Dynamic Texture Recognition Using Multiscale Binarized Statistical Image Features

Shervin Rahimzadeh Arashloo and Josef Kittler, *Life Member, IEEE*

***Abstract*—A spatio-temporal descriptor for representation and recognition of time-varying textures is proposed [binarized statistical image features on three orthogonal planes (BSIF-TOP)] in this paper. The descriptor, similar in spirit to the well known local binary patterns on three orthogonal planes approach, estimates histograms of binary coded image sequences on three orthogonal planes corresponding to spatial/spatio-temporal dimensions. However, unlike some other methods which generate the code in a heuristic fashion, binary code generation in the BSIF-TOP approach is realized by filtering operations on different regions of spatial/spatio-temporal support and by binarizing the filter responses. The filters are learnt via independent component analysis on each of three planes after preprocessing using a whitening transformation. By extending the BSIF-TOP descriptor to a multiresolution scheme, the descriptor is able to capture the spatio-temporal content of an image sequence at multiple scales, improving its representation capacity. In the evaluations on the UCLA, Dyntex, and Dyntex + + dynamic texture databases, the proposed method achieves very good performance compared to existing approaches.**

***Index Terms*—Binarized statistical image features, multiresolution analysis, spatio-temporal descriptors, time-varying texture.**

I. Introduction

DYNAMIC textures are sequences of spatial patterns varying over time. They are typically image sequences of moving scenes exhibiting stochastic motion. Some readily observable instances of such visual phenomena in the natural world are sea waves, smoke, fire, swarms of birds, vegetation in the wind, etc. Normally, the changes in dynamic patterns are due to motion (e.g., swarms of birds, fluttering flag) but may also be the result of variations in the intensity of the emitted light (e.g., fire) (see Fig. 1). In literature, such patterns are referred to with different terminologies, including spatio-temporal textures, turbulent flow/motion, time-varying textures, dynamic textures, textured motion, etc. The terms spatio-temporal and dynamic textures will be used in this work interchangeably.

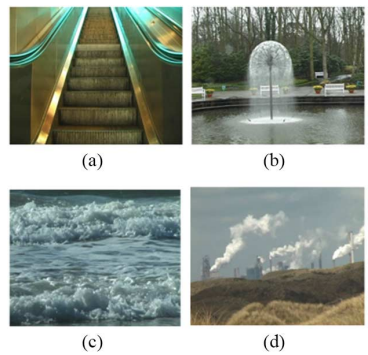


Fig. 1. Example frames of dynamic textures. (a) Escalator. (b) Fountain. (c) Sea. (d) Smoke.

Sometimes, the term dynamic texture is used for image sequences of natural processes with nondeterministic behavior; however, in this work it is assumed that it also applies to simpler patterns of events such as a moving escalator or a walking crowd and as a result it encompasses a larger set of time-varying textures.

Conventionally, the classification of textures was based on static cues only. However, it has been observed that in many situations static information was not sufficient and it is advantageous to employ temporal statistics in conjunction with spatial information where applicable and attempt to recognize a scene not only based on the statistics of a single image but on the statistics of an image sequence over time. It has been argued that in the perception of the world, the vision system in humans exploits motion in addition to supplementary information in other forms [1]. In particular, motion is known to be an influential sensory signal for visual recognition of objects and scenes. In this respect, Johansson's moving dot displays [2] illustrates that highly ambiguous objects from a single view become easily identified once motion is provided. The study of dynamic textures in computer vision as a topic ranging from dynamic texture modelling and synthesis to classification and recognition has emerged very recently. The focus of this paper is on the development of an effective representation and classification technique for time-varying textures. Unlike static patterns, dynamic textures not only depend on the spatial distribution of textons, but also on their dynamics over time. The main challenge in the study of time-varying textures is then how to effectively compute the properties of dynamics over a period of time.

Manuscript received March 15, 2014; revised July 29, 2014 and September 19, 2014; accepted October 05, 2014. Date of publication October 13, 2014; date of current version November 13, 2014. This work was supported in part by EPSRC/DSTL project Signal Processing in a Networked Battlespace under Contract EP/K014307/1 and the European Union project Beat. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Chia-Wen Lin.

S. R. Arashloo is with the Department of Electrical Engineering, Urmia University, Urmia 57135, Iran, and also with the Centre for Vision, Speech, and Signal Processing, Department of Electronic Engineering, University of Surrey, Guildford GU2 7XH, U.K. (e-mail: Sh.Rahimzadeh@hotmail.co.uk).

J. Kittler is with the Centre for Vision, Speech, and Signal Processing, Department of Electronic Engineering, University of Surrey, Guildford GU2 7XH, U.K. (e-mail: J.Kittler@surrey.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2014.2362855

The capability to recognize time-varying patterns is of significant importance to various applications of visual processing. In surveillance, the ability to discern dynamic patterns can serve to isolate activities of interest from background clutter. For example, one may need to know whether a fire has started or not, and if so, identify its location and extent. Traffic monitoring, homeland security, or studying animal behavior can be considered as other applications, just to name a few.

In the context of automatic face recognition [3]–[5], psychological and neural studies [6] indicate that both static and dynamic facial features and personal characteristics are appropriate for recognizing faces. It has also been found that the facial motion is the basis to the recognition of facial expressions. A further application of dynamic texture recognition is concerned with lip reading. It is known that human speech perception is a multimodal process where visual information of lips, teeth, and tongue offers important information about the place of pronunciation articulation. Description and recognition of dynamic patterns is also of practical importance in video indexing/retrieval, and registration and editing systems, which have attracted growing attention. In summary, the span of applications of time-varying textures is very wide, including video indexing/retrieval [7], surveillance [8], environmental monitoring [9], isolating background from activities of interest [10], detection of various emergency conditions such as fire [11], leaks in pipes, space-time stereo [12], grouping [13], [14], activity recognition [15], tracking [16], lip reading [17], etc.

There are different approaches proposed in the literature for representation and recognition of spatio-temporal textures. A line of research is based on the so-called physics-based approaches [18]. In these methods, by analyzing the underlying generating process, a model for the characterization of a specific dynamic pattern is constructed. Once the model parameters are recovered for an input pattern, they are used for recognition. A major disadvantage of these approaches is that the models are derived for specific patterns separately and thus suffer from poor generalization to other kinds of time-varying textures. A second group of approaches are the methods which directly use the motion field for recognition. Using the estimated instantaneous motion patterns in sequences, the flow-based method proposed in [19] converts the analysis of spatio-temporal sequences to the analysis of sequences of static information. In other works [20], [21], dynamic texture is represented using statistical measurements of optical flow information. Based on the velocity and acceleration fields estimated at various spatio-temporal scales, a metric for video sequences is defined in [22]. The main drawbacks of these methods include the high correlation of normal flow and dynamic texture spatial appearance in addition to the assumptions of brightness constancy and local smoothness which generally do not hold for stochastic dynamics scenes [23].

