A novel filtering kernel based on difference of derivative Gaussians with applications to dynamic texture representation

Thanh Tuan Nguyen, Thanh Phuong Nguyen, Frédéric Bouchara

▶ **To cite this version:**

Thanh Tuan Nguyen, Thanh Phuong Nguyen, Frédéric Bouchara. A novel filtering kernel based on difference of derivative Gaussians with applications to dynamic texture representation. Signal Processing: Image Communication, 2021, 98, pp.116394. 10.1016/j.image.2021.116394 . hal-03378945

HAL Id: hal-03378945
https://hal.science/hal-03378945

Submitted on 22 Aug 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Version of Record: https://www.sciencedirect.com/science/article/pii/S0923596521001879
Manuscript_10d65ea2d8633aa03a09068c4cdabe5e

A novel filtering kernel based on difference of derivative Gaussians with applications to dynamic texture representation

Thanh Tuan Nguyen[a,b], Thanh Phuong Nguyen[a,*], Frédéric Bouchara[a]

[a] *Université de Toulon, Aix Marseille Univ, CNRS, LIS, Marseille, France*
[b] *HCMC University of Technology and Education, Faculty of IT, Thu Duc City, Ho Chi Minh City, Vietnam*

Abstract

Efficiently representing spatio-temporal features of dynamic textures (DTs) in videos has been restricted due to negative impacts of the well-known issues of environmental changes, illumination, and noise. In order to mitigate those, this paper proposes a new approach for an efficient DT representation by addressing the following novel concepts. Firstly, a novel filtering kernel, called Difference of Derivative Gaussians (DoDG), is introduced for the first time based on high-order derivative of a Gaussian kernel. It allows to point out DoDG-based filtered outcomes which are prominently resistant to noise for DT representation compared to exploiting the conventional Difference of Gaussians (DoG). A new framework in low computational complexity is then presented to take DoDG into account video denoising as an effective preprocessing of DT encoding. Finally, a simple variant of Local Binary Patterns (LBPs) is addressed to extract local features from these DoDG-filtered outcomes for constructing discriminative DoDG-based descriptors in small dimension, expected as one of appreciated solutions for mobile applications. Experimental results for DT recognition have verified that our proposal significantly performs well compared to all non-deep-learning methods, while being very close to deep-learning approaches. Also, ours are eminently better than those based on the traditional DoG.

Keywords: Dynamic texture, Feature extraction, Gaussian-based filterings, LBP, CLBP, Video representation

1. Introduction

Dynamic textures (DTs) are textural features repeated in a temporal domain [1]. Efficiently analyzing them is one of crucial missions in applications of computer vision: human interaction [2, 3, 4, 5, 6], tracking motions [7, 8], object detection [9, 10, 11], background subtraction [12, 13, 14, 15], etc. Due to the negative impacts of environmental changes, illumination and noise, describing their chaotic motions is a notable challenge for DT representation. Many efforts have been introduced to deal with those problems, which can be grouped into six main categories: *model-based*, *geometry-based*, *optical-flow-based*, *learning-based*, *local-feature-based*, and *filter-based* (refer to Section 2.3 for the literature in detail). Among of them, the filter-based approaches, taking filters into account video analysis for noise reduction, have recently obtained promising results in reasonable dimension, expected to be potential for mobile applications needing restricted resources to execute functions. Concretely, 2D/3D Gaussian-based filterings were addressed for video analyses to figure out its filtered images/volumes. The filtered responses were then encoded by a simple operator CLBP [16] to construct descriptors named FoSIG^{2D} [17]/V-BIG^{3D} [18] correspondingly, while the filtered volumes of the 3D Gaussian-based filterings were encoded

*Corresponding author
Email addresses: tuannt@hcmute.edu.vn (Thanh Tuan Nguyen), tpnguyen@univ-tln.fr (Thanh Phuong Nguyen), bouchara@univ-tln.fr (Frédéric Bouchara)

© 2021 published by Elsevier. This manuscript is made available under the CC BY NC user license
https://creativecommons.org/licenses/by-nc/4.0/

by a Local Rubik Pattern (LRP) operator to form another descriptor in more discriminative power [19]. In another effort, the gradients of these Gaussian-based filterings were also introduced in [20] to enhance the performance. Moreover, learned filters have been addressed to construct DT descriptors, e.g., BSIF-TOP [23], BSDF [24], and B3DF_SMC [25]. Experiments in DT recognition have shown that these learned-filter-based approaches in DT encoding have usually obtained results at moderate levels compared to those addressing non-learned filters, e.g., Gaussian-based kernels [20, 19]. In the meanwhile, some of the learned filters need more computational cost for the learning processes as well as the obtained descriptors are in big dimension, e.g., up to 2^{15} bins for B3DF_SMC [25].

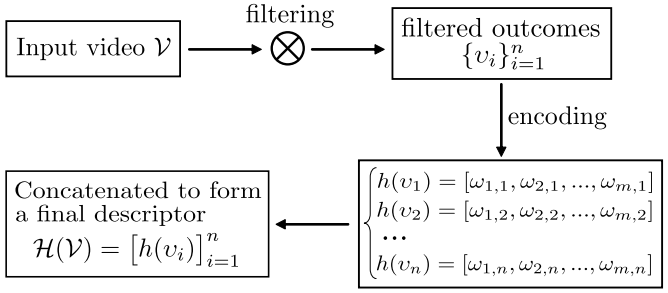


Figure 1: A general framework of encoding a video based on filtering.

Based on the encoding processes of above filter-based methods, it can be deduced a general diagram for analysis of a given video as shown in Figure 1. Accordingly, it can be verified that the filtering plays an important role in the whole framework. Due to mostly addressing filters as pre-processing, it should be done in rapid time while pointing out filtered outcomes as robust to noise for local DT encoding as possible. To this end, motivated by the concept of the well-known DoG, we propose in this work a novel DoDG filtering kernel formed subject to difference of high-order partial derivatives of a Gaussian kernel to point out robust and discriminative features at various levels. DoDG is then taken into account video analysis as an efficient preprocessing for denoising. Finally, the obtained DoDG-filtered outcomes are encoded by CLBP [16], one of the most popular and simple local operators, in order to structure robust descriptors with a slight dimension. It should be emphasized that in our prior work [20], we directly exploited the Gaussian gradients in multi-scales of standard deviations to filter a video. Contrary to that, we simply address the difference of two scales of them to form the novel DoDG kernel, thereby allowing to reduce about two thirds dimensions of DT representation (refer to Tables 2 and 8) while inheriting and enhancing robust characteristics (refer to Tables 4 and 8). This formulation is the same as forming the conventional DoG kernel but DoDG is much better in the denoising treatment (refer to Table 4). Experimental results have validated the interest of our proposal. Generally, it can be listed our significant contributions as

- A novel DoDG kernel based on difference of high-order Gaussian-gradients is introduced to efficiently deal with the negative impacts of the well-known issues on DT representation.
- A comprehensive investigation has been made to evaluate the prominent effectiveness of DoDG filterings in local DT encoding compared to that of the conventional DoG one.
- DoDG is considered in multi-order analysis to exploit more high-order DoDG-filtered features for further improvement of discrimination power. Moreover, addressing the odd and even orders is carefully analyzed and recommended thanks to their effectiveness.
- An efficient framework is introduced to take the DoDG kernel into account video analysis. Robust DoDG-based descriptors are shallowly structured by addressing a simple operator on the obtained DoDG-filtered outcomes.
- Having a small dimension, our DoDG-based descriptors have very good performance compared to all non-deep-learning models, while being close to that of the deep-learning approaches.

2. Related works

2.1. A brief review of LBP and CLBP

For describing an image $\mathcal{I}$, Ojala *et al.* [26] introduced a LBP pattern as a binary string by measuring differences of intensities between a pixel $\mathbf{q} \in \mathcal{I}$ and its local neighbors as

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

in which $\{\mathbf{p}_i\}_{i=1}^{P}$ ($P \in \mathbb{Z}^+$) is a set of P neighbors that are interpolated by a circle sample at center $\mathbf{q}$ with radius R, $\mathcal{I}(.)$ returns the gray-level of a pixel, and $s(.)$ is defined as: $s(x) = 1, \text{if } x \geq 0$, and $s(x) = 0$ in otherwise.

Accordingly, it takes 2^P bins for describing a textual image. In reality, the following mappings are often addressed to deal with this burden of dimension: $u2$ for uniform patterns, $riu2$ for rotation-invariant $u2$ patterns, $TAP^{\mathcal{A}}$ mapping [27] for topological patterns, LBC [28] - an alternative of $riu2$ patterns.

In order to address LBP in diverse encoding, Guo *et al.* [16] proposed its completed model (CLBP) in which CLBP's properties are different integrating ways of three components: CLBP_S is identical to the typical LBPs, CLBP_M for magnitude patterns, and CLBP_C for global gray-level differences of center pixels. Among of them, the integration of 3D joint (i.e., $\mathrm{CLBP}_{S/M/C}$) is recommended owing to its discriminative power (refer to [16] for the specific formulas of CLBP's components as well as evaluations of their different combinations).

2.2. Gaussian filtering kernel and its derivatives

A well-known Gaussian filtering is a convolving function of a n-dimensional Gaussian kernel subject to a spacial domain $\gamma_n = \{x_i\}_{i=1}^{n}$ so that its outcomes are in accordance with the Gaussian distribution. In general, the kernel is defined as

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

where $\sigma \in \mathbb{R}^+$ means a pre-defined standard deviation. Accordingly, a k-order partial derivative of $\mathrm{G}_{\sigma}^{n}(\gamma_n)$ with respect to a direction $x_i \in \gamma_n$ is formed as

$$\mathrm{G}_{\sigma,\partial x_i^k}^{n}(\gamma_n) = \frac{\partial^k \mathrm{G}_{\sigma}^{n}(\gamma_n)}{\partial x_i^k} \tag{3}$$

in which "∂" denotes an operation of partial derivatives.

2.3. Literature of DT representation

Efficiently representing DTs plays an important role in real applications of computer vision. Many approaches have been introduced and can be resumed in six main groups as follows.

Model-based methods: Most of model-based methods have taken advantage of the concept of Linear Dynamical System (LDS), introduced by Saisan *et al.* [29], in order to model turbulent movements of DTs. Chan *et al.* [30] introduced a kernel-PCA (Principal Component Analysis) to adapt the observation component of LDS so that it could be in accordance with issues of analyzing DTs in more complex contexts: chaotic properties of motions, moving camera, etc. Also motivated by the LDS's concept, Mumtaz *et al.* [31] proposed DT mixture (DTM) model to cluster DT features based on their similarities that were estimated by Hierarchical Expectation-Maximization (HEM) algorithm. In another aspect of LDS's leverage, Wang *et al.* [32] adapted LDS to be agreed with bag-of-words (BoW) to capture turbulent characteristics of DTs in videos, while Ravichandran *et al.* [33] took into account bag-of-systems (BoS) to attempt a spatio-temporal concern in DT representation. In addition, several model-based approaches have relied on Hidden Markov Model (HMM) to model DT motions. Qiao *et al.* [34] addressed HMM to encode spatial information of DTs in consideration of their appearance along time of a sequence. After that, Qiao *et al.* [35] proposed a model of multivariate HMM to investigate the dependence of the spatial adjacent pixels, which has been lacking in the former work [34]. Regarding the effectiveness in representing DTs, the

model-based approaches have been at moderate levels on DT recognition due to a lack of temporal properties of DTs taking into account their modeling processes [29]. Furthermore, in case of addressing the above issue, the modelings can become more complicated [33].

Geometry-based methods: In order to represent appearance information of DTs, geometry-based methods have mostly based on fractal analyses to mitigate negative impacts of environmental changes on understanding sequences. Accordingly, Xu *et al.* [36, 37] introduced a typical Dynamic Fractal Spectrum (DFS) and its crucial extension, called Multi-Fractal Spectrum (MFS), in which analyses of stochastic self-similarities and fractal patterns were addressed for DT representation. However, they mainly focused on spectral information rather than spatial domain. Ji *et al.* [38] then located this problem by taking spatial information into account MFS with wavelet coefficients in order to construct Wavelet-based MFS (WMFS) for efficiently representing DTs. Lately, Quan *et al.* [39] proposed a technique of Spatio-Temporal Lacunarity Spectrum (STLS) to capture lacunarity-based features based on lacunarity analysis for local binary patterns in DT slices. In another aspect of DT analysis, Baktashmotlagh *et al.* [40] utilized Stationary Subspace Analysis (SSA) in order to investigate stationary components for video description. In terms of effectiveness in DT recognition, experiments have shown that the geometry-based methods principally have good performances on simple datasets, e.g., UCLA [29], but not on the more challenging ones, e.g., DynTex [41] and DynTex++ [42]. It may be due to lack of temporal information involved in their encodings.

Optical-flow-based methods: Most of optical-flow-based methods have represented DTs based on magnitudes and directions of normal flow. Peh *et al.* [43] proposed to shape and trace paths of DT motions in a video. In the meanwhile, on one side, Péteri *et al.* [44, 45] took advantage of normal vector field and criteria of sequences in order to extract DT features. On the other side, they combined the normal flow and filtering regularity for the feature extraction. Lu *et al.* [46] attempted to exploit the beneficial characteristics of the velocity and acceleration to structure spatio-temporal multi-resolution probability distribution. Lately, Nguyen *et al.* [47, 48] encoded a DT video by addressing motion points subject to their trajectories and local motive neighbors. With regard to effectiveness in DT recognition, the optical-flow-based methods have been at moderate abilities due to assumption of brightness constancy and local smoothness as mentioned by Rivera *et al.* [49]. In addition, their moderation can be caused by the less regard of textural appearances, one of important clues for DT understanding.

Learning-based methods: In general, there are two trends of learning-based methods for DT representation as follows. The first one is based on deep learning techniques. Qi *et al.* [50] proposed Transferred ConvNet Features (TCoF), which were deep spatio-temporal structures learned from an implementation of Convolutional Neural Network (CNN) (i.e., AlexNet [51]) for the frames of a video. In the meanwhile, Andrearczyk *et al.* [52] also implemented AlexNet [51] and GoogleNet [53] frameworks to learned DT features on the three orthogonal planes of a given video. Also addressing the video's planes, Arashloo *et al.* [54] proposed a combination of a multi-layer convolutional model and PCA's function to learned filters. In other works, Hong *et al.* [55] introduced a learning concept of "key frames" and "key segments" to construct a deep dual descriptor based on static and dynamic learned features. Hadji *et al.* [56] composed a new challenging large scale dataset (DTDB). They then implemented some deep-learning methods for learning DTs on DTDB: Convolutional 3D (C3D) [57], RGB/Flow Stream [58], Marginalized Spatio-temporal Oriented Energy (MSOE) in two learning streams (MSOE-two-Stream) [56]. The second trend concerns with dictionary learning approaches. Quan *et al.* [59] considered patches of a given video as atoms fed into a sparse coding method to learn a dictionary for DT

representation, while Quan *et al.* [60] proposed equiangular kernel to learn a dictionary in reasonable dimension. In regard to efficiency of the learning-based methods in DT recognition, just the deep models have outperformed the others. However, most of them utilized complex learning algorithms to learn tremendous parameters in deep network architectures. For instance, it takes ∼61M for AlexNet and ∼6.8M for GoogleNet learned in the deep model of [52], while ∼80M learned parameters are for C3D [57], ∼88M by MSOE-two-Stream [56]. Recently, many efforts for object detection have attempted to propose deep-learning models with less resource requirements: MobileNets [61, 62], CenterNet [63]. They can be potential alternatives for DT description in further contexts.

Local-feature-based methods: Taking advantage of simple computation of Local Binary Pattern (LBP) [26] and its variants, many efforts have been made and achieved noteworthy performances in DT recognition. For encoding a given video, Zhao *et al.* [64] introduced VLBP patterns in consideration of a voxel and its local neighbors that are interpolated by addressing three consecutive frames in a given video. Because of this encoding, it is up to 2^{3P+2} bins for DT representation, where P denotes a number of the concerned neighbors. This leads to remarkable barriers for real applications due to the curse of dimension. In order to mitigate that drawback, Zhao *et al.* [64] proposed LBP-TOP patterns in consideration of a voxel and its P neighbors sampled on each of its three orthogonal plane-images in a given video. In addition, it is also possible to apply popular mappings (e.g., $riu2$ mapping) on each plane-image to drastically reduce the dimension. Motivated by these fundamental concepts, many works have been proposed to address LBP's conventional shortcomings for further improvement of discrimination: rotation-invariant problems [65], sensitivity to noise [17, 66, 67, 19], near-uniform regions [68, 69, 21, 70], etc.

Filter-based methods: Filter-bank approaches, which have been early applied to texture analysis [71], have had promising results in DT recognition by mitigating influence of noise on video representation. Arashloo *et al.* [23, 24] exploited filters, learned by implementing Independent Component Analysis (ICA) transformation, to point out Binarized Statistical Image Features (BSIF-TOP) based on Three Orthogonal Planes of a given video [23], and binarised statistical dynamic features (BSDF) [24]. In the meanwhile, Zhao *et al.* [25] proposed to use Completed Local Binary Pattern (CLBP) [16] for capturing spatio-temporal features from filtered responses computed by learned filters, where these filters were learned by different unsupervised procedures: PCA, ICA, sparse filtering, and k-means clustering.

Recently, Gaussian-based kernels were addressed in the previous works to diminish the well-known problems in local DT encoding. Concretely, the original Gaussian filterings were used to point out original Gaussian filtered responses. After that, CLBP [16] was addressed to capture spatio-temporal features for DT representations of FoSIG [17], V-BIG [18], and RUBIG [19] (see Figure 2(a)). In the meantime, the gradients of Gaussian kernels were exploited in [20] for HoGF descriptors, as a visual instance in Figure 2(b)). Different from them, we propose in this work the novel DoDG kernel for the filterings. As highlighted by the light-blue boxes in the flowchart of Figure 2(c), three main significant points can be taken out as follows: *i)* Our DoDG is based on the difference of two scales of Gaussian gradients. Experiments would verify that the DoDG-based responses have more robustness compared to the original Gaussian kernels, the Gaussian gradients as well as the conventional DoG. *ii)* We also propose and validate the noteworthy contribution of the absolute features of the DoDG-based responses in the performance improvement. *iii)* The obtained DoDG-based descriptors have more discrimination compared to the former onces in the same conditions of local encoding with CLBP operator.

