Enhanced Local Binary Covariance Matrices (ELBCM) for texture analysis and object tracking

Andrés Romero
Laboratoire de Recherche en Informatique
Bât 650, Université Paris Sud
91405 Orsay Cedex France
andres.romero@u-psud.fr

Michèle Gouiffès
Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur
Bat 508, Université Paris Sud
91405 Orsay Cedex France
michele.guiffes@u-psud.fr

Lionel Lacassagne
Laboratoire de Recherche en Informatique
Bât 650, Université Paris Sud
91405 Orsay Cedex France
lionel.lacassagne@u-psud.fr

ABSTRACT

This paper presents a novel way to embed local binary texture information in the form of local binary patterns (LBP) into the covariance descriptor. Contrary to previous publications, our method is not based on the LBP decimal values where arithmetic operations have no texture meaning. Our method uses the angles described by the uniform LBP patterns and includes them into the set of features used to build the covariance descriptor. Our representation is not only more compact but more robust because it is less affected by noise and small neighborhood rotations. Experimental evaluations corroborate the performance of our descriptor for texture analysis and tracking applications. Our descriptor rivals with state-of-the-art methods and beats other covariance-based descriptors.

Categories and Subject Descriptors

H.4 [**Object, body, and face tracking in image sequences**]: Covariance descriptors, local binary patterns, texture analysis

Keywords

Covariance descriptors, LBP, Local log-Euclidean, Riemannian manifolds

1. INTRODUCTION

Texture is a phenomenon which depends on the scale at which different objects or artifacts appear in an image. In computer vision textures provide strong cues useful for object and material recognition even some of their properties can be identified by texture analysis (e.g., hardness, smoothness, opaqueness). Texture algorithms deal with several difficulties such as the intra-class variation, clutter, occlusions and changes in illumination and pose. Discriminant representations are required to notice when two textures are closely similar or significantly different. Textures can be modeled as the combination of primitive elements (*textones*) such as spots, bars and edges considering their scale and orientation and how these element repeat to form different types of patterns. Some special type of filters (e.g., Gabor filters [12]) are specifically defined to deal with textones. Other representations use local descriptors in regions of interest (ROI) given by popular state of the art key-point detectors (e.g., Harris [5], SIFT [11], SURF [2]) or over dense grid of points evenly distributed over the image's region of interest [10] and [3].

Covariance matrix representations [18, 16] and [9] offer a very practical alternative that allows us to build models based on local image correlations between different image cues (e.g., intensity, directional gradients, spatial coordinates, texture representations, optical flow, etc). Up to some limits, these representations are insensitive to scale and rotations changes and to illumination variations. Covariance matrices are symmetric and positive-definite (SPD), each matrix in this set represents a point in a smooth Riemannian manifold (i.e. Euclidean geometry is not applicable) and matrix distances are measured using geodesics defined in the Riemannian manifold. To improve texture discriminant capabilities Pang et al. proposed GRCM which results from the embedding of the Gabor filter bank responses into the set of features used to build covariance matrix descriptor [15] and [17]. Due to the large number of features used (each one for each filter) Gabor-based covariances are expensive to compute and match. Some publications replace Gabor filters and include local binary patterns (LBP) [14] into the covariance matrix (e.g., local binary covariance matrix (LBCM)[4] and Gabor-LBP based region covariance descriptor (GLRCD) [20]). But the right way to embed LBP's into the covariance matrix descriptor is not obvious. GLRCD[20] directly uses LBP's decimal value which is not only very unstable for some very common phenomenons such as local neighborhood rotations but for which typical arithmetic operations have no meaning (i.e. adding or averaging two or more LBP decimal values does not mean anything in terms of texture). LBCM uses bit-strings (i.e. each bit acts as an independent feature) which are more stable and for which arithmetic operations are correctly defined, the problem is that the number of features to embed into the covariance matrix depends on the number of bits forming the LBP pat-

tern and this can severely reduce execution speed.

In this article we propose an enhanced LBP-based descriptor (ELBCM) which is based on the angles defined by the uniform LBP patterns. ELBCM's descriptor size is independent of the number of neighbors used in the LBP pattern. ELBCM has been evaluated experimentally for different texture analysis applications such as classification, facial expression recognition and tracking and its performance surpasses previous covariance-based configurations and rivals with some state-of-the-art methods.

The remainder of this paper is organized as follows. Section 2 explains basics aspects about the different LBP configurations. Section 3 presents the underlying theory of covariance matrix descriptors. ELBCM's details are presented in section 4. The description of our experiments and their respective results are given in section 5. Finally our conclusion is given in section 6.

