SMOOTH-INVARIANT GAUSSIAN FEATURES FOR DYNAMIC TEXTURE RECOGNITION

Thanh Tuan Nguyen[1,2,3], Thanh Phuong Nguyen[1,2], Frédéric Bouchara[1,2]

[1] Université de Toulon, CNRS, LIS, UMR 7020, 83957 La Garde, France
[2] Aix-Marseille Université, CNRS, ENSAM, LIS, UMR 7020, 13397 Marseille, France
[3] HCMC University of Technology and Education, Faculty of IT, HCM City, Vietnam

ABSTRACT

An efficient framework for dynamic texture (DT) representation is proposed by exploiting local features based on Local Binary Patterns (LBP) from filtered images. First, Gaussian smoothing filter is used to deal with near uniform regions and noise which are typical restrictions of LBP operator. Second, the receptive field of Difference of Gaussians (DoG), which is exploited in DT description for the first time, allows to make the descriptor more robust against the changes of environment, illumination, and scale which are main challenges in DT representation. Experimental results of DT recognition on different benchmark datasets (i.e., UCLA, DynTex, and DynTex++), which give outstanding performance compared to the state of the art, verify the interest of our proposal.

Index Terms— Dynamic Texture, Dynamic Texture Recognition, DoG, Gaussian Filter, LBP, CLBP.

1. INTRODUCTION

A dynamic texture is a repetition of its features in a spatio-temporal domain, such as fountain, fire, clouds, blowing flag, trees, waves, etc. Efficiently describing DTs is one of the crucial factors in computer vision applications. Many efforts, which have addressed different techniques for DT representation, can be mainly grouped into six categories as follows. Firstly, *optical-flow-based methods* [16, 14] effectively capture the characteristics of chaotic motions in natural ways for DT recognition. Secondly, *model-based methods* [25, 30] are mostly based on Linear Dynamical System [25] to model spatial-temporal features. Thirdly, *filter-based methods* [3, 24] have taken various filters into account dealing with the problems of noise and illumination in encoding videos. Fourthly, *geometry-based methods* [31, 32, 21] are focused on fractal analysis of dynamic properties for DT description. Fifthly, *learning-based methods* are in two main trends: deep learning techniques (e. g., Convolutional Neural Network (CNN)) are primarily utilized for learning features [18, 1, 2] while the kernel sparse coding is taken into account in dictionary-learning-based approaches [19, 20]. Lastly, *local-feature-based methods* [12, 26, 27, 29, 28, 13] take advantages of LBP operator in simple and efficient computation for DT representation. Most of them are based on two following variants: Volume LBP (VLBP) [33] and LBP on three orthogonal planes (LBP-TOP) [33].

In spite of obtaining the promising rates on DT recognition tasks, several restrictions in local-feature-based methods have been addressed in following works: illumination and noise problems [3, 11] in filter-based methods; near uniform regions, sensitivity to noise [27, 12]; and large dimensional puzzles [33, 23, 26]. To mitigate these shortcomings, we propose in this paper an efficient framework to encode a DT video in two main stages. First, Gaussian-based kernels (Gaussian and DoG) are applied on each orthogonal plane of the video sequence to capture their smooth and invariant features against illumination and noise problems. Second, a local feature extractor (e.g., CLBP [7]) is exploited to encode the relationships of local patterns in these filtered images. Finally, a prominent descriptor is formed by concatenating and normalizing the obtained probability distributions. Evaluations of the proposed framework on different benchmark datasets in DT classification task validate that our method significantly outperforms in comparison with the state-of-the-art results.

2. PROPOSED METHOD

The typical local-feature-based methods are often limited by the problems of sensitivity to illumination and noise. To deal with those, hereafter, we firstly take a look of LBP and its variants as well as Gaussian-based filtering kernels. We then propose a robust descriptor based on smooth-invariant Gaussian features to enhance discrimination power. It should be noted that DoG is involved in DT encoding for the first time.

