A PCA-Based Technique to Detect Moving Objects

Nicolas Verbeke and Nicole Vincent

Laboratoire CRIP5-SIP, Université René Descartes Paris 5, 45 rue des Saints-Pères, 75270 Paris Cedex 06, France
{nicolas.verbeke,nicole.vincent}@math-info.univ-paris5.fr
http://www.sip-crip5.org

Abstract. Moving objects detection is a crucial step for video surveillance systems. The segmentation performed by motion detection algorithms is often noisy, which makes it hard to distinguish between relevant motion and noise motion. This article describes a new approach to make such a distinction using principal component analysis (PCA), a technique not commonly used in this domain. We consider a ten-frame subsequence, where each frame is associated with one dimension of the feature space, and we apply PCA to map data in a lower-dimensional space where points picturing coherent motion are close to each other. Frames are then split into blocks that we project in this new space. Inertia ellipsoids of the projected blocks allow us to qualify the motion occurring within the blocks. The results obtained are encouraging since we get very few false positives and a satisfying number of connected components in comparison to other tested algorithms.

Keywords: Data analysis, motion detection, principal component analysis, video sequence analysis, video surveillance.

1 Introduction

Digital video cameras available on the market are less and less expensive and more and more compact. At the same time, nowadays the computation power of computers enables us to consider real-time processing of video sequences in a serious way. That is why industrialists now tend to choose vision-based solutions to solve problems, for which, a few years back, they would have chosen another type of solution, such as human surveillance or more mechanical sensors. Video sequences thus obtained are three-dimensional data (two spatial dimensions and one temporal dimension) and may be considered as 2D+T volumes. Various issues are encountered, but the first task of a video sequence analysis system is always motion detection, and if possible, moving objects detection (segmentation). This task can be more or less difficult depending on the light conditions and the expected processing speed and accuracy. A detailed list of problems related to light conditions and to the scene content can be found in [1]. In this paper we will address the case of a static video camera.

B.K. Ersbøll and K.S. Pedersen (Eds.): SCIA 2007, LNCS 4522, pp. 641–650, 2007.

Most motion detection algorithms in the literature are described as background subtraction methods. A survey of recent advances can be found in [2]. The process is to build a background model, then to apply a decision function over each new frame in order to label each point as background or foreground. In other words, video data is not considered as a 2D+T volume but as a series of two-dimensional image pairs.

When the video is a set of frames indexed by time, the simplest background model is to consider frame $t-1$ as the background in frame t and to classify every pixel that changed significantly between t and $t-1$ as belonging to the foreground. In other words, motion detection is achieved by temporal derivation of the sequence. This derivative can be computed very quickly, but it is also very unstable because of its sensitiveness to any kind of noise. In addition, only short term past is considered so that slow or arrhythmic motions are ill-detected. Thus, temporal derivative is almost always post-processed, as in [3] where it is regularized with respect to the optical flow history at each point.

Instead of using frame $t-1$ as a background model, one may use a "reference image". Unfortunately such an image is not always available, and if it is, it becomes quickly out-of-date, especially in outdoor environment. That is why the authors who use this technique always offer a function to update the reference image. This method is called "background maintenance", as in [4] where the reference image is continuously updated when temporal derivative is negligible.

The background model is often a statistical model built to estimate the probability of a gray value or a color at a given point. If this probability is high the pixel is considered as belonging to the background, otherwise it must belong to an object. Sometimes the model is the parameter set of a distribution from a known family, as in [5] where the color values are modeled as Gaussian distributions. In other cases, no prior assumption is made about the distribution to be estimated, and a non-parametric estimation is achieved, as in [6]. The background model is then a set of past measurements used to punctually estimate probability densities thanks to a Parzen window or a kernel function. Background subtraction may also be viewed as a prediction problem. The most commonly used technique is Wiener [1] or Kalman [7] filtering.

