Dynamic texture representation based on oriented magnitudes of Gaussian gradients

Thanh Tuan Nguyen, Thanh Phuong Nguyen, Frédéric Bouchara

Journal of Visual Communication and Image Representation (2021)

Contents lists available at ScienceDirect

Journal of Visual Communication and Image Representation

journal homepage: www.elsevier.com/locate/jvci

Dynamic texture representation based on oriented magnitudes of Gaussian gradients

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*

ARTICLE INFO

Article history:
Received xx xx 2020
Received in final form xx xx 2021
Accepted xx xx 2021
Available online xx xx 2021

Keywords: Dynamic textures, Gaussian–filtered derivatives, Oriented magnitudes, LBP, CLBP, Video representation

ABSTRACT

Efficiently capturing shape and turbulent motions of dynamic textures (DTs) for video description is a challenge in real applications due to the negative influences of the well-known problems: environmental elements, illumination, scale, and noise. In this paper, we propose an efficient and simple framework for DT representation based on oriented features of high-order Gaussian gradients. Firstly, 2D/3D Gaussian-based filtering kernels in high-order partial derivatives are taken into account video analysis as a preprocessing to obtain corresponding gradient-filtered images/volumes. After that, oriented features, which are robust against above issues, are extracted by decomposing the Gaussian derivative magnitudes into oriented components. Finally, a shallow local encoding is utilized for structuring spatio-temporal features from these oriented magnitudes. This allows to construct discriminative descriptors with promising performances compared to those based on the non-oriented ones. Experimental results for DT classification task on benchmark datasets have verified the interest of our proposal.

1. Introduction

Dynamic textures (DTs) are textural characteristics repeated in a temporal domain. Understanding them in effect is one of crucial issues in many applications of computer vision, such as human interaction [1, 2, 3], detection and object tracking [4, 5], background subtraction [6, 7], crowded people [8, 9], etc. Due to the turbulent motions along with impacts of environmental changes and illumination, efficiently capturing dynamic characteristics is a major challenge for DT representation. In order to deal with those problems, many techniques have been proposed and they can be categorized into the following groups.

Geometry-based methods: Based on fractal analysis, geometry-based methods attempt to deal with the influence of environmental changes. Dynamic Fractal Spectrum (DFS) [10] and its variant (Multi-Fractal Spectrum (MFS) [11]) were introduced in order to take advantage of stochastic self-similarities and fractal features for DT representation. After that, Ji *et al.* [12] adapted the MFS model using wavelet coefficients to construct Wavelet-based MFS (WMFS) descriptor with more effectiveness. In addition, Quan *et al.* [13] proposed a technique of lacunarity analysis, named Spatio-Temporal Lacunarity Spectrum (STLS), to extract lacunarity-based patterns for DT description. Baktashmotlagh *et al.* [14] introduced Stationary Subspace Analysis (SSA) to extract stationary components in videos for DT encoding. In respect of DT recognition, experimental results have validated that the geometry-based methods seem to be adaptive for recognizing DTs with simple motions (e.g., those in UCLA [15] dataset), while being difficult to understand complex DTs (e.g., those in DynTex [16] and DynTex++ [17] datasets). It may be due to a lack of the temporal information addressed in these analyses.

Optical-flow-based methods: Magnitudes and directions of the normal flow were taken into account video analysis in natural ways [18, 19, 20]. However, a supposition of local smooth-

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

ness and brightness in stability can be a limitation for dealing with the chaotic motions of DTs in videos [21]. Besides, dense trajectories and angles of DT motions were addressed in [22] to capture directional dynamic features in various directions of motion points. Lu *et al.* [23] exploited characteristics of velocity and acceleration in multi-resolution analysis to structure probability distributions for DT description. Nevertheless, just motion components were considered in [22] and [23], lack of textural appearance information for encoding spatial features.

Model-based methods: Most of them are based on Linear Dynamical System (LDS) [15] and its variants to address turbulent dynamic properties of DTs in videos. *Kernel*-PCA was exploited to adapt the LDS's observation in order to handle DTs with complex motions [24]. Chan *et al.* [25] introduced a model of DT mixtures (DTM) for addressing characteristics of movable objects in videos. The outcomes were then arranged into k clusters by employing a method of hierarchical expectation-maximization (HEM-DTM). In addition, other efforts also focused on the LDS's concept to be in accordance with analyzing DT features: bag-of-words (BoW) [26], bag-of-systems (BoS) [27], and BoS Tree [28]. With respect to their ability of classifying DTs, the model-based techniques have achieved moderate results due to without regard to dynamic features, one of important information for DT description [15]. Furthermore, the processes in constructing the models can become more complicated if the dynamic properties are taken into account [27].

Learning-based methods: It can group them into two main kinds of approaches as follows. Deep learning techniques often utilize Convolutional Neural Networks (CNNs) to learn DT features in several ways. Such methods are Transferred ConvNet Features (TCoF) [29] - learning deep structures in still images; DT-CNN [30] and PCANet-TOP [31] - learning DT features based on three orthogonal planes of sequences; D3 [32] - concentrating on "key frames" and "key segments" of videos to extract static and dynamic patterns. Although the deep-learning methods achieved outstanding results in classifying DTs, they addressed tremendous parameters for the learning processes with high complexity of net-computing algorithms. This leads to a strict barrier for mobile applications in practice. The remain group learns dictionaries of DT features based on kernel sparse coding. Quan *et al.* [33] introduced a dictionary learned from atoms of sequences. In the meantime, an equiangular kernel was proposed in [34] to build a dictionary in reasonable dimension. Like the geometry-based approaches, the dictionary-based methods have arduously faced with "understanding" complex dynamic properties of DTs in DynTex [16] and DynTex++ [17].

Filter-based methods: Thanks to robustness against changes of environmental elements, illumination and noise, filter-based methods have achieved potential results of DT recognition. Arashloo *et al.* [35] proposed to employ filters learned by transformation of ICA (independent component analysis). They then extracted Multi-scale Binarized Statistical Image Features based on three orthogonal planes of sequences (MBSIF-TOP). In the meanwhile, Zhao *et al.* [36] utilized CLBP [37] (Completed Local Binary Pattern) to capture spatio-temporal characteristics from $3D$ filtered volumes. Therein, the $3D$ filters were learned from various unsupervised techniques: PCA (Principal Component Analysis), ICA, sparse filtering, and k-means clustering. Recently, Nguyen *et al.* [38, 39, 40] addressed filtering methods as a pre-processing to mitigate the negative impacts of environmental changes, illumination noise on DT encoding: moment image model [38] - a filtering technique based on pre-defined supporting regions; Gaussian-based filtering kernels [39, 40]. Experimental results for DT classification have validated that the filter-based methods seem to work well for describing simple motions rather than for complicated ones.

Local-feature-based methods: Most of them are based on Local Binary Pattern (LBP) [41] and its completed model (CLBP [37]) to encode shape and motion clues for DT representation. Zhao *et al.* [42] introduced two approaches taking advantage of LBP to investigate local relationships in spatio-temporal domain of video analysis as follows. For a video, Volume-LBP (VLBP) patterns were formed by using LBP on its three consecutive frames while LBP-TOP patterns were computed by exploiting LBP on its three orthogonal planes. Motivated by VLBP and LBP-TOP, many efforts have addressed LBP's conventional limitations in order to enhance the discrimination power: rotation-invariant problems [43], sensitivity to noise, and near-uniform regions [44, 45, 46, 38]. Furthermore, Ren *et al.* [47] proposed data-driven LBP (DDLBP) features to deal with problems of grand dimension, while PCA was involved in local encoding to eliminate noise features [48].

In spite of having promising performances, the local-feature-based techniques [44, 45, 46, 38] have yet encountered with the well-known issues of DT representation: environmental elements, illumination, and noise. To mitigate those problems, Nguyen *et al.* [39, 40, 49] used Gaussian-based filtering kernels for denosing before encoding local spatio-temporal features of DTs. In other aspects, Nguyen *et al.* [38, 50] introduced local DT features structured from moment-filtered outcomes, while other efforts [35, 36] addressed the learning-based filtered features. The abilities of the achieved descriptors for DT classification task are encouraging in comparison with other local-feature-based attempts. However, the conventional limitations seem not to be thoroughly dealt with. To this end, we propose in this work a novel approach for exploiting oriented local features containing rich textural information. An efficient and simple framework is then introduced for DT representation based on local patterns of high-order oriented magnitudes. This allows to efficiently reduce the negative impacts of above problems on capturing shape and turbulent motions of DTs in local regions. Contrary to the complicated models of deep-learning-based methods, our proposal can have competitive performances by just using the shallow analysis for encoding spatio-temporal oriented magnitudes.

Generally, our proposed framework takes the following steps for DT description in effect. Firstly, k-order gradients of 2D (resp. 3D) Gaussian filtering kernels are taken into account video analysis as a preprocessing to obtain corresponding gradient-filtered images $\mathcal{I}_\sigma^{\partial x_i^k}$ (resp. volumes $\mathcal{V}_\sigma^{\partial x_i^k}$). Magnitudes $\|\nabla\mathcal{I}_\sigma\|$ (resp. $\|\nabla\mathcal{V}_\sigma\|$) and gradient directions θ_σ (resp. ϕ_σ) are then referred from these Gaussian-gradient filterings. After that, k-order oriented magnitudes IOM_σ^k (resp. VOM_σ^k)

are pointed out by decomposing the magnitudes $\|\nabla\mathcal{I}_\sigma\|$ (resp. $\|\nabla\mathcal{V}_\sigma\|$) with respect to different ranges of gradient directions θ_σ and ϕ_σ. Finally, a simple local operator (e.g., CLBP [37]) is utilized for extracting spatio-temporal features from the IOM/VOM-based outcomes. As a result, robust descriptors are then structured with high performances in reasonable dimensions. Experiments for classifying DTs on benchmark datasets have verified the considerable ability of our proposed descriptors in comparison with state of the art. In short, our major contributions can be listed as follows.

- To the best of our knowledge, it is the first time that the oriented features of high-order Gaussian-gradient magnitudes are exploited to make DT representation more robust against above typical issues.
- Multi-order of gradients and multi-scale analysis of Gaussian-based filtering are also addressed to forcefully investigate benefits of informative magnitudes.
- A modified soft-assignment is introduced to efficiently quantize the Gaussian-gradient magnitudes subject to a pre-defined orientation in comprehensive comparison with the traditional quantification models of oriented features.
- Based on the oriented features of Gaussian gradients, discriminative IOM/VOM-based descriptors are structured by just using a simple operator. Furthermore, our experiments have also proved the advantages of the oriented magnitudes for DT representation using LBP-based variants compared to non-oriented ones involved in.
- In reasonable dimension, our proposed descriptors perform well in comparison with all non-deep-learning approaches, while being very close to performances of deep-learning models.

2. Related works

2.1. A brief of LBP and its completed model

Let $\mathcal{I}$ denote a gray-scale textural image. In consideration of relationships between a center pixel $\mathbf{q} \in \mathcal{I}$ and its local neighbors, Ojala *et al.* [41] introduced a LBP pattern as a binary string by measuring differences of their intensities as

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

where $\mathcal{I}(.)$ points out the gray-value of a pixel; $\{\mathbf{p}_i\}_{i=1}^{P}$ ($P \in \mathbb{Z}^+$) is a collection of P neighbors which are interpolated by a circle sample with center $\mathbf{q}$ and radius R; and $s(.)$ is defined as

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

As a result, a histogram of 2^P bins is structured to describe a textural image. This leads to a restriction for real implementations. In practice, two following mapping techniques are usually addressed to overcome this curse of dimension: $u2$ mapping with $P(P-1)+3$ bins for uniform patterns, $riu2$ mapping with $P+2$ bins for rotation-invariant uniform patterns. Furthermore, other mappings can be also considered: $TAP^{\mathcal{A}}$ mapping [51] for topological features, LBC [52] - an alternative of $riu2$.

In order to conduct the LBP encoding in diversity, Guo *et al.* [37] presented completed model of LBP (CLBP). Principally, CLBP consists of three components integrated into various ways to enhance the performance: CLBP_S is identical to the typical LBP; CLBP_M captures informative magnitudes; and CLBP_C measures the global gray-differences of center pixels. In practice, two following integration types are often used due to their better performances: "S_M/C" means jointing CLBP_M and CLBP_C patterns before concatenating with CLBP_S; "S/M/C" denotes a 3D joint of those components.

