Machine Vision and Applications
DOI 10.1007/s00138-010-0251-6

SPECIAL ISSUE PAPER

Dynamic texture as foreground and background

Dmitry Chetverikov · Sándor Fazekas · Michal Haindl

Received: 6 October 2008 / Revised: 4 August 2009 / Accepted: 18 January 2010

Abstract Depending on application, temporal texture can be viewed as either foreground or background. We address two related problems: finding regions of dynamic texture in a video and detecting moving targets in a dynamic texture. We propose efficient and fast methods for both cases. The methods can be potentially used in real-time applications of machine vision. First, we show how the optical flow residual can be used to find dynamic texture in video. The algorithm is a practical, real-time simplification of the sophisticated and powerful but time-consuming method (Fazekas et al. in Int J Comput Vis 82:48–63, 2009). We give numerous examples of detecting and segmenting fire, smoke, water and other dynamic textures in real-world videos acquired by static and moving cameras. Then we apply the singular value decomposition (SVD) to a temporal data window in a video to detect targets in dynamic texture via the residual of the largest singular value. For a dynamic background of low-temporal periodicity, such as water, no temporal periodicity analysis is needed. For a highly periodic background such as an escalator, we show that periodicity analysis can improve detection results. Applying the method proposed in Chetverikov and Fazekas (Proceedings of British machine vision conference, vol 1, pp 167–176, 2006), we find the temporal period and use the resonant SVD to detect moving targets against a time-periodic background.

D. Chetverikov (✉) · S. Fazekas
Computer and Automation Research Institute (SZTAKI),
Budapest, Hungary
e-mail: csetverikov@sztaki.hu

M. Haindl
Institute of Information Theory and Automation,
Prague, Czech Republic

Keywords Dynamic texture · Detection · Optical flow · Background modelling · SVD · Photometric invariants · Temporal periodicity

1 Introduction

Detection of specific dynamic textures (DTs), such as smoke or fire [37,38], is one of the most frequent surveillance applications of temporal texture analysis. In these applications, the detection is usually based on specific features (colour, frequency, etc.) of the phenomena to be detected. A different task is detection of any dynamic texture in a video. To solve this task, one has to define and use generic properties of dynamic texture rather than specific properties of certain classes of dynamic textures. In the study [12], intrinsic dynamics is identified as a generic property of dynamic texture. This property can be expressed in terms of greyscale or colour optical flow, resulting in robust and precise DT detection and segmentation in real-world videos acquired by static and moving cameras [11,12].

Most of the previous work on temporal texture segmentation is based on DT modelling by linear models [3,7,10]. These models have been successful in synthesising different dynamic textures, recognising pre-segmented DTs, and segmenting artificial DT mosaics consisting of a small number of distinct dynamic textures, such as fire on the water. However, due to their complexity, the linear models do not seem to be suitable for applications where real-time processing of real-world video data is needed, especially when the data are acquired by a moving camera. Crivelli et al. [8] use mixed-state Markov random fields to model motion textures and segment artificial and natural texture mosaics consisting of two classes of DTs. Ghoreyshi and Vidal [15] apply Ising descriptors and autoregressive exogenous models in a

Published online: 21 February 2010

level set framework to segment similar data. Rahman and Murshed [33] consider a set of DTs combined to form different two-class mosaics. They use spatiotemporal co-occurrence [34] to determine which two of the set are in the mosaic.

In DT detection and segmentation, temporal textures are regions of interest treated as foreground. A different but related surveillance application of dynamic texture processing is background modelling and target detection for dynamic backgrounds such as waving trees, sea waves, or escalators [22,26,39]. Here, moving targets are regions of interest while temporal texture is the background of the targets.

Numerous methods and practical solutions for background subtraction and target (change, novelty) detection are available [20,32]. In video surveillance, adaptivity to illumination changes and shadows is a basic requirement [30,36]. A fundamental assumption of most existing methods for background modelling is that of the static background against which changes and motion in the scene are detected and tracked. Some recent techniques [22,26] allow for a limited, low-periodicity motion of the background. Very few studies (e.g., [39]) consider background formed by a dynamic texture with high temporal periodicity. The reason is that most of the statistical models used for background modelling cannot appropriately incorporate temporal or spatial periodicity.

In this paper, we propose two efficient and fast methods to handle dynamic texture as foreground or background. Both methods are based on the residual of a representation. To find dynamic texture in video, we use the optical flow representation which assumes that image motion complies with the intensity constancy constraint [19]. We treat regions with large residual of the optical flow as regions of high intrinsic dynamics, which we attribute to dynamic texture. This algorithm is a real-time simplification of the method [12] which is accurate and robust but time-consuming. We demonstrate that when the camera motion is relatively slow and smooth, the performance of the proposed method is comparable to that of the sophisticated algorithm [12].

Our second contribution is a novel SVD-based algorithm for moving target detection in a dynamic texture. We use a sliding temporal data window of n frames, obtain the SVD of the window by a fast running algorithm, then detect targets by thresholding the residual of the largest singular value. Previous attempts to use the PCA or SVD for dynamic background modelling [22,26] applied approximate solutions to speed up the re-calculation of the SVD as the temporal data window moves on. This leads to error accumulation and the necessity to re-initialise the process when the error grows beyond an acceptable limit. We use a precise (approximation-free) algorithm which is fast and needs no re-initialisation. Another advantage of our method is that it is applied to the entire image, while the methods [22,26] split the image into blocks in order to achieve acceptable processing times. This solution introduces an additional sensitive parameter, the block size, and makes the result patchy.

