Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions

Guoying Zhao and Matti Pietikäinen, *Senior Member, IEEE*

***Abstract*— Dynamic texture is an extension of texture to the temporal domain. Description and recognition of dynamic textures have attracted growing attention. In this paper, a novel approach for recognizing dynamic textures is proposed and its simplifications and extensions to facial image analysis are also considered. First, the textures are modeled with volume local binary patterns (VLBP), which are an extension of the LBP operator widely used in ordinary texture analysis, combining motion and appearance. To make the approach computationally simple and easy to extend, only the co-occurrences on three orthogonal planes (LBP-TOP) are then considered. A block-based method is also proposed to deal with specific dynamic events, such as facial expressions, in which local information and its spatial locations should also be taken into account. In experiments with two dynamic texture databases, DynTex and MIT, both the VLBP and LBP-TOP clearly outperformed the earlier approaches. The proposed block-based method was evaluated with the Cohn-Kanade facial expression database with excellent results. Advantages of our approach include local processing, robustness to monotonic gray-scale changes and simple computation.**

***Index Terms*— temporal texture, motion, facial image analysis, facial expression, local binary pattern**

I. Introduction

DYNAMIC or temporal textures are textures with motion [1]. They encompass the class of video sequences that exhibit some stationary properties in time [2]. There are lots of dynamic textures (DT) in the real world, including sea-waves, smoke, foliage, fire, shower and whirlwind. Description and recognition of DT are needed, for example, in video retrieval systems, which have attracted growing attention. Because of their unknown spatial and temporal extent, the recognition of DT is a challenging problem compared with the static case [3]. Polana and Nelson classified visual motion into activities, motion events and dynamic textures [4]. Recently, a brief survey of DT description and recognition was given by Chetverikov and Péteri [5].

Key issues concerning dynamic texture recognition include: 1) combining motion features with appearance features, 2) processing locally to catch the transition information in space and time, for example the passage of burning fire changing gradually from a spark to a large fire, 3) defining features which are robust against image transformations such as rotation, 4) insensitivity to illumination variations, 5) computational simplicity, and 6) multi-resolution analysis. To our knowledge, no previous method satisfies all these requirements. To address these issues, we propose a novel, theoretically and computationally simple approach based on local binary patterns. First, the textures are modeled with volume local binary patterns (VLBP), which are an extension of the LBP operator widely used in ordinary texture analysis [6], combining the motion and appearance. The texture features extracted in a small local neighborhood of the volume are not only insensitive with respect to translation and rotation, but also robust with respect to monotonic gray-scale changes caused, for example, by illumination variations. To make the VLBP computationally simple and easy to extend, only the co-occurrences on three separated planes are then considered. The textures are modeled with concatenated Local Binary Pattern histograms from Three Orthogonal Planes (LBP-TOP). The circular neighborhoods are generalized to elliptical sampling to fit the space-time statistics.

As our approach involves only local processing, we are allowed to take a more general view of dynamic texture recognition, extending it to specific dynamic events such as facial expressions. A block-based approach combining pixel-level, region-level and volume-level features is proposed for dealing with such non-traditional dynamic textures in which local information and its spatial locations should also be taken into account. This will make our approach a highly valuable tool for many potential computer vision applications. For example, the human face plays a significant role in verbal and non-verbal communication. Fully automatic and real-time facial expression recognition could find many applications, for instance, in human-computer interaction, biometrics, telecommunications and psychological research. Most of the research on facial expression recognition has been based on static images [7], [8], [9], [10], [11], [12], [13]. Some research on using facial dynamics has also been carried out [14], [15], [16]; however, reliable segmentation of lips and other moving facial parts in natural environments has proved to be a major problem. Our approach is completely different, avoiding error prone segmentation.

II. Related Work

Chetverikov and Péteri [5] placed the existing approaches of temporal texture recognition into five classes: methods based on optic flow, methods computing geometric properties in the spatiotemporal domain, methods based on local spatiotemporal filtering, methods using global spatiotemporal transforms and, finally, model-based methods that use estimated model parameters as features.

The methods based on optic flow [3], [4], [17], [18], [19], [20], [21], [22], [23], [24] are currently the most popular ones [5], because optic flow estimation is a computationally efficient and natural way to characterize the local dynamics of a temporal texture. Péteri and Chetverikov [3] proposed a method that combines normal flow features with periodicity features, in an attempt to explicitly characterize motion magnitude, directionality and periodicity. Their features are rotation-invariant, and the results are promising. But they did not consider the multi-scale properties of DT. Lu *et al.* proposed a new method using spatiotemporal

Manuscript received June 1, 2006; revised October 4, 2006.

The authors are with the Machine Vision Group, Infotech Oulu and Department of Electrical and Information Engineering, University of Oulu, P. O. Box 4500, FI-90014, Finland. E-mail: {gyzhao, mkp}@ee.oulu.fi.

multi-resolution histograms based on velocity and acceleration fields [21]. Velocity and acceleration fields of different spatiotemporal resolution image sequences were accurately estimated by the structure tensor method. This method is also rotation-invariant and provides local directionality information. Fazekas and Chetverikov compared normal flow features and regularized complete flow features in dynamic texture classification [25]. They concluded that normal flow contains information on both dynamics and shape.

Saisan *et al.* [26] applied a dynamic texture model [1] to the recognition of 50 different temporal textures. Despite this success, their method assumed stationary DTs that are well-segmented in space and time, and the accuracy drops drastically if they are not. Fujita and Nayar [27] modified the approach [26] by using impulse responses of state variables to identify model and texture. Their approach showed less sensitivity to non-stationarity. However, the problem of heavy computational load and the issues of scalability and invariance remain open. Fablet and Bouthemy introduced temporal co-occurrence [19], [20] that measures the probability of co-occurrence in the same image location of two normal velocities (normal flow magnitudes) separated by certain temporal intervals. Recently, Smith *et al.* dealt with video texture indexing using spatiotemporal wavelets [28]. Spatiotemporal wavelets can decompose motion into the local and global, according to the desired degree of detail.

Otsuka *et al.* [29] assumed that DTs can be represented by moving contours whose motion trajectories can be tracked. They considered trajectory surfaces within 3D spatiotemporal volume data, and extracted temporal and spatial features based on the tangent plane distribution. The spatial features include the directionality of contour arrangement and the scattering of contour placement. The temporal features characterize the uniformity of velocity components, the ash motion ratio and the occlusion ratio. These features were used to classify four DTs. Zhong and Scarlaro [30] modified [29] and used 3D edges in the spatiotemporal domain. Their DT features were computed for voxels taking into account the spatiotemporal gradient.

It appears that nearly all of the research on dynamic texture recognition has considered textures to be more or less 'homogeneous', i.e., the spatial locations of image regions are not taken into account. The dynamic textures are usually described with global features computed over the whole image, which greatly limits the applicability of dynamic texture recognition. Using only global features for face or facial expression recognition, for example, would not be effective since much of the discriminative information in facial images is local, such as mouth movements. In their recent work, Aggarwal *et al.* [31] adopted the Auto-Regressive and Moving Average (ARMA) framework of Doretto *et al.* [2] for video-based face recognition, demonstrating that temporal information contained in facial dynamics is useful for face recognition. In this approach, the use of facial appearance information is very limited. We are not aware of any dynamic texture based approaches to facial expression recognition [7], [8], [9].

III. Volume Local Binary Patterns

The main difference between dynamic texture (DT) and ordinary texture is that the notion of self-similarity, central to conventional image texture, is extended to the spatiotemporal domain [5]. Therefore combining motion and appearance to analyze DT is well justified. Varying lighting conditions greatly affect the gray scale properties of dynamic texture. At the same time, the textures may also be arbitrarily oriented, which suggests using rotation-invariant features. It is important, therefore, to define features which are robust with respect to gray scale changes, rotations and translation. In this paper, we propose the use of volume local binary patterns (VLBP, which could also be called 3D-LBP) to address these problems [32].

A. Basic VLBP

To extend LBP to DT analysis, we define dynamic texture V in a local neighborhood of a monochrome dynamic texture sequence as the joint distribution v of the gray levels of $3P+3(P>1)$ image pixels. P is the number of local neighboring points around the central pixel in one frame.

$$\begin{aligned} V = \; & v(g_{t_c-L,c},\ g_{t_c-L,0},\cdots,g_{t_c-L,P-1},\ g_{t_c,c},\ g_{t_c,0},\\ & \cdots,g_{t_c,P-1},\ g_{t_c+L,0},\cdots,g_{t_c+L,P-1},\ g_{t_c+L,c}). \end{aligned} \tag{1}$$

where the gray value $g_{t_c,c}$ corresponds to the gray value of the center pixel of the local volume neighborhood, $g_{t_c-L,c}$ and $g_{t_c+L,c}$ correspond to the gray value of the center pixel in the previous and posterior neighboring frames with time interval L; $g_{t,p}(t=t_c-L,t_c,t_c+L;p=0,\cdots,P-1)$ correspond to the gray values of P equally spaced pixels on a circle of radius $R(R>0)$ in image t, which form a circularly symmetric neighbor set.

Suppose the coordinates of $g_{t_c,c}$ are (x_c,y_c,t_c), the coordinates of $g_{t_c,p}$ are given by $(x_c+R\cos(2\pi p/P), y_c-R\sin(2\pi p/P), t_c)$, and the coordinates of $g_{t_c\pm L,p}$ are given by $(x_c+R\cos(2\pi p/P), y_c-R\sin(2\pi p/P), t_c\pm L)$. The values of the neighbors that do not fall exactly on pixels are estimated by bilinear interpolation.