2. LOCAL BINARY PATTERNS (LBP)

Local binary patterns (LBP) are very efficient texture operators which label the neighborhood of each pixel as a binary number according to a predefined set of comparisons. One important advantage of LBP operators is that they are invariant against monotonic gray level changes such as the ones caused by illumination variations. LBP operators have a considerable success in the computer vision community because they are computationally simple and very discriminant. LBP's are proving that filter banks with large support areas are not always necessary to achieve high performance in texture classification. The original LBP pattern operator was introduced in [13] where the authors highlighted two important aspects of a texture: a pattern and its strength. The original and most basic operator consists of a 3×3 square neighborhood where its center is used as a threshold, the resulting thresholded values are accumulated considering a weight associated to their position, these weights grow as powers of two: $\{2^0, 2^1, \cdots, 2^{P-1}\}$. The first row in Figure 1 illustrates the operator concept, the idea is to build a binary code of length P (i.e., for a 8-pixel neighborhood there are $2^8 = 256$ possible labels) which depends on the relative gray values of the center and the pixels in the neighborhood. Several years after the original publication, this basic operator was presented in a more generic form in [14]. In its generic form, the operator has no limitations on the neighborhood's size or on the number of sampling points (P).

The local texture of the monochrome image $I(x, y)$ is characterized by the joint distribution of $P + 1$ gray values

$$T = t(g_c, g_0, g_1, \cdots, g_{P-1}) \tag{1}$$

where g_c is the gray value of an arbitrary pixel (x, y) (i.e., $g_c = I(x, y)$), and $g_p = I(x_p, y_p)$ is the gray value of a sampling point $p \in \{0, \cdots, P-1\}$ located at

$$\begin{aligned} x_p &= x + Rcos(2\pi p/P) \\ y_p &= y - Rsin(2\pi p/P) \end{aligned} \tag{2}$$

where P is the total number of sampling pixels around (x_c, y_c), these samples are taken sampling the circle of radius R at evenly spaced degrees.

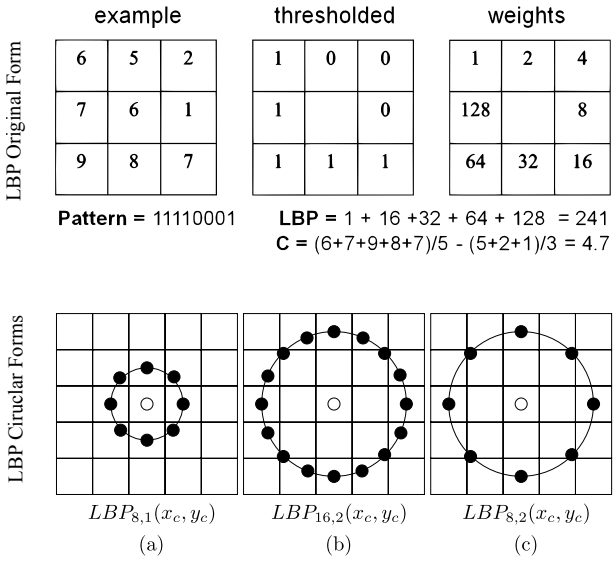

Figure 1: An example of the LBP original forms appears in the first row: all pixels in the neighborhood are thresholded by the pixel at the center, then a binary code is formed by the dot product of the thresholded values and their weights. Different sized neighborhoods of the form $LBP_{P,R}(x_c, y_c)$ appear on the second row. Here P is the number of pixels considered by LBP pattern and R the radius distance from the neighborhood center.

The T distribution is approximated by the local neighborhood differences

$$T \approx t(g_0 - g_c, g_1 - g_c, \cdots, g_{P-1} - g_c), \tag{3}$$

this makes the operator invariant to changes of the mean gray value. But these values are not invariant to other changes in gray levels, to alleviate this limitation only signs of the differences are considered

$$t(s(g_0 - g_c), s(g_1 - g_c), \cdots, s(g_{P-1} - g_c)) \tag{4}$$

where $s(z)$ is the step function

$$s(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{if } z < 0 \end{cases}. \tag{5}$$

Accumulating the thresholded differences weighted by powers of two, the operator is finally defined as

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c)2^p. \tag{6}$$

Equation (6) interprets the sign of the differences in a neighborhood as a P-bit binary number, which results in 2^P different labels. The local gray-scale texture is thus, approximately described with the 2^P-bin distribution of LBP labels

$$T \approx t(LBP_{P,R}(x_c, y_c)). \tag{7}$$

Pixels located at the diagonals are interpolated. While this

may seem costly, this generic form is advantageous as it is adaptable for different scales sizes, and rotation invariant descriptors are easily derived from this circular neighborhood indexing.

Uniform patterns(ULBP)[14] are the result of another modifications to the original definition of the LBP operator. For this, a uniformity measure that counts the number of bitwise transitions from 0 to 1 or vice versa is employed. A pattern is considered uniform if its uniformity measure is ≤ 2. For example, 00000000 and 01110000 are uniform patterns but 11001001 and 10010011 are not. It has been proven experimentally that ULBP patterns account for the majority of texture patterns in natural images, and they are indications that uniform patterns are more stable or less prone to noise. ULBP patterns are also convenient because they reduce the possible number of labels conducting to more concentrated (less sparse) histogram distributions. Our proposed descriptor (defined in section 4.2) is based on the set of uniform LBP patterns $ULBP_{8,R}$ and the covariance matrix descriptor presented next.

