Pattern Recognition Letters 33 (2012) 1710–1716

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

# Space–time spectral model for object detection in dynamic textured background

Imtiaz Ali [a,b,d,*], Julien Mille [a,c], Laure Tougne [a,b]

[a] Université de Lyon, CNRS, France
[b] Université Lyon 2, LIRIS, UMR5205, F-69676, France
[c] Université Lyon 1, LIRIS, UMR5205, F-69622, France
[d] Optics Laboratories, P.O. 1021, Islamabad, Pakistan

ARTICLE INFO

*Article history:*
Received 28 November 2011
Available online 21 June 2012

Communicated by G. Borgefors

*Keywords:*
Background model
Local Fourier transform
Dynamic texture
Object detection

ABSTRACT

Background models are used for object detection in many computer vision algorithms. In this article, we propose a novel background modeling method based on frequency for spatially varying and time repetitive textured background. The local Fourier transform is applied to construct a pixel-wise representation of local frequency components. We apply our method for object detection in moving background conditions. Experimental results of our frequency-based background model are evaluated both qualitatively and quantitatively.



## 1. Introduction

Unknown object detection in complex backgrounds is one of the main tasks in automated surveillance systems. It is often the primary task before higher level analysis, like event understanding process. In fixed camera surveillance systems, background subtraction techniques are commonly used for this purpose. Low computational costs and no requirements of *a priori* knowledge of target objects are two prominent features which make this technique widely popular in the computer vision community. The methods often use background models, most of them being probabilistic representations of the background process. The probability of background color/intensity is usually independently modeled at each pixel. These background models work well in applications with limited background perturbations. However, in some cases, background environment is composed of repetitive moving objects, for example, water ripples, moving vegetation in the wind, fire, moving escalators *etc*. We address object detection in videos in which background is composed of time repetitive textures which are observable visually. In such conditions, individual pixel-based background models are not able to represent these regional changes. Similarly, standard background models do not take into account the temporal evolution of background regions. Thus, these algorithms produce a lot of false detections when they are applied in repetitively moving background conditions.

For spatial textures with time extent, one of the methods is to consider these patterns as time series, which has been referred to as dynamic texture (Doretto et al., 2003) in the literature. However, working with videos that contain textures of unknown spatiotemporal extent is different from working with static textured images. Dynamic textures have proven the interest of considering patterns in the (2D+T) space. Our method is inspired from frequency based 2D texture segmentation. We propose a pixel-wise background model based on local spectral analysis, that captures the frequency components of a spatiotemporal region around each pixel. We propose to use local Fourier transform on neighborhoods of pixels (both spatial and temporal). We construct the background model based on the observations of the background process during a training period. Once the background model is constructed, then object detection is performed on incoming frames. To our knowledge, spatiotemporal frequency analysis has not been explored for background modeling in the literature.

This paper is organized as follows. Previous work on non-stationary background modeling is presented in Section 2. Local space–time Fourier transform and the method of scene modeling are presented in Sections 3. We present object detection by using the background model in Section 4. We apply our background model on different videos from DynTex database (Péteri et al., 2010). Also, we compare our results with the GMM background model (Stauffer and Grimson, 2000) both qualitatively and quantitatively in Section 5. In Section 6, we conclude and present future ways of research linked with the proposed method.

* Corresponding author at: Université Lyon 2, LIRIS, UMR5205, F-69676, France.
*E-mail addresses:* imtiaz.ali@liris.cnrs.fr (I. Ali), julien.mille@liris.cnrs.fr (J. Mille), laure.tougne@liris.cnrs.fr (L. Tougne).

0167-8655/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2012.06.011

## 2. Related work

In fixed camera situations, the object detection can be done by learning background process and forming a statistical representation of underlying physical phenomena. Background subtraction is performed pixel-wise, comparing each new frame with the background model.

A recent comparative survey of background subtraction techniques is presented by Brutzer et al. (2011). They evaluate the results obtained with various background models. Among these techniques, the Gaussian mixture model (GMM) (Stauffer and Grimson, 2000) is an example of parametric background model which is commonly used for object detection. In this background modeling technique, color distribution of a pixel is assumed to be represented by a mixture of $K$ normal density distributions.

Conversely, in non-parametric background models like (Elgammal et al., 2002) density functions are estimated directly from the data without any assumption being made about underlying distributions. They use intensity histogram of fixed Gaussian kernels for modeling the probability density at a given pixel.

We can also find a codebook-based background model by Kim et al. (2005), that uses the appearance of a pixel value in the image sequence. It is a quantization technique that uses long scene observations for each pixel. One or several codewords are stored in the codebook for each pixel. The number of codewords for a pixel depends on the background intensity variations.

A last well-known method is the VuMeter method by Goyat et al. (2006). It is a non parametric model, based on a discrete estimation of the probability distribution, using color histograms for each pixel. They estimate the likelihood of the current pixel value to belong to background.

However, the previous pixel-based background models are not suited to moving background scenarios. As a matter of fact, pixel based background models consider each pixel independently. These methods neither take into account spatial neighborhoods of pixels for background modeling nor frequency of colors.

Spatial neighborhoods of pixels are often considered in texture analysis methods. In this context, a texture based background model is proposed by Heikkilä and Pietikäinen (2006). The method works in the spatial domain and for each pixel, 8 neighborhood pixels are considered for background modeling. They propose to use local binary pattern (LBP) as texture operator that relates to the earlier work by Ojala et al. (1996, 2002). This approach gives better background representation compared to pixel based approaches. But, it does not work very robustly on flat image areas where the gray values of the neighboring pixels are very close to the value of the center pixel. Similarly, LBP is strictly in spatial domain and does not take into account temporal evolution of background region which may change local texture temporally, therefore, gives poor results in case of time repetitive moving background.