Apart from the physics-based methods, there are other methods for spatio-temporal texture characterization based on statistical generative representations to jointly model the appearance and dynamics of textures. Recognition in these methods is performed by comparing the estimated model parameters. Several variants of this approach include autoregressive (AR) models [24]–[27] and multiresolution schemes [28], the most popular one being the joint photometric-dynamic, AR-based linear dynamic system (LDS) model [26]. Although good performance has been reported in the literature for these methods, the performance of these approaches is highly dependent on the spatial appearance captured by these models rather than the underlying dynamics [29], [30]. Other methods [31] have addressed the variability in viewpoint in the LDS model within the bag-of-features framework. There are also some LDS variants which have discarded the appearance component of the model and have restricted the attention to the dynamic component for recognition purposes [29], [30]. Methods employing certain transforms such as wavelet and 3D-surfacelet also exist. One may cite, for example, the work in [32] which uses spatio-temporal wavelet transforms in order to decompose dynamic texture into different spatio-temporal scales.

A successful category of methods for dynamic texture categorization which do not explicitly model the underlying dynamic patterns is known as discriminative method [33], [34]. Using a number of training samples, different spatio-temporal filters are constructed in [33] which are specifically tuned for certain local dynamic pattern structures. The original local binary pattern (LBP) descriptor in 2D images proposed by Zhao and Pietikinen in [35] is extended to the 3D spatio-temporal volume based on local spatio-temporal statistics in [34]. To compare different descriptors efficiently, the co-occurrence of LBPs was computed in three orthogonal planes, known as the local binary patterns on three orthogonal planes (LBP-TOP) descriptor [34].

Other work in [36] addresses the problem of separating a video sequence into its two constituent layers. One layer corresponds to the video of the unoccluded background, and the other to that of the dynamic texture. For this purpose an approach that uses the image motion information to simultaneously obtain a model of the dynamic texture and separate it from the background which is required to be still is proposed. The frames of a sequence are modelled as being produced by a continuous hidden Markov model (HMM), characterized by transition probabilities based on the Navier-Stokes equations for fluid dynamics, and by generation probabilities based on the convex matting of the fluid dynamic texture (FDT) with the background.

The work in [37] presents a model of spatio-temporal variations in a dynamic texture sequence. In this work, a model is proposed which relates texture dynamics to the variation of the Fourier phase, that captures the relationships between the motions of all pixels within the texture, as well as the appearance of the texture. The proposed approach does not require segmentation or cropping during the training, which allows it to handle time-varying sequences containing a static background.

In addition to the aforementioned methods, there is a further category of approaches which are constructed by stacking multiple modules of nonlinear operations atop of each other, commonly referred to as deep learning networks [38], [39]. However, the majority of these methods are designed to operate on static data and are rarely employed to model time-varying patterns. Among few attempts for dealing with time-varying patterns are some works on using variants of the restricted Boltzmann machine (RBM) for specific time series data, i.e., human motion analysis in [40], [41]. Some other deep-learning methods address video data with convolutional learning of spatio-temporal features or stacked auto-encoders [42]–[44].

However, as these methods have not been evaluated on the most commonly employed standard dynamic texture databases (UCLA50, Dyntex, and Dyntex + +) in a recognition scenario, a quantitative comparison of the proposed method against these approaches is not feasible.

In [45], the hierarchical matching pursuit (HMP) method, which builds a feature hierarchy layer-by-layer using an efficient matching pursuit encoder, is proposed. The proposed method includes different modules as batch (tree) orthogonal matching pursuit, spatial pyramid max pooling, and contrast normalization. The proposed HMP method is shown to be scalable and handle full-size images efficiently. The HMP approach is then compared with state-of-the-art algorithms including convolutional deep belief networks, scale invariant feature transform (SIFT)-based single layer sparse coding, and kernel-based feature learning and shown to yield superior accuracy on three types of image classification problems. However, no extension of the approach is considered to the dynamic texture classification and evaluations are conducted only on static image databases.

The current work addresses the representation and recognition of temporally varying textures using the binarized statistical image features (BSIF) [46]. Similar to some other descriptors such as local binary pattern [35] and local phase quantization [47], the static BSIF produces a binary code for each pixel. This is realized by binarizing the responses of a set of filters operating on local image patches. The outputs of the filters are generated by linearly projecting local image patches onto a subspace whose basis vectors are learnt via statistical analysis of natural images. For characterization of dynamic patterns, the original BSIF operator which was proposed to operate on static images is extended in this work to the spatio-temporal domain. This objective is realized by analyzing the spatial and temporal variations of an image sequence by applying spatial/spatio-temporal filters on various regions of a 3D signal similar in spirit to the LBP-TOP approach [34]. The filters are learnt in a similar fashion as the static BSIF descriptor but working in a spatio-temporal domain instead. To this end, different filters are learnt on three orthogonal planes of XY, XT, and YT. In the process of filter learning, a whitening transformation on the pixels in various regions of spatio-temporal support is applied, followed by an independent coefficient analysis (ICA) transformation. The ICA transformation serves to maximize the independence of filter responses while the whitening transformation captures spatial and spatio-temporal domain correlations in addition to neutralizing the dominant effects of the low-frequency content and contrast gain and luminance control [48]. As the filter responses are made statistically independent via ICA, they can be binarized independently to produce a binary code for each temporal pixel. This is a fundamental prerequisite for independent processing of filter responses which is not completely fulfilled in other descriptors such as LBP and its variants. By varying the sizes of dynamic BSIF filters, a multiscale dynamic texture representation is then derived which is able to capture spatio-temporal information at multiple resolutions. The responses of filters are finally summarized via histograms. The recognition of a dynamic pattern then boils down to a comparison of distributions of coded patterns. In summary, the main contributions of the current work include a dynamic texture descriptor based on the BSIF representation. The proposed dynamic descriptor has computational overhead by a small factor compared to the static case and compares very favorably with the existing representations. An extension of the proposed dynamic descriptor to a multiscale version, applicable to image sequences which improves the discriminatory capacity of the representation by a great extent, is proposed next. It is shown that a simple χ^2 distance-based classifier on the obtained representations can achieve promising performance on different databases. The remainder of the paper is organized as follows: In Section II, we overview the original static BSIF descriptor. Section III details the description of the proposed spatio-temporal texture descriptor (BSIF-TOP). The discussion is then followed by a multiscale extension of the proposed BSIF-TOP representation, i.e., MBSIF-TOP. In Section IV, an experimental evaluation of the proposed method along with a comparison to other approaches on different databases is presented. The paper is drawn to conclusions in Section V.

II. Binarized Static Image Features

A. Overview of Static BSIF

The BSIF is a generative model based on the ICA [48]. ICA represents the data as a linear transformation of some latent independent components. Let $\mathbf{p}$ denote the pixel grey values in an image patch concatenated into a vector. Using ICA, $\mathbf{p}$ can be represented using a feature matrix $\mathcal{F}$ as

$$\mathbf{p} = \mathcal{F}\mathbf{r} \tag{1}$$

where the elements of the vector $\mathbf{r}$ are some unknown random variables which differ from one patch to another. Conversely, the elements of $\mathcal{F}$ are constant and the same for all different image patches. A fundamental assumption regarding this linear generative model is that the elements of $\mathbf{r}$ are statistically independent. In this case, one may, using a large enough number of training samples, recover a reasonable approximation to $\mathcal{F}$ up to a multiplicative constant without explicitly knowing the latent vector $\mathbf{r}$ [48]. Estimation of $\mathcal{F}$ is equivalent to determining the matrix $\mathbf{F}$ which produces $\mathbf{r}$ as the output of a number of linear filters as

$$\mathbf{r} = \mathbf{F}\mathbf{p} \tag{2}$$

where each row of $\mathbf{F}$ represents a filter applied to the pixels in $\mathbf{p}$.

