In *2nd Workshop on Statistical Methods in Video Processing (SMVP)*, European Conference on Computer Vision, Prague, 2004.

Novelty Detection in Image Sequences with Dynamic Background

Fredrik Kahl[1], Richard Hartley[1,2] and Volker Hilsenstein[3]

[1] Dept. of Systems Engineering, Australian National University, Australia
[2] National ICT Australia
[3] Institute for Environmental Physics, University of Heidelberg, Germany

Abstract. We propose a new scheme for novelty detection in image sequences capable of handling non-stationary background scenarious, such as waving trees, rain and snow. Novelty detection is the problem of classifying new observations from previous samples, as either novel or belonging to the background class. An adaptive background model, based on a linear PCA model in combination with local, spatial transformations, allows us to robustly model a variety of appearences. An incremental PCA algorithm is used, resulting in a fast and efficient detection algorithm. The system has been successfully applied to a number of different (outdoor) scenarious and compared to other approaches.

1 Introduction

Given a stationary camera taking images of an outdoor scene, the problem considered in this paper is to detect novel events in the sequence of images. A *novel* event can loosely be defined as an event that cannot be inferred from previous images. Many times, it comes down to separating foreground objects from the (possibly changing) background.

The detection of novel events is often the first stage in many visual surveillance systems. Typically, background subtraction is a method employed to alert when new events occur in the sequence. The information provided by this low-level system is an important cue and it can be used for more high-level tasks, such as motion tracking, recognition of events, etc. See [2] for a recent review of the current state-of-the-art in video surveillance.

The requirements for such a novelty detection system are that it must work in (near) real-time, the background should be adaptive and be able to deal with illumination changes, and preferably work with both grey-scale and colour imagery. A common assumption is that the background is static, which makes

things simple on one hand, but limits the applicability on the other hand. In this paper we try to relax this assumption and to cope with non-stationary data in the background by introducing a more flexible background model. It is a linear PCA model, which makes it possible to model different appearances. In order to allow for local deformations, the image plane is partitioned into a set of (possibly overlapping) regions which may move locally around its origin. The model is adaptive, and to speed up computations an incremental PCA algorithm is developed based an algorithm called PowerFactorization ([4]).

1.1 Related Work

There is a huge literature on background subtraction/modelling for static (or slowly changing) background scenes. For example, recursive updates of the background model using an adaptive filter was used in [11] and a Kalman filter in [8]. In [7], a statistical background model was presented focusing on handling illumination changes. A breakthrough in background modelling was made by Stauffer and Grimson in [10] who used a mixture of Gaussians to model multiple hypotheses. To reduce the complexity of the model, each pixel was treated independently. A similar approach was taken by Elgammal et al. [3], but instead kernel methods were applied to obtain non-parametric estimates of the probability distributions of the pixels. Again, no interdependencies between pixels were assumed. Even though these approaches have proven to be successful for many scenarios, they do not handle very much dynamic motion in the background.

In a recent paper by Monnet et al. [5], a background model that explicitly tries to model dynamic textures is presented. The dynamic model is based on the work in [9]. Our model has many resemblances with their model as they employ linear PCA as well. However, they pursue another line of thought. The coefficients of the linear model are fed into an autoregressive dynamical system and then they use the system as a prediction/detection mechanism for novel behaviour. Their model is adaptively updated, but it is unclear how they handle older model coefficients. When the PCA basis is updated, these coefficients become obsolete.

2 Novelty Detection

Before delving into our approach to the problem, we will give a more formal problem formulation. Novelty detection is concerned with first, estimating the unknown parameters (or latent variables) of a statistical model from a set of observations and then, for subsequent observations, deciding whether they should be regarded as novel or not. In a statistical setting, this can be formulated as follows. Given a set of observations $\{x_1, \ldots, x_t\}$, we want to estimate the latent variables θ for some probability distribution $p(x|\theta)$, for example, by applying the Maximum Likelihood method. Then, a new observation x_{t+1} can be considered as *novel* if $p(x_{t+1}) < p_0$ for some threshold p_0.