2.2. Gaussian filtering kernel and its derivatives

A well-known Gaussian filtering is a process of convolving a μ-dimensional Gaussian kernel on a spatial domain. Its results agree with the Gaussian distribution. The Gaussian filtering kernel is defined in general as

$$\mathrm{G}_\sigma^\mu(\gamma_\mu) = \frac{1}{(\sigma\sqrt{2\pi})^\mu}\exp\Big(-\frac{x_1^2 + x_2^2 + ... + x_\mu^2}{2\sigma^2}\Big) \tag{3}$$

in which $\gamma_\mu = \{x_i\}_{i=1}^{\mu}$ denotes a collection of μ spatial directions, σ is a pre-defined standard deviation. Appropriately, a k-order ($k \in \mathbb{Z}^+$) partial derivative of $\mathrm{G}_\sigma^\mu(\gamma_\mu)$ is calculated with respect to a direction $x_i \in \gamma_\mu$ as

$$\mathrm{G}_{\sigma,\partial x_i^k}^\mu(\gamma_\mu) = \frac{\partial^k \mathrm{G}_\sigma^\mu(\gamma_\mu)}{\partial x_i^k} \tag{4}$$

in which "∂" denotes a gradient operation.

2.3. Exploiting oriented features

Oriented features play an important role in representation of local features. Gabor filter [53] has been early used to extract oriented features in textural analysis. Dalal and Triggs [54] presented histograms of oriented gradient on each local patch to form HoG descriptor for pedestrian detection. Inspired by this well-known descriptor, many other works have been introduced to deal with different problems. The oriented features have been also utilized for key-point description in different detectors, e.g., SIFT detector [55]. Exploiting this kind of features from local patch around keypoints leads to a powerful description of detected keypoints making these detectors be effective in various applications of computer vision in the years of 2000s.

3. Proposed method

In our prior works [39, 40, 49], we have indicated that taking Gaussian-based filtering kernels into account DT representation could improve the discrimination power of local DT encoding. This is thanks to mitigating the negative impacts the typical problems on DT encoding. However, the achieved improvements are still at a moderate level since those problems may not be dealt with thoroughly. Instead of exploiting Gaussian-based filtered features as in [39, 40], this work is motivated by HoG descriptor [54] where oriented information has been successfully exploited for local feature representation. We propose

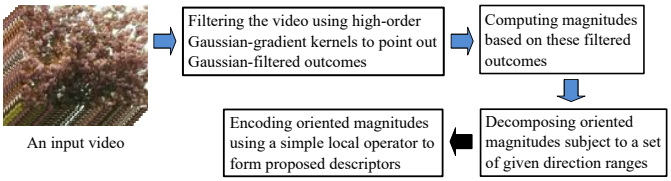

Fig. 1. (Best viewed in color) A proposed framework for encoding a video in general. Therein, the blue arrows denote progresses of pre-processing, the black one denotes progress of encoding features of oriented magnitudes.

an efficient framework for DT representation based on high-order oriented magnitudes that are decomposed from Gaussian-gradient outcomes, as graphically illustrated in Fig. 1. Accordingly, high-order Gaussian-gradient kernels are used to filter a given video for noise reduction. Magnitude features are then extracted from the gradient-filtered outcomes. Different decomposing models are then addressed to separate these obtained magnitude features into oriented magnitudes subject to a given orientation range (see Section 3.1). Finally, robust descriptors are structured by using a simple local operator to encode the oriented magnitudes (see Section 3.2). Experiments for DT classification have validated the good performance of oriented magnitudes compared to Gaussian-based filtered features in [39, 40] (see Section 4.5). Hereafter, we express above proposed processes in detail.

3.1. Oriented magnitudes of Gaussian gradients

In order to compute Gaussian-oriented magnitudes, we conduct the kernel $\mathrm{G}^{\mu}_{\sigma,\partial x^k_i}$ in 2D and 3D filtering dimensions, i.e., $\mathrm{G}^{2D/3D}_{\sigma,\partial x^k_i}$. Appropriately, for a given image $\mathcal{I}$, a pixel $\mathbf{q} \in \mathcal{I}$ is filtered by the 2D filtering kernel with respect to spatial coordinates (x, y) as

$$\begin{cases} \mathcal{I}^{\partial x^k}_{\sigma}(\mathbf{q}) = \mathrm{G}^{2D}_{\sigma,\partial x^k}(x,y) * \mathcal{I}(\mathbf{q}) \\ \mathcal{I}^{\partial y^k}_{\sigma}(\mathbf{q}) = \mathrm{G}^{2D}_{\sigma,\partial y^k}(x,y) * \mathcal{I}(\mathbf{q}) \end{cases} \tag{5}$$

in which "*" denotes a convolving operator; $\mathcal{I}^{\partial x^k}_{\sigma}$ and $\mathcal{I}^{\partial y^k}_{\sigma}$ are k-order Gaussian-filtered images. Similarly, for a given video $\mathcal{V}$, a voxel $\mathbf{u} \in \mathcal{V}$ is filtered by the 3D filtering kernel with respect to spatial coordinates (x, y) and temporal direction z as

$$\begin{cases} \mathcal{V}^{\partial x^k}_{\sigma}(\mathbf{u}) = \mathrm{G}^{3D}_{\sigma,\partial x^k}(x,y,z) * \mathcal{V}(\mathbf{u}) \\ \mathcal{V}^{\partial y^k}_{\sigma}(\mathbf{u}) = \mathrm{G}^{3D}_{\sigma,\partial y^k}(x,y,z) * \mathcal{V}(\mathbf{u}) \\ \mathcal{V}^{\partial z^k}_{\sigma}(\mathbf{u}) = \mathrm{G}^{3D}_{\sigma,\partial z^k}(x,y,z) * \mathcal{V}(\mathbf{u}) \end{cases} \tag{6}$$

where $\mathcal{V}^{\partial x^k}_{\sigma}$, $\mathcal{V}^{\partial y^k}_{\sigma}$, and $\mathcal{V}^{\partial z^k}_{\sigma}$ are k-order filtered volumes.

Based on above k-order Gaussian-filtered images/volumes, we correspondingly propose 2D/3D oriented magnitudes which are decomposed subject to a direction range. In order to thoroughly investigate the influences of the decomposing process, the following quantification strategies are addressed as

Quantification strategies: In consideration of an uniform quantification of an oriented feature f, which is defined at an arbitrary pixel $\mathbf{q}$ as $f(\mathbf{q})$, into n bins, it can be decomposed into two components: orientation $\overline{f}(\mathbf{q}) \in [0, 2\pi)$ and magnitude $\|f(\mathbf{q})\|$. Let us suppose that $(i-1)\lambda \leq \overline{f}(\mathbf{q}) < i\lambda$, where $i \in \{1, 2, .., n\}$ and $\lambda = \frac{2\pi}{n}$. We investigate hereunder 3 following quantification modes for decomposition of an image of oriented features. The two first modes are often used in the literature of feature quatification, while the last one is our proposal for this task. Traditional methods address two possible strategies for decomposition of f into n images of oriented features: $\{m_i\}_{i=1}^n$.

- *Hard assignment:* $f(\mathbf{q})$ is totally assigned to pixel $\mathbf{q}$ of image m_i with value $\|f(\mathbf{q})\|$. It means as

$$\begin{cases} m_i(\mathbf{q}) = \|f(\mathbf{q})\| \\ m_j(\mathbf{q}) = 0 \qquad \forall j \neq i \end{cases} \tag{7}$$

- *Soft assignment:* $f(\mathbf{q})$ is partially assigned to pixel $\mathbf{q}$ of image m_i with value $\frac{i\lambda - \overline{f}(\mathbf{q})}{\lambda}\|f(\mathbf{q})\|$ and to pixel $\mathbf{q}$ of image m_{i+1} with value $\frac{\overline{f}(\mathbf{q}) - (i-1)\lambda}{\lambda}\|f(\mathbf{q})\|$, where $m_{n+1} \equiv m_1$. It means as

$$\begin{cases} m_i(\mathbf{q}) = \frac{i\lambda - \overline{f}(\mathbf{q})}{\lambda}\|f(\mathbf{q})\| \\ m_{i+1}(\mathbf{q}) = \frac{\overline{f}(\mathbf{q}) - (i-1)\lambda}{\lambda}\|f(\mathbf{q})\| \\ m_j(\mathbf{q}) = 0, \text{ where } j \notin \{i, i+1\} \end{cases} \tag{8}$$

We introduce in this work an another version of soft assignment, called modified soft assignment, which allows to quantize $f(\mathbf{q})$ into $2n$ bins $\{m_i^+, m_i^-\}_{i=1}^n$ as follows.

- *Modified soft assignment:* $f(\mathbf{q})$ is partially assigned to pixel $\mathbf{q}$ of image m_i^+ with value $\frac{i\lambda - \overline{f}(\mathbf{q})}{\lambda}\|f(\mathbf{q})\|$ and to pixel $\mathbf{q}$ of image m_{i+1}^- with value $\frac{\overline{f}(\mathbf{q}) - (i-1)\lambda}{\lambda}\|f(\mathbf{q})\|$, where $m_{n+1}^- \equiv m_1^-$. It means as

$$\begin{cases} m_i^+(\mathbf{q}) = \frac{i\lambda - \overline{f}(\mathbf{q})}{\lambda}\|f(\mathbf{q})\| \\ m_j^+(\mathbf{q}) = 0 \qquad \forall j \neq i \\ m_{i+1}^-(\mathbf{q}) = \frac{\overline{f}(\mathbf{q}) - (i-1)\lambda}{\lambda}\|f(\mathbf{q})\| \\ m_j^-(\mathbf{q}) = 0 \qquad \forall j \neq i+1 \end{cases} \tag{9}$$

The main difference between the soft assignment and our modified model is that for n ranges of orientations, the first one produces n bins while the second one generates $2n$ bins. In other words, each bin m_i in the typical approach is now separated into 2 components (m_i^+ and m_i^-) to express the quantized feature with more discriminative power in the new approach[1].

Decomposition of gradient-filtered images $\mathcal{I}^{\partial x^k}_{\sigma}$ and $\mathcal{I}^{\partial y^k}_{\sigma}$: Following the quantification strategies presented in the previous section, we introduce herefater the decomposition of gradient-filtered images. The high-order oriented magnitude of a pixel $\mathbf{q} \in \mathcal{I}$ is determined so that its gradient direction is agreed with a given range of direction $d = [\alpha, \beta) = [(i-1)\lambda, i\lambda)$, where $\lambda = \frac{2\pi}{n}$, $\alpha = (i-1)\lambda$, and $\beta = i\lambda$, $i \in \{1, 2, .., n\}$. Let us suppose that $\theta^{\partial x^k, \partial y^k}_{\sigma}(\mathbf{q}) \in d$.

[1] A simple MATLAB code of our modified soft assignment to decompose high-order 2D/3D Gaussian gradients subject to a pre-defined orientation range is available at `http://tpnguyen.univ-tln.fr/download/MATCodeIVOM`

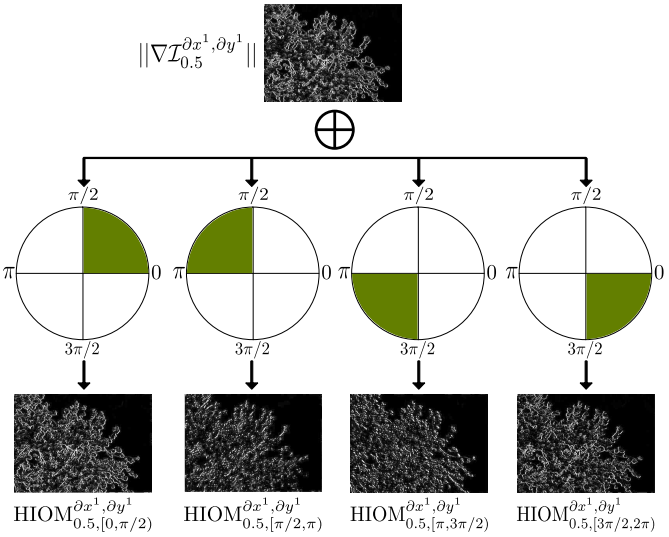

Fig. 2. (Best viewed in color) A hard-assignment model for decomposing the magnitudes of two Gaussian-gradient images $\mathcal{I}_{0.5}^{\partial x^1}$ and $\mathcal{I}_{0.5}^{\partial y^1}$ into 4 HIOM images subject to a set of 4 equal ranges of direction $\mathcal{D}^4 = \{[0,\pi/2),[\pi/2,\pi),[\pi,3\pi/2),[3\pi/2,2\pi)\}$.