The fast running SVD can be applied to single-channel as well as multi-channel data. In target detection against a dynamic background, the motion of the background can be random, slightly periodic, or highly periodic. We discuss the issues of colour representation and background motion periodicity in relation to the proposed approach. Recently, the photometric invariants based on the dichromatic reflection model [35] have been successfully used for illumination-robust variational optical flow calculation [25]. We demonstrate that for some backgrounds such as water, photometric invariants are preferable to the intensity or the original colour representation. Then we show that for a highly periodic background such as an escalator, temporal periodicity analysis can improve detection results. To achieve this, we introduce the notion of the resonant SVD as the SVD for the temporal spacing equal to the motion period of the background dynamic texture. The method needs a good estimate of the resonant spacing which is obtained by an algorithm based on the principles proposed in Kanjilal et al. [23] for signals and applied in Chetverikov and Fazekas [5] to dynamic texture. The periodicity estimation algorithm can also be useful in video mosaicking and dynamic texture recognition [13].

The structure of this paper is as follows. In Sect. 2, the method [12] is briefly discussed to provide motivation and background for the proposed simplified real-time algorithm. The presentation of the proposed algorithm is followed by test results on real-world data acquired by static and moving cameras. The results are qualitatively compared to those obtained by the original method [12]. In Sect. 3, the fast running SVD algorithm is presented and tested. A discussion of colour representation, temporal periodicity and other practical issues closes the section. Finally, conclusions are drawn in Sect. 4.

2 Detecting dynamic texture in video

In this section, we demonstrate that the optical flow residual can be used as a tool to detect and segment dynamic texture in video. Motion based segmentation methods often use optical flow algorithms to estimate the motion. Efficient, fast and precise optical flow estimation procedures are available [1,2,18,24]. Most of them are based on the brightness constancy assumption and minimise the optical flow residual

$$R_{\mathrm{of}} = \left\langle |I(x+u, y+v, t+1) - I(x, y, t)| \right\rangle_w, \quad (1)$$

where $I(x, y, t)$ is the image function at discrete time t, (u, v) the displacement field, $\langle \cdot \rangle_w$ a convolution with a Gaussian kernel w. $I(x+u, y+v, t+1)$ is calculated with subpixel accuracy by interpolation. Additional constraints are used to overcome the aperture problem and provide a smooth

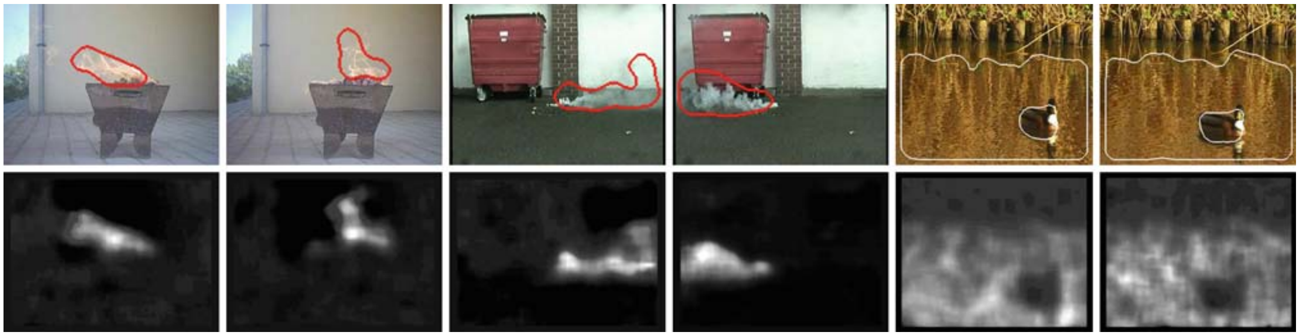

Fig. 1 *Top row*: Sample frames of three video sequences with the dynamic texture regions detected. The videos were acquired by static camera. *Bottom row*: The optical flow residual maps

flow. The accuracy and speed of optical flow calculation can be enhanced by using a multi-resolution coarse-to-fine approach.

Many methods for dynamic texture analysis are based on optical flow [6,34]. Recently, Fazekas et al. [12] proposed a sophisticated method for finding regions of dynamic texture in video. In such regions, the traditional brightness constancy assumption is not valid because of the intrinsic dynamics which is a generic property of temporal textures. To better account for this dynamics, in regions of temporal texture brightness constancy should be substituted by the more general assumption of brightness conservation. The task of finding dynamic texture in video is formulated as a variational problem which is solved by a level set method. The Lagrangian incorporates an indicator function that switches on the border of a dynamic texture region in a kind of competition between the two assumptions: areas with translational motion are better described by brightness constancy, while areas with intrinsic dynamics are better described by brightness conservation.

The method [12] has been applied to numerous real-world videos taken by static and moving cameras and containing smoke, fire, water and other natural dynamic textures in natural contexts. A collection of impressive results is available on the web site [11]. However, the method [12] is too slow to be applied in real time. Providing a simplified fast approach is desirable for real-time applications such as fire and smoke detection in surveillance video.

We observe that the traditional brightness constancy based optical flow mainly errs in areas of dynamic texture and areas of occlusion. Next, we assume that occlusion borders are narrow while areas of dynamic texture are usually wider and larger. The proposed algorithm then involves the following steps:

1. Calculate the optical flow.
2. Calculate its residual (1) by mapping the first frame onto the second one with subpixel accuracy using the displacement field.
3. When three or more consecutive residual frames are available, process them by a fast spatiotemporal median filter of large spatial size and small temporal depth.

The result of this simple algorithm is the flow residual map which can be thresholded to find regions of dynamic textures. In the tests, the size of the spatiotemporal median filter for the initial video resolution was $25 \times 25 \times 3$. When reduced resolution was used for faster operation, the spatial size of the filter was reduced accordingly. The residual threshold can be set manually; in most case, the value of 3 is appropriate for greyscale images with 256 levels. Alternatively, one can set the threshold in an adaptive way. Assuming that the dynamic texture occupies a significant part of the data in the first n frames, one can calculate the residual histogram in the n frames, obtain the threshold by any histogram thresholding technique (we use [28]), then update the histogram and the threshold by sliding the n-frame data window by one frame.

Figure 1 shows frames of three sequences with dynamic texture regions detected. The bottom row shows the optical flow residual maps where the areas of dynamic texture are visible. The first two sequences showing fire and smoke are low-resolution videos taken by a static camera. These videos were kindly provided by the Bilkent University in the framework of the MUSCLE Network of Excellence [27]. (See [9,38] on the related research in fire and smoke detection.) The third sequence, also acquired by a static camera, comes from the public DynTex dynamic texture database [29] created by the MUSCLE NoE.