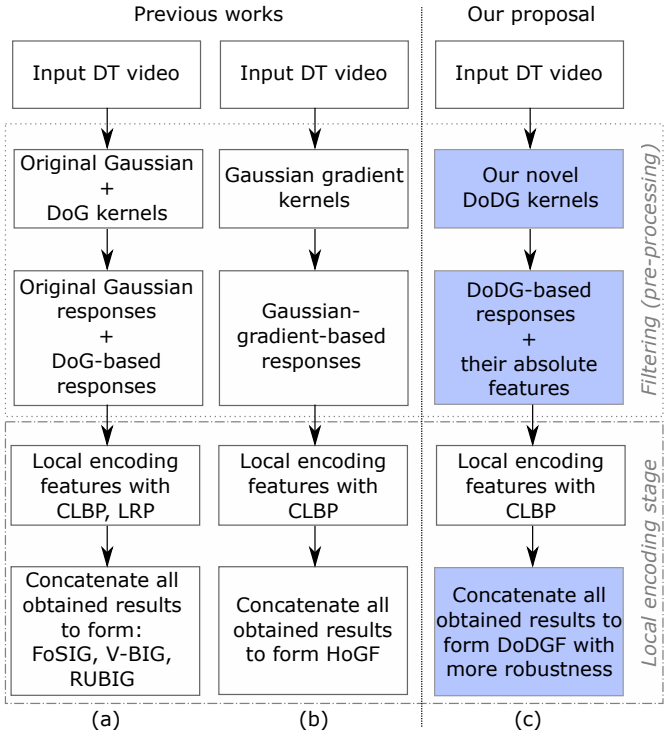


Figure 2: Comparison with previous works based on the Gaussian-based filterings: (a) – FoSIG [17], V-BIG [18], and RUBIG [19]; (b) – HoGF [20]. The distinctive ways come from the light-blue boxes.

3. Proposed method

3.1. An overview

Our proposed framework is graphically illustrated as Figure 5. In general, it takes two major steps to structure a given video $\mathcal{V}$: *i)* a novel filtering for an efficient reduction of the negative impacts of the problems on DT representation; *ii)* a local DT encoding of the obtained filtered-outcomes in simplicity of computation. For the filtering, we introduce a novel DoDG kernel based on the difference of high-order Gaussian-gradients (see Section 3.2). This allows to extract DoDG-filtered outcomes that effectively deal with the well-known issues thanks to robustness of invariant Gaussian-gradient-filtered features in comparison with those done by the conventional DoG. It should be noted that DoG was exploited in local DT encodings: FoSIG [17], V-BIG [18], and RUBIG [19] but its ability is just at a moderate level due to a lack of complementary filtered components involved in those encodings, i.e., only one DoG-filtered outcome was obtained by a DoG filtering operation with each pre-defined pair of standard deviations (see Figure 4 line (a)). Section 4.4 gives more thorough discussions of this significant point. For the local DT encoding, we investigate the effectiveness of our DoDG for the pre-processing step. CLBP [16], a simple operator, can be then addressed for capturing local DoDG-based features from the obtained DoDG-filtered outcomes (see Section 3.4). As a result, robust DoDG-based descriptors are constructed which of their performance in DT recognition is very good compared to recent methods. Hereunder, we detail the above processes.

3.2. A novel DoDG filtering kernel

As mentioned above, DoG, the well-known Gaussian-based filtering kernel, was exploited as a pre-processing step in the former works [17, 18, 19] to reduce the negative impacts of the issues on DT representation. However, its responses are not robust enough for those DT encodings due to the weakness of complementary features. To deal with this shortcoming, we hereafter introduce a novel filtering kernel by forming the difference of high-order Gaussian gradients in simple computation. Experiments in Section 4 have substantiated that its achieved responses efficiently maintain invariant spatial features as well as provide various robust filtered outcomes to forcefully capture rich information for DT description.

Let (σ, σ') denote a pre-defined pair of standard deviations, so that $0 < \sigma < \sigma'$. Based on high-order Gaussian gradients formulated as in Eq. (3), a k-order filtering kernel of DoDG for a direction $x_i \in \gamma_n$, named $\mathrm{DoDG}^n_{\sigma,\sigma',\partial x_i^k}(\gamma_n)$, is defined as the difference of two scales of k-order Gaussian gradients corresponding to (σ, σ') as

$$\mathrm{DoDG}^n_{\sigma,\sigma',\partial x_i^k}(\gamma_n) = \mathrm{G}^n_{\sigma,\partial x_i^k}(\gamma_n) - \mathrm{G}^n_{\sigma',\partial x_i^k}(\gamma_n) \quad (4)$$

Figure 3 at (b) and (c) respectively shows plots of the densities of DoDG^{1D} kernel in the first ($k = 1$) and second ($k = 2$) orders of $(\sigma, \sigma') = (0.7, 1)$. Appreciably, it can be

deduced in general that the proposed DoDG kernels for the spatial domain $\gamma_n = \{x_i\}_{i=1}^n$ as

$$\begin{cases} \mathrm{DoDG}^n_{\sigma,\sigma',\partial x_1^k}(\gamma_n) = \mathrm{G}^n_{\sigma,\partial x_1^k}(\gamma_n) - \mathrm{G}^n_{\sigma',\partial x_1^k}(\gamma_n) \\ \mathrm{DoDG}^n_{\sigma,\sigma',\partial x_2^k}(\gamma_n) = \mathrm{G}^n_{\sigma,\partial x_2^k}(\gamma_n) - \mathrm{G}^n_{\sigma',\partial x_2^k}(\gamma_n) \\ \quad\vdots \qquad\qquad\quad \vdots \qquad\qquad\quad \vdots \\ \mathrm{DoDG}^n_{\sigma,\sigma',\partial x_n^k}(\gamma_n) = \mathrm{G}^n_{\sigma,\partial x_n^k}(\gamma_n) - \mathrm{G}^n_{\sigma',\partial x_n^k}(\gamma_n) \end{cases} \tag{5}$$

As a result, for each k-order, it is possible to obtain n DoDG-filtered outcomes corresponding to n directions that are taken into account a filtering operation.

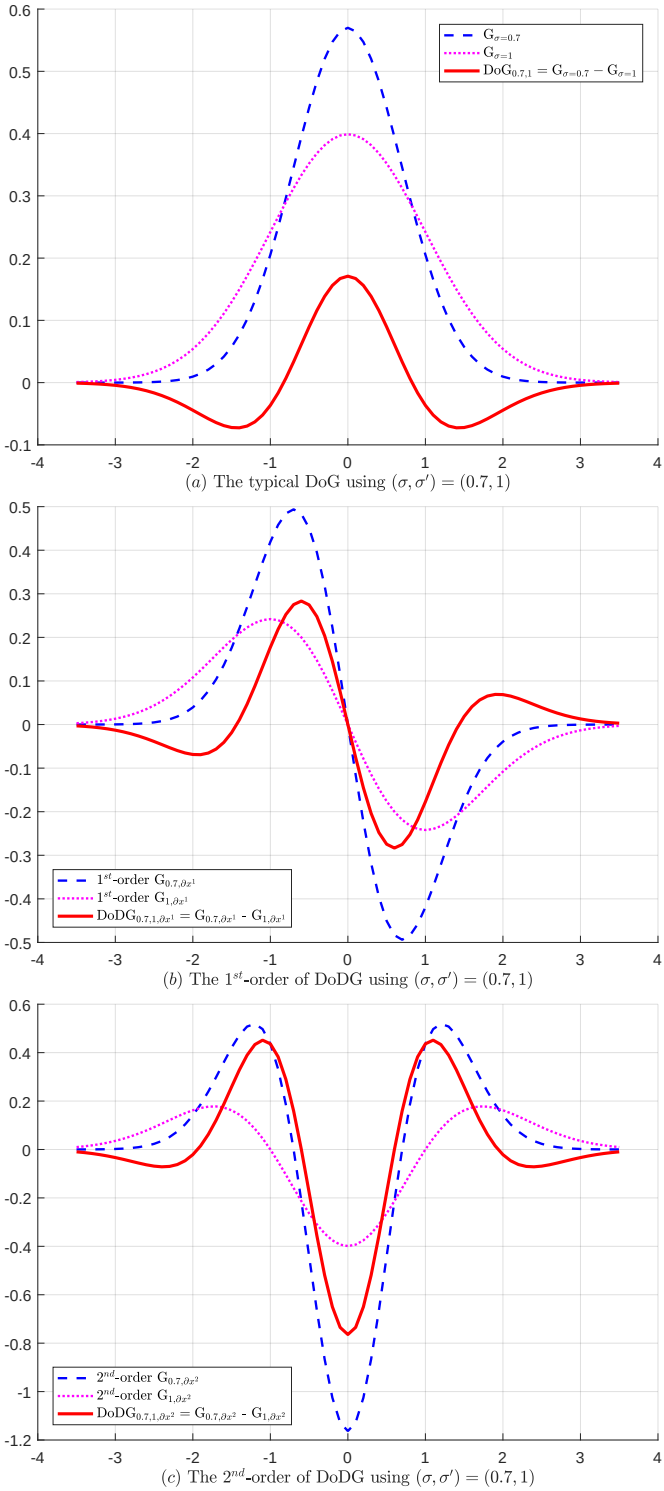
3.3. Beneficial properties of DoDG filtering kernel

Hereafter, we point out some beneficial properties of DoDG for DT representation. For the simplicity of presentation, let us consider k-order DoDG in 1D space, which their profiles are shown in Figure 3.

- When k is odd, DoDG's responses are semi-symmetric since $\mathrm{DoDG}_{\sigma,\sigma'}(x) = -\mathrm{DoDG}_{\sigma,\sigma'}(-x)$ (see Figure 3(b)).
- When k is even, DoDG's responses are symmetric since $\mathrm{DoDG}_{\sigma,\sigma'}(x) = \mathrm{DoDG}_{\sigma,\sigma'}(-x)$ (see Figure 3(c)). Its responses are somewhat similar to that of the DoG kernel (also see Figure 3(a)).
- Also, being a Gaussian-based kernel, DoDG naturally produces robust features against noise.

Accordingly, our DoDG can be structured into two groups: odd and even order kernels. It is evident that two groups are complementary together since they exploit local features in a totally different way. A combination of those allows to take into account both symmetric and asymmetric features, thereby enhancing the informative richness and discrimination power.

On the other hand, since the $\mathrm{G}^n_{\sigma,\partial x_i^k}$ filtering kernel has separable and linear properties, the computational complexity of our $\mathrm{DoDG}^n_{\sigma,\sigma',\partial x_i^k}$ is also inherited from those advantages. Those allow to compute our DoDG[1] in different partial derivatives to forcefully consider DoDG-filtered features in multi-scale analysis of higher orders. Figure 4 in lines (b) and (c) shows DoDG-filtered images obtained



(a) The typical DoG using $(\sigma, \sigma') = (0.7, 1)$

(b) The 1^{st}-order of DoDG using $(\sigma, \sigma') = (0.7, 1)$

(c) The 2^{nd}-order of DoDG using $(\sigma, \sigma') = (0.7, 1)$

Figure 3: Profile of 1D DoG kernel (a) using a pre-defined pair of standard deviations $(\sigma, \sigma') = (0.7, 1)$ in comparison with those of 1D DoDG kernels at the first (b) and second (c) orders.

[1] A simple MATLAB code for 2D/3D DoDG filterings is available at `http://tpnguyen.univ-tln.fr/download/MATCodeDoDG`

by using the DoDG^{2D} filtering kernel with $(\sigma, \sigma') = (0.7, 1)$ in four levels of partial derivatives, i.e., $k \in \{1, 2, 3, 4\}$.

In addition, it is worth noting that the conventional DoG kernel can be also conducted as a degeneration of our DoDG at the zero-order (i.e., $k = 0$). It means that Eq. (4) can be rewritten for the band-pass filter DoG as

$$\mathrm{DoG}^{n}_{\sigma,\sigma'}(\gamma_n) = \mathrm{G}^{n}_{\sigma}(\gamma_n) - \mathrm{G}^{n}_{\sigma'}(\gamma_n) \tag{6}$$

Consequently, it can be stated several crucial points making a better execution of DoDG in noise reduction compared to DoG and Gaussian gradients [17, 18, 20] as

- For filtering processes, each spatial domain in γ_n is often truncated by a scale range of $[-3\sigma, 3\sigma]$ for the convolving operation to optimally capturing the energy of Gaussian distribution. Figure 3 illustrates a graphical view of exploiting both DoG and DoDG to filter an image with a specific pair of standard deviations $(\sigma, \sigma') = (0.7, 1)$. Accordingly, it can be visually realized that our DoDG has figured out less closed-to-zero bipolar features than DoG, those which make the encoding more sensitive to noise as claimed by Vu *et al* [72].
- Our DoDG has extracted more diversity of bipolar filtered-image partitions than DoG (see Figure 3), thereby allowing to capture forceful features for DT representation.
- Also, conducted from Figure 3 (b) and (c), our DoDG could maintain invariant spatial information in better stable frequencies thanks to an adaptive conservation of DoDG's distribution in accordance with that of the concerning Gaussian gradients. In the meanwhile, it is not for DoG since the subtraction of non-Gaussian-gradient filterings is agreed with an approximation of the Laplacian of Gaussian (LoG) (see Figure 3(a)).
- It can be verified from Eqs. (5) and (6) that for a pre-defined pair of (σ, σ') taking into account a filtering process, our DoDG can figure out more complementary filtered outcomes than the only one done by DoG (see Figure 4 for an instance of these filterings). This allows to comprehensively investigate DoDG-filtered features for further enhancement.
- Furthermore, our DoDG can inherit and enhance robust characteristics of their corresponding Gaussian gradients (see Figure 3 (b) and (c)). It can be verified this benefit by experimental instances in Table 8 with standard deviations $(\sigma, \sigma') = (0.7, 1)$).

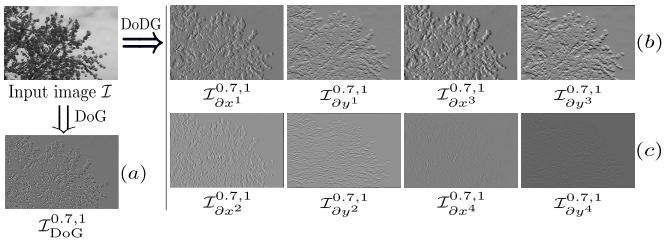


Figure 4: Instances of 2D Gaussian-based filterings for an given image $\mathcal{I}$ using a pre-defined pair of standard deviations $(\sigma, \sigma') = (0.7, 1)$. Therein, (a): a DoG-filtered image of the conventional DoG^{2D} filtering, (b) and (c): DoDG-based images of odd and even DoDG^{2D} filterings respectively.

To validate above advantageous points, both DoG and DoDG are addressed for video analysis as a pre-processing step to handle the well-known issues of DT description (see Section 3.4). After that, the obtained results in DT recognition are thoroughly discussed in Sections 4.4, 4.5, and 4.6.

3.4. DT representation with DoDG-based filterings

To verify the DoDG's ability in dealing with the negative influences on DT description, we take its 2D and 3D variations into account the pre-processing step of encoding a given video $\mathcal{V}$ for noise-resistance (see Figure 5). The DoDG-filtered outputs are then encoded by a simple operator CLBP [16] to correspondingly form robust DT descriptors. Hereafter, we express these processes in detail.