In our setting, the observables are images $\{I_i\}_{i=1}^{t}$, where $I_i : \Omega \to \mathbf{R}$ and Ω denotes the image plane . Suppose we have a model $\mathcal{M}$ describing this set (or

some representation of it), then novel detection becomes:

$$\text{Is } I_{t+1} \in \mathcal{M} \ ?$$

Or, in words, is I_{t+1} close to $\mathcal{M}$ in some suitable metric?

In addition, we want our statistical model to evolve over time, that is, to adapt to new data, such that only sudden changes are detected. Hence, given $\{I_i\}_{i=1}^{t+1}$ and the current model $\mathcal{M}_t$, it should be easy to infer an updated model $\mathcal{M}_{t+1}$. The influence of older observations should generally be lower than for new ones - the system has only limited memory.

If the resulting novelty detection scheme is to be practically useful, it needs to be able to process the data in real-time. Hence, there is a compromise between the level of complexity of the model and the computational speed. Therefore, it is essential to consider solutions that are able to update the model parameters incrementally.

3 Background Model

This section introduces the statistical background model $\mathcal{M}$ for our novelty detection scheme. Again, remember that the set of observations consists of 2D images of a scene and often the background is non-stationary. For example, trees sway or it could be raining. A sequence of images contains a lot of data, and hence the complexity of the model and the methods have to be restricted.

Linear models are a good compromise between low complexity and reasonable approximation of the observed data – witness the success of Active Appearance Models [1]. Suppose an image I_t at time t can be modelled by an affine function plus a noise term:

$$I_t = I_{mean} + \sum_{i=1}^{r} \lambda_{it} \Phi_i + \epsilon, \tag{1}$$

where I_{mean} can be thought of as the "mean image", the Φ_i are some vectors describing modes of possible variation, and the λ_{it} are some time varying scalars. The noise ϵ is supposed to be zero-mean and normally distributed[4].

The above model for $\mathcal{M}$ maintains interdependencies between pixels in a natural way, and it is capable of modelling a variety of different (global) appearences [1]. However, it does not account for local perturbations well (such as swaying trees) if only a few modes of variation are incorporated into the model.

In order to achieve more locality, we partition the image plane Ω into a set of (possibly overlapping) regions Ω_i such that $\Omega = \cup_i \Omega_i$. Then, we can apply the model in (1) to each region independently. In addition, we will allow for small spatial transformations of each region. Let $T : \mathbf{R}^2 \to \mathbf{R}^2$ denote a spatial transformation, then an image region Ω_i can be modelled by

$$I_t(\mathbf{x}) = I_{mean}(T(\mathbf{x})) + \sum_{i=1}^{r} \lambda_{it} \Phi_i(T(\mathbf{x})) + \epsilon. \tag{2}$$

[4] It is possible to assume a prior on λ_{it} as well, but in our experiments that assumption has had no noticeable effect.

We will only allow for small translations, i.e., $T(\mathbf{x}) = \mathbf{x} + \boldsymbol{\Delta}x$ where $||\boldsymbol{\Delta}x||$ is small, on the order of a few pixels. Otherwise, the region might start tracking moving foreground objects. For more details, see Section 5 on experimental results.

In principle, there are many ways to partition the image plane. For simplicitly, we have partitioned the image into blocks. The size of these blocks is a crucial parameter. If the blocks are too large, then the model is less capable of capturing local dynamics. And too small blocks will tend to, in the limit, to a pixel-based model. The issue is further discussed in the experimental section.

4 Estimation of Model parameters

There are essentially three sets of parameters that need to be estimated in the algorithm: the mean, the linear basis and the spatial transformations. The computation of the first two are described in the next section. Then follows an outline of the complete algorithm (including the estimation of the third set of parameters).