Figure 2 gives more examples. The videos are from the DynTex database. Here, the first three videos were taken by a panning video camera of good quality. The 'Cooking' sequence contains both smoke and fire which are detected in the same way, by a generic method selective to any dynamic texture. In the 'Candles' video, the hand-held camera was scanning the scene in a more irregular way. In all cases, the camera motion was relatively slow.

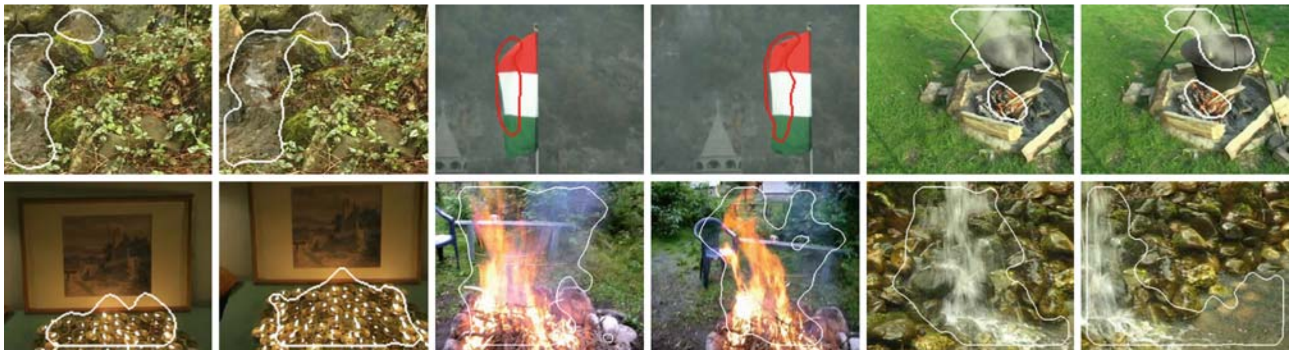

Fig. 2 Sample frames of six video sequences with the dynamic texture regions detected. The videos were acquired by moving camera

In the above test sequences, four different dynamic textures, smoke, fire, water and flag, were successfully detected. The borders of fire and, especially, smoke are fuzzy, so one cannot speak of precise segmentation. In the case of the flag, only the blowing part can be detected, as the rest is not a dynamic texture. The borders of the water stream are distinct; here, one can speak of segmentation which was successful.

Using the proposed method, we have processed the videos presented at the web site [11]. Most of these videos come from the DynTex dynamic texture database [29]. Currently, the DynTex contains no ground truth. Qualitative comparison to the results [11] is still possible, and some conclusions can be made. The sophisticated method [12] is more precise and more robust to camera motion. It works for faster and more irregular camera motion than our simple method because it has no threshold to set, and its motion estimation is more precise. However, many of the videos [11] can be successfully processed by our method as well, at video rate or close to video rate, depending on the resolution and the optical flow used. We have tested our method with the fast Lucas–Kanade [24] and the slower but more precise Horn-Schunck [18] optical flow algorithms. Multi-resolution versions of the OpenCV implementations [21] were used to cope with fast motion. We concluded that the precision of the Lucas–Kanade algorithm is sufficient to provide thresholdable residual maps in most cases. Reducing the full video frame size by a factor of four (to 144×176) and using the fast Lucas–Kanade algorithm, real-time operation can be achieved on a modern laptop.

3 Target detection in dynamic texture

Detecting objects against a dynamic background is a challenging problem important for traffic and surveillance applications. Classical target detection algorithms using adaptive background models [20] and more recent approaches, such as those presented at the IEEE Workshops on Visual Surveillance, have a limited capability to cope with varying backgrounds such as trees in the wind. When the background is time-periodic, like an escalator or conveyor, target detection may become difficult. Very few studies show results for time-periodic backgrounds. Zhong and Sclaroff [39] describe dynamic textures by an autoregressive moving average (ARMA) model and use the Markov assumption (dependence on the previous state only) and a Kalman filter to estimate the dynamic background and detect objects. Examples of successful detection are shown for an object floating in a waving river and a ball jumping on a moving escalator. However, this approach is, in general, not suitable for time-periodic dynamic textures where the Markov assumption is not valid.

The singular value decomposition (SVD) is a robust and powerful tool having numerous applications in signal processing, computer vision, pattern recognition and other areas. (See Appendix for a definition of the SVD.) In particular, it is suitable for separation of dominant, 'typical' data from 'untypical' data. Recently, it has been successfully used for adaptive modelling of non-stationary background and detection of novelty in image sequences [22,26]. For greyscale images, the data matrix $\boldsymbol{A}$ of size $m \times n$ is formed by m consecutive frames of a sequence, where n is the number of pixels in a frame. Each image is read row-by-row and stored in the appropriate column of $\boldsymbol{A}$. n is the size of the temporal data window which scans the sequence. For colour images, each pixel is represented in $\boldsymbol{A}$ by three values, e.g., the RGB codes. The detection is based on the residual error of the SVD $\boldsymbol{A} = \boldsymbol{USV}^T$ for a number of the largest singular values s_i which are the elements of the diagonal matrix $\boldsymbol{S}$.

The results presented in [22,26] are promising. However, the algorithmic solutions used in these studies limit the applicability of the SVD method. To decrease the computational load of the batch PCA, images are split into blocks and an approximate incremental PCA is applied in each block separately. In this section, we propose a fast and approximation-free incremental SVD which is applied to the whole image at once. Then we use the proposed method for target detection in dynamic textures with different degrees of temporal periodicity and discuss the issue of colour representation. For highly periodic backgrounds, we introduce and use the resonant SVD. We demonstrate that using the resonant SVD for periodic data and photometric invariants for DTs with shadows and shading can improve detection results.