Proposed DoDGF$^{2D}_{\sigma,\sigma',\mathcal{F}}$ ***descriptor:*** To be compliant with the DoDG^{2D} filtering, the video $\mathcal{V}$ is decomposed subject to its orthogonal planes to obtain separate collections of plane-images f_{XY}, f_{XT}, and f_{YT}. With respect to each

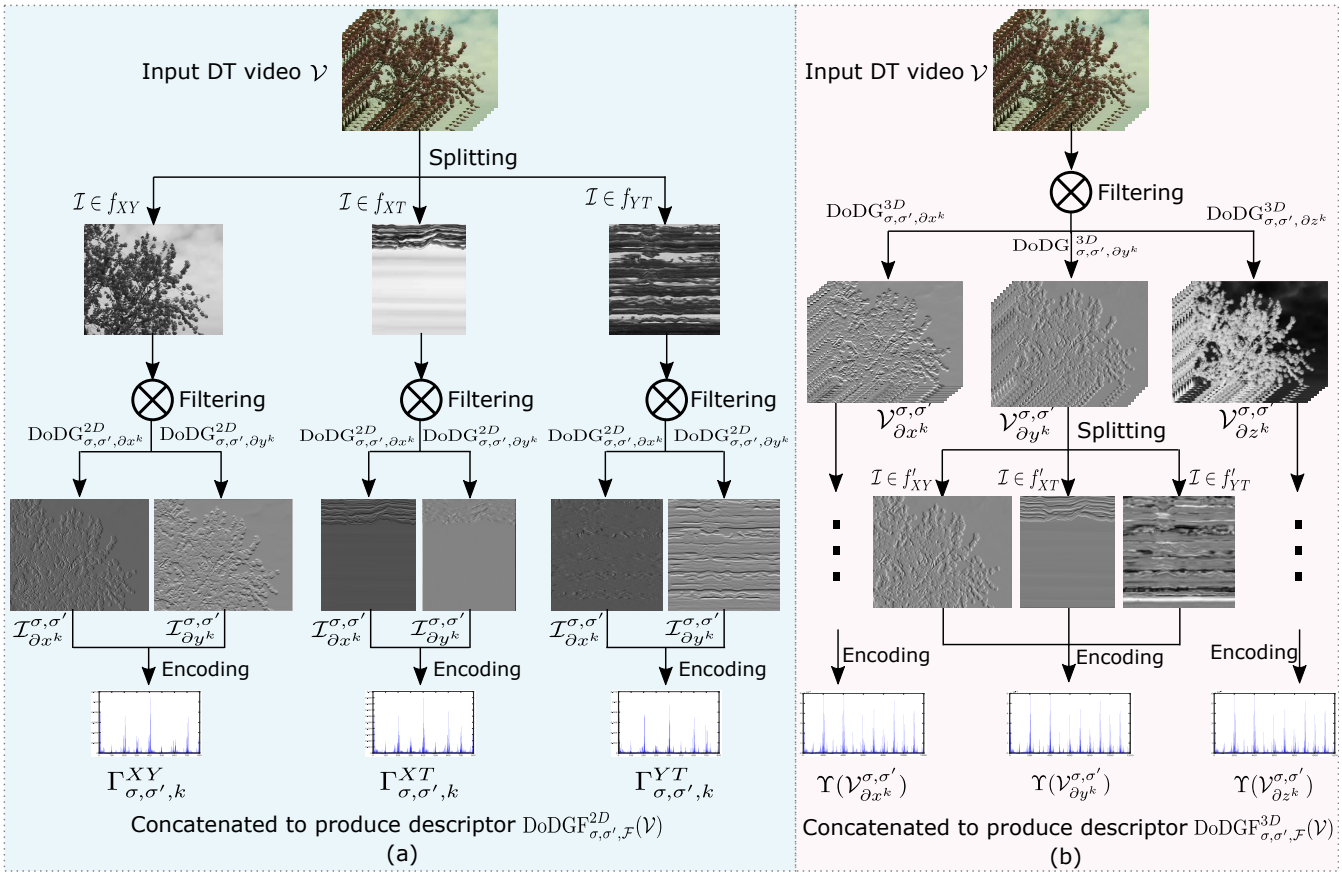


Figure 5: Our proposed framework for encoding a video $\mathcal{V}$ based on its DoDG-filtered outcomes computed by the novel DoDG filterings.

image $\mathcal{I} \in f_{XY}$, a k-order DoDG^{2D} kernel is convolved on it to extract DoDG-filtered images as

$$\begin{cases} \mathcal{I}^{\sigma,\sigma'}_{\partial x^k} = \mathrm{DoDG}^{2D}_{\sigma,\sigma',\partial x^k}(x,y) * \mathcal{I} \\ \mathcal{I}^{\sigma,\sigma'}_{\partial y^k} = \mathrm{DoDG}^{2D}_{\sigma,\sigma',\partial y^k}(x,y) * \mathcal{I} \end{cases} \tag{7}$$

where "*" stands for a convolving operator; x, y are spatial coordinates. Samples of this filtering can be seen in Figure 4: line (b) for the odd gradients and line (c) for the even ones. Since $\mathcal{I}^{\sigma,\sigma'}_{\partial x^k}$ and $\mathcal{I}^{\sigma,\sigma'}_{\partial y^k}$ are bipolar-filtered images, it is possible to consider their absolute outcomes (i.e., $|\mathcal{I}^{\sigma,\sigma'}_{\partial x^k}|$ and $|\mathcal{I}^{\sigma,\sigma'}_{\partial y^k}|$) to explore more textural appearances for further improving discrimination (see Table 6 for their contributions). As a result, all plane-images $\mathcal{I} \in f_{XY}$ are encoded as

$$\Gamma^{XY}_{\sigma,\sigma',k} = \tfrac{1}{\mathcal{N}_{XY}} \sum_{\mathcal{I} \in f_{XY}} \left[\Psi(\mathcal{I}^{\sigma,\sigma'}_{\partial x^k}), \Psi(|\mathcal{I}^{\sigma,\sigma'}_{\partial x^k}|), \Psi(\mathcal{I}^{\sigma,\sigma'}_{\partial y^k}), \Psi(|\mathcal{I}^{\sigma,\sigma'}_{\partial y^k}|) \right] \tag{8}$$

where $\mathcal{N}_{XY}$ denotes a number of plane-images in f_{XY}, $\Psi(.)$ is a simple function using a local operator (e.g., LBP, CLBP, etc.) in order to compute the corresponding histogram. Similarly, this encoding is considered for plane-images f_{XT} and f_{YT} to capture temporal characteristics of DTs. Figure 5(a) shows a visual view of the construction, while Alg. 1 is for the computing structure. Consequently, a robust descriptor based on the high-order 2D DoDG-filtered Features ($\mathrm{DoDGF}^{2D}_{\sigma,\sigma',\mathcal{F}}$) is constructed in simplicity by concatenating these histograms $\Gamma_{\sigma,\sigma',k}$ as

$$\mathrm{DoDGF}^{2D}_{\sigma,\sigma',\mathcal{F}}(\mathcal{V}) = \biguplus_{k \in \mathcal{F}} \left[\Gamma^{XY}_{\sigma,\sigma',k}, \Gamma^{XT}_{\sigma,\sigma',k}, \Gamma^{YT}_{\sigma,\sigma',k} \right] \tag{9}$$

where $\mathcal{F}$ denotes a set of high-orders taking into account the DT encoding; $\biguplus$ stands for incorporation of histograms computed subject to the specific k-orders. For instance, $\mathcal{F} = \{1^{st}, 2^{nd}\}$ means that both first and second gradients of DoDG^{2D} are addressed for multi-order analysis.

Proposed $\mathrm{DoDGF}^{3D}_{\sigma,\sigma',\mathcal{F}}$ ***descriptor:*** The DoDG^{3D} fil-

Algorithm 1: Encoding of DoDGF^{2D} descriptor.

Input: A video $\mathcal{V}$; a set $\mathcal{F}$ of k orders; a pair of standard deviations (σ, σ').

Output: A DoDGF^{2D} descriptor of $\mathcal{V}$.

Split $\mathcal{V}$ into collections of plane-images: $\{f_{XY}, f_{XT}, f_{YT}\}$;

$\Gamma^{XY}_{\sigma,\sigma',k} = \Gamma^{XT}_{\sigma,\sigma',k} = \Gamma^{YT}_{\sigma,\sigma',k}$ = an array of zeros;

for $k \in \mathcal{F}$ **do**

//Compute histograms according to order k

  for $\mathcal{I} \in f_{XY}$ **do**

    //Filtering $\mathcal{I}$ with kernel DoDG^{2D}

    $\mathcal{I}^{\sigma,\sigma'}_{\partial x^k} = \text{DoDG}^{2D}_{\sigma,\sigma',\partial x^k} * \mathcal{I}$;

    $\mathcal{I}^{\sigma,\sigma'}_{\partial y^k} = \text{DoDG}^{2D}_{\sigma,\sigma',\partial y^k} * \mathcal{I}$;

    //Encoding the DoDG^{2D}-based responses

    $\Gamma^{\mathcal{I}}_{\sigma,\sigma',k} = \left[\Psi(\mathcal{I}^{\sigma,\sigma'}_{\partial x^k}), \Psi(|\mathcal{I}^{\sigma,\sigma'}_{\partial x^k}|), \Psi(\mathcal{I}^{\sigma,\sigma'}_{\partial y^k}), \Psi(|\mathcal{I}^{\sigma,\sigma'}_{\partial y^k}|)\right]$;

    $\Gamma^{XY}_{\sigma,\sigma',k} = \Gamma^{XY}_{\sigma,\sigma',k} + \Gamma^{\mathcal{I}}_{\sigma,\sigma',k}$;

  end for

  $\Gamma^{XY}_{\sigma,\sigma',k} = \frac{1}{\mathcal{N}_{XY}} \Gamma^{XY}_{\sigma,\sigma',k}$; *//Normalized*

  Repeat the steps 4 and 5 on f_{XT} and f_{YT} for $\Gamma^{XT}_{\sigma,\sigma',k}$ and $\Gamma^{YT}_{\sigma,\sigma',k}$ respectively.

  $\text{DoDGF}^{2D}_{\sigma,\sigma',k} = \left[\Gamma^{XY}_{\sigma,\sigma',k}, \Gamma^{XT}_{\sigma,\sigma',k}, \Gamma^{YT}_{\sigma,\sigma',k}\right]$;

end for

//Concatenate all the obtained histograms

$\text{DoDGF}^{2D}_{\sigma,\sigma',\mathcal{F}} = \biguplus_{k\in\mathcal{F}} \text{DoDGF}^{2D}_{\sigma,\sigma',k}$;

tering is exploited for pre-processing video $\mathcal{V}$ as

$$\begin{cases} \mathcal{V}^{\sigma,\sigma'}_{\partial x^k} = \text{DoDG}^{3D}_{\sigma,\sigma',\partial x^k}(x,y,z) * \mathcal{V} \\ \mathcal{V}^{\sigma,\sigma'}_{\partial y^k} = \text{DoDG}^{3D}_{\sigma,\sigma',\partial y^k}(x,y,z) * \mathcal{V} \\ \mathcal{V}^{\sigma,\sigma'}_{\partial z^k} = \text{DoDG}^{3D}_{\sigma,\sigma',\partial z^k}(x,y,z) * \mathcal{V} \end{cases} \tag{10}$$

where z denotes the temporal direction of $\mathcal{V}$. To encode the obtained DoDG-filtered volume $\mathcal{V}^{\sigma,\sigma'}_{\partial x^k}$, it is firstly split into collections of filtered plane-images, $\{f'_{XY}, f'_{XT}, f'_{YT}\}$, subject to its three orthogonal planes. The simple operator $\Psi(.)$ is then taken into account encoding these collections to efficiently capture spatio-temporal features as

$$\Upsilon(\mathcal{V}^{\sigma,\sigma'}_{\partial x^k}) = \left[\Psi(\mathcal{I} \in f'_{XY}), \Psi(\mathcal{I} \in f'_{XT}), \Psi(\mathcal{I} \in f'_{YT})\right] \tag{11}$$

Similarly, this encoding is applied to DoDG-filtered volumes $\mathcal{V}^{\sigma,\sigma'}_{\partial y^k}$ and $\mathcal{V}^{\sigma,\sigma'}_{\partial z^k}$ to correspondingly construct histograms of $\Upsilon(\mathcal{V}^{\sigma,\sigma'}_{\partial y^k})$ and $\Upsilon(\mathcal{V}^{\sigma,\sigma'}_{\partial z^k})$. Because these DoDG-filtered outcomes are also bipolar-filtered volumes, it can be possible to consider their absolute volumes (i.e., $|\mathcal{V}^{\sigma,\sigma'}_{\partial x^k}|$, $|\mathcal{V}^{\sigma,\sigma'}_{\partial y^k}|$, and $|\mathcal{V}^{\sigma,\sigma'}_{\partial z^k}|$) to investigate more spatio-temporal features for further enhancement of discrimination power. Figure 5(b) shows a visual view of the construction, while Alg. 2 is for the computing structure. Finally, the obtained histograms are normalized and concatenated to form a local robust descriptor of the high-order 3D DoDG-filtered Features (DoDGF$^{3D}_{\sigma,\sigma',\mathcal{F}}$) as

$$\begin{aligned} \text{DoDGF}^{3D}_{\sigma,\sigma',\mathcal{F}}(\mathcal{V}) = \biguplus_{k\in\mathcal{F}} \Big[&\Upsilon(\mathcal{V}^{\sigma,\sigma'}_{\partial x^k}), \Upsilon(\mathcal{V}^{\sigma,\sigma'}_{\partial y^k}), \Upsilon(\mathcal{V}^{\sigma,\sigma'}_{\partial z^k}), \\ &\Upsilon(|\mathcal{V}^{\sigma,\sigma'}_{\partial x^k}|), \Upsilon(|\mathcal{V}^{\sigma,\sigma'}_{\partial y^k}|), \Upsilon(|\mathcal{V}^{\sigma,\sigma'}_{\partial z^k}|)\Big] \end{aligned} \tag{12}$$

where $\mathcal{F}$ denotes a set of high-orders taking into account the DT encoding; $\biguplus$ stands for incorporation of histograms computed subject to the specific k-orders of $\mathcal{F}$. For instance, $\mathcal{F} = \{1^{st}, 2^{nd}\}$ means that both first and second partial derivatives of DoDG^{3D} are addressed for analysis of multi-orders.

DoG-based descriptors for assessment: To verify the interest of our DoDG in local DT description compared to the well-known DoG kernel, we also implement local DoG-based descriptors based on the corresponding DoG filterings for comprehensive evaluations in Sections 4.4 and 4.6. Accordingly, the 2D and 3D DoG kernels are addressed for the filtering of video $\mathcal{V}$ as

$$\begin{aligned} \mathcal{I}^{\sigma,\sigma'}_{DoG} &= \text{DoG}^{2D}_{\sigma,\sigma'}(x,y) * \mathcal{I} \\ \mathcal{V}^{\sigma,\sigma'}_{DoG} &= \text{DoG}^{3D}_{\sigma,\sigma'}(x,y,z) * \mathcal{V} \end{aligned} \tag{13}$$

Following the construction of the DoDGF^{2D} descriptor, the 2D DoG-filtered features (DoGF$^{2D}_{\sigma,\sigma'}$) are structured as

$$\text{DoGF}^{2D}_{\sigma,\sigma'}(\mathcal{V}) = \left[\Lambda^{XY}_{\sigma,\sigma'}, \Lambda^{XT}_{\sigma,\sigma'}, \Lambda^{YT}_{\sigma,\sigma'}\right] \tag{14}$$

in which $\Lambda^{XY}_{\sigma,\sigma'}$, $\Lambda^{XT}_{\sigma,\sigma'}$, $\Lambda^{YT}_{\sigma,\sigma'}$ are similarly defined as Eq. (8), but for structuring DoG-filtered plane-images instead

Algorithm 2: Encoding of DoDGF^{3D} descriptor.

Input: A video $\mathcal{V}$; a set $\mathcal{F}$ of k orders; a pair of standard deviations (σ, σ').

Output: A DoDGF^{3D} descriptor of $\mathcal{V}$.

for $k \in \mathcal{F}$ **do**

//Compute filtered volumes according to order k

$\mathcal{V}^{\sigma,\sigma'}_{\partial x^k} = \text{DoDG}^{3D}_{\sigma,\sigma',\partial x^k} * \mathcal{V}$;

$\mathcal{V}^{\sigma,\sigma'}_{\partial y^k} = \text{DoDG}^{3D}_{\sigma,\sigma',\partial y^k} * \mathcal{V}$;

$\mathcal{V}^{\sigma,\sigma'}_{\partial z^k} = \text{DoDG}^{3D}_{\sigma,\sigma',\partial z^k} * \mathcal{V}$;

$\Omega = \{\mathcal{V}^{\sigma,\sigma'}_{\partial x^k}, \mathcal{V}^{\sigma,\sigma'}_{\partial y^k}, \mathcal{V}^{\sigma,\sigma'}_{\partial z^k}, |\mathcal{V}^{\sigma,\sigma'}_{\partial x^k}|, |\mathcal{V}^{\sigma,\sigma'}_{\partial y^k}|, |\mathcal{V}^{\sigma,\sigma'}_{\partial z^k}|\}$;

for $\mathcal{V}^k_G \in \Omega$ **do**

Split $\mathcal{V}^k_G$ into collections of plane-images: $\{f'_{XY}, f'_{XT}, f'_{YT}\}$;

$t_{XY} = t_{XT} = t_{YT} =$ an array of zeros;

for $\mathcal{I}_1 \in f'_{XY}, \mathcal{I}_2 \in f'_{XT}, \mathcal{I}_3 \in f'_{YT}$ **do**

$t_{XY} = t_{XY} + \Psi(\mathcal{I}_1)$;

$t_{XT} = t_{XT} + \Psi(\mathcal{I}_2)$;

$t_{YT} = t_{YT} + \Psi(\mathcal{I}_3)$;

end for

$\Upsilon(\mathcal{V}^k_G) = \left[\frac{1}{\mathcal{N}'_{XY}} t_{XY}, \frac{1}{\mathcal{N}'_{XT}} t_{XT}, \frac{1}{\mathcal{N}'_{YT}} t_{YT}\right]$;

end for

//Concatenate results of computing $\mathcal{V}^k_G \in \Omega$

$\text{DoDGF}^{3D}_{\sigma,\sigma',k} = \biguplus_{\mathcal{V}^k_G \in \Omega} \Upsilon(\mathcal{V}^k_G)$;

end for

//Concatenate all the obtained histograms

$\text{DoDGF}^{3D}_{\sigma,\sigma',\mathcal{F}} = \biguplus_{k \in \mathcal{F}} \text{DoDGF}^{3D}_{\sigma,\sigma',k}$;

of addressing the DoDG-filtered ones. For instance of encoding the collection f_{XY} of raw plane-images, $\Lambda^{XY}_{\sigma,\sigma'}$ is formed as

$$\Lambda^{XY}_{\sigma,\sigma'} = \frac{1}{\mathcal{N}} \sum_{\mathcal{I} \in f_{XY}} \left[\Psi(\mathcal{I}^{\sigma,\sigma'}_{DoG}), \Psi(|\mathcal{I}^{\sigma,\sigma'}_{DoG}|)\right] \quad (15)$$

Also based on the construction of DoDGF^{3D}, the 3D DoG-filtered features (DoGF$^{3D}_{\sigma,\sigma'}$) are structured as

$$\text{DoGF}^{3D}_{\sigma,\sigma'}(\mathcal{V}) = \left[\Upsilon(\mathcal{V}^{\sigma,\sigma'}_{DoG}), \Upsilon(|\mathcal{V}^{\sigma,\sigma'}_{DoG}|)\right] \quad (16)$$

It should be noted that the 2D/3D DoG filterings were exploited in the prior works (i.e., FoSIG [17], V-BIG [18], RUBIG [19]), but for capturing the absolute-filtered features. In the meanwhile, the DoGF$^{2D/3D}_{\sigma,\sigma'}$ descriptors are here proposed to capture more the bipolar-filtered ones of those filterings due to an objective comparison to DoDGF$^{2D/3D}_{\sigma,\sigma',\mathcal{F}}$ in abilities of DT classification.