2.1. A brief review of LBP and CLBP

Ojala et al. [15] introduced a LBP as a string of binary values to depict the local relationships of textural features in an image. Let $\mathcal{I}$ denote a 2D gray-scale image. A LBP code of a center pixel $\mathbf{q}_c$ is formed by considering the difference of $\mathbf{q}_c$'s gray-level and its local neighbors' as follows.

$$\mathrm{LBP}_{P,R}(\mathbf{q}_c) = \sum_{i=0}^{P-1} f\big(\mathcal{I}(\mathbf{p}_i) - \mathcal{I}(\mathbf{q}_c)\big)2^i \qquad (1)$$

in which $\{\mathbf{p}_i\}_{i=0}^{P-1}$ means P local neighbors of $\mathbf{q}_c$ that are sampled by a circle of radius R, $\mathcal{I}(.)$ returns the gray-scale value of a pixel, and function $f(.)$ is defined as

$$f(x) = \begin{cases} 1, x \geq 0 \\ 0, \text{otherwise.} \end{cases} \tag{2}$$

The LBP code has thus large dimension of 2^P bins. In practice, different mappings are used to reduce the curse of dimensionality: $u2$ for capturing uniform patterns, $riu2$ for addressing rotation invariant uniform patterns, Local Binary Count [35] - an alternative of uniform features, $TAP^{\mathcal{A}}$ mapping [10] for taking into account topological information.

Guo et al. [7] introduced two more complementary parts so that they can be integrated together with LBP to form CLBP model as follows: CLBP_S that is identical to LBP, CLBP_M that exploits local variations of magnitudes, and CLBP_C that addresses the gray-value difference of a center pixel against the average level of the entire image. In practice, the joint of these components (i.e., CLBP_S/M/C) is utilized because of its effectiveness in performance.

2.2. Smooth-Invariant Gaussian features

A Gaussian filter is a convolution filter whose convoluted results comply with the rule of the Gaussian distribution. Its 2D kernel is defined as follows.

$$\mathrm{G}_\sigma(x, y) = \frac{1}{2\pi\sigma^2}\exp\big(-\frac{x^2+y^2}{2\sigma^2}\big) \tag{3}$$

where σ is a standard derivation, and $x, y \in [-3\sigma, 3\sigma]$ are settings for our implementation. Accordingly, a kernel of the difference of Gaussian filters is defined as

$$\mathrm{DoG}_{\sigma_1,\sigma_2}(x, y) = \mathrm{G}_{\sigma_1}(x, y) - \mathrm{G}_{\sigma_2}(x, y) \tag{4}$$

Two above kernels are used to produce corresponding smooth ($\mathcal{I}_{\mathrm{G}}$) and invariant ($\mathcal{I}_{\mathrm{DoG}}$) filtered images as follows.

$$\begin{cases} \mathcal{I}_{\mathrm{G}} = \mathrm{G}_{\sigma_1}(x, y) * \mathcal{I} \\ \mathcal{I}_{\mathrm{DoG}} = |\mathrm{DoG}_{\sigma_1,\sigma_2}(x, y)| * \mathcal{I} \end{cases} \tag{5}$$

where $\mathcal{I}$ denotes a 2D image texture, $\sigma_1 < \sigma_2$, "*" is the convolution operator. Figure 1 shows an instance of this filtering.

(a) (b) (c) (d)

Fig. 1. A sample of Gaussian filtering. (a): an input gray-scale image; (b), (c): smoothed images of $\sigma_1 = 0.5$, $\sigma_2 = 4$ respectively; (d): the difference of Gaussians of (b) and (c).