To get gray-scale invariance, the distribution is thresholded similar to [6]. The gray value of the volume center pixel $(g_{t_c,c})$ is subtracted from the gray values of the circularly symmetric neighborhood $g_{t,p}(t=t_c-L,t_c,t_c+L;p=0,\cdots,P-1)$, giving:

$$\begin{aligned} V = \; & v(g_{t_c-L,c}-g_{t_c,c},\ g_{t_c-L,0}-g_{t_c,c},\cdots,\\ & g_{t_c-L,P-1}-g_{t_c,c},\ g_{t_c,c},\ g_{t_c,0}-g_{t_c,c},\cdots,\\ & g_{t_c,P-1}-g_{t_c,c},\ g_{t_c+L,0}-g_{t_c,c},\cdots,\\ & g_{t_c+L,P-1}-g_{t_c,c},\ g_{t_c+L,c}-g_{t_c,c}). \end{aligned} \tag{2}$$

Then we assume that differences $g_{t,p}-g_{t_c,c}$ are independent of $g_{t_c,c}$, which allow us to factorize (2):

$$\begin{aligned} V \approx \; & v(g_{t_c,c})v(g_{t_c-L,c}-g_{t_c,c},\ g_{t_c-L,0}-g_{t_c,c},\cdots,\\ & g_{t_c-L,P-1}-g_{t_c,c},\ g_{t_c,0}-g_{t_c,c},\cdots,g_{t_c,P-1}-g_{t_c,c},\\ & g_{t_c+L,0}-g_{t_c,c},\cdots,g_{t_c+L,P-1}-g_{t_c,c},\ g_{t_c+L,c}-g_{t_c,c}). \end{aligned}$$

In practice, exact independence is not warranted; hence, the factorized distribution is only an approximation of the joint distribution. However, we are willing to accept a possible small loss of information as it allows us to achieve invariance with respect to shifts in gray scale. Thus, similar to LBP in ordinary texture analysis [6], the distribution $v(g_{t_c,c})$ describes the overall luminance of the image, which is unrelated to the local image texture and, consequently, does not provide useful information for dynamic texture analysis. Hence, much of the information in the original joint gray level distribution (1) is conveyed by the joint difference distribution:

$$\begin{aligned} V_1 = \; & v(g_{t_c-L,c}-g_{t_c,c},\ g_{t_c-L,0}-g_{t_c,c},\cdots,\\ & g_{t_c-L,P-1}-g_{t_c,c},\ g_{t_c,0}-g_{t_c,c},\cdots,g_{t_c,P-1}-g_{t_c,c},\\ & g_{t_c+L,0}-g_{t_c,c},\cdots,g_{t_c+L,P-1}-g_{t_c,c},\ g_{t_c+L,c}-g_{t_c,c}). \end{aligned}$$

This is a highly discriminative texture operator. It records the occurrences of various patterns in the neighborhood of each pixel in a $(2(P+1)+P=3P+2)$-dimensional histogram.

We achieve invariance with respect to the scaling of the gray scale by considering simply the signs of the differences instead of their exact values:

$$\begin{array}{rl} V_2 = & v(s(g_{t_c-L,c}-g_{t_c,c}), s(g_{t_c-L,0}-g_{t_c,c}), \cdots, \\ & s(g_{t_c-L,P-1}-g_{t_c,c}), s(g_{t_c,0}-g_{t_c,c}), \cdots, \\ & s(g_{t_c,P-1}-g_{t_c,c}), s(g_{t_c+L,0}-g_{t_c,c}), \cdots, \\ & s(g_{t_c+L,P-1}-g_{t_c,c}), s(g_{t_c+L,c}-g_{t_c,c})). \end{array} \quad (3)$$

where $s(x) = \left\{ \begin{array}{l} 1, x \geq 0 \\ 0, x < 0 \end{array} \right\}$.

To simplify the expression of V_2, we use $V_2 = v(v_0, \cdots, v_q, \cdots, v_{3P+1})$, and q corresponds to the index of values in V_2 orderly. By assigning a binomial factor 2^q for each sign $s(g_{t,p}-g_{t_c,c})$, we transform Eq. (3) into a unique $VLBP_{L,P,R}$ number that characterizes the spatial structure of the local volume dynamic texture:

$$VLBP_{L,P,R} = \sum_{q=0}^{3P+1} v_q 2^q. \quad (4)$$

Fig. 1 shows the whole computing procedure for $VLBP_{1,4,1}$. We begin by sampling neighboring points in the volume, and then thresholding every point in the neighborhood with the value of the center pixel to get a binary value. Finally we produce the VLBP code by multiplying the thresholded binary values with weights given to the corresponding pixel and we sum up the result.

Let us assume we are given an $X \times Y \times T$ dynamic texture $(x_c \in \{0, \cdots, X-1\}, y_c \in \{0, \cdots, Y-1\}, t_c \in \{0, \cdots, T-1\})$. In calculating $VLBP_{L,P,R}$ distribution for this DT, the central part is only considered because a sufficiently large neighborhood cannot be used on the borders in this 3D space. The basic VLBP code is calculated for each pixel in the cropped portion of the DT, and the distribution of the codes is used as a feature vector, denoted by D:

$$\begin{array}{l} D = v(VLBP_{L,P,R}(x,y,t)), x \in \{\lceil R \rceil, \cdots, X-1-\lceil R \rceil\}, \\ y \in \{\lceil R \rceil, \cdots, Y-1-\lceil R \rceil\}, t \in \{\lceil L \rceil, \cdots, T-1-\lceil L \rceil\}. \end{array}$$

The histograms are normalized with respect to volume size variations by setting the sum of their bins to unity.

Because the dynamic texture is viewed as sets of volumes and their features are extracted on the basis of those volume textons, VLBP combines the motion and appearance to describe dynamic textures.

B. Rotation Invariant VLBP

Dynamic textures may also be arbitrarily oriented, so they also often rotate in the videos. The most important difference between rotation in a still texture image and DT is that the whole sequence of DT rotates around one axis or multi-axes (if the camera rotates during capturing), while the still texture rotates around one point. We cannot, therefore, deal with VLBP as a whole to get rotation invariant code as in [6], which assumed rotation around the center pixel in the static case. We first divide the whole VLBP code from (3) into 5 parts:

$$\begin{array}{rl} V_2 = & v([s(g_{t_c-L,c}-g_{t_c,c})], \\ & [s(g_{t_c-L,0}-g_{t_c,c}), \cdots, s(g_{t_c-L,P-1}-g_{t_c,c})], \\ & [s(g_{t_c,0}-g_{t_c,c}), \cdots, s(g_{t_c,P-1}-g_{t_c,c})], \\ & [s(g_{t_c+L,0}-g_{t_c,c}), \cdots, s(g_{t_c+L,P-1}-g_{t_c,c})], \\ & [s(g_{t_c+L,c}-g_{t_c,c})]). \end{array}$$

Then we mark those as $V_{preC}, V_{preN}, V_{curN}, V_{posN}, V_{posC}$ in order, and $V_{preN}, V_{curN}, V_{posN}$ represent the LBP code in the previous, current and posterior frames, respectively, while V_{preC} and V_{posC} represent the binary values of the center pixels in the previous and posterior frames.

$$LBP_{t,P,R} = \sum_{p=0}^{P-1} s(g_{t,p}-g_{t_c,c}) 2^p, t = t_c - L, t_c, t_c + L. \quad (5)$$

Using formula (5), we can get $LBP_{t_c-L,P,R}$, $LBP_{t_c,P,R}$ and $LBP_{t_c+L,P,R}$.

To remove the effect of rotation, we use:

$$\begin{array}{l} VLBP^{ri}_{L,P,R} = min\{(VLBP_{L,P,R} \ and \ 2^{3P+1}) \\ +ROL(ROR(LBP_{t_c+L,P,R}, i), 2P+1) \\ +ROL(ROR(LBP_{t_c,P,R}, i), P+1) \\ +ROL(ROR(LBP_{t_c-L,P,R}, i), 1) \\ +(VLBP_{L,P,R} \ and \ 1) | i = 0, 1, \cdots, P-1\} \end{array} \quad (6)$$

where $ROR(x,i)$ performs a circular bit-wise right shift on the P-bit number x i times [6], and $ROL(y,j)$ performs a bit-wise left shift on the $3P+2$-bit number y j times. In terms of image pixels, formula (6) simply corresponds to rotating the neighbor set in three separate frames clockwise and this happens synchronously so that a minimal value is selected as the VLBP rotation invariant code.

For example, for the original VLBP code $(1, 1010, 1101, 1100, 1)_2$, its codes after rotating clockwise 90, 180, 270 degrees are $(1, 0101, 1110, 0110, 1)_2$, $(1, 1010, 0111, 0011, 1)_2$ and $(1, 0101, 1011, 1001, 1)_2$ respectively. Their rotation invariant code should be $(1, 0101, 1011, 1001, 1)_2$, and not $(00111010110111)_2$ as obtained by using the VLBP as a whole.

In [6], Ojala *et al.* found that the vast majority of the LBP patterns in a local neighborhood are so called "uniform patterns". A pattern is considered uniform if it contains at most two bitwise transitions from 0 to 1 or vice versa when the bit pattern is considered circular. When using uniform patterns, all non-uniform LBP patterns are stored in a single bin in the histogram computation. This makes the length of the feature vector much shorter and allows us to define a simple version of rotation invariant LBP [6]. In the remaining sections, the superscript $riu2$ will be used to denote these features, while the superscript $u2$ means that the uniform patterns without rotation invariance are used. For example, $VLBP^{riu2}_{1,2,1}$ denotes rotation invariant $VLBP_{1,2,1}$ based on uniform patterns.

IV. Local Binary Patterns from Three Orthogonal Planes

In the proposed VLBP, the parameter P determines the number of features. A large P produces a long histogram, while a small P makes the feature vector shorter, but also means losing more information. When the number of neighboring points increases, the number of patterns for basic VLBP will become very large,