In practice, the statistical models are applied on preprocessed data. Suppose that the data variables of a single patch after preprocessing are collected into the vector $\mathbf{z} = (z_1, \ldots, z_N)$. Commonly, a linear transformation is used for preprocessing. In this case, z_is would be linear transformations of the independent components r_is. This can be readily observed by multiplying both sides of (1) by the matrix performing the preprocessing and obtain

$$\mathbf{z} = \mathcal{U}\mathbf{r} \tag{3}$$

where matrix $\mathcal{U}$ is obtained by multiplying matrix $\mathcal{F}$ by the preprocessing transformation matrix, $\mathbf{V}$. In practice, a whitening transformation is used as the preprocessing step as it is found to

be instrumental in contrast gain and luminance control [48]. In this case, for matrix $\mathcal{U}$ to be invertible, the number of independent components should be chosen in a way that it equals the number of variables produced after the whitening transformation. Under this condition, the system in (3) would be invertible in a unique way, producing the vector $\mathbf{r}$ as a linear function of $\mathbf{z}$ as

$$\mathbf{r} = \mathbf{U}\mathbf{z} \tag{4}$$

where matrix $\mathbf{U}$ is obtained by inverting matrix $\mathcal{U}$. The filter matrix $\mathbf{F}$ in (2) can then be obtained by multiplying the linear transformations $\mathbf{U}$ and $\mathbf{V}$, i.e.,

$$\mathbf{F} = \mathbf{U}\mathbf{V}. \tag{5}$$

As a result, the independent components r_i's of vector $\mathbf{r}$ are obtained as

$$\mathbf{r} = \mathbf{U}\mathbf{V}\mathbf{p}. \tag{6}$$

In summary, given an image $\mathbf{p}$ of size $d \times d$ pixels, one applies N filters on the pixels of $\mathbf{p}$ using the filter matrix $\mathbf{F}^{N \times d^2}$ and obtains N responses which are stacked into the vector $\mathbf{r}$. As the filter responses (r_is) are independent, they can be processed independently. This is in contrast to some other approaches [47], [35] where the independence assumption is imposed only approximately. A useful post-processing step is binarizing r_is by thresholding at zero to produce the binarized features b_is as

$$b_i = \begin{cases} 1, & r_i > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

The binarized features of b_is can then be summarized using aggregate statistics such as histograms.

Learning BSIF Filters: The training procedure for filter matrix $\mathbf{F}$ can be summarized as follows. Using a training set of image patches randomly taken from natural images their covariance matrix is estimated and eigen-decomposed. The dimensionality of each patch is then reduced using N (number of the filters used) principal eigenvectors of the covariance matrix divided by their standard deviations. At the end of this step, whitened data samples $\mathbf{z}$ are obtained. In more detail, if the eigen decomposition of the covariance matrix $\mathbf{\Sigma}$ is $\mathbf{\Sigma} = \mathbf{Y}\mathbf{D}\mathbf{Y}^\top$, where $\mathbf{D}$ is the diagonal matrix of eigen values of $\mathbf{\Sigma}$ in a descending order and the columns of $\mathbf{Y}$ are the corresponding eigenvectors of $\mathbf{\Sigma}$, then matrix $\mathbf{V}$ which is used for whitening and dimensionality reduction is given by

$$\mathbf{V} = [\mathbf{D}^{-1/2}\mathbf{Y}]_{1:N} \tag{8}$$

where $[.]_{1:N}$ denotes the first N rows of a matrix. Next, given the whitened data samples $\mathbf{z}$, the independent component analysis is employed to estimate an orthogonal matrix $\mathbf{U}$. Having estimated the matrices of $\mathbf{U}$ and $\mathbf{V}$, the final filter matrix is obtained using (5) (see Fig. 2). The BSIF descriptor has been found to yield very useful features for static texture description in various applications [46], [49]. In particular, the BSIF descriptor has been found to provide more discriminative representations than other well known descriptors such as LBP [35] or LPQ [47].

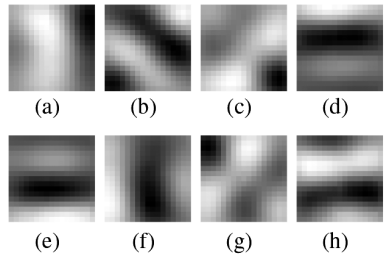


Fig. 2. Eight 17×17 BSIF filters ($N = 8$) obtained for the XY plane.

III. Spatio-Temporal BSIF: BSIF-TOP

The BSIF approach can be naturally extended to the spatio-temporal domain by considering a three-dimensional slice of an image sequence instead of a 2D patch as the data to be filtered. In this case, the data to learn the filters is formed by vectorizing 3D image blocks which are then processed by PCA and ICA transformations as in the spatial case to produce the filters. This procedure, in fact, leads to learning 3D filters applicable to (slices of) image sequences. However, there are certain drawbacks in following this approach. The whitening transformation used reduces the dimension of the data down to N, where N is the number of filters employed. In the case of 3D spatio-temporal data, the size of the data grows linearly with the temporal dimension. For instance, considering an image patch of size 9×9 in the spatial case, a corresponding slice of image sequence would be of size $9 \times 9 \times t$, where t is the temporal dimension. Considering the same value of 9 for t, the dimensionality of the data after vectorizing would be 729. In this case, if one reduces the dimensionality from 729 to a similar dimension as in the spatial case (i.e., 8), a large portion of high-frequency content of the data would be lost (note that the dimensionality in the spatial case would be reduced from 81 to 8). Hence, in practice a larger number of eigenvectors in the whitening transformation should be retained and as a result more filters are required to extract informative features from 3D data. However, increasing N would lead to an exponential growth in histogram bins (the number of histogram bins is 2^N) and thus increases the computational cost drastically. An alternative approach is to consider the image sequence on three orthogonal planes, similar in spirit to the LBP-TOP approach [34]. Such an approach helps the LBP operator handle a larger number of neighboring points on each plane while reducing the computational complexity of the dynamic texture descriptor. In the LBP-TOP approach, an image sequence is considered as a stack of XT planes in axis Y, YT planes in axis X, and XY planes in axis T, where the YT and XT planes capture information about the space-time transitions of the video sequence and correlations between the time and spatial domains and the XY plane represents spatial information, as seen in Fig. 3. Following the same procedure for the BSIF descriptor would have a computational overhead only by a factor of three compared to the BSIF descriptor operating in the spatial case only. Moreover, larger temporal support region variations can be considered for filtering operations if desired. Once the BSIF codes for all frames are computed, a histogram is constructed for each plane. Finally, three histograms corresponding to three orthogonal planes are concatenated to form the BSIF-TOP descriptor.