Consequently, it can be stated that our DoDG-based descriptors have some following benefits to enhance the performance compared to other local Gaussian-based ones:

- Our DoDGF$^{2D/3D}_{\sigma,\sigma',\mathcal{F}}$ descriptors are enriched more spatio-temporal characteristics extracted from both bipolar and absolute DoDG-filtered outcomes instead of only absolute features from the DoG-filtered ones in FoSIG, V-BIG, and RUBIG (see Table 6 for evaluations of their contributions).
- It can take advantage of more complementary features by addressing DoDG in high-order gradients. This allows DoDGF$^{2D/3D}_{\sigma,\sigma',\mathcal{F}}$ to capture more scale-filtered information to enhance the performance (see Table 7).
- Addressing our DoDG$^{2D/3D}$ kernels could produce more DoDG-filtered outcomes which are complementary for the local DT encoding due to Eqs. (7) and (10). In the meanwhile, only one outcome done by the DoG$^{2D/3D}$ filterings is exploited in FoSIG, V-BIG, RUBIG, and DoGF$^{2D/3D}_{\sigma,\sigma'}$ due to Eq. (13).

4. Experiments and evaluations

4.1. Datasets and protocols

The benchmark datasets for evaluating our proposed descriptors in DT classification are expressed in this section. A brief of those is shown in Table 1 for a quick reference.

UCLA dataset [29] consists of 200 videos recorded in $110 \times 160 \times 75$ dimension to capture textural motions (see Figure 6(a) for several instances of DT videos). The following protocols are usually addressed for DT recognition.

- *50-class:* 200 videos are grouped into 50 classes with 4 sequences for each. Two popular protocols are used for assessments: leave-one-out [23, 66] and 4-fold cross validation [73, 67].

- *9-class and 8-class:* 200 videos are arranged into 9 classes with different numbers of sequences to form *9-class* scheme: "boiling water" (8), "plants" (108), "sea" (12), "fire" (8), "flowers" (12), "fountains" (20), "smoke" (4), "water" (12), and "waterfall" (16), where the numbers in parentheses indicate their quantities. Due to the dominance of "plants" class, it is removed to form *8-class* with more challenges [74]. Following settings in [42, 73], a half of videos in each group is randomly selected for training and the rest for testing. The trial is repeated 20 times and the final rate is obtained from the average of those.

DynTex dataset [41] consists of 650 high-quality sequences recorded in various conditional environments (see Figure 6(b) for several instances of DT videos). Following settings in [23, 67], DynTex's challenging schemes are often addressed for DT recognition using leave-one-out protocol.

- *DynTex35* is composed as follows. Each of 35 selected videos is split into 10 sub-videos subject to its spacial axes to correspondingly form 35 categories.
- *Alpha* includes three categories as "Sea"(20), "Trees"(20), and "Grass"(20).
- *Beta* includes 162 sequences divided into 10 classes with different numbers of videos in each: "sea(20)", "escalator(7)", "fountains(20)", "calm water(20)", "smoke(16)", "vegetation(20)", "trees(20)", "flags(20)", "traffic(9)", and "rotation(10)".
- *Gamma* has 10 classes of 264 sequences as "flags(31)", "naked trees(25)", "flowers(29)", "calm water(30)", "foliage(35)", "sea(38)", "escalator(7)", "grass(23)", "fountains(37)", and "traffic(9)".

Herein, the numbers in parentheses indicate cardinality of corresponding categories.

DynTex++ dataset [42] consists of 36 categories with 100 sub-videos of $50 \times 50 \times 50$ dimension for each, i.e., 3600 sequences in total. These sub-videos are composed by capturing the major turbulent DTs from 345 raw videos of DynTex. Following settings in [23, 42], a half number of videos in each category is randomly addressed for training and the rest for testing. The final result is then reported by the average of 10 trials.

Figure 6: Samples of videos in UCLA (a), DynTex (b), DTDB (c).

DTDB dataset [56] is recently a large scale dataset of DT videos for principally evaluating effectiveness of CNN-based proposals. It consists of over 10000 DT videos with a total of ∼3.5M frames captured from different sources: websites, handled cameras, etc. (see Figure 6(c) for some samples). Two challenging schemes are addressed for DT recognition as follows.

- *Dynamics* scheme is arranged into 18 categories so that its DT videos just include features of dynamics, i.e., independent of spatial appearance.
- *Appearance* scheme has 45 classes only including features of spatial appearance, i.e., independent of dynamics.

Following settings in [56], 70% of samples in each category is randomly selected for training and the rest (30%) for testing. This trial is repeated 10 runtimes and the final rate is reported by the average of them.

4.2. Experimental settings

For DoDG filtering processes: In experiments of this work, we conduct $\mathrm{DoDG}^{2D/3D}_{\sigma,\sigma',\partial x_i^k}$ in four orders (i.e., $\{1^{st}, 2^{nd}, 3^{rd}, 4^{th}\}$) of gradients with direction axes for the convolving operation $x, y, z \in [-3\sigma, 3\sigma]$. Pairs of standard deviations are empirically investigated as $\{(\sigma, \sigma')\} = \{(0.5, 0.7), (0.5, 1), (0.7, 1), (1, 1.3), (1, 1.5)\}$.

For structuring $\mathrm{DoDGF}^{2D/3D}_{\sigma,\sigma',\mathcal{F}}$ *descriptors:* To construct

Table 1: A brief of properties of benchmark DT datasets.

Dataset	Sub-dataset	#Videos	Resolution	#Classes	Protocol
UCLA	50-class	200	$110 \times 160 \times 75$	50	LOO and 4fold
	9-class	200	$110 \times 160 \times 75$	9	50%/50%
	8-class	92	$110 \times 160 \times 75$	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%
DTDB	Dynamics	> 10000	different dimensions	18	70%/30%
	Appearance	> 9000	different dimensions	45	70%/30%

Note: LOO and 4fold are leave-one-out and four cross-fold validation respectively. 50%/50% is 50% random samples for training and the remain (50%) for testing.

Table 2: A comparison of various bins of LBP-based descriptors.

Method	#bins	$P = 8$
LBP-TOP^{u2} [64]	$3(P(P-1)+3)$	177
VLBP [64]	2^{3P+2}	-
CVLBP [68]	$3 \times 2^{3P+2}$	-
HLBP [67]	6×2^P	1536
CLSP-TOP^{riu2} [73]	$6(P+2)^2$	600
WLBPC [66]	6×2^P	1536
MEWLSP [70]	6×2^P	1536
CVLBC [69]	$2(3P+3)^2$	1458
CSAP-TOP^{riu2}[75]	$12(P+2)^2$	1200
FD-MAP$^{u2}_{L=2}$ [47]	$216P((P-1)+3))+16$	12760
HILOP [76]	$3P(P(P-1)+3)$	1416
FoSIG [17]	$12(P+2)^2$	1200
V-BIG [18]	$12(P+2)^2$	1200
RUBIG [19]	$36(P+2)^2$	3600
HoGF$^{2D}_{\sigma,1^{st}}$ [20]	$36(P+2)^2$	3600
HoGF$^{3D}_{\sigma,1^{st}}$ [20]	$48(P+2)^2$	4800
DoGF$^{2D}_{\sigma,\sigma'}$	$12(P+2)^2$	1200
DoGF$^{3D}_{\sigma,\sigma'}$	$12(P+2)^2$	1200
Our DoDGF$^{2D}_{\sigma,\sigma',1^{st}}$	$24(P+2)^2$	2400
Our DoDGF$^{3D}_{\sigma,\sigma',1^{st}}$	$36(P+2)^2$	3600

Note: P denotes the concerned neighbors. "-" means "not available". Dimension of all above descriptors is referred to their basic parameters used for encoding a given video.

our DoDG-based descriptors, we simply utilize CLBP[2], one of the most popular local operators, with the 3D-joint setting of $riu2$ mapping and a supporting region $(P, R) = (8, 1)$. It means $\Psi = \text{CLBP}^{\text{riu2}}_{8,1}$ corresponding to $\mathcal{H}_\Psi = 2(P+2)^2$ bins for a pattern description, where P denotes a number of neighbors involved in the DT encoding. Consequently, it takes a small dimension for single-scale analysis of high-order DoDG filterings (i.e., $|\mathcal{F}|$=1) to describe a given video, just $4 \times 3 \times |\mathcal{F}| \times \mathcal{H}_\Psi = 2400$ bins for DoDGF$^{2D}_{\sigma,\sigma',\mathcal{F}}$ and $6 \times 3 \times |\mathcal{F}| \times \mathcal{H}_\Psi = 3600$ bins for DoDGF$^{3D}_{\sigma,\sigma',\mathcal{F}}$, where $|\mathcal{F}| = \mathbf{card}(\mathcal{F})$ denotes the number of k-orders in $\mathcal{F}$ taking into account multi-order analysis. Table 2 shows a comprehensive comparison between dimension of DoDGF$^{2D/3D}$ descriptors and the dimension of other LBP-based ones.

For structuring DoGF$^{2D/3D}_{\sigma,\sigma'}$ *descriptors:* In order to make an objective comparison, the same settings should be addressed for the construction of the DoG-based descriptors. It means that the pre-defined pairs of $\{(\sigma, \sigma')\}$ are also used for the DoG filterings, while $\Psi = \text{CLBP}^{\text{riu2}}_{8,1}$ is exploited for the local encoding of the DoG-filtered outcomes. As a result, it takes $2 \times 3 \times \mathcal{H}_\Psi = 1200$ bins for both of DoGF$^{2D/3D}_{\sigma,\sigma'}$ descriptors.

For classifying DTs: To evaluate the performances of our DoDGF$^{2D/3D}_{\sigma,\sigma',\mathcal{F}}$ and the DoG-based ones (i.e., DoGF$^{2D/3D}_{\sigma,\sigma'}$), the linear multi-class SVM classifier of LIBLINEAR [79] is used with the default parameters.

4.3. Complexity of proposed DoDG-based descriptors

In this section, the complexity of computing DoDGF$^{2D/3D}_{\sigma,\sigma',\mathcal{F}}$ descriptors is comprehensively discussed and compared to DoGF$^{2D/3D}_{\sigma,\sigma'}$ and other LBP-based ones. In general, it can be verified that the computational cost of DoDGF$^{2D/3D}_{\sigma,\sigma',\mathcal{F}}$ is as simple as that of other LBP-based ones. This is thanks to the separable and linear properties of DoDG's convolving operations which are inherited from the well-known Gaussian filtering kernel.

Indeed, for a video $\mathcal{V}$ with $\mathcal{H} \times \mathcal{W} \times \mathcal{T}$ dimension, let $\mathcal{Q}_{\text{LBP}} = \mathcal{O}(P \times \mathcal{H} \times \mathcal{W})$ be the complexity of LBP [26] for encoding a plane-image, where $P \in \mathbb{Z}^+$ denotes a number of concerning neighbors. So the complexity of LBP-TOP [64] for encoding $\mathcal{V}$: $\mathcal{Q}_{\text{LBP-TOP}} \approx \mathcal{T} \times \mathcal{Q}_{\text{LBP}}$. Since CLBP

[2]Operator CLBP [16] is addressed in this work for a purpose of simplicity in implementing and evaluating the effectiveness of our novel DoDG filtering for DT representation compared to the well-known DoG. It could be absolutely replaced by other robust ones for further improvement in practice, e.g., CLBC [28], LDP-based [22, 21], LVP-based [77, 48], LRP [19], MRELBP [78], etc.

[16] with its three complementary components is taken into account encoding a plane-image: $\mathcal{Q}_{\text{CLBP}} \approx 3 \times \mathcal{Q}_{\text{LBP}}$, it could be inferred that the computational cost of CLBP for encoding $\mathcal{V}$ is $\mathcal{Q}_{\text{CLBP-TOP}} \approx 3 \times \mathcal{Q}_{\text{LBP-TOP}}$. Accordingly, the complexity of $\text{DoDGF}^{2D/3D}_{\sigma,\sigma',\mathcal{F}}$ is estimated as $\mathcal{Q}_{\text{DoDGF}^{2D}} \approx |\mathcal{F}| \times \mathcal{T} \times (4 \times \mathcal{Q}_{\text{CLBP}} + \mathcal{Q}_{\text{DoDG}^{2D}})$ for $\text{DoDGF}^{2D}_{\sigma,\sigma',\mathcal{F}}$, and $\mathcal{Q}_{\text{DoDGF}^{3D}} \approx |\mathcal{F}| \times (6 \times \mathcal{Q}_{\text{CLBP-TOP}} + \mathcal{Q}_{\text{DoDG}^{3D}})$ for $\text{DoDGF}^{3D}_{\sigma,\sigma',\mathcal{F}}$, where $\mathcal{Q}_{\text{DoDG}^{2D/3D}}$ is the cost of corresponding $\text{DoDG}^{2D/3D}$ filterings involved in the DT representation (refer to Algs. 1 and 2 for their computing structures). Due to the separable and linear properties of the DoDG filterings as well as the much smallness of $|\mathcal{F}|$ (e.g., $|\mathcal{F}| = 2$ for two orders in Table 7), $\mathcal{Q}_{\text{DoDG}^{2D/3D}}$ and $|\mathcal{F}|$ can be ignored. Consequently, $\mathcal{Q}_{\text{DoDGF}^{2D/3D}} \approx \mathcal{O}(P \times \mathcal{H} \times \mathcal{W} \times \mathcal{T})$. Also, addressing CLBP for encoding $\mathcal{V}$ (see Sections 3.4 and 4.2), the computational cost of $\text{DoGF}^{2D/3D}_{\sigma,\sigma'}$ can be conducted as $\mathcal{Q}_{\text{DoGF}^{2D/3D}} \approx \mathcal{O}(P \times \mathcal{H} \times \mathcal{W} \times \mathcal{T})$. Furthermore, referred to complexity estimation of other LBP-based methods presented in [19], our $\mathcal{Q}_{\text{DoDGF}^{2D/3D}}$ is also the same order as FoSIG [17], V-BIG [18], RUBIG [19], CSAP-TOP [75], CVLBP [68], CVLBC [69], VLBP [64], HoGF [20], etc. (refer to those works for more detail of computation).

In regard to processing time, our $\text{DoDGF}^{2D/3D}_{\sigma,\sigma',1^{st}}$ descriptors and those based on the DoG (i.e., $\text{DoGF}^{2D/3D}_{\sigma,\sigma'}$) are implemented on the alike computing system: a 64-bit Linux desktop of CPU Core i7 3.4GHz 16G RAM. This is to make an impartial evaluation with other LBP-based ones done in [19]. Table 3 shows that runtime of encoding the DoDG-based descriptors of a $50 \times 50 \times 50$ video is nearly the same as that of other LBP-based ones. In addition, it should be noted that all runtimes in Table 3 are reported using the CPU in only one thread for running their raw MATLAB codes. In the case of addressing 4 multi-threads, it takes about 0.26s and 0.29s for encoding $\text{DoDGF}^{2D}_{\sigma,\sigma',1^{st}}$ and $\text{DoDGF}^{3D}_{\sigma,\sigma',1^{st}}$ respectively.

Table 3: 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 [64]	-	$\{(4,1)\}$	-	≈ 0.22
LBP-TOP [64]	-	$\{(8,1)\}$	u2	≈ 0.15
CLSP-TOP [73]	-	$\{(8,1)\}$	riu2	≈ 0.27
CSAP-TOP [75]	-	$\{(8,1)\}$	riu2	≈ 0.50
HILOP [76]	-	$\{(8,\{1,2\})\}$	u2	≈ 0.42
FoSIG [17]	$\{(0.5,6)\}$	$\{(8,1)\}$	riu2	≈ 0.37
V-BIG [18]	$\{(0.5,6)\}$	$\{(8,1)\}$	riu2	≈ 0.35
RUBIG [19]	$\{(0.5,6)\}$	$\{(8,1)\}$	riu2	≈ 0.56
$\text{HoGF}^{2D}_{\sigma,1^{st}}$ [20]	$\{\sigma = 1\}$	$\{(8,1)\}$	riu2	≈ 0.54
$\text{HoGF}^{3D}_{\sigma,1^{st}}$ [20]	$\{\sigma = 1\}$	$\{(8,1)\}$	riu2	≈ 0.70
$\text{DoGF}^{2D}_{\sigma,\sigma'}$	$\{(0.7,1)\}$	$\{(8,1)\}$	riu2	≈ 0.37
$\text{DoGF}^{3D}_{\sigma,\sigma'}$	$\{(0.7,1)\}$	$\{(8,1)\}$	riu2	≈ 0.35
Our $\mathbf{DoDGF^{2D}_{\sigma,\sigma',1^{st}}}$	$\{(0.7,1)\}$	$\{(8,1)\}$	riu2	≈ 0.58
Our $\mathbf{DoDGF^{3D}_{\sigma,\sigma',1^{st}}}$	$\{(0.7,1)\}$	$\{(8,1)\}$	riu2	≈ 0.79

Note: "-" means "not available". Runtimes of HILOP [76] and $\text{DoGF}^{2D/3D}$ are implemented by this work while the others are referred to implementations of [19]. It should be noted that all above runtimes are reported using a CPU in only one thread for running their raw MATLAB codes.

4.4. Advantages of DoDG filterings

4.4.1. Robustness to the issues of DT description

Thanks to taking our DoDG into account the filterings, all DoDG-filtered outcomes are complementary and robust to environmental changes, illumination, and noise. This allows that local spatio-temporal features extracted from these outcomes are more insensitive for DT encoding compared to those extracted from a raw video. Indeed, in order to evaluate this advantageous property, we investigate DoDG on noisy datasets to evaluate its ability of noise-resistance.

Accordingly, we address the Gaussian zero-mean noise model with different signal-to-noise ratio (SNR) levels, i.e., $\text{SNR}_{dB} \in \{1,2,3,4,5\}$, to add noise into UCLA [29] - the simple dataset, and DynTex [41] - the more challenging one (see Table 1 for their attributes in detail). For each of them, we have achieved 5 noise-datasets corresponding to 5 SNR_{dB} levels used for the noise-adding process (see Figure 7 for noise-instances). We then evaluate our $\text{DoDGF}^{2D/3D}_{\sigma,\sigma',\mathcal{F}}$ exploiting the 1^{st}-order DoDG-filtered outcomes with $(\sigma,\sigma') = (0.7,1)$ on those noisy datasets.

Table 4: Performances (%) on different Gaussian noise subsets: *50-4fold* of UCLA and *Gamma* of DynTex.

