Rubik Gaussian-based patterns for dynamic texture classification

Thanh Tuan Nguyen, Thanh Phuong Nguyen, Frédéric Bouchara

Rubik Gaussian-based patterns for dynamic texture classification

Thanh Tuan Nguyen and Thanh Phuong Nguyen and Frédéric Bouchara [*]

Abstract

Illumination, noise, and changes of environments, scales negatively impact on encoding chaotic motions for dynamic texture (DT) representation. This paper proposes a new method to overcome those issues by addressing the following novel concepts. First, different Gaussian-based kernels are taken into account as an effective filtered pre-processing with low computational cost to point out robust and invariant features. Second, a discriminative operator, named Local Rubik-based Pattern (LRP), is introduced to adequately capture both shape and motion cues of DTs by proposing a new concept of complemented components together with an effective encoding method. In addition, it also addresses a novel thresholding to take into account rich spatio-temporal relationships extracted from a new model of neighborhood supporting region. Finally, an efficient framework for DT description is presented by exploiting operator LRP for encoding various instances of Gaussian-based volumes in order to form a robust descriptor against noise, changes of illumination, scale, and environment. Experiments for DT classification on benchmark datasets have authenticated the interest of our proposal.

1 Introduction

Analysis to perceive dynamic textures (DTs), textural features "moving" in a temporal domain, plays an important role in numerous applications of computer vision. Due to the turbulent expansions of DTs in sequences, those intensify challenges in video representation. Many efforts have nominated diverse techniques to effectively characterize their complex motions for DT recognition issue. It can be practically categorized them into the following groups. First, for DT description in natural ways, *optical-flow-based methods* take into account direction and magnitude information of normal flows for capturing chaotic motion features of DTs [1, 2]. Second, thanks to discriminant power against the changes of environmental elements in video encoding, *filter-based methods* have favorable results in DT classification [3, 4].

[*] All authors work at Université de Toulon, Aix Marseille Université, CNRS, LIS, Marseille, France

In the meanwhile, *model-based methods* principally exploit Linear Dynamical System (LDS) [5] to model textural appearance and motions [6]. LDS's derivations are then extended to be in accordance with modeling DT properties, such as bag-of-words (BoW) [7], bag-of-systems (BoS) [8], and BoS Tree [9]. Fourth, instead of using filtering techniques, *geometry-based methods* address fractal analysis to deal with the environmental changes, such as Dynamic Fractal Spectrum (DFS) [10], Multi-Fractal Spectrum (MFS) [11], Wavelet-based MFS (WMFS) [12], Spatio-Temporal Lacunarity Spectrum (STLS) [13], and Stationary Subspace Analysis (SSA) [14]. Fifth, in terms of *learning-based methods*, two kinds of techniques are frequently taken into account for learning DT features: *i)* based on derivations of Convolutional Neural Networks (CNNs) to capture deep structures of DTs [15, 16, 17], and *ii)* utilizing kernel sparse coding for learning dictionaries to make the complex motions of DTs more receptive [18, 19]. Finally, *local-feature-based methods* have considerable attention thanks to their performances with low computation for DT representation. [20] nominated Local Binary Pattern (LBP) [21] for analyzing videos in two ways: VLBP patterns, which are structured from three consecutive frames of a sequence, and LBP-TOP patterns from three orthogonal planes. After that, some works proposed various schemes to enhance the distinguishing power by dealing with limitations of the typical LBPs in DT encoding such as problems of rotation-invariant [22], near-uniform regions, and sensitivity to noise [23, 24, 25, 26].

Although many efforts have been made for DT description, most of them gain modest results due to partly the negative impacts of environmental changes and other affected elements. Recently, the deep-learning methods can cover those problems using deep models. However, the following restrictions have prevented them from deployment in real-time applications of computer vision: a huge volume of parameters which is needed for modeling DT features prevents an application on embedded devices, non-unique parameters are addressed for learning from all datasets [16], failure of transfer-based learning approaches in case of extracting DT features from strange videos [17]. In the meanwhile, the local-feature-based methods have achieved promising results in DT recognition using simple computations contrary to those in the deep learning techniques. In spite of that, they remain several limitations, such as sensitivity to environmental changes: illumination, scales, noise, and near-uniform regions. To mitigate these drawbacks, we proposed in this paper crucial extensions of our previous work [27] to completely construct an efficient framework for DT representation. Our proposal has the following prominent contributions:

- Analysis of Gaussian-based kernels to reduce the negative impacts from noise and changes of environments.
- A completed model for better exploiting rich spatio-temporal relationships extracted from a novel concept of supporting region.
- A novel operator LRP, constructed from complementary components,

allows to effectively capture both shape and motion cues around a cube centered at each voxel thanks to a novel machanism of encoding and thresholding.

- A robust descriptor is structured using a simple computation for exploiting blurred-invariant features of Gaussian-based volumes at different scales.

2 Related works

Extracting DT features with a simple computation, the LBP operator [21] and its variants have formed robust descriptors with competitive performances for DT classification task. In this section, we take a brief of them as well as an overview of the n-dimensional Gaussian-based filtering functions used as pre-processing steps against problems of environmental changes in video sequences.

2.1 A brief review of LBP and its completed model

A well-known operator LBP encodes a center pixel $\mathbf{q}_c$ of 2D gray-scale image $\mathcal{I}$ as a binary string in consideration of local relationships of $\mathbf{q}_c$ and its regional neighbors $\{\mathbf{p}_i\}$ as follows.

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

in which P means local regarding neighbors addressed by a interpolated computation on a circle with radius R, $\mathcal{I}(.)$ returns the gray-value of a pixel, and function $f(.)$ is defined as

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

Consequently, it takes 2^P diverse values to structure a histogram for textural image description. Due to the curse of dimensionality, LBP's utilization for real applications can be unfeasible. In practice, the LBP patterns are often matched with two following popular mappings for dimensional reduction: $u2$ mapping for uniform patterns with $P(P-1)+3$ bins, and $riu2$ for uniform rotation-invariant patterns with only $P+2$ bins [21]. Other influential mappings have been introduced to refine patterns, such as Local Binary Count - a substitution for choosing uniform patterns [28], $TAP^{\mathcal{A}}$ for mapping topological information [29].

[30] presented a completed model of LBP (CLBP) by incorporating three following crucial components: CLBP_S that is equal to the typical LBP, CLBP_M for capturing magnitudes of gray-level differences between a center pixel and its neighbors with the average of those on the whole image, CLBP_C

for measuring the gray-value divergence of a pixel versus the mean of that on the entire image. Those components are integrated into various modes to boost the performance.