However, spatiotemporal approaches have been proposed to address the problem of dynamic textures. An earlier work by Szummer and Picard (1996) focuses on temporal texture modeling. They proposed spatiotemporal auto regressive model (STAR) for temporal textures recognition. STAR is a three dimensional extension of autoregressive models. It works on entire image and models the image sequence as time series. It imposes a neighborhood causality constraint even for the spatial domain. The method has been modified by incorporating spatial correlation without imposing causal restrictions by Doretto et al. (2003, 2006). A probabilistic generative model using a mixture of dynamic textures for clustering and segmentation is presented in (Chan and Vasconcelos, 2008, 2010). In this method, an EM algorithm is derived for maximum-likelihood estimation of the parameters of a dynamic texture mixture and video segmentation is availed through the clustering of spatiotemporal patches. In (Ravichandran et al., 2009), the authors use bags of features relying on linear dynamic systems to handle texture motion. To model spatiotemporal variation in dynamic textures, a Fourier phase based method is proposed by Ghanem and Ahuja (2007), which captures the phase changes in dynamic texture over time. To detect foreground objects in a dynamic textured background, an approach has been presented by Zhong and Sclaroff (2003), that uses auto regressive moving average (ARMA) model. They proposed a robust Kalman filter to iteratively update the state of the dynamic texture ARMA model. If the estimated value for a pixel is different from the predicted value then the pixel is labeled as foreground.

The main idea of our approach, inspired by these last articles, is to model the spatiotemporal color patterns for object detection. In moving backgrounds, these color patterns appear repeatedly with time. A background model can be built on the frequency analysis in such dynamic textured background. The use of image frequency analysis for texture segmentation is common in computer vision. For example, Gabor transform is used for texture segmentation (Bovik et al., 1990). It is essentially a Fourier transform windowed by a Gaussian envelope. To select appropriate Gabor filters, the power spectrum analysis of Fourier transform of the textured image is performed (Manjunath and Ma, 1996; Puzicha et al., 1997; Wang et al., 2006). Local Fourier transform in spatial domain is applied by Zhou et al. (2001) for texture classification and content based image retrieval. In (Abraham et al., 2005) dynamic texture synthesis is carried out by using Fourier descriptors. They apply 2D Fourier transform on the whole image and the most significant frequencies that are contributed by all pixels are retained. In their approach, they assume temporal stationarity of spatial 2D textures. Therefore, they do not consider temporal evolution of spatial texture.

Variations in the background are both spatial and temporal in case of moving background. Therefore, the background model should be constructed by using spatiotemporal data in the region around a pixel by applying frequency analysis. The following section describes our proposed background model.

## 3. Scene modeling based on space–time local Fourier transform

First, we describe some abbreviations which are used for a pixel representation in space–time and frequency domain. Let a pixel in space be represented by $\boldsymbol{x} = (x, y)$ and in space–time by $\boldsymbol{p} = (\boldsymbol{x}, t)$. Let $\boldsymbol{u} = (u, v, w)$ be a space–time frequency vector. A spatiotemporal cuboid centered at a pixel is denoted as:

$$\Omega(\boldsymbol{p}) = \left[x - \frac{N_x}{2}, \ldots, x + \frac{N_x}{2}\right] \times \left[y - \frac{N_y}{2}, \ldots, y + \frac{N_y}{2}\right] \times \left[t - \frac{N_t}{2}, \ldots, t + \frac{N_t}{2}\right]$$

It is important to note that $N_x$, $N_y$ and $N_t$ should be chosen according to the maximal period (spatial and temporal, respectively) with no objects. Let us consider a gray scale image sequence as a real-valued function $f(\boldsymbol{p})$ defined for each pixel $\boldsymbol{p}$. Let us introduce a complex-valued function $\widehat{F}(\boldsymbol{u}, \boldsymbol{p})$, corresponding to space–time local Fourier transform for a pixel $\boldsymbol{p}$ given frequency $\boldsymbol{u}$. It is expressed as:

$$\widehat{F}(\boldsymbol{u}, \boldsymbol{p}) = \sum_{\boldsymbol{p}' \in \Omega(\boldsymbol{p})} f(\boldsymbol{p}')\omega(\boldsymbol{p} - \boldsymbol{p}')e^{-i2\pi((\boldsymbol{p}-\boldsymbol{p}')\cdot\boldsymbol{u})} \quad (1)$$

where

$$\omega(x, y, t) = \frac{1}{\sqrt{2\pi\sigma_x^2\sigma_y^2\sigma_t^2}} e^{\left(-\frac{x^2}{2\sigma_x^2} - \frac{y^2}{2\sigma_y^2} - \frac{t^2}{2\sigma_t^2}\right)}$$

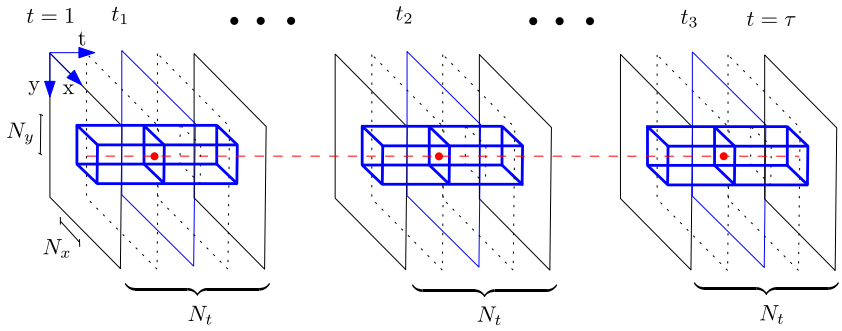


**Fig. 1.** An example of sequence containing $\tau$ images for learning background. Three spectrum feature vectors $n = 3$ are learned at time instants $t_1$, $t_2$ and $t_3$ during the training period (*i.e.* $t = 1, \ldots, \tau$). The location of pixel is represented by red dots in spatiotemporal window $\Omega = N_x \times N_y \times N_t$.