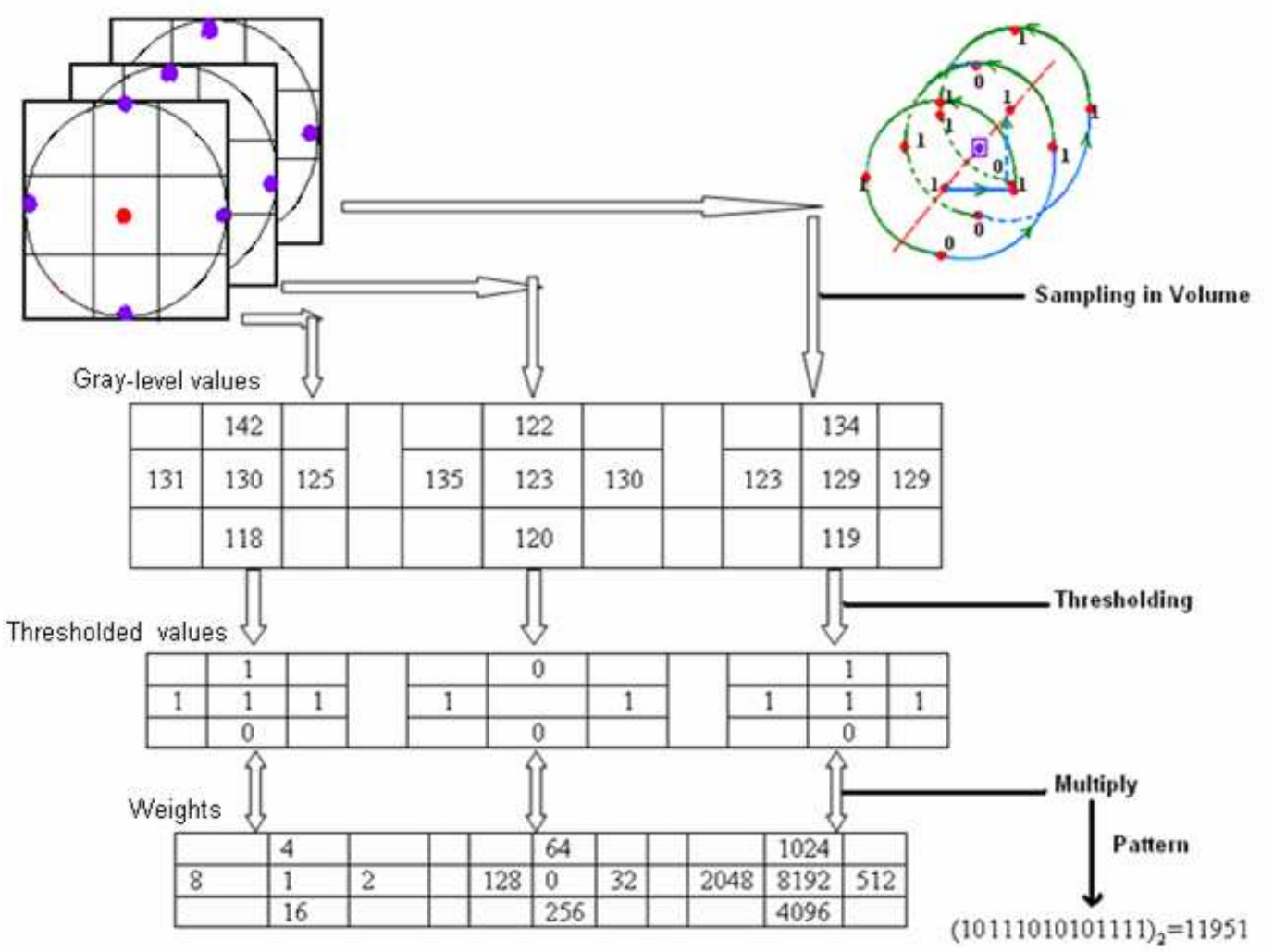

Fig. 1. Procedure of $VLBP_{1,4,1}$.

2^{3P+2}, as shown in Fig. 2. Due to this rapid increase, it is difficult to extend VLBP to have a large number of neighboring points, and this limits its applicability. At the same time, when the time interval $L > 1$, the neighboring frames with a time variance less than L will be omitted.

To address these problems, we propose simplified descriptors by concatenating local binary patterns on three orthogonal planes (LBP-TOP): XY, XT and YT, considering only the co-occurrence statistics in these three directions (shown in Fig. 3). Usually a video sequence is thought of as a stack of XY planes in axis T, but it is easy to ignore that a video sequence can also be seen as a stack of XT planes in axis Y and YT planes in axis X, respectively. The XT and YT planes provide information about the space-time transitions. With this approach, the number of bins is only $3 \cdot 2^P$, much smaller than 2^{3P+2}, as shown in Fig. 2, which makes the extension to many neighboring points easier and also reduces the computational complexity.

There are two main differences between VLBP and LBP-TOP. Firstly, the VLBP uses three parallel planes, of which only the middle one contains the center pixel. The LBP-TOP, on the other hand, uses three orthogonal planes which intersect in the center pixel. Secondly, VLBP considers the co-occurrences of all neighboring points from three parallel frames, which tends to

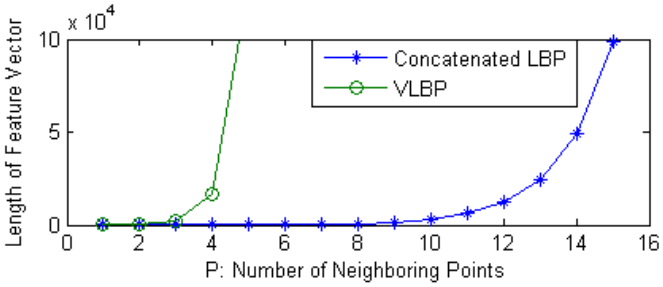

Fig. 2. The number of features versus the number of LBP codes.

make the feature vector too long. LBP-TOP considers the feature distributions from each separate plane and then concatenates them together, making the feature vector much shorter when the number of neighboring points increases.

To simplify the VLBP for DT analysis and to keep the number of bins reasonable when the number of neighboring points increases, the proposed technique uses three instances of co-occurrence statistics obtained independently from three orthogonal planes [33], as shown in Fig. 3. Because we do not know the motion direction of textures, we also consider the

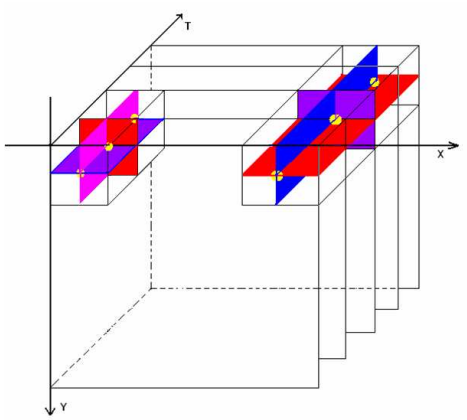

Fig. 3. Three planes in DT to extract neighboring points.

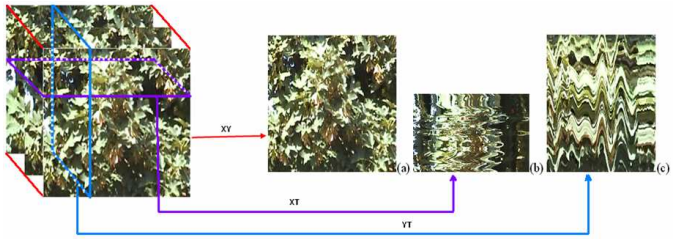

Fig. 4. (a) Image in XY plane (400×300) (b) Image in XT plane (400×250) in $y = 120$ (last row is pixels of $y = 120$ in first image) (c) Image in TY plane (250×300) in $x = 120$ (first column is the pixels of $x = 120$ in first frame).

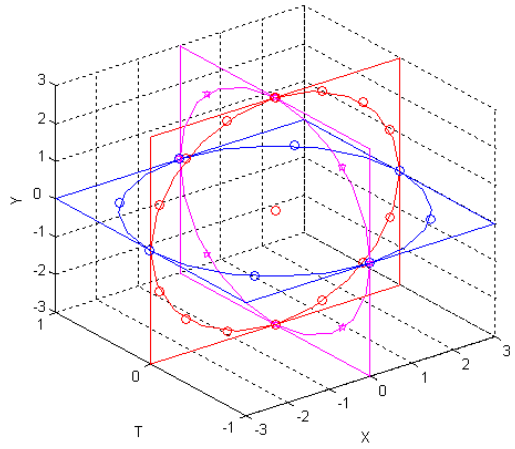

Fig. 6. Different radii and number of neighboring points on three planes.

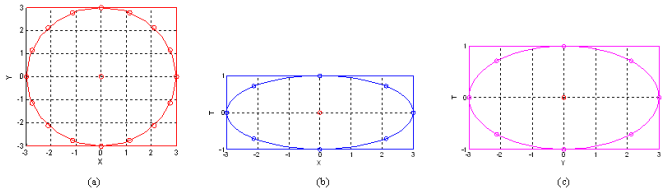

Fig. 7. Detailed sampling for Fig. 6. with $R_X = R_Y = 3$, $R_T = 1$, $P_{XY} = 16$, $P_{XT} = P_{YT} = 8$. (a) XY plane; (b) XT plane; (c) YT plane.

neighboring points in a circle, and not only in a direct line for central points in time. Compared with VLBP, not all the volume information, but only the features from three planes are applied. Fig. 4 demonstrates example images from three planes. (a) shows the image in the XY plane, (b) in the XT plane which gave the visual impression of one row changing in time, while (c) describes the motion of one column in temporal space. The LBP code is extracted from the XY, XT and YT planes, which are denoted as $XY - LBP$, $XT - LBP$ and $YT - LBP$, for all pixels, and statistics of three different planes are obtained, and then concatenated into a single histogram. The procedure is demonstrated in Fig. 5. In such a representation, DT is encoded by the $XY-LBP$, $XT-LBP$ and $YT-LBP$, while the appearance and motion in three directions of DT are considered, incorporating spatial domain information ($XY-LBP$) and two spatial temporal co-occurrence statistics ($XT - LBP$ and $YT - LBP$).

Setting the radius in the time axis to be equal to the radius in the space axis is not reasonable for dynamic textures. For instance, for a DT with an image resolution of over 300 by 300, and a frame rate of less than 12, in a neighboring area with a radius of 8 pixels in the X axis and Y axis the texture might still keep its appearance; however, within the same temporal intervals in the T axis, the texture changes drastically, especially in those DTs with high image resolution and a low frame rate. So we have different radius parameters in space and time to set. In the XT and YT planes, different radii can be assigned to sample neighboring points in space and time. With this approach the traditional circular sampling is extended to elliptical sampling.

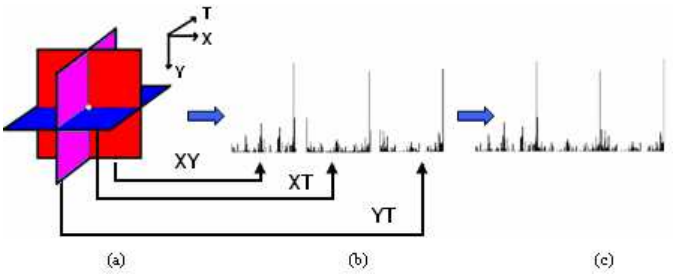

Fig. 5. (a) Three planes in dynamic texture (b) LBP histogram from each plane (c) Concatenated feature histogram.