Accordingly, a feature of Image of Oriented Magnitudes (IOM) could be quantified by the hard-assignment principle (also see Eq. 7) as

$$\text{HIOM}_{\sigma,i}^{\partial x^k,\partial y^k}(\mathbf{q}) = \|\nabla \mathcal{I}_{\sigma}^{\partial x^k,\partial y^k}(\mathbf{q})\|, \text{so that } \theta_{\sigma}^{\partial x^k,\partial y^k}(\mathbf{q}) \in d \quad (10)$$

by the soft-assignment (also see Eq. 8) as

$$\begin{cases} \text{SIOM}_{\sigma,i}^{\partial x^k,\partial y^k}(\mathbf{q}) = \|\nabla \mathcal{I}_{\sigma}^{\partial x^k,\partial y^k}(\mathbf{q})\| \times \frac{\beta - \theta_{\sigma}^{\partial x^k,\partial y^k}(\mathbf{q})}{\beta-\alpha} \\ \text{SIOM}_{\sigma,i+1}^{\partial x^k,\partial y^k}(\mathbf{q}) = \|\nabla \mathcal{I}_{\sigma}^{\partial x^k,\partial y^k}(\mathbf{q})\| \times \frac{\theta_{\sigma}^{\partial x^k,\partial y^k}(\mathbf{q}) - \alpha}{\beta-\alpha} \end{cases} \quad (11)$$

and by the modified soft-assignment (also see Eq. 9) as

$$\begin{cases} \text{pMSIOM}_{\sigma,i}^{\partial x^k,\partial y^k}(\mathbf{q}) = \|\nabla \mathcal{I}_{\sigma}^{\partial x^k,\partial y^k}(\mathbf{q})\| \times \frac{\beta - \theta_{\sigma}^{\partial x^k,\partial y^k}(\mathbf{q})}{\beta-\alpha} \\ \text{nMSIOM}_{\sigma,i+1}^{\partial x^k,\partial y^k}(\mathbf{q}) = \|\nabla \mathcal{I}_{\sigma}^{\partial x^k,\partial y^k}(\mathbf{q})\| \times \frac{\theta_{\sigma}^{\partial x^k,\partial y^k}(\mathbf{q}) - \alpha}{\beta-\alpha} \end{cases} \quad (12)$$

where $\text{SIOM}_{\sigma,n+1}^{\partial x^k,\partial y^k}(\mathbf{q}) \equiv \text{SIOM}_{\sigma,1}^{\partial x^k,\partial y^k}(\mathbf{q})$, $\text{nMSIOM}_{\sigma,n+1}^{\partial x^k,\partial y^k}(\mathbf{q}) \equiv \text{nMSIOM}_{\sigma,1}^{\partial x^k,\partial y^k}(\mathbf{q})$, and $\|\nabla \mathcal{I}_{\sigma}^{\partial x^k,\partial y^k}(\mathbf{q})\|$ denotes the k-order magnitude information of $\mathbf{q}$ and is calculated as follows.

$$\|\nabla \mathcal{I}_{\sigma}^{\partial x^k,\partial y^k}(\mathbf{q})\| = \sqrt{\left(\mathcal{I}_{\sigma}^{x^k}(\mathbf{q})\right)^2 + \left(\mathcal{I}_{\sigma}^{y^k}(\mathbf{q})\right)^2} \quad (13)$$

In the meanwhile, $\theta_{\sigma}^{\partial x^k,\partial y^k}(\mathbf{q})$ denotes the gradient direction of pixel $\mathbf{q}$ and is inferred as

$$\theta_{\sigma}^{\partial x^k,\partial y^k}(\mathbf{q}) = \arctan\left(\mathcal{I}_{\sigma}^{\partial y^k}(\mathbf{q}) / \mathcal{I}_{\sigma}^{\partial x^k}(\mathbf{q})\right) \quad (14)$$

Let us consider an intuitive example of decomposition in Fig. 2 which graphically illustrates an instance of decomposing the magnitudes of two Gaussian-gradient images $\mathcal{I}_{0.5}^{\partial x^1}$ and $\mathcal{I}_{0.5}^{\partial y^1}$ in order to obtain 4 HIOM images subject to a set of 4 equal ranges of direction $\mathcal{D}^4 = \{[0,\pi/2),[\pi/2,\pi),[\pi,3\pi/2),[3\pi/2,2\pi)\}$.

Decomposition of gradient-filtered volumes $\mathcal{V}_{\sigma}^{\partial x^k}$, $\mathcal{V}_{\sigma}^{\partial y^k}$, and $\mathcal{V}_{\sigma}^{\partial z^k}$: The high-order oriented magnitudes of a voxel $\mathbf{u} \in \mathcal{V}$ are addressed subject to its pairs of gradient directions being in accordance with the pre-defined direction range $d = [\alpha,\beta)$. Therein, $\lambda = \frac{2\pi}{n}$; $\alpha = (i-1)\lambda$ and $\beta = i\lambda$ are two extremities of d, $i \in \{1,2,..,n\}$. Without loss of generality, let us suppose that $\phi_{\sigma}^{\partial x^k,\partial y^k}(\mathbf{u}) \in d$ (similarly for two other cases: $\phi_{\sigma}^{\partial y^k,\partial z^k}(\mathbf{u}) \in d$ and $\phi_{\sigma}^{\partial z^k,\partial x^k}(\mathbf{u}) \in d$). Accordingly, a feature of Volumes of Oriented Magnitudes (VOM) could be quantified to a bin by the hard assignment principle as

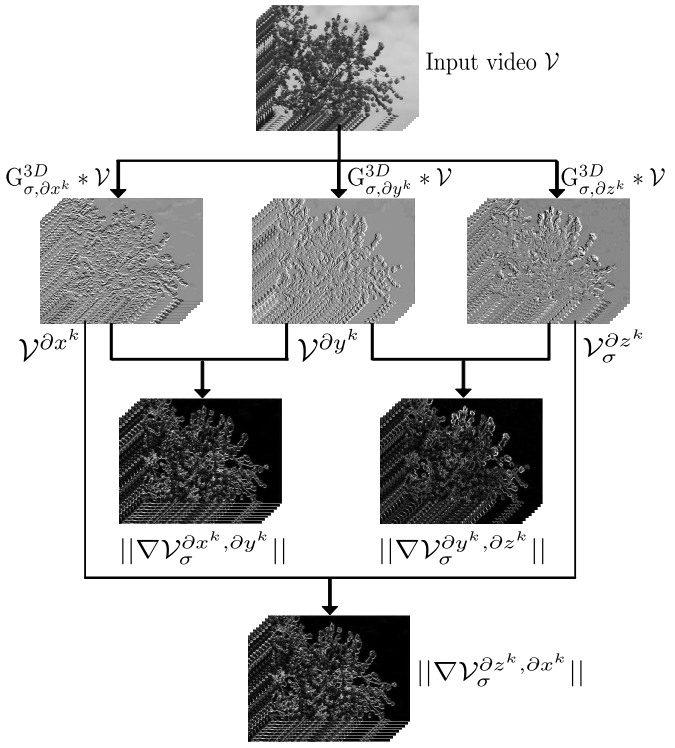

Fig. 3. An instance of 3D Gaussian-gradient filtering and computing volumes of magnitude features.

$$\begin{cases} \text{HVOM}_{\sigma,i}^{\partial x^k,\partial y^k}(\mathbf{u}) = \|\nabla \mathcal{V}_{\sigma}^{\partial x^k,\partial y^k}(\mathbf{u})\|, \text{so that } \phi_{\sigma}^{\partial x^k,\partial y^k}(\mathbf{u}) \in d \\ \text{HVOM}_{\sigma,i}^{\partial y^k,\partial z^k}(\mathbf{u}) = \|\nabla \mathcal{V}_{\sigma}^{\partial y^k,\partial z^k}(\mathbf{u})\|, \text{so that } \phi_{\sigma}^{\partial y^k,\partial z^k}(\mathbf{u}) \in d \\ \text{HVOM}_{\sigma,i}^{\partial z^k,\partial x^k}(\mathbf{u}) = \|\nabla \mathcal{V}_{\sigma}^{\partial z^k,\partial x^k}(\mathbf{u})\|, \text{so that } \phi_{\sigma}^{\partial z^k,\partial x^k}(\mathbf{u}) \in d \end{cases} \quad (15)$$

by the soft-assignment as

$$\begin{cases} \text{SVOM}_{\sigma,i}^{\partial x^k,\partial y^k}(\mathbf{u}) = \|\nabla \mathcal{V}_{\sigma}^{\partial x^k,\partial y^k}(\mathbf{u})\| \times \frac{\beta - \phi_{\sigma}^{\partial x^k,\partial y^k}(\mathbf{u})}{\beta-\alpha} \\ \text{SVOM}_{\sigma,i+1}^{\partial x^k,\partial y^k}(\mathbf{u}) = \|\nabla \mathcal{V}_{\sigma}^{\partial x^k,\partial y^k}(\mathbf{u})\| \times \frac{\phi_{\sigma}^{\partial x^k,\partial y^k}(\mathbf{u}) - \alpha}{\beta-\alpha} \\ \text{SVOM}_{\sigma,i}^{\partial y^k,\partial z^k}(\mathbf{u}) = \|\nabla \mathcal{V}_{\sigma}^{\partial y^k,\partial z^k}(\mathbf{u})\| \times \frac{\beta - \phi_{\sigma}^{\partial y^k,\partial z^k}(\mathbf{u})}{\beta-\alpha} \\ \text{SVOM}_{\sigma,i+1}^{\partial y^k,\partial z^k}(\mathbf{u}) = \|\nabla \mathcal{V}_{\sigma}^{\partial y^k,\partial z^k}(\mathbf{u})\| \times \frac{\phi_{\sigma}^{\partial y^k,\partial z^k}(\mathbf{u}) - \alpha}{\beta-\alpha} \\ \text{SVOM}_{\sigma,i}^{\partial z^k,\partial x^k}(\mathbf{u}) = \|\nabla \mathcal{V}_{\sigma}^{\partial z^k,\partial x^k}(\mathbf{u})\| \times \frac{\beta - \phi_{\sigma}^{\partial z^k,\partial x^k}(\mathbf{u})}{\beta-\alpha} \\ \text{SVOM}_{\sigma,i+1}^{\partial z^k,\partial x^k}(\mathbf{u}) = \|\nabla \mathcal{V}_{\sigma}^{\partial z^k,\partial x^k}(\mathbf{u})\| \times \frac{\phi_{\sigma}^{\partial z^k,\partial x^k}(\mathbf{u}) - \alpha}{\beta-\alpha} \end{cases} \quad (16)$$

where $\text{SVOM}_{\sigma,n+1}^{\partial z^k,\partial x^k}(\mathbf{u}) \equiv \text{SVOM}_{\sigma,1}^{\partial z^k,\partial x^k}(\mathbf{u})$, $\text{SVOM}_{\sigma,n+1}^{\partial y^k,\partial z^k}(\mathbf{u}) \equiv \text{SVOM}_{\sigma,1}^{\partial y^k,\partial z^k}(\mathbf{u})$, and $\text{SVOM}_{\sigma,n+1}^{\partial x^k,\partial y^k}(\mathbf{u}) \equiv \text{SVOM}_{\sigma,1}^{\partial x^k,\partial y^k}(\mathbf{u})$.
In the meanwhile, a feature of VOM can be quantified to two

bins by the modified soft-assignment as

$$\begin{cases} \text{pMSVOM}^{\partial x^k,\partial y^k}_{\sigma,i}(\mathbf{u}) = \|\nabla\mathcal{V}^{\partial x^k,\partial y^k}_{\sigma}(\mathbf{u})\| \times \frac{\beta-\phi^{\partial x^k,\partial y^k}_{\sigma}(\mathbf{u})}{\beta-\alpha} \\ \text{nMSVOM}^{\partial x^k,\partial y^k}_{\sigma,i+1}(\mathbf{u}) = \|\nabla\mathcal{V}^{\partial x^k,\partial y^k}_{\sigma}(\mathbf{u})\| \times \frac{\phi^{\partial x^k,\partial y^k}_{\sigma}(\mathbf{u})-\alpha}{\beta-\alpha} \\ \text{pMSVOM}^{\partial y^k,\partial z^k}_{\sigma,i}(\mathbf{u}) = \|\nabla\mathcal{V}^{\partial y^k,\partial z^k}_{\sigma}(\mathbf{u})\| \times \frac{\beta-\phi^{\partial y^k,\partial z^k}_{\sigma}(\mathbf{u})}{\beta-\alpha} \\ \text{nMSVOM}^{\partial y^k,\partial z^k}_{\sigma,i+1}(\mathbf{u}) = \|\nabla\mathcal{V}^{\partial y^k,\partial z^k}_{\sigma}(\mathbf{u})\| \times \frac{\phi^{\partial y^k,\partial z^k}_{\sigma}(\mathbf{u})-\alpha}{\beta-\alpha} \\ \text{pMSVOM}^{\partial z^k,\partial x^k}_{\sigma,i}(\mathbf{u}) = \|\nabla\mathcal{V}^{\partial z^k,\partial x^k}_{\sigma}(\mathbf{u})\| \times \frac{\beta-\phi^{\partial z^k,\partial x^k}_{\sigma}(\mathbf{u})}{\beta-\alpha} \\ \text{nMSVOM}^{\partial z^k,\partial x^k}_{\sigma,i+1}(\mathbf{u}) = \|\nabla\mathcal{V}^{\partial z^k,\partial x^k}_{\sigma}(\mathbf{u})\| \times \frac{\phi^{\partial z^k,\partial x^k}_{\sigma}(\mathbf{u})-\alpha}{\beta-\alpha} \end{cases} \tag{17}$$

in which $\text{nMSVOM}^{\partial z^k,\partial x^k}_{\sigma,n+1}(\mathbf{u}) \equiv \text{nMSVOM}^{\partial z^k,\partial x^k}_{\sigma,1}(\mathbf{u})$, $\text{nMSVOM}^{\partial y^k,\partial z^k}_{\sigma,n+1}(\mathbf{u}) \equiv \text{nMSVOM}^{\partial y^k,\partial z^k}_{\sigma,1}(\mathbf{u})$, and $\text{nMSVOM}^{\partial x^k,\partial y^k}_{\sigma,n+1}(\mathbf{u}) \equiv \text{nMSVOM}^{\partial x^k,\partial y^k}_{\sigma,1}(\mathbf{u})$.