2.2 Gaussian-based filtering functions

A Gaussian filtering is a process of convolving a Gaussian kernel on a spatial domain. Its outcome should be agreed with the regulation of a Gaussian distribution. In general, a n-dimensional Gaussian kernel is defined as follows.

$$\mathrm{G}_{\sigma}^{n}\big(\{x_i\}_{i=1}^{n}\big) = \frac{1}{(\sigma\sqrt{2\pi})^n}\exp\Big(-\frac{x_1^2 + x_2^2 + ... + x_n^2}{2\sigma^2}\Big) \tag{3}$$

in which σ means a pre-defined standard deviation, n denotes a number of spatial axes $\varphi_n = \{x_i\}_{i=1}^{n}$ that are taken into account in the convolving operation. Accordingly, a kernel of the Difference of Gaussian (DoG) filters with $\sigma < \sigma'$ is defined as

$$\mathrm{DoG}_{\sigma,\sigma'}^{n}(\varphi_n) = \mathrm{G}_{\sigma}^{n}(\varphi_n) - \mathrm{G}_{\sigma'}^{n}(\varphi_n) \tag{4}$$

3 Proposed method

3.1 Overview

We introduce an efficient framework, as illustrated in Fig. 1, to construct a robust DT descriptor based on extracting blurred and invariant spatio-temporal features with forceful robustness to illumination and noise. To this end, first, Gaussian-based kernels are taken into account in Section 3.2 for pre-processing an input video $\mathcal{V}$ in order to point out blurred volumes $\mathcal{V}^G$ with more intensity to noise and invariant sequences $\mathcal{V}^{DoG}$ against changes of environmental elements. Second, we introduce the notion of complemented components in Section 3.3, inspired from [30], in order to better capture spatio-temporal relationships than typical LBP-based variants for DT representation such as LBP-TOP, VLBP, etc. Third, this allows us to propose a novel operator LRP in Section 3.4 based on the concept of neighborhood configuration considering local relationships between a voxel and its neighbors interpolated from 6 sides and 3 orthogonal plane-images of a cube. From now on, we call it a rubik cube because the neighborhood supporting region has a similar shape to a rubik cube (see Fig. 2(b)). In addition, it is somewhat homonym with RUBIG features presented in Section 3.5. Taking advantage of beneficial properties of VLBP, LBP-TOP, and CLBP allows LRP to effectively capture spatio-temporal features in concern with a full space investigation. We then introduce a discriminative descriptor, namely RUBIG (Rubik Blurred-Invariant Gaussian features), which is formed by utilizing LRP for capturing space-completed features in different scales of

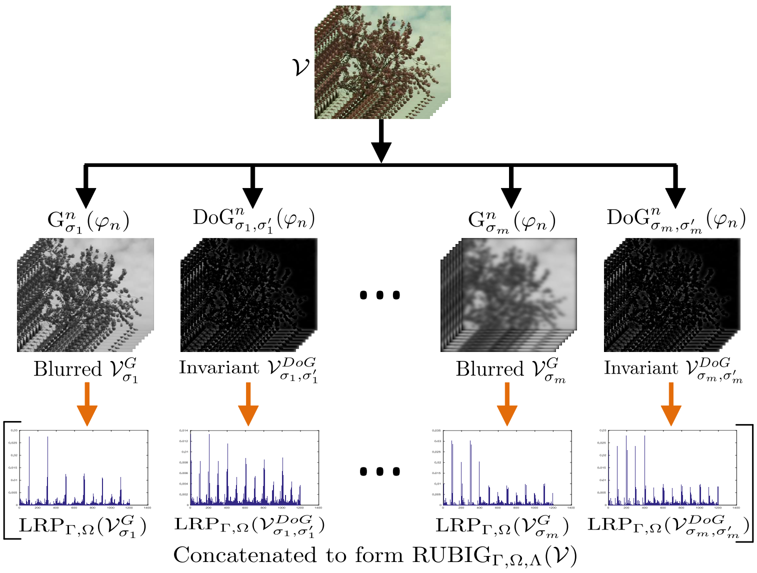
$\mathcal{V}$
$\mathrm{G}^n_{\sigma_1}(\varphi_n)$
$\mathrm{DoG}^n_{\sigma_1,\sigma'_1}(\varphi_n)$
$\mathrm{G}^n_{\sigma_m}(\varphi_n)$
$\mathrm{DoG}^n_{\sigma_m,\sigma'_m}(\varphi_n)$
Blurred $\mathcal{V}^G_{\sigma_1}$
Invariant $\mathcal{V}^{DoG}_{\sigma_1,\sigma'_1}$
Blurred $\mathcal{V}^G_{\sigma_m}$
Invariant $\mathcal{V}^{DoG}_{\sigma_m,\sigma'_m}$
$\mathrm{LRP}_{\Gamma,\Omega}(\mathcal{V}^G_{\sigma_1})$
$\mathrm{LRP}_{\Gamma,\Omega}(\mathcal{V}^{DoG}_{\sigma_1,\sigma'_1})$
$\mathrm{LRP}_{\Gamma,\Omega}(\mathcal{V}^G_{\sigma_m})$
$\mathrm{LRP}_{\Gamma,\Omega}(\mathcal{V}^{DoG}_{\sigma_m,\sigma'_m})$
Concatenated to form $\mathrm{RUBIG}_{\Gamma,\Omega,\Lambda}(\mathcal{V})$

Figure 1: Illustration of proposed framework for DT representation.

Gaussian-based filtered volumes (see Section 3.5). Finally, Section 3.6 discusses a computational analysis of LBP-based models for DT representation.

3.2 Gaussian-based filtering

Filter-bank approach, which has been early applied for texture analysis since years of 90s [31] , was also considered for DT representation in recent works [32, 3, 33]. Moreover, filter-bank and LBP-based approaches have been also addressed together in [34] for an effective texture representation. Inspired from this approach, we address Gaussian-based filters to overcome well-known issues in DT description: the influence of noise, changes of environments, scales and illumination, etc. Indeed, two complementary families of filtering are taken into account for this purpose. First, Gaussian filters G^n_σ are used to produce blurred volumes $\mathcal{V}^G$ which are more robust against noise. Second, DoG filters are addressed to figure out invariant volumes $\mathcal{V}^{DoG}$ which is robust against changes of illuminations and scales. It should be noted that Gaussian distribution has been also used in a totally different way in [35] to simulate image texture by stationary Gaussian random fields. We point out hereafter the following beneficial properties of our approach inheriting from these Gaussian-based filters.

- *Robustness to changes of illumination, scales, and environment:* Gaussian-based filtered volumes $\mathcal{V}^{DoG}_{\sigma_i,\sigma'_i}$ are invariant sequences against illumination thanks to exploiting various scales of Gaussian filtering kernels. In addition, the receptive $\mathcal{V}^{DoG}_{\sigma_i,\sigma'_i}$ volumes, formed by two different Gaussian kernels, allow to capture features with more robustness to the major remaining problems of DT description: illumination, scale, and environmental changes.

- *Robustness to noise:* Instead of extracting features from a raw video $\mathcal{V}$, its Gaussian-based filtered volumes $\mathcal{V}^G$ allow to capture local features with more intensity to noise. On the other hand, DoG features are also exploited in our proposal to make descriptor more robust against changes of environment and illumination.
- *Forceful incisive elements:* Well-known as an approximation of Laplacian of Gaussian (LoG), $\mathcal{V}^{DoG}_{\sigma_i,\sigma'_i}$ sequences provide beneficial receptive clues for feature encoding. Meantime, $\mathcal{V}^G_{\sigma_i}$ volumes produce robust blurred features for the description. Consequently, the performance of DT recognition is enhanced thanks to these supplementary filtered volumes (see Table 3 for their contributions).