3. COVARIANCE MATRIX DESCRIPTOR

Covariance matrices and symmetric positive-definite matrices (SPD) in general are gaining importance in a variety of applications such as Diffusion tensor imaging (DTI) [19], brain-computer interfaces (BCI) [1] and conventional imaging. All of them operating with very heterogeneous signal sources. In computer vision, Tuzel et al. [18] proposed to use the covariance matrices obtained from several image statistics computed inside a region of interest as the region descriptor (known in the literature as covariance descriptor). Covariance matrix descriptors dimensionality is much smaller than other representations based on histograms. And even better, they can be calculated super fast using integral images and the cost is independent from region's size. Unfortunately, covariance matrices do not lie on the Euclidean space, they are points in a Riemannian manifold and as such classical geometry and machine learning algorithms need to be adapted to work in this type of geometry.

Let I represent a luminance (grayscale) or a three dimensional color image (other types of images can also be considered: infrared, depth images, etc.), and let F be the $W \times H \times d$ dimensional feature image extracted from I

$$F(x, y) = \phi(I, x, y) \tag{8}$$

where ϕ is any d-dimensional mapping forming a feature combination for each pixel including features such as intensity, color (in any color space), gradients, filter responses, or any possible set of images obtained from I. Let now $\{\mathbf{z}_k\}_{k=1\cdots n}$ be a set of d-dimensional points inside the rectangular region $R \subset F$ (region size is n). The region R is represented with the $d \times d$ coavariance matrix

$$\mathbf{C}_R = \frac{1}{n-1} \sum_{k=1}^{n} (\mathbf{z}_k - \boldsymbol{\mu})(\mathbf{z}_k - \boldsymbol{\mu})^T \tag{9}$$

where $\boldsymbol{\mu}$ is the mean vector of the points.

The covariance matrix is a $d \times d$ square matrix which fuses multiple features naturally measuring their correlations. Elements in the main diagonal represent the variance of each feature, while elements outside this diagonal represent the correlations. Thanks to the averaging in the covariance computation, noisy pixels are largely filtered out which contrasts with raw-pixel methods. Covariance matrices are low-dimensional compared to other descriptors, and due to symmetry $\mathbf{C}_R$ has only $(d^2 + d)/2$ different values. The covariance descriptor $\mathbf{C}_R$ does not use any information regarding the ordering or the number of the points implying a certain scale and rotation invariance. Nevertheless the covariance descriptor ceases to be rotationally invariant when orientation information is introduced in the feature vector such as the norm of gradients with respect to x and y directions.

4. PROPOSED DESCRIPTOR

Covariance descriptors are very convenient and effective for object detection, human tracking and texture classification. The previously mentioned Gabor filter-banks approach [15] has two inconveniences: the convolution of texture images with multiple Gabor filters is expensive to compute and secondly the number of features inside the covariance matrix grows up to $d = 42$ which increases matching time. LBP-based covariance descriptors have been proposed to deal with this situation [4, 20]. The key advantages of these descriptors are that LBPs are invariant to illumination changes and computationally cheaper to compute than Gabor features. LBPs are traditionally used to describe texture micro patterns such as edges, spots and flat areas by measuring their statistical distribution in the form of histograms calculated inside a defined region. The problem with LBP's is that each LBP decimal label corresponds to its own bin in a histogram and traditional arithmetic operations between two or more of these labels are not well defined (e.g., the mean of two ore more LBP decimal values not necessarily renders a meaningful texture result). Disregarding this fact, some methods [20] directly introduce these LBP decimal values to the set the features that form the covariance matrix. While this description may work in some particular cases, in our view it is ill defined and in the general case this configuration can significantly reduce the final performance. Observing this situation, [4] proposed to use the string of local binary values obtained by comparing the pixels in the neighborhood that surrounds the central pixel in the place of the LBP decimal value so that each one of the considered neighbors represents a feature component, we believe this approach is correct but in our experiments we have witnessed an increased presence of non-positive-definite matrices. We believe this is due to the sparsity of the feature vector in smooth regions. Observing this problem we have decided to avoid these representations and to propose a new one based on the angles defined by the LBP regular patterns. This section continues with an exploration of the existent LBP-based covariance descriptor (LBCM), and then we explain our own descriptor.

4.1 LBP-based covariance matrices (LBCM)

LBP-based covariance matrices provide an complementary representation to the traditional LBP histogram analysis. While LBP histograms estimate the statistical distribution of local micro-patterns such as edges, corners and flat areas inside a texture, covariance matrices can go further and integrate other types of information such as the spatial, luminance and color correlations between neighbors. To utilize

this information the local binary feature

$$\mathbf{LBP}(x,y) = \{LBP_i^{(x,y)} | i = 1, 2, \cdots, P\} \quad (10)$$

is proposed. Here, $LBP_i^{(x,y)}$ is the label of the i-th neighbor of the central pixel located at (x,y) and P is the total number of neighbors of (x,y). The value of $LBP_i^{(x,y)}$ is calculated as

$$LBP_i^{(x,y)} = \begin{cases} 1 & \text{if } I(LN_i^{(x,y)}) \geq I(x,y) \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

where $LN_i^{(x,y)}$ gives the coordinates of the i-th (x,y) LBP neighbor.