Here, the k-order magnitudes $\|\nabla\mathcal{V}^{\partial x^k,\partial y^k}_{\sigma}(\mathbf{u})\|$, $\|\nabla\mathcal{V}^{\partial y^k,\partial z^k}_{\sigma}(\mathbf{u})\|$, and $\|\nabla\mathcal{V}^{\partial z^k,\partial x^k}_{\sigma}(\mathbf{u})\|$ are computed as

$$\begin{cases} \|\nabla\mathcal{V}^{\partial x^k,\partial y^k}_{\sigma}(\mathbf{u})\| = \sqrt{\left(\mathcal{V}^{x^k}_{\sigma}(\mathbf{u})\right)^2 + \left(\mathcal{V}^{y^k}_{\sigma}(\mathbf{u})\right)^2} \\ \|\nabla\mathcal{V}^{\partial y^k,\partial z^k}_{\sigma}(\mathbf{u})\| = \sqrt{\left(\mathcal{V}^{y^k}_{\sigma}(\mathbf{u})\right)^2 + \left(\mathcal{V}^{z^k}_{\sigma}(\mathbf{u})\right)^2} \\ \|\nabla\mathcal{V}^{\partial z^k,\partial x^k}_{\sigma}(\mathbf{u})\| = \sqrt{\left(\mathcal{V}^{z^k}_{\sigma}(\mathbf{u})\right)^2 + \left(\mathcal{V}^{x^k}_{\sigma}(\mathbf{u})\right)^2} \end{cases} \tag{18}$$

In order to illustrate the decomposition of gradient-filtered volumes, Fig. 3 shows an example of computing magnitude volumes of Gaussian gradients. Gradient directions $\phi^{\partial x^k,\partial y^k}_{\sigma}(\mathbf{u})$, $\phi^{\partial y^k,\partial z^k}_{\sigma}(\mathbf{u})$, and $\phi^{\partial z^k,\partial x^k}_{\sigma}(\mathbf{u})$ are inferred as

$$\begin{cases} \phi^{\partial x^k,\partial y^k}_{\sigma}(\mathbf{u}) = \arctan\left(\mathcal{V}^{\partial y^k}_{\sigma}(\mathbf{u})/\mathcal{V}^{\partial x^k}_{\sigma}(\mathbf{u})\right) \\ \phi^{\partial y^k,\partial z^k}_{\sigma}(\mathbf{u}) = \arctan\left(\mathcal{V}^{\partial z^k}_{\sigma}(\mathbf{u})/\mathcal{V}^{\partial y^k}_{\sigma}(\mathbf{u})\right) \\ \phi^{\partial z^k,\partial x^k}_{\sigma}(\mathbf{u}) = \arctan\left(\mathcal{V}^{\partial x^k}_{\sigma}(\mathbf{u})/\mathcal{V}^{\partial z^k}_{\sigma}(\mathbf{u})\right) \end{cases} \tag{19}$$

In the meanwhile, Fig. 5 graphically illustrates a general model of decomposing a volume of magnitude features.

It can be seen that for a given direction range, the modified soft decomposition has produced a double number of oriented magnitude outcomes compared to the hard-assignment and the classic soft-assignment. For convenience in further presentation, we could generally refer the above decomposing results: HIOM/SIOM/MSIOM as IOM-based images, HVOM/SVOM/MSVOM as VOM-based volumes.

3.2. DT representation based on oriented magnitudes

In order to generally investigate oriented magnitudes for DT representation, we address the IOM and VOM computations in n ($n \in \mathbb{Z}^+$) equal ranges of direction as $\mathcal{D}^n = \{[(i-1)\lambda, i\lambda)\}_{i=1}^{n}$, where $\lambda = \frac{2\pi}{n}$ denotes an angle coefficient for decomposing the k-order image/volume magnitudes. For example, with respect to $\lambda = \pi/2$, we have $n = 4$ direction ranges in equality (i.e., $\mathcal{D}^4 = \{[0,\pi/2), [\pi/2,\pi), [\pi,3\pi/2), [3\pi/2,2\pi)\}$). In Fig. 2, we respectively used direction ranges of $\mathcal{D}^4$ to decompose a magnitude image $\|\nabla\mathcal{I}^{\partial x^k,\partial y^k}_{\sigma}\|$. Hereunder, we propose robust descriptors structured corresponding to the IOM-based and VOM-based outcomes.

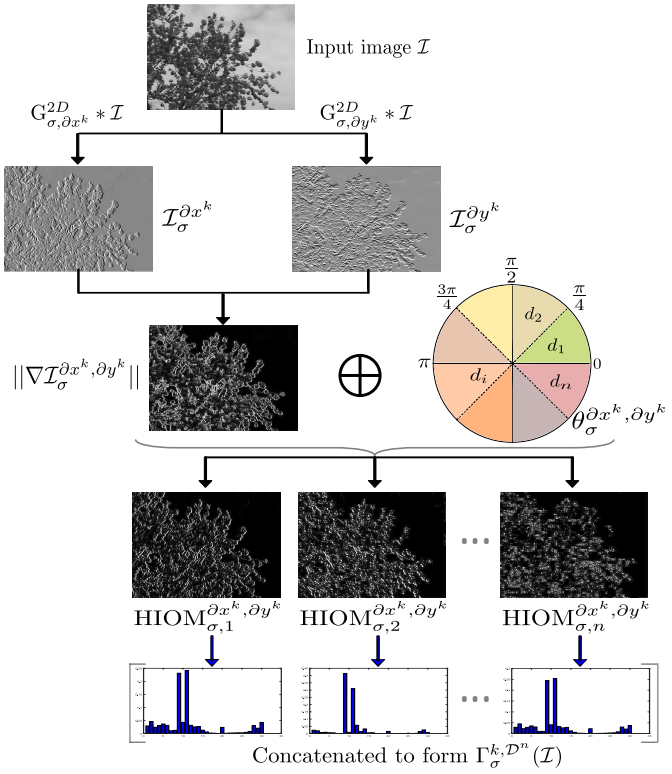

Fig. 4. (Best viewed in color) Flowchart of HIOM model subject to direction ranges $d_i = [(i-1)\lambda, i\lambda)$ in $\mathcal{D}^n$. Therein, the black arrows are noted for pre-processing while the blue ones are for encoding.

Proposed IOM-based descriptors: To be compliant with the k-order 2D Gaussian-gradient filtering, a given video $\mathcal{V}$ is separated subject to its three orthogonal planes $\{XY, XT, YT\}$ to obtain corresponding collections of plane-images f_{XY}, f_{XT}, and f_{YT}. For the plane-image collection f_{XY}, its spatial HIOM, SIOM, MSIOM features of DTs are respectively encoded as

$$\Gamma^{k,\mathcal{D}^n}_{\sigma}(f_{XY}) = \frac{1}{\mathcal{N}_{XY}} \sum_{\mathcal{I}\in f_{XY}} \Big[\xi(\text{HIOM}^{\partial x^k,\partial y^k}_{\sigma,1}(\mathcal{I})), ..., \xi(\text{HIOM}^{\partial x^k,\partial y^k}_{\sigma,n}(\mathcal{I}))\Big] \tag{20}$$

and

$$\Upsilon^{k,\mathcal{D}^n}_{\sigma}(f_{XY}) = \frac{1}{\mathcal{N}_{XY}} \sum_{\mathcal{I}\in f_{XY}} \Big[\xi(\text{SIOM}^{\partial x^k,\partial y^k}_{\sigma,1}(\mathcal{I})), \xi(\text{SIOM}^{\partial x^k,\partial y^k}_{\sigma,2}(\mathcal{I})), ..., \xi(\text{SIOM}^{\partial x^k,\partial y^k}_{\sigma,n}(\mathcal{I}))\Big] \tag{21}$$

and

$$\Omega^{k,\mathcal{D}^n}_{\sigma}(f_{XY}) = \frac{1}{\mathcal{N}_{XY}} \sum_{\mathcal{I}\in f_{XY}} \Big[\xi(\text{pMSIOM}^{\partial x^k,\partial y^k}_{\sigma,1}(\mathcal{I})), \xi(\text{nMSIOM}^{\partial x^k,\partial y^k}_{\sigma,1}(\mathcal{I})), ..., \xi(\text{pMSIOM}^{\partial x^k,\partial y^k}_{\sigma,n}(\mathcal{I})), \xi(\text{nMSIOM}^{\partial x^k,\partial y^k}_{\sigma,n}(\mathcal{I}))\Big] \tag{22}$$

in which $\mathcal{N}_{XY}$ means a number of plane-images in f_{XY}; $\xi(.)$ denotes a simple function using a local operator (e.g., LBP [41],

CLBP [37], etc.) in order to figure out the corresponding histograms. Fig. 4 illustrates a graphical view of filtering an input image, hard-decomposing its filtered magnitudes, and encoding the obtained HIOM outcomes correspondingly. In similarity, these encodings could be used for the remaining plane-image collections f_{XT} and f_{YT} to capture temporal IOM-based features for DT representation. As a result, robust local descriptors are structured in simplicity by concatenating the probability distributions of $\Gamma_{\sigma}^{k,\mathcal{D}^n}(.)$, $\Upsilon_{\sigma}^{k,\mathcal{D}^n}(.)$, and $\Omega_{\sigma}^{k,\mathcal{D}^n}(.)$ as

$$\text{HIOMF}_{\sigma}^{k,\mathcal{D}^n}(\mathcal{V}) = [\Gamma_{\sigma}^{k,\mathcal{D}^n}(f_{XY}), \Gamma_{\sigma}^{k,\mathcal{D}^n}(f_{XT}), \Gamma_{\sigma}^{k,\mathcal{D}^n}(f_{YT})] \quad (23)$$

and

$$\text{SIOMF}_{\sigma}^{k,\mathcal{D}^n}(\mathcal{V}) = [\Upsilon_{\sigma}^{k,\mathcal{D}^n}(f_{XY}), \Upsilon_{\sigma}^{k,\mathcal{D}^n}(f_{XT}), \Upsilon_{\sigma}^{k,\mathcal{D}^n}(f_{YT})] \quad (24)$$

and

$$\text{MSIOMF}_{\sigma}^{k,\mathcal{D}^n}(\mathcal{V}) = [\Omega_{\sigma}^{k,\mathcal{D}^n}(f_{XY}), \Omega_{\sigma}^{k,\mathcal{D}^n}(f_{XT}), \Omega_{\sigma}^{k,\mathcal{D}^n}(f_{YT})] \quad (25)$$

Proposed VOM-based descriptors: As mentioned in Section 3.1 for the hard decomposition (refer to Eq. 15), three filtered volumes of oriented magnitudes are pointed out corresponding to three pairs of spacial domains convolved on a given video $\mathcal{V}$. Those volumes are taken into account local analysis to construct a robust descriptor as follows. For an obtained volume $\text{HVOM}_{\sigma,i}^{\partial x^k,\partial y^k}$, $(i \in \{1, 2, .., n\})$, it is firstly split into collections of filtered plane-images (f'_{XY}, f'_{XT}, and f'_{YT}) subject to its three orthogonal planes $\{XY, XT, YT\}$. The simple operator $\xi(.)$ is then utilized to capture local spatio-temporal features of DTs as

$$\Psi(\text{HVOM}_{\sigma,i}^{\partial x^k,\partial y^k}) = \left[\frac{\sum\limits_{\mathcal{I}\in f'_{XY}} \xi(\mathcal{I})}{\mathcal{N}'_{XY}}, \frac{\sum\limits_{\mathcal{I}\in f'_{XT}} \xi(\mathcal{I})}{\mathcal{N}'_{XT}}, \frac{\sum\limits_{\mathcal{I}\in f'_{YT}} \xi(\mathcal{I})}{\mathcal{N}'_{YT}}\right] \quad (26)$$

in which $\mathcal{N}'_{XY}$, $\mathcal{N}'_{XT}$, and $\mathcal{N}'_{YT}$ are numbers of plane-images f'_{XY}, f'_{XT}, and f'_{YT} of $\text{HVOM}_{\sigma,i}^{\partial x^k,\partial y^k}$ respectively. Fig. 5 illustrates a graphical view of encoding a HVOM volume. This encoding is similarly deployed for the remaining volumes $\text{HVOM}_{\sigma,i}^{\partial y^k,\partial z^k}$ and $\text{HVOM}_{\sigma,i}^{\partial z^k,\partial x^k}$. As a result, a discriminative descriptor based on the k-order HVOM features is constructed by concatenating these obtained histograms as

$$\text{HVOMF}_{\sigma}^{k,\mathcal{D}^n}(\mathcal{V}) = \biguplus\Big[\Psi(\text{HVOM}_{\sigma,i}^{\partial x^k,\partial y^k}), \Psi(\text{HVOM}_{\sigma,i}^{\partial y^k,\partial z^k}), \Psi(\text{HVOM}_{\sigma,i}^{\partial z^k,\partial x^k})\Big]_{i=1}^{n} \quad (27)$$