3.1 Using fast running SVD for target detection

Incremental eigenanalysis has been of interest for machine vision since the publication of the first promising results by appearance-based and active contour methods that rely on the principal component analysis [17]. Eigenspace models of large databases need to be incrementally modified when the data are modified, added or deleted. When eigenanalysis is applied in a sliding spatial or temporal data window, one needs fast re-calculation of the eigenvalue decomposition or SVD as a column enters the data matrix and a column leaves the matrix. Algorithmically, the re-calculation consists of two different operations, the update step and the downdate step. Updating SVD models is usually considered to be a relatively simple and robust operation, lending itself to stable and accurate approximate solutions. Assuming that the SVD basis does not change drastically, an iterative update procedure can be used [22]. Downdating SVD models is a more difficult problem: it was even argued that downdating is impossible in closed form [17]. Fortunately, this is not the case.

We call the precise incremental method the running SVD using the word 'running' in the same sense as it is used in the term 'running filter'. Our running algorithm combines two existing procedures and includes a fast update step and a fast downdate step [16]. Both steps are precise: no approximation is used. When applied to a video sequence, the update step adds a new frame to the SVD of the current data window, while the downdate step removes the exiting frame from the updated SVD. A detailed description of the method [4] is quite lengthy and involved. For completeness, a brief description and the Matlab (Octave) codes of the update and downdate procedures are given in Appendix.

The complexity of the complete running algorithm is $\mathcal{O}((m+n)m^2)$, where n is the image size, m the number of frames in a temporal data window. (See [4,16] for derivation and discussion.) Typically, $m \leq 50$ and $n \gg m$, so in practice the computational load is $\mathcal{O}(nm^2)$. The linear dependence on image size n means that splitting the image into blocks as in [22,26] will not speed up the algorithm and is unnecessary. (For comparison, note that in [39] this dependence is quadratic which limits the image size.) The quadratic dependence on the number of frames m is prohibitive for very large $m > 100$; however, this problem can be tackled by reducing the temporal resolution.

To make the method less sensitive to illumination changes, shadow and shading, one can use the photometric invariants of the dichromatic reflection model [35], as proposed in [25] for illumination-robust optical flow estimation. (Note that this can only be done for colour videos.) For an RGB-coded video, the options insensitive to shadow and shading are the normalised RGB values or the angles ϕ, θ of the spherical (conical) transformation. The normalised RGB values are defined as

$$\left(\frac{R}{N}, \frac{G}{N}, \frac{B}{N}\right)^T,$$

where N is either the arithmetic mean $R+G+B$ or the geometric mean $\sqrt[3]{RGB}$. The spherical transformation is given by

$$r = \sqrt{R^2+G^2+B^2} \qquad (2)$$

$$\theta = \arctan\left(\frac{G}{R}\right) \qquad (3)$$

$$\phi = \arcsin\left(\frac{\sqrt{R^2+G^2}}{\sqrt{R^2+G^2+B^2}}\right) \qquad (4)$$

Note that r is not a photometric invariant, hence for the spherical transformation two channels are only used.

Figures 3, 4, 5 and 6 show examples of target detection in natural dynamic textures with low temporal periodicity. In the 'Bush' sequence, a man passes a bush waving in the wind. The motion of the bush becomes stronger in the end of the sequence. Figures 3 and 4 compare the proposed methods to the classical adaptive Gaussian mixture method [36]. The latter fails when the background motion is strong, while the proposed method works properly in the whole sequence. One can observe, however, that the contours of the target as detected by our method are less precise. This can be improved by using a more sophisticated decision procedure: in this study, we simply threshold the residual.

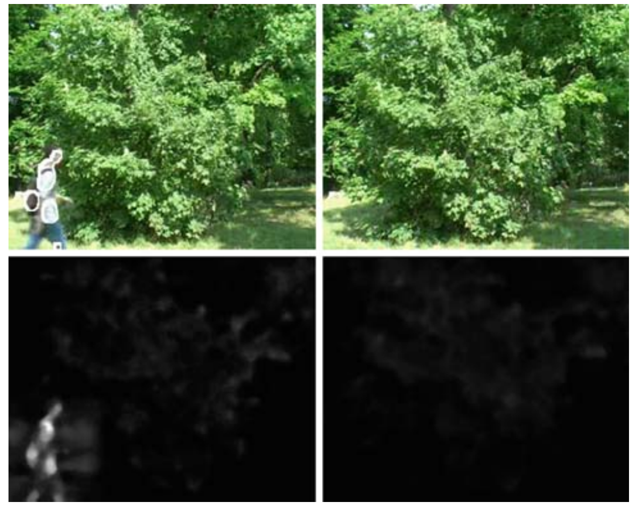

Fig. 3 Sample frames of the 'Bush' sequence. *Top row*: Results of target detection by our method using the original RGB colours. *Bottom row*: Residual maps

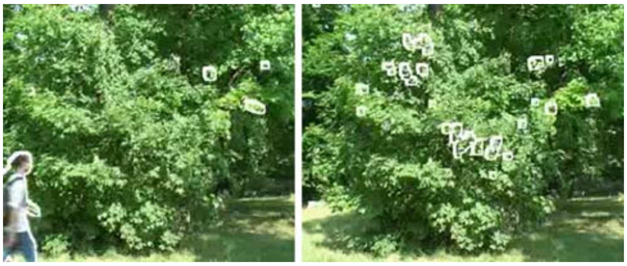

Fig. 4 Results of target detection in the 'Bush' sequence by the method [36]

Fig. 5 Sample frames of the 'Duck' sequence. *Top row*: Results of target detection using our method with photometric invariants. *Bottom row*: Residual maps

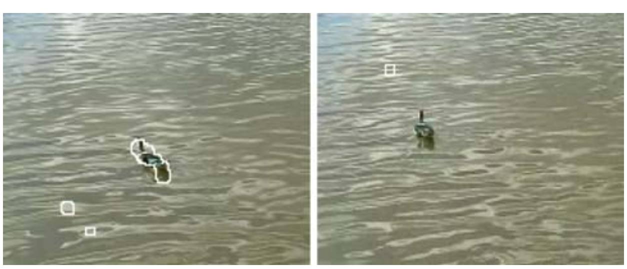

Fig. 6 Results of target detection using our method with the original RGB colours

Figures 5 and 6 illustrate the application of the photometric invariants to a dynamic background with frequent shading. In the 'Duck' sequence, a duck floats away on the wavy water. The proposed SVD based method applied to the normalised RGB (Fig. 5), or to the angles ϕ, θ provides a stable solution, while using the original RGB values (Fig. 6) results in errors and loss of the target when it becomes smaller. The Gaussian mixture method [36] cannot cope with this dynamic background at all.