						SNR$_{dB}$ for *50-4fold*						SNR$_{dB}$ for *Gamma*					
Descriptor	Filter	Derivative	$\{(\sigma[,\sigma'])\}$	$\{(P,\{R\})\}$	Mapping	dB=1	dB=2	dB=3	dB=4	dB=5	No-dB	dB=1	dB=2	dB=3	dB=4	dB=5	No-dB
VLBP* [64]	None	-	-	$\{(4,1)\}$	-	91.00	93.00	92.00	94.00	94.00	96.00	87.12	88.64	89.02	90.91	90.53	92.80
LBP-TOP* [64]	None	-	-	$\{(8,1)\}$	u2	97.50	99.00	99.50	99.00	98.50	97.50	77.65	81.82	84.47	86.36	87.12	93.56
CLSP-TOP* [73]	None	-	-	$\{(8,1)\}$	riu2	98.00	**100**	99.50	99.50	99.00	99.00	82.95	84.85	84.47	86.36	87.50	93.18
HILOP* [76]	None	-	-	$\{(8,\{1,2\})\}$	u2	99.50	99.50	99.50	99.50	99.50	99.50	88.64	89.77	90.91	90.91	91.29	92.42
CLBP$_{S/M/C}$ [16]	None	-	-	$\{(8,1)\}$	riu2	99.50	99.50	99.50	99.00	99.50	99.50	85.98	87.12	87.88	88.64	89.39	92.80
ZoGF^{2D*}	Orig. Gau.	0^{th}-order	$\{\sigma=1\}$	$\{(8,1)\}$	riu2	**100**	**100**	99.50	99.00	99.00	**100**	88.64	90.15	89.39	89.39	88.64	92.42
ZoGF^{3D*}	Orig. Gau.	0^{th}-order	$\{\sigma=1\}$	$\{(8,1)\}$	riu2	**100**	**100**	**100**	**100**	**100**	**100**	90.53	90.53	90.91	90.15	90.91	93.56
HoGF^{2D*} [20]	Gau. gradi.	1^{st}-order	$\{\sigma=1\}$	$\{(8,1)\}$	riu2	**100**	**100**	**100**	**100**	**100**	**100**	90.53	90.53	90.15	90.91	90.53	93.56
HoGF^{3D*} [20]	Gau. gradi.	1^{st}-order	$\{\sigma=1\}$	$\{(8,1)\}$	riu2	**100**	**100**	**100**	**100**	**100**	**100**	**90.91**	**92.05**	**93.18**	**92.05**	**92.05**	**96.21**
DoGF^{2D}	DoG	0^{th}-order	$\{(0.7,1)\}$	$\{(8,1)\}$	riu2	**100**	**100**	**100**	**100**	**100**	**100**	81.06	86.74	88.64	89.02	88.26	92.42
DoGF^{3D}	DoG	0^{th}-order	$\{(0.7,1)\}$	$\{(8,1)\}$	riu2	**100**	99.50	**100**	99.50	**100**	**100**	87.88	89.77	90.15	91.29	89.77	94.70
Our DoDGF^{2D}	DoDG	1^{st}-order	$\{(0.7,1)\}$	$\{(8,1)\}$	riu2	**100**	**100**	**100**	**100**	**100**	**100**	89.77	90.53	89.77	90.53	91.29	95.08
Our DoDGF^{3D}	DoDG	1^{st}-order	$\{(0.7,1)\}$	$\{(8,1)\}$	riu2	**100**	**100**	**100**	**100**	**100**	**100**	**90.91**	91.67	91.67	91.67	**92.80**	**96.21**

Note: "-" means "not available". No-dB denotes results without Gaussian noise added into. "$_{S/M/C}$" denotes a 3D-jointed histogram of CLBP's components. CLSP-TOP [73] is formed with thresholding parameters $a=0, b=1$. "*" denotes results transferred from the prior work [20].

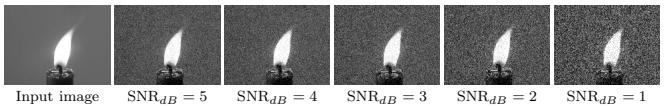


Figure 7: Noise-instances obtained by using different levels of SNR$_{dB}$ on a plane-image in a video of UCLA dataset.

In addition, for a comprehensive evaluation of the noise-resistant ability, several relevant DT descriptors are also addressed on these datasets: DoG-based descriptors, i.e., DoGF$^{2D/3D}_{\sigma,\sigma'}$ defined in Section 3.4, and other LBP-based ones without taking any filters into account their DT encoding, e.g., VLBP [64], LBP-TOP [64], CLSP-TOP [73], HILOP [76]. Their specific parameters along with the achieved results of DT recognition are presented in Table 4.

It can be seen from Table 4 that taking DoDG into account the DT encoding makes our proposed DoDG-based descriptors more robust against noise compared to the DoG-based ones and other LBP-based variants as well. Specifically, our DoDGF$^{2D/3D}$ descriptors absolutely resist to the Gaussian noise for the simple scheme, i.e., *50-4fold*. In the meanwhile, except that the VLBP's performance has sharply decreased by 3%, the noise-resistant ability of the rest is approximately the same execution in general (see Table 4). On the challenging scheme, i.e., *Gamma*, the performance of both DoDGF^{2D} and DoDGF^{3D} has dropped by about 2%, but that of DoDGF^{3D} has the better rates in more "stability". In comparison to the DoG-based descriptors, DoGF^{3D} is much better the 2D one in both rates and noise-resistant ability. In terms of the ability of other LBP-based variants, all of them have a sharp decrease compared to ours (see Figure 8 for a graphical view). This has proved the impressive property of DoDG making our DoDG-based descriptors more robust in noisy conditions. In addition, it can be seen from Table 4 that DoDGF$^{2D/3D}$ has the nearly same levels of noise resistance as HoGF$^{2D/3D}$'s [20].

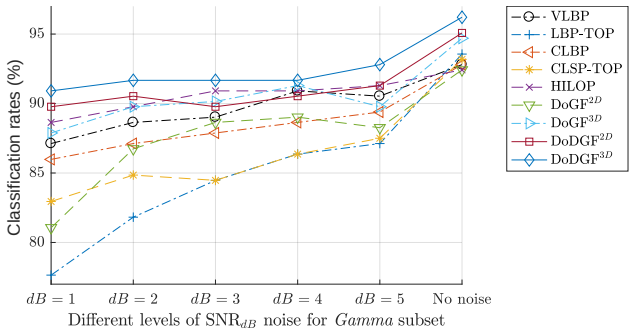


Figure 8: Impacts of Gaussian noise on DoDGF$^{2D/3D}$ compared to DoGF$^{2D/3D}$ and other LBP-based descriptors.

Besides, we also investigate the affects of different density levels of salt-and-pepper noise ρ on the performance of our DoDGF$^{2D/3D}$ in comparison with the conventional DoG-based descriptors (i.e., DoGF$^{2D/3D}$). Accordingly, we add the salt-and-pepper noise with levels

Table 5: Performances (%) on different density levels of salt-and-pepper noise on *Gamma* of DynTex.

Descriptor	ρ=10%	ρ=20%	ρ=30%	ρ=40%	ρ=50%	ρ=60%	ρ=70%
DoGF^{2D}	92.42	88.26	78.79	71.97	70.45	65.53	59.09
DoGF^{3D}	91.67	90.15	87.50	81.06	72.73	61.36	57.20
Our DoDGF^{2D}	**93.18**	**92.05**	89.02	88.64	87.12	81.44	76.14
Our DoDGF^{3D}	91.67	89.77	**89.39**	**89.39**	**87.50**	**86.74**	**83.71**

Table 6: Comparing contributions of DoG and 1^{st}-order DoDG^{2D}.

DoG/DoDG filtered complement(s)	#bins	Dyn35	Beta	Gamma	Dyn++
$\mathcal{I}^{0.7,1}_{\partial x^1}$	600	98.86	92.59	91.29	92.93
$\mathcal{I}^{0.7,1}_{\partial y^1}$	600	99.43	92.59	93.18	93.89
$\lvert\mathcal{I}^{0.7,1}_{\partial x^1}\rvert$	600	97.43	91.36	90.91	93.94
$\lvert\mathcal{I}^{0.7,1}_{\partial y^1}\rvert$	600	96.57	93.21	90.53	93.83
$\lvert\mathcal{I}^{0.7,1}_{\partial x^1}\rvert + \lvert\mathcal{I}^{0.7,1}_{\partial y^1}\rvert$	1200	98.00	95.06	93.18	95.62
$\mathcal{I}^{0.7,1}_{\partial x^1} + \mathcal{I}^{0.7,1}_{\partial y^1}$	1200	98.86	95.06	93.94	95.19
$\mathcal{I}^{0.7,1}_{\partial x^1} + \mathcal{I}^{0.7,1}_{\partial y^1} + \lvert\mathcal{I}^{0.7,1}_{\partial x^1}\rvert + \lvert\mathcal{I}^{0.7,1}_{\partial y^1}\rvert$	2400	**99.43**	**95.68**	**95.08**	**96.40**
$\mathcal{I}^{0.7,1}_{\text{DoG}} + \lvert\mathcal{I}^{0.7,1}_{\text{DoG}}\rvert$	1200	98.00	91.98	92.42	94.86

Note: Dyn35 and Dyn++ stand for DynTex35 sub-set and DynTex++ respectively.

$\rho \in \{10\%, 20\%, 30\%, 40\%, 50\%, 60\%, 70\%\}$ into the challenging scheme *Gamma*. The DoDGF$^{2D/3D}$ descriptors are then computed with the same settings that was addressed for the performing estimation in above SNR-noise conditions. It can be seen from Table 5 that the resistant ability of our DoDGF$^{2D/3D}$ is much better than the conventional DoG-based ones.

4.4.2. Rich and discriminative features

A DoDG filtering points out more filtered outcomes than the well-known DoG (see Figure 4 in line (a) and two first ones in line (b)). This allows to exploit spatio-temporal features in more forceful contexts to enhance the discrimination power (see Table 6 for their contributions). Also, a DoDG kernel can be computed in higher partial derivatives to conduct high-gradient features for further improvement (see Figure 4 in lines (b) and (c) for instances of high-order DoDG filterings, and Table 7 for their performances). Moreover, addressing various orders of DoDG allows to take into account a hierarchical representation of DTs that somewhat shares the similar links to hierarchical representation in deep-learning models.

4.5. Assessments of DoDG-based descriptors

Based on the experimental results in Table 7, it can be stated that our DoDG is the major factor in order to boost the discrimination of DoDGF$^{2D/3D}$. Hereafter, we discuss their performance thoroughly.

- The performance of DoDG-based descriptors is diminished subject to the increasing high-orders of DoDG involved in the filterings. It is due to the weakness of appearances in the larger-orders. Therein, the odd DoDG kernels often handle denoising in more effect (see Table 7).
- Local patterns extracted from each of the DoDG-filtered outcomes are complementary to enhance the robustness. Indeed, Table 6 shows that DoDGF^{2D} has higher rates when integrating all those features, as mentioned in Section 3.4.
- It can be seen from Table 7 that the DoDGF$^{2D/3D}$ descriptors have the nearly same rates on simple datasets (e.g., UCLA). However, for the challenging schemes (i.e., *Beta* and *Gamma*), the DoDGF^{3D} one has much better results. This has proved that exploiting the 3D DoDG kernel can enrich more robust spacial-filtered information for DT representation compared to using the 2D one. Figure 9 intuitively shows this prominent point.
- Taking a coherence of both odd and even DoDG filterings into account multi-order analysis gives better rates compared to doing that with the whole either odd or even ones (see Table 7 for results in 2-scale of orders). It is due to the fact that the odd and even orders are complementary since the first ones are semi-symmetric shapes while the second ones are symmetric shapes (see Section 3.3 for these properties and Figure 3 for illustration with 1D DoDG kernels).

In general, the single-order DoDGF$^{2D/3D}_{\sigma,\sigma',\mathcal{F}}$ descriptors with the setting of $(\sigma, \sigma') = (0.7, 1)$ often point out the best results on UCLA and Alpha datasets (see Table 7). Moreover, the odd-even DoDGF$^{2D/3D}_{(0.7,1),\mathcal{F}}$ descriptors in

Table 7: DT Classification rates (%) of $\text{DoDGF}^{2D/3D}_{\sigma,\sigma',\mathcal{F}}$ and $\text{DoGF}^{2D/3D}_{\sigma,\sigma'}$ descriptors on benchmark datasets.