3.3 Complemented components

Motivated by the conception of complemented components in [30, 23, 36], three prominent components are proposed to address forceful discrimination of local textural features by adapting the concept to the supporting region constructed from 6 sides of a rubik cube and by introducing new concepts of encoding and thresholding dedicated to this neighborhood configuration for three completed components (see Equations (6), (8), (9), and (11)). Accordingly, let $\mathbf{q}$ be a voxel in a video $\mathcal{V}$; $\mathbf{q}_f$ be its projection on a plane-image $f \in \mathcal{V}$ (see Fig. 2(a) for a graphical illustration). Figure 2(b) presents our neighborhood supporting region which is constructed from 6 sides of a rubik cube centered at the voxel together with 3 orthogonal planes passing through this voxel. The first component captures the differences between the mean gray-level center points (i.e., $\mathbf{q}$, $\mathbf{q}_f$) and each of $\{\mathbf{p}_{i,f}\}$ local neighbors of $\mathbf{q}_f$ as follows.

$$\mathrm{D}_{P,R,f}(\mathbf{q},\mathbf{q}_f) = \big\{s\big(\mathcal{I}(\mathbf{p}_{i,f}),\mathcal{I}(\mathbf{q}_f),\mathcal{I}(\mathbf{q})\big)\big\}_{i=0}^{P-1} \tag{5}$$

where P denotes the number of considered neighbors interpolated on a circle of radius R, $\mathcal{I}(.)$ returns the gray-scale of an image pixel, the binary function s(.) is defined as

$$s(x,y,z) = \begin{cases} 1, x \geq \frac{y+z}{2} \\ 0, \text{otherwise.} \end{cases} \tag{6}$$

The second conducts informative magnitudes by comparing the gray-level differences in the first component with the average of them $\overline{m}_f$ computed for the whole textural region as follows.

$$\mathrm{M}_{P,R,f}(\mathbf{q},\mathbf{q}_f) = \big\{h\big(\mathcal{I}(\mathbf{p}_{i,f}),\mathcal{I}(\mathbf{q}_f),\mathcal{I}(\mathbf{q}),\overline{m}_f\big)\big\}_{i=0}^{P-1} \tag{7}$$

in which $\overline{m}_f$ and function $h(.)$ are defined in (8) and (9) respectively, $\mathcal{N}$ means the quantity of pixels $\{\mathbf{q}_j\}$ in current image f.

$$\overline{m}_f = \frac{1}{P \times \mathcal{N}} \sum_{j=0}^{\mathcal{N}} \sum_{i=0}^{P-1} \Big(\mathcal{I}(\mathbf{p}_{i,f}) - \frac{\mathcal{I}(\mathbf{q}_{j,f}) + \mathcal{I}(\mathbf{q})}{2} \Big) \tag{8}$$

$$h(x, y, z, t) = \begin{cases} 1, x - \frac{y+z}{2} \geq t \\ 0, \text{otherwise.} \end{cases} \tag{9}$$

The third component features central differences of the mean gray-level of the center points (i.e., $\mathbf{q}$ and $\mathbf{q}_f$) versus the average of them $\overline{c}_f$ calculated for the entire plane image f as follows.

$$\mathrm{C}_{P,R,f}(\mathbf{q}, \mathbf{q}_f) = g\big(\mathcal{I}(\mathbf{q}_f) + \mathcal{I}(\mathbf{q}) - \overline{c}_f\big) \tag{10}$$

where $g(.)$ is identical to Equation (2) and $\overline{c}_f$ is computed as

$$\overline{c}_f = \frac{1}{\mathcal{N}} \sum_{j=0}^{\mathcal{N}} \Big(\mathcal{I}(\mathbf{q}_{j,f}) + \mathcal{I}(\mathbf{q}) \Big) \tag{11}$$

Those components are complementary [30]. Therefore, their integration is recommended in order to improve the discriminant power. Let $\mathrm{DMC}_{P,R,\Omega}(.)$ be an integration Ω of the complemented components (i.e., $\mathrm{D}_{P,R,f}(.)$, $\mathrm{M}_{P,R,f}(.)$, $\mathrm{C}_{P,R,f}(.)$) subject to each voxel. For instance, $\mathrm{DMC}_{P,R,\Omega}(\mathbf{q}, \mathbf{q}_{f_{i-1}})$ computes $\mathrm{D}_{P,R,f_{i-1}}(\mathbf{q}, \mathbf{q}_{f_{i-1}})$, $\mathrm{M}_{P,R,f_{i-1}}(\mathbf{q}, \mathbf{q}_{f_{i-1}})$, and $\mathrm{C}_{P,R,f_{i-1}}(\mathbf{q}, \mathbf{q}_{f_{i-1}})$ based on $\mathbf{q}$'s central symmetry voxel $\mathbf{q}_{f_{i-1}}$ at image f_{i-1} in plane XY (see Fig. 2(c) for a sample of this computation). Those are then integrated into different ways Ω to form space-completed patterns. Therein, $\Omega = \{_{D\_M/C}, {}_{D/M/C}, \text{etc.}\}$ where signs “_” and “/” mean operations of concatenating and jointing probability distributions of the components respectively, e.g., “$_{D\_M/C}$” indicates that a joint histogram of M(.) and C(.) is concatenated to that of D(.).

3.4 Local Rubik-based Pattern (LRP)

Based on the concept of complemented model in the previous section, we introduce hereafter the novel LRP operator. For a video $\mathcal{V}$, let a center voxel $\mathbf{q} \in \mathcal{V}$ be an intersection point of orthogonal plane images $f_i \in XY$, $f_j \in XT$, and $f_k \in YT$ where $\{XY, XT, YT\}$ are planes of $\mathcal{V}$. A rubik cube Γ of $\mathbf{q}$ is addressed in consideration of the previous and posterior plane-images of f_i, f_j, and f_k respectively (i.e., f_{i-1}, f_{i+1} for XY, f_{j-1}, f_{j+1} for XT, f_{k-1}, f_{k+1} for YT, see Fig. 2(b) for a graphical instance). A local rubik-based pattern for $\mathbf{q}$ is structured by integrating complementary components computed on 6 sides and 3 orthogonal plane-images of rubik cube Γ as follows.

$$\mathrm{LRP}_{\Gamma,\Omega}(\mathbf{q}) = \biguplus_{f \in \mathcal{F}} \big[\mathrm{DMC}_{P,R,\Omega}(\mathbf{q}, \mathbf{q}_f)\big] \tag{12}$$