4.1 PowerFactorization

At the heart of our background subtraction algorithm is the extraction of a Principal Component Analysis (PCA) basis for the image blocks. Consider a particular region Ω_i in the image. The pixels in this block at time t may be represented by a vector $\mathbf{v}_{it}$. (Note that the pixels in $\mathbf{v}_{it}$ need not come from precisely the same position in the image at each time, since small motions of the blocks are allowed for.) For instance, in the experiments described later, each block is a block of size 17×23, so $\mathbf{v}_t$ is a vector of dimension $D = 17 \times 23 = 391$. For simplicity, we drop the subscript i, and bear in mind that the succeeding discussion refers to each block Ω_i in the image.

In order to keep a finite memory of the previous appearances of the block, we consider some number N of previous states of the block, represented by N vectors $\mathbf{v}_{t-N+1}, \ldots, \mathbf{v}_t$. In the experiments described below, $N = 50$. We denote this set of vector by $V_t = \{\mathbf{v}_{t-N+1}, \ldots, \mathbf{v}_t\}$. Our purpose is to carry out PCA on these vectors at each time t to find a small number r of such vectors that most nearly span the space containing the complete sequence $\mathbf{v}_{t-N+1}, \ldots, \mathbf{v}_t$.

At each time t, we may compute the vector $\bar{\mathbf{v}}_t$, which is the mean of all the vectors in V_t. Subtracting this mean from each $\mathbf{v}_t$, and numbering appropriately, we obtain a zero-mean set of vectors $W_t = \{\mathbf{w}_{t1}, \mathbf{w}_{t2}, \ldots \mathbf{w}_{tN}\}$, where $\mathbf{w}_{ti} = \mathbf{v}_{t-i+1} - \bar{\mathbf{v}}_t$.

To carry out PCA, on the set of vectors W_t, we form a matrix $\mathtt{M}$, the columns of which are the vectors in W_t. Thus, $\mathtt{M}$ has dimension $D \times N = 391 \times 50$ in the experiments. We wish to represent the column space of $\mathtt{M}$ by a small number r of vectors. In the experiments, $r = 4$. Equivalently, we wish to find the matrix $\widehat{\mathtt{M}}$ of rank r that is closest to $\mathtt{M}$ in an appropriate norm – the Frobenius norm. A common way that this may be done is by using the Singular Value Decomposition

(SVD). Let $\mathtt{M} = \mathtt{UDV}^\top$, where the diagonal entries of $\mathtt{D}$ (the singular values) are in descending order. Then the first r columns of $\mathtt{U}$ form an orthonormal basis for the dimension-r subspace that we require, our PCA basis. Equivalently, the closest rank-r matrix to $\mathtt{M}$ is the matrix $\widehat{\mathtt{M}} = \mathtt{UD}^{(r)}\mathtt{V}^\top$, where $\mathtt{D}^{(r)}$ represents the (diagonal) matrix formed from $\mathtt{D}$ by setting all but the first r diagonal entries to zero.

The SVD method gives an exact solution. Unfortunately, the computational cost of this algorithm is quite high, particularly since we need to carry it out for each block Ω_i in each frame. Instead, we make use of the fact that the PCA basis for a given block will not change very much from one time instant to the next. Consequently, an iterative update method is more suitable. We use the method of PowerFactorization presented in [4] to accomplish this. It should be noted that another (similar though not identical) method for iterative low-rank approximation has been suggested by [6].

We are carrying out PCA on a sliding window of N vectors derived from each block Ω_i at successive times, indexed by t. It is clear that the mean vector $\bar{\mathbf{v}}_t$ can easily be computed recursively with minimal effort at each time step. Similarly a matrix containing (as columns) the last N vectors is easily maintained by writing each new vector $\mathbf{v}_t$ over the top of the vector $\mathbf{v}_{t-N}$ that it is replacing. Then, the mean $\bar{\mathbf{v}}_t$ can be subtracted from each column to obtain the matrix $\mathtt{M}$ of dimension $D \times N$.