Thus, most methods aim at estimating a background model to detect moving objects. Nonetheless, other works can be mentioned such as [8] where moving areas are those where spatiotemporal entropy of the sequence reaches a maximum. Unlike foregoing techniques, temporal dimension is fully used by a local analysis algorithm through the 2D+T video volume. In [9], this approach is slightly modified to compute the temporal derivative's entropy instead of the input sequence's entropy, in order to prevent the detection of spatial edges as moving areas. Our study lies in the same category: we aim at detecting moving object by analyzing the video volume with no explicit background model. We will first introduce the chosen feature space and then we will explain the criteria used to select moving objects. At last we will analyze the results of our experiments and evaluate them.

2 Feature Space

Initially we consider video data represented by a function defined in a three-dimensional space: two spatial dimensions (x, y) and a temporal one (t). With each point of this space is associated a gray value at point (x, y) and time t. So the semantic entities (background, moving objects) are subsets of points from this space. In order to identify them, we have to aggregate them into classes with respect to shared features. In such a space, the amount of points to consider is huge, that is why the background model-based approach is so commonly used: the only points to consider are those from current frame, while the background model is supposed to sum up all past observations. We would prefer to keep a knowledge of the past less synthetic because relevant information we have to extract is not always the same. Thus we need a feature space that fits better to the sequence itself, rather than to each frame, and that enables us to consider motion without modifying input data. With each point (x, y) in the image space, is associated a vector containing gray values at location (x, y) along the considered time interval. Moreover, in the prospect of using data analysis techniques, the sequence is not regarded as a function anymore but as a set of individuals, which are the pixels we observe when we watch the sequence. During this step, spatial relationships between pixels are therefore ignored. To avoid doing a fine analysis, we do not track objects anymore but we focus on a fixed location within the image. We will keep the evolution of gray values for each pixel along time. A dozen values may be kept (let p be that number), and each pixel becomes an individual characterized by a set of parameters. Individuals have p coordinates. As our algorithm processes p frames at a time, we can allow the computation to be p times slower than if we would have processed each frame individually; so, we can use more time-consuming techniques. Nevertheless, for the process to be fast enough, we have to reduce the amount of information. There are many dimension reduction techniques, such as principal component analysis (PCA), factor analysis (FA), the whole family of independent component analysis (ICA) methods, or neural network-based algorithms as Kohonen self-organizing maps. For an extended survey of dimension reduction techniques, one may refer to [10]. As PCA is known to be the best linear dimension reduction technique in terms of mean squared error, we chose this method to avoid losing information that best discriminates points.

PCA was developed in the early 19th century to analyze data from human science. It is a statistical technique to simplify a dataset representation by expressing it in a new coordinate system so that the greatest variance comes to lie on the first axis. Thus we can reduce the search space dimensionality by keeping the first few coordinates in the new frame of reference. A basis of this space consists of the eigenvectors of the covariance matrix of the data sorted in the decreasing order of the corresponding eigenvalues magnitude.

The coordinates of background pixels are likely to be more or less equal to each other, while the coordinates of moving objects pixels should vary. We want to detect this variation, it is therefore interesting to find the axis, that is, the good basis in the p-dimensional space where the factor's variance is maximum.

In the case of a video sequence, the data matrix $\mathbf{X}$ contains all the features of the points (x, y, t) to consider. From now on, let n be the number of rows of $\mathbf{X}$, that is the number of pixels in the image, and let p be its number of columns, that is the number of features associated with each pixel. The two first coordinates (x, y) can get a finite number of values; let $\mathcal{D}_P$ be the pixel domain. On the other hand, the time domain is assumed to be infinite. So we have to choose a range that should contain all relevant information. We decide to use $\mathcal{D}_t = \{t - \Delta t, \ldots, t\}$ as time domain, where t is current time. Each row of $\mathbf{X}$ is a data element corresponding to a pixel $(x, y) \in \mathcal{D}_P$, and each column of $\mathbf{X}$ is an observed variable, that is a gray value at time $\tau \in \mathcal{D}_t$. The new basis of the feature space is then associated with the eigenvectors of the data covariance matrix $\mathbf{C}$.