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

Similarly, this HVOMF encoding could be applied to 3 SVOM (resp. 6 MSVOM) outcomes extracted by the soft decomposition (refer to Eq. 16) subject to the direction range $\mathcal{D}^n$. Accordingly, other robust descriptors based on the k-order SVOM (resp. MSVOM) features are formed by concatenating the corresponding histograms as

$$\text{SVOMF}_{\sigma}^{k,\mathcal{D}^n}(\mathcal{V}) = \biguplus\Big[\Psi(\text{SVOM}_{\sigma,i}^{\partial x^k,\partial y^k}), \Psi(\text{SVOM}_{\sigma,i}^{\partial y^k,\partial z^k}), \Psi(\text{SVOM}_{\sigma,i}^{\partial z^k,\partial x^k})\Big]_{i=1}^{n} \quad (28)$$

and

$$\begin{aligned}\text{MSVOMF}_{\sigma}^{k,\mathcal{D}^n}(\mathcal{V}) = \biguplus\Big[&\Psi(\text{pMSVOM}_{\sigma,i}^{\partial x^k,\partial y^k}),\\ &\Psi(\text{nMSVOM}_{\sigma,i}^{\partial x^k,\partial y^k}),\\ &\Psi(\text{pMSVOM}_{\sigma,i}^{\partial y^k,\partial z^k}),\\ &\Psi(\text{nMSVOM}_{\sigma,i}^{\partial y^k,\partial z^k}),\\ &\Psi(\text{pMSVOM}_{\sigma,i}^{\partial z^k,\partial x^k}),\\ &\Psi(\text{nMSVOM}_{\sigma,i}^{\partial z^k,\partial x^k})\Big]_{i=1}^{n}\end{aligned} \quad (29)$$

Our proposed IOM/VOM-based descriptors take the following benefits to improve the performance compared to other local Gaussian-based descriptors (also see Sections 4.3, 4.5 for comprehensive evaluations):

- Different from exploiting Gaussian-based filtered features to construct local descriptors FoSIG [40] and V-BIG [39], in this work, the high-order oriented magnitudes are taken into account DT representation. Thanks to the decomposing models presented in Section 3.1, the magnitudes of Gaussian-gradient-filtered outcomes are addressed in diversity of invariant features to enhance the robustness against the well-known issues in more effect. In the mean while, exploiting oriented features makes those outcomes still more discriminative for texture description.
- The Gaussian-gradient filterings allow to produce more filtered outcomes for the DT encoding. In the meanwhile, just one DoG-based element was used in FoSIG [40] and V-BIG [39] due to taking the Different of Gaussians (DoG) kernel into account the filterings.
- To enhance the discrimination power, it is possible to address the IOM/VOM-based descriptors for a multi-analysis of high-orders along with different Gaussian filtering scales, while keeping their representation in reasonable dimensions thanks to the tiny size of single-scale ones (see Table 2). In the meantime, just single-scale of Gaussian filtering was addressed in FoSIG [40] and V-BIG [39].
- It should be noted that the 2D-magnitude information (i.e., non-decomposition applied to) is also exploited in [56] for structuring textual images. However, taking it into account DT representation is not more adaptive than taking its oriented properties (see Table 3 for a fact of this statement). It has proved the interest of our proposed framework.

4. Experiments and evaluations

4.1. Datasets and protocols

Hereafter, benchmark datasets and protocols for evaluating our proposal are detailed. A brief of their properties is then shown at a glance in Table 1.

UCLA dataset: It has 200 DT videos fixed in $110\times160\times75$ dimension [15]. Those mainly characterize disorder motions of waterfall, plant, flower, fountain, fire, boiling water, etc. (see Fig. 6 for several instances of them). For DT classification task, UCLA is often organized in challenging scenarios as follows:

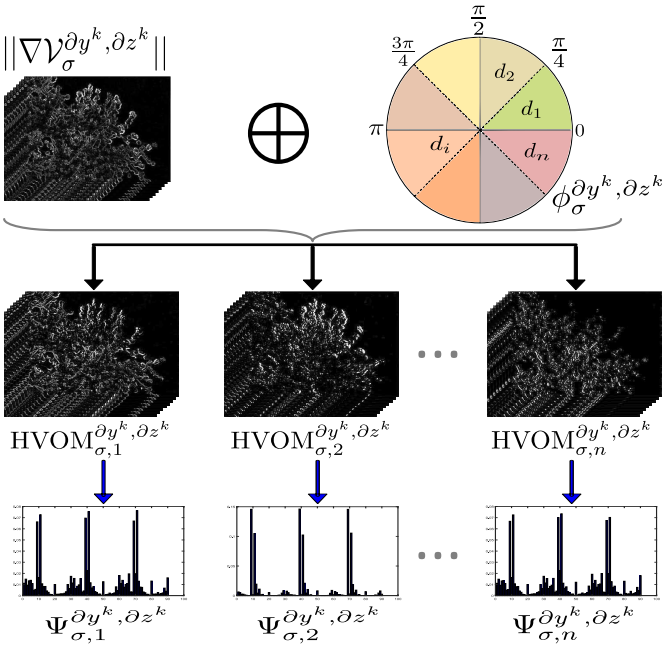

Fig. 5. (Best viewed in color) Flowchart of HVOM model subject to direction ranges $d_i = [(i-1)\lambda, i\lambda)$ in $\mathcal{D}^n$. Therein, the black arrows are pre-processing steps while the blue ones are for encoding.

- *50-class breakdown:* 200 DT sequences are arranged into 50 classes with 4 videos for each category. Protocols of leave-one-out (LOO) and four cross-fold validation are used for classifying DTs [35, 45, 39, 40].
- *9-class and 8-class breakdowns:* 200 DT sequences are arranged into 9 classes to form *9-class* scheme [27, 10]. It includes "boiling water(8)", "fire(8)", "flowers(12)", "fountains(20)", "plants(108)", "sea(12)", "smoke(4)", "water(12)", and "waterfall(16)", where the numbers in parentheses denote quantities for the corresponding classes. Because of the dominance of "plants(108)", it is removed to form *8-class* scheme with more challenges [27, 10]. In order to evaluate DT classification in two schemes, following to the experimental protocol in [17, 45, 40], a half of DT sequences in each class is randomly addressed for testing and the remain for training. The average of 20 trials for each scheme is reported as a final rate.

Fig. 6. Several DT sequences of UCLA

DynTex dataset: It is a challenging dataset for DT classification [16]. DynTex has 679 DT videos recorded in AVI format with dimension of $352 \times 288 \times 250$, in which turbulent motions of DTs are captured in different conditions of environmental changes (see Fig. 7 for several DT samples). For DT classification, it is often arranged into the following subsets. The LOO protocol [31, 57, 39] is utilized to evaluate the performances.

- *DynTex35* is composed by splitting from 35 DynTex videos as follows. Each video is split into 8 non-overlapping sub-DTs at random cutting points with respect to axes X, Y, and T, but not half of those. For instance, cutting coordinates can be $x = 170$, $y = 130$, and $t = 100$ as in [42, 35, 45, 38, 40]. In addition, two more sub-DTs are also collected subject to the T axis of the splitting process. As a result, 10 sub-videos for each of 35 videos are obtained in different spatio-temporal dimensions to form a challenging scheme with 35 categories.
- *Alpha* includes 60 DT videos which are grouped into 3 classes: "grass", "sea", and "trees". Each category has 20 sequences.
- *Beta* includes 162 DT videos which are grouped into 10 categories: "sea(20)", "vegetation(20)", "trees(20)", flags(20)", "calm water(20)", "fountains(20)", "traffic(9)", "smoke(16)", "escalator(7)", and "rotation(10)", in which the numbers in parentheses mean quantities of videos in the corresponding categories.
- *Gamma* includes 264 DT videos which are also arranged into 10 categories: "flowers(29)", "sea(38)", "naked trees(25)", "foliage(35)", "escalator(7)", "calm water(30)", "flags(31)", "grass(23)", "traffic(9)", and "fountains(37)", where the numbers in parentheses denote quantities of sequences in the corresponding categories.

Table 1. A brief of main properties of 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%

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

Fig. 7. Several samples of DynTex dataset

DynTex++ dataset: It is composed as follows. 345 DynTex's videos are split and filtered so that only major textural motions are captured [17]. The obtained sub-videos are then grouped into 36 classes with 100 sub-videos for each of them, i.e., 3600 sub-videos in total. Be similar to experimental protocol in [17, 35, 58], a half of samples in each class is randomly taken out for training, and the remain for testing. The average of 20 trials is reported as a final result.

4.2. Experimental settings

For computing high-order Gaussian-gradient-based responses: We investigate 2D/3D Gaussian filtering kernels in high-order gradients of $k \in \{1, 2, 3, 4\}$. Therein, standard deviation $\sigma \in \{0.5, 0.7, 1, 1.3, 1.5, 2\}$ and spatio-temporal coordinates

of convolution $x, y, z \in [-3\sigma, 3\sigma]$ could be empirically conducted for each Gaussian-gradient kernel in order to compute corresponding filtered outcomes.

For the decomposition of oriented magnitudes: With respect to addressing direction ranges for decomposing these obtained responses to achieve IOM-based images and VOM-based volumes, it can take into account various numbers of equal direction ranges, e.g., $n \in \{4, 6, 8\}$ respectively corresponding to $\lambda \in \{\pi/2, \pi/3, \pi/4\}$. Furthermore, as mentioned in Section 3.1 (refer to Eqs. (10), (11), (12), (15), (16), (17)), the modified soft-assignment decomposition has produced a double number of oriented magnitude outcomes than the others. To take an objective evaluation in effectiveness of these decomposing models, we address $n = 8$ (i.e., $\mathcal{D}^8$) for the traditional models (i.e., hard and soft) and $n = 4$ (i.e., $\mathcal{D}^4$) for our modified soft assignment in order to obtain the same numbers of outcomes. This could be appropriate since for a direction range $[0, \pi/2)$, the soft model and its modified version respectively decompose a magnitude image into 2 SIOMs (refer to Eq. (11)) and 4 MSIOMs (refer to Eq. (12)) by adopting the pixels which their gradient directions are close to $\pi/4$. It is nearly the same that the hard model is addressed in two ranges $[0, \pi/4)$ and $[\pi/4, \pi/2)$ to obtain 2 HIOMs (refer to Eq. (10)) correspondingly.

For structuring IOM-based and VOM-based descriptors: In order to encode the obtained outcomes of oriented magnitudes, we use a simple operator CLBP [37], one of the most popular local operator, with *riu*2 mapping and local supporting region $\{(P, R)\} = \{(8, 1)\}$, i.e., $\xi = \text{CLBP}^{riu2}_{8,1}$. To structure our proposed descriptors in reasonable dimension, the integration of "S_M/C" should be utilized for jointing CLBP's components. That means it generally needs $\omega = 3(P + 2) \times 3 \times |\nabla|$ bins for representing the oriented magnitudes decomposed by a direction range. Therein, $|\nabla|$ denotes a number of Gaussian-gradient magnitudes fed into a decomposing model. As a result, the final dimension to describe a DT video is subject to which the decomposing model is taken into account. For instance, using $\mathcal{D}^8$ for the traditional decomposition (i.e., $n = 8$), dimension of single-scale $\text{HIOMF}^{k,\mathcal{D}^8}_{\sigma}$ (i.e., $|\nabla| = 1$) is $\omega \times 8 = 720$ bins, while that of single-scale $\text{HVOMF}^{k,\mathcal{D}^8}_{\sigma}$ (i.e., $|\nabla| = 3$) is $\omega \times 8 = 2160$ bins. Those are the same bins for $\text{SIOMF}^{k,\mathcal{D}^8}_{\sigma}$ and $\text{SVOMF}^{k,\mathcal{D}^8}_{\sigma}$ respectively. Due to addressing $\mathcal{D}^4$ for the modified soft-assignment, the dimensions in single-scale analysis of $\text{MSIOMF}^{k,\mathcal{D}^4}_{\sigma}$ and $\text{MSVOMF}^{k,\mathcal{D}^4}_{\sigma}$ is also the same as those above, i.e., $\omega \times 2 \times 4 = 720$ and $\omega \times 2 \times 4 = 2160$ bins respectively. Table 2 shows the dimensions of our descriptors in comparison with those of current local methods. Due to these tiny bins, it is possible to take advantage of the IOM/VOM-based outcomes in multi-oriented magnitudes by addressing multi-scale of standard deviations and multi-order of Gaussian-gradient kernels. This analysis is to enrich more discriminative information for improvement of their performances.