The final (LBCM) mapping function is

$$\phi(I,x,y) = \left[\begin{array}{cccc} x & y & I(x,y) & \mathbf{LBP}(x,y) \end{array} \right]. \quad (12)$$

Although interesting, the approach has two main drawbacks. First, the dimension of this feature vector is at least 11 and depends on the neighborhood size P of the selected $LBP_{P,R}$ operator. Second, it frequently produces non-positive-definite matrices, because of the sparsity of the feature vector. To alleviate these problem, the current paper proposes to use the angles defined by the LBP regular patterns.

4.2 Enhanced LBP-based covariance matrices (ELBCM)

In the case of the $LBP_{8,R}$ operator there are 58 possible uniform patterns, but the strings formed by all ones and all zeros are ignored because their angle representations are the same, which can can be misleading. Figure 2 shows the remaining 56 different patterns used to construct our ELBCM covariance descriptor when applying an $LBP_{8,R}$ operator. The starting and ending angles θ_0 and θ_1 are marked by the curved arrows inside each pattern. Since computations have to be made on circular quantities, the angles are converted to points on the unit circle, e.g., θ_0 is converted to $(cos\theta_0, sin\theta_0)$, which is equivalent to convert from polar coordinates to Cartesian coordinates.

Finally, vector

$$\mathbf{v}(\theta_0,\theta_1) = [cos(\theta_0) \;\; sin(\theta_0) \;\; cos(\theta_1) \;\; sin(\theta_1)] \quad (13)$$

uniquely describes each local LBP pattern as a function of (θ_0,θ_1). An example for calculating vector $\mathbf{v}(\theta_0,\theta_1)$ is provided in Figure 2-(b) using decimal value 63 (bit string 00111111). Our enhanced LBP-based covariance matrix (ELBCM) texture descriptor is built using the mapping function

$$\phi(I,x,y) = \left[\begin{array}{cccc} x & y & I(x,y) & \mathbf{v}(LBP(x,y)) \end{array} \right]. \quad (14)$$

where $LBP(x,y)$ provides the pair of angles (θ_0,θ_1) at (x,y) to $\mathbf{v}$. Local neighborhoods with non-uniform LBP patterns are ignored and their feature do not contribute to the covariance matrix computation. Because $\mathbf{v}$ is a four dimensional vector, the total dimension of $\phi(I,x,y)$ is $d = 7$ and the resulting covariance matrices are of size 7×7.

The use of trigonometric formulas in (14) should make ELBCM computation more time consuming, but the problem is fully solved by using a look-up table that maps directly the 56 LBP decimal numbers to the sine and cosine values of θ_0 and θ_1.

ELBCM has multiple advantages in comparison to previous covariance-based texture descriptors (e.g., Tuzel[18] and LBCM [4]): it is more compact than [4] (7 components vs. 11 for LBCM), more stable *i.e.*, small rotations of the LBP pattern slightly affect $\mathbf{v}(\theta_0,\theta_1)$ components, whereas in LBCM the same rotations irregularly affect different bits in $\mathbf{LBP}(x,y)$ depending on their local position. In practice, special care must be taken because some image regions may have non-positive-definite covariance matrices this problem was more frequently observed for the LBCM matrices than with our ELBCM approach because bit-strings $\mathbf{LBP}(x,y)$ can be very sparse in smooth regions.

5. EXPERIMENTS

In this section, we exhibit some possible applications of our ELBCM descriptor such as texture and facial expression classification and object tracking.

5.1 Texture classification

For performance evaluation we used two widely used datasets in texture classification. The Brodatz dataset and KTH-TIPS dataset [6].

5.1.1 Brodatz Dataset

The Brodatz dataset contains 111 different textures each one represented by a single 640×640 image. Scale, viewpoint or illumination changes are not a problem in this dataset, the main difficulty of this dataset in terms of classification is that it includes non-homogeneous textures. To have comparable results, we have followed the same methodology of [9] where each image is subdivided into 49 overlapping blocks of size 160×160 using a pixel block stride of 80. Each class is trained using twenty four randomly selected images from their respective 49 images set (the remaining ones are used for testing). During training, each image is now divided into four 80×80 patches and their ELBCM covariances are then computed. During testing, each covariance matrix of the testing image is compared to all the covariance matrices inside the training set using the Riemannian metric proposed in [18]. A label is assigned to each test image using a KNN algorithm ($k = 5$) which counts the number of votes gathered by the four test image covariance matrices for each class. In Figure 3-(a) the classification accuracy results for the Brodatz dataset are reported. Different feature map configurations in the covariance matrices were tested: Tuzel[18], LBCM[4] and ELBCM, other state-of-the-art results are included too (taken from [9]). The reported results are the average of twenty different experiment runs.

