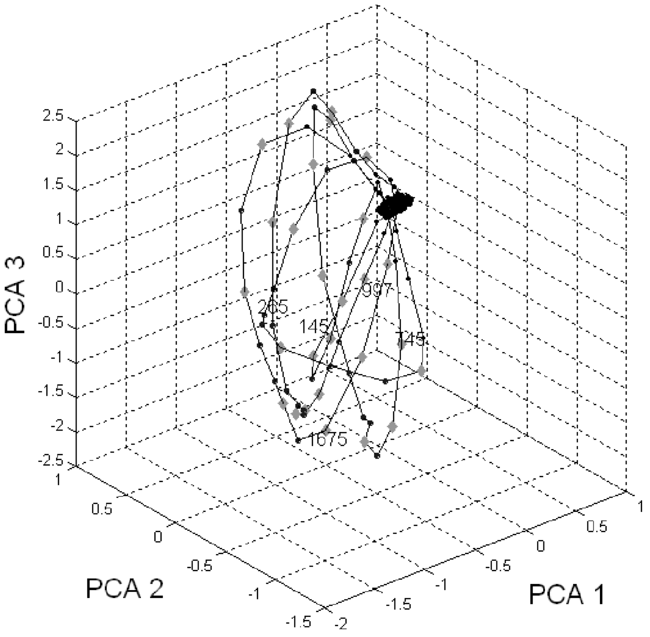
3.5 Motion orbits evaluation

In comparison to any pixel-based approach (e.g., S&G), 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). We demonstrate this on *Campus* 1 video (http://knight.cis.temple.edu/~video/VA/); in Figure 7(b) we plot a trajectory over time of RGB colour values that occur at the pixel (185, 217), which is one of the pixels in the block (26, 28). For better visualisation, in Figure 7(a) we show the linearly transformed space of PCA projections of the original RGB colour values (the trajectory in the space of original RGB colours is similar). Also in Figure 7(b), we superimpose green and blue dots computed by our algorithm for block (26, 28), that correctly correspond to moving objects at this position.

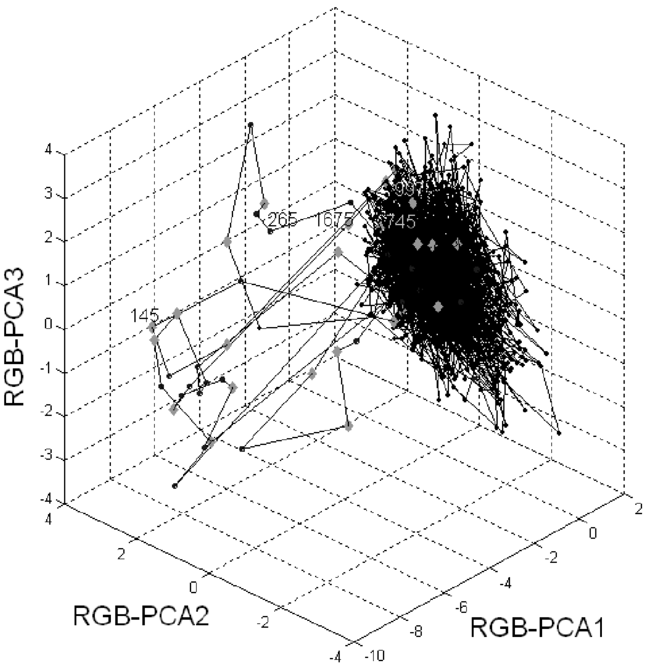
3.6 Visual frame evaluation

The simplest way to evaluate motion and tracking methods is to simply view the video. In Figure 8(c), we see the effects of false-positives when a noisy video is streamed from a live surveillance camera (the original frame is shown in Figure 8(a)). In Figure 8(b), the effect of noise is minimal owing to spatiotemporal block evaluation of motion vectors. Some noise is visible in Figure 8(b), right at the boundaries of light and shadow. The jitters in video create false motion of texture, however this effect is minimal compared with Figure 8(c).

Figure 7 Trajectories at location *I* = 26, *J* = 28 of the outdoor video in feature space of (a) 3-PCA components of block vectors and b) standardised PCA components of RGB pixel coordinates at pixel location (185, 217) (inside block *I* = 26, *J* = 28)