For DT classification: In order to evaluate performances of our IOM/VOM-based descriptors in classifying DTs, we use the linear multi-class SVM classifier of LIBLINEAR [59], which the default parameters are involved in.

Table 2. A comparison of various dimensions of LBP-based descriptors.

Method	#bins	$P = 8$
LBP-TOP^{u2} [42]	$3(P(P-1)+3)$	177
VLBP [42]	2^{3P+2}	-
CVLBP [44]	$3 \times 2^{3P+2}$	-
HLBP [45]	6×2^P	1536
CLSP-TOP^{riu2} [46]	$6(P+2)^2$	600
WLBPC [60]	6×2^P	1536
MEWLSP [58]	6×2^P	1536
CVLBC [61]	$2(3P+3)^2$	1458
CSAP-TOP^{riu2}[38]	$12(P+2)^2$	1200
FDT^{u2} [22]	$216P((P-1)+3)$	12744
FD-MAP$^{u2}_{L=2}$ [22]	$216P((P-1)+3)+16$	12760
$\textbf{HIOMF}^{k,\mathcal{D}^8}_{\sigma}$, $\textbf{SIOMF}^{k,\mathcal{D}^8}_{\sigma}$, $\textbf{MSIOMF}^{k,\mathcal{D}^4}_{\sigma}$	$72(P+2)$	720
$\textbf{HVOMF}^{k,\mathcal{D}^8}_{\sigma}$, $\textbf{SVOMF}^{k,\mathcal{D}^8}_{\sigma}$, $\textbf{MSVOMF}^{k,\mathcal{D}^4}_{\sigma}$	$216(P+2)$	2160

Note: P denotes the concerned neighbors. "-" means "not available".

4.3. Assessments of effectiveness of decomposing models

As mentioned in Sections 3.1 and 3.2, corresponding to the decomposing models, we address the proposed IOM/VOM-based descriptors for DT classification task on the challenging schemes, i.e., *Beta*, *Gamma*, and DynTex++. For an objective comparison, we also take non-oriented Gaussian-gradient magnitudes into account DT representation with the same encoding parameters (i.e., $\xi = \text{CLBP}^{riu2}_{8,1}$) in order to construct corresponding descriptors of image/volume non-oriented magnitude features (IMF^{k}_{σ} and VMF^{k}_{σ}). Experimental results in Table 3 have shown classification rates of these descriptors in various scale analyses. Based on those, it could be pointed out two crucial statements as follows.

- In general, it can be seen from Table 3 that the ability of the basic soft-assignment does not perform well in decomposing Gaussian-gradient magnitudes for DT encoding compared to the hard one. Even, it is inferior to the non-decomposing model (i.e., exploiting IMF and VMF features of non-oriented magnitudes) in some cases, e.g., DT recognition on *Beta* as shown in Fig. 8. It may be due to the intensified textural appearances caused by quantizing oriented magnitudes in adjacent orientation ranges instead of softly separating as in our modified model.
- As expected, our modified soft-assignment has much improved the performance compared to its original model (see classification rates in columns "3D-S" and "3D-B" of Table 3). Furthermore, its discriminative power is significantly better than that of the non-decomposing and hard ones (see Table 3). This is thanks to the adjusted voting strategy as proposed in Section 3.1. It has appropriately adopted the magnitude features subject to a given direction range to obtain filtered outcomes in more robustness for DT encoding (refer to Eqs. (12) and (17) for detail).

Due to the good discrimination in the extraction of oriented magnitudes, the modified soft decomposition should be recommended for processing Gaussian-gradient magnitudes in practice. Accordingly, in the rest of this work, we mainly discuss the performances of the MSIOMF and MSVOMF descriptors in comprehensive comparison with those of recent approaches.

Table 3. Classification rates (%) on the challenging schemes of descriptors based on non-oriented-magnitude and IOM/VOM-based features.

Scheme		Beta							Gamma							DynTex++						
Order	$\{\sigma_i\}$	2D-H	2D-S	3D-H	3D-S	3D-B	IMF	VMF	2D-H	2D-S	3D-H	3D-S	3D-B	IMF	VMF	2D-H	2D-S	3D-H	3D-S	3D-B	IMF	VMF
1^{st}	{0.7}	91.36	90.74	90.74	93.21	90.74	91.36	93.83	92.80	**93.94**	91.29	**94.32**	91.29	**90.91**	91.67	**95.77**	**96.08**	**97.13**	**97.01**	96.66	**87.99**	**93.68**
	{1.0}	91.36	91.36	91.98	92.59	90.12	91.98	93.21	92.42	92.80	93.18	93.18	92.80	89.39	90.53	94.72	95.73	96.18	96.76	95.57	88.92	93.19
	{1.3}	91.98	91.36	91.98	92.59	89.51	89.51	93.83	90.15	92.80	91.67	92.42	92.42	87.50	90.53	94.61	95.05	96.05	96.05	95.47	85.51	91.09
	{1.5}	89.51	91.36	91.36	91.98	90.74	91.36	92.59	92.05	92.05	92.42	92.05	93.18	89.02	91.67	93.90	94.98	95.51	95.85	95.08	86.96	91.10
2^{nd}	{0.7}	91.36	**93.83**	91.36	94.44	91.36	91.36	**94.44**	**93.18**	93.56	93.56	93.18	93.56	89.39	91.67	95.66	95.76	96.51	96.82	**96.77**	85.73	93.09
	{1.0}	93.21	93.21	92.59	**95.06**	90.74	**92.59**	91.98	92.42	93.18	92.80	93.56	**94.32**	**90.91**	93.56	94.88	95.39	96.44	96.23	96.09	86.03	92.10
	{1.3}	91.36	91.36	91.36	93.83	91.98	88.27	90.74	90.53	93.56	91.67	93.94	89.77	89.02	91.67	94.10	94.51	95.31	96.28	94.88	84.76	92.17
	{1.5}	90.74	92.59	93.21	93.21	91.36	90.74	92.59	92.42	92.80	**93.94**	93.18	93.18	86.64	91.67	94.19	94.07	95.14	95.93	95.40	83.51	91.35
3^{rd}	{0.7}	89.51	89.51	91.98	92.59	**92.59**	89.51	93.83	91.67	**93.94**	90.53	93.18	93.94	86.74	91.67	95.54	95.67	96.51	96.81	96.23	85.49	92.57
	{1.0}	91.36	92.59	93.21	92.59	91.36	88.89	93.83	91.67	93.18	90.53	91.29	92.80	89.39	92.80	93.52	95.34	95.82	96.18	95.04	85.71	91.88
	{1.3}	**95.06**	93.21	**95.06**	93.83	91.36	88.27	93.21	92.42	91.29	93.18	93.18	92.05	89.77	92.42	93.88	94.34	95.27	96.16	94.63	83.84	92.31
	{1.5}	90.74	91.98	93.21	93.83	88.89	90.12	90.74	90.91	91.29	91.67	91.67	90.53	90.53	92.80	94.20	94.38	94.83	95.66	93.79	85.00	91.26
4^{th}	{0.7}	92.59	**93.83**	93.83	93.83	**92.59**	90.12	93.21	90.91	93.18	93.56	93.94	93.94	86.74	90.53	94.81	95.02	96.39	96.07	95.99	85.62	93.07
	{1.0}	90.74	91.36	92.59	**95.06**	89.51	88.89	93.83	89.77	90.53	92.05	**94.32**	90.91	86.74	**94.32**	94.27	95.22	95.55	96.57	95.46	85.46	92.47
	{1.3}	90.12	90.74	90.74	94.44	90.12	89.51	92.59	91.29	91.29	92.80	93.56	92.42	88.64	92.05	93.58	94.77	95.56	95.82	94.37	86.73	**93.68**
	{1.5}	89.51	91.98	91.36	94.44	90.74	89.51	93.83	90.53	92.42	92.80	**94.32**	93.18	90.15	93.56	92.72	93.90	94.89	95.62	94.44	84.19	91.09

Note: Respectively, 2D-H and 3D-H denote for oriented magnitude descriptors $\text{HIOMF}_{\sigma}^{k,\mathcal{D}^8}$ and $\text{HVOMF}_{\sigma}^{k,\mathcal{D}^8}$ using the hard-decomposing model, while 2D-S and 3D-S are for $\text{MSIOMF}_{\sigma}^{k,\mathcal{D}^4}$ and $\text{MSVOMF}_{\sigma}^{k,\mathcal{D}^4}$ with the modified soft decomposition. 3D-B denotes for $\text{SVOMF}_{\sigma}^{k,\mathcal{D}^8}$ based on the basic soft-assignment model. IMF and VMF stand for non-oriented magnitude ones IMF_{σ}^{k} and VMF_{σ}^{k}, i.e., none of the decomposing models is involved in the DT encoding.

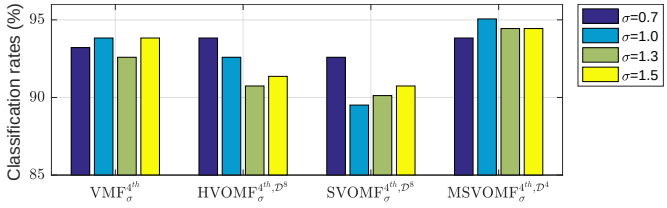

Fig. 8. (Best viewed in color) Performances (%) on *Beta* of descriptors based on the 4^{th}-order 3D Gaussian-gradient magnitudes using both decomposing and non-decomposing models.

4.4. Complexity of IOM/VOM-based descriptors

In general, it could be seen from the construction in Section 3.2 that it takes three main stages to structure our IOM/VOM-based descriptors: *i)* the filtering using Gaussian-gradient kernels; *ii)* the processes of decomposition; *iii)* the local feature extraction from the obtained IOM/VOM outcomes. Hereunder, we thoroughly discuss the complexity of encoding our proposed descriptors as well as measure the corresponding runtimes compared to other LBP-based ones.

Let $\mathcal{Q}_{\text{LBP}} = O(P \times \mathcal{H} \times \mathcal{W})$ be the computational cost of the basic LBP [41] operator for encoding an image with $\mathcal{H} \times \mathcal{W}$ dimension, in which P denotes a number of concerned neighbors. For encoding a video $\mathcal{V}$, Zhao *et al.* [42] addressed LBP on three orthogonal planes $\{XY, XT, YT\}$ of $\mathcal{V}$ to form LBP-TOP patterns with the cost of $\mathcal{Q}_{\text{LBP-TOP}} = O(P \times \mathcal{H} \times \mathcal{W} \times \mathcal{T})$, where $\mathcal{T}$ denotes the quantity of $\mathcal{V}$'s frames. As mentioned in Section 4.2, ξ = CLBP [37] was addressed in this work to encode the IOM/VOM outcomes. The CLBP's complexity is approximately estimated as $\mathcal{Q}_{\xi} \approx 3 \times \mathcal{Q}_{\text{LBP}}$ because its complementary components (i.e., CLBP_S, CLBP_M, CLBP_C) could be computed independently (refer to [37] for more details). In addition, it can be deduced from Eqs. (10), (11), (12) that the cost of the decomposition for the gradient-filtered images is estimated as $\mathcal{Q}_{\text{IOM}} = \mathcal{Q}_{\nabla_I} + \mathcal{Q}_{\theta}$, where $\mathcal{Q}_{\nabla_I}$ and $\mathcal{Q}_{\theta}$ denote the cost of computing the magnitude image and the gradient direction respectively. Due to Eqs. (13) and (14), $\mathcal{Q}_{\nabla_I} = \mathcal{Q}_{\theta} = O(\mathcal{H} \times \mathcal{W})$, i.e., $\mathcal{Q}_{\text{IOM}} \approx O(\mathcal{H} \times \mathcal{W})$ in general. Similarly, referring to Eqs. (15), (16), (17), we also have the cost of decomposing the gradient-filtered volumes $\mathcal{Q}_{\text{VOM}} = \mathcal{Q}_{\nabla_{\mathcal{V}}} + \mathcal{Q}_{\phi}$, where $\mathcal{Q}_{\nabla_{\mathcal{V}}}$ and $\mathcal{Q}_{\phi}$ mean the cost of computing magnitude volumes and the gradient directions respectively. Due to Eqs. (18) and (19), $\mathcal{Q}_{\nabla_{\mathcal{V}}} = \mathcal{Q}_{\phi} = O(\mathcal{H} \times \mathcal{W} \times \mathcal{T})$, i.e., $\mathcal{Q}_{\text{VOM}} \approx O(\mathcal{H} \times \mathcal{W} \times \mathcal{T})$ in general. Based on those above, the complexity of our proposed descriptors can be deduced as follows.