is the gaussian window function which is truncated beyond 3 times the standard deviation in each dimension. We chose $\sigma_x = \frac{N_x}{6}$, such that $\omega$ is negligible when $x = \pm\frac{N_x}{2}$ and similarly for $\sigma_y$ and $\sigma_t$. In our method, we take the magnitude of Fourier coefficients which have information of the quantity of each frequency component present inside spatiotemporal cuboid $\Omega$ around the pixel $\boldsymbol{p}$. We denote it as a spectrum $\mathcal{S}(\boldsymbol{u}, \boldsymbol{p})$. This can be expressed as:

$$\mathcal{S}(\boldsymbol{u}, \boldsymbol{p}) = |\widehat{F}(\boldsymbol{u}, \boldsymbol{p})| \tag{2}$$

The space–time local Fourier transform produces $N_x \times N_y \times N_t$ frequency components. A spectrum feature vector is constructed for the pixel $\boldsymbol{p}$, by concatenating the Fourier coefficient values in an 1D vector as:

$$\boldsymbol{v}(\boldsymbol{p}) = [\mathcal{S}(\boldsymbol{u}_1, \boldsymbol{p}), \mathcal{S}(\boldsymbol{u}_2, \boldsymbol{p}), \ldots, \mathcal{S}(\boldsymbol{u}_M, \boldsymbol{p})] \tag{3}$$

where

$$M = N_x \times N_y \times N_t$$

For color images, we compute the local Fourier transform independently on each channel values. For a given pixel, three spectra are concatenated in $\boldsymbol{v}(\boldsymbol{p})$ (in this case, $M = 3N_x \times N_y \times N_t$). The background learning process is as follows. We take the spatiotemporal input data from $\tau$ learning images to compute local Fourier transform. We learn $n$ spectra per pixel during this training period. The $i$th learned spectrum vector is:

$$\boldsymbol{v}^i_{\text{background}}(\boldsymbol{x}) = \boldsymbol{v}(\boldsymbol{x}, t_i) \quad \forall i = 1, \ldots, n$$

Frequency background model in space can be expressed as the set of learned spectrum vectors:

$$\mathcal{M}(\boldsymbol{x}) = \left\{ \boldsymbol{v}^i_{\text{background}}(\boldsymbol{x}) \right\}_{i=1,\ldots,n}$$

Fig. 1 shows the space–time neighborhoods over which training spectra are computed (in this example, $n = 3$).

To learn the dynamic temporal texture in the background, the parameter $N_t$ is crucial. If we have small period of temporal texture repetition (*i.e.* fast background motion) in an application then we can limit ourselves to use a small value for $N_t$. Otherwise, repetitive motions with large periods (*i.e.* slow background motion) need a high value of $N_t$ to be captured. The remarks are also valid for $N_x$ and $N_y$ (*i.e.* slow and fast varying background in space can be modeled with large and small values of these parameters, respectively).

## 4. Object detection

For object detection, we buffer a set of $N_t$ incoming frames in the memory. We take spatiotemporal data around each pixel of this set of incoming frames as explained in Section 3. We compute a spectrum vector for each pixel, by applying Eq. (2) and (3) on current image data of $N_t$ frames.

For each pixel $\boldsymbol{p}$, the current spectrum feature vector $\boldsymbol{v}(\boldsymbol{x}, t)$ is compared with the set of $n$ background learned spectrum feature vectors. Let $d$ be the dissimilarity function between $\boldsymbol{v}(\boldsymbol{x}, t)$ and the model associated to pixel $\boldsymbol{x}$, namely $\mathcal{M}(\boldsymbol{x})$. We can write mathematically as:

$$d((\boldsymbol{x}, t), \mathcal{M}(\boldsymbol{x})) = \min_{i=1,\ldots,n} \mathcal{D}(\boldsymbol{v}(\boldsymbol{x}, t), \boldsymbol{v}^i_{\text{background}}(\boldsymbol{x})) \tag{4}$$

where $\mathcal{D}$ is a distance function between two spectrum feature vectors. We choose to define it by:

$$\mathcal{D}(\boldsymbol{v}, \boldsymbol{v}_{\text{background}}) = \left\| \boldsymbol{v} - \boldsymbol{v}_{\text{background}} \right\|^2$$

High value of the distance measure $d$ leads to the following interpretation: the current spectrum vector $\boldsymbol{v}(\boldsymbol{x}, t)$ is not close to any of the $n$ learned spectrum vectors of training sequence, and may corresponds to an object pixel in the scene. In other words, current spectrum vector is composed of frequencies which do not exist in the background. We can consider a pixel $\boldsymbol{x}$ as a moving object pixel if $d$ is greater than a threshold $\epsilon$. Therefore, a foreground image $\mathcal{F}(\boldsymbol{x}, t)$ is produced by using the following equation:

$$\mathcal{F}(\boldsymbol{x}, t) = \begin{cases} 1 & \text{if} \quad d((\boldsymbol{x}, t), \mathcal{M}(\boldsymbol{x})) \geqslant \epsilon \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

In this way the perturbations in the scene, apart from the spatially varying and time repetitive textures, are identified and used for object detection.