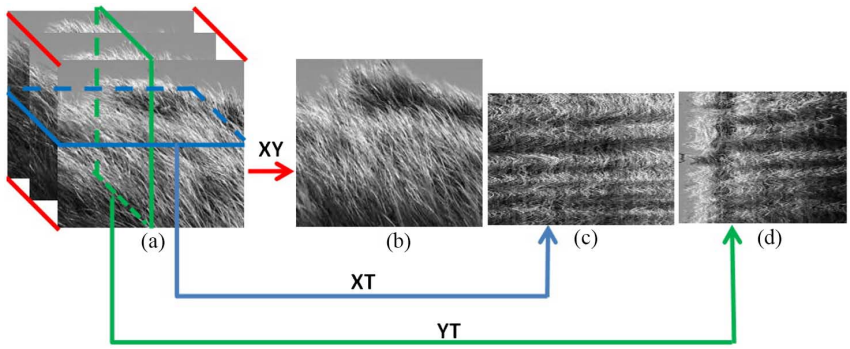


Fig. 3. (a) Image sequence. (b) Sample frame in the XY plane. (c) Sample frame in the XT plane. (d) Sample frame in the YT plane.

For the spatio-temporal BSIF descriptor, the same procedure as in the static BSIF is followed to learn three sets of filters, each on a different plane: XY, XT, and YT. In other words, the training data is constructed using randomly chosen patches on each of the three XY, XT, and YT planes. Once the covariance matrices are estimated for each plane and the data is preprocessed using the inferred whitening transformations, three matrices are learnt via ICA and finally used to produce three sets of filters, each of which is specific to a different plane. Some filters obtained for the XY plane are depicted in Fig. 2. For training, we use an external set of random image sequences of dynamic textures collected from the web.

A. Multiscale Analysis

Suppose the size of each individual BSIF filter is fixed at d^2. In this case, using a larger number of filters (increasing N) would include more high-frequency components into the descriptor. This is because the N eigenvectors of the covariance matrix of the training data are sorted in a descending order with respect to their corresponding eigenvalues and increasing N would include more eigenvectors corresponding to smaller eigenvalues into the whitening transformation. Conversely, using a fixed number of filters N, by increasing the size of each filter, the variations of the signal over a larger support region are taken into account. In others words, the descriptor now captures low-frequency contents of image sequence. It has been observed that using eight filters ($N = 8$) results in an acceptable frequency response, able to capture a wide range of frequency content of images [46]. Hence, the number of filters in all experiments in this work is fixed to eight, producing an eight-bit binary code for each pixel.

As noted earlier, the other parameter controlling the frequency content of the feature is the filter size. While smaller filters capture high-frequency variations of texture, larger filters are better suited to deal with blurring effects and low-frequency content. In this work, this trade-off is moderated via a multiscale extension of the BSIF-TOP descriptor. In order to construct a multiscale representation, the sizes of the filters are varied to capture information at multiple scales in the spatial and temporal domain, i.e., $d = \{3, 5, \ldots 17\}$. We have learnt filters in eight scales and in each scale eight filters are learnt. As such, the dimension of the final spatio-temporal multiscale BSIF descriptor is 6144. The results of BSIF coding in the XY plane for a sample frame is depicted in Fig. 4.

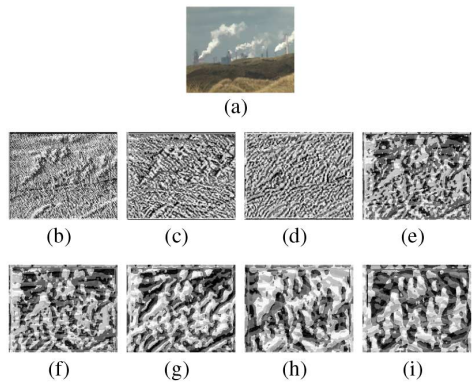


Fig 4 (a) Sample frame. (b)–(i) BSIF code images at different scales in the XY plane.

B. MBSIF-TOP Spatio-Temporal Texture Descriptor

In the proposed approach to multiresolution analysis, BSIF-TOP operators at Z scales are first applied to all frames of an image sequence in three orthogonal planes. A grey level code for each pixel at each resolution is thus obtained. The BSIF-TOP pattern histogram for the resulting coded sequence in the scale of s in each plane, $\mathbf{h}_s$, is computed by

$$\begin{aligned} \mathbf{h}_s &= [h_s^0, h_s^1, \ldots, h_s^{L-1}] \\ h_s^i &= \sum_{p \in \text{plane}} \mathbb{1}\{\text{BSIF}_s(p) = i\} \times s \in [1, 2, \ldots, Z], \\ L &= 256 \end{aligned} \tag{9}$$

where p is a pixel in a specific plane, $\mathbb{1}\{.\}$ is the indicator function equal to one when its argument is true and zero otherwise, and L is the number of histogram bins. The size of the BSIF filter at scale s is $d \times d$ where $d = 2 \times s + 1$. When the dynamic textures to be compared are of different temporal/spatial sizes, $\mathbf{h}_s$ is normalized to yield a coherent description, i.e., probability density.

$$\tilde{\mathbf{h}}_s = \frac{\mathbf{h}_s}{\sum_{i=0}^{L-1} h_s^i} \tag{10}$$

By concatenating all the histograms computed at different scales into a single vector, the multiresolution dynamic texture descriptor on each plane is obtained.

$$\mathbf{q}^{\text{plane}} = [\tilde{h}_1, \tilde{h}_2, \ldots, \tilde{h}_Z] \tag{11}$$

Once the histograms are obtained on the XY, XT, and YT planes, three histograms are concatenated to form the final spatio-temporal descriptor for the image sequence as

$$\mathbf{q} = [\mathbf{q}^{XY}\mathbf{q}^{XT}\mathbf{q}^{YT}]. \tag{12}$$

The proposed spatio-temporal descriptor multiscale binarised statistical image features on three orthogonal planes (MBSIF-TOP) has different levels of locality. First, it captures the information at multiple resolutions at the pixel level. This is achieved via a multiresolution representation on three orthogonal planes. The distribution of codes in each plane is represented via plane-specific histograms. At a higher level, the global content is represented by the concatenation of histograms from different planes at different scales. The histograms of two image sequences are finally compared using the χ^2 distance measure

$$\chi^2(\mathbf{q}_1, \mathbf{q}_2) = \sum_i \frac{(\mathbf{q}_1(i) - \mathbf{q}_2(i))^2}{\mathbf{q}_1(i) + \mathbf{q}_2(i)}. \tag{13}$$

IV. Experimental Evaluation

In this section a detailed evaluation of the proposed method is provided on various databases for the task of dynamic texture representation and recognition.

A. UCLA Data Set

In order to assess the performance of the proposed descriptor, in this section a detailed evaluation of our method in various scenarios on the UCLA database [26], [50] is presented. The UCLA database comprises 50 sets, corresponding to 50 scenes, each represented by four sequences of a dynamic texture, for a total of 200 sequences. These include boiling water, fountains, fire, waterfalls, plants and flowers swaying in the wind, etc. Each sequence is comprised of 75 frames in a resolution of 160×110 pixels. We use a version of this database where each sequence is clipped to a 48×48 window that contains the key statistical and dynamical features [50], [26].