3.2 Periodicity estimation for time-periodic background

When the dominant data are periodic, care should be taken to understand the typical periodic structure and find untypical, salient data as significant local deviations from this structure. As discussed by Kanjilal et al. [23], understanding a periodic signal means extracting the pattern which is repeated, the period length, and the scaling factor of each period. The signal periodicity analysis approach proposed in [23] involves signal pre-processing and normalisation steps followed by the SVD of the signal data matrix for a range of potential period lengths. For a signal $x(k)$ and a hypothesised period n, the columns of the data matrix $\boldsymbol{A}_n$ are the consecutive signal intervals of length n. When n coincides with the actual period length N, the rank of $\boldsymbol{A}_n$ is close to one. This rank can be robustly estimated by decomposing the matrix with the SVD: $\boldsymbol{A}_n = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$, where the diagonal matrix $\boldsymbol{S}$ contains the sorted singular values s_i. The periodicity spectrum (P-spectrum)

$$P(n) = 1 - \frac{s_2(n)}{s_1(n)} \tag{5}$$

has distinct maxima at $n = N, 2N, 3N$, etc. All maxima of $P(n)$ are tested and the periodicity index (P-index) is calculated to separate the true periodicity from noisy maxima.

In [5], we applied this method to estimation of the temporal periodicity of dynamic textures. For more technical details, including pre-processing and normalisation, the reader is referred to the papers [5,23]. We use the periodicity estimation method to find the dominant periodic structure as the data representation $\boldsymbol{u}_1 s_1 \boldsymbol{v}_1^T$ for the largest singular value s_1 at the resonant value $n = N$. The vector $\boldsymbol{v}_1$ is the (normalised) periodic pattern, the elements of $\boldsymbol{u}_1 s_1$ are the scaling factors. The residual of this representation,

$$R_{\mathrm{svd}} = \boldsymbol{A}_N - \boldsymbol{u}_1 s_1 \boldsymbol{v}_1^T, \tag{6}$$

indicates defects in spatially periodic textures and targets in temporally periodic sequences. We call the period length the 'resonant spacing' and refer to the corresponding SVD representation for the largest singular value as the 'resonant SVD representation'.

For better understanding of the method, we first illustrate its operation on a static greyscale texture with orthogonal axes of periodicity, assuming for simplicity that one of the axes is vertical. (See Fig. 7.) The texture contains a defect which is not easy to perceive and detect. The period length in the vertical direction is estimated as described above, by putting two, three or more image rows into each column of the data matrix $\boldsymbol{A}_n$ to test the period length $n = 2, 3$, etc. Figure 8 shows the P-spectrum and P-index of the test pattern. In the P-index, the true period of 36 rows is clearly visible.

Once the period N has been obtained, the matrix $\boldsymbol{u}_1 s_1 \boldsymbol{v}_1^T$ is transformed into the resonant SVD model, while the

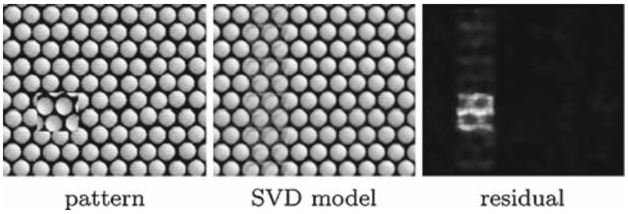

Fig. 7 *Left*: Periodic test pattern with an artificial defect. *Middle*: The resonant SVD model for the row spacing equal to the period which is 36 pixels. *Right*: The residual map indicating the defect

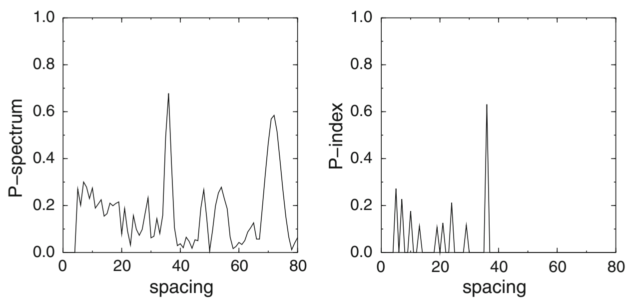

Fig. 8 The P-spectrum and P-index of the test pattern shown in Fig. 7

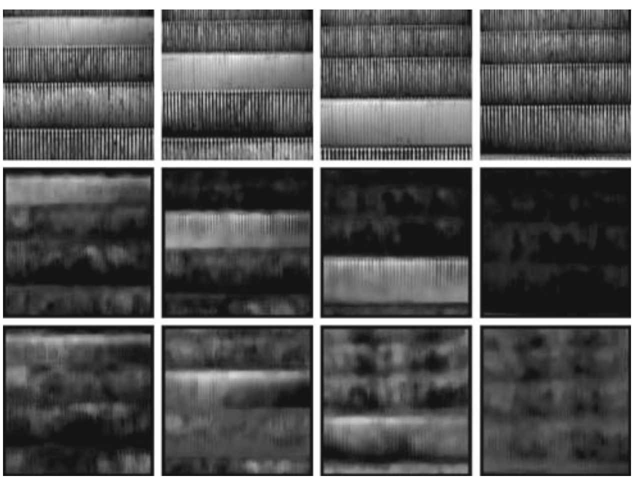

Fig. 9 *Top row*: Sample frames of the 'Escalator Steps' sequence. *Middle row*: The resonant residual maps highlighting the irregular step. The period is 21 frames. *Bottom row*: The residual map for the temporal spacing of 19 frames

residual matrix (6) is converted into the residual map taking the absolute values of the matrix. Figure 7 shows the resonant SVD representation of the artificial test pattern and the corresponding residual map highlighting the defect.