Dataset		UCLA								DynTex								DynTex++	
DoGF/DoDGF		50-LOO		50-4fold		9-class		8-class		DynTex35		Alpha		Beta		Gamma			
Order(s)	(σ,σ')	2D	3D	2D	3D	2D	3D	2D	3D	2D	3D	2D	3D	2D	3D	2D	3D	2D	3D
	$(0.5, 0.7)$	**100**	99.50	**100**	99.50	98.10	98.85	96.41	97.83	97.17	97.14	95.00	96.67	93.83	93.83	93.18	92.42	94.94	95.04
	$(0.5, 1)$	**100**	**100**	**100**	**100**	98.25	98.45	96.74	97.83	97.71	97.43	98.33	98.33	92.59	92.59	91.29	92.42	94.62	95.09
0^{th}	$(0.7, 1)$	**100**	**100**	**100**	**100**	97.70	98.75	96.20	97.93	98.00	96.57	**100**	**100**	91.98	91.98	92.42	94.70	94.86	94.98
	$(1, 1.3)$	**100**	**100**	**100**	**100**	97.15	98.70	97.83	97.93	97.71	97.14	98.33	96.67	91.98	90.74	93.18	93.56	94.08	93.63
	$(1, 1.5)$	**100**	**100**	**100**	**100**	97.50	98.90	97.83	97.39	98.29	98.86	98.33	98.33	92.59	92.59	90.53	93.56	94.18	93.36
	$(0.5, 0.7)$	99.50	99.50	99.50	99.50	98.05	98.40	98.70	98.59	97.71	98.29	98.33	98.33	95.06	96.91	94.32	95.45	97.03	97.19
	$(0.5, 1)$	**100**	**100**	**100**	**100**	98.90	98.20	96.52	98.15	99.14	99.43	98.33	98.33	95.68	96.30	94.70	95.45	97.08	96.88
1^{st}	$(0.7, 1)$	**100**	**100**	**100**	**100**	99.05	99.10	98.04	99.24	99.43	**100**	**100**	98.33	95.68	97.53	95.08	96.21	96.40	97.15
	$(1, 1.3)$	**100**	**100**	**100**	**100**	98.50	98.90	96.09	98.04	99.43	**100**	**100**	98.33	95.68	96.91	94.32	96.59	96.51	96.99
	$(1, 1.5)$	**100**	**100**	**100**	**100**	98.70	99.40	96.96	98.26	99.43	98.86	98.33	98.33	95.68	97.53	94.32	95.83	96.21	96.52
	$(0.5, 0.7)$	99.00	99.00	99.00	99.00	98.90	98.65	98.15	97.39	98.00	98.29	**100**	**100**	95.68	96.30	93.56	96.21	95.86	97.09
	$(0.5, 1)$	99.50	99.50	99.50	99.50	99.15	98.90	96.96	98.48	97.71	98.00	**100**	**100**	95.06	94.44	93.56	96.21	96.11	96.84
2^{nd}	$(0.7, 1)$	**100**	**100**	**100**	**100**	99.00	98.60	97.61	**99.57**	98.57	99.14	**100**	**100**	95.06	96.30	93.94	95.08	95.54	96.47
	$(1, 1.3)$	**100**	**100**	**100**	**100**	99.30	99.10	98.70	98.26	98.00	98.29	**100**	**100**	94.44	94.44	92.80	96.21	96.09	96.89
	$(1, 1.5)$	**100**	**100**	**100**	**100**	98.75	99.25	98.37	99.13	98.86	98.29	**100**	98.33	93.83	94.44	93.18	95.45	95.72	96.27
	$(0.5, 0.7)$	**100**	99.50	**100**	99.50	98.70	98.80	98.37	98.70	99.14	99.14	98.33	98.33	94.44	96.91	94.70	95.45	96.91	97.15
	$(0.5, 1)$	**100**	99.50	**100**	99.50	99.05	98.75	98.70	98.15	99.43	99.14	98.33	98.33	96.30	96.30	94.70	95.08	96.82	96.51
3^{rd}	$(0.7, 1)$	**100**	**100**	**100**	**100**	99.10	99.35	96.63	99.46	99.43	98.57	98.33	98.33	95.68	96.30	92.80	95.83	95.95	96.39
	$(1, 1.3)$	**100**	**100**	**100**	**100**	98.60	99.10	95.22	98.70	99.43	98.57	98.33	98.33	94.44	96.30	92.80	95.83	96.15	96.24
	$(1, 1.5)$	**100**	**100**	**100**	**100**	98.45	**99.75**	98.04	97.83	99.14	98.86	98.33	96.67	95.06	96.30	94.32	96.59	96.06	96.72
	$(0.5, 0.7)$	99.00	99.00	98.50	99.00	98.10	98.00	96.30	95.76	98.29	97.71	98.33	96.67	96.30	95.06	92.80	93.18	95.56	96.72
	$(0.5, 1)$	**100**	**100**	**100**	**100**	99.35	98.70	97.39	98.48	98.29	96.86	98.33	**100**	93.21	93.21	91.67	95.45	95.69	96.57
4^{th}	$(0.7, 1)$	**100**	**100**	**100**	**100**	98.80	98.15	97.93	98.70	99.14	98.00	98.33	96.67	96.91	95.06	92.42	93.56	96.30	95.69
	$(1, 1.3)$	99.00	98.00	99.00	98.00	98.35	98.50	97.61	95.33	96.86	98.00	98.33	**100**	94.44	94.44	92.80	95.45	95.27	96.11
	$(1, 1.5)$	**100**	**100**	**100**	99.50	99.30	99.10	97.93	98.59	97.71	98.57	**100**	98.33	93.21	92.59	93.18	95.45	95.49	95.62
	$(0.5, 0.7)$	99.50	99.50	99.00	99.50	98.50	98.55	99.02	98.70	96.57	96.29	**100**	**100**	95.68	96.92	94.70	96.21	96.93	97.62
	$(0.5, 1)$	**100**	99.50	**100**	99.50	99.10	98.50	97.39	97.61	99.43	99.43	**100**	**100**	96.30	95.68	94.70	96.97	97.20	97.55
$\{1^{st}, 2^{nd}\}$	$(0.7, 1)$	**100**	**100**	**100**	**100**	**99.55**	99.25	99.13	**99.57**	**99.71**	99.71	**100**	**100**	**97.53**	**98.15**	**96.21**	96.97	97.14	97.52
	$(1, 1.3)$	**100**	**100**	**100**	**100**	98.20	99.40	98.80	96.96	**99.71**	99.71	**100**	**100**	95.06	96.30	94.70	96.97	97.02	97.40
	$(1, 1.5)$	**100**	**100**	**100**	**100**	99.05	98.75	98.26	97.72	**99.71**	99.71	**100**	98.33	95.68	95.68	94.70	96.59	96.96	96.97
	$(0.5, 0.7)$	**100**	**100**	**100**	**100**	99.40	98.65	97.39	97.93	98.86	98.86	98.33	98.33	95.68	96.91	94.32	95.83	**97.54**	97.46
	$(0.5, 1)$	**100**	**100**	**100**	**100**	99.40	98.60	98.91	99.13	99.43	99.71	98.33	98.33	96.30	96.91	94.70	96.59	97.23	97.48
$\{1^{st}, 3^{rd}\}$	$(0.7, 1)$	**100**	**100**	**100**	**100**	98.70	98.75	98.70	99.13	99.43	99.71	98.33	98.33	95.68	96.30	93.56	96.97	96.73	97.18
	$(1, 1.3)$	**100**	**100**	**100**	**100**	98.50	99.60	95.76	97.83	99.43	99.71	98.33	98.33	95.68	96.30	93.18	**97.35**	96.82	96.83
	$(1, 1.5)$	**100**	**100**	**100**	**100**	99.15	99.05	95.00	98.26	99.14	98.57	98.33	96.67	96.91	96.30	95.45	96.59	96.86	97.07
	$(0.5, 0.7)$	99.00	99.50	99.00	99.50	98.65	98.45	96.85	96.52	97.71	96.00	98.33	98.33	96.91	97.53	**96.21**	96.21	96.91	97.46
	$(0.5, 1)$	**100**	**100**	**100**	**100**	99.25	98.35	99.13	96.96	99.14	99.71	**100**	**100**	**97.53**	95.68	93.18	96.21	97.47	97.49
$\{1^{st}, 4^{th}\}$	$(0.7, 1)$	**100**	**100**	**100**	**100**	99.35	99.35	98.70	96.63	**99.71**	**100**	**100**	**100**	96.91	97.53	94.32	96.97	97.21	**97.95**
	$(1, 1.3)$	99.00	98.00	99.00	98.00	98.55	98.70	96.20	95.11	**99.71**	99.43	**100**	**100**	96.91	95.68	94.32	95.83	97.07	97.47
	$(1, 1.5)$	**100**	**100**	**100**	**100**	99.10	98.70	97.83	97.50	99.14	99.14	**100**	**100**	95.68	95.06	94.32	96.21	96.79	97.34
	$(0.5, 0.7)$	**100**	99.50	99.50	99.50	98.75	98.35	**99.24**	98.37	99.43	99.43	**100**	**100**	96.30	96.91	94.70	96.59	97.09	97.37
	$(0.5, 1)$	**100**	99.50	**100**	99.50	98.95	99.15	97.72	98.26	99.43	99.14	**100**	**100**	**97.53**	97.53	94.70	96.21	96.95	97.08
$\{2^{nd}, 3^{rd}\}$	$(0.7, 1)$	**100**	**100**	**100**	**100**	98.40	99.30	98.70	98.70	99.43	99.43	**100**	**100**	96.91	**98.15**	**96.21**	96.59	96.93	97.03
	$(1, 1.3)$	**100**	**100**	**100**	**100**	97.80	98.90	96.52	98.04	99.43	99.43	**100**	**100**	96.30	96.91	95.45	96.97	97.04	96.62
	$(1, 1.5)$	**100**	**100**	**100**	**100**	98.95	99.35	97.83	97.50	99.14	98.86	**100**	96.67	96.30	95.70	94.70	96.97	97.03	97.01
	$(0.5, 0.7)$	98.00	99.50	98.00	99.50	98.00	98.15	95.76	97.93	98.86	97.14	**100**	**100**	96.91	95.06	94.32	95.83	96.13	97.01
	$(0.5, 1)$	**100**	99.50	**100**	99.50	99.40	99.30	98.70	97.39	98.57	98.86	**100**	**100**	95.06	94.44	92.80	**97.35**	96.49	97.11
$\{2^{nd}, 4^{th}\}$	$(0.7, 1)$	**100**	**100**	**100**	**100**	99.10	99.20	98.70	97.39	99.43	98.00	**100**	**100**	95.68	96.91	93.56	94.70	96.19	96.88
	$(1, 1.3)$	99.00	98.00	99.00	98.00	98.20	98.80	96.63	97.83	97.71	97.71	**100**	**100**	95.68	95.06	92.80	95.83	96.53	96.87
	$(1, 1.5)$	**100**	**100**	**100**	**100**	99.15	99.25	98.04	99.13	98.86	98.86	**100**	**100**	93.83	94.44	92.80	96.21	95.92	96.62
	$(0.5, 0.7)$	99.00	99.50	99.00	99.50	99.00	98.80	95.76	97.28	97.71	98.86	**100**	98.33	96.91	**98.15**	95.45	96.21	97.31	97.16
	$(0.5, 1)$	**100**	**100**	**100**	**100**	99.30	99.50	97.83	98.59	99.43	99.43	**100**	**100**	**97.53**	97.53	94.32	95.83	97.04	96.64
$\{3^{rd}, 4^{th}\}$	$(0.7, 1)$	**100**	**100**	**100**	**100**	99.45	99.55	99.13	97.39	99.43	99.43	**100**	98.33	**97.53**	97.53	**96.21**	96.59	97.09	97.55
	$(1, 1.3)$	99.00	98.00	99.00	98.00	97.95	98.85	95.65	97.18	99.43	99.43	**100**	**100**	**97.53**	96.91	95.08	96.59	96.91	96.53
	$(1, 1.5)$	**100**	**100**	**100**	**100**	98.25	99.35	96.96	97.39	98.86	99.14	**100**	98.33	96.91	95.06	**96.21**	96.59	96.88	96.45

Note: 50-LOO and 50-4fold denote results on 50-class breakdown using leave-one-out and four cross-fold validation. 2D and 3D columns denote results of the 2D and 3D descriptors correspondingly. The 0^{th}-order denotes results of $\text{DoGF}^{2D/3D}_{\sigma,\sigma'}$, while the other orders denote rates of $\text{DoDGF}^{2D/3D}_{\sigma,\sigma',\mathcal{F}}$.

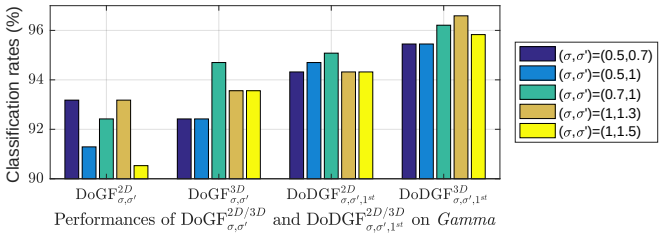


Figure 9: Rates of $\mathrm{DoGF}^{2D/3D}_{\sigma,\sigma'}$ and 1^{st}-order $\mathrm{DoGF}^{2D/3D}_{\sigma,\sigma',1^{st}}$.

multi-order analysis (i.e., $\{1^{st}, 2^{nd}\}$, $\{1^{st}, 4^{th}\}$, $\{2^{nd}, 3^{rd}\}$, $\{3^{rd}, 4^{th}\}$) have produced better performances than the others on all datasets (they also obtain the best results on UCLA and Alpha datasets). It means that on the more challenging schemes (Beta, Gamma, and DynTex++), exploiting complementary information by odd and even orders of DoDG allows to enhance the discrimination power. Among of above those, the 1^{st}-order $\mathrm{DoDGF}^{2D/3D}_{(0.7,1),\{1^{st}\}}$ descriptors should be addressed for mobile applications due to their small dimension, i.e., just 2400 bins for the 2D one and 3600 bins for the 3D. For more strict requirement of accuracy, the setting of multi-gradients $\mathcal{F} = \{1^{st}, 2^{nd}\}$ should be addressed for $\mathrm{DoDGF}^{2D/3D}_{(0.7,1),\mathcal{F}}$ due to the best results. Hereafter, if no settings are specified, the default ones are in the following comprehensive evaluations.

4.6. Comprehensive comparison to DoG-based descriptors

It can be verified from Table 7 that our proposed descriptors using the novel DoDG filterings are much powerful execution compared to those using the well-known DoGs, i.e., rates in the 0^{th}-order rows of Table 7. In consideration of the contributions of complementary filtered outcomes as shown in Table 6, it can assert the prominent performance of DoDG's parts compared to DoG's. This has proved that our novel filtering kernel is more influential for the local DT description.

Furthermore, the DoDG-filtered features are also more discriminative than those of DoGs in FoSIG [17] and V-BIG [18], where both blurred and invariant Gaussian-based characteristics are taken into account the DT encoding. It can be seen from Figure 10 for a comprehensive

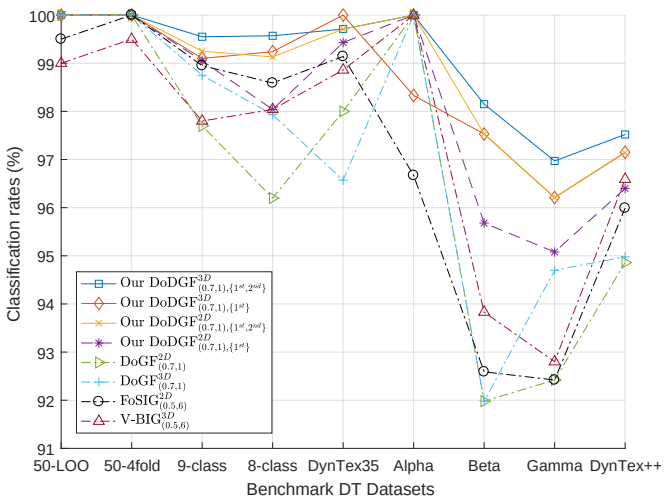


Figure 10: Rates of several local-feature-based descriptors using the same $\mathrm{CLBP}^{riu2}_{8,1}$ for encoding DoG/DoDG-based outcomes.

comparison of their performances, where all the descriptors are constructed by the same $\mathrm{CLBP}^{riu2}_{8,1}$ for capturing spatio-temporal features in DoG/DoDG-based outcomes.

4.7. Comprehensive comparison to HoGF descriptors

In our prior work [20], the 2D/3D Gaussian gradients in multi-scales of standard deviations were directly exploited to filter a video. The obtained $\mathrm{HoGF}^{2D/3D}$ descriptors have good recognition rates compared to state of the art. In this work, thanks to $\mathrm{DoDG}^{2D/3D}$ filterings, our $\mathrm{DoDGF}^{2D/3D}$ descriptors obtain better performance than $\mathrm{HoGF}^{2D/3D}$. Indeed, it can be verified from Table 8 that rates of $\mathrm{DoDGF}^{2D/3D}_{(0.7,1),\{1^{st},2^{nd}\}}$ are generally higher than those of two separate deviation scales of 2-order HoGFs, i.e., $\mathrm{HoGF}^{2D/3D}_{\{\sigma=1\},\{1^{st},2^{nd}\}}$ and $\mathrm{HoGF}^{2D/3D}_{\{\sigma=0.7\},\{1^{st},2^{nd}\}}$, for DT recognition on challenging datasets. Particularly, on DTDB, rates of our $\mathrm{DoDGF}^{2D/3D}_{(0.7,1),\{1^{st},2^{nd}\}}$ are about 1% to 2% better than those of HoGFs (see Table 8). In the case of addressing the best settings, rates of our DoDGFs are just a little lower on *Gamma* (see Table 9) but about 1% higher than those of HoGFs on both DTDB's large-scale schemes (see Table 11). Furthermore, the dimension of DoDGFs is about two thirds smaller than that of HoGFs (see Table 8). Those evidences have proved the significant denoising treatment of our DoDG compared to the Gaussian-

Table 8: Comparing rates of DoDGF$^{2D/3D}$ and HoGF$^{2D/3D}$ [20].

Descriptor	#bins	DynTex			Dyn++	DTDB	
		Dyn35	Beta	Gamma		Dyna	Appe
HoGF$^{2D}_{\{\sigma=0.7\},\{1^{st},2^{nd}\}}$ [20]	7200	99.43	95.06	95.83	**97.43**	67.95	68.84
HoGF$^{2D}_{\{\sigma=1\},\{1^{st},2^{nd}\}}$ [20]	7200	**99.71**	96.91	95.08	97.39	68.84	68.66
DoDGF$^{2D}_{(0.7,1),\{1^{st},2^{nd}\}}$	4800	**99.71**	**97.53**	**96.21**	97.14	**69.81**	**69.84**
HoGF$^{3D}_{\{\sigma=0.7\},\{1^{st},2^{nd}\}}$ [20]	9600	99.14	96.91	96.21	97.71	70.47	71.06
HoGF$^{3D}_{\{\sigma=1\},\{1^{st},2^{nd}\}}$ [20]	9600	**99.71**	96.91	96.59	97.34	70.89	71.11
DoDGF$^{3D}_{(0.7,1),\{1^{st},2^{nd}\}}$	7200	**99.71**	**98.15**	**96.97**	**97.52**	**72.06**	**72.10**

Note: Dyn35 and Dyn++ are shortened for DynTex35 sub-set and DynTex++ respectively while Dyna and Appe stand for DTDB's schemes Dynamics and Appearance. Results of HoGFs [20] on DTDB are reported by this work.

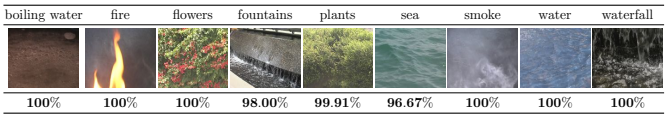


Figure 11: Specific rates on each category of *9-class*.

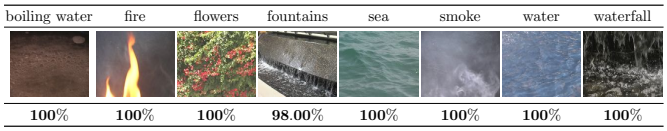


Figure 12: Specific rates on each category of *8-class*.

gradient kernels. In addition, our DoDG-based descriptors obtain the best performance by the same settings of the 2-order form $\{1^{st}, 2^{nd}\}$ for both the DoDG$^{2D/3D}$ filterings, whilst those of HoGFs [20] are different: $\{2^{nd}, 3^{rd}\}$ for the 2D Gaussian-gradient filtering, and $\{3^{rd}, 4^{th}\}$ for the 3D one (see Tables 9 and 11). This assures that our DoDGFs can be more adaptative in real implementations.

4.8. Comprehensive comparison to state of the art

Generally, it can be seen from Table 9 that our DoDG-based descriptors have obtained the best rates compared to all non-deep-learning methods. Their performances are also better than those of deep-learning-based approaches on UCLA as well as very close to those on DynTex and DynTex++. This is certainly thanks to the leverage contribution of our DoDG. Hereunder, we detail particular discussions of those on each benchmark dataset.

4.8.1. Classification on UCLA

It can be verified from Table 9 that thanks to the efficiently denoising processes of DoDG filterings, our DoDG-based descriptors perform very well compared to state of the art, including the deep-learning methods, i.e., DT-CNNs [52]. More specifically, they obtain the best rates of 100% on both schemes of *50-class* and *50-4fold*. In terms of classifying DTs on *9-class* and *8-class*, our proposal is just a little inferior to DNGP [49] (99.6%) on *9-class*, while achieving the highest rate of 99.57% on *8-class* by DoDGF$^{3D}_{(0.7,1),\{1^{st},2^{nd}\}}$, the same as FD-MAP's [47]. It should be noted that DNGP's and FD-MAP's are not better than ours on other schemes (see Table 9). In addition, CVLBC [69] also obtains the nearly same performance as ours but it performs less on DynTex35 and DynTex++. Also, it has not been verified on the challenging scenarios: *Alpha*, *Beta*, and *Gamma* (also see Table 9). For further consideration of enhancement, we present the specific rates of DoDGF$^{3D}_{(0.7,1),\{1^{st},2^{nd}\}}$ in Figure 11 for the *9-class* scheme and Figure 12 for the *8-class* one.

4.8.2. Classification on DynTex

It can be observed from Table 9 that our DoDGF$^{2D/3D}$ descriptors obtain the best rates compared to all non-deep-learning approaches, from over 1% to 3% higher improvement on the challenging schemes (i.e., *Beta* and *Gamma*) than those of MDP-based [21] and RUBIG [19] descriptors, very recent robust methods based on local features for DT representation. Moreover, with the highest rates of 100%, 100%, 98.15%, and 96.97% on *DynTex35*, *Alpha*, *Beta*, and *Gamma* respectively, these results are very close to those of the deep-learning techniques, i.e., DT-CNNs [52], st-TCoF [50], and D3 [55]. It is worth noting that we just use the shallow framework for DT representation versus complicated algorithms addressed by those deep-learning models which their deployment is restricted on mobile devices. For further consideration of improvement, we present the specific rates of DoDGF$^{3D}_{(0.7,1),\{1^{st},2^{nd}\}}$ in Figure 13 for the *Beta* scheme and Figure 14 for the *Gamma* one.