In the next section, we analyze the proposed frequency based model and show the relevance of the model in the particular case of rivers.

## 5. Background spectral analysis and object detection results

The background representation using frequency analysis requires some further explanation and needs to be clarified with examples. We use a video containing a floating object in a river to illustrate our method. We show that using frequency analysis, the discrimination between different background regions and moving objects can be obtained. Two background pixels $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are marked in an image from the video (see Fig. 2). An object passes through the pixel $\boldsymbol{x}_2$ in the water region. We take spatiotemporal region $N_x \times N_y \times N_t = 5 \times 5 \times 3$ and $n = 8$ for respective points in the video.

### 5.1. Background spectral analysis

We expect the spectra to be distant between different points in the image and also during an object passage. We can remark that

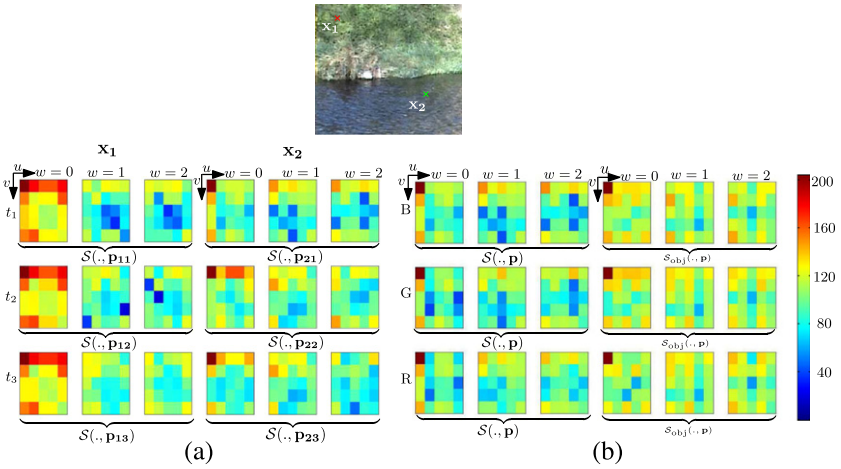


**Fig. 2.** Graphical representation of values of local Fourier transform coefficients: (a) $\mathcal{S}(.,\boldsymbol{p}_{1t})$ and $\mathcal{S}(.,\boldsymbol{p}_{2t})$ represent spectra for the B channel at spatiotemporal locations $(\boldsymbol{x}_1,t)$ and $(\boldsymbol{x}_2,t)$ at time $t=t_1,\ t_2,\ t_3$ and (b) Two spectra $\mathcal{S}(.,\boldsymbol{p})$ and $\mathcal{S}_{\text{obj}}(.,\boldsymbol{p})$ for Blue (B), Green (G) and Red (R) color channels of spatial point $\boldsymbol{x}_2$ for background and object passage, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the spatiotemporal frequencies of background pixels contain similar Fourier coefficient values. We represent these values for visual comparison in Fig. 2(a). The magnitude values of local Fourier transform of the two pixel locations ($\boldsymbol{x}_1$ and $\boldsymbol{x}_2$) are shown in the frequency domain. For this data, we found that the spectra of the three RGB components were similar (color saturation is relatively low, causing colors to be located near the black and white axis as shown in the histogram of Fig. 6). Therefore, we show only one spectrum $\mathcal{S}(\boldsymbol{u},\boldsymbol{x})$ at multiple time instances. For $\boldsymbol{p}_{1t}=(\boldsymbol{x}_1,t)$ and $\boldsymbol{p}_{2t}=(\boldsymbol{x}_2,t)$, we show, the three spectra $\mathcal{S}(\boldsymbol{u},.)$ at time $t=t_1,\ t_2$ and $t_3$. First three columns represent spectra $\mathcal{S}(\boldsymbol{u},\boldsymbol{p}_{1t})$ and last three columns represent spectra $\mathcal{S}(\boldsymbol{u},\boldsymbol{p}_{2t})$ in Fig. 2(a). It must be noted that the time instances (*i.e.* $t_1,\ t_2$ and $t_3$) are not consecutive in time. Furthermore, the Fourier coefficient values are normalized by using logarithmic transformation such that the values remain in the range from 0 to 255. Two prominent properties are highlighted here. The first one is that local Fourier coefficient values of the corresponding frequencies within a spatiotemporal region are similar at different time instances with few variations. Therefore, it implies that the values of local Fourier transform can be used as a feature for background modeling. The second property is that the values of local Fourier transform are dissimilar for two different regions.

We also present the analysis of spatiotemporal frequency components in case of object motion. In the river video, a floating object passes through the pixel $\boldsymbol{x}_2$. To illustrate the effects of object passage on the spatiotemporal frequencies, we show 3 spectra $\mathcal{S}(\boldsymbol{u},\boldsymbol{p})$ for RGB color channels at $\boldsymbol{x}_2$. First three columns in Fig. 2(b) represent the spectrum $\mathcal{S}(\boldsymbol{u},\boldsymbol{p})$ at $\boldsymbol{x}_2$ with only background. Last three columns in Fig. 2(b) show the spectra $\mathcal{S}_{\text{obj}}(\boldsymbol{u},\boldsymbol{p})$ at the same spatial position during the object passage through the point. For these spatiotemporal positions, the coefficient values of the two respective spectra are different. We can observe an increase of the corresponding spatiotemporal frequencies values. This difference is used for object detection.