$$\mathbf{C} = \bar{\mathbf{X}}^T \cdot \mathbf{D}_p \cdot \bar{\mathbf{X}}, \tag{1}$$

where $\bar{\mathbf{X}}$ is the centered data matrix, and $\mathbf{D}_p = \frac{1}{p}\mathbf{I}_p$ ($\mathbf{I}_p$ being the p-order identity matrix.)

The method must be as invariant as possible towards the various acquisition conditions, therefore we will focus on gray variations rather than on gray values. Thus we can remove one dimension from the feature space by filling $\mathbf{X}$ with the temporal derivative rather than with the raw grayscale images. We have then a $(p-1)$-dimensional space. Let $\mathbf{Y}$ be the data matrix in this space.

Let us consider a 10-frame sub-sequence with 288 rows and 720 columns. Figure 1(a) pictures the first frame from this sub-sequence. Matrix $\mathbf{X}$ has therefore 288×720 rows and 10 columns, while $\mathbf{Y}$, over which we will perform PCA, has 288×720 rows and 9 columns. Figure 1 pictures the nine projections of $\mathbf{Y}$ on the principal axes given by the PCA algorithm. More precisely, we consider the image domain and we build an image where gray values are proportional to the values of the feature vectors projected on a given principal axis.

According to Fig. 1, moving areas are clearly shown when $\mathbf{Y}$ is projected on the two first principal axes. The difference between a static area and a moving area is emphasized on these axes. This observation is confirmed by the histogram of variance explained by the factors (Fig. 2). The variance explained by an axis is defined as the ratio between the eigenvalue associated with this axis and the sum of all eigenvalues of the covariance matrix.

Thus, if we choose to keep only the two first principal axes, 20% of the initial amount of data is enough to preserve 80% of the observed variance. This first experimentation confirms our approach, which remains very global. In next section, we will use this feature space. The process is then to compute a PCA for every set of ten successive frames. To achieve greater consistency, we will now consider a more local approach relying on this first global study.

3 Detection of Coherent Moving Areas

The data representation as shown on Fig. 1(b) enables to detect local motion (pixel motion) more easily. Indeed, selecting the pixels whose absolute

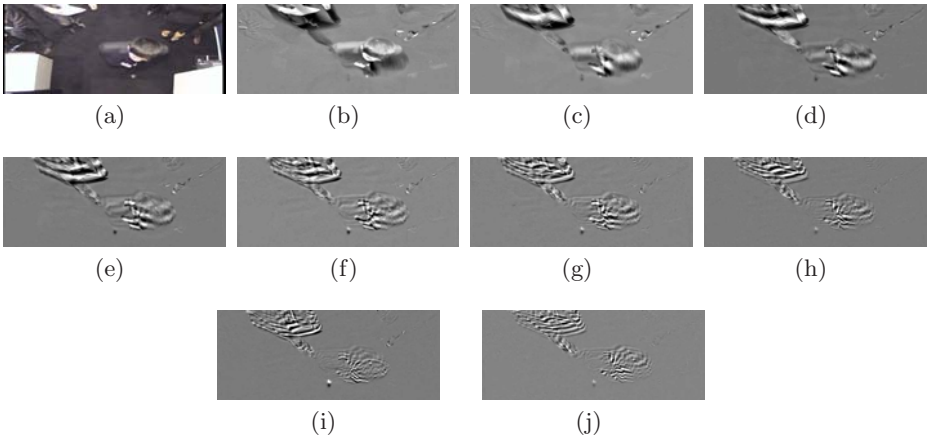

Fig. 1. (a) Input sequence. (b)—(j) Projections of $\mathbf{Y}$ on each of the nine principal axes outlined by PCA. Subfigures are ordered with respect to the rank of the associated factor.