Complexity of MSIOMF *descriptor:* According to Eq. (22), it can be deduced that the computational cost of encoding plane-images $\mathcal{I} \in f_{XY}$ is $\mathcal{Q}_{\Omega_{f_{XY}}} = 2n \times \mathcal{N}_{XY} \times (\mathcal{Q}_{\xi} + \mathcal{Q}_{\text{IOM}} + \mathcal{Q}_{G^{2D}})$. Therein, $\mathcal{Q}_{G^{2D}}$ denotes the cost of the 2D Gaussian-gradient filtering; $\mathcal{N}_{XY} = \mathcal{T}$ means the number of plane-images in f_{XY}. Because of the much smaller value of n (e.g., $n = 4$ for the modified soft-assignment (see Section 4.2)), as well as the separable property of the 2D Gaussian-gradient filtering, they can be disregarded. It means $\mathcal{Q}_{\Omega_{f_{XY}}} = \mathcal{T} \times (\mathcal{Q}_{\xi} + \mathcal{Q}_{\text{IOM}}) \approx O(P \times \mathcal{H} \times \mathcal{W} \times \mathcal{T})$. Since MSIOMF is structured on the separate collections of plane-images f_{XY}, f_{XT}, and f_{YT} (see Eq. (25)), its complexity is estimated as $\mathcal{Q}_{\text{MSIOMF}} \approx max(\{\mathcal{Q}_{\Omega_{f_{XY}}}, \mathcal{Q}_{\Omega_{f_{XT}}}, \mathcal{Q}_{\Omega_{f_{YT}}}\})$. Consequently, $\mathcal{Q}_{\text{MSIOMF}} \approx O(P \times \mathcal{H} \times \mathcal{W} \times \mathcal{T})$.

Complexity of MSVOMF *descriptor:* It can be seen from Eq. (26) that the cost for encoding a VOM-based volume can be estimated as $\mathcal{Q}_{\Psi} \approx \mathcal{T} \times \mathcal{Q}_{\xi}$. Subject to Eq. (29), the complexity of MSVOMF is formed as $\mathcal{Q}_{\text{MSVOMF}} = 6n \times (\mathcal{Q}_{\Psi} + \mathcal{Q}_{\text{VOM}} + \mathcal{Q}_{G^{3D}})$, where $\mathcal{Q}_{G^{3D}}$ is the cost of the 3D Gaussian-gradient filtering. Due to the much smaller value of n as well as the separable property of the 3D Gaussian-gradient filtering, they can be disregarded. Consequently, $\mathcal{Q}_{\text{MSVOMF}} \approx O(P \times \mathcal{H} \times \mathcal{W} \times \mathcal{T})$.

As a result, the complexity of encoding IOM/VOM-based features is the same simple order as that of other LBP-based methods: CVLBC [61], CSAP-TOP [38], CVLBP [44], VLBP [42], V-BIG [39], FoSIG [40], etc. (refer to these works for more detail). In the meantime, the performance of our proposed descriptors on DT recognition is significantly better than theirs, as thoroughly discussed in Sections 4.5, 4.6, and 4.7. With respect to the processing time, we measure runtime of encoding the IOM/VOM-based descriptors in comparison with the LBP-based others implemented by our prior work [49]. It can be seen from Table 4 that our runtimes are as similar as the others. It should be emphasized that all those executions have been implemented by raw MATLAB codes and run in single-threading on a 64-bit Linux desktop with a configuration of CPU Core i7

Table 4. Processing time of several LBP-based methods to structure a $50 \times 50 \times 50$ video of DynTex++.

Descriptor	Gradient	(σ [,σ'])	(P, R)	Mapping	Runtime(s)
VLBP [42]	-	-	(4, 1)	-	≈ 0.22
LBP-TOP [42]	-	-	(8, 1)	u2	≈ 0.15
CLSP-TOP [46]	-	-	(8, 1)	riu2	≈ 0.27
CSAP-TOP [38]	-	-	(8, 1)	riu2	≈ 0.50
FoSIG^{2D} [40]	-	(0.5, 6)	(8, 1)	riu2	≈ 0.37
V-BIG^{3D} [39]	-	(0.5, 6)	(8, 1)	riu2	≈ 0.35
RUBIG [49]	-	(0.5, 6)	(8, 1)	riu2	≈ 0.56
Our **MSIOMF**$_{\sigma}^{k,\mathcal{D}^4}$	1^{st}	$\sigma = 0.7$	(8, 1)	riu2	≈ 0.48
Our **MSVOMF**$_{\sigma}^{k,\mathcal{D}^4}$	1^{st}	$\sigma = 0.7$	(8, 1)	riu2	≈ 0.62

Note: "-" means "not available". Runtime of other LBP-based approaches is reported by implementations of our former work [49].

Table 5. Evaluation of hardware scability.

Number of threads	1	2	3	4
Times (in s)	622.74	334.23	242.08	195.46
Speed-up	1	1.863	2.572	3.186
Sequential coefficient (ω)	NA	0.0734	0.0831	0.0851

3.4GHz 16G RAM.

Scalability of the proposed method: We consider hereunder the scalability of our method. Table 5 shows the necessary time for descriptor construction of a video of size $352 \times 288 \times 250$ by using different numbers of threads of processing cores in the CPU to evaluate its hardware scalability. Let ω denote the sequential coefficient. According to the Amdahl's law [62], the maximal speedup which can be achieved by using C threads is determined as follows: $\left(\omega + \frac{1-\omega}{C}\right)^{-1}$ (refer to [63] for more detail). This allows to deduce the sequential coefficients when the number of threads is changed, as presented in the last row of Table 5. Accordingly, the proposed method can be highly parallelized since the sequential coefficient is relatively small (it only varies around 0.08). This beneficial property is an advantage of the proposed method for hardware scalability since the calculation of descriptor can be effectively sped up thanks to the parallelizing mechanism with the involved threads.

4.5. Assessments of MSIOMF$_{\sigma}^{k,\mathcal{D}^4}$ *and* MSVOMF$_{\sigma}^{k,\mathcal{D}^4}$

We thoroughly discuss the significant effectiveness of taking high-order oriented magnitudes into account DT representation in comparison with other Gaussian-based filtered features. Based on the experimental results in Tables 6, 7, 8, and 9, it could be stated the following crucial assessments:

- Firstly, to prove the validation of our proposal, we have also implemented other local DT descriptors, named IMF$_{\sigma}^{k}$ and VMF$_{\sigma}^{k}$, that are correspondingly based on the 2D/3D non-oriented magnitudes of Gaussian gradients (i.e., non-decomposing models involved in). It can be seen from Tables 3, 8, and 9 that IMF$_{\sigma}^{k}$ and VMF$_{\sigma}^{k}$ are not generally efficient compared to taking advantage of their oriented ones.
- Instead of exploiting Gaussian-based filtered characteristics as done in FoSIG [40] and V-BIG [39], taking the high-order oriented magnitudes into account DT representation has significantly improved the discrimination power (see Tables 10 and 11).

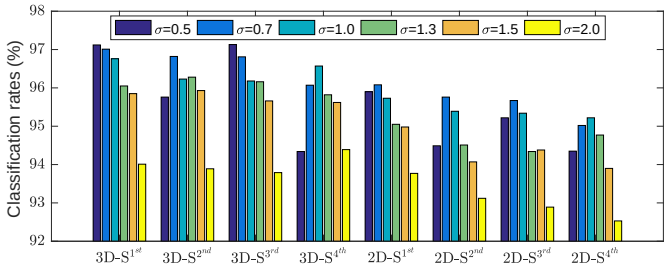

Fig. 9. (Best viewed in color) Performances on DynTex++ of high-order MSIOMF$_{\sigma}^{k,\mathcal{D}^4}$ **and** MSVOMF$_{\sigma}^{k,\mathcal{D}^4}$ **descriptors (represented by** 2D-S^k **and** 3D-S^k **respectively) are sharply decreased when the higher level of standard deviation σ is used for the gradient filterings.**

- The higher level of standard deviation σ is used for the Gaussian-gradient filterings, the less robustness of our MSIOMF$_{\sigma}^{k,\mathcal{D}^4}$ and MSVOMF$_{\sigma}^{k,\mathcal{D}^4}$ descriptors is mostly achieved. Absolutely, it can be verified in Fig. 9 that with an increase of σ from 0.5 to 2, their performances on DynTex++ dataset are decreased about from 1% to 3% in general. This is due to lack of appearance features caused by the Gaussian-gradient filterings with large levels of σ.
- Decomposing the Gaussian-gradient filtered outcomes in the same ranges of direction, the obtained MSVOM features are more discriminative than the MSIOM ones (see Fig. 10 for a graphical view of those in settings of $\mathcal{D}^4$ and $\sigma = 1.3$, see Table 6 for other circumstances in general). This is because there are complements from the intensification of pairs of gradients in the MSVOM decomposition.
- It can be found out that for the challenging datasets (i.e., *DynTex35*, *Beta*, *Gamma*), the proposed descriptors with the odd derivatives often give better effectiveness of DT classification (see Tables 6 and 7). Therefore, they should be nominated for applications in practice.
- As expected in Section 3.2, the multi-analysis has significantly improved the discrimination power. Indeed, it can be seen from Tables 6 and 7 that using 2-scale of Gaussian filterings with different standard deviations, the abilities of MSIOMF$_{\{\sigma\}}^{k,\mathcal{D}^4}$ and MSVOMF$_{\{\sigma\}}^{k,\mathcal{D}^4}$ are enhanced and more "stable" than those of the single-scale. Also, the 2-order descriptors are better than the single-order ones (see Tables 6 and 8). Furthermore, an incorporation of 2-scale and 2-order features points out the best (see Table 9).

Consequently, based on the effectiveness of MSIOMF$_{\{\sigma\}}^{\{k\},\mathcal{D}^4}$ and MSVOMF$_{\{\sigma\}}^{\{k\},\mathcal{D}^4}$ in classifying DTs, the settings of those: MSIOMF$_{\{0.5,1.0\}}^{\{1^{st},2^{nd}\},\mathcal{D}^4}$ and MSVOMF$_{\{0.7,1.0\}}^{\{1^{st},4^{th}\},\mathcal{D}^4}$ should be recommended for real applications as well as for comprehensive comparison with recent methods due to their best performances. In further evaluations, if parameters of the MSIOM/MSVOM-based descriptors are not explicit, these default settings are mentioned.

4.6. Comprehensive comparison to shallow methods

Classification on UCLA: It can be seen from Tables 6, 7, 8, and 9 that our MSIOM/MSVOM-based descriptors achieve very good rates on the schemes of UCLA. Therein, thanks to

Table 6. Classification rates (%) on DT benchmark datasets of $\text{MSIOMF}_{\sigma}^{k,\mathcal{D}^4}$ and $\text{MSVOMF}_{\sigma}^{k,\mathcal{D}^4}$ descriptors.