### 5.2. Projection into discriminative subspace

Since the magnitudes of neighboring frequencies are highly related, we expect to have high correlation between several components of the feature vectors. This leads us to study the relevance of our feature space using dimensionality reduction technique. We use Fisher linear discriminant analysis (LDA), in order to project the feature points onto a subspace that maximizes interclass variance while minimizing intraclass variance. We present the results of discriminant analysis applied to the video with a moving object and two background pixels $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ in Fig. 3. The projected data is clustered into distinct areas when projected onto the first three eigenvectors ($\boldsymbol{v_1}$, $\boldsymbol{v_2}$ and $\boldsymbol{v_3}$). This Figure shows that it will be possible, using these features, to distinguish the different color patterns of the respective pixels.

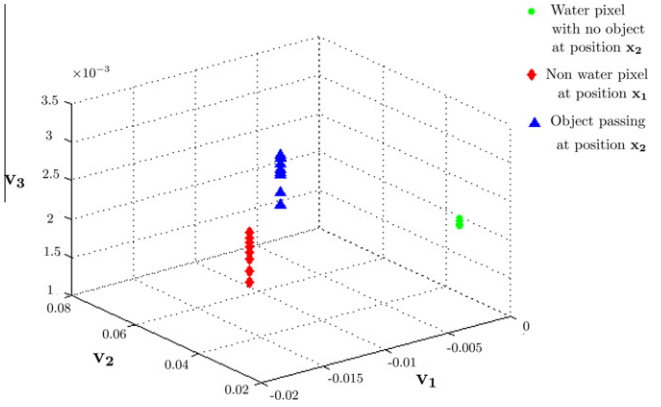


**Fig. 3.** A subspace linear discriminant analysis (LDA) for the two background pixels $\boldsymbol{x}_1,\ \boldsymbol{x}_2$ (Fig. 2) and an object pixel with spatiotemporal region $N_x\times N_y\times N_t = 5\times 5\times 3$ at time $t$, data is projected onto first 3 eigenvectors.

### 5.3. Object detection results

We present our experiments on both synthetic and real natural videos. We use three videos from the DynTex database (Péteri et al., 2010), which contains multiple videos with dynamic textures. Apart from the database, we also test our algorithm on a video of floating objects in a river, which we have made.

But first, let us explain the effects of changing various model parameters on the escalator video (extracted from Dyntex database) in detail. In this video, an escalator moves from top to bottom in the image plane. The motion is an example of dynamic texture with large temporal extent. Our method is composed of two steps, background learning and object detection. We use $\tau$ images from the video for background learning. In this experiment, we change both spatial and temporal neighborhoods per pixel in order to show the effects of these parameters. Number of spectra per pixel

$n$ is a user-defined parameter and we fix $n = 8$ in our experiments. The original video does not contain any object to detect. Therefore, we move synthetically a $30 \times 30$ square portion of escalator as an object. Motion of this square block is from left to right in the image plane. The block is simultaneously translated and rotated with an angle of 5 degrees clockwise per image in 100 consecutive images. It is important to note that we use different sets of images for training and detection.

We show the final foreground image, that is obtained by using Eq. (5), with the corresponding parameters in Fig. 4. We show one foreground image of the image sequence. The effects of changing size of the spatiotemporal neighborhood can be observed. We use odd values from $1 \times 1$ to $7 \times 7$ for $N_x \times N_y$. When $N_x \times N_y = 1 \times 1$, implies that no spatial neighborhood per pixel is considered, which boils down to extracting purely temporal patterns. The results of these values of the parameters are shown in the first row of Fig. 4. Similarly, we vary the value of $N_t$ per pixel from 1 to 11. We show the results of our method when only spatial neighborhoods are used (*i.e.* $N_t = 1$) in the first column of Fig. 4. The escalator motion in the background is slow and increasing

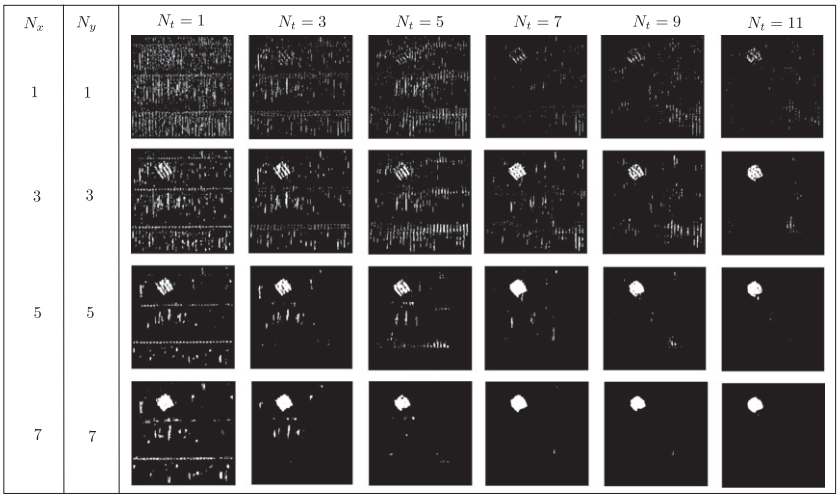


**Fig. 4.** Foreground image results with different model parameters of spatiotemporal region $N_x \times N_y \times N_t$ per pixel for a moving square block with moving escalator in the background.