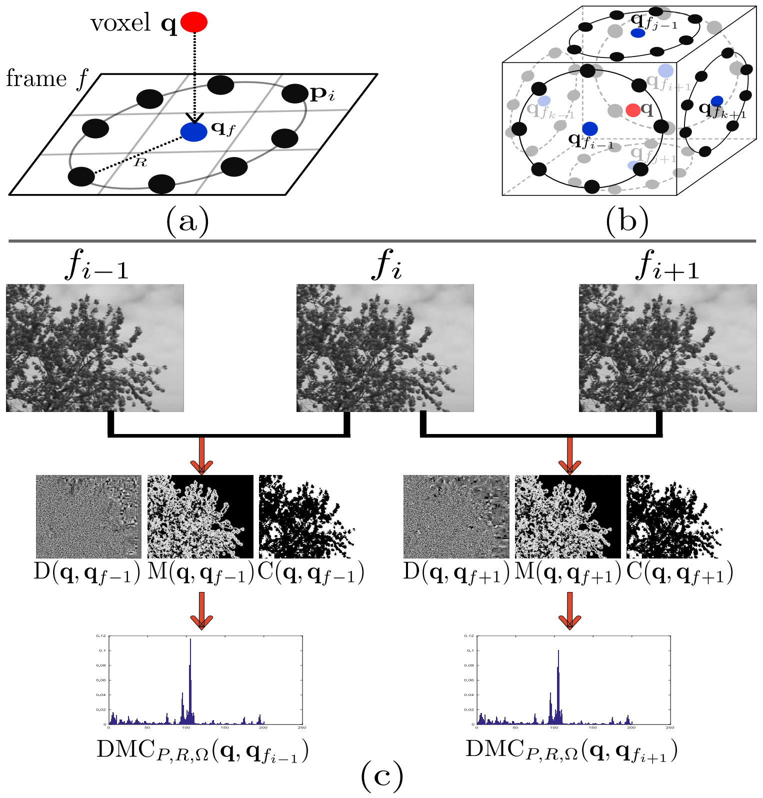
voxel $\mathbf{q}$
frame f
$\mathbf{p}_i$
$\mathbf{q}_f$
R
(a)
$\mathbf{q}_{f_{j-1}}$
$\mathbf{q}_{f_{i+1}}$
$\mathbf{q}_{f_{k-1}}$
$\mathbf{q}$
$\mathbf{q}_{f_{k+1}}$
$\mathbf{q}_{f_{i-1}}$
$\mathbf{q}_{f_{j+1}}$
(b)
f_{i-1}
f_i
f_{i+1}
$\mathrm{D}(\mathbf{q}, \mathbf{q}_{f-1})$ $\mathrm{M}(\mathbf{q}, \mathbf{q}_{f-1})$ $\mathrm{C}(\mathbf{q}, \mathbf{q}_{f-1})$
$\mathrm{D}(\mathbf{q}, \mathbf{q}_{f+1})$ $\mathrm{M}(\mathbf{q}, \mathbf{q}_{f+1})$ $\mathrm{C}(\mathbf{q}, \mathbf{q}_{f+1})$
$\mathrm{DMC}_{P,R,\Omega}(\mathbf{q}, \mathbf{q}_{f_{i-1}})$
$\mathrm{DMC}_{P,R,\Omega}(\mathbf{q}, \mathbf{q}_{f_{i+1}})$
(c)

Figure 2: (Best viewed in color) Computing parts of our framework. (a): A model of encoding feature for a voxel $\mathbf{q}$ (in red) based on its central symmetry voxel $\mathbf{q}_f$ (in blue) on plane image f. (b): A graphical illustration of LRP construction at voxel $\mathbf{q}$. (c): A calculation of an integrated histogram DMC(.) for voxels $\{\mathbf{q} \in f_i\}$ along with their symmetry points in images f_{i-1} and f_{i+1} of plane XY in a video.

in which $\biguplus$ denotes a concatenating function of histograms,

$$\mathcal{F} = \{f_{i-1}, f_i, f_{i+1}, f_{j-1}, f_j, f_{j+1}, f_{k-1}, f_k, f_{k+1}\}$$

is a set of 6 sides and 3 orthogonal plane-images of rubik cube Γ, $\mathbf{q}_f$ is the central symmetry voxel of $\mathbf{q}$ that is orthogonally projected on plane-image f (see Fig. 2(a) for an instance of a projection of $\mathbf{q}$).

Our LRP operator is different from LBP-based variants in several properties to improve the performance:

- LRP structures a voxel in consideration of rich spatio-temporal relationships extracted from 6 sides of the rubik cube (see Fig. 2(b)) while other LBP-based variants mostly based on three orthogonal planes for DT representation [24, 37].
- Discriminative information of a center voxel is embedded into encoding side patterns against near-uniform regions.

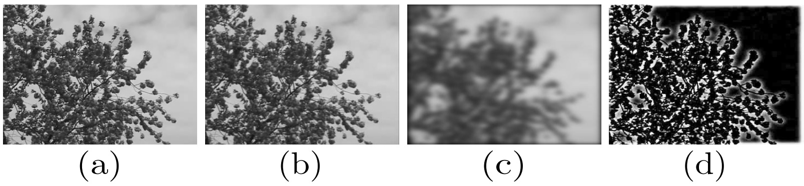

Figure 3: An instance of 3D Gaussian-based filters. (a) is an input gray-scale frame of a DT video. (b) and (c) are 3D smoothed frames of (a) using $\sigma_1 = 0.5$ and $\sigma_2 = 4$ respectively. (d) denotes the 3D DoG of (b) and (c).

- Based on a block shape, LRP is more suitable for encoding DT videos than LBP-based variants which are separately applied to still images of the planes in videos.
- By addressing previous and posterior plane-images, LRP can capture changes of a voxel in global spatio-temporal appearances. In the meanwhile, VLBP for structuring temporal appearances in plane XY, and LBP-TOP for addressing local orthogonal patterns [20].

3.5 RUbik Blurred-Invariant Gaussian (RUBIG) features

As a derivation of the LBP-based computation, encoding rubik-based patterns can be faced with sensitivity to noise and illumination problems. To treat those, Gaussian-based filtering kernels in Equations (3) and (4) are addressed as a pre-processing step to reduce the negative impacts of environmental changes on DT representation. It should be noted that Gaussian filter has been addressed together with LBP operator in [38]. However, it employed a 2D Gaussian kernel to analyze neighborhoods at different area scales of a pixel for texture description, while [39] utilized it for capturing spatio-temporal features from filtered images of planes in a video. Accordingly, for a video $\mathcal{V}$ along with pre-defined couples of standard deviations $\Lambda = \{(\sigma_i, \sigma'_i)\}_{i=1}^{m}$, a series of volumes of blurred Gaussian features $\mathcal{V}_{\sigma_i}^{G}$ and the difference of Gaussians $\mathcal{V}_{\sigma_i,\sigma'_i}^{DoG}$ are computed as follows. Figure 3 shows several samples of this filtering.

$$\mathcal{V}_{\sigma_i}^{G} = \mathrm{G}_{\sigma_i}^{n}(\varphi_n) * \mathcal{V}, \text{ and } \mathcal{V}_{\sigma_i,\sigma'_i}^{DoG} = |\mathrm{DoG}_{\sigma_i,\sigma'_i}^{n}(\varphi_n)| * \mathcal{V} \tag{13}$$

where "$*$" is a convolving operator, and $\sigma_i < \sigma'_i$. We then utilize the proposed LRP operator for each filtered volume to capture RUbik Blurred-Invariant Gaussian (RUBIG) features for DT description (see Fig. 1 for a graphical illustration of this construction). The obtained histograms are then normalized and concatenated to form a discriminative descriptor.

$$\mathrm{RUBIG}_{\Gamma,\Omega,\Lambda}(\mathcal{V}) = \left[\mathrm{LRP}_{\Gamma,\Omega}(\mathcal{V}_{\sigma_i}^{G}), \mathrm{LRP}_{\Gamma,\Omega}(\mathcal{V}_{\sigma_i,\sigma'_i}^{DoG})\right]_{i=1}^{m} \tag{14}$$

Our RUBIG is based on two important properties to boost its performance compared to that of V-BIG [27] (see Table 3 for a specific performing comparison): *i)* RUBIG is enriched by rich spatio-temporal features thanks to our novel, discriminative operator LRP. *ii)* RUBIG can be better resistant to the illumination and noise since its blurred-invariant features are encoded from multi-scale Gaussian-based volumes. Besides the beneficial properties inheriting from Gaussian-based filtering (see Section 3.2), our RUBIG has also following properties.