In the PowerFactorization algorithm, we estimate two matrices $\mathtt{A}$ of dimension $D \times r$ and $\mathtt{B}$ of dimension $N \times r$, such that $\widehat{\mathtt{M}} = \mathtt{AB}^\top$ is the closest rank-r approximation to $\mathtt{M}$. In addition $\mathtt{A}$ has orthonormal columns, which therefore form the vectors of the desired PCA basis. This is done by a simple iterative procedure as will be described soon. At each time instant t, the matrices $\mathtt{A}$ and $\mathtt{B}$ are computed iteratively, and for simplicity, we will omit the index t representing the time instant (or frame number). Subscripts represent the iteration number within the iterative procedure at a given time instant.

The iteration consists of three steps, starting from an initial value for the matrix $\mathtt{A}_0$.

1. Define $\mathtt{B}_k = \mathtt{M}^\top \mathtt{A}_k$.
2. Define $\mathtt{A}_{k+1} = \mathtt{MB}_k$.
3. Apply the Gram-Schmidt algorithm (otherwise known as QR-factorization) to orthonormalize the columns of $\mathtt{A}_{k+1}$.

The product $\mathtt{A}_k {\mathtt{B}_k}^\top$ is guaranteed ([4]) to converge linearly (even from a random starting point) to the closest rank-r matrix $\widehat{\mathtt{M}}$ to $\mathtt{M}$. In particular, for some constant K,

$$\|\widehat{\mathtt{M}} - \mathtt{A}_k {\mathtt{B}_k}^\top\| \le K(s_{r+1}/s_r)^k$$

where s_i is the i-th greatest singular value of $\mathtt{M}$.

It remains to explain how to choose an initial value for $\mathtt{A}_0$. Under the assumption that the PCA basis does not change much from one time instant to the next, the best strategy is to start from where we left off at the previous time instant. In particular, since $\mathtt{A}$ contains the PCA basis, we start with A_0 equal to

the final matrix $\mathtt{A}_{k+1}$ from the previous time instant. At the very start, when no previous value for $\mathtt{A}$ is available, we may begin with a random value of $\mathtt{A}_0$, and the algorithm will boot-strap itself.

Note that convergence of this algorithm is quickest when the ratio s_{r+1}/s_r is small – that is, the matrix is close to having rank r. This may not be the case in the present problem, but it does not cause significant difficulty, since it is not essential to have the absolute best PCA basis. In practice, a small number (we use 3) of iterations at each time step are sufficient. In our tests, this was enough to ensure that the vectors of the orthonormal PCA basis found in this way were within 1% of a true spanning set for the subspace found using the exact SVD algorithm.

4.2 Outline of the Algorithm

As the residuals between the model and the measurements are assumed to be Gaussian, cf. (2), it implies that the Frobenius norm of the residuals is the statistically correct measure to use for deciding whether a new sample is novel or not. For a given position, the coffecients λ_{it}, $i = 1, \ldots, r$ are computed by a projecting onto the basis spanned by Φ_i (which is done by a scalar product as the basis vectors are orthogonal). The residual error is computed by

$$\epsilon^2(I_t) = ||I_t - I_{mean}||_F^2 - \sum_{i=1}^{r} \lambda_{it}^2.$$

For each region in the image, the following steps are performed at time t.

1. Let $\mathbf{x}_{t-1}$ be the position in the previous image of the region. Compute the residual error ϵ at this position.
2. If $\epsilon < p_{low}$, then set $\mathbf{x}_t = \mathbf{x}_{t-1}$ and go to 5.
3. For each integer position in a neighbourhood of the previous position, compute the residual error and set $\mathbf{x}_t$ to the one with lowest error.
4. If the residual error ϵ at $\mathbf{x}_t$ is greater than p_0, then declare the region as novel.
5. Update the mean and the PCA basis incrementally using PowerFactorization.

The test in the second step above is done to speed up the algorithm. Most regions in the image do not move and hence it is not necessary to check for other translations if the reconstruction error is low for no movement. (In the experiments, $p_{low} = p_0/3$.) The neighbourhood of $\mathbf{x}_{t-1}$ is defined to include all points within a circle of the region's original position. The radius of the circle is set to 10 pixels and for computational reasons, we assume that a region moves at maximum two pixels per frame. The threshold p_0 is based on the estimated standard deviation of the $N = 50$ previous residual errors (which can be computed incrementally).