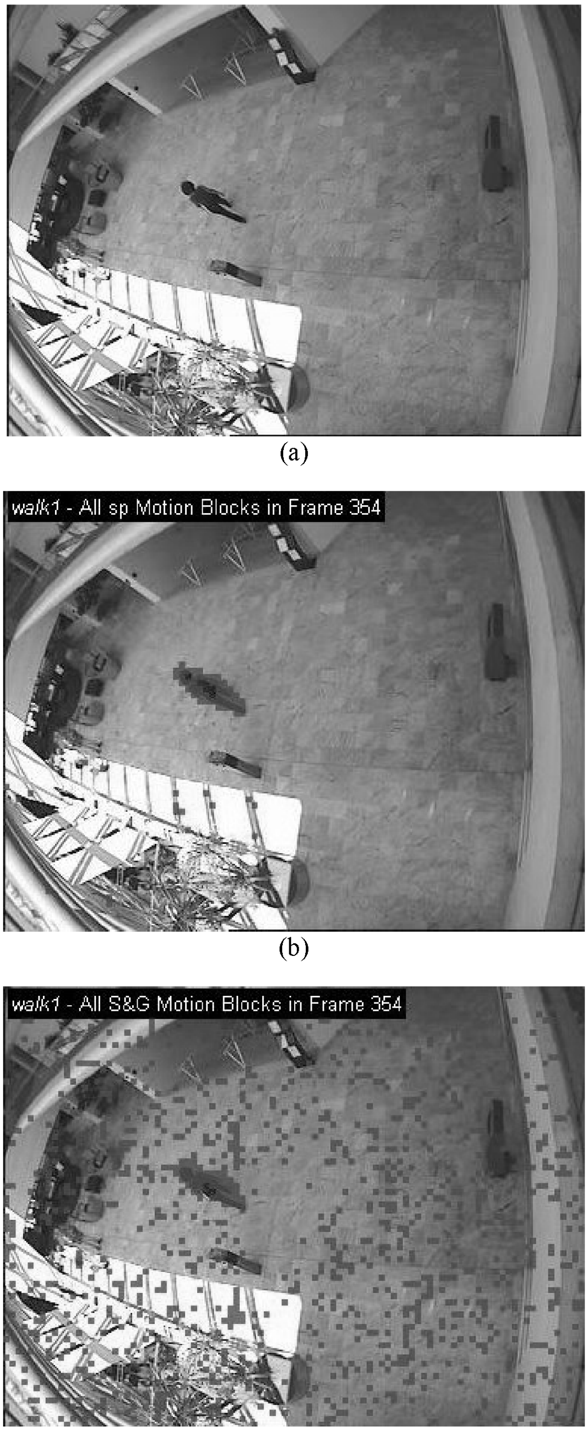


(a)



(b)

Figure 8 (a) *Walk*1 video, frame 354, no motion blocks; (b) spatiotemporal blocks detect motion of central person and (c) S&G method detects motion of central person plus some noise


walk1 - All sp Motion Blocks in Frame 354
walk1 - All S&G Motion Blocks in Frame 354

(a)

(b)

(c)

4 Conclusion

In this paper, we propose a new method for dealing with uncertainty in detecting moving objects in video. We decompose a given video into spatiotemporal blocks and apply a dimensionality reduction technique to obtain a compact representation of colour or grey 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. The power of 3D block representation has already been recognised in video compression, where 3D discrete cosine and 3D wavelet transforms have been developed. We propose to use 3D block vectors as primary input elements to video analysis algorithms moving 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 experiments show that detection and tracking of moving objects is substantially improved if it is based on spatiotemporal blocks instead on pixels. Since each 3D block is represented as a vector of a few real numbers, we significantly improve the performance of video analysis algorithms. We show that the proposed local variation is not only a much simpler but also a more robust model for motion detection for surveillance videos. It can significantly reduce the processing time in comparison to the Gaussian mixture model, owing 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 possible. 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.

Acknowledgements

Ground truth data are provided by the EC Funded CAVIAR project/IST 2001 37540, found at URL: http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.

References

Butler, D., Sridharan, S. and Bove, V.M. (2003) 'Real-time adaptive background segmentation', *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing* (ICASSP), Hong Kong, No. 3, pp.349–353.

Collins, R.T., Lipton, A.J. and Kanade, T. (2000) 'Introduction to the special section on video surveillance', *IEEE PAMI*, Vol. 22, No. 8, pp.745–746.

Duda, R., Hart, P. and Stork, D. (2001) *Pattern Classification*, 2nd ed., John Wiley & Sons, New York.

Flury, B. (1997) *A First Course in Multivariate Statistics*, Springer Verlag, New York.

Haralick, R.M. and Shapiro, L.G. (1992) *Computer and Robot Vision*, Vol. I, Addison-Wesley, Reading, MA.

Haritaoglu, I., Harwood, D. and Davis, L. (2000) 'W4: real-time surveillance of people and their activities', *IEEE PAMI*, Vol. 22, No. 8, pp.809–830.

Jain, R., Militzer, D. and Nagel, H. (1977) 'Separating nonstationary from stationary scene components in a sequence of real world TV images', *Proc. IJCAI*, Cambridge, MA, pp.612–618.

Javed, O., Shafique, K. and Shah, M.A. (2002) 'Hierarchical approach to robust background subtraction using colour and gradient information', *Proc. IEEE Workshop MOTION*, Orlando, pp.22–27.

Jolliffe, I.T. (2002) *Principal Component Analysis*, 2nd ed., Springer Verlag, New York.

Oliver, N.M., Rosario, B. and Pentland, A.P. (2000) 'A Bayesian computer vision system for modeling human interactions', *IEEE PAMI*, Vol. 22, No. 8, pp.831–843.

Pokrajac, D. and Latecki, L.J. (2003) 'Spatiotemporal blocks-based moving objects identification and tracking', *IEEE Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, Nice, France, pp.70–77.

Pokrajac, D. and Latecki, L.J. (2004) 'Entropy-based approach for detecting feature reliability', Invited Paper, *48th Conf. for Electronics, Telecommunications, Computers, Automation, and Nuclear Engineering (ETRAN)*, Cacak, Serbia.

Remagnino, P., Jones, G.A., Paragios, N. and Regazzoni, C.S. (Eds.) (2002) *Video-Based Surveillance Systems*, Kluwer Academic Publishers, Boston.

Stauffer, C. and Grimson, W.E.L. (2000) 'Learning patterns of activity using real-time tracking', *IEEE PAMI*, Vol. 22, No. 8, pp.747–757.

Toyama, K., Krumm, J., Brumitt, B. and Meyers, B. (1999) 'Wallflower: principles and practice of background maintenance', *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Corfu, Greece, Vol. 1, Nos. 20–27, pp.255–261.

Westwater, R. and Furht, B. (1997) *Real-Time Video Compression: Techniques and Algorithms*, Kluwer Academic Publishers, Boston.

Wren, C., Azarbayejani, A., Darrell, T. and Pentland, A.P. (1997) 'Pfinder: real-time tracking of the human body', *IEEE PAMI*, Vol. 19, No. 7, pp.780–785.

Note

[1] Monochromatic IR cameras generate pixel values as a single function of IR radiation picked up by the camera.