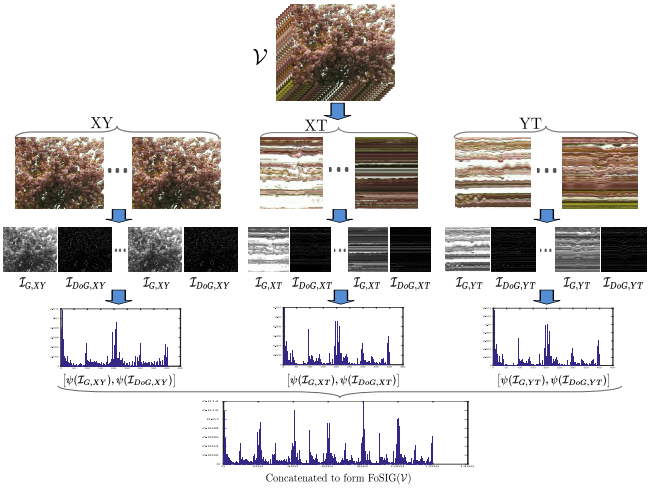

Fig. 2. Our proposed framework of encoding a video $\mathcal{V}$.

2.3. Proposed DT descriptor

Given a video $\mathcal{V}$. For DT representation, we analysis and compute its features in three stages (see Figure 2 for graphical illustration). First, $\mathcal{V}$ is split into separative images according to its three orthogonal planes $\{XY, XT, YT\}$. Second, the Gaussian filtering kernels are taken into account on these planes to extract smooth and invariant images against noise and illumination. Finally, a local encoding function is used for each of them to capture spatio-temporal patterns of DTs. The obtained probability distributions are concatenated and normalized to construct a robust descriptor with features of smooth-invariant Gaussians $\mathrm{FoSIG}(\mathcal{V})$ as follows.

$$\begin{aligned} \mathrm{FoSIG}(\mathcal{V}) = \big[&\psi(\mathcal{I}_{G,XY}), \psi(\mathcal{I}_{DoG,XY}) \\ &\psi(\mathcal{I}_{G,XT}), \psi(\mathcal{I}_{DoG,XT}) \\ &\psi(\mathcal{I}_{G,YT}), \psi(\mathcal{I}_{DoG,YT})\big] \end{aligned} \tag{6}$$

where $\psi(.)$ denotes a local encoding operator (e.g., LBP, CLBP, ...) for capturing local features in an image.

2.4. Advantages of our descriptor

FoSIG improves the performance based on several properties:

- *Robust to illumination and environment changes:* It is well-known that DoG filtered images are invariant against illumination changes. Moreover, by considering the difference between two Gaussians of different scales, they are robust against scale changes. This makes our encoding more robust against following main challenges of DT representation: illumination, scale, and environment changes.
- *Robust to noise:* Local DT features captured on $\mathcal{I}_{\mathrm{G}}$ and $\mathcal{I}_{\mathrm{DoG}}$ are more insensitive to noise in comparison to that in the original image. It should be noted that Gaussian filtering has been introduced together with LBP for

2D texture representation [9]. However, contrariwise to [9], the Gaussian filtering is directly calculated on the entire image in our encoding instead of on neighborhoods at various local scale areas of a pixel.

- *Forceful discrimination:* Mentioned as an approximation of Laplacian of Gaussian (LoG), DoG produces beneficial invariant features for recognition. It is along with the Gaussian smooth filtering to form two critical complemented components in order to significantly enhance the discriminant power (see Table 2).
- *Low computational cost:* It only takes less than 0.92s to structure a video of $48 \times 48 \times 75$ dimension with a raw MATLAB code of our algorithm which is run on a Linux laptop of CPU Intel Core i7 1.9 Ghz, 4G RAM.

3. EXPERIMENTS

In this section, we verify our proposed method for recognition issue on various benchmark DT datasets. The obtained results are evaluated in comparison with the existing approaches. A linear multi-class SVM algorithm of the LIBLINEAR[1] library [5] with the default parameters are used for DT classification.

3.1. Experimental settings

We conduct $\sigma_1 = 0.5$ and $\sigma_2 = \{3, 4, 5, 6\}$ for computing smooth-invariant filtered features. To capture local relationships, we use the popular operator CLBP with joint settings of $riu2$ mapping and $(P, R) = (8, 1)$, i.e., $\psi = \text{CLBP}^{\text{riu2}}_{8,1}$. As the result, the obtained descriptor has dimension of 1200 bins. The pair of $(\sigma_1, \sigma_2) = (0.5, 6)$ is addressed for comparison with the existing methods due to its outperformance. Empirically, values of σ_1 and σ_2 should be in the range of above parameters because the informative appearance of DTs will be sharply diminished when σ_2 is close to σ_1 or $\sigma_2 > 6$.