5.1.2 KTH-TIPS Dataset

This dataset [6] is composed by ten different texture classes represented by 81 different samples. This dataset is more challenging than Brodatz because here each texture class is represented by images taken at different scales, illumination and pose. The size of the samples is 200×200 pixels. Similar to the previous experiment each image is uniformly subdivided into four blocks and a covariance matrix is computed on each one. Figure 3-(b) shows the classification accuracy

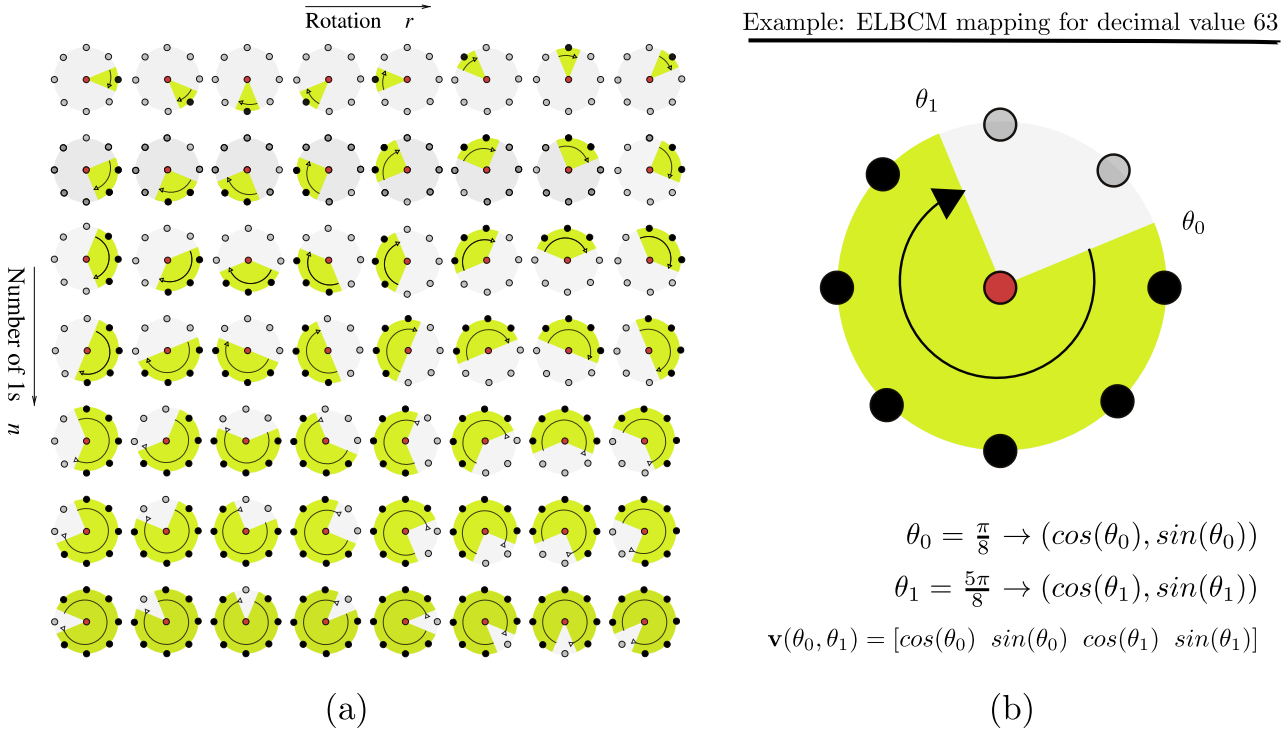

Figure 2: (a)The set of 56 different uniform LBP patterns in a $(8, R)$ neighborhood. For each pattern a curved arrow indicates the start and end angles (θ_0 and θ_1). (b)In ELBCM, the uniform LBP pattern defined by two angles θ_0 and θ_1 is described by $\mathbf{v}(\theta_0, \theta_1) = [cos(\theta_0) \;\; sin(\theta_0) \;\; cos(\theta_1) \;\; sin(\theta_1)]$. The LBP pattern represented by decimal value 63 maps to $\mathbf{v}(\pi/8, 5\pi/8) = [cos(\pi/8) \;\; sin(\pi/8) \;\; cos(5\pi/8) \;\; sin(5\pi/8)]$.

depending of each feature combination or method depending on the number of training images.

5.2 Facial expression classification

To evaluate the performance of our covariance descriptor in the facial expression recognition problem we used the JAFFE dataset[1] which consists of 213 images from Japanese female individuals, covering seven expression categories (neutral, anger, disgust, fear, happiness, sadness ans surprise). To compare with [4] we followed their same methodology: images were automatically cropped to align the eyes, then they were resized to 100×100 pixels and 21 samples of each expression were used for training and only one training sample for testing. The experiments were repeated several times and their averages are displayed in Figure 4.