More generally, the radii in axes X, Y and T, and the number of neighboring points in the XY, XT and YT planes can also be different, which can be marked as R_X, R_Y and R_T, P_{XY}, P_{XT} and P_{YT}, as shown in Fig. 6 and Fig. 7. The corresponding feature is denoted as $LBP - TOP_{P_{XY},P_{XT},P_{YT},R_X,R_Y,R_T}$. Suppose the coordinates of the center pixel $g_{t_c,c}$ are (x_c, y_c, t_c), the coordinates of $g_{XY,p}$ are given by $(x_c - R_X \sin(2\pi p/P_{XY}), y_c + R_Y \cos(2\pi p/P_{XY}), t_c)$, the coordinates of $g_{XT,p}$ are given by $(x_c - R_X \sin(2\pi p/P_{XT}), y_c, t_c - R_T \cos(2\pi p/P_{XT}))$, and the coordinates of $g_{YT,p}$ $(x_c, y_c - R_Y \cos(2\pi p/P_{YT}), t_c - R_T \sin(2\pi p/P_{YT}))$. This is different from the ordinary LBP widely used in many papers, and it extends the definition of LBP.

Let us assume we are given an $X\times Y\times T$ dynamic texture $(x_c \in \{0, \cdots, X-1\}, y_c \in \{0, \cdots, Y-1\}, t_c \in \{0, \cdots, T-1\})$. In calculating $LBP-TOP_{P_{XY},P_{XT},P_{YT},R_X,R_Y,R_T}$ distribution for this DT, the central part is only considered because a sufficiently large neighborhood cannot be used on the borders in this 3D space.

A histogram of the DT can be defined as

$$H_{i,j} = \sum_{x,y,t} I\left\{f_j(x,y,t) = i\right\}, \quad i = 0, \cdots, n_j - 1; j = 0, 1, 2. \tag{7}$$

in which n_j is the number of different labels produced by the LBP operator in the jth plane ($j = 0 : XY,\ 1 : XT\ and\ 2 : YT$), $f_i(x, y, t)$ expresses the LBP code of central pixel (x, y, t) in the jth plane, and $I\{A\} = \begin{cases} 1, if\ A\ is\ true; \\ 0, if\ A\ is\ false. \end{cases}$

When the DTs to be compared are of different spatial and temporal sizes, the histograms must be normalized to get a coherent description:

$$N_{i,j} = \frac{H_{i,j}}{\sum_{k=0}^{n_j - 1} H_{k,j}}. \tag{8}$$

In this histogram, a description of DT is effectively obtained based on LBP from three different planes. The labels from the XY plane contain information about the appearance, and in the labels from the XT and YT planes co-occurrence statistics of motion in horizontal and vertical directions are included. These three histograms are concatenated to build a global description of DT with the spatial and temporal features.

V. Local Descriptors for Facial Image Analysis

Local texture descriptors have gained increasing attention in facial image analysis due to their robustness to challenges such as pose and illumination changes. Recently, Ahonen *et al.* proposed a novel facial representation for face recognition from static images based on LBP features [34], [35]. In this approach, the face image is divided into several regions (blocks) from which the LBP features are extracted and concatenated into an enhanced feature vector. This approach is proving to be a growing success. It has been adopted and further developed by many research groups, and has been successfully used for face recognition, face detection and facial expression recognition [35]. All of these have applied LBP-based descriptors only for static images, i.e. they do not utilize temporal information as proposed in this paper.

In this section, a block-based approach for combining pixel-level, region-level and temporal information is proposed. Facial expression recognition is used as a case study, but a similar approach could be used for recognizing other specific dynamic events such as faces from video, for example. The goal of facial expression recognition is to determine the emotional state of the face, for example, happiness, sadness, surprise, neutral, anger, fear, and disgust, regardless of the identity of the face.

Most of the proposed methods use a mug shot of each expression that captures the characteristic image at the apex [10], [11], [12], [13]. However, according to psychologists [36], analyzing a sequence of images produces more accurate and robust recognition of facial expressions. Psychological studies have suggested that facial motion is fundamental to the recognition of facial expressions. Experiments conducted by Bassili [36] demonstrate that humans are better at recognizing expressions from dynamic images as opposed to mug shots. For using dynamic information to analyze facial expression, several systems attempt to recognize fine-grained changes in the facial expression. These are based on the Facial Action Coding System (FACS) developed by Ekman and Friesen [37] for describing facial expressions by action units (AUs). Some papers attempt to recognize a small set of prototypical emotional expressions, i.e. joy, surprise, anger, sadness, fear, and disgust, for example [14], [15], [16]. Yeasin *et al.* [14] used the horizontal and vertical components of the flow as features. At the frame level, the k-NN rule was used to derive a characteristic temporal signature for every video sequence. At the sequence level, discrete Hidden Markov Models (HMMs) were trained to recognize the temporal signatures associated with each of the basic expressions. This approach cannot deal with illumination variations, however. Aleksic and Katsaggelos [15] proposed facial animation parameters as features describing facial expressions, and utilized multi-stream HMMs for recognition. The system is complex, making it difficult to perform in real-time. Cohen *et al.* [16] introduced a Tree-Augmented-Naive Bayes classifier for recognition. However, they only experimented on a set of five people, and the accuracy was only around 65% for person-independent evaluation.

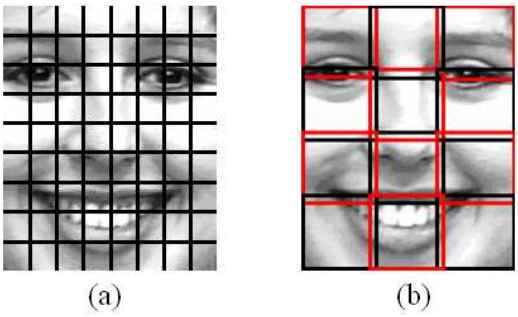

Fig. 8. (a) non-overlapping blocks (9×8) (b) overlapping blocks (4×3, overlap size = 10).

Considering the motion of the facial region, we propose region-concatenated descriptors on the basis of the algorithm in Section 4 for facial expression recognition. An LBP description computed over the whole facial expression sequence encodes only the occurrences of the micro-patterns without any indication about their locations. To overcome this effect, a representation in which the face image is divided into several non-overlapping or overlapping blocks is introduced. Fig. 8 (a) depicts non-overlapping 9×8 blocks and Fig. 8 (b) overlapping 4×3 blocks with an overlap of 10 pixels, respectively. The LBP-TOP histograms in each block are computed and concatenated into a single histogram, as Fig. 9 shows. All features extracted from each block volume are connected to represent the appearance and motion of the facial expression sequence, as shown in Fig. 10.

In this way, we effectively have a description of the facial expression on three different levels of locality. The labels (bins) in the histogram contain information from three orthogonal planes, describing appearance and temporal information at the pixel level. The labels are summed over a small block to produce information on a regional level expressing the characteristics for the appearance and motion in specific locations, and all information from the regional level is concatenated to build a global description of the face and expression motion.

Ojala *et al.* noticed in their experiments with texture images that uniform patterns account for slightly less than 90% of all patterns when using the (8,1) neighborhood and for around 70% in the (16,2) neighborhood [6]. In our experiments, for histograms from the two temporal planes, the uniform patterns account for slightly less than those from the spatial plane, but also follow the rule that most of the patterns are uniform. So the following is the notation for the LBP operator: $LBP -$

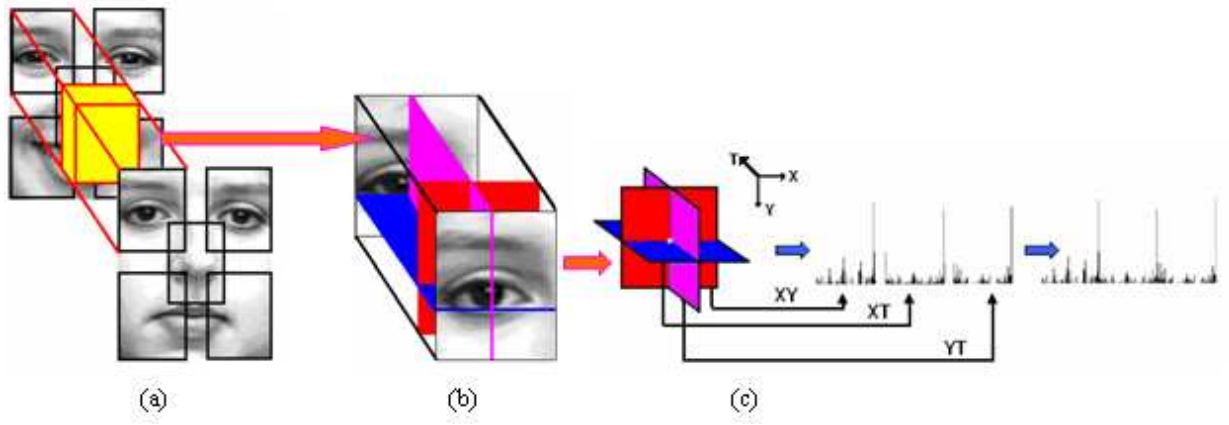

Fig. 9. Features in each block volume. (a) Block volumes; (b) LBP features from three orthogonal planes; (c) Concatenated features for one block volume with the appearance and motion

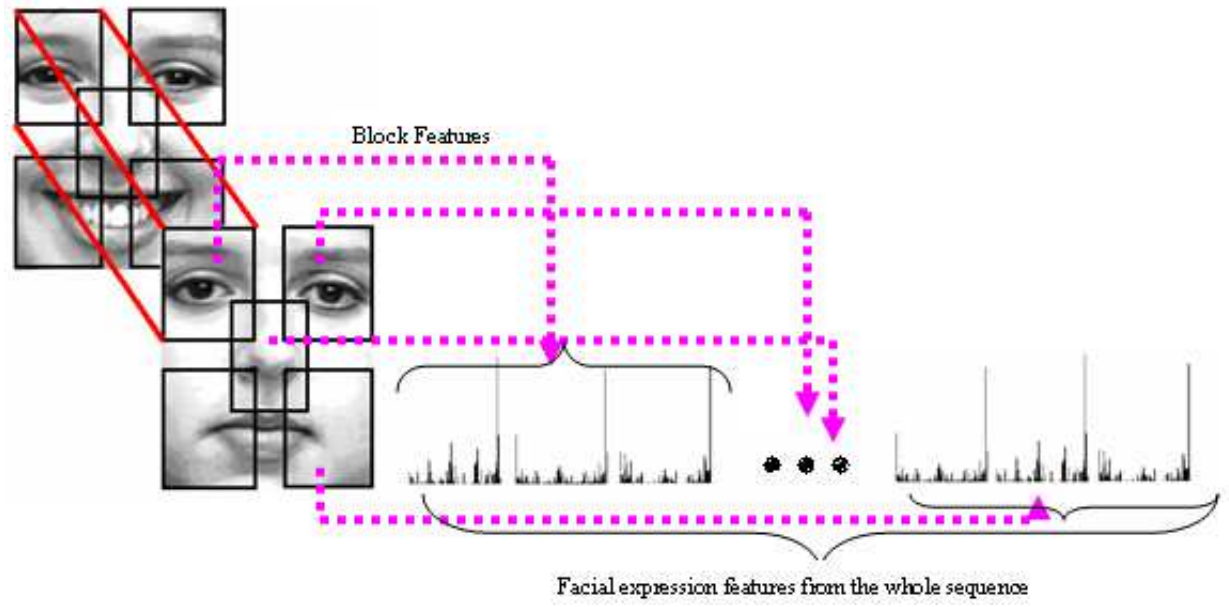

Fig. 10. Facial expression representation.