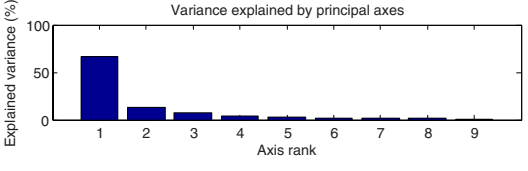

Fig. 2. Variance explained by the principal axes

Fig. 3. (a) Projection of $\mathbf{Y}$ on the first principal axis, (b) background segmentation achieved from (a), (c) segmentation improved by morphological operations

value is high (the darkest and the lightest ones) is enough to get a moving objects/background segmentation. Figure 3(b) shows the automatic segmentation of the projection of $\mathbf{Y}$ on the first principal axis.

We find ourselves in the same situation as most methods present in the literature; such connected components of a binary picture would be labeled to obtain a moving object detection. Like in most cases, in Fig. 3, image post-processing would be necessary to remove the false positives and restore the connectivity of the objects (Fig. 3(c)). Such an approach provides an accurate segmentation, but choosing the morphological operations to carry out is often a delicate job. A mistake in choosing a structuring element could erase an important object, connect two different objects, validate a false positive, etc. A learning phase is

necessary to adapt the general method to the particular case of the sequence studied. The need of post-processing is due to the global aspect of the method all over the image. Therefore, we prefer to avoid having to carry out such a step, but we still need to define the connected areas associated with a unique moving object. To gain coherence we have to lose in accuracy. To achieve this we are going to introduce a more local approach that nevertheless is based on the results of the previous study. From the global representation studied above, sub-populations will be isolated and compared in the pixel population. That is why we will start to split the data ($\mathbf{Y}$) into many subsets. Each subset is associated with a $b \times b$ pixel block defined by the nine values of the data matrix $\mathbf{Y}$ which constitute the values of the factors revealed in the global study. On the initial video volume three-dimensional blocks of size $b \times b \times 10$ are thus studied through 9 new features. To get more continuous results without increasing the computation time too much, we chose blocks that overlay in half along the space dimensions.

The subsets thus obtained are represented in a $(p-1)$-dimensional space. We will study the relative locations of those subsets. To simplify the computation, we will represent each subset by its inertia ellipsoid. Projections of the ellipsoids will be compared in the plane formed by the first two factors of the global representation.

4 Comparison of the Detected Areas

Figure 4 shows a set of inertia ellipsoids projected on the first factorial plane of the global image. Each corresponds to a spatiotemporal block as described in Sect. 3.

The observed ellipses differ by way of their location in the plane, their area and their orientation. As far as this study is concerned, we will not focus on the orientation of the ellipses. The data being centered, the frame of Fig. 4 has for origin the mean of $\mathbf{Y}$ (or more exactly the projection of the mean.) As a result, an ellipse that is far from the origin represents a block of which many points are in motion. The area of the ellipses gives a hint about the variability of the points in the block it represents. Thus, we can distinguish four cases:

1. A small ellipse close to the origin represents a block in which no motion occurs.
2. A large ellipse close to the origin represents a block in which the different points have dissimilar motions, but in which the mean of the motions is close to zero. In other words, we can speak of noise.
3. A small ellipse far from the origin represents a block in which the mean motion is important, and whose points have almost the same motion. They are blocks fully included in a moving object.
4. A large ellipse far from the origin represents a block in which the mean motion is important and whose points show various motions. They are blocks that could be found for example on the edge of a moving object.