50-Class Breakdown: Fifty distinct classes are considered in this scenario. Previous methods evaluated in this case have used different portions of the database as training and test data as follows.

Leave-one-out scheme: Similar to the work in [23] and [50], a leave-one-out classification procedure is followed here where a correct decision for a test sequence is defined as having one of the three remaining sequences of the same scene as its nearest neighbor. The results of an evaluation of the MBSIF-TOP descriptor in this scenario using filters of varying scales are reported in Table I. Using a single resolution, the proposed approach yields an encouraging performance of a 93% correct classification rate. However, by incorporating descriptors corresponding to different resolutions into the representation, better performance is expected. Increasing the number of scales used, the recognition performance is improved as expected. This can be observed in the table, where an impressive performance of 99.5% is achieved using seven scales. Beyond this point, it was observed that employing a larger number of resolutions did not improve the performance. This can be attributed to the decreased relative sizes of each frame compared to the filter size at the coarser scales, in addition to the correlation between histograms of different scales. A comparison of the seven-scale version of the proposed descriptor to the methods of [23] and [50] in the 50-class scenario using a leave-one-out classification rule is presented in Table II. It is evident that the proposed method achieves the best recognition performance among other competitors with a large margin. The only misclassified sequence in this case is a smoke sequence confused with a water sequence which, when visually analyzed, have very similar appearance/dynamic characteristics, as seen in Fig. 5.

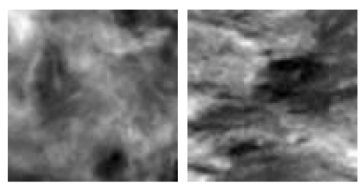
Four cross-fold validation scheme: In [51] and [52], a different split of the data set is used for training and test. In these works, 75% of the data is used for training (three sequences per class) and the rest for test (one sequence per class). The test is performed four times, each time using a different sequence as the test sample. Finally, the results were averaged over four trials. The results of the proposed method are compared against

TABLE I
Effect of Using a Multiresolution BSIF-TOP Descriptor With Different Number of Scales on the Recognition Performance in the 50-Class Leave-One-Out Scenario on the UCLA Data Set

No of Scales Used	Recognition Performance
1	93%
2	95.5%
3	96%
4	96.5%
5	97%
6	97.5%
7	99.5%

TABLE II
Comparison of the Performance of the Proposed Method to Other Approaches in the 50-Class Scenario on the UCLA Data Set in a Leave-One-Out Scheme

Method	Recognition Performance
Martin Dist.[50]	89.5%
L2 Bhattacharyya [23]	81%
MBSIF-TOP	**99.5%**



(a) (b)

Fig. 5. (a) Only misclassified sequence in the UCLA database (Smoke) in the 50-class leave-one-out evaluation scheme. (b) Water sequence: the sequence with which the system has confused smoke with.

TABLE III
COMPARISON OF THE PERFORMANCE OF THE PROPOSED METHOD TO OTHER APPROACHES IN THE 50-CLASS SCENARIO ON THE UCLA DATA SET IN A FOUR CROSS-FOLD VALIDATION SCHEME

Method	Recognition Performance
Distance Learning [51]	99%
KDT-MD [52]	97.5%
MBSIF-TOP	**99.5%**

other approaches in this scenario in Table III. The results reported in the table clearly illustrate the superiority of the proposed descriptor to some other approaches.

Nine-Class Breakdown: As pointed out in other works [31], [51], examining the UCLA database reveals that many of the sequences are semantically capturing the same scene. Examples include different scenes of windblown vegetation which share fundamental spatio-temporal similarities as well as different sequences of water and fire, etc. Taking into account these observations, in [51] the UCLA data set is reorganized into nine semantic categories being boiling water (eight), fire (eight), flowers (12), fountains (20), plants (108), sea (12), smoke (four), water (12), and waterfall (16), where the numbers in the parentheses indicate the number of sequences for each category. Similar to [51], we use half of the data for training and half for test. The experiment is repeated 20 times with random splits of the data into train and test sets. A correct classification in this case is defined as assigning the test sequence into one of the training sequences from the same category based on a χ^2 nearest neighbor distance. The method in [51] achieves a 95.6% average correct classification rate using a maximum margin distance learning method. The proposed MBSIF-TOP + χ^2 method achieves a 98.75% correct classification rate in this scenario using a much simpler distance metric, i.e., χ^2 distance.

Eight-Class Breakdown: Similar to earlier work [31], as the number of sequences corresponding to plants far exceeds that of the other classes, eight classes are used in this experiment after discarding the plant sequences. In this case, similar to [31], 50% of the dataset is used for training and 50% for test. A correct classification for the MBSIF-TOP + χ^2 method in this case is defined as assigning the test sequence into one of other training sequences from the same category based on a nearest neighbor rule. The results of this test for the proposed method along with two other approaches reporting the best known results are given as confusion matrices in Table IV. As in the 50-class scenario, in the proposed approach the only misclassified sequence is a smoke sequence confused with water. The average recognition rate in this scenario for the proposed descriptor is $\sim$97.8%, significantly better than other two methods having averages of 80% for the method in [31] and 54.12% obtained by the approach in [50]. The impressive performance in the proposed approach is achieved despite using a simple nearest neighbor classifier based on the χ^2 distance on the MBSIF-TOP histogram which emphasizes the discriminatory capacity of the proposed descriptor.

B. Dyntex Data Set

The DynTex dataset is a varied data set of dynamic textures [53]. Although it has been used for dynamic texture recognition, different methods have used different experimental configurations such as different categories and subsets. Among others, the work in [34], [54] provides a precise characterization of the configuration used. In their experiments, a version of the data set comprised of 35 categories is employed. Each sequence is divided into eight nonoverlapping sub-sequences by randomly cutting the sequence in X and Y dimensions. Two additional sequences were generated by randomly cutting in the temporal axis. In this way, 10 samples of each class are obtained which are all different from each other in image size and sequence length. All these samples are finally used in the dynamic texture recognition experiment. Such sampling scheme in effect makes the recognition more challenging. The experiment is conducted using the leave-one-group out scheme, i.e., a single sample per class is picked to form the test set and the rest are used as the training data. In our experiments, each class is represented by all the feature vectors of the samples in the training set. A test sample is assumed to be classified correctly if it has one of the training samples as its nearest neighbor in the feature space. The classification rates of the proposed method along with other approaches are summarized in Table V. It can be observed from Table V that the proposed method performs very well compared to other approaches.

More recently, in order to make the evaluations on the DynTex consistent, different subsets have been compiled and labelled. In these subsets, only a single dynamic texture is present in each sequence without any panning or zooming of the camera. These are the Alpha, Beta, and Gamma subsets. The Alpha dataset is composed of 60 dynamic textures divided into three classes, the Beta dataset is composed of 162 dynamic textures divided into 10 classes, and the Gamma dataset is comprised of 275 dynamic textures, each belonging to one of 10 classes. Similar to [55], we use a leave-one-out cross validation scheme. The recognition results obtained on the aforementioned subsets along with the results obtained in [55] are presented in Table VI. From the table, one can observe that the proposed approach achieves better performance on all three subsets. While on the Alpha subset the improvement compared to the method of [55] is moderate, on the Beta and Gamma subsets the proposed descriptor achieves considerably better recognition performance by more than 20%.