5.3 Object tracking

Using the covariance matrix as a global region descriptor, we compare ELBCM descriptors with other configurations: Tuzel's [18], LBCM [17] and LRCD [20] (LBP decimal value without Gabor filters configuration). The initial trackers position is marked in the first frame by hand. We implemented a very simple tracker that is limited to fixed-scale targets and which follows targets locally (i.e. if targets are located far from the trackers positions they are not found). Three different sequences are discussed here and what interest us is the precision at which the different feature combinations follow the targets and their tracking times. When ground-truth information is available precision is measured using following weighted error

$$w_{error} = \frac{\|C_{gt} - C_{tracked}\|}{R} \tag{15}$$

where C_{gt} and $C_{tracked}$ are the ground truth and tracked centers respectively, and R is the target size, calculated as the average of the vertical and horizontal bounding box lengths. Unfortunately, ground-truth is only available for the first two sequences. Some samples from the tracking results are shown in Figures 5, 6 and 7.

5.3.1 Jumping sequence

This sequence (313 frames, size: 352×288) appears in gray-scale, it shows a guy jumping rope in a garden, a building and some trees are behind. The object of interest is the guys face. Due to the jumping speed and random camera movements images appear extremely blurred. After frame $t = 41$ Tuzel and LRCD confuse the guys face with the background losing the target and never being able to re-catch it again. In contrast, LBCM and ELBCM follow the target until the end. In some frames and due to the shaking of the camera, the guys face appears extremely blurred and trackers drift considerably from the real location, but LBCM and ELBCM are able to catch it again as soon as the target appears reasonably clear again. This sequence comes from the TLD dataset [7]. Results for this sequence are given in **Table 1**, because LRCD matrices are the smallest (4×4)

[1] http://www.kasrl.org/jaffe.html

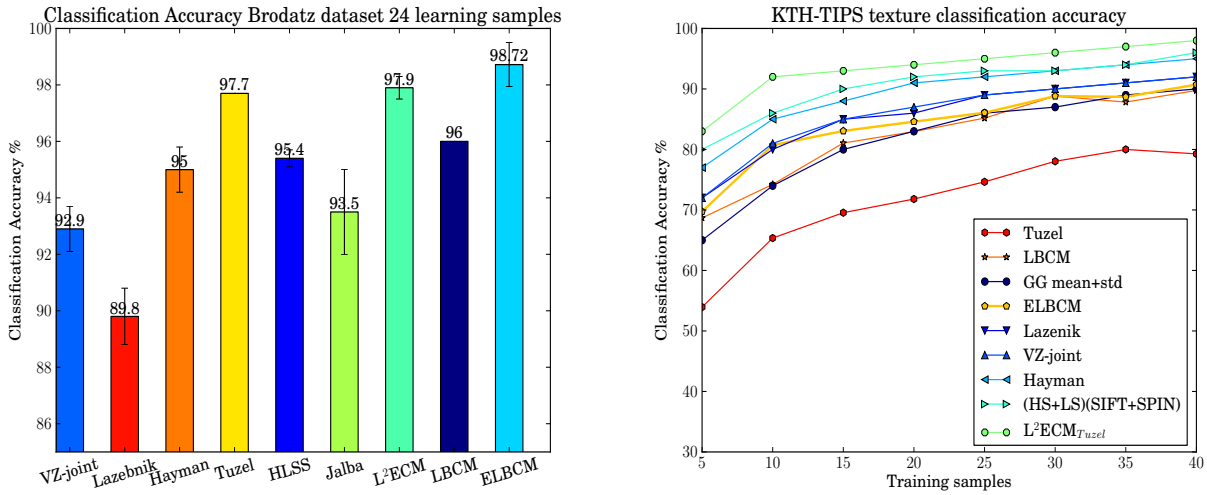

Figure 3: Texture classification accuracy for the (a)Brodatz and (b)KTH-TIPS dataset.

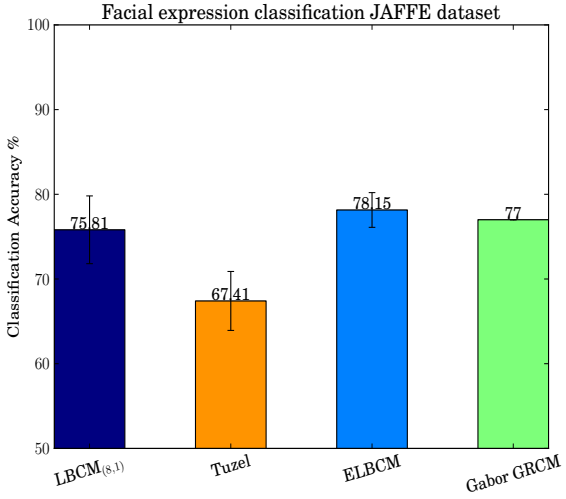

Figure 4: Facial expression classification results for the JAFFE database using 21 learning samples.

they are the fastest ones but less precise. The best descriptor in terms of performance is ELBCM which is two times faster than LBCM and more precise in terms of the w_{error}.