$TOP^{u2}_{P_{XY},P_{XT},P_{YT},R_X,R_Y,R_T}$ is used. The subscript represents using the operator in a $(P_{XY}, P_{XT}, P_{YT}, R_X, R_Y, R_T)$ neighborhood. Superscript $u2$ stands for using only uniform patterns and labeling all remaining patterns with a single label.

VI. Experiments

The new large dynamic texture database DynTex was used to evaluate the performance of our DT recognition methods. Additional experiments with the widely used MIT dataset [1], [3] were also carried out. In facial expression recognition, the proposed algorithms were evaluated on the Cohn-Kanade Facial Expression Database [9].

A. DT recognition

1) Measures

After obtaining the local features on the basis of different parameters of L, P and R for VLBP, or $P_{XY}, P_{XT}, P_{YT}, R_X, R_Y, R_T$ for LBP-TOP, a leave-one-group-out classification test was carried out for DT recognition based on the nearest class. If one DT includes m samples, we separate all DT samples into m groups, evaluate performance by letting each sample group be unknown and train on the remaining $m-1$ samples groups. The mean VLBP features or LBP-TOP features of all the $m-1$ samples are computed as the feature for the class. The omitted sample is classified or verified according to its difference with respect to the class using the k nearest neighbor method $(k = 1)$.

In classification, the dissimilarity between a sample and model feature distribution is measured using the log-likelihood statistic: $L(S, M) = -\sum_{b=1}^{B} S_b \log M_b$, where B is the number of bins and S_b and M_b correspond to the sample and model probabilities at bin b, respectively. Other dissimilarity measures like histogram intersection or Chi square distance could also be used.

When the DT is described in the XY, XT and YT planes, it can be expected that some of the planes contain more useful information than others in terms of distinguishing between DTs. To take advantage of this, a weight can be set for each plane based on the importance of the information it contains. The weighted log-likelihood statistic is defined as: $L_w(S, M) =$

Fig. 11. DynTex database.

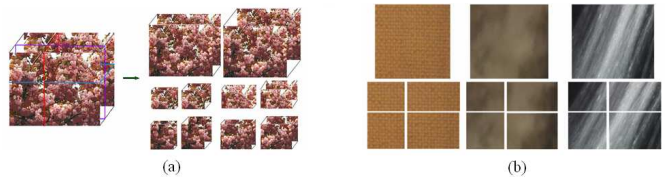

Fig. 12. (a) Segmentation of DT sequence. (b) Examples of segmentation in space.

$-\sum_{i,j}(w_j S_{j,i}\log M_{j,i})$, in which w_j is the weight of plane j.

2) Multi-resolution analysis

By altering L, P and R for VLBP, $P_{XY}, P_{XT}, P_{YT}, R_X, R_Y, R_T$ for LBP-TOP, we can realize operators for any quantization of the time interval, the angular space and spatial resolution. Multi-resolution analysis can be accomplished by combining the information provided by multiple operators of varying (L, P, R) and $(P_{XY}, P_{XT}, P_{YT}, R_X, R_Y, R_T)$.

The most accurate information would be obtained by using the joint distribution of these codes [6]. However, such a distribution would be overwhelmingly sparse with any reasonable size of image and sequence. For example, the joint distribution of $VLBP^{riu2}_{1,4,1}$, $VLBP^{riu2}_{2,4,1}$ and $VLBP^{riu2}_{2,8,1}$ would contain $16 \times 16 \times 28 = 7168$ $bins$. So, only the marginal distributions of the different operators are considered, even though the statistical independence of the outputs of the different VLBP operators or simplified concatenated bins from three planes at a central pixel cannot be warranted.

In our study, we perform straightforward multi-resolution analysis by defining the aggregate dissimilarity as the sum of individual dissimilarity between the different operators on the basis of the additivity property of the log-likelihood statistic [6]: $L_N = \sum_{n=1}^{N} L(S^n, M^n)$, where N is the number of operators and S^n and M^n correspond to the sample and model histograms extracted with operator $n (n = 1, 2, \cdots, N)$.

3) Experimental setup

The DynTex dataset (http://www.cwi.nl/projects/dyntex/) is a large and varied database of dynamic textures. Fig. 11 shows example DTs from this dataset. The image size is 400×300.

In the experiments on the DynTex database, each sequence was divided into 8 non-overlapping subsets, but not half in X,Y and T. The segmentation position in volume was selected randomly. For example in Fig. 12, we select the transverse plane with $x = 170$, the lengthways plane with $y = 130$, and in the time direction with $t = 100$. These eight samples do not overlap each other, and they have different spatial and temporal information. Sequences with the original size but only cut in the time direction are also included in the experiments. So we can get 10 samples of each class and all samples are different in image size and sequence length from each other. Fig. 12 (a) demonstrates the segmentation, and Fig. 12 (b) shows some segmentation examples in space. We can see that this sampling method increases the challenge of recognition in a large database.

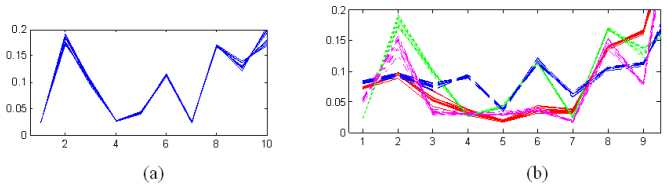

Fig. 13. Histograms of dynamic textures. (a) Histograms of up-down tide with 10 samples for $VLBP^{riu2}_{2,2,1}$. (b) Histograms of four classes each with 10 samples for $VLBP^{riu2}_{2,2,1}$.

4) Results of VLBP

Fig. 13 (a) shows the histograms of 10 samples of a dynamic texture using $VLBP^{riu2}_{2,2,1}$. We can see that for different samples of the same class, their VLBP codes are very similar to each other, even if they are different in spatial and temporal variation. Fig. 13 (b) depicts histograms of 4 classes each with 10 samples as in Fig. 12 (a). We can clearly see that the VLBP features have good similarity within classes and good discrimination between classes.

Table 1 presents the overall classification rates. The selection of optimal parameters is always a problem. Most approaches get locally optimal parameters by experiments or experience. According to our earlier studies on LBP, e.g. [6,10,34,35], the best radii are usually not bigger than 3, and the number of neighboring points (P) is 2^n $(n = 1, 2, 3, \cdots)$. In our proposed VLBP, when the number of neighboring points increases, the number of patterns for basic VLBP will become very large: 2^{3P+2}. Due to this rapid increase the feature vector will soon become too long to handle. Therefore only the results for $P = 2$ and $P = 4$ are given in Table 1. Using all 16384 bins of the basic $VLBP_{2,4,1}$ provides a 94.00% rate, while $VLBP_{2,4,1}$ with $u2$ gives a good result of 93.71% using only 185 bins. When using rotation invariant $VLBP^{ri}_{2,4,1}$(4176 bins), we get a result of 95.71%. With more complex features or multi-resolution analysis, better results could be expected. For example, the multi-resolution features $VLBP^{riu2}_{2,2,1+2,4,1}$ obtain a good rate of 90.00%, better than the results from the respective features $VLBP^{riu2}_{2,2,1}$(83.43%) and $VLBP^{riu2}_{2,4,1}$ (85.14%). However, when using multi-resolution analysis for basic patterns, the results are not improved, partly because the feature vectors are too long.

It can be seen that the basic VLBP performs very well, but it does not allow the use of many neighboring points P. A higher value for P is shown to provide better results. With uniform patterns, the feature vector length can be reduced without much loss in performance. Rotation invariant features perform almost as well as the basic VLBP for these textures. They further reduce the feature vector length and can handle the recognition of DTs after rotation.

5) Results for LBP-TOP

Table 2 presents the overall classification rates for LBP-TOP. The first three columns give the results using only one histogram from the corresponding plane; they are much lower than those from direct concatenation (fourth column) and weighed measures (fifth column). Moreover, the weighted measures of three histograms achieved better results than direct concatenation because they consider the different contributions of features. When using

TABLE I

RESULTS (%) IN DYNTEX DATASET (SUPERSCRIPT $riu2$ MEANS ROTATION INVARIANT UNIFORM PATTERNS, $u2$ IS THE UNIFORM PATTERNS WITHOUT ROTATION INVARIANCE, AND ri REPRESENTS THE ROTATION INVARIANT PATTERNS. THE NUMBERS INSIDE THE PARENTHESES DEMONSTRATE THE LENGTH OF CORRESPONDING FEATURE VECTORS).

	$VLBP_{1,2,1}$	$VLBP_{2,2,1}$	$VLBP_{1,4,1}$	$VLBP_{2,4,1}$
Basic	91.71 (256)	91.43 (256)	94.00 (16384)	94.00 (16384)
$u2$	87.71 (59)	90.00 (59)	93.43 (185)	93.71 (185)
ri	89.43 (144)	90.57 (144)	93.14 (4176)	95.71 (4176)
$riu2$	83.43 (10)	83.43 (10)	88.57 (16)	85.14 (16)
multi-resolution	$VLBP^{riu2}_{2,2,1+2,4,1}$: 90.00 ($VLBP^{riu2}_{2,2,1}$: 83.43, $VLBP^{riu2}_{2,4,1}$: 85.14)		$VLBP^{riu2}_{2,2,1+1,2,1}$: 86.00 ($VLBP^{riu2}_{1,2,1}$: 83.43, $VLBP^{riu2}_{2,2,1}$: 83.43)	

TABLE II

RESULTS (%) IN DYNTEX DATASET (VALUES IN SQUARE BRACKET ARE WEIGHTS ASSIGNED FOR THREE SETS OF LBP BINS).