C. Dyntex + + Data Set

The DynTex database [53] is a diverse set of dynamic texture videos intended for the evaluation of spatio-temporal texture classification. Although some evaluations have been performed on the DynTex database, a second version of this data set is compiled (Dyntex + +) having a number of appealing properties [51]. In creating this new version of the Dyntex data set the goal was to organize the raw data in the DynTex database so as to provide a richer benchmark that will be nearly similar to the UCLA benchmark data set. The sequences in the Dyntex + + data set include only a representative dynamic scene without any background or other dynamic patterns. In addition, there is only one spatio-temporal texture present in each sequence in this

TABLE IV
CONFUSION MATRICES FOR THE EIGHT SEMANTIC CATEGORIES OF THE UCLA DATA SET

Ground Truth	Boiling Water	Fire	Flowers	Fountains	Sea	Smoke	Water	Waterfall
Boiling Water (8)	100%	0%	0%	0%	0%	0%	0%	0%
Fire (8)	0%	100%	0%	0%	0%	2%	0%	0%
Flowers (12)	0%	0%	100%	0%	0%	0%	0%	0%
Fountains (20)	0%	0%	0%	100%	0%	0%	0%	0%
Sea (12)	0%	0%	0%	0%	100%	0%	0%	0%
Smoke (4)	0%	0%	0%	0%	0%	75%	25%	0%
Water (12)	0%	0%	0%	0%	0%	0%	100%	0%
Waterfall (16)	0%	0%	0%	0%	0%	0%	0%	100%

Predicted Class Using the method in [31]

Ground Truth	Boiling Water	Fire	Flowers	Fountains	Sea	Smoke	Water	Waterfall
Boiling Water (8)	100%	0%	0%	0%	0%	0%	0%	0%
Fire (8)	0%	98%	0%	0%	0%	2%	0%	0%
Flowers (12)	0%	0%	100%	0%	0%	0%	0%	0%
Fountains (20)	0%	0%	2%	50%	0%	0%	0%	48%
Sea (12)	0%	0%	0%	0%	100%	0%	0%	0%
Smoke (4)	0%	90%	0%	0%	0%	10%	0%	0%
Water (12)	0%	46%	0%	0%	0%	3%	51%	0%
Waterfall (16)	0%	0%	0%	0%	0%	0%	0%	100%

Predicted Class Using the method in [50]

Ground Truth	Boiling Water	Fire	Flowers	Fountains	Sea	Smoke	Water	Waterfall
Boiling Water (8)	0%	0%	0%	0%	0%	0%	0%	100%
Fire (8)	0%	50%	0%	0%	0%	0%	0%	50%
Flowers (12)	0%	0%	50%	0%	17%	0%	0%	33%
Fountains (20)	0%	0%	0%	0%	0%	0%	0%	100%
Sea (12)	0%	0%	0%	0%	100%	0%	0%	0%
Smoke (4)	0%	0%	0%	0%	0%	100%	0%	0%
Water (12)	0%	0%	0%	0%	0%	67%	33%	0%
Waterfall (16)	0%	0%	0%	0%	0%	0%	0%	100%

TABLE V
COMPARISON OF THE PERFORMANCE OF THE PROPOSED METHOD TO OTHER APPROACHES ON THE DYNTEX DATA SET

Method	Recognition performance
LBP-TOP [34]	97.14%
DFS [54]	97.63%
MBSIF-TOP	**98.61**%

TABLE VI
COMPARISON OF THE PERFORMANCE OF THE PROPOSED METHOD TO THE METHOD OF [55] ON THE ALPHA, BETA, AND GAMMA SUBSETS OF THE DYNTEX DATASET

Subset	The proposed method	The method of [55]
Alpha	90.0 %	88.3%
Beta	90.7%	69.8%
Gamma	91.3%	68.3%

data set. There is no zooming or panning in the sequences and the ground truth labels of all the sequences are provided. In this data set, the sequences are filtered, preprocessed, and labelled. In total, there are 36 classes, each represented by 100 sequences.

Dynamic Texture Recognition: In this section, the MBSIF-TOP descriptor is evaluated for recognition of dynamic textures on the Dyntex++ data set. Following the same test procedure as in [54] and [51], half of the database is used for training and the other half for testing. In the proposed MBSIF-TOP + χ^2 method a test sample is assumed to be classified correctly if it has one of the 50 training sequences of the same class as its nearest neighbor. As in the previous experiments, a χ^2 distance measure is used for comparing

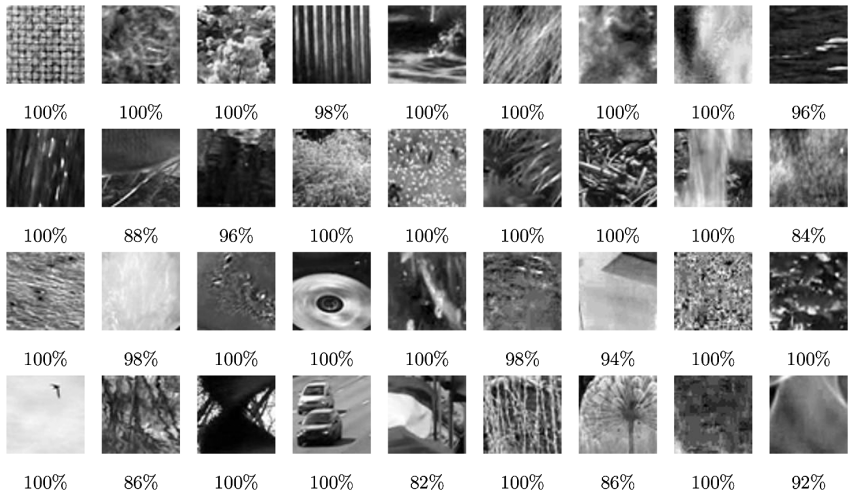


Fig. 6. Correct classification rates for different classes in the Dyntex $++$ database.

TABLE VII
COMPARISON OF THE PERFORMANCE OF THE PROPOSED METHOD TO OTHER APPROACHES ON THE Dyntex $++$ DATA SET USING 50% TRAINING AND 50% TEST DATA

Method	Recognition performance
Distance Learning [51]	63.7%
DFA [54]	89.9%
LBP-TOP [34]	89.5%
MBSIF-TOP	**97.17%**

TABLE VIII
CORRECT CLASSIFICATION RATES FOR DIFFERENT CLASSES IN THE Dyntex $++$ DATABASE

Descriptor used	Correct recognition rate
MBSIF-TOP	97.17%
MBSIF-XY	90.50%
MBSIF-XT	94.17%
MBSIF-YT	92.94%

the histograms obtained from three orthogonal planes. The results of this evaluation are reported in Table VII. It can be observed that the proposed MBSIF-TOP descriptor achieves an impressive performance of 97.17% correct classification rate on this data set, improving the previous best result by more than 7%. This is obtained despite the fact that the a simple classifier based on χ^2 distance is used in the current method whereas more sophisticated classifiers are employed in some other works. Using more complex classifiers, such as SVMs or kernel discriminant analysis, better performance is expected using the MBSIF-TOP descriptor; however, this is beyond the scope of the current work as we aim at objectively gauging the capacity of the proposed BSIF-TOP descriptor. It is worth reiterating that the employed MBSIF-TOP filters are trained on a separate external dataset of randomly chosen sequences and evaluated on the Dyntex data set which demonstrates its generalization capacity across different data sets. The correct recognition rates for different classes of this database are given separately in Fig. 6.