5.3.2 *Panda sequence*

This sequence (3000 frames, size: 312×233) was filmed in color. It shows a panda enclosed in a park. During the first frames of the sequence, camera zooms in-out while the panda is walking. The panda is a non-rigid target, its appearance changes a lot along the sequence: it turns on itself and its shape and color distribution drastically change. In addition, the panda passes behind a tree, and the background changes. All methods follow the target correctly until frame $t = 1001$ when it is no longer visible. After this, it re-enters the scene at $t = 1182$ LRCD, LBCM and ELBCM are able to re-identify it and track it until frame $t = 2656$ when the camera zooms in and the target size changes significantly affecting the appearance. Execution time and w_{error} results are shown in **Table 2** and as for the previous sequence, ELBCM provides the fastest and more precise results. This sequence comes from the TLD dataset [7] too.

5.3.3 *Pedxing 3 sequence*

Used in [8], this sequence (188 frames long from $t = 7484$ to $t = 7672$, size: 640×480) was registered in color. The target is a pedestrian who is crossing the street, and passes behind a panel. Due to compression noise, the quality of the image is very poor. All methods but ELBCM fail to catch the pedestrian after he passes behind the panel. Tracking results are given in **Table 3**, w_{error} information is not provided here because there is no ground-truth for this sequence what is interesting here is to observe how tracking time grows depending on the target's size.

6. CONCLUSIONS

In this article we have presented a feature vector which embeds local binary texture information based on the uniform LBP patterns into the covariance descriptor. Our experiments show the advantages of our descriptor in texture analysis and tracking applications. Compared to similar approaches our descriptor is more robust and due to its compactness it is faster (to compute and to match), and as our results show it rivals in performance with some of the state-of-the-art methods.

7. FUTURE WORK

In future work we are interested in evaluating ELBCM in the form of a local log-Euclidean covariance matrix L^2ECM[9] for face recognition and multiple target tracking applications.

Tracking results Jumping sequence		
Features	Tracking time/frame [ms]	w_{error}
Tuzel	4.83	1.83294
LBCM	10.3	0.362136
ELBCM	4.82	0.268516
LRCD	2.25	2.35453

Table 1: Tracking results Jumping sequence

Tracking results Panda sequence		
Features	Tracking time/frame [ms]	w_{error}
Tuzel	3.65	1.9342
LBCM	7.57	0.282194
ELBCM	3.65	0.279785
LRCD	1.71	0.388786

Table 2: Tracking results Panda sequence

Tracking results Pedxing3 sequence		
Features	Tracking time/frame [ms]	w_{error}
Tuzel	20.5	-
LBCM	40.6	-
ELBCM	20.5	-
LRCD	6.69	-

Table 3: Tracking results Pedxing3 sequence

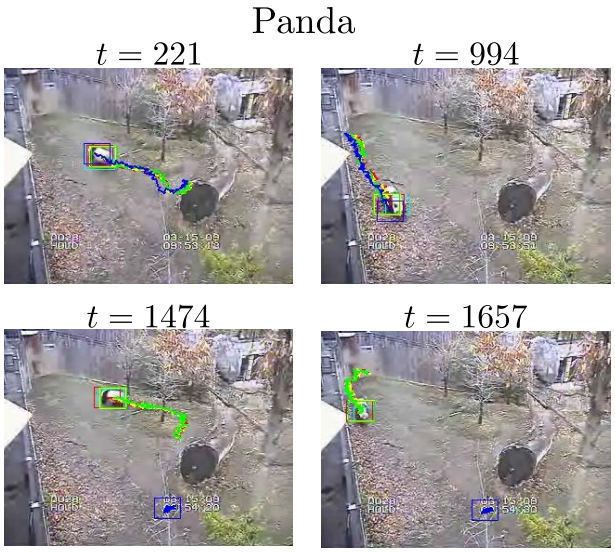

Figure 6: Tracking results in the Panda sequence. During the first 1000 frames of the sequence all targets follow correctly the target. Tuzel's tracker (blue) loses the target when it gets out of the image. When it re-enters, LRCD(green), LBCM(red) and ELBCM(yellow) follow the target correctly until $t = 2656$ when the camera zooms in and the object appearance changes significantly. Ground-truth information is colored in cyan.