3.2. Datasets and protocols

UCLA dataset: It includes 200 DT videos which are divided into 50 categories with four sequences per one group [25]. Each original video is recorded in $110 \times 160 \times 75$ dimension to capture textural motions, such as fountain, waterfall, flower, plant, etc. In experiment, its small version with $48 \times 48 \times 75$ sequences of main appearances of DTs is usually addressed and composed more challenging subsets for DT recognition as follows. Table 1 shows a brief of key features for a glance.

- *50-class:* 50 groups using two protocols: *leave-one-out* [3, 29] and *4-fold cross validation* [12, 27].
- *9-class and 8-class:* A sub-dataset of 9 classes is rearranged from the 50 classes consisting of "boiling water" (8), "plants" (108), "sea" (12), "fire" (8), "flowers" (12), "fountains" (20), "smoke" (4), "water" (12), and "waterfall" (16), where the numbers beside indicate their quantities. The "plants" is removed from 9-class to structure 8-class with more challenging [32]. Following [6, 12], a half of samples is randomly chosen for training and the rest for testing. The final rate is returned from the average of 20 runtimes.

[1] https://www.csie.ntu.edu.tw/∼cjlin/liblinear

Table 1. A summary of main properties of DT datasets.

Dataset	Sub-dataset	#Videos	Resolution	#Classes	Protocol
UCLA	50-class	200	$48 \times 48 \times 75$	50	Loo and 4fold
	9-class	200	$48 \times 48 \times 75$	9	50%/50%
	8-class	92	$48 \times 48 \times 75$	8	50%/50%
DynTex	DynTex35	350	different dimensions	10	Loo
	Alpha	60	$352 \times 288 \times 250$	3	Loo
	Beta	162	$352 \times 288 \times 250$	10	Loo
	Gamma	264	$352 \times 288 \times 250$	10	Loo
DynTex++		3600	$50 \times 50 \times 50$	36	50%/50%

Note: Loo and 4fold are leave-one-out and four cross-fold validation respectively. 50%/50% denotes a protocol of taking randomly 50% samples for training and the remain (50%) for testing.

DynTex dataset: It includes more than 650 high-quality videos captured in diversities of environmental conditions [17]. Following [3, 27, 33], a challenging sub-dataset (named DynTex35) with 10 categories is composed by cropping 35 videos as follows: 8 non-overlapping sub-sequences obtained by randomly splitting each video with various cutting points but not in the half of X, Y, and T axes; 2 sub-videos by only splitting along its T axis. Other challenging sub-datasets are also assembled from the original DynTex using leave-one-out protocol for DT recognition as follows: *Alpha* − three categories with 20 videos in each of them, *Beta* − 162 sequences divided into 10 classes with different numbers of items for each, and *Gamma* − 10 classes of 264 DT videos [2, 3].

DynTex++ dataset: Ghanem et al. [6] selected 345 raw videos of DynTex to compose DynTex++. These videos, which are filtered and fixed in size of $50 \times 50 \times 50$ to capture the major chaotic motions, are categorized into 36 classes with 100 items for each, i.e., 3600 DTs in total. Following [3, 6], a half of samples of each class is randomly selected for training and the rest for testing. The final recognition rate is the mean of 10 repetitions.

3.3. Experimental results

It can be seen in Table 2 that taking smooth-invariant features into account describing DTs points out a robust descriptor with outperformance. Specific evaluations of FoSIG descriptor on the datasets are presented in Table 3, in which the highest recognition rates are in bold. The experimental results are then compared to the state of the art in Tables 4 and 5, in which the highest rates are in bold. Evaluations of VLBP and LBP-TOP are referred to the implementations in [27, 18] while the rest are directed from the original works. In general, FoSIG significantly outperforms in comparison to most of the existing methods on various DT datasets for recognition issue. It just operates on DynTex less efficiently than deep-learning-based methods taking enormous cost of computation with complex algorithms for learning DT features. Hereafter, the performance of our proposed descriptor on the

Table 2. Contribution of $\mathcal{I}_{\mathrm{DoG}}$ on DynTex++.