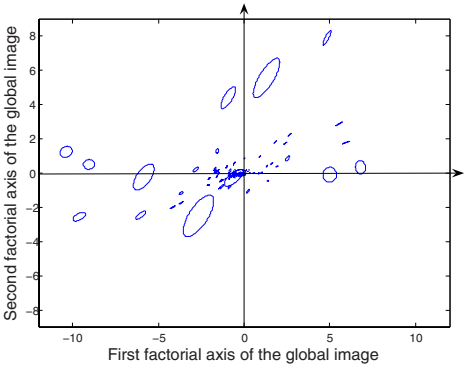

Fig. 4. Each three-dimensional block is modeled by the inertia ellipsoid of the points which constitute it, and each ellipsoid is projected on the plane formed by the two first factors from the global study (PCA)

To detect moving objects, the most interesting blocks are those corresponding to cases 3 and 4, in other words the ellipses far from the origin. Therefore, it is necessary to threshold with respect to the distance from the origin to the center of the ellipses, that is to say the mean or the sum of the points belonging to the corresponding blocks.

5 Results

The size of spatiotemporal blocks introduced in section 3 is still to be discussed. The edges of the detected moving objects could be not accurate enough if the blocks are too large, while blocks too small would imply greater computation time, and impair the objects connectivity. Figure 5 pictures the results obtained for the same ten-frame sequence, with blocks of size $b \times b \times 9$, where b equals successively 16, 32, 64 and 128. Besides we measured the computation time necessary to get these results, as well as their accuracy. Accuracy is given by the number of false positives and false negatives observed when the result is compared to an ideal segmentation. These measurements are put down in Table 1.

We notice that the computation time does not depend a lot on the blocks size. As a consequence it can be chosen depending only on the sequence to be analyzed, without worrying about computation time. In this case, the image size is 720×288 pixels and the blocks size providing the best results is 32×32.

To evaluate our algorithm, we use five video sequences which differ in the issue raised by the application and/or the intrinsic difficulties of the sequence. The first sequence represents a people counting application, where most people stand still for a long time in the field of vision before passing the monitored gate. It is then necessary that the algorithm does not detect insignificant motion. The second sequence is another people counting application, where people tend to move in connected groups. Therefore, the algorithm has to be accurate enough to

(a) (b) (c) (d) (e)

Fig. 5. Results obtained from one sequence by only changing the spatiotemporal blocks size. (a) Input sequence. (b) $b = 16$. (c) $b = 32$. (d) $b = 64$. (e) $b = 128$.

Table 1. Algorithm performances achieved with different block sizes

	$b = 16$	$b = 32$	$b = 64$	$b = 128$
Computation time (%)	100	97.4	87.8	63.9
False positives (pixels)	975	3375	10,372	28,445
False negatives (pixels)	10,851	6874	2149	506

discern the different members of each group. The third one is a vehicle monitoring application. The images are very noisy, due to the sun passing through the trees on the left side of the picture and the vehicles moving in the background. The two last sequences are classical benchmark sequences used in numerous articles[1]. They are used to facilitate the comparison of our results with other methods.

In Fig. 6 we can see the foreground motion detections achieved on those five video sequences thanks to four different algorithms. Row 2 shows the results obtained with the algorithm presented in this article; row 3 is the smoothed temporal derivative of the sequence (with a fixed smoothing coefficient); row 4 shows a non-parametric background subtraction [6]; row 5 is the difference-based spatiotemporal entropy of the sequence [9]. As methods 3 to 5 usually require a post-processing step before labeling connected components, we applied a morphological closing by a 5-pixel-diameter disk followed by an opening by the same structuring element to obtain results shown in rows 3 to 5.

With the smoothed temporal derivation method, the smoothing coefficient is a critical issue and has to be carefully chosen depending on the sequence to be analyzed. A coefficient too high produces a "ghost effect" (row 3, column 1), while a coefficient too low tends to detect only the edges and ignore the inside (row 3, columns 4 and 5).

The non-parametric background modeling algorithm (row 4) detects accurately the edges of the moving objects. Still, this method is very noise-sensitive and unless the post-processing is chosen very specifically contingent on the sequence, the results obtained do not constitute a good segmentation of moving objects.