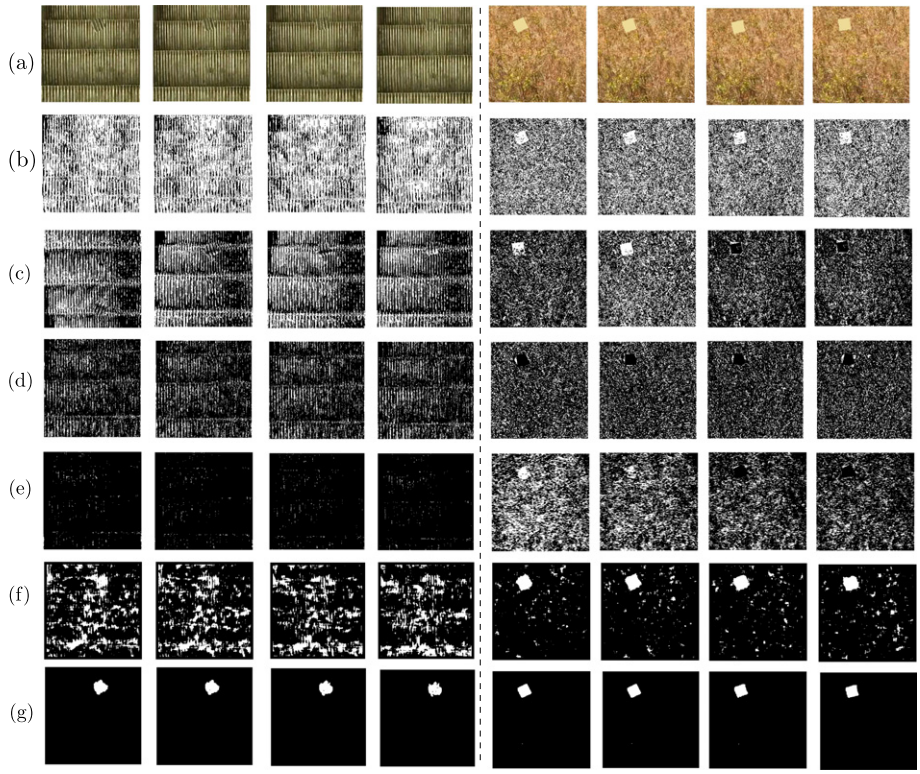


**Fig. 5.** (a) A synthetic square of $30 \times 30$ pixels moving from left to right with (*left*) moving escalator and (*right*) moving wheat field in the background. Results of (b) frame differencing, (c) approximate median filtering (McFarlane and Schofield, 1995), (d) the GMM (Stauffer and Grimson, 2000), (e) the Vumeter (Goyat et al., 2006), (f) the LBP (Heikkilä and Pietikäinen, 2006) and (g) our method. Computation time during training period for escalator video is 193.70 s with $7 \times 7 \times 11$ and for wheat video 89.76 s with $5 \times 5 \times 5$, detection time per frame are 15.4 s and 9.47 s, respectively.

the value of $N_t$ enables the background model to capture the periodicity of moving escalator. The optimal value of spatiotemporal region $N_x \times N_y \times N_t$ per pixel is $7 \times 7 \times 11$ for this video.

We also test the method on another video of moving wheat field from DynTex (Péteri et al., 2010). In this video, there are continuous motions in the background. As in the escalator experiment, we move a square block of the same size synthetically over the original images. For both videos, we test frame differencing (FD), approximated median filtering (AMD) (McFarlane and Schofield, 1995) as well as three existing background modeling (the GMM) (Stauffer and Grimson, 2000), the VuMeter (VM) (Goyat et al., 2006) and LBP-based method (Heikkilä and Pietikäinen, 2006). The results for the corresponding images with these methods are shown in Fig. 5. Existing methods produce lower detection rates than our model. We show that the frequency-based background model can be used to detect an object even if it has similar colors as background. The poor results of LBP-based in these dynamic texture applications is due to the fact that the method remains local in space. Therefore, LBP-based method may not capture temporal extent of textures.

In another application, we consider a river background, in which water ripples have region-wise temporal texture (see Fig. 6). The color histograms show 100 consecutive color values of two regions of $4 \times 4$ pixels that are highlighted by squares in the top row of Fig. 6. The green leaves in the surroundings of river contain repetitive textures from green to light green. The pixels in the water region contain almost all intermediate values between black and white as shown in Figure. We can see that pixels values have a wide distribution especially in the aquatic region. The color histograms reveal the fact that pixel-based background models such as GMM (Stauffer and Grimson, 2000) will not be able to model correctly in such conditions. The distributions tell us about the color diversity, however, the temporal variations of pixel values in successive frames are not evident from the histograms.

We apply our background model to a video of floating objects in a river. The quasi-periodic changes which occur in background regions are learned during the first $\tau$ frames with no objects. In this application, the optimal results are obtained with a spatiotemporal neighborhood size of $N_x \times N_y \times N_t = 5 \times 5 \times 5$. Results of our method and concurrent algorithms are shown in the left part of Fig. 6. We can notice that frame differencing, AMD and the GMM results contain many false detected background pixels. The results of the Vumeter and LBP method are slightly better than the GMM. The results show that our frequency based background model is able to capture the dynamic changes in the aquatic region, waving grass and leaves in the background.

Finally, we test our model on another video from the Dyntex database (Péteri et al., 2010). Here, the aquatic background contains water ripples and dark cast shadows of the surroundings. A duck enters the scene from the top right corner and moves across the scene to the middle of the image plane. When the duck moves under the cast shadows, it shares the same color with the background. Results are shown in the right part of Fig. 6. We can notice that existing methods generate many false detections and miss