- *Multi-scale and rich spatio-temporal features:* RUBIG is concerned with analysis of rich spatio-temporal features to form an effective descriptor that is more discriminative than CLBP features of V-BIG. Moreover, it is enriched by robust clues based on various scales of Gaussian kernels taken into account the filtering, while V-BIG is lack of multi-scale analysis due to just a single-scale involved in.
- *Informative voxel discrimination:* Shape and motion cues are jointly structured thanks to voxels in a DT video enriched by discriminative information with 3D Gaussian kernels. In the meanwhile, FoSIG [39] just captures spatio-temporal features of voxels on 2D Gaussian filtered images of the planes in the video.

3.6 Complexity of RUBIG and other LBP-based descriptors

Typically, the complexity of structuring our RUBIG is as simple as that of the LBP-based variants for DT representation. Indeed, for a video $\mathcal{V}$ with $\mathcal{H} \times \mathcal{W} \times \mathcal{T}$ dimension, let $Q_{\text{LBP-TOP}} = \mathcal{O}(P \times \mathcal{H} \times \mathcal{W} \times \mathcal{T})$ be the complexity of computing the basic LBP-TOP patterns for encoding $\mathcal{V}$ based on three orthogonal planes $\{XY, XT, YT\}$, where P is the number of considered neighbors (refer to [20] for more detail). It can be conduced that the complexities of calculating descriptors FoSIG and V-BIG are $Q_{\text{FoSIG}} = Q_{\text{V-BIG}} \approx 3 \times Q_{\text{LBP-TOP}} + Q_G$ since they are based on 3 CLBP's components that are computed independently (refer to [30] for more detail). Therein, Q_G denotes the computational cost of Gaussian-based filterings in general. Also, our LRP is formed by independent computations of its components (i.e., D, M, and C (see Section 3.3)) that are based on 6 sides and 3 orthogonal plane-images of a rubik cube (see Section 3.4), i.e., $Q_{\text{LRP}} \approx 9 \times Q_{\text{LBP-TOP}}$. Therefore, it can be deduced from Equation (14) that the RUBIG's complexity is estimated as $Q_{\text{RUBIG}} = 2 \times m \times (Q_{\text{LRP}} + Q_G)$. Due to the much smaller value of parameter m compared to the others, as well as the separable property of Gaussian filtering, they can be ignored. Consequently, $Q_{\text{RUBIG}} \approx \mathcal{O}(P \times \mathcal{H} \times \mathcal{W} \times \mathcal{T})$. In addition, our Q_{RUBIG} is also the same order as that of other LBP-based descriptors: CSAP-TOP [37], CVLBC [36], CVLBP [23], VLBP [20], etc. (refer to these works for detail). In terms of processing time, we have implemented some of them and made a comparison with ours (see Table 1). It is noteworthy that raw MATLAB codes of these implementations are run on a 64-bit Linux desktop

Table 1: Comparison of processing time of encoding a video with $50 \times 50 \times 50$ dimension in DynTex++ dataset.

Descriptor	$\{(\sigma, \sigma')\}$	$\{(P, R)\}$	Mapping	Runtime (s)
VLBP [20]	-	$\{(4, 1)\}$	-	≈ 0.22
LBP-TOP [20]	-	$\{(8, 1)\}$	u2	≈ 0.15
CLSP-TOP [40]	-	$\{(8, 1)\}$	riu2	≈ 0.27
CSAP-TOP [37]	-	$\{(8, 1)\}$	riu2	≈ 0.50
FoSIG [39]	$\{(0.5, 6)\}$	$\{(8, 1)\}$	riu2	≈ 0.37
V-BIG [27]	$\{(0.5, 6)\}$	$\{(8, 1)\}$	riu2	≈ 0.35
Our RUBIG	$\{(0.5, 6)\}$	$\{(8, 1)\}$	riu2	≈ 0.56

Note: "-" means "not available".

of CPU Core i7 3.4GHz 16G RAM.

4 Experiments

In this section, our proposal is judged for DT classification task on various benchmark datasets (UCLA, DynTex, and DynTex++). We address a linear multi-class SVM algorithm of LIBLINEAR library [41] with the default settings for classifying DTs in comparison with state-of-the-art results.

4.1 Experimental settings

The 3-dimensional Gaussian-based kernels are exploited to capture volumes of blurred-invariant features, where the kernel width of each axis is traditionally truncated to $[-3\sigma, 3\sigma]$ (σ is the standard deviation of Gaussian distribution) for optimally capturing the energy of Gaussian distribution. We then consider a set of couples of standard deviations $\Lambda = \{(\sigma_i, \sigma_i')\}_{i=1}^{m} = \{(0.5, 6), (0.75, 5), (1, 4)\}$ (i.e., $m = 3$) in order to compute DoG together with Gaussian-filtered outcomes. In brief, for each couple (σ_i, σ_i'), two following outcomes $\mathcal{V}_{\sigma_i}^{G}$, and $\mathcal{V}_{\sigma_i, \sigma_i'}^{DoG}$ are produced and then are encoded by our LRP operator in the next step. It should be noted that the large scale ratios between two scales of each couple of standard deviations are taken into account. Our idea is to highlight the invariant features of DoG outcome extracted from two different scales of Gaussian filtering. Empirically, the more two standard deviations are different, the more DoG outcome contains rich, discriminative, and robust features for LRP operator. Therefore, this concept justifies the large scale ratios of standard deviations between two scales in our model. For DT representation, LRP features are extracted from the filtered volumes by utilizing parameters of $riu2$ mapping, $\{(P, R)\} = \{(8, 1)\}$

for single-scale relationships, and $\{(8,1),(8,2)\}$ for multi-scale in further local regions. The achieved components are integrated in two investigations of $\Omega = \{_{D\_M/C},\ _{D/M/C}\}$ to form corresponding RUBIG descriptors with dimensions of 540 and 3600 bins respectively.

4.2 Datasets and protocols

In this section, benchmark datasets and corresponding protocols for experiments of DT classification are detailed. A brief of those is then indicated in Table 2 for at a glance.

UCLA dataset includes 200 videos with dimension of 110 × 160 × 75, which are categorized into 50 groups of DT sequences [5]. The first line in Fig. 4 shows some samples of those. The following sub-datasets are usually addressed and rearranged for evaluations of DT classification. *50-class breakdown* uses two following protocols to recognize DTs on the original 50 groups: *leave-one-out* (LOO) [3, 42] and *4-fold cross validation* [40, 16]. Two more challenging breakdowns are constructed by composing from the original 50 categories. *9-class* includes "*plants*" (108), "*sea*" (12), "*fire*" (8), "*flowers*" (12), "*fountains*" (20), "*smoke*" (4), "*water*" (12), "*boiling water*" (8), and "*waterfall*" (16), where the numbers of videos are indicated in parentheses. Due to the dominant quantity of the *"plants"*, it is removed to form *8-class* with more challenging [43]. The protocol for two schemes is similar to that in [44, 40], where a half of samples in each category is randomly picked out for testing and the rest for training. The final rates are reported from 20 runtimes.