$LBP-TOP$	XY	XT	YT	Con	Weighted
$8,8,8,1,1,1\ u2$	92.86	88.86	89.43	94.57	96.29[4,1,1]
$2,2,2,1,1,1 Basic$	70.86	60.86	78.00	88.86	90.86[3,1,4]
$4,4,4,1,1,1 Basic$	94.00	86.29	91.71	93.71	94.29[6,1,5]
$8,8,8,1,1,1 Basic$	95.14	90.86	90.00	95.43	97.14[5,2,1]
$8,8,8,3,3,3 Basic$	90.00	91.17	94.86	95.71	96.57[1,2,5]
$8,8,8,3,3,1 Basic$	89.71	91.14	92.57	94.57	95.14[1,2,3]

only the $LBP-TOP_{8,8,8,1,1,1}$, we get good results of 97.14% with the weight [5,2,1] for the three histograms. Because the frame rate or the resolution in the temporal axis in the DynTex is high enough for using the same radius, it can be seen from Table 2 that results from a smaller radius in the time axis T (the last row) are as good as the same radius to the other two planes.

For selecting the weights, a heuristic approach was used which takes into account the different contributions of the features from the three planes. First, by computing the recognition rates for each plane separately, we get three rates $X = [x1, x2, x3]$; then, we can assume that the lower the minimum rate, the smaller the relative advantage will be. For example, the recognition rate improvement from 70% to 80% is better than from 50% to 60%, even though the differences are both 10%. The relative advantage of the two highest rates to the lowest one can now be computed as: $Y = (X - min(X))/((100 - min(X))/10)$. Finally, considering that the weight of the lowest rate is 1, the weights of the other two histograms can be obtained according to a linear relationship of their differences to that with the lowest rate. The following presents the final computing step, and W is the generated weight vector corresponding to the three histograms. $Y1 = round(Y)$; $Y2 = (Y \times ((max(Y1) - 1))/max(Y) + 1$; $W = round(Y2)$.

As demonstrated in Table 2, the larger the number of neighboring points, the better the recognition rate. For $LBP-TOP_{8,8,8,1,1,1}$, the result 97.14% is much better than that for $LBP-TOP_{4,4,4,1,1,1}$ (94.29%) and $LBP-TOP_{2,2,2,1,1,1}$ (90.86%). It should be noted, however, that $P = 4$ can be preferred in some cases because it generates a feature vector with much lower dimensionality than $P = 8$ (48 vs. 768), while it does not suffer too much loss in accuracy (94.29% vs. 97.14%). So the choice of optimal P really depends on the accuracy requirement as well as the size of data set.

As Table 1 shows, an accuracy of 91.71% is obtained for VLBP using $P = 2$ with a feature vector length of 256. The length of the LBP-TOP feature vector is only 12, achieving 90.86% accuracy. For $P = 4$, the long VLBP feature vector (16384) provides an accuracy of 94.00%, while the LBP-TOP achieves 94.29% with a feature vector of 48 elements. This clearly demonstrates that the LBP-TOP is a more effective representation for dynamic textures.

Fig. 14. Examples of misclassified sequences.

There were two kinds of commonly occuring misclassifications. One of these is the different sequences of similar classes. For example, the two steam sequences (Fig. 14(a)) and two parts (TopLeft and TopRight in the blocks) of the steam context in Figs. 14(b) were discriminated into the same class. Actually, they should be thought of as the same DT. Therefore, our algorithm considering them as one class seems to prove its effectiveness in another aspect. The other one is a mixed DT, as shown in Fig. 14(c), which includes more than one dynamic texture: water and shaking grass. So it shows both characteristics of these two dynamic textures. If we think of (a) and the top-left, top-right parts of (b) as the same class, our resulting recognition rate using concatenated $LBP-TOP_{8,8,8,1,1,1}$ is 99.43%.

Our results are very good compared to the state-of-the-art. In [25], a classification rate of 98.1% was reported for 26 classes of the DynTex database. However, their test and training samples were only different in the length of the sequence, but the spatial variation was not considered. This means that their experimental setup was much simpler. When we experimented using all 35 classes with samples having the original image size and only different in sequence length, a 100% classification rate using $VLBP^{u2}_{1,8,1}$ or using $LBP-TOP_{8,8,8,1,1,1}$ was obtained.

We also performed experiments on the MIT dataset [1] using a similar experimental setup as with DynTex, obtaining a 100% accuracy both for the VLBP and LBP-TOP. None of the earlier methods have reached the same performance, see for example [1], [3], [25], [29]. Except for [3], which used the same segmentation as us but with only 10 classes, all other papers used simpler datasets which did not include the variation in space and time.

Comparing VLBP and LBP-TOP, they both combine motion and appearance together, and are robust to translation and illumination variations. Their differences are: a) the VLBP considers the co-occurrence of all the neighboring points in three frames of volume at the same time, while LBP-TOP only considers the

Fig. 15. Examples of in-plane rotated faces.

three orthogonal planes making it easy to be extended to use more neighboring information; b) when the time interval $L > 1$, the neighboring frames with a time variance of less than L will be missed out in VLBP, but the latter method still keeps the local information from all neighboring frames; c) computation of the latter is much simpler with the complexity $O(XYT \cdot 2^P)$, compared to the VLBP with $O(XYT \cdot 2^{3P})$.

B. Facial expression recognition experiments

1) Dataset

The Cohn-Kanade facial expression database [9] consists of 100 university students ranging in age from 18 to 30 years. Sixty-five percent were female, fifteen percent African-American, and three percent Asian or Latino. The subjects were instructed by an experimenter to perform a series of 23 facial displays that included single action units and combinations of action units, six of which were based on descriptions of prototypical emotions of anger, disgust, fear, joy, sadness, and surprise. The image sequences from neutral to the target display were digitized into 640 by 480 or 490 pixel arrays with an 8-bit precision for grayscale values. The video rate is 30 frames per second. For our study, 374 sequences from the dataset are selected from the database for basic emotional expression recognition. The selection criterion was that any sequence to be labeled was one of the six basic emotions. The sequences came from 97 subjects, with one to six emotions per subject. The positions of the two eyes in the first frame of each sequence were given manually, and then these positions were used to determine the facial area for the whole sequence. The whole sequence was used to extract the proposed spatiotemporal LBP features.

Just the positions of the eyes from the first frame of each sequence were used for alignment, and this alignment was used not only for the first frame, but also for the whole sequence. Though there may be some translation - in-plane rotation (as shown in Fig. 15) and out-of-plane rotation (subjects S045 and S051 in the database, permission not granted to be shown in publications) of the head - no further alignment of facial features, such as alignment of the mouth, was performed in our algorithm. We do not need to 1) track faces or eyes in the following frames as in [38], 2) segment eyes and outer-lips [15], 3) select constant illumination [15] or perform illumination correction [14], 4) align facial features with respect to the canonical template [14] or normalize the faces to a fixed size as done by most of the papers [11], [12], [13], [38]. In this way our experimental setup is more realistic. Moreover, there are also some illumination and skin color variations (as shown in Fig. 16). Due to our methods' robustness to monotonic gray-level changes, there was no need for image correction by histogram equalization or gradient correction, for example, which is normally needed as a preprocessing step in facial image analysis.

Fig. 16. Variation of illumination.

TABLE III
OVERLAPPING RATIO BEFORE AND AFTER OVERLAPPING ADJUST FOR 9X8 BLOCKS

$ra(\%)$		30	40	50	60	70	80
$ra'(\%)$	height	23.7	29.5	34.6	39.1	43.2	46.8
	width	23.8	29.6	34.8	39.3	43.4	47.1

2) Evaluation and comparison

As expected, the features from the XY plane contribute less than features extracted from the XT and YT planes. This is because the facial expression is different from ordinary dynamic textures, its appearance features are not as important as those of DTs. The appearance of DT can help greatly in recognition, but the appearance of a face includes both identity and expression information, which could make accurate expression classification more difficult. Features from the XT and YT planes explain more about the motion of facial muscles.

After experimenting with different block sizes and overlapping ratios, as shown in Fig. 17, we chose to use 9×8 blocks in our experiments. The best results were obtained with an overlap ratio of 70% of the original non-overlapping block size. We use the overlapping ratio with the original block size to adjust and get the new overlapping ratio. Suppose the overlap ratio to the original block width is ra, after adjusting, the new ratio is ra', and the following equation represents the ra' in height.

$$ra' = \frac{ra \cdot h/r}{\frac{(ra \cdot h \cdot (r-1))/r+h}{r}} = \frac{ra \cdot h}{(ra \cdot h \cdot (r-1))/r+h} = \frac{ra \cdot h \cdot r}{ra \cdot h(r-1)+h \cdot r} = \frac{ra \cdot r}{ra \cdot (r-1)+r}$$

where, h is the height of the block, and r is the row number of blocks. So the final overlapping ratio corresponding to 70% of the original block size is about 43%, as Table 3 shows.

For evaluation, we separated the subjects randomly into n groups of roughly equal size and did a "leave one group out" cross validation which also could be called an "n-fold cross-validation" test scheme. They are subject-independent. The same subjects did not always appear in both training and testing. The testing was therefore done with "novel faces" and was person-independent.