Let us now consider periodic dynamic texture viewed as background. For time-periodic video, the columns of the data matrix are composed of multiple consecutive images. Note that a static camera is assumed to ensure correspondence between pixels. Figure 9 shows frames of the greyscale sequence 'Escalator Steps' whose total length is about 300 frames. In this video an irregular white step enters, passes the

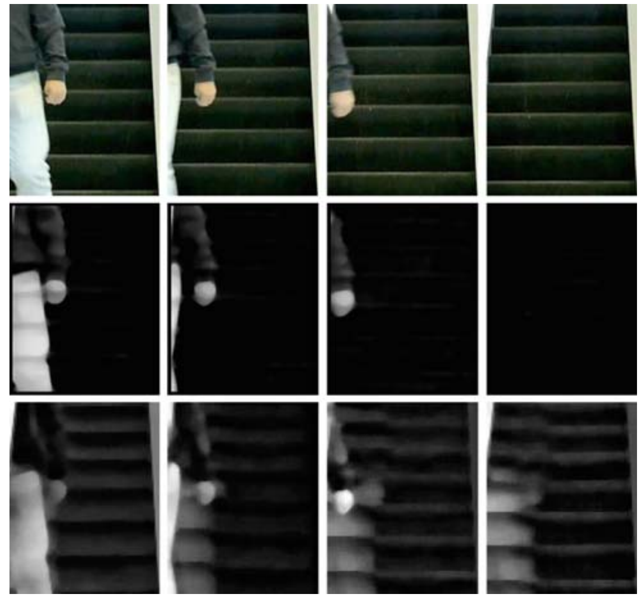

Fig. 10 *Top row*: Sample frames of the 'Man and Escalator' sequence. *Middle row*: The resonant residual maps. *Bottom row*: Residual maps obtained in a sliding temporal data window with unit temporal spacing

viewfield, then exits. The rest of the video shows the empty escalator in periodic motion. To speed up the period estimation procedure, the original spatial resolution of 200×200 was significantly reduced. Then the algorithm was run again, at the full resolution for the resonant spacing, to obtain the full-resolution resonant SVD representation. The second row of Fig. 9 shows the resonant residual map selective to the irregular step. The bottom row illustrates the importance of obtaining a good estimate of the period: for the temporal spacing of 19 instead of the true period of 21, the result is much worse.

Figure 10 shows frames of the greyscale sequence 'Man and Escalator' whose total length is about 200 frames. In this video a man leaves the escalator and exits. The rest of the video shows the empty escalator in periodic motion. The second row of the figure shows the residual map indicating the man against the time-periodic background. Note that the escalator has almost disappeared in the residual. On the other hand, the contrast of the map is low where the intensity difference between the man and the escalator is small. Finally, the bottom row again illustrates the importance of obtaining the resonant SVD representation in the case of periodic data.

The periodicity estimation method and the resonant SVD are only used for dynamic textures of high temporal periodicity. In target detection applications, it is usually known if the background is temporally periodic, or not. In the tests presented in Sects. 3.1 and 3.2, we assumed that this is known and applied the corresponding version of the method. When the periodicity is unknown, one should first calculate the

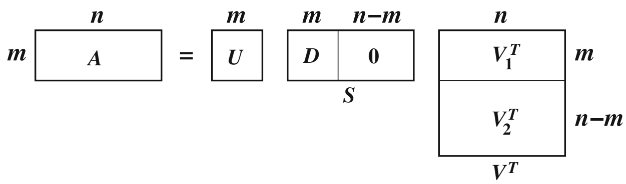

Fig. 11 Illustration to the SVD and the update procedure

P-spectrum (5) and decide if the dynamic texture if highly periodic, or not.

4 Conclusion

We have given a summary of our recent results in dynamic texture detection and target detection in dynamic background. Both tasks are relevant for machine vision applications such as surveillance, security and traffic or crowd monitoring where fast and robust solutions are needed. The proposed method for real-time temporal texture detection can enhance the existing feature-based methods for detection of smoke and fire, by efficiently discarding false positives like moving objects whose colour resembles fire or smoke. An important advantage is that this can be done by a moving camera as well, allowing to survey a larger area. Our fast running SVD can handle backgrounds with significant, possibly periodic, dynamics. Depending on the video resolution and the size of the temporal data window, the method can be used at video rate, or close to video rate. We have successfully tested the method on numerous traffic monitoring videos, including snow, fog, and shadows. Using photometric invariants makes the method applicable to backgrounds with shadows and frequent shading. For periodic backgrounds, temporal periodicity analysis may be desirable. We have provided a method for periodicity estimation in video based on the resonant SVD. The method tests a range of potential periods, which can be done at a low, reduced resolution. This procedure is not real-time. Fortunately, the period of man-made dynamic textures, e.g., escalators, is usually stable. The period can be learned prior to the real-time operation in which minor, fast corrections will only be necessary.

Acknowledgements This research was supported in part by the NKTH-OTKA grant CK 78409 and by the MUSCLE Network of Excellence (FP6-507752).

Appendix: Matlab (Octave) codes of the update and downdate procedures

Let $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ be a real-valued matrix. Then there always exists a decomposition $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$, where

- The matrix $\boldsymbol{U} \in \mathbb{R}^{m\times m}$, $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$, consists of m orthonormal eigenvectors which belong to the m eigenvalues of $\boldsymbol{A}\boldsymbol{A}^T$. ($\boldsymbol{I}$ is a unit matrix.)
- The matrix $\boldsymbol{V} \in \mathbb{R}^{n\times n}$, $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$, consists of n orthonormal eigenvectors of $\boldsymbol{A}^T\boldsymbol{A}$.
- $\boldsymbol{S} \in \mathbb{R}^{m\times n}$ is a diagonal matrix $\mathrm{diag}(s_1, \ldots, s_m)$ containing the singular values of $\boldsymbol{A}$ which are the square roots of the eigenvalues of $\boldsymbol{A}^T\boldsymbol{A}$. It is usually assumed that $s_1 \geq s_2 \geq \cdots \geq s_m \geq 0$.