DynTex dataset consists of over 650 high-resolution DT videos in AVI format which are captured in various circumstances of environment [45]. Following settings in [46, 47], the version of sequences with 352 × 288 × 250 dimension is addressed for our evaluations of DT classification using the LOO protocol (see the second line in Fig. 4 for some samples). There are 4 challenging subsets rearranged from the original as follows. *DynTex35* includes 10 classes formed by splitting from 35 videos as follows. Eight non-overlapping sub-videos are obtained by randomly clipping each video at separating points along X, Y, and T axes, but not at the half of them. Two more are resulted using another cutting operation according to its T axis [20, 24]. Three remaining challenges are composed as follows. *Alpha* consists of 60 DTs divided into three groups: "*trees*", "*sea*", and "*grass*" with 20 samples for each. *Beta* contains 162 videos in 10 labels with various numbers of samples for each of them. Lastly, *Gamma* includes 10 classes of 264 DT videos with different quantities.

DynTex++ dataset is constructed from 345 raw DynTex videos, in which each of them is pre-processed to capture major chaotic DTs and settled in 50 × 50 × 50 dimension [44]. The obtained DTs (3600 sequences) are then divided into 36 classes with 100 samples for each of them. For

Table 2: A brief review of DT datasets' properties.

Dataset	Sub-dataset	#Videos	Resolution	#Classes	Protocol
UCLA	50-class	200	110 × 160 × 75	50	LOO and 4fold
	9-class	200	110 × 160 × 75	9	50%/50%
	8-class	92	110 × 160 × 75	8	50%/50%
DynTex	DynTex35	350	different dimensions	10	LOO
	Alpha	60	352 × 288 × 250	3	LOO
	Beta	162	352 × 288 × 250	10	LOO
	Gamma	264	352 × 288 × 250	10	LOO
DynTex++		3600	50 × 50 × 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.

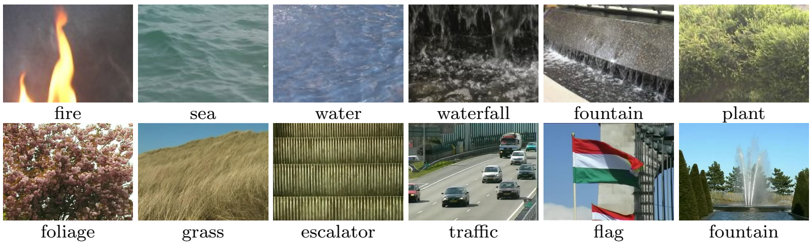

Figure 4: Sample videos of UCLA (above row) and DynTex (bottom row).

evaluations, a half of items is randomly selected from each class for testing and the rest for training. The final rate is reported from the average of 10 repetitions [3].

4.3 Experimental results

Specific experimental results of our descriptor RUBIG on benchmark datasets are shown in Table 4 with the highest rates in bold. It should be noted that only results of the setting of $_{D/M/C}$ are reported due to its high performance. As expected, it can be verified from Tables 3, 4 that RUBIG outperforms compared to those of FoSIG and V-BIG thanks to the crucial contributions of the proposed operator LRP utilized for capturing rich spatio-temporal patterns in the Gaussian-based filtered volumes. The experiments have also validated that RUBIG's performance becomes more "stable" in consideration of various scales of Gaussian-based kernels (see Table 4). In general, our framework performs very well in comparison with the state-of-the-art approaches, including deep-learning-based methods in several circumstances (see Tables 5, 6). Due to these recognition rates on most of DT datasets, the settings of $_{D/M/C}$ and $\{(0.5,6),(0.75,5),(1,4)\}$ for the multi-scale LRP encoding are addressed for comparison (see Table 4). Hereinafter, we detail evaluations of RUBIG's performances.

UCLA dataset: It can be verified from Table 4 that RUBIG obtains very good results on DT recognition. In comparison with the state-of-the-

Table 3: Comparison contributions in rates (%) on DynTex++ between components of descriptors FoSIG, V-BIG and RUBIG.

$(\sigma, \sigma') = (0.5, 6)$	FoSIG$^{riu2}_{8,1}$	V-BIG$^{riu2}_{8,1}$	our RUBIG$^{riu2}_{8,1}$
$G^{2D/3D}_{\sigma}$	95.73	96.01	96.23
$DoG^{2D/3D}_{\sigma,\sigma'}$	93.78	94.43	95.06
$G^{2D/3D}_{\sigma} + DoG^{2D/3D}_{\sigma,\sigma'}$	95.99	96.59	96.68

Table 4: Classification rates (%) on benchmark datasets.

Dataset	UCLA				DynTex				Dyn++
$\{(\sigma_i, \sigma'_i)\}, \{(8,1)\}$	50-LOO	50-4fold	9-class	8-class	Dyn35	Alpha	Beta	Gamma	
$\{(0.5, 6)\}$	**100**	**100**	98.25	98.04	98.57	**100**	92.59	93.18	96.68
$\{(0.75, 5)\}$	**100**	**100**	99.15	98.48	98.00	**100**	92.59	92.42	96.22
$\{(1, 4)\}$	**100**	**100**	98.60	98.80	98.29	**100**	93.83	92.80	95.94
$\{(0.5, 6), (0.75, 5)\}$	**100**	**100**	98.65	98.26	97.71	**100**	93.83	93.18	96.48
$\{(0.75, 5), (1, 4)\}$	**100**	**100**	98.15	99.13	98.86	**100**	93.21	93.18	96.66
$\{(0.5, 6), (0.75, 5), (1, 4)\}$	**100**	**100**	98.50	97.07	97.43	**100**	94.44	93.18	96.79
$\{(\sigma_i, \sigma'_i)\}, \{(8,1), (8,2)\}$									
$\{(0.5, 6)\}$	**100**	**100**	98.90	99.13	99.14	**100**	93.83	**93.56**	96.76
$\{(0.75, 5)\}$	**100**	**100**	99.05	98.80	**99.43**	**100**	94.44	93.18	96.64
$\{(1, 4)\}$	**100**	**100**	98.95	98.37	98.57	**100**	94.44	**93.56**	96.12
$\{(0.5, 6), (0.75, 5)\}$	**100**	**100**	98.95	**99.24**	**99.43**	**100**	94.44	93.18	96.92
$\{(0.75, 5), (1, 4)\}$	**100**	**100**	98.20	99.13	98.57	**100**	94.44	**93.56**	96.54
$\{(0.5, 6), (0.75, 5), (1, 4)\}$	**100**	**100**	**99.20**	99.13	98.86	**100**	**95.68**	**93.56**	**97.08**

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 shortened for DynTex35 and DynTex++ sub-datasets respectively.

are approaches, our descriptor gains the best performance in schemes of 50 categories, 100% for both *50-LOO* and *50-4fold*. In the meanwhile, the deep-learning methods are just at 99.5% for PCANet-TOP [46] and DT-CNN [16] (see Table 5). In terms of two remaining breakdowns, with the settings for comparison, RUBIG achieves rates of 99.2% for *9-class* and 99.13% for *8-class*, the highest compared to the LBP-based methods, including two methods FoSIG and V-BIG utilizing 2D/3D Gaussian kernels against illumination and noise. It should be noted that ours are also better than those of DT-CNN on these schemes. In the meanwhile, STRF N-jet [33] has nearly the same our performances; FD-MAP [2] and DNGP [4] obtain a little higher than ours but not on 50 categories. More evaluations of confusion matrices and F-measure on these schemes are detailed in a supplementary material of this work.