A support vector machine (SVM) classifier was selected since it is well founded in statistical learning theory and has been successfully applied to various object detection tasks in computer vision. Since SVM is only used for separating two sets of points, the six-expression classification problem is decomposed into 15 two-class problems (happiness-surprise, anger-fear, sadness-disgust, etc.), then a voting scheme is used to accomplish recognition. Sometimes more than one class gets the highest number of votes. In this case, 1-NN template matching is applied to these classes to reach the final result. This means that in training, the spatiotemporal LBP histograms of face sequences belonging to a given class are averaged to generate a histogram template for that class. In recognition, a nearest-neighbor classifier is adopted, i.e. the VLBP or LBP-TOP histogram of the input sequence sample s is classified to the nearest class template n: $L(s,n) < L(s,c)$ for all $c \neq n$ (c and n are the indices of the expression classes

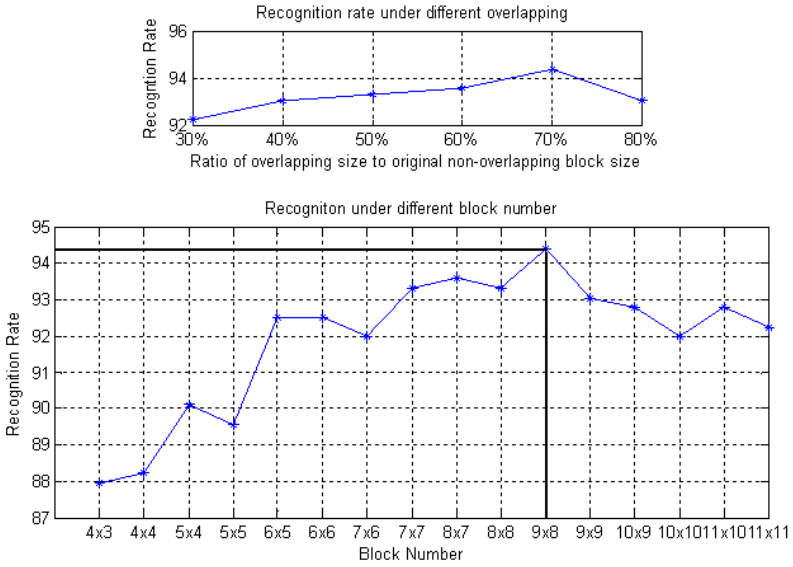

Fig. 17. Recognition with $LBP-TOP^{u2}_{8,8,8,3,3,3}$ under different overlapping sizes (top) and number of blocks (bottom).

having the same highest number of votes in the SVM classifier).

Here, after the comparison of linear, polynomial and RBF kernels in experiments, we use the second degree polynomial kernel function $K(\cdot,\cdot)$ defined by: $K(x,t_i)=(1+x\cdot t_i)^2$, which provided the best results.

We did some experiments using $P=2,4,8,16$ with different radii, overlapping rates and number of blocks. The value $P=8$ gave better results than the other values. Then lots of facial expression recognition experiments were performed with the value $P=8$, radii 1 and 3, overlapping rates from 30% to 80%, and number of blocks with $m\times n(m=4,\cdots,11;n=m-1,m)$. The experimental results given in Fig. 18 and Table 4 were selected to give a concise presentation.

Table 4 presents the results of two-fold cross-validation for each expression using different features. As we can see, a larger number of features does not necessarily provide better results; for example, the recognition rate of $LBP-TOP^{u2}_{16,16,16,3,3,3}$ is lower than that of the $LBP-TOP^{u2}_{8,8,8,3,3,3}$. With a smaller number of features, $LBP-TOP^{u2}_{8,8,8,3,3,3}$ (177 patterns per block) outperforms $VLBP_{3,2,3}$ (256 patterns per block). The combination of $LBP-TOP^{u2}_{8,8,8,3,3,3}$and $VLBP_{3,2,3}$ (last row) does not improve much compared to the results with separate features. It also can be seen that expressions labeled as surprise, happiness, sadness and disgust are recognized with very high accuracy (94.5% - 100%), whereas fear and anger present a slightly lower recognition rate.

Fig. 18 shows a comparison to some other dynamic analysis approaches using the recognition rates given in each paper. It should be noted that the results are not directly comparable due to different experimental setups, preprocessing methods, the number of sequences used etc., but they still give an indication of the discriminative power of each approach. In the left part of the figure are the detailed results for every expression. Our method outperforms the other methods in almost all cases, being the clearest winner in the case of fear and disgust. The right part of the figure gives the overall evaluation. As we can see, our algorithm works best with the greatest number of people and sequences. Table 5 compares our method with other static and dynamic analysis methods in terms of the number of people, the number of sequences and expression classes, with different measures, providing the overall results obtained with the Cohn-Kanade facial expression database. We can see that with the experiments on the greatest number of people and sequences, our results are the best ones compared not only to the dynamic methods but also to the static ones.

Table 6 summarizes the confusion matrix obtained using a ten-fold cross-validation scheme on the Cohn-Kanade facial expression database. The model achieves a 96.26% overall recognition rate of facial expressions. Table 7 gives some misclassified examples. The pair sadness and anger is difficult even for a human to recognize accurately. The discrimination of happiness and fear failed because these expressions had a similar motion of the mouth.

The experimental results clearly show that our approach outperforms the other dynamic and static methods. Our approach is quite robust with respect to variations of illumination, as seen from the pictures in Fig. 16. It also performed well with some in-plane and out-of-plane rotated sequences. This demonstrates robustness to errors in alignment.

VII. Conclusion

A novel approach to dynamic texture recognition was proposed. A volume LBP method was developed to combine the motion and appearance together. A simpler LBP-TOP operator based on concatenated LBP histograms computed from three orthogonal planes was also presented, making it easy to extract co-occurrence features from a larger number of neighboring points. Experiments on two DT databases with a comparison to the state-of-the-art results showed that our method is effective for DT recognition. Classification rates of 100% and 95.7% using VLBP, and 100% and 97.1% using LBP-TOP were obtained for the MIT and DynTex databases, respectively, using more difficult experimental setups than in the earlier studies. Our approach is computationally simple and robust in terms of grayscale and rotation variations, making it very promising for real application problems. The

TABLE IV
RESULTS (%) OF DIFFERENT EXPRESSIONS FOR TWO-FOLD CROSS-VALIDATION

	Surprise	Happiness	Sadness	Fear	Anger	Disgust	Total
$LBP-TOP^{u2}_{4,4,4,3,3,3}$	98.65	89.11	90.41	83.93	93.75	92.11	91.18
$LBP-TOP^{u2}_{8,8,8,3,3,3}$	100.00	97.03	94.52	85.71	84.38	97.37	94.38
$LBP-TOP^{u2}_{8,8,8,1,1,1}$	97.30	90.10	89.04	82.14	84.38	97.37	90.37
$LBP-TOP^{u2}_{16,16,16,3,3,3}$	98.65	91.10	89.04	76.80	75.00	92.10	88.77
$VLBP_{3,2,3}$	95.95	94.06	89.04	83.93	87.50	92.10	91.18
$LBP-TOP^{u2}_{8,8,8,3,3,3}+VLBP_{3,2,3}$	100.00	97.03	94.52	89.29	87.50	97.37	95.19

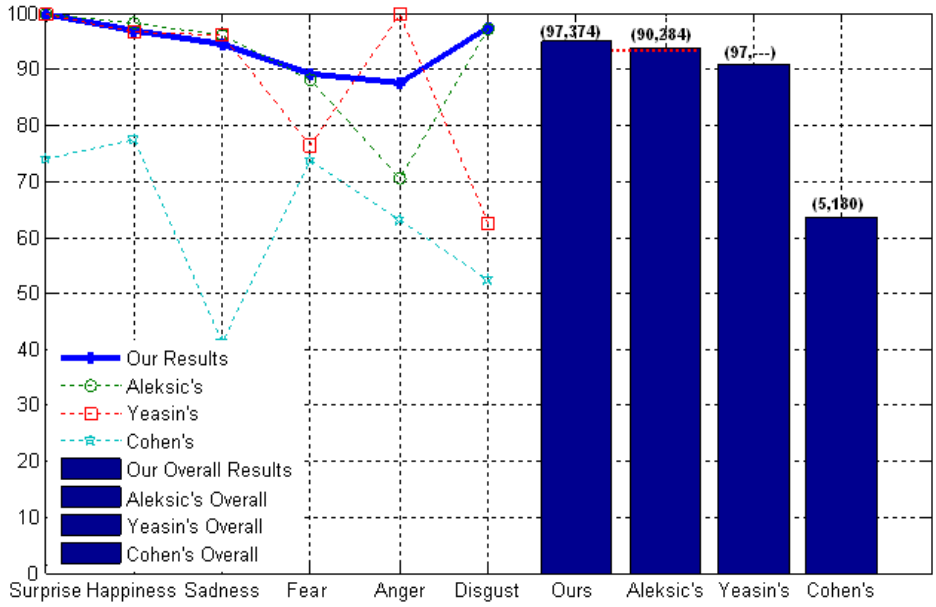

Fig. 18. Comparison with recent dynamic facial expression recognition methods. The left line graphs are recognition results for each expression, while the right histograms are the overall rate. The numbers in the brackets express the number of people and sequences used in different approaches. In Yeasin's work, the number of sequences in experiments is not mentioned.

TABLE V
COMPARISON WITH DIFFERENT APPROACHES

	PeopleNum	SequenceNum	ClassNum	Dynamic	Measure	Recognition Rate (%)
[11]	96	320	7(6)	N	10-fold	88.4 (92.1)
[12]	90	313	7	N	10-fold	86.9
[13]	90	313	7	N	leave-one-subject-out	93.8
[38]	97	375	6	N	—	93.8
[14]	97	—	6	Y	five-fold	90.9
[15]	90	284	6	Y	—	93.66
Ours	97	374	6	Y	two-fold	95.19
Ours	97	374	6	Y	10-fold	96.26

processing is done locally catching the transition information, which means that our method could also be used for segmenting dynamic textures. With this the problems caused by sequences with more than one dynamic texture could be avoided.

A block-based method, combining local information from the pixel, region and volume levels, was proposed to recognize specific dynamic events such as facial expressions in sequences. In experiments on the Cohn-Kanade facial expression database, our $LBP-TOP^{u2}_{8,8,8,3,3,3}$ feature gave an excellent recognition accuracy of 94.38% in two-fold cross-validation. Combining this feature with the $VLBP_{3,2,3}$ resulted in further improvement achieving 95.19% in two-fold and 96.26% in ten-fold cross-validation, respectively. These results are better than those obtained in earlier studies, even though in our experiments we used simpler image preprocessing and a larger number of people (97) and sequences (374) than most of the others have used. Our approach was shown to be robust with respect to errors in face alignment, and it does not require error prone segmentation of facial features such as lips. Furthermore, no gray-scale normalization is needed prior to applying our operators to the face images.

TABLE VI
CONFUSION MATRIX FOR $LBP-TOP^{u2}_{8,8,8,3,3,3}+VLBP_{3,2,3}$ FEATURES

	Surprise	Happiness	Sadness	Fear	Anger	Disgust
Surprise	**98.65**		1.35			
Happiness		**96.04**		2.97		0.99
Sadness	1.37		**95.89**		2.74	
Fear		3.57	1.79	**94.64**		
Anger			3.12		**96.88**	
Disgust			2.63		2.63	**94.74**

TABLE VII
EXAMPLES OF MISCLASSIFIED EXPRESSIONS

Apex image in sequences				
Ground truth	Happiness	Fear	Sadness	Anger
Misclassified results	Fear	Happiness	Anger	Sadness

ACKNOWLEDGMENTS

This work was supported in part by the Academy of Finland. The authors would like to thank Dr. Renaud Péteri for providing the DynTex database and Dr. Jeffrey Cohn for providing the Cohn-Kanade facial expression database used in experiments. The authors appreciate the helpful comments and suggestions of Prof. Janne Heikkilä, Timo Ahonen, Esa Rahtu and the anonymous reviewers.