(σ_1, σ_2)	$(0.5, 3)$	$(0.5, 4)$	$(0.5, 5)$	$(0.5, 6)$
G_{σ_1}	95.73	95.73	95.73	95.73
DoG_{σ_1,σ_2}	93.19	93.33	93.52	93.78
G_{σ_1} + DoG_{σ_1,σ_2}	**96.38**	**96.39**	**96.12**	**95.99**

Table 3. Recognition rates (%) on DT benchmark datasets.

Dataset	UCLA				DynTex				
(σ_1, σ_2)	50-Loo	50-4fold	9-class	8-class	Dyn35	Alpha	Beta	Gamma	Dyn++
$(0.5, 3)$	**99.50**	99.50	97.25	**98.59**	98.29	**96.67**	91.98	**93.56**	96.38
$(0.5, 4)$	99.00	**100**	**99.15**	98.48	98.57	**96.67**	91.98	92.05	**96.39**
$(0.5, 5)$	99.00	99.50	97.20	98.04	98.57	**96.67**	**92.59**	91.67	96.12
$(0.5, 6)$	**99.50**	**100**	98.95	**98.59**	**99.14**	**96.67**	**92.59**	92.42	95.99

Note: 50-Loo and 50-4fold denote results on 50-class breakdown using leave-one-out and four cross-fold validation. Dyn35 and Dyn++ are abbreviated for DynTex35 and DynTex++ sub-datasets respectively.

particular datasets is evaluated in detail.

UCLA dataset: It can be verified from Tables 3 and 4 that the proposed descriptor obtains the best results on scenarios of *50-class* (99.50%) and *50-4fold* (100%) compared to all existing methods, including deep-learning-based. In terms of *9-class* and *8-class* with rates of 98.95% and 98.59%, FoSIG nearly has the same performance as DT-CNN's [1]. In the meanwhile, those are the best evaluations among the LBP-based methods, except CVLBC [34]. However, it is not better than ours on other datasets (see Table 5). The highest rate of *9-class* is 99.15% with setting of $(\sigma_1, \sigma_2) = (0.5, 4)$ and of *8-class* is 98.59% at $\sigma_1 = 0.5$ and $\sigma_2 = \{3, 6\}$ (see Table 3).

Table 4. Comparison of recognition rates (%) on UCLA.

Group	Encoding method	50-Loo	50-4fold	9-class	8-class
A	FD-MAP [14]	**99.50**	99.00	99.35	**99.57**
B	AR-LDS [25]	89.90^{N}	-	-	-
	Chaotic vector [30]	-	-	85.10^{N}	85.00^{N}
C	3D-OTF [31]	-	87.10	97.23	99.50
	DFS [32]	-	**100**	97.50	99.20
	STLS [21]	-	99.50	97.40	99.50
D	MBSIF-TOP [3]	**99.50**^{N}	-	-	-
	DNGP [24]	-	-	**99.60**	99.40
E	VLBP [33]	-	89.50^{N}	96.30^{N}	91.96^{N}
	LBP-TOP [33]	-	94.50^{N}	96.00^{N}	93.67^{N}
	CVLBP [26]	-	93.00^{N}	96.90^{N}	95.65^{N}
	HLBP [27]	95.00^{N}	95.00^{N}	98.35^{N}	97.50^{N}
	CLSP-TOP [12]	99.00^{N}	99.00^{N}	98.60^{N}	97.72^{N}
	MEWLSP [28]	96.50^{N}	96.50^{N}	98.55^{N}	98.04^{N}
	WLBPC [29]	-	96.50^{N}	97.17^{N}	97.61^{N}
	CVLBC [34]	98.50^{N}	99.00^{N}	99.20^{N}	99.02^{N}
	CSAP-TOP [13]	**99.50**	99.50	96.80	95.98
	FoSIG	**99.50**	**100**	98.95	98.59
F	DL-PEGASOS [6]	-	97.50	95.60	-
	PI-LBP+super hist [22]	-	**100**^{N}	98.20^{N}	-
	Orthogonal Tensor DL [20]	-	99.80	98.20	99.50
	PCANet-TOP [2]	**99.50***	-	-	-
	DT-CNN-AlexNet [1]	-	99.50*	98.05*	98.48*
	DT-CNN-GoogleNet [1]	-	99.50*	98.35*	99.02*