5 Experimental Results

We have tested the proposed algorithm on a number of sequences with promising results. Some typical behaviour of the algorithm on a representative selection of sample sequences are presented in this section. The algorithm has also been compared to the approach of Elgammal et al. [3] and to the Stauffer-Grimson approach [10], which can be considered to be the state-of-the-art in terms of real-time background modelling. Both of these two alternative approaches are pixel-based (see Section 1.1) and can handle multi-modal probability pixel distributions. The main difference between them is in the way the probabilitiy distributions are estimated, but the performance is similar.

The sequences presented below consist of frames with 240×320 grey-scale pixels. It is straightforward to apply the algorithm on colour images as well, though with increased computational requirements. The image has been divided into rectangle regions of 17×23 pixels, such that it is covered by a total of 15×15 rectangles. Each region is allowed to translate up to 10 pixels from its origin (but at most 2 pixels per frame for computational reasons). In the model, the number of principal components is set to 4 for each rectangle. The model is updated using the 50 previous frames (those that are not classified as novel) using the incremental PowerFactorization algorithm. The number of iterations in the innerloop of PowerFactorization is set to 3. We have experimentally validated that even with so few iterations, we approximate the true linear subspace within one percent in Frobenius norm. Our matlab implementation can perform novelty detection with 3 frames per second on a Pentium 4, 2.9 GHz.

In the left of Figure 1, one image of the forest sequence of Elgammal[5] is shown. It consists of a person walking in the woods. Partial occlusions occur due to branches and the trees sway slightly - the wind is quite moderate. Just looking at a single frame, it is hard to detect the person, however, the algorithm has no problem in tracking the person. The detected squares are marked in the middle of Figure 1. The sequence has also been tested with the algorithm of Elgammal [3] and here the result is even better. As this approach is pixel-based, one gets a more precise localization of the foreground object compared to a region-based approach.

The next sequence shows a traffic intersection (also from Elgammal), Figure 2. The difficulty in this sequence is due to the heavy rain, which may be hard to spot in a single frame. The results of both our and Elgammal's algorithms are given in the same figure. As can be seen, the two algorithms perform well, though, the techniques for handling the distortions caused by the rain are quite different. In the first case, the PCA model finds a suitable subspace for each region, while in the latter case, the effects are modelled by a non-parametric estimate of the pdf on a pixel basis.

The next sequence is similar to the previous one, except that the camera is not stable - it shakes a bit and as a result the image jumps up and down a

[5] The Elgammal sequences can be downloaded from `http://www.cs.rutgers.edu/~elgammal`.

Fig. 1. Left: An example frame in the Elgammal forest sequence. **Middle**: Output of our algorithm. Rectangles detected as novel are marked. **Right:** Thresholded result of Elgammal [3].

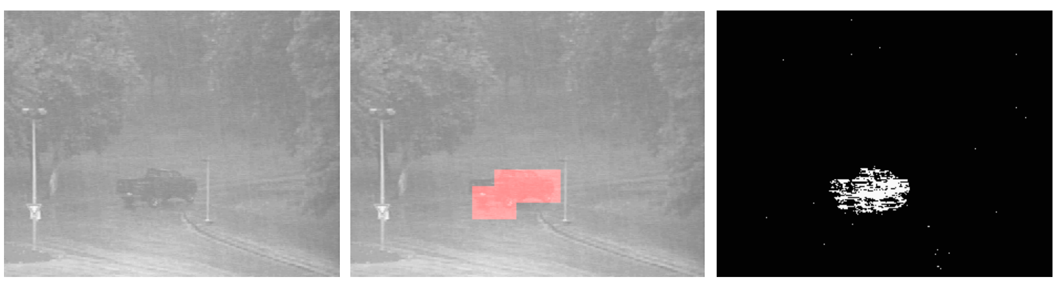

Fig. 2. Left: One frame in the Elgammal rain sequence, followed by the results of our algorithm (**middle**) and Elgammal (**right**).

few pixels in the sequence. The output of our algorithm is presented in Figure 3 and as can be seen, the spatial transformations (i.e., the translations) are able to compensate well for the shaking image. In the left of Figure 4 the result of Elgammal's standard pixel-based algorithm is shown - a lot of pixels are classified as novel. Elgammal et al. have also developed a motion-compensated version of their algorithm, which tests if a given pixel can be explained by neighbouring pixel distributions (see right of Figure 4). Now, the shakiness is eliminated from the output.