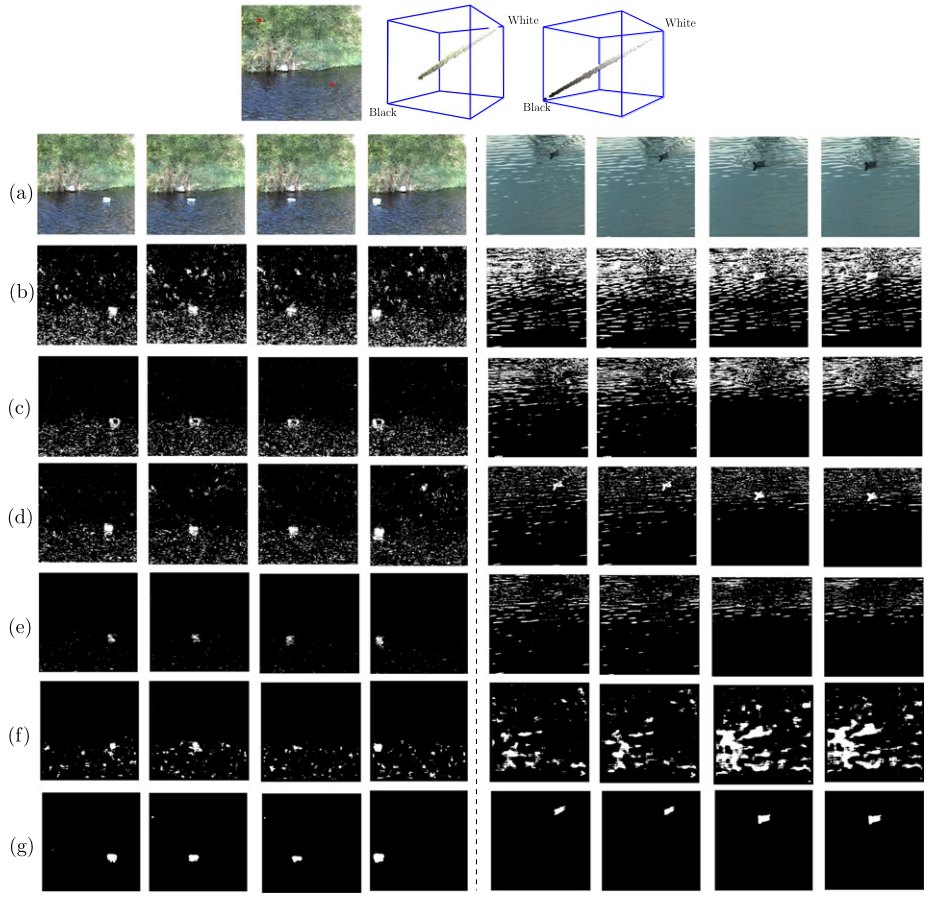


**Fig. 6.** *(Top)* An image from river videos where two regions of $4 \times 4$ pixels outside water and in water are highlighted with histograms of such regions, respectively. (*Bottom*) (a) Original images from a floating bottle and moving duck videos with corresponding results of (b) frame differencing, (c) approximate median filtering (McFarlane and Schofield, 1995), (d) the GMM (Stauffer and Grimson, 2000), (e) the Vumeter (Goyat et al., 2006), (f) the LBP (Heikkilä and Pietikäinen, 2006) and (g) our method. Computation time during training period for bottle is 89.76 s with $5 \times 5 \times 5$ and for duck is 124.04 s with $5 \times 5 \times 11$ and detection time per frame are 15.4 s and 9.47 s, respectively.

**Table 1**
Quantitative comparison of Dice similarity measure of proposed background model and the GMM for four videos.

| | Escalator | Wheat | Duck | Bottle |
|---|---|---|---|---|
| FD | 0.04 | 0.05 | 0.02 | 0.06 |
| AMF | 0.06 | 0.03 | 0.01 | 0.07 |
| GMM | 0.03 | 0.01 | 0.11 | 0.14 |
| VM | 0.10 | 0.07 | 0.05 | 0.30 |
| LBP | 0.23 | 0.58 | 0.01 | 0.45 |
| Our method | 0.87 | 0.96 | 0.81 | 0.75 |

many foreground pixels. Moreover, many parts of the foreground object are mis-classified as the background. The results of our frequency based background model show that not only we have minimum false detections but also the foreground object has very few mis-detected pixels. It is important to mention that we do not use any morphological operations. In this application, we use spatiotemporal neighborhood of $N_x \times N_y \times N_t = 5 \times 5 \times 11$ per pixel.

Image segmentation results of our approach and existing methods are evaluated with the Dice similarity measure, which is a commonly used measure of segmentation quality (Cárdenes et al., 2008). It is expressed as

$$S = \frac{2|X \cap Y|}{|X| + |Y|} \tag{6}$$

where $X$ and $Y$ are the sets of object pixels in the generated segmentation and in th ground truth image, respectively. $S$ is equal to 1 when the segmented region and the ground truth region perfectly overlap, and 0 when they are disjoint. Ground truth images are available for synthetic motions in the escalator and the wheat videos. For other videos, we manually obtained ground truth images. For this purpose, we randomly select (15%) images per video for the two videos. The average Dice coefficient values obtained with each method are shown in Table 1. Image segmentation results obtained with frame difference, approximate median filtering, the GMM, the VuMeter and LBP method have very low dice values for all videos. High Dice values are obtained with our background model. We can remark that in the bottle video, the average Dice value is smaller than in other cases due to the reflection in water that create some false detections. Image segmentation results indicate strong superiority of the frequency-based background model for object detection over existing methods in dynamic textured and moving background. Indeed, GMM and VuMeter are only color based and do not take into account frequency of colors. The LBP-based background model is texture-based but relies on a purely 2D representation.

Finally, we give the computation time taken by our method during the training period and object detection. The image size is $256 \times 256$ and $n = 8$ for all videos. The method is tested on an Intel Core2 Duo 2.66 GHz with 4 GB RAM, running a C code. We mention the computation time taken during training period and detection time per frame in the legends with corresponding results in Figs. 5 and 6. One may note that the detection time taken by the other tested methods is in the order of 200 ms.