Comparison of Different Planes: In this section, the discriminatory capacity of MBSIF-TOP descriptor on each individual plane is investigated. In the MBSIF-TOP descriptor, the XT and YT planes capture temporal variations of patterns whereas the XY plane captures mostly the appearance. However, as the frames in the XY plane change over time, the histogram obtained on the XY plane includes some dynamic information of the sequence in addition to appearance. The same experimental set up as in the previous section is followed. Table VIII reports the correct classification rates obtained using all three orthogonal planes and also the XT, YT, and XY planes separately on the Dyntex $++$ data set. As expected, the correct recognition rates obtained using either one of the individual planes are much lower than considering three planes together. An interesting observation from the table is that the information on each of the XT or YT planes is more discriminative than the XY plane. This clearly demonstrates that the proposed MBSIF-TOP descriptor effectively captures the dynamic information present in the sequence conveyed by the XT and YT planes.

V. CONCLUSIONS

We addressed the representation and recognition of dynamic textures. A multiscale descriptor based on the binarized statistical image features was proposed for this purpose. The proposed descriptor (MBSIF-TOP) was similar in spirit to the well known LBP-TOP approach in the sense that it estimates histograms of binary coded image sequences on three orthogonal planes (XY, XT, and YT). However, unlike the LBP-TOP approach, the MBSIF-TOP descriptor used the statistics of natural image sequences to enhance its representation capacity. The code generation in the proposed approach was realized via filtering operations on different regions of spatial/spatio-temporal support. The filters employed in the proposed descriptor were estimated via ICA in conjunction with a whitening transformation. The experimental evaluations of the proposed methodology on different databases clearly illustrated the merits of the proposed descriptor for dynamic texture recognition compared to other alternatives.

REFERENCES

[1] V. Bruce, P. R. Green, and M. Georgeson, *Visual Perception*. London, U.K.: Psychol. Press, 1996.

[2] G. Johansson, "Visual perception of biological motion and a model for its analysis," *Perception Psychophys.*, vol. 14, no. 2, pp. 201–211, 1973.

[3] S. Rahimzadeh Arashloo, J. Kittler, and W. Christmas, "Facial feature localization using graph matching with higher order statistical shape priors and global optimization," in *Proc. 4th IEEE Int. Conf. Biometrics: Theory Appl. Syst.*, Sep. 2010, pp. 1–8.

[4] S. Rahimzadeh Arashloo and J. Kittler, , D. Cremers, Y. Boykov, A. Blake, and F. Schmidt, Eds., "Pose-invariant face matching using mrf energy minimization framework," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*, ser. Lecture Notes in Comput. Sci. Berlin, Germany: Springer, 2009, vol. 5681, pp. 56–69.

[5] S. Rahimzadeh Arashloo and J. Kittler, D. Cremers, Y. Boykov, A. Blake, and F. Schmidt, Eds., "Efficient processing of mrfs for unconstrained-pose face recognition," in *Proc. 6th IEEE Int. Conf. Biometr.: Theory, Appl. Syst.*, Sep. 2013, pp. 1–8.

[6] A. J. Otoole, D. A. Roark, H. Abdi, and A. J. Otoole, "Recognizing moving faces: A psychological and neural synthesis," *Trends Cognitive Sci.*, vol. 6, no. 6, pp. 261–266, 2002.

[7] M. Haas, J. T. Rijsdam, B. Thomee, and M. S. Lew, "Relevance feedback: Perceptual learning and retrieval in bio-computing, photos, and video," in *Proc. 6th ACM SIGMM Int. Workshop Multimedia Inf. Retrieval*, 2004, pp. 151–156.

[8] X. Wang, C. Zhang, and Z. Zhang, "Boosted multi-task learning for face verification with applications to web image and video search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2009, pp. 142–149.

[9] W. Ali, F. Georgsson, and T. Hellstrom, "Visual tree detection for autonomous navigation in forest environment," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2008, pp. 560–565.

[10] S. Liao, G. Zhao, V. Kellokumpu, M. Pietikainen, and S. Li, "Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes," in *Proc. IEEE Comput. Vis. Pattern Recog.*, Jun. 2010, pp. 1301–1306.

[11] J. Huang, J. Zhao, W. Gao, C. Long, L. Xiong, Z. Yuan, and S. Han, "Local binary pattern based texture analysis for visual fire recognition," in *Proc. IEEE 3rd Int. Congr. Image Signal Process.*, 2010, pp. 1887–1891.

[12] M. Sizintsev and R. P. Wildes, "Spatiotemporal stereo via spatiotemporal quadric element (stequel) matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 493–500.

[13] K. G. Derpanis and R. P. Wildes, "Early spatiotemporal grouping with a distributed oriented energy representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 232–239.

[14] K. G. Derpanis and R. P. Wildes, , H. Zha, R. ichiro Taniguchi, and S. J. Maybank, Eds., "Detecting spatiotemporal structure boundaries: Beyond motion discontinuities," in *ACCV (2)*, ser. Lecture Notes in Comput. Sci. New York, NY, USA: Springer, 2009, vol. 5995, pp. 301–312.

[15] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Proc IEEE Int. Workshop Perform. Eval. Track. Surveillance*, Beijing, China, Oct. 2005, pp. 65–72.

[16] K. J. Cannons, J. M. Gryn, and R. P. Wildes, , K. Daniilidis, P. Maragos, and N. Paragios, Eds., "Visual tracking using a pixelwise spatiotemporal oriented energy representation," in *ECCV (4)*, ser. Lecture Notes in Comput. Sci. New York, NY, USA: Springer, 2010, vol. 6314, pp. 511–524.

[17] G. Zhao, M. Barnard, and M. Pietikinen, "Lipreading with local spatiotemporal descriptors," *IEEE Trans. Multimedia*, vol. 11, no. 7, pp. 1254–1265, Nov. 2009.

[18] T. Kung and W. Richards, "Inferring water from images," *Natural Comput.*, pp. 224–233, 1988.

[19] D. Chetverikov and R. Pteri, "A brief survey of dynamic texture description and recognition," in *Proc. Int. Conf. Comput. Recog. Syst.*, 2005, vol. 30, pp. 17–26.

[20] R. Pteri and D. Chetverikov, "Dynamic texture recognition using normal flow and texture regularity," in *Proc. Iberian Conf. Pattern Recog. Image Anal.*, 2005, vol. 3523, pp. 223–230.