Note: "-" means "not available". Superscript "*" indicates result using deep learning algorithms. "N" is rate with 1-NN classifier. 50-Loo and 50-4fold denote results on 50-class breakdown using leave-one-out and four cross-fold validation respectively. Group A denotes *optical-flow-based methods*, B: *model-based*, C: *geometry-based*, D: *filter-based*, E: *local-feature-based*, F: *learning-based*.

DynTex dataset: With rate of 99.14%, FoSIG is the best performance on DynTex35, except CSAP-TOP [13] (100%) and MEWLSP [28] (99.71%). However, MEWLSP has not been verified on more challenging datasets (i.e., *Alpha*, *Beta*, *Gamma*) while CSAP-TOP is about 2% lower than ours on *Gamma* and has been missing on DynTex++ (see Table 5). In respect of *Alpha* (96.67%), *Beta* (92.59%), and *Gamma* (92.42%), our proposed descriptor outperforms significantly compared to most of the state of the art, except st-TCoF [18], D3 [8], DT-CNN [1] (see Table 5). These methods use deep learning techniques with sophisticated algorithms to learn DT features while the cost of time is crucial in real applications.

DynTex++ dataset: It can be observed from Table 5 that our result (95.99%) is the best in comparison with all approaches, except MEWLSP [28], HLBP [27], and DT-CNN [1]. However, MEWLSP and HLBP are not better than ours on UCLA as well as have not been verified on more challenging sub-datasets of DynTex (i.e., *Alpha*, *Beta*, *Gamma*). In the meanwhile, DT-CNN with complicated frameworks of AlexNet and GoogleNet takes much time for learning DTs to achieve over 2% better than ours.

Table 5. Comparison of rates (%) on DynTex and DynTex++.

Group	Encoding method	Dyn35	Alpha	Beta	Gamma	Dyn++
A	FD-MAP [14]	98.86	98.33	92.59	91.67	95.69
C	3D-OTF [31]	96.70	83.61	73.22	72.53	89.17
	DFS [32]	97.16	85.24	76.93	74.82	91.70
	2D+T [4]	-	85.00	67.00	63.00	-
	STLS [21]	98.20	89.40	80.80	79.80	94.50
D	MBSIF-TOP [3]	98.61^{N}	90.00^{N}	90.70^{N}	91.30^{N}	97.12^{N}
	DNGP [24]	-	-	-	-	93.80
E	VLBP [33]	81.14^{N}	-	-	-	94.98^{N}
	LBP-TOP [33]	92.45^{N}	98.33	88.89	84.85^{N}	94.05^{N}
	DDLBP with MJMI [23]	-	-	-	-	95.80
	CVLBP [26]	85.14^{N}	-	-	-	-
	HLBP [27]	98.57^{N}	-	-	-	96.28^{N}
	CLSP-TOP [12]	98.29^{N}	95.00^{N}	91.98^{N}	91.29^{N}	95.50^{N}
	MEWLSP [28]	99.71^{N}	-	-	-	98.48^{N}
	WLBPC [29]	-	-	-	-	95.01^{N}
	CVLBC [34]	98.86^{N}	-	-	-	91.31^{N}
	CSAP-TOP [13]	**100**	96.67	92.59	90.53	-
	FoSIG	99.14	96.67	92.59	92.42	95.99
F	DL-PEGASOS [6]	-	-	-	-	63.70
	PCA-cLBP/PI/PD-LBP [22]	-	-	-	-	92.40
	Orthogonal Tensor DL [20]	-	87.80	76.70	74.80	94.70
	Equiangular Kernel DL [19]	-	88.80	77.40	75.60	93.40
	st-TCoF [18]	-	**100***	**100***	98.11*	-
	PCANet-TOP [2]	-	96.67*	90.74*	89.39*	-
	D3 [8]	-	**100***	**100***	98.11*	-
	DT-CNN-AlexNet [1]	-	**100***	99.38*	**99.62***	98.18*
	DT-CNN-GoogleNet [1]	-	**100***	**100***	**99.62***	**98.58***