Table 9: Comparison of DT recognition rates (%) on benchmark DT datasets

Category	Dataset	UCLA				DynTex				Dyn++
	Encoding method	50-LOO	50-4fold	9-class	8-class	Dyn35	Alpha	Beta	Gamma	
Optical-flow-based	FDT [47]	98.50	99.00	97.70	99.35	98.86	98.33	93.21	91.67	95.31
	FD-MAP [47]	99.50	99.00	99.35	**99.57**	98.86	98.33	92.59	91.67	95.69
	DDTP [48]	99.00	99.50	98.75	98.04	99.71	96.67	93.83	91.29	95.09
Model-based	AR-LDS [29]	89.90^{N}	-	-	-	-	-	-	-	-
	KDT-MD [30]	-	97.50	-	-	-	-	-	-	-
	NLDR [33]	-	-	-	80.00	-	-	-	-	-
	Chaotic vector [32]	-	-	85.10^{N}	85.00^{N}	-	-	-	-	-
Geometry-based	3D-OTF [37]	-	87.10	97.23	99.50	96.70	83.61	73.22	72.53	89.17
	WMFS [38]	-	-	97.11	96.96	-	-	-	-	-
	NLSSA [40]	-	-	-	-	-	-	-	-	92.40
	KSSA [40]	-	-	-	-	-	-	-	-	92.20
	DKSSA [40]	-	-	-	-	-	-	-	-	91.10
	DFS [74]	-	**100**	97.50	99.20	97.16	85.24	76.93	74.82	91.70
	2D+T [80]	-	-	-	-	-	85.00	67.00	63.00	-
	STLS [39]	-	99.50	97.40	99.50	98.20	89.40	80.80	79.80	94.50
Filter-based	MBSIF-TOP [23]	99.50^{N}	-	-	-	98.61^{N}	90.00^{N}	90.70^{N}	91.30^{N}	97.12^{N}
	B3DF_SMC [25]	99.50^{N}	99.50^{N}	98.85^{N}	98.15^{N}	99.71^{N}	95.00^{N}	90.12^{N}	90.91^{N}	95.58^{N}
	DNGP [49]	-	-	**99.60**	99.40	-	-	-	-	93.80
Local-feature-based	VLBP [64]	-	89.50^{N}	96.30^{N}	91.96^{N}	81.14^{N}	-	-	-	94.98^{N}
	LBP-TOP [64]	-	94.50^{N}	96.00^{N}	93.67^{N}	92.45^{N}	98.33	88.89	84.85^{N}	94.05^{N}
	DDLBP with MJMI [81]	-	-	-	-	-	-	-	-	95.80
	CVLBP [68]	-	93.00^{N}	96.90^{N}	95.65^{N}	85.14^{N}	-	-	-	-
	HLBP [67]	95.00^{N}	95.00^{N}	98.35^{N}	97.50^{N}	98.57^{N}	-	-	-	96.28^{N}
	CLSP-TOP [73]	99.00^{N}	99.00^{N}	98.60^{N}	97.72^{N}	98.29^{N}	95.00^{N}	91.98^{N}	91.29^{N}	95.50^{N}
	MEWLSP [70]	96.50^{N}	96.50^{N}	98.55^{N}	98.04^{N}	99.71^{N}	-	-	-	98.48^{N}
	WLBPC [66]	-	96.50^{N}	97.17^{N}	97.61^{N}	-	-	-	-	95.01^{N}
	CVLBC [69]	98.50^{N}	99.00^{N}	99.20^{N}	99.02^{N}	98.86^{N}	-	-	-	91.31^{N}
	VSCR [82]	99.43	-	-	-	95.43	-	-	-	-
	CSAP-TOP [75]	99.50	99.50	96.80	95.98	**100**	96.67	92.59	90.53	-
	FoSIG [17]	99.50	**100**	98.95	98.59	99.14	96.67	92.59	92.42	95.99
	V-BIG [18]	99.50	99.50	97.95	97.50	99.43	**100**	95.06	94.32	96.65
	HILOP [76]	99.50	99.50	97.80	96.30	99.71	96.67	91.36	92.05	96.21
	MMDP$_{D_M/C}$ [21]	**100**	**100**	98.70	98.70	99.43	98.33	96.91	92.05	95.86
	MEMDP$_{D_M/C}$ [21]	**100**	**100**	98.90	98.70	99.71	96.67	96.91	93.94	96.03
	RUBIG [19]	**100**	**100**	99.20	99.13	98.86	**100**	95.68	93.56	97.08
	HoGF$^{2D}_{\{\sigma=1\},\{2^{nd},3^{rd}\}}$ [20]	**100**	**100**	99.20	98.91	99.71	**100**	97.53	96.59	97.19
	HoGF$^{3D}_{\{\sigma=1\},\{3^{rd},4^{th}\}}$ [20]	**100**	**100**	99.25	**99.57**	99.43	98.33	98.15	97.53	97.63
	Our DoDGF$^{2D}_{(0.7,1),\{1^{st}\}}$	**100**	**100**	99.05	98.04	99.43	**100**	95.68	95.08	96.40
	Our DoDGF$^{2D}_{(0.7,1),\{1^{st},2^{nd}\}}$	**100**	**100**	99.25	99.13	99.71	**100**	97.53	96.21	97.14
	Our DoDGF$^{3D}_{(0.7,1),\{1^{st}\}}$	**100**	**100**	99.10	99.24	**100**	98.33	97.53	96.21	97.15
	Our DoDGF$^{3D}_{(0.7,1),\{1^{st},2^{nd}\}}$	**100**	**100**	99.55	**99.57**	99.71	**100**	98.15	96.97	97.52
Learning-based	DL-PEGASOS [42]	-	97.50	95.60	-	-	-	-	-	63.70
	PI-LBP+super hist [83]	-	**100^{N}**	98.20^{N}	-	-	-	-	-	-
	PD-LBP+super hist [83]	-	**100^{N}**	98.10^{N}	-	-	-	-	-	-
	PCA-cLBP/PI-LBP/PD-LBP [83]	-	-	-	-	-	-	-	-	92.40
	Orthogonal Tensor DL [59]	-	99.80	98.20	99.50	-	87.80	76.70	74.80	94.70
	Equiangular Kernel DL [60]	-	-	-	-	-	88.80	77.40	75.60	93.40
	SOE-Net [84]	-	-	-	-	-	96.70	95.70	92.20	94.40
	st-TCoF [50]	-	-	-	-	-	**100***	**100***	98.11*	-
	PCANet-TOP [54]	99.50*	-	-	-	-	96.67*	90.74*	89.39*	-
	D3 [55]	-	-	-	-	-	**100***	**100***	98.11*	-
	DT-CNN-AlexNet [52]	-	99.50*	98.05*	98.48*	-	**100***	99.38*	**99.62***	98.18*
	DT-CNN-GoogleNet [52]	-	99.50*	98.35*	99.02*	-	**100***	**100***	**99.62***	**98.58***

Note: "-" means "not available". Superscript "*" indicates results using deep learning algorithms. "N" indicates rates 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. Dyn35 and Dyn++ are abbreviated for DynTex35 and DynTex++ datasets respectively. Evaluations of VLBP and LBP-TOP operators are referred to the evaluations of implementations in [67, 50].

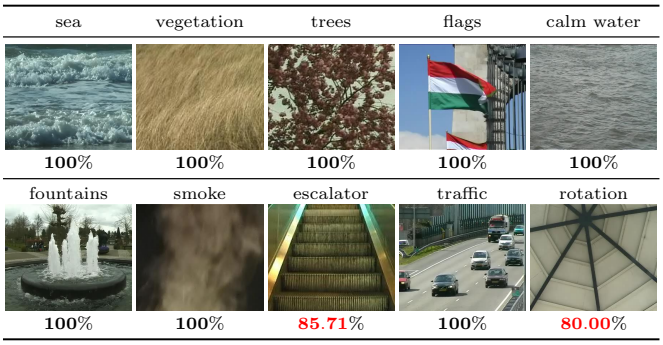


Figure 13: Specific rates on each category of *Beta*. The challenging categories are highlighted in red rates.

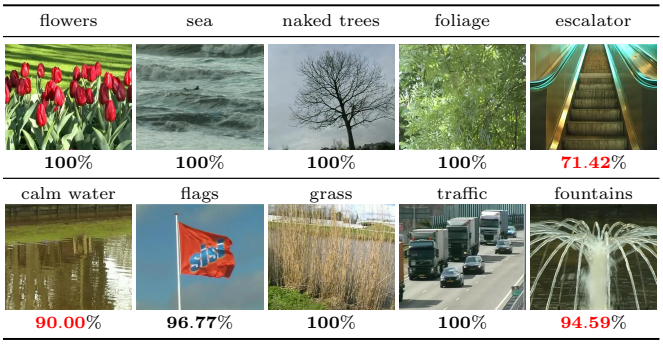


Figure 14: Specific rates on each category of *Gamma*. The challenging categories are highlighted in red rates.

4.8.3. Classification on DynTex++

Our proposal has significant performance on this scheme with over 97% for DoDG-based descriptors in 2-scale analyses of orders (see Table 9). These rates are the best compared to all methods, excluding MEWLSP (98.48%) [70], and DT-CNNs [52] (98.18% for AlexNet and 98.58% for GoogleNet frameworks). Specifically, $\text{DoDGF}^{3D}_{(0.7,1),\{1^{st},2^{nd}\}}$ just obtains 97.52% due to the challenging categories highlighted in red rates in Figure 15. It is noteworthy that MEWLSP's performance is inferior to ours on UCLA (see Table 9). Also, it has not been verified on more challenging schemes, i.e., *Alpha*, *Beta*, *Gamma*. In the meantime, DT-CNNs taking a large number of learned parameters for those frameworks just obtain about 0.5∼1% higher than ours.

4.8.4. Classification on DTDB dataset

Table 10 shows results of our $\text{DoDGF}^{2D/3D}$ on two challenging subsets of DTDB, *Dynamics* and *Appearance*. In

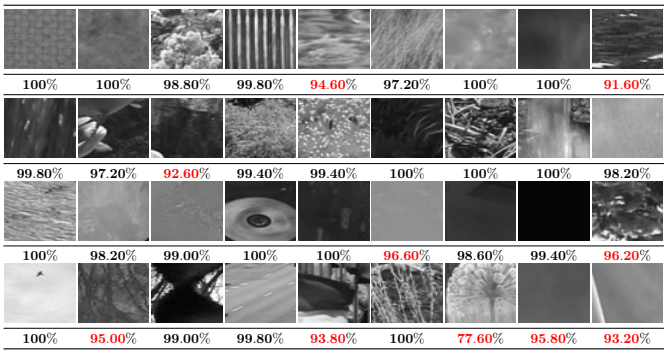


Figure 15: Specific results of DT recognition of $\text{DoDGF}^{3D}_{(0.7,1),\{1^{st},2^{nd}\}}$ on each category of DynTex++. The challenging categories are highlighted in red rates.

general, the rates of DoDGF^{3D} are about 3% better than those of DoDGF^{2D}. This has consolidated the prominence of the filtering kernel DoDG^{3D} for DT representation by jointly addressing spatial and temporal clues.

To validate the prominent performance of DoDGs on DTDB in comparison with several other methods, we also implement and evaluate the Gaussian-gradient-based descriptors (HoGFs [20]) using their best settings, i.e., $\text{HoGF}^{2D}_{\{\sigma=1\},\{2^{nd},3^{rd}\}}$ and $\text{HoGF}^{3D}_{\{\sigma=1\},\{3^{rd},4^{th}\}}$ with supporting regions $\{(P,R)\} = \{(8,1),(8,2)\}$. To the best of our knowledge, HoGFs currently are the best local descriptors for DT representation (see Table 9). In this work, two basic local operators, LBP-TOP [64] and CLBP [16] are also implemented in the same set of neighbors $\{(P,R)\} = \{(8,1)\}$ for objective evaluations in recognizing DTs on DTDB. Table 11 presents results of $\text{DoDGF}^{2D/3D}_{\sigma,\sigma',\mathcal{F}}$ utilizing the best settings discussed in Section 4.5. Also, those of the other LBP-based ones and learning-based methods are expressed in this table for a purpose of comprehensive comparison. It should be noted that rates of the learning-based methods are referred to the implementations of Hadji *et al.* [56].

It can be seen from Table 11 that our DoDG-based descriptors have performed very well in DT recognition on both *Dynamics* and *Appearance*. Those results are about 7∼9% better than those of the DoG-based ones. For instance, on *Dynamics*, $\text{DoGF}^{3D}_{(0.7,1)}$ just obtains

Table 10: Classification rates (%) of $\text{DoDGF}^{2D/3D}_{\sigma,\sigma',\mathcal{F}}$ on DTDB.

		Dynamics		Appearance	
Order(s)	(σ,σ')	DoDGF^{2D}	DoDGF^{3D}	DoDGF^{2D}	DoDGF^{3D}
1^{st}	$(0.5,0.7)$	65.81	69.61	66.31	68.74
	$(0.5,1)$	69.00	70.85	68.65	70.74
	$(0.7,1)$	68.03	70.52	68.54	70.94
	$(1,1.3)$	68.22	71.04	68.46	70.87
	$(1,1.5)$	68.38	69.05	68.51	69.18
2^{nd}	$(0.5,0.7)$	65.85	69.61	66.09	68.28
	$(0.5,1)$	67.06	70.08	66.58	69.32
	$(0.7,1)$	67.17	69.72	66.93	69.75
	$(1,1.3)$	67.37	70.47	66.62	69.13
	$(1,1.5)$	66.66	69.70	65.59	68.71
3^{rd}	$(0.5,0.7)$	65.47	68.79	66.04	68.33
	$(0.5,1)$	66.66	68.47	66.49	69.18
	$(0.7,1)$	66.14	69.61	66.08	69.70
	$(1,1.3)$	65.74	68.92	65.45	68.21
	$(1,1.5)$	66.91	69.03	67.19	68.17
4^{th}	$(0.5,0.7)$	64.97	68.29	64.20	67.65
	$(0.5,1)$	65.97	69.75	65.41	68.22
	$(0.7,1)$	65.49	68.78	65.33	68.05
	$(1,1.3)$	65.50	69.72	65.86	69.17
	$(1,1.5)$	65.89	69.84	67.01	68.58
$\{1^{st},2^{nd}\}$	$(0.5,0.7)$	67.86	70.25	67.06	69.97
	$(0.5,1)$	69.61	72.24	69.50	71.72
	$(0.7,1)$	69.81	72.06	69.84	72.10
	$(1,1.3)$	69.89	71.97	69.22	72.11
	$(1,1.5)$	69.47	71.87	69.46	72.08
$\{1^{st},3^{rd}\}$	$(0.5,0.7)$	66.97	69.49	66.49	68.93
	$(0.5,1)$	69.00	71.81	68.77	71.19
	$(0.7,1)$	69.85	71.49	68.51	71.60
	$(1,1.3)$	69.14	71.73	69.24	70.96
	$(1,1.5)$	69.03	71.44	68.22	71.26
$\{1^{st},4^{th}\}$	$(0.5,0.7)$	67.24	70.37	66.66	69.51
	$(0.5,1)$	69.70	72.01	69.40	71.74
	$(0.7,1)$	69.07	71.47	69.17	71.26
	$(1,1.3)$	69.54	72.36	69.57	71.68
	$(1,1.5)$	69.29	71.45	69.60	71.11
$\{2^{nd},3^{rd}\}$	$(0.5,0.7)$	67.16	70.20	67.21	69.50
	$(0.5,1)$	68.38	71.32	67.51	69.71
	$(0.7,1)$	67.90	70.42	67.86	70.45
	$(1,1.3)$	67.85	70.25	67.55	69.75
	$(1,1.5)$	68.14	70.08	67.76	69.53
$\{2^{nd},4^{th}\}$	$(0.5,0.7)$	66.62	70.11	65.98	68.22
	$(0.5,1)$	67.83	70.38	67.17	70.19
	$(0.7,1)$	67.85	69.68	67.21	70.20
	$(1,1.3)$	68.51	70.81	68.15	69.04
	$(1,1.5)$	67.72	69.54	67.32	69.26
$\{3^{rd},4^{th}\}$	$(0.5,0.7)$	66.49	69.54	66.78	68.99
	$(0.5,1)$	67.95	70.51	66.79	69.86
	$(0.7,1)$	66.99	68.96	66.90	69.89
	$(1,1.3)$	67.38	69.70	66.83	69.70
	$(1,1.5)$	67.15	70.35	66.17	68.60

rate of 65.07%, inferior to ∼7% compared to ours, i.e., $\text{DoDGF}^{3D}_{(0.7,1),\{1^{st},2^{nd}\}}$ with rate of 72.06%. In the meanwhile, in smaller dimension, our $\text{DoDGF}^{2D/3D}$ descriptors also have about 1% better than $\text{HoGF}^{2D/3D}$ [20]. Those have consolidated the prominent ability of DoDG filterings in noise reduction compared to the traditional DoGs and the Gaussian-gradient-based filterings. In terms of comparison to CLBP and LBP-TOP without addressing any filters in their encodings, our DoDG-based descriptors obtain about ∼12% and ∼24% higher than CLBP's [64] and LBP-TOP's [16] respectively (see Table 11). In the meantime, the $\text{DoGF}^{2D/3D}$ descriptors based on the well-known DoGs are also ∼5% and ∼17% better than CLBP's and LBP-TOP's respectively. This has proved the importance of filterings in noise reduction for DT representation, especially, the prominent contribution of our DoDGs.

Regarding comparison to the learning-based methods, in general, our DoDG-based descriptors have performance being very close to most of those methods, particularly, better than some of them. Indeed, with 72.10% on *Appearance*, our $\text{DoDGF}^{3D}_{(0.7,1),\{1^{st},2^{nd}\}}$ is about 8% better than deep-learning-based Flow Stream (64.80%) [58] while being as good as learning-based MSOE Stream [85]. For DT recognition on *Dynamics*, ours (72.06%) is the same execution as that of Flow Stream [58] while being very close to that of C3D (74.90%) [57] and RGB Stream (76.40%) [58] (see Table 11). Furthermore, it should be pointed out that SOE-Net [84] obtains the nearly highest rates on both schemes of DTDB, but not mean that it also has the same performance on other datasets. Certainly, all SOE-Net's performances on DynTex and DynTex++ are much lower than our DoDG-based descriptors. For instance, it could be seen from Table 9 that SOE-Net just obtains 96.70%, 95.70%, 92.20%, and 94.40% on *Alpha*, *Beta*, *Gamma*, and DynTex++ respectively. In the meanwhile, our $\text{DoDGF}^{3D}_{(0.7,1),\{1^{st},2^{nd}\}}$ is 100%, 98.15%, 96.97%, and 97.52% respectively. This has restated the interest of our proposal.