Fig. 3. Left: One frame in the shaking camera sequence. **Middle**: The output of our algorithm.

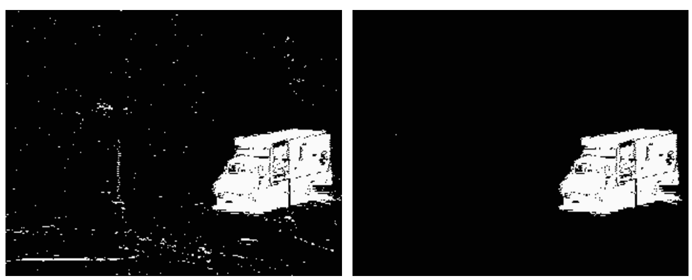

Fig. 4. The result of Elgammal witout motion-compensation (**left**) and with motion-compensation (**right**).

So far, we have only demonstrated that our algorithm works fine where alternative approaches may also work well. The next sequence is somewhat harder. The scene depicts a road with a swaying tree and a lot of cast shadows all over the image. Intermittent gusts of wind cause vivid motion of the tree's branches and the corresponding shadows to move vigorously. Three frames in the sequence are shown in Figure 5 and the corresponding detection results of our scheme in Figure 6 and Stauffer-Grimson in Figure 7.

Fig. 5. Three frames of the Daley road sequence. Notice, in the right image a cyclist can be spotted.

Fig. 6. The detection results of our algorithm for the images in Figure 5.

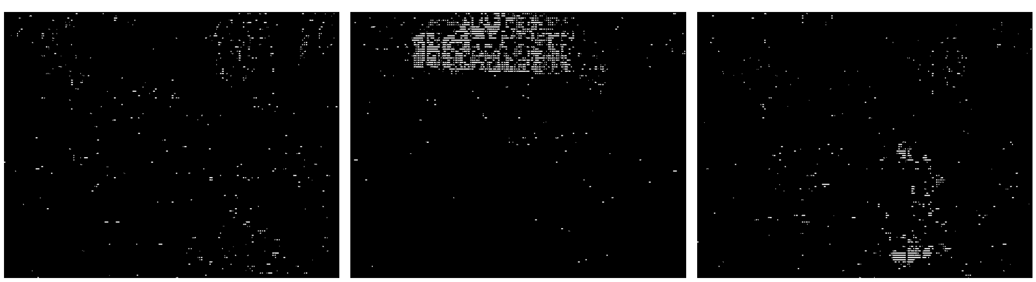

Fig. 7. The detection results of Stauffer-Grimson algorithm for the images in Figure 5.

We have tested this sequence with the Stauffer-Grimson algorithm[6] with up to 8 Gaussians in the mixture, but without great success. The pixelwise distributions are not able to capture the non-stationary texture well. As a comparison, we have graphed the percentage of pixels that are classified as novel with their algorithm and similarly with ours (Figure 8). Notice that even though there are high peaks where foreground objects appear, there are also peaks at other times where there is no actual novelty. The situation looks much better in the second case, even though a few blocks are occasionally misclassified.