DynTex dataset: The proposed descriptor achieves the highest rate of 99.43% using the settings of Gaussian-based kernels $\{(0.75, 5)\}$ and $\{(0.5, 6), (0.75, 5)\}$ along with multi-local-region relationships $\{(8, 1), (8, 2)\}$ (see Table 4). With the settings for comparison, it just gains 98.86%, a little lower than V-BIG (99.43%), FoSIG (99.14%), MEWLSP [48] (99.71%) but those methods are either not validated (MEWLSP) or not better than ours in other datasets (V-BIG, FoSIG) (see Tables 5, 6). In respect of DT classification on *Alpha*,

Table 5: Comparison of recognition rates (%) on UCLA.

Group	Encoding method	50-LOO	50-4fold	9-class	8-class
A	FDT [2]	98.50	99.00	97.70	99.35
	FD-MAP [2]	99.50	99.00	99.35	**99.57**
B	AR-LDS [5]	89.90^{N}	-	-	-
	Chaotic vector [7]	-	-	85.10^{N}	85.00^{N}
C	3D-OTF [11]	-	87.10	97.23	99.50
	DFS [43]	-	**100**	97.50	99.20
	STLS [13]	-	99.50	97.40	99.50
D	MBSIF-TOP [3]	99.50^{N}	-	-	-
	DNGP [4]	-	-	**99.60**	99.40
	STRF N-jet [33]	-	**100**	99.20	99.00
E	VLBP [20]	-	89.50^{N}	96.30^{N}	91.96^{N}
	LBP-TOP [20]	-	94.50^{N}	96.00^{N}	93.67^{N}
	CVLBP [23]	-	93.00^{N}	96.90^{N}	95.65^{N}
	HLBP [24]	95.00^{N}	95.00^{N}	98.35^{N}	97.50^{N}
	CLSP-TOP [40]	99.00^{N}	99.00^{N}	98.60^{N}	97.72^{N}
	MEWLSP [48]	96.50^{N}	96.50^{N}	98.55^{N}	98.04^{N}
	WLBPC [42]	-	96.50^{N}	97.17^{N}	97.61^{N}
	CVLBC [36]	98.50^{N}	99.00^{N}	99.20^{N}	99.02^{N}
	CSAP-TOP [37]	99.50	99.50	96.80	95.98
	FoSIG [39]	99.50	**100**	98.95	98.59
	V-BIG [27]	99.50	99.50	97.95	97.50
	Our RUBIG$^{riu2}_{\{(8,1),(8,2)\}}\{(0.5,6),(0.75,5),(1,4)\}$	**100**	**100**	99.20	99.13
F	DL-PEGASOS [44]	-	97.50	95.60	-
	PI-LBP+super hist [49]	-	**100^{N}**	98.20^{N}	-
	Orthogonal Tensor DL [18]	-	99.80	98.20	99.50
	PCANet-TOP [46]	99.50*	-	-	-
	DT-CNN-AlexNet [16]	-	99.50*	98.05*	98.48*
	DT-CNN-GoogleNet [16]	-	99.50*	98.35*	99.02*

Note: "-" means "not available". "*" 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.*

Beta, and *Gamma*, it can be seen from Table 6 that RUBIG outperforms the same as V-BIG, DT-CNN, and D3 [17] with the best rate of 100% on subset *Alpha*. It also obtains significant rates of 95.68% on *Beta*, 93.56% on *Gamma* while only 92.59%, 92.42% for FoSIG, and 95.06%, 94.32% for V-BIG respectively. In comparison with other approaches, STRF N-jet [33] obtains significant results, especially with rate of 95.5% on *Gamma*, about 2% better than ours (see Table 6). In the meanwhile, the deep-learning methods (i.e., DT-CNN and D3) achieve the best performances. However, they take a long time for learning DT features along with complicated algorithms involved with.

DynTex++ dataset: It can be observed from Table 6 that with the comparing settings, RUBIG obtains the highest rate of 97.08% compared to most of the existing methods, including Gaussian-based approaches FoSIG and V-BIG. Ours is the same as filter-based method MBSIF-TOP (97.12%),

Table 6: Comparison of rates (%) on DynTex and DynTex++.

Group	Encoding method	Dyn35	Alpha	Beta	Gamma	Dyn++
A	FDT [2]	98.86	98.33	93.21	91.67	95.31
	FD-MAP [2]	98.86	98.33	92.59	91.67	95.69
C	3D-OTF [11]	96.70	83.61	73.22	72.53	89.17
	DFS [43]	97.16	85.24	76.93	74.82	91.70
	2D+T [47]	-	85.00	67.00	63.00	-
	STLS [13]	98.20	89.40	80.80	79.80	94.50
D	MBSIF-TOP [3]	98.61$^{\text{N}}$	90.00$^{\text{N}}$	90.70$^{\text{N}}$	91.30$^{\text{N}}$	97.12$^{\text{N}}$
	DNGP [4]	-	-	-	-	93.80
	STRF N-jet [33]	-	**100**	95.10	95.50	-
E	VLBP [20]	81.14$^{\text{N}}$	-	-	-	94.98$^{\text{N}}$
	LBP-TOP [20]	92.45$^{\text{N}}$	98.33	88.89	84.85$^{\text{N}}$	94.05$^{\text{N}}$
	DDLBP with MJMI [50]	-	-	-	-	95.80
	CVLBP [23]	85.14$^{\text{N}}$	-	-	-	-
	HLBP [24]	98.57$^{\text{N}}$	-	-	-	96.28$^{\text{N}}$
	CLSP-TOP [40]	98.29$^{\text{N}}$	95.00$^{\text{N}}$	91.98$^{\text{N}}$	91.29$^{\text{N}}$	95.50$^{\text{N}}$
	MEWLSP [48]	99.71$^{\text{N}}$	-	-	-	98.48$^{\text{N}}$
	WLBPC [42]	-	-	-	-	95.01$^{\text{N}}$
	CVLBC [36]	98.86$^{\text{N}}$	-	-	-	91.31$^{\text{N}}$
	CSAP-TOP [37]	**100**	96.67	92.59	90.53	-
	FoSIG [39]	99.14	96.67	92.59	92.42	95.99
	V-BIG [27]	99.43	**100**	95.06	94.32	96.65
	Our $\mathbf{RUBIG}^{riu2}_{\{(8,1),(8,2)\}}\{(0.5,6),(0.75,5),(1,4)\}$	98.86	**100**	95.68	93.56	97.08
F	DL-PEGASOS [44]	-	-	-	-	63.70
	PCA-cLBP/PI/PD-LBP [49]	-	-	-	-	92.40
	Orthogonal Tensor DL [18]	-	87.80	76.70	74.80	94.70
	Equiangular Kernel DL [19]	-	88.80	77.40	75.60	93.40
	st-TCoF [15]	-	**100***	**100***	98.11*	-
	PCANet-TOP [46]	-	96.67*	90.74*	89.39*	-
	D3 [17]	-	**100***	**100***	98.11*	-
	DT-CNN-AlexNet [16]	-	**100***	99.38*	**99.62***	98.18*
	DT-CNN-GoogleNet [16]	-	**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*, C: *geometry-based*, D: *filter-based*, E: *local-feature-based*, F: *learning-based.*

and just about 1% lower than DT-CNN with 98.18% for AlexNet framework and 98.58% for GoogleNet. MEWLSP gains 98.48% but as mentioned above, it has not been verified on challenging subsets of DynTex and not better than our performance on UCLA.