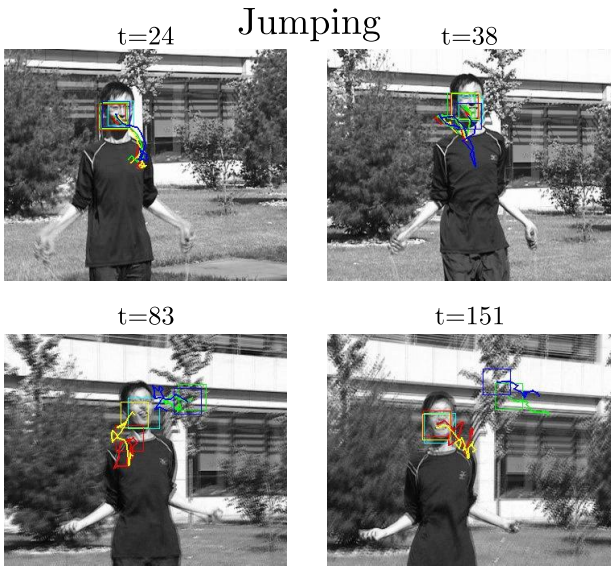

Figure 5: Tracking results in the Jumping sequence. Tuzel's(blue) tracker and LRCD(green) get stuck in the background after the first frames. LBCM(red) is less precise than ELBCM(yellow) but both follow the target until the end. Ground-truth information is colored in cyan.

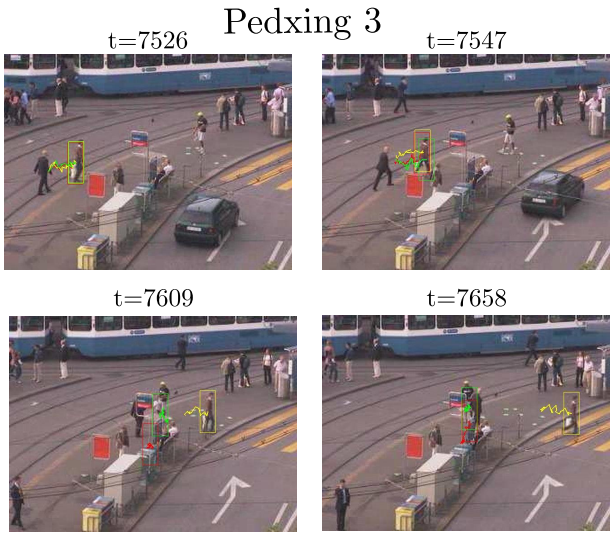

Figure 7: Tracking results for Pedxing3 sequence. Tuzel's tracker(blue), LRCD(green) and LBCM(red) lose the target when it passes behind the panel at the center of the image. Only ELBCM (yellow) succeeds to follow the target until the end.

8. REFERENCES

[1] A. Barachant, S. Bonnet, M. Congedo, C. Jutten, et al. Bci signal classification using a riemannian-based kernel. In *Proceeding of the 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 97–102, 2012.

[2] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *Computer Vision–ECCV 2006*, page 404–417, 2006.

[3] A. Bosch, A. Zisserman, and X. Munoz. Scene classification via pLSA. *Computer Vision–ECCV 2006*, page 517–530, 2006.

[4] S. Guo and Q. Ruan. Facial expression recognition using local binary covariance matrices. In *Wireless, Mobile & Multimedia Networks (ICWMMN 2011), 4th IET International Conference on*, page 237–242, 2011.

[5] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50, 1988.

[6] E. Hayman, B. Caputo, M. Fritz, and J.-O. Eklundh. On the significance of real-world conditions for material classification. *Computer Vision-ECCV 2004*, pages 253–266, 2004.

[7] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7):1409–1422, 2012.

[8] B. Leibe, K. Schindler, and L. Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[9] P. Li and Q. Wang. Local log-euclidean covariance matrix (L2ECM) for image representation and its applications. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision – ECCV 2012*, volume 7574 of *Lecture Notes in Computer Science*, pages 469–482. Springer Berlin Heidelberg, 2012.

[10] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):978–994, 2011.

[11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[12] J. R. Movellan. Tutorial on gabor filters. *Open Source Document*, 2002.

[13] T. Ojala, M. Pietikainen, and D. Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision amp; Image Processing., Proceedings of the 12th IAPR International Conference on*, volume 1, pages 582 –585 vol.1, oct 1994.

[14] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002.

[15] Y. Pang, Y. Yuan, and X. Li. Gabor-based region covariance matrices for face recognition. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(7):989 –993, july 2008.

[16] F. Porikli, O. Tuzel, and P. Meer. Covariance tracking using model update based on lie algebra. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 728–735. IEEE, 2006.

[17] J. Tou, Y. Tay, and P. Lau. Gabor filters as feature images for covariance matrix on texture classification problem. *Advances in Neuro-Information Processing*, page 745–751, 2009.

[18] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. *Computer Vision–ECCV 2006*, pages 589–600, 2006.

[19] B. Vemuri, M. Liu, S.-I. Amari, and F. Nielsen. Total bregman divergence and its applications to DTI analysis. *Medical Imaging, IEEE Transactions on*, 30(2):475 –483, Feb. 2011.

[20] Y. Zhang and S. Li. Gabor-LBP based region covariance descriptor for person re-identification. In *Image and Graphics (ICIG), 2011 Sixth International Conference on*, page 368–371, 2011.