Note: "-" means "not available". Superscript "*" indicates result using deep learning algorithms. "N" is rate with 1-NN classifier. Dyn35 and Dyn++ are stood for DynTex35 and DynTex++ sub-datasets. Group A denotes *optical-flow-based methods*, B: *model-based*, C: *geometry-based*, D: *filter-based*, E: *local-feature-based*, F: *learning-based*.

4. CONCLUSIONS

We have presented an effective framework for DT description in which the advantages of Gaussian and DoG filtering kernels are taken into account feature encoding to make the proposed descriptor more robust against noise and illumination changes. Assessments of DT recognition on various datasets have validated that our proposal significantly outperforms in comparison with the state-of-the-art results. Analysis in multi-scale solutions can be considered in future works.

5. REFERENCES

[1] V. Andrearczyk and P. F. Whelan. Convolutional neural network on three orthogonal planes for dynamic texture classification. *Pattern Recognition*, 76:36 – 49, 2018.

[2] S. R. Arashloo, M. C. Amirani, and A. Noroozi. Dynamic texture representation using a deep multi-scale convolutional network. *JVCIR*, 43:89 – 97, 2017.

[3] S. R. Arashloo and J. Kittler. Dynamic texture recognition using multiscale binarized statistical image features. *IEEE Trans. Multimedia*, 16(8):2099–2109, 2014.

[4] S. Dubois, R. Péteri, and M. Ménard. Characterization and recognition of dynamic textures based on the 2d+t curvelet transform. *SIVP*, 9(4):819–830, 2015.

[5] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin. LIBLINEAR: A library for large linear classification. *JMLR*, 9:1871–1874, 2008.

[6] B. Ghanem and N. Ahuja. Maximum margin distance learning for dynamic texture recognition. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *ECCV*, volume 6312 of *LNCS*, pages 223–236, 2010.

[7] Z. Guo, L. Zhang, and D. Zhang. A completed modeling of local binary pattern operator for texture classification. *IEEE Trans. IP*, 19(6):1657–1663, 2010.

[8] S. Hong, J. Ryu, W. Im, and H. S. Yang. D3: recognizing dynamic scenes with deep dual descriptor based on key frames and key segments. *Neurocomputing*, 273:611–621, 2018.

[9] T. Mäenpää and M. Pietikäinen. Multi-scale binary patterns for texture analysis. In *SCIA*, pages 885–892, 2003.

[10] T. P. Nguyen, A. Manzanera, W. G. Kropatsch, and X. S. N'Guyen. Topological attribute patterns for texture recognition. *Pattern Recogn Lett*, 80:91–97, 2016.

[11] T. P. Nguyen, N. Vu, and A. Manzanera. Statistical binary patterns for rotational invariant texture classification. *Neurocomputing*, 173:1565–1577, 2016.

[12] T. T. Nguyen, T. P. Nguyen, and F. Bouchara. Completed local structure patterns on three orthogonal planes for dynamic texture recognition. In *IPTA*, pages 1–6, 2017.

[13] T. T. Nguyen, T. P. Nguyen, and F. Bouchara. Completed statistical adaptive patterns on three orthogonal planes for recognition of dynamic textures and scenes. *J. Electronic Imaging*, 27(05):053044, 2018.