4.4 Global discussion

- Empirically, closed values of σ and σ' lead to reduction of performance due to lack of differences of blurred-invariant features. For instance, using $(\sigma, \sigma') = (1.5, 2)$, $\text{RUBIG}^{riu2}_{8,1}$ just obtains 99.5% for 50 categories and about 98% for *9-class* and *8-class* breakdowns in UCLA.
- For DT recognition on simple datasets (e.g., UCLA), the setting of "$_{D\_M/C}$" should be exploited in practice since its performance is nearly

the same that of "$_{D/M/C}$" but in much lower dimension, 540 bins versus 3600 for each video encoding. In fact, using this jointing prototype, RUBIG$^{riu2}_{(8,1)}$ with kernel of $\{(0.5, 6)\}$ obtains rates of 100% for 50 categories, 98.55% for *9-class* and 98.04% for *8-class*.

- For a trade-off between the dimension of descriptor and accuracy rates, RUBIGs with two Gaussian-based filtering scales (e.g., $\{(0.5, 6), (0.75, 5)\}$) may be taken into account real applications due to their reasonable performance on most of the benchmark datasets (see Table 4).

5 Conclusions

An efficient model for DT description has been introduced in this work in which Gaussian-based filtering outcomes, computed from two different scales, are taken into account to extract blurred-invariant features from a DT scene. Those outputs are then addressed by a discriminative LRP operator thanks to a novel concept of supporting region, and an effective model of completed components. Moreover, we have presented a new mechanism of thresholding/encoding to capture rich spatio-temporal relationships from a rubik cube centered at each voxel in order to structure a robust descriptor against the negative impacts of environmental changes. The experiments for DT recognition on the benchmark datasets have validated the interest of our proposal in comparison with the existing approaches.

References

[1] Péteri, R., Chetverikov, D.: Dynamic texture recognition using normal flow and texture regularity. In Marques, J.S., de la Blanca, N.P., Pina, P., eds.: IbPRIA. Volume 3523 of LNCS. (2005) 223–230

[2] Nguyen, T.T., Nguyen, T.P., Bouchara, F., Nguyen, X.S.: Directional beams of dense trajectories for dynamic texture recognition. In Blanc-Talon, J., Helbert, D., Philips, W., Popescu, D., Scheunders, P., eds.: ACIVS. (2018) 74–86

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

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

[5] Saisan, P., Doretto, G., Wu, Y.N., Soatto, S.: Dynamic texture recognition. In: CVPR. (2001) 58–63

[6] Mumtaz, A., Coviello, E., Lanckriet, G.R.G., Chan, A.B.: Clustering dynamic textures with the hierarchical EM algorithm for modeling video. IEEE Trans. PAMI **35**(7) (2013) 1606–1621

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

[8] Ravichandran, A., Chaudhry, R., Vidal, R.: View-invariant dynamic texture recognition using a bag of dynamical systems. In: CVPR. (2009) 1651–1657

[9] Mumtaz, A., Coviello, E., Lanckriet, G.R.G., Chan, A.B.: A scalable and accurate descriptor for dynamic textures using bag of system trees. IEEE Trans. PAMI **37**(4) (2015) 697–712

[10] Xu, Y., Quan, Y., Ling, H., Ji, H.: Dynamic texture classification using dynamic fractal analysis. In: ICCV. (2011) 1219–1226

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

[12] Ji, H., Yang, X., Ling, H., Xu, Y.: Wavelet domain multifractal analysis for static and dynamic texture classification. IEEE Trans. IP **22**(1) (2013) 286–299

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

[14] Baktashmotlagh, M., Harandi, M.T., , A., C. Lovell, B.C., Salzmann, M.: Discriminative non-linear stationary subspace analysis for video classification. IEEE Trans. PAMI **36**(12) (2014) 2353–2366

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

[16] Andrearczyk, V., Whelan, P.F.: Convolutional neural network on three orthogonal planes for dynamic texture classification. PR **76** (2018) 36 – 49

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

[18] Quan, Y., Huang, Y., Ji, H.: Dynamic texture recognition via orthogonal tensor dictionary learning. In: ICCV. (2015) 73–81

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

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

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

[22] Zhao, G., Ahonen, T., Matas, J., Pietikäinen, M.: Rotation-invariant image and video description with local binary pattern features. IEEE Trans. IP **21**(4) (2012) 1465–1477

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

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

[25] Nguyen, T.T., Nguyen, T.P., Bouchara, F.: Dynamic texture representation based on hierarchical local patterns. In Blanc-Talon, J., Delmas, P., Philips, W., Popescu, D., Scheunders, P., eds.: ACIVS. (2020) 277–289

[26] Nguyen, T.T., Nguyen, T.P., Bouchara, F., Nguyen, X.S.: Momental directional patterns for dynamic texture recognition. CVIU **in press** (2020)

[27] Nguyen, T.T., Nguyen, T.P., Bouchara, F., Vu, N.: Volumes of blurred-invariant gaussians for dynamic texture classification. In Vento, M., Percannella, G., eds.: CAIP. (2019) 155–167

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

[29] Nguyen, T.P., Manzanera, A., Kropatsch, W.G., N'Guyen, X.S.: Topological attribute patterns for texture recognition. PRL **80** (2016) 91–97

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

[31] Jain, A.K., Farrokhnia, F.: Unsupervised texture segmentation using gabor filters. Pattern Recognition **24**(12) (1991) 1167–1186

[32] Derpanis, K.G., Wildes, R.P.: Spacetime texture representation and recognition based on a spatiotemporal orientation analysis. IEEE Trans. PAMI **34**(6) (2012) 1193–1205

[33] Jansson, Y., Lindeberg, T.: Dynamic texture recognition using time-causal and time-recursive spatio-temporal receptive fields. Journal of Mathematical Imaging and Vision **60**(9) (2018) 1369–1398

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

[35] Lee, T.C.M., Berman, M.: Nonparametric estimation and simulation of two-dimensional gaussian image textures. CVGIP: Graphical Model and Image Processing **59**(6) (1997) 434–445

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

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

[38] Mäenpää, T., Pietikäinen, M.: Multi-scale binary patterns for texture analysis. In: SCIA. (2003) 885–892

[39] Nguyen, T.T., Nguyen, T.P., Bouchara, F.: Smooth-invariant gaussian features for dynamic texture recognition. In: ICIP. (2019) 4400–4404

[40] Nguyen, T.T., Nguyen, T.P., Bouchara, F.: Completed local structure patterns on three orthogonal planes for dynamic texture recognition. In: IPTA. (2017) 1–6

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

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

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

[44] Ghanem, B., Ahuja, N.: Maximum margin distance learning for dynamic texture recognition. In Daniilidis, K., Maragos, P., Paragios, N., eds.: ECCV. Volume 6312 of LNCS. (2010) 223–236

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

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

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

[48] Tiwari, D., Tyagi, V.: Dynamic texture recognition using multiresolution edge-weighted local structure pattern. Computers & Electrical Engineering **62** (2017) 485–498

[49] Ren, J., Jiang, X., Yuan, J.: Dynamic texture recognition using enhanced LBP features. In: ICASSP. (2013) 2400–2404

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