REFERENCES

[1] M. Szummer and R.W. Picard, "Temporal Texture Modeling," *Proc. IEEE Conf. Image Processing*, vol. 3, pp. 823-826, 1996.

[2] G. Doretto, A. Chiuso, S. Soatto, and Y.N. Wu, "Dynamic Textures," *International Journal of Computer Vision*, vol. 51, no. 2, pp. 91-109, 2003.

[3] R. Péteri and D. Chetverikov, "Dynamic Texture Recognition using Normal Flow and Texture Regularity," *Proc. Iberian Conference on Pattern Recognition and Image Analysis*, pp. 223-230, 2005.

[4] R. Polana and R. Nelson, "Temporal Texture and Activity Recognition," *Motion-based Recognition*, pp. 87-115, 1997.

[5] D. Chetverikov and R. Péteri, "A Brief Survey of Dynamic Texture Description and Recognition," *Proc. Int'l Conf. Computer Recognition Systems*, pp. 17-26, 2005.

[6] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution Gray Scale and Rotation Invariant Texture Analysis with Local Binary Patterns," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971-987, 2002.

[7] M. Pantic and L.L.M. Rothkrantz, "Automatic Analysis of Facial Expressions: The State of the Art," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no.12, pp. 1424-1455, 2000.

[8] B. Fasel and J. Luettin, "Automatic Facial Expression Analysis: A Survey," *Pattern Recognition*, vol. 36, pp. 259-275, 2003.

[9] T. Kanade, J.F. Cohn, and Y. Tian, "Comprehensive Database for Facial Expression Analysis," *Proc. Int'l Conf. Automatic Face and Gesture Recognition*, pp. 46-53, 2000.

[10] X. Feng, M. Pietikäinen, and A. Hadid, "Facial Expression Recognition with Local Binary Patterns and Linear Programming," *Pattern Recognition and Image Analysis*, vol. 15, no. 2, pp. 546-548, 2005.

[11] C. Shan, S. Gong and P.W. McOwan, "Robust Facial Expression Recognition Using Local Binary Patterns," *Proc. IEEE Int'l Conf. Image Procession*, pp. 370-373, 2005.

[12] M.S. Bartlett, G. Littlewort, I. Fasel, and R. Movellan, "Real Time Face Detection and Facial Expression Recognition: Development and

Application to Human Computer Interaction," *Proc. CVPR Workshop on Computer Vision and Pattern Recognition for Human-Computer Interaction,* 2003.
[13] G. Littlewort, M. Bartlett, I. Fasel, J. Susskind, and J. Movellan, "Dynamics of Facial Expression Extracted Automatically from Video," *Proc. IEEE Workshop Face Processing in Video,* 2004.
[14] M. Yeasin, B. Bullot, and R. Sharma, "From Facial Expression to Level of Interest: A Spatio-temporal Approach," *Proc. Conf. Computer Vision and Pattern Recognition,* pp. 922-927, 2004.
[15] S.P. Aleksic and K.A. Katsaggelos, "Automatic Facial Expression Recognition Using Facial Animation Parameters and Multi-stream HMMS," *IEEE Trans. Signal Processing, Supplement on Secure Media,* 2005.
[16] I. Cohen, N. Sebe, A. Garg, L.S. Chen, and T.S. Huang, "Facial Expression Recognition from Video Sequences: Temporal and Static Modeling," *Computer Vision and Image Understanding,* vol. 91, pp. 160-187, 2003.
[17] R.C. Nelson and R. Polana, "Qualitative Recognition of Motion Using Temporal Texture," *Computer Vision, Graphics, and Image Processing,* vol. 56, no. 1, pp. 78-99, 1992.
[18] P. Bouthemy and R. Fablet, "Motion Characterization from Temporal Co-occurrences of Local Motion-based Measures for Video Indexing," *Proc. Int'l Conf. Pattern Recognition,* vol.1, pp. 905-908, 1998.
[19] R. Fablet and P. Bouthemy, "Motion Recognition Using Spatio-temporal Random Walks in Sequence of 2D Motion-related Measurements," *IEEE Int'l Conf. Image Processing,* pp. 652-655, 2001.
[20] R. Fablet and P. Bouthemy, "Motion Recognition Using Nonparametric Image Motion Models Estimated from Temporal and Multiscale Co-occurrence Statistics," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, pp. 1619-1624, 2003.
[21] Z. Lu, W. Xie, J. Pei, and J. Huang, "Dynamic Texture Recognition by Spatiotemporal Multiresolution Histogram," *Proc. IEEE Workshop Motion and Video Computing,* vol. 2, pp. 241-246, 2005.
[22] C.H. Peh and L.-F. Cheong, "Exploring Video Content in Extended Spatiotemporal Textures," *Proc. 1st European Workshop Content-Based Multimedia Indexing,* pp. 147-153, 1999.
[23] C.H. Peh and L.-F. Cheong, "Synergizing Spatial and Temporal Texture," *IEEE Trans. Image Processing,* vol. 11, pp. 1179-1191, 2002.
[24] R. Péteri and D. Chetverikov, "Qualitative Characterization of Dynamic Textures for Video Retrieval," *Proc. Int'l Conf. Computer Vision and Graphics,* 2004.
[25] S. Fazekas and D. Chetverikov, "Normal Versus Complete Flow in Dynamic Texture Recognition: A Comparative Study," *4th Int'l Workshop Texture Analysis and Synthesis,* pp. 37-42, 2005.
[26] P. Saisan, G. Doretto, Y.N. Wu, and S. Soatto, "Dynamic Texture Recognition," *Proc. Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 58-63, 2001.
[27] K. Fujita and S.K. Nayar, "Recognition of Dynamic Textures Using Impulse Responses of State Variables," *Proc. Third Int'l Workshop Texture Analysis and Synthesis,* pp. 31-36, 2003.
[28] J.R. Smith, C.-Y. Lin, and M. Naphade, "Video Texture Indexing Using Spatiotemporal Wavelets," *Proc. IEEE Int'l Conf. Image Processing,* vol. 2, pp. 437-440, 2002.
[29] K. Otsuka, T. Horikoshi, S. Suzuki, and M. Fujii, "Feature Extraction of Temporal Texture Based on Spatiotemporal Motion Trajectory," *Proc. Int'l Conf. Pattern Recognition,* vol. 2, pp. 1047-1051, 1998.
[30] J. Zhong and S. Scarlaro, "Temporal Texture Recognition Model Using 3D Features," *Technical report,* MIT Media Lab Perceptual Computing, 2002.
[31] G. Aggarwal, A.R. Chowdhury, and R. Chellappa, "A System Identification Approach for Video-based Face Recognition," *Proc. 17th Int'l Conf. Pattern Recognition,* vol. 1, pp. 175-178, 2004.
[32] G. Zhao and M. Pietikäinen, "Dynamic Texture Recognition Using Volume Local Binary Patterns," *Proc. Workshop on Dynamical Vision WDV 2005/2006,* LNCS 4358, pp. 165-177, 2007.
[33] G. Zhao and M. Pietikäinen, "Local Binary Pattern Descriptors for Dynamic Texture Recognition," *Proc. Int'l Conf. Pattern Recognition,* vol. 2, pp. 211-214, 2006.
[34] T. Ahonen, A. Hadid, and M.Pietikäinen, "Face Recognition with Local Binary Patterns," *Proc. Eighth European Conf. Computer Vision,* pp. 469-481, 2004.
[35] T. Ahonen, A. Hadid, and M.Pietikäinen, "Face Description with Local Binary Patterns: Application to Face Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 12, pp. 2037-2041, 2006.
[36] J. Bassili, "Emotion Recognition: The Role of Facial Movement and the Relative Importance of Upper and Lower Areas of the Face," *Journal of Personality and Social Psychology,* vol. 37, pp. 2049-2059, 1979.
[37] P. Ekman and W.V. Friesen, "The Facial Action Coding System: A Technique for the Measurement of Facial Movement," *Consulting Psychologists Press, Inc.,* San Francisco, CA, 1978.
[38] Y. Tian, "Evaluation of Face Resolution for Expression Analysis," *Proc. IEEE Workshop on Face Processing in Video,* 2004.

Guoying Zhao received the MS in computer science from North China University of Technology, in 2002, and the PhD degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China in 2005. Since July 2005, she has been a postdoctoral researcher in Machine Vision Group at the University of Oulu. Her research interests include dynamic texture recognition, facial expression recognition, human motion analysis and person identification. She has authored over 30 papers in journals and conferences.

Matti Pietikäinen received his Doctor of Science in Technology degree from the University of Oulu, Finland, in 1982. In 1981, he established the Machine Vision Group at the University of Oulu. This group has achieved a highly respected position in its field and its research results have been widely exploited in industry. Currently, he is Professor of Information Engineering, Scientific Director of Infotech Oulu Research Center, and Leader of the Machine Vision Group at the University of Oulu. From 1980 to 1981 and from 1984 to 1985, he visited the Computer Vision Laboratory at the University of Maryland, USA. His research interests are in texture-based computer vision, face analysis, and their applications in human-computer interaction, person identification and visual surveillance. He has authored about 195 papers in international journals, books, and conference proceedings, and about 100 other publications or reports. He has been Associate Editor of IEEE Transactions on Pattern Analysis and Machine Intelligence and Pattern Recognition journals. He was Guest Editor (with L.F. Pau) of a two-part special issue on "Machine Vision for Advanced Production" for the International Journal of Pattern Recognition and Artificial Intelligence (also reprinted as a book by World Scientific in 1996). He was also Editor of the book Texture Analysis in Machine Vision (World Scientific, 2000) and has served as a reviewer for numerous journals and conferences. He was President of the Pattern Recognition Society of Finland from 1989 to 1992. Since 1989, he has served as Member of the Governing Board of the International Association for Pattern Recognition (IAPR) and became one of the founding fellows of the IAPR in 1994. He has also served on committees of several international conferences. He has been Area Chair of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007) and is Co-chair of Workshops of International Conference on Pattern Recognition (ICPR 2008). He is Senior Member of the IEEE and was Vice-Chair of IEEE Finland Section.