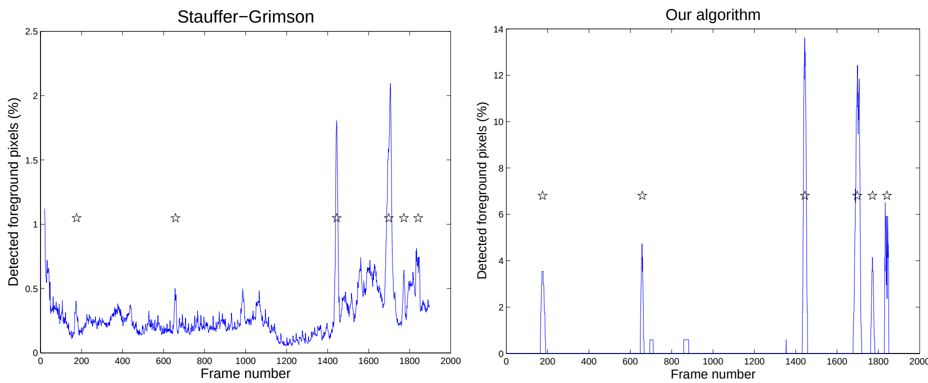

Fig. 8. Results of Stauffer-Grimson (**left**) and our algorithm (**right**) on Daley Road sequence. The stars indicate where actual cars or cyclists pass by.

5.1 Limitations

Testing the algorithm on a number of sequences, we have found some drawbacks. One parameter that is crucial is the size of regions in the images.

- We have noticed at some occasions that if the novel foreground object is relatively small compared to the region size, the region may not flag it as novel. Instead, the small foreground object is temporarily merged into the background model.
- Compared to a pixel-based methods, the detection boundary of the foreground objects is less precise.
- Slowly moving foreground objects may adaptively be merged into the background model. In the current setting, this hardly ever happens, as objects would need to move quite slowly.

A limitation of the current paper is of course that it lacks a full experimental comparison to the work of Monnet et al. [5]. It seems likely that their model is

[6] Thanks to D. Magee, University of Leeds, for making the algorithm publicly available.

more suitable for truly, repetitive patterns like ocean waves, but our approach would perform better on scenes with local, spatial dynamics like waving trees. The implementation of Elgammal's algorithm was neither available, therefore only a comparison of the demonstration scenes provided on his web page was possible.

6 Conclusions

In this paper, we have developed a new scheme for novelty detection in image sequences, capable of handling non-stationary background scenarious. The main advantages of the system are its simplicity, the powerful combination of a PCA model with local, spatial transformations and a fast, incremental PCA through PowerFactorization.

References

1. T. F. Cootes, G. J. Edwards, and Taylor C. J. Active appearance models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.
2. A. Dick and M.J. Brooks. Issues in automated visual surveillance. In *International Conference on Digital Image Computing: Techniques and Applications*, Sydney, Australia, 2003.
3. A. Elgammal, D. Harwood, and L. S. Davis. Non-parametric model for background subtraction. In *European Conf. Computer Vision*, volume II, pages 751–767, Dublin, Ireland, 2000.
4. R. Hartley and F. Schaffalitzky. Powerfactorization : Affine reconstruction with missing or uncertain data. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Washington DC*, 2004 (submitted).
5. A. Monnet, A. Mittal, N. Paragios, and V. Ramesh. Background modeling and subtraction of dynamic scenes. In *Int. Conf. Computer Vision*, pages 1305–1312, Nice, France, 2003.
6. T. Morita and T Kanade. A sequential factorization method for recovering shape and motion from image streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(8):858–867, 1997.
7. N. Ohta. A statistical approach to background substraction for surveillance systems. In *Int. Conf. Computer Vision*, volume II, pages 481–486, Vancouver, Canada, 2001.
8. C. Ridder, O. Munkelt, and H. Kirchner. Adaptive background estimation and foreground detection using kalman filtering. In *Int. Conf. on Recent Advances in Mechatronics*, pages 193–199, 1995.
9. S. Soatto, G. Doretto, and Y.N. Wu. Dynamic textures. In *Int. Conf. Computer Vision*, volume II, pages 439–446, Vancouver, Canada, 2001.
10. C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Conf. Computer Vision and Pattern Recognition*, volume II, pages 246–252, Fort Collins, USA, 1999.
11. C.R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland. Pfinder: real-time tracking of the human body. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.