4.9. Further discussions

In addition to the thorough evaluations discussed in Section 4.5, it can be asserted the DoDG-based descriptors in

Table 11: Comparison of performances (%) on two challenging subsets of the large scale DTDB [56] dataset.

Group	Encoding method	$\{(P,R)\}$	Dynamics	Appearance
A	LBP-TOP^{u2} [64]	$\{(8,1)\}$	48.30	47.50
	CLBP$^{riu2}_{S/M/C}$ [16]	$\{(8,1)\}$	60.35	60.72
	HoGF$^{2D}_{\{\sigma=1\},\{2^{nd},3^{rd}\}}$ [20]	$\{(8,1),(8,2)\}$	69.38	69.56
	HoGF$^{3D}_{\{\sigma=1\},\{3^{rd},4^{th}\}}$ [20]	$\{(8,1),(8,2)\}$	71.08	71.03
	DoGF$^{2D}_{(0.7,1)}$	$\{(8,1)\}$	63.27	64.14
	DoGF$^{3D}_{(0.7,1)}$	$\{(8,1)\}$	65.07	65.11
	Our DoDGF$^{2D}_{(0.7,1),\{1^{st}\}}$	$\{(8,1)\}$	68.03	68.54
	Our DoDGF$^{2D}_{(0.7,1),\{1^{st},2^{nd}\}}$	$\{(8,1)\}$	69.81	69.84
	Our DoDGF$^{3D}_{(0.7,1),\{1^{st}\}}$	$\{(8,1)\}$	70.52	70.94
	Our DoDGF$^{3D}_{(0.7,1),\{1^{st},2^{nd}\}}$	$\{(8,1)\}$	72.06	72.10
B	MSOE Stream [85]	-	80.10	72.20
	SOE-Net [84]	-	**86.80**	79.00
	C3D [57]	-	74.90*	75.50*
	RGB Stream [58]	-	76.40*	76.10*
	Flow Stream [58]	-	72.60*	64.80*
	MSOE-two-Stream [56]	-	84.00*	**80.00***

Note: "-" means "not available". Superscript "*" expresses results using deep learning algorithms. Group A denotes *local-feature-based* methods, while B: *learning-based*. Results of above learning-based methods are referred to [56]. Results of HoGFs [20] on DTDB are reported by this work.

Table 12: Rates (%) of DoDGF$^{2D/3D}_{\{(\sigma,\sigma')\},\mathcal{F}}$ in further scale analysis.

	DoDG-based Descriptor	#bins	Beta	Gamma	DynTex++
(a)	DoDGF$^{2D}_{\{(0.7,1),(0.5,1)\},\{1^{st}\}}$	4800	95.06	95.45	97.02
	DoDGF$^{2D}_{\{(0.7,1),(1,1.3)\},\{1^{st}\}}$	4800	95.68	94.32	96.51
	DoDGF$^{2D}_{\{(0.5,1),(0.7,1),(1,1.3)\},\{1^{st}\}}$	7200	95.06	94.70	97.19
	DoDGF$^{3D}_{\{(0.7,1),(0.5,1)\},\{1^{st}\}}$	7200	**97.53**	96.21	97.19
	DoDGF$^{3D}_{\{(0.7,1),(1,1.3)\},\{1^{st}\}}$	7200	**97.53**	96.59	96.87
	DoDGF$^{3D}_{\{(0.5,1),(0.7,1),(1,1.3)\},\{1^{st}\}}$	10800	**97.53**	97.35	97.52
(b)	DoDGF$^{2D}_{(0.7,1),\{1^{st},2^{nd},3^{rd}\}}$	7200	96.91	95.08	97.09
	DoDGF$^{2D}_{(0.7,1),\{1^{st},2^{nd},3^{rd},4^{th}\}}$	9600	96.91	95.45	97.44
	DoDGF$^{3D}_{(0.7,1),\{1^{st},2^{nd},3^{rd}\}}$	10800	98.15	96.59	97.51
	DoDGF$^{3D}_{(0.7,1),\{1^{st},2^{nd},3^{rd},4^{th}\}}$	14400	**97.53**	96.97	97.53
(c)	DoDGF$^{2D}_{\{(0.7,1),(0.5,1)\},\{1^{st},2^{nd}\}}$	10800	96.30	95.45	97.27
	DoDGF$^{2D}_{\{(0.5,1),(0.7,1),(1,1.3)\},\{1^{st},2^{nd}\}}$	14400	95.68	95.08	97.56
	DoDGF$^{3D}_{\{(0.7,1),(0.5,1)\},\{1^{st},2^{nd}\}}$	14400	**97.53**	97.35	97.43
	DoDGF$^{3D}_{\{(0.5,1),(0.7,1),(1,1.3)\},\{1^{st},2^{nd}\}}$	21600	**97.53**	**97.73**	**97.81**

further contexts based on more experimental results as follows.

- The experimental results in Table 7 have verified that the 3D filtering is better than the 2D one in most cases. It may be deduced that addressing the higher directions of DoDG can improve the performance. In other words, addressing jointly shape and motion cues based on the 3D filtering is more effective than a separate consideration in the 2D one.
- Taking multi-scale analysis of $\{(\sigma,\sigma')\}$ into account the DT encoding does not make the DoDG-based descriptors more robust, except 97.35%, a little higher rate on *Gamma* of DoDGF$^{3D}_{\{(0.5,1),(0.7,1),(1,1.3)\},\{1^{st}\}}$ (see Table 12(a)).
- Also, addressing multi-scale of high-order DoDGs is not for further enhancement (see Table 12(b)).
- In addition, combining two kinds of above multi-scale analyses obtains a better rate of 97.73% on *Gamma* for DoDGF$^{3D}_{\{(\sigma,\sigma')\},\mathcal{F}}$, while facing with the cruse of larger dimension, up to 21600 bins, (see Table 12(c)).
- Taking odd and even orders of DoDG is recommended because their outcomes are more complementary.

Presently, deep-learning-based methods are going on the major stream for computer vision community. They often obtain significant results in DT recognition (see Tables 9 and 11). However, it takes much time for them to learn millions of parameters using complex learning algorithms in multi-deep-layer networks. For instance, it takes ∼80M for C3D [57], ∼88M for MSOE-two-Stream [56], while ∼61M for AlexNet and ∼6.8M for GoogleNet for DT-CNN [52]. This is one of crucial barriers in order to bring those into real applications for mobile devices as well as embedded sensor systems, those which strictly require tiny resources for their functions.

5. Conclusions

The simple and efficient DoDG kernel has been proposed to deal with the well-known problems of local DT representation. To take our DoDG into account video analysis, an adaptative framework has been presented for local DT representation, which is also available for different LBP-based encodings on robust DoDG-filtered outcomes. Just using a shallow analysis to exploit DoDG-filtered features, we have constructed discriminative DoDGF$^{2D/3D}_{\sigma,\sigma',\mathcal{F}}$ descriptors in slight dimension. Indeed, Tables 9 and 11 show the very good performances of our 2-order DoDGF$^{3D}_{(0.7,1),\{1^{st},2^{nd}\}}$ with 7200 bins as well as those of the single order DoDGF$^{2D}_{(0.7,1),\{1^{st}\}}$ with only 2400 bins on different datasets. Those can be easily applied to edge

devices, while maintaining a comparable performance related to deep learning models. For perspectives, instead of using CLBP [16], it is able to take other LBP-based operators into account our proposed framework for a purpose of further enhancement, e.g., CLBC [28], LDP-based [22, 21], LVP-based [77, 48], LRP [19], MRELBP [78], etc. In addition, analysis in multi-scale solutions of supporting regions (e.g., $\{(P,R)\} = \{(8,1),(8,2),(8,3)\}$) can be considered for these operators to investigate more extensively local relationships for further improvement.

Acknowledgment

We would like to express our sincere appreciation for the editors and reviewers, who pointed out the valuable and insightful remarks allowing us to clarify the presentation of this work. Also, we would like to send many thanks to those in Faculty of IT, HCMC University of Technology and Education, Thu Duc City, Ho Chi Minh City, Vietnam, who gave us enthusiastic support with high-performing computers for the experiments on the large scale datasets.

References

[1] G. Doretto, A. Chiuso, Y. N. Wu, S. Soatto, Dynamic textures, IJCV 51 (2) (2003) 91–109.

[2] X. Li, Human-robot interaction based on gesture and movement recognition, Signal Process. Image Commun. 81 (2020) 115686.

[3] W. Zhang, M. L. Smith, L. N. Smith, A. R. Farooq, Gender and gaze gesture recognition for human-computer interaction, CVIU 149 (2016) 32–50.

[4] X. S. Nguyen, T. P. Nguyen, F. Charpillet, N.-S. Vu, Local derivative pattern for action recognition in depth images, Multimedia Tools Appl 77 (7) (2018) 8531–8549.

[5] M. Deng, Robust human gesture recognition by leveraging multi-scale feature fusion, Signal Process. Image Commun. 83 (2020) 115768.

[6] A. I. Maqueda, C. R. del-Blanco, F. Jaureguizar, N. N. García, Human-computer interaction based on visual hand-gesture recognition using volumetric spatiograms of local binary patterns, CVIU 141 (2015) 126–137.

[7] T. P. Nguyen, A. Manzanera, M. Garrigues, N. Vu, Spatial motion patterns: Action models from semi-dense trajectories, IJPRAI 28 (07) (2014) 1460011.

[8] O. J. Makhura, J. C. Woods, Learn-select-track: An approach to multi-object tracking, Sig. Proc.: Image Comm. 74 (2019) 153–161.

[9] P. Barmpoutis, K. Dimitropoulos, N. Grammalidis, Smoke detection using spatio-temporal analysis, motion modeling and dynamic texture recognition, in: EUSIPCO, 2014, pp. 1078–1082.

[10] C. Zhang, F. Zhou, B. Xue, W. Xue, Stabilization of atmospheric turbulence-distorted video containing moving objects using the monogenic signal, Sig. Proc.: Image Comm. 63 (2018) 19–29.

[11] P. Mettes, R. T. Tan, R. C. Veltkamp, Water detection through spatio-temporal invariant descriptors, CVIU 154 (2017) 182–191.

[12] H. Sajid, S. S. Cheung, N. Jacobs, Motion and appearance based background subtraction for freely moving cameras, Sig. Proc.: Image Comm. 75 (2019) 11–21.

[13] D. Ortego, J. C. SanMiguel, J. M. Martínez, Stand-alone quality estimation of background subtraction algorithms, CVIU 162 (2017) 87–102.

[14] Z. Xu, B. Min, R. C. C. Cheung, A robust background initialization algorithm with superpixel motion detection, Sig. Proc.: Image Comm. 71 (2019) 1–12.

[15] Z. Zeng, J. Jia, Z. Zhu, D. Yu, Adaptive maintenance scheme for codebook-based dynamic background subtraction, CVIU 152 (2016) 58–66.

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

[17] T. T. Nguyen, T. P. Nguyen, F. Bouchara, Smooth-invariant gaussian features for dynamic texture recognition, in: ICIP, 2019, pp. 4400–4404.

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

[19] T. T. Nguyen, T. P. Nguyen, F. Bouchara, Rubik gaussian-based patterns for dynamic texture classification, Pattern Recognition Letters 135 (2020) 180–187.

[20] T. T. Nguyen, T. P. Nguyen, F. Bouchara, Prominent local representation for dynamic textures based on high-order gaussian-gradients, IEEE Trans. on Multimedia 23 (2021) 1367–1382.

[21] T. T. Nguyen, T. P. Nguyen, F. Bouchara, X. S. Nguyen, Momental directional patterns for dynamic texture recognition, CVIU 194 (2020) 102882.

[22] B. Zhang, Y. Gao, S. Zhao, J. Liu, Local derivative pattern

versus local binary pattern: Face recognition with high-order local pattern descriptor, IEEE Trans. IP 19 (2) (2010) 533–544.

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

[24] S. R. Arashloo, Sparse binarised statistical dynamic features for spatio-temporal texture analysis, Signal Image Video Process. 13 (3) (2019) 575–582.

[25] X. Zhao, Y. Lin, L. Liu, J. Heikkilä, W. Zheng, Dynamic texture classification using unsupervised 3d filter learning and local binary encoding, IEEE Trans. Multimedia 21 (7) (2019) 1694–1708.

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

[27] T. P. Nguyen, A. Manzanera, W. G. Kropatsch, X. S. N'Guyen, Topological attribute patterns for texture recognition, Pattern Recog. Letters 80 (2016) 91–97.

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

[29] P. Saisan, G. Doretto, Y. N. Wu, S. Soatto, Dynamic texture recognition, in: CVPR, 2001, pp. 58–63.

[30] A. B. Chan, N. Vasconcelos, Classifying video with kernel dynamic textures, in: CVPR, 2007, pp. 1–6.

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

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

[33] A. Ravichandran, R. Chaudhry, R. Vidal, View-invariant dynamic texture recognition using a bag of dynamical systems, in: CVPR, 2009, pp. 1651–1657.

[34] Y. Qiao, L. Weng, Hidden markov model based dynamic texture classification, IEEE Signal Process. Lett. 22 (4) (2015) 509–512.

[35] Y. Qiao, Z. Xing, Dynamic texture classification using multivariate hidden markov model, IEICE Trans. Fundam. Electron. Commun. Comput. Sci. 101-A (1) (2018) 302–305.

[36] Y. Xu, Y. Quan, H. Ling, H. Ji, Dynamic texture classification using dynamic fractal analysis, in: ICCV, 2011, pp. 1219–1226.

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

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

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

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

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

[42] B. Ghanem, N. Ahuja, Maximum margin distance learning for dynamic texture recognition, in: ECCV, 2010, pp. 223–236.

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

[44] R. Péteri, D. Chetverikov, Qualitative characterization of dynamic textures for video retrieval, in: ICCVG, 2004, pp. 33–38.

[45] R. Péteri, D. Chetverikov, Dynamic texture recognition using normal flow and texture regularity, in: IbPRIA, 2005, pp. 223–230.

[46] Z. Lu, W. Xie, J. Pei, J. Huang, Dynamic texture recognition by spatio-temporal multiresolution histograms, in: WACV/MOTION, 2005, pp. 241–246.

[47] T. T. Nguyen, T. P. Nguyen, F. Bouchara, X. S. Nguyen, Directional beams of dense trajectories for dynamic texture recognition, in: ACIVS, 2018, pp. 74–86.

[48] T. T. Nguyen, T. P. Nguyen, F. Bouchara, Directional dense-trajectory-based patterns for dynamic texture recognition, IET Computer Vision 14 (4) (2020) 162–176.

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

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

[51] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: NIPS, 2012, pp. 1106–1114.

[52] V. Andrearczyk, P. F. Whelan, Convolutional neural network on three orthogonal planes for dynamic texture classification, Pattern Recognition 76 (2018) 36 – 49.

[53] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: CVPR, 2015, pp. 1–9.

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

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

[56] I. Hadji, R. P. Wildes, A new large scale dynamic texture dataset with application to convnet understanding, in: ECCV, 2018, pp. 334–351.

[57] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, M. Paluri,

Learning spatiotemporal features with 3d convolutional networks, in: ICCV, 2015, pp. 4489–4497.

[58] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: NIPS, 2014, pp. 568–576.

[59] Y. Quan, Y. Huang, H. Ji, Dynamic texture recognition via orthogonal tensor dictionary learning, in: ICCV, 2015, pp. 73–81.

[60] Y. Quan, C. Bao, H. Ji, Equiangular kernel dictionary learning with applications to dynamic texture analysis, in: CVPR, 2016, pp. 308–316.

[61] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, L. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in: CVPR, 2018, pp. 4510–4520.

[62] A. Howard, R. Pang, H. Adam, Q. V. Le, M. Sandler, B. Chen, W. Wang, L. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, Searching for mobilenetv3, in: ICCV, 2019, pp. 1314–1324.

[63] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, Q. Tian, Centernet: Keypoint triplets for object detection, in: ICCV, 2019, pp. 6568–6577.

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

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

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

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

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

[69] X. Zhao, Y. Lin, J. Heikkilä, 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.

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

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

[72] N. Vu, T. P. Nguyen, C. Garcia, Improving texture categorization with biologically-inspired filtering, Image Vision Comput. 32 (6-7) (2014) 424–436.

[73] T. T. Nguyen, T. P. Nguyen, F. Bouchara, Completed local structure patterns on three orthogonal planes for dynamic texture recognition, in: IPTA, 2017, pp. 1–6.

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

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

[76] T. T. Nguyen, T. P. Nguyen, F. Bouchara, Dynamic texture representation based on hierarchical local patterns, in: ACIVS, 2020, pp. 277–289.

[77] K. Fan, T. Hung, A novel local pattern descriptor - local vector pattern in high-order derivative space for face recognition, IEEE Trans. IP 23 (7) (2014) 2877–2891.

[78] L. Liu, S. Lao, P. W. Fieguth, Y. Guo, X. Wang, M. Pietikäinen, Median robust extended local binary pattern for texture classification, IEEE Trans. IP 25 (3) (2016) 1368–1381.

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

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

[81] J. Ren, X. Jiang, J. Yuan, G. Wang, Optimizing LBP structure for visual recognition using binary quadratic programming, IEEE Signal Process. Lett. 21 (11) (2014) 1346–1350.

[82] J. Xie, Y. Fang, Dynamic texture recognition with video set based collaborative representation, Image and Vision Computing 55 (2016) 86–92.

[83] J. Ren, X. Jiang, J. Yuan, Dynamic texture recognition using enhanced LBP features, in: ICASSP, 2013, pp. 2400–2404.

[84] I. Hadji, R. P. Wildes, A spatiotemporal oriented energy network for dynamic texture recognition, in: ICCV, 2017, pp. 3085–3093.

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