The above decomposition is called the singular value decomposition, or SVD. Its structure is illustrated in Fig. 11. Several methods to calculate the SVD exist. The Golub–Reinsch procedure [14,31] and its modification for rectangular matrices are probably the best known. We use this procedure and assume that $m < n$.

The update procedure receives the components of the previous SVD $\boldsymbol{U}$, $\boldsymbol{S}$, $\boldsymbol{V}_1$, as well as the vector (row) $\boldsymbol{a}^T$ to be added to the matrix $\boldsymbol{A}$. $\boldsymbol{V}_1$ is the result of the following decomposition:

$$\boldsymbol{V} = \begin{pmatrix} \boldsymbol{V}_1 & \boldsymbol{V}_2 \end{pmatrix}, \quad \boldsymbol{S} = \begin{pmatrix} \boldsymbol{D} & \boldsymbol{0} \end{pmatrix}, \tag{7}$$

where $\boldsymbol{V}_1 \in \mathbb{R}^{n\times m}$, $\boldsymbol{V}_2 \in \mathbb{R}^{n\times(n-m)}$, $\boldsymbol{D} \in \mathbb{R}^{m\times m}$, and $\boldsymbol{0}$ is a zero matrix. (See Fig. 11.) In other words, $\boldsymbol{V}_1$ is the 'useful' part of $\boldsymbol{V}$ that needs to be updated. The update procedure outputs the updated versions of the components. The downdate procedure receives the components of the previous SVD $\boldsymbol{U}$, $\boldsymbol{S}$, $\boldsymbol{V}$ and outputs the downdated versions of the components after removing the last row of the matrix $\boldsymbol{A}$.

```
function [_U, _S, _V1] = svdUpdate
                        (U, S, V1, a);

    N = size(V1)(1);
    M = size(U)(1);
    D = S(:,1:M);

    z1 = V1'*a;
    w = a - V1*z1;

    [v,eta] = qr(w,0);

    L = [D, zeros(M,1);
         z1', eta];

    [Q,Omega,W] = svd(L);

    _U = [ U , zeros(M,1);
           zeros(1,M), 1 ] * Q;
     _S = Omega;
    _V1 = [V1, v]*W;
```

```
function [_U,_S,_V] = svdDowndate
                         (U,S,V);

    N = size(V)(1);
    M = size(U)(1);

    mu = U(M,M);

    if(mu < 0)
       U = -1*U;
       V = -1*V;
       mu = -1*mu;
    end

    D    = S(:,1:M);
    d    = S(M,M);
    U_11= U(1:M-1,1:M-1);
    x    = U(1:M-1,M);
    u1   = U(M,1:M-1)';

    MM   = eye(M-1) - (1/(1+mu))*u1*u1';
    X    = U_11*MM-x*u1';
    C    = [MM, -u1]*D;

    [Q,Omega,W] = svd(C);
    Omega    = Omega(:,1:M-1);

    _U   = X*Q;
    _S   = Omega;
    _V   =  (V*W)(:,1:M-1);
```

References

1. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Proceedings of European Conference on Computer Vision, vol. IV, pp. 25–36 (2004)
2. Bruhn, A., Weickert, J., Schnörr, C.: Lucas/Kanade meets Horn/Schunck: combining local and global optic flow methods. Int. J. Comput. Vis. **61**(3), 211–231 (2005)
3. Chan, A.B., Vasconcelos, N.: Layered dynamic textures. In: Weiss, Y., Schölkopf, B., Platt, J. (eds.) Advances in Neural Information Processing Systems, vol. 18, pp. 203–210. MIT Press, Cambridge (2006)
4. Chetverikov, D., Axt, A.: Approximation-free running SVD and its application to motion detection. Pattern Recognit. Lett. (2009, submitted)
5. Chetverikov, D., Fazekas, S.: On motion periodicity of dynamic textures. In: Proceedings of British Machine Vision Conference, vol. 1, pp. 167–176 (2006)
6. Chetverikov, D., Péteri, R.: A brief survey of dynamic texture description and recognition. In: 4th International Conference on Computer Recognition Systems (CORES'05), Springer Advances in Soft Computing, pp. 17–26 (2005)
7. Cooper, L., Liu, J., Huang, K.: Spatial segmentation of temporal texture using mixture linear models. In: LNCS, vol. 4358, pp. 142–150. Springer, New York (2007)
8. Crivelli, T., Cernuschi-Frias, B., Bouthemy, P., Yao, J.-F.: Mixed-state markov random fields for motion texture modeling and segmentation. In: Proceedings of International Conference on Image Processing (2006)
9. Dedeoglu, Y., Toreyin, B.U., Gudukbay, U., Cetin, A.E.: Real-time fire and flame detection in video. In: IEEE International Conference Acoustics, Speech, and Signal Processing, vol. 2, pp. 669–672 (2005)
10. Doretto, G., Cremers, D., Favaro, P., Soatto, S.: Dynamic texture segmentation. In: Ninth International Conference on Computer Vision, p. 1236 (2003)
11. Fazekas, S., Amiaz, T., Chetverikov, D., Kiryati, N.: Demo of dynamic texture detection and segmentation. http://vision.sztaki.hu/~fazekas/dtsegm/result/index.html (2007)
12. Fazekas, S., Amiaz, T., Chetverikov, D., Kiryati, N.: Dynamic texture detection based on motion analysis. Int. J. Comput. Vis. **82**, 48–63 (2009)
13. Fazekas, S., Chetverikov, D.: Dynamic texture recognition using optical flow features and temporal periodicity. In: Fifth International Workshop on Content-Based Multimedia Indexing (CBMI), pp. 25–32 (2007)
14. Free Software Foundation, Inc: GNU Scientific Library (GSL), version gsl-1.10. www.gnu.org/software/gsl/ (2007)
15. Ghoreyshi, A., Vidal, R.: Segmenting dynamic textures with ising descriptors, ARX models and level sets. In: LNCS, vol. 4358, pp. 121–141. Springer, New York (2007)
16. Gu, M.: Studies in numerical linear algebra. PhD thesis, Yale University, New Haven (1993)
17. Hall, P., Marshall, D., Martin, R.: On adding and subtracting eigenspaces with EVD and SVD. In: Proceedings of British Machine Vision Conference, pp. 453–462 (1999)
18. Horn, B.K.P.: Robot Vision. McGraw-Hill, New York (1986)
19. Horn, B.K.P., Schunck, B.G.: Determining optical flow. Artif. Intell. **17**, 185–203 (1981)
20. Hull, D., Nascimento, J., Ribeiro, P., et al.: Comparison of target detection algorithms using adaptive background models. In: Joint IEEE Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS), pp. 117–128 (2005)
21. Intel Corporation, Microprocessor Research Labs. OpenCV: Open Source Computer Vision Library. http://sourceforge.net/projects/opencvlibrary/ (2007)
22. Kahl, F., Hartley, R., Hilsenstein, V.: Novelty detection in image sequences with dynamic background. In: LNCS, vol. 3247, pp. 117–128. Springer, New York (2004)
23. Kanjilal, P.P., Bhattacharya, J., Saha, G.: Robust method for periodicity detection and characterization of irregular cyclical series in terms of embedded periodic components. Phys. Rev. E **59**, 4013–4025 (1999)
24. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: International Joint Conference on Artificial Intelligence, pp. 674–679 (1981)
25. Mileva, Y., Bruhn, A., Weickert, J.: Illumination-robust variational optical flow with photometric invariants. In: Lecture Notes in Computer Science, vol. 4713, pp. 152–162. Springer, New York (2007)
26. Monnet, A., Mittal, A., Paragios, N., Ramesh, V.: Background modeling and subtraction of dynamic scenes. In: Proceedings of International Conference on Computer Vision, vol. 2, pp. 1305–1312 (2003)
27. MUSCLE Network of Excellence: Multimedia Understanding through Semantics, Computation and Learning. http://muscle.ercim.org/ (2005)
28. Otsu, N.: A threshold selection method from gray level histograms. IEEE Trans. Syst. Man Cybern. **9**, 62–66 (1979)