[21] R. Polana and R. Nelson, , M. Shah and R. Jain, Eds., "Temporal texture and activity recognition," in *Motion-Based Recognition*, ser. Comput. Imag. Vis. Dordrecht, The Netherlands: Springer, 1997, vol. 9, pp. 87–124.

[22] Z. Lu, W. Xie, J. Pei, and J. Huang, "Dynamic texture recognition by spatio-temporal multiresolution histograms," in *Proc. IEEE Workshop Appl. Comput. Vision/IEEE Workshop Motion Video Comput.*, 2005, pp. 241–246.

[23] K. G. Derpanis and R. P. Wildes, "Spacetime texture representation and recognition based on a spatiotemporal orientation analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1193–1205, Jun. 2012.

[24] Y. Wang and S. chun Zhu, "Modeling textured motion: Particle, wave and sketch," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2003, pp. 213–220.

[25] M. Szummer and R. W. Picard, "Temporal texture modeling," in *Proc. IEEE Int. Conf. Image Process.*, 1996, pp. 823–826.

[26] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, "Dynamic textures," *Int.l J. Comput. Vis.*, vol. 51, no. 2, pp. 91–109, 2003.

[27] A. W. Fitzgibbon, "Stochastic rigidity: Image registration for nowhere-static scenes," in *Proc. Int. Conf. Comput. Vis.*, 2001, pp. 662–669.

[28] Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman, "Texture mixing and texture movie synthesis using statistical learning," *IEEE Trans. Vis. Comput. Graph.*, vol. 7, no. 2, pp. 120–135, Apr.-Jun. 2001.

[29] A. B. Chan and N. Vasconcelos, "Probabilistic kernels for the classification of auto-regressive visual processes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2005, pp. 846–851.

[30] F. Woolfe and A. W. Fitzgibbon, "Shift-invariant dynamic texture recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2006, vol. 3952, pp. 549–562.

[31] A. Ravichandran, R. Chaudhry, and R. Vidal, "View-invariant dynamic texture recognition using a bag of dynamical systems," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 1651–1657.

[32] M. R. Naphade, C.-Y. Lin, and J. R. Smith, "Video texture indexing using spatio-temporal wavelets," in *Proc. IEEE Int. Conf. Image Process.*, 2002, pp. 437–440.

[33] R. P. Wildes and J. R. Bergen, "Qualitative spatiotemporal analysis using an oriented energy representation," in *Proc. Eur. Conf. Comput. Vis.*, 2000, vol. 1843, pp. 768–784.

[34] G. Zhao and M. Pietikinen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 915–928, Jun. 2007.

[35] T. Ojala, M. Pietikinen, and T. Menp, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[36] B. Ghanem and N. Ahuja, "Extracting a fluid dynamic texture and the background from video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.

[37] B. Ghanem and N. Ahuja, "Phase based modelling of dynamic textures," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[38] K. Yu, Y. Lin, and J. D. Lafferty, "Learning image representations from the pixel level via hierarchical sparse coding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 1713–1720.

[39] Y. Bengio, A. C. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[40] G. W. Taylor, G. E. Hinton, and S. T. Roweis, "Modeling human motion using binary latent variables," in *Proc. Neural Inf. Process. Syst.*, 2006, pp. 1345–1352.

[41] G. W. Taylor and G. E. Hinton, "Factored conditional restricted Boltzmann machines for modeling motion style," in *Proc. Int. Conf. Mach. Learn.*, 2009, vol. 382, p. 129.

[42] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *Proc. Eur. Conf. Comput. Vis.*, 2010, vol. 6316, pp. 140–153.

[43] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 3361–3368.

[44] X. Yan, H. Chang, S. Shan, and X. Chen, "Modeling video dynamics with deep dynencoder," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 215–230.

[45] L. Bo, X. Ren, and D. Fox, "Hierarchical matching pursuit for image classification: Architecture and fast algorithms," in *Proc. Neural Inf. Process. Syst.*, 2011, pp. 2115–2123.

[46] J. Kannala and E. Rahtu, "BSIF: Binarized statistical image features," in *Proc. Int. Conf. Image Process.*, 2012, pp. 1363–1366.
[47] E. Rahtu, J. Heikkilä, V. Ojansivu, and T. Ahonen, "Local phase quantization for blur-insensitive image analysis," *Image Vis. Comput.*, vol. 30, no. 8, pp. 501–512, 2012.
[48] A. Hyvrinen, J. Hurri, and P. Hoyer*, Natural Image Statistics: A Probabilistic Approach to Early Computational Vision*, ser. Comput. Imag. Vis.. Berlin, Germany: Springer-Verlag, 2009, vol. 39.
[49] S. Rahimzadeh Arashloo and J. Kittler, "Class-specific kernel fusion of multiple descriptors for face verification using multiscale binarized statistical image features," *IEEE Trans. Inf. Forensics Security*, no. 99, p. 1, 2014, to be published.
[50] P. Saisan, G. Doretto, Y. N. Wu, and S. Soatto, "Dynamic texture recognition," in *Proc. IEEE. Comput. Vis. Pattern Recog.*, 2001, pp. 58–63.
[51] B. Ghanem and N. Ahuja, , K. Daniilidis, P. Maragos, and N. Paragios, Eds., "Maximum margin distance learning for dynamic texture recognition," in *Computer Vision – ECCV 2010*, ser. Lecture Notes Comput. Sci. Berlin, Germany: Springer, 2010, vol. 6312, pp. 223–236.
[52] A. Chan and N. Vasconcelos, "Classifying video with kernel dynamic textures," in *Proc. IEEE Comput. Vis. Pattern Recog.*, Jun. 2007, pp. 1–6.
[53] R. Pteri, S. Fazekas, and M. J. Huiskes, "Dyntex: A comprehensive database of dynamic textures," *Pattern Recog. Lett.*, vol. 31, no. 12, pp. 1627–1632, 2010.
[54] Y. Xu, Y. Quan, H. Ling, and H. Ji, D. N. Metaxas, L. Quan, A. Sanfeliu, and L. J. V. Gool, Eds., "Dynamic texture classification using dynamic fractal analysis," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 1219–1226.
[55] S. Dubois, R. Pteri, and M. Mnard, "Characterization and recognition of dynamic textures based on the 2d+t curvelet transform," *Signal, Image Video Process.*, pp. 1–12, 2013.

Shervin Rahimzadeh Arashloo received the Ph.D. degree in computer vision and pattern recognition from the Centre for Vision, Speech, and Signal Processing, University of Surrey, Guildford, U.K.

He is currently an Assistant Professor of Electrical Engineering with Urmia University, Urmia, Iran, and a Visiting Senior Fellow with the University of Surrey, U.K. His research interests include cognitive vision, biometrics, and representation learning.

Josef Kittler (M'74–LM'12) is Professor of Machine Intelligence with the Centre for Vision, Speech, and Signal Processing, University of Surrey, Guildford, U.K. He conducts research in biometrics, video and image database retrieval, medical image analysis, and cognitive vision. He authored the textbook *Pattern Recognition: A Statistical Approach* (Prentice-Hall, 1982), as well as more than 170 journal papers.

Dr. Kittler serves on the Editorial Board of several scientific journals in pattern recognition and computer vision.