Our method (row 2), as well as spatiotemporal entropy (row 5), both sacrifice the edges' accuracy for greater robustness. However, there is more noise with entropy than with the method presented here.

[1] They come from the EC funded CAVIAR project/IST 2001 37540, found at URL: `http://homepages.inf.ed.ac.uk/rbf/CAVIAR/`

Fig. 6. Results obtained on 5 sequences with 4 algorithms: (1) Input sequence, (2) our algorithm, (3) smoothed temporal derivative, (4) non-parametric modeling, (5) difference-based spatiotemporal entropy

6 Conclusion

In this paper, we have presented a new coherent motion detection technique in a video sequence. Unlike most of the methods present in the literature, we do not aim at modeling the background of the scene to detect objects, but rather to select significant information and to express it in a lower-dimensional space, in which classifying motion areas and still areas is easier. To get this space, we carry out a principal components analysis on the input data, and we only keep the first two principal factors. Then the sequence is split into spatiotemporal blocks which are classified with respect to the location of their respective inertia ellipse in the first factorial plane. The results are satisfactory if we consider that the number of connected components matches the expected number of objects. However, the edges of the objects are less accurately detected than with statistical background modeling algorithms. Nevertheless, in the context of an industrial use of the method, edge accuracy is not a capital issue. It is far more important to know

precisely the number of objects present in the scene as well as their approximate location and area. If very accurate edges are required, one may use an active contour [11] initialized on the contour given by our algorithm. Besides, in order to take better advantage of the input data, we plan to study the way we could use the color information in our data model.

References

1. Toyama, K., Krumm, J., Brummit, B., Meyers, B.: Wallflower: Principles and practice of background maintenance. In: Proc. IEEE Int. Conf. on Computer Vision (ICCV'99). Kerkyra, Corfu, Greece, vol. 1, pp. 255–261 (1999)
2. Moeslund, T.B., Hilton, A., Krüger, V.: A survey of advances in vision-based human motion capture and analysis. Computer Vision and Image Understanding 104(2), 90–126 (2006)
3. Tian, Y.L., Hampapur, A.: Robust salient motion detection with complex background for real-time video surveillance. In: IEEE Workshop on Motion and Video Computing. Breckenridge, CO, vol. II, pp. 30–35 (2005)
4. Yang, T., Li, S.Z., Pan, Q., Li, J.: Real-time and accurate segmentation of moving objects in dynamic scene. In: Proc. ACM 2nd Int. Workshop on Video Surveillance & Sensor Networks (VSSN 2004), New York, NY pp. 136–143 (2004)
5. McKenna, S.J., Jabri, S., Duric, Z., Wechsler, H., Rosenfeld, A.: Tracking goups of people. Computer Vision and Image Understanding 80(1), 42–56 (2000)
6. Elgammal, A.M., Harwood, D., Davis, L.S.: Non-parametric model for background subtraction. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1842, pp. 751–767. Springer, Heidelberg (2000)
7. Koller, D., Weber, J., Malik, J.: Robust multiple car tracking with occlusion reasoning. Technical Report UCB/CSD-93-780, University of California at Berkeley, EECS Department, Berkeley, CA (1993)
8. Ma, Y.F., Zhang, H.J.: Detecting motion object by spatio-temporal entropy. In: Proc. IEEE Int. Conf. on Multimedia and Expo (ICME 2001), Tokyo, Japan pp. 265–268 (2001)
9. Guo, J., Chng, E.S., Rajan, D.: Foreground motion detection by difference-based spatial temporal entropy image. In: Proc. IEEE Region 10 Conf. (TenCon 2004), Chiang Mai, Thailand pp. 379–382 (2004)
10. Fodor, I.K.: A survey of dimension reduction techniques. Report UCRL-ID-148494, Lawrence Livermore National Laboratory, Livermore, CA (2002)
11. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International Journal of Computer Vision 1(4), 321–331 (1988)