## 6. Conclusion

In this paper, we present a novel frequency based background model. Such model is dedicated to moving backgrounds containing repetitive structures. We consider spatiotemporal neighborhoods of the pixels in the scene, on which we apply local Fourier transform. The generated spectral feature vectors are used to build a background model. Spatially varying and time repetitive textures in the background regions are very efficiently modeled using the frequency-based method. We apply our method for moving object detection in moving backgrounds, on both synthetic and real image sequences. We obtain high accuracy in foreground/background segmentation and outperform classic and commonly used background models. In outdoor scenarios, our background model leads to better detection and segmentation than the existing methods, which fail to capture the time-repetitive background motions. As future work, we plan to include an update mechanism in the background model. Among other image phenomena, this could handle global brightness change through the image sequence.

## References


Abraham, B., Camps, O.I., Sznaier, M., 2005. Dynamic texture with Fourier descriptors. In: Internat. Workshop on Texture Analysis and Synthesis, pp. 53–58.

Bovik, A.C., Clark, M., Geisler, W.S., 1990. Multichannel texture analysis using localized spatial filters. IEEE Trans. Pattern Anal. Machine Intell. 12 (1), 55–73.

Brutzer, S., Höferlin, B., Heidemann, G., 2011. Evaluation of background subtraction techniques for video surveillance. IEEE Comput. Vision Pattern Recognition, 1937–1944.

Cárdenes, R., Bach, M., Chi, Y., Marras, I., de Luis Garca, R., Anderson, M., Cashman, P., Bultelle, M., 2008. Multimodal evaluation for medical image segmentation. In: Internat. Conf. on Computer Analysis of Images and Patterns, pp. 229–236.

Chan, A.B., Vasconcelos, N., 2008. Modeling, clustering, and segmenting video with mixtures of dynamic textures. IEEE Trans. Pattern Anal. Machine Intell. 30 (5), 909–926.

Chan, A.B., Coviello, E., Lanckriet, G., 2010. Clustering dynamic textures with the hierarchical EM algorithm. In: IEEE Internat. Conf. on Computer Vision and Pattern Recognition, pp. 2022–2029.

Doretto, G., Chiuso, A., Wu, Y.N., Soatto, S., 2003. Dynamic textures. Internat. J. Comput.Vision 51 (2), 91–109.

Doretto, G., Soatto, S., 2006. Dynamic shape and appearance models. IEEE Trans. Pattern Anal. Machine Intell. 28 (12), 2006–2019.

Elgammal, A., Duraiswami, R., Harwood, D., Davis, L.S., 2002. Background and foreground modeling using nonparametric kernel density for visual surveillance. Proceedings of the IEEE (90), 1151–1163.

Ghanem, B., Ahuja, N., 2007. Phase based modelling of dynamic textures. In: IEEE Internat. Conf. on Computer Vision, pp. 1–8.

Goyat, Y., Chateau, T., Malaterre, L., Trassoudaine, L., 2006. Vehicle trajectories evaluation by static video sensors. In: IEEE Internat. Conf. on Intelligent Transportation Systems, pp. 864–869.

Heikkilä, M., Pietikäinen, M., 2006. A texture-based method for modeling the background and detecting moving objects. IEEE Trans. Pattern Anal. Machine Intell. 28 (4), 657–662.

Kim, K., Thanarat, T., Chalidabbhognse, H., Harwood, D., Davis, L., 2005. Real time foreground-background segmentation using codebook model. Real-Time Imaging 11 (3), 172–185.

Manjunath, B., Ma, W., 1996. Texture features for browsing and retrieval of image data. IEEE Trans. Pattern Anal. Machine Intell. 18 (8), 837–842.

McFarlane, N.J.B., Schofield, C.P., 1995. Segmentation and tracking of piglets in images. Machine Vision Appl. 8 (3), 187–193.

Ojala, T., Pietikäinen, M., Harwood, D., 1996. A comparative study of texture measures with classification based on feature distributions. Pattern Recognition 29 (1), 51–59.

Ojala, T., Pietikäinen, M., Mäenpää, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Machine Intell. 24 (7), 971–987.

Péteri, R., Fazekas, S., Huiskes, M.J., 2010. DynTex: A comprehensive database of dynamic textures. Pattern Recognition Lett. 31 (12), 1627–1632.

Puzicha, J., Hofmann, T., Buhmann, J.M., 1997. Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 267–272.

Ravichandran, A., Chaudhry, R., Vidal, R., 2009. View-invariant dynamic texture recognition using a bag of dynamical systems. In: IEEE Internat. Conf. Computer Vision and Pattern Recognition, pp. 1651–1657.

Stauffer, C., Grimson, W., 2000. Learning patterns of activity using real-time tracking. IEEE Trans. Pattern Anal. Machine Intell. 22 (8), 747–757.

Szummer, M., Picard, R.W., 1996. Temporal texture modeling. In: IEEE Internat. Conf. on Image Processing, pp. 823–826.

Wang, H., Wang, X., Zhou, Y., Yang, J., 2006. Colour texture segmentation using quaternion-Gabor filters. In: IEEE Conf. on Image Processing, pp. 745–748.

Zhong, J., Sclaroff, S., 2003. Segmenting foreground objects from a dynamic textured background via a robust Kalman filter. In: Internat. Conf. on Computer Vision, pp. 44–50.

Zhou, F., Feng, J.F., Shi, Q.Y., 2001. Texture feature based on local Fourier transform. In: IEEE Internat. Conf. on Image Processing, vol. 2, pp. 610–613.