29. Péteri, R., Huskies, M., Fazekas, S.: DynTex: a comprehensive database of dynamic textures. www.cwi.nl/projects/dyntex/ (2005)
30. Porikli, F., Tuzel, O.: Human body tracking by adaptive background models and mean-shift analysis. In: IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (2003)
31. Press, W., Teukolsky, S., Vetterling, W., Flannery, B.: Numerical Recipes in C. 2nd edn. Cambridge University Press, Cambridge (1992)
32. Radke, R.J., Andra, S., Al-Kofahi, O., Roysam, B.: Image change detection algorithms: a systematic survey. IEEE Trans. Image Process. **3**, 294–307 (2005)
33. Rahman, A., Murshed, M.: Multiple temporal texture detection using feature space mapping. In: Sixth ACM International Conference on Image and Video Retrieva, pp. 417–424 (2007)
34. Rahman, A., Murshed, M.: Temporal texture characterization: a review. In: Studies in Computational Intelligence, vol. 96, pp. 291–316. Springer, New York (2008)
35. Shafer, S.A.: Using color to separate reflection components. Color Res. Appl. **10**(4), 210–218 (1985)
36. Stauffer, C., Grimson, W.E.L.: Learning patterns of activity using real-time tracking. IEEE Trans. Pattern Anal. Mach. Intell. **22**, 747–757 (2000)
37. Toreyin, B.U., Dedeoglu, Y., Cetin, A.E.: Wavelet based real-time smoke detection in video. In: European Signal Processing Conference (2005)
38. Toreyin, B.U., Dedeoglu, Y., Gudukbay, U., Cetin, A.E.: Computer vision based method for real-time fire and flame detection. Pattern Recognit. Lett. **27**, 149–158 (2006)
39. Zhong, J., Sclaroff, S.: Segmenting foreground objects from a dynamic textured background via a robust Kalman filter. In: Proceedings of International Conference on Computer Vision, vol. 1, pp. 44–50 (2003)

Author Biographies

Dmitry Chetverikov was born in 1952. He studied theoretical physics at the Faculty of Physics of the Moscow State (Lomonosov) University, Russia. He received his PhD and DSc degrees from the Hungarian Academy of Sciences in 1988 and 2004, respectively. Currently, he is a Scientific Adviser and head of Geometric Modelling and Computer Vision Lab at the Computer and Automation Research Institute (SZTAKI), Budapest, Hungary. Dmitry is also a Professor at the Eotvos Lorand University in Budapest. He is the author or co-authors of about 150 research papers. His research interests include texture, motion and shape analysis, and scene reconstruction from multiple views.

Sándor Fazekas was born in 1975. He studied Physics and later Computer Science and Mathematics at the Eotvos Lorand University in Budapest, Hungary. He received his PhD degree in Physics from the Budapest University of Technology and Economics in 2008. For over 5 years Sandor was with the Geometric Modelling and Computer Vision Analysis Lab of the Computer and Automation Research Institute doing research in Digital Video Processing and Dynamic Texture Analysis. Currently, he is with the Budapest division of Ericsson.

Michal Haindl graduated in control engineering from the Czech Technical University, Prague, in 1979. He received his PhD in technical cybernetics from the Czechoslovak Academy of Sciences (1983) and the ScD (DSc) degree from the Czech Technical University (2001). He is an IAPR Fellow and a senior member of the IEEE. Michal is head of the Pattern Recognition department at the Institute of Information Theory and Automation (UTIA). He is the author or co-authors of about 250 research papers. His current research interests are in pattern recognition applications of random fields, image processing, and automatic acquisition of virtual reality models.