[14] T. T. Nguyen, T. P. Nguyen, F. Bouchara, and X. S. Nguyen. Directional beams of dense trajectories for dynamic texture recognition. In J. Blanc-Talon, D. Helbert, W. Philips, D. Popescu, and P. Scheunders, editors, *ACIVS*, pages 74–86, 2018.

[15] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. PAMI*, 24(7):971–987, 2002.

[16] C. Peh and L. F. Cheong. Synergizing spatial and temporal texture. *IEEE Trans. IP*, 11(10):1179–1191, 2002.

[17] R. Péteri, S. Fazekas, and M. J. Huiskes. Dyntex: A comprehensive database of dynamic textures. *Pattern Recogn Lett*, 31 (12):1627–1632, 2010.

[18] X. Qi, C.-G. Li, G. Zhao, X. Hong, and M. Pietikainen. Dynamic texture and scene classification by transferring deep image features. *Neurocomputing*, 171:1230 – 1241, 2016.

[19] Y. Quan, C. Bao, and H. Ji. Equiangular kernel dictionary learning with applications to dynamic texture analysis. In *CVPR*, pages 308–316, 2016.

[20] Y. Quan, Y. Huang, and H. Ji. Dynamic texture recognition via orthogonal tensor dictionary learning. In *ICCV*, pages 73–81, 2015.

[21] Y. Quan, Y. Sun, and Y. Xu. Spatiotemporal lacunarity spectrum for dynamic texture classification. *CVIU*, 165:85–96, 2017.

[22] J. Ren, X. Jiang, and J. Yuan. Dynamic texture recognition using enhanced LBP features. In *ICASSP*, pages 2400–2404, 2013.

[23] J. Ren, X. Jiang, J. Yuan, and G. Wang. Optimizing LBP structure for visual recognition using binary quadratic programming. *SPL*, 21(11):1346–1350, 2014.

[24] A. R. Rivera and O. Chae. Spatiotemporal directional number transitional graph for dynamic texture recognition. *IEEE Trans. PAMI*, 37(10):2146–2152, 2015.

[25] P. Saisan, G. Doretto, Y. N. Wu, and S. Soatto. Dynamic texture recognition. In *CVPR*, pages 58–63, 2001.

[26] D. Tiwari and V. Tyagi. Dynamic texture recognition based on completed volume local binary pattern. *MSSP*, 27(2):563–575, 2016.

[27] D. Tiwari and V. Tyagi. A novel scheme based on local binary pattern for dynamic texture recognition. *CVIU*, 150:58–65, 2016.

[28] D. Tiwari and V. Tyagi. Dynamic texture recognition using multiresolution edge-weighted local structure pattern. *Comput Electr Eng*, 62:485–498, 2017.

[29] D. Tiwari and V. Tyagi. Improved weber's law based local binary pattern for dynamic texture recognition. *Multimedia Tools Appl.*, 76(5):6623–6640, 2017.

[30] Y. Wang and S. Hu. Chaotic features for dynamic textures recognition. *Soft Computing*, 20(5):1977–1989, 2016.

[31] Y. Xu, S. B. Huang, H. Ji, and C. Fermüller. Scale-space texture description on sift-like textons. *CVIU*, 116(9):999–1013, 2012.

[32] Y. Xu, Y. Quan, Z. Zhang, H. Ling, and H. Ji. Classifying dynamic textures via spatiotemporal fractal analysis. *Pattern Recognition*, 48(10):3239–3248, 2015.

[33] G. Zhao and M. Pietikäinen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Trans. PAMI*, 29(6):915–928, 2007.

[34] X. Zhao, Y. Lin, and J. Heikkilä. Dynamic texture recognition using volume local binary count patterns with an application to 2d face spoofing detection. *IEEE Trans. Multimedia*, 20(3): 552–566, 2018.

[35] Y. Zhao, D.-S. Huang, and W. Jia. Completed Local Binary Count for Rotation Invariant Texture Classification. *IEEE Trans. IP*, 21(10):4492–4497, 2012.