Dataset		UCLA								DynTex								DynTex++	
Sub-set		50-LOO		50-4fold		9-class		8-class		DynTex35		Alpha		Beta		Gamma			
Order	$\{\sigma_i\}$	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S
1^{st}	{0.5}	99.50	**100**	99.00	99.50	98.90	98.55	96.74	98.80	98.29	95.43	96.67	96.67	90.74	**95.68**	92.42	93.94	95.90	97.12
	{0.7}	99.00	99.50	99.00	99.50	**99.70**	99.60	96.63	97.50	98.29	96.57	95.00	96.67	90.74	93.21	93.94	94.32	**96.08**	97.01
	{1.0}	99.50	99.50	**100**	99.50	99.30	98.75	97.50	97.17	98.86	98.29	96.67	96.67	91.36	92.59	92.80	93.18	95.73	96.76
	{1.3}	98.50	99.00	98.50	99.50	99.05	97.65	95.11	97.39	98.57	98.86	96.67	96.67	91.36	92.59	92.80	92.42	95.05	96.05
	{1.5}	99.00	99.00	99.50	99.50	96.85	98.70	96.85	98.04	99.14	98.86	96.67	96.67	91.36	91.98	92.05	92.05	94.98	95.85
	{2.0}	**100**	99.50	**100**	99.50	98.75	99.35	96.74	98.04	98.57	98.86	**98.33**	96.67	91.98	91.98	91.67	93.18	93.77	94.01
2^{nd}	{0.5}	**100**	99.00	**100**	99.50	97.15	98.75	95.87	97.83	97.71	98.00	96.67	**98.33**	91.36	90.12	89.77	88.64	94.49	95.76
	{0.7}	**100**	**100**	**100**	**100**	98.90	99.40	97.28	98.04	98.00	97.14	96.67	96.67	93.83	94.44	93.56	93.18	95.76	96.82
	{1.0}	99.50	**100**	99.00	**100**	98.60	99.00	98.49	97.39	98.57	97.71	96.67	96.67	93.21	95.06	93.18	93.56	95.39	96.23
	{1.3}	99.50	**100**	99.50	**100**	99.25	98.70	98.15	97.07	97.71	98.57	96.67	96.67	91.36	93.83	93.56	93.94	94.51	96.28
	{1.5}	99.00	99.00	99.00	99.00	98.10	99.35	99.02	98.04	98.86	97.43	96.67	96.67	92.59	93.21	92.80	93.18	94.07	95.93
	{2.0}	99.00	**100**	99.00	**100**	98.60	98.50	97.07	97.93	97.71	97.71	96.67	96.67	91.36	92.59	93.18	95.08	93.12	93.89
3^{rd}	{0.5}	99.50	99.50	**100**	99.00	99.10	99.15	97.61	98.04	98.29	98.29	96.67	**98.33**	**95.06**	92.59	92.80	91.29	95.22	**97.13**
	{0.7}	99.00	99.50	99.50	99.50	98.40	98.90	97.72	97.39	98.86	98.86	96.67	96.67	89.51	92.59	**93.94**	93.18	95.67	96.81
	{1.0}	**100**	**100**	**100**	99.50	98.30	98.45	99.13	97.50	**99.71**	**99.43**	96.67	96.67	92.59	92.59	93.18	91.29	95.34	96.18
	{1.3}	**100**	**100**	**100**	**100**	98.45	99.05	94.67	96.74	98.57	98.57	96.67	96.67	93.21	93.83	91.29	93.18	94.34	96.16
	{1.5}	99.00	99.00	99.00	99.50	98.55	98.40	96.30	97.17	98.86	**99.43**	96.67	96.67	91.98	93.83	91.29	91.67	94.38	95.66
	{2.0}	99.50	98.50	99.50	99.50	98.70	98.45	98.49	96.20	98.00	98.86	96.67	96.67	93.21	93.21	92.05	93.18	92.89	93.79
4^{th}	{0.5}	**100**	99.50	**100**	99.50	96.35	97.80	96.96	97.07	96.29	96.29	96.67	96.67	91.36	90.12	90.53	89.39	94.35	94.34
	{0.7}	99.00	**100**	99.50	**100**	97.95	98.65	98.04	98.70	98.29	98.86	96.67	96.67	93.83	93.83	93.18	93.94	95.02	96.07
	{1.0}	99.50	**100**	**100**	**100**	98.65	98.85	98.80	98.04	92.86	97.14	96.67	96.67	91.36	95.06	90.53	94.32	95.22	96.57
	{1.3}	99.00	**100**	99.00	**100**	98.55	97.85	97.83	99.02	96.29	97.43	96.67	96.67	90.74	94.44	91.29	93.56	94.77	95.82
	{1.5}	99.50	99.50	99.50	99.50	98.45	**99.80**	**99.35**	98.49	98.00	96.57	96.67	96.67	91.98	94.44	92.42	94.32	93.90	95.62
	{2.0}	99.50	**100**	99.50	**100**	98.50	99.20	98.59	**99.24**	93.71	97.43	96.67	96.67	91.36	95.06	92.42	**95.45**	92.53	94.39

Note: 2D-S and 3D-S are shortened for $\text{MSIOMF}_{\sigma}^{k,\mathcal{D}^4}$ and $\text{MSVOMF}_{\sigma}^{k,\mathcal{D}^4}$ respectively. 50-LOO and 50-4fold denote results on 50-class breakdown using leave-one-out and four cross-fold validation.

Table 7. Classification rates (%) on DT benchmark datasets of $\text{MSIOMF}_{\{\sigma\}}^{k,\mathcal{D}^4}$ and $\text{MSVOMF}_{\{\sigma\}}^{k,\mathcal{D}^4}$ descriptors.

Dataset		UCLA								DynTex								DynTex++	
Sub-set		50-LOO		50-4fold		9-class		8-class		DynTex35		Alpha		Beta		Gamma			
Order	$\{\sigma_i\}$	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S
1^{st}	{0.5, 0.7}	99.00	99.50	99.50	**100**	97.95	98.85	95.22	96.41	98.57	98.00	95.00	96.67	93.83	95.06	93.18	93.18	93.77	97.36
	{0.5, 1.0}	99.50	99.50	99.50	99.50	99.20	98.80	**99.02**	97.50	99.14	99.43	96.67	96.67	93.83	94.44	92.42	93.56	96.67	97.28
	{0.7, 1.0}	99.00	99.50	99.50	99.50	99.25	98.10	97.72	98.48	99.14	98.86	95.00	96.67	92.59	93.83	92.80	93.18	96.72	96.78
	{1.0, 1.3}	98.50	99.50	99.00	99.50	98.90	98.65	98.04	96.41	**99.71**	98.86	96.67	96.67	91.98	93.83	92.42	93.18	96.43	96.89
	{1.0, 1.5}	99.00	99.50	99.50	99.50	98.25	98.85	97.50	97.50	99.43	98.86	96.67	96.67	91.36	92.59	92.05	92.80	96.27	96.88
	{1.5, 2.0}	99.50	99.50	99.50	99.50	98.15	99.15	96.85	96.85	99.43	98.86	**98.33**	96.67	93.21	91.98	91.67	92.05	95.51	95.81
2^{nd}	{0.5, 0.7}	**100**	**100**	**100**	**100**	98.85	97.75	96.52	97.50	98.57	98.86	96.67	96.67	93.83	93.83	93.56	93.94	96.66	**97.37**
	{0.5, 1.0}	**100**	**100**	**100**	**100**	98.45	98.50	97.28	97.39	97.14	98.00	96.67	96.67	91.98	93.21	93.94	93.56	96.63	97.08
	{0.7, 1.0}	**100**	**100**	**100**	**100**	99.15	98.65	97.17	97.39	98.86	97.43	96.67	96.67	93.83	93.83	92.80	93.56	96.45	97.08
	{1.0, 1.3}	99.50	**100**	99.50	**100**	**99.35**	99.00	96.20	97.83	98.29	98.00	96.67	96.67	92.59	93.83	93.18	93.94	95.91	96.45
	{1.0, 1.5}	99.50	99.50	98.50	**100**	98.80	99.00	98.26	99.02	99.43	92.57	96.67	96.67	93.21	93.21	93.56	93.18	96.39	96.67
	{1.5, 2.0}	98.50	**100**	98.00	**100**	98.90	98.95	98.70	98.04	98.29	97.71	96.67	96.67	92.59	91.98	92.80	94.70	94.86	95.50
3^{rd}	{0.5, 0.7}	99.50	99.50	99.50	99.50	98.90	98.70	97.39	98.37	98.57	99.14	96.67	96.67	92.59	92.59	93.56	93.18	96.69	97.32
	{0.5, 1.0}	**100**	**100**	**100**	99.50	99.25	98.15	97.61	97.72	99.43	**99.71**	96.67	96.67	**94.44**	93.21	93.56	91.67	**96.72**	97.06
	{0.7, 1.0}	99.50	99.50	99.50	99.50	98.65	98.70	97.83	97.28	99.14	99.43	96.67	96.67	91.98	92.59	92.80	92.04	96.36	97.25
	{1.0, 1.3}	**100**	**100**	**100**	**100**	99.25	98.20	95.87	98.59	98.86	99.14	96.67	96.67	93.83	93.83	93.56	93.94	96.33	96.76
	{1.0, 1.5}	99.00	**100**	99.50	99.50	98.30	97.90	97.72	97.61	99.43	**99.71**	96.67	96.67	92.59	93.83	90.91	91.67	96.26	96.48
	{1.5, 2.0}	99.50	99.00	99.50	99.50	98.15	**99.40**	96.63	96.85	98.00	99.14	96.67	96.67	91.98	93.83	90.53	92.80	94.24	95.24
4^{th}	{0.5, 0.7}	**100**	**100**	**100**	**100**	97.20	98.30	97.50	97.50	97.14	97.43	96.67	96.67	91.98	91.36	**94.32**	92.80	96.29	96.80
	{0.5, 1.0}	**100**	**100**	**100**	**100**	98.90	98.40	98.48	97.72	96.86	98.29	96.67	96.67	93.21	93.21	93.18	93.56	94.47	96.88
	{0.7, 1.0}	99.50	**100**	99.50	**100**	98.20	99.05	97.39	**99.57**	98.29	97.71	96.67	96.67	93.83	94.44	92.42	94.70	96.19	96.93
	{1.0, 1.3}	99.50	**100**	99.50	**100**	98.85	98.80	98.70	98.91	95.43	98.57	96.67	96.67	91.98	**96.30**	92.05	93.94	96.17	96.81
	{1.0, 1.5}	99.50	**100**	**100**	**100**	98.65	99.15	98.26	98.70	97.43	96.86	96.67	96.67	92.59	94.44	92.80	**95.08**	95.62	96.24
	{1.5, 2.0}	99.50	**100**	99.50	**100**	98.90	99.25	97.93	99.15	98.29	96.86	96.67	96.67	91.36	93.83	93.18	94.70	94.68	95.61

Note: 2D-S and 3D-S are shortened for $\text{MSIOMF}_{\{\sigma\}}^{k,\mathcal{D}^4}$ and $\text{MSVOMF}_{\{\sigma\}}^{k,\mathcal{D}^4}$ respectively. 50-LOO and 50-4fold denote results on 50-class breakdown using leave-one-out and four cross-fold validation.

exploiting more oriented magnitudes from pairs of Gaussian-gradients (see Section 3.1), the MSVOM-based ones have the performances in more "stability". With respect to the settings for comparison, our results are mostly the best in comparison with all current methods (see Table 10). Particularly, both MSIOMF and MSVOMF obtain the best rate of 100% on both schemes *50-LOO* and *50-4fold*. In the meantime, $\text{MSIOMF}_{\{0.5,1.0\}}^{\{1^{st},2^{nd}\}}$ obtains 99.00% and 98.59% for DT classification on *9-class* and *8-class* breakdowns respectively, while $\text{MSVOMF}_{\{0.7,1.0\}}^{\{1^{st},4^{th}\}}$ achieves better rates of 99.35% for both these breakdowns (see Fig. 11 for its specific confusions on *9-class*, and Fig. 12 on *8-class*). In addition, the highest rate on *9-class* is 99.80% obtained by $\text{MSVOMF}_{\{1.5\}}^{4^{th}}$ (see Table 6); on *8-class*

Table 8. Classification rates (%) on DT benchmark datasets of multi-order $\text{MSIOMF}_{\sigma}^{\{k\},\mathcal{D}^4}$ and $\text{MSVOMF}_{\sigma}^{\{k\},\mathcal{D}^4}$ descriptors.

Dataset		UCLA								DynTex								DynTex++	
Sub-set		50-LOO		50-4fold		9-class		8-class		DynTex35		Alpha		Beta		Gamma			
Multi-order	$\{\sigma_i\}$	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S
$\{1^{st}, 2^{nd}\}$	{0.5}	**100**	**100**	**100**	**100**	97.65	98.35	97.61	96.63	97.71	98.57	96.67	96.67	92.59	95.68	93.56	93.18	96.61	97.07
	{0.7}	**100**	**100**	**100**	**100**	97.95	**99.70**	96.74	**99.57**	98.57	98.86	96.67	96.67	93.21	95.06	**94.70**	95.08	97.04	97.18
	{1.0}	99.50	99.50	99.50	99.50	99.20	98.25	97.72	98.15	98.86	98.57	96.67	96.67	93.21	94.44	**94.70**	95.08	97.07	97.28
	{1.3}	99.50	**100**	99.50	**100**	99.35	99.40	95.98	98.80	97.71	**99.71**	96.67	96.67	91.98	93.21	92.42	93.56	96.57	96.78
	{1.5}	99.00	**100**	99.50	99.50	**99.65**	97.95	97.39	98.04	99.14	99.14	96.67	96.67	91.98	92.59	93.18	95.45	96.24	96.44
	{2.0}	**100**	**100**	**100**	99.50	99.15	99.10	96.74	98.26	98.86	99.14	96.67	96.67	93.21	92.59	93.56	95.45	94.94	95.67
$\{1^{st}, 3^{rd}\}$	{0.5}	**100**	99.50	**100**	**100**	99.15	98.45	98.15	98.59	98.86	**99.71**	96.67	96.67	93.83	95.06	92.42	92.05	96.68	97.25
	{0.7}	99.50	99.50	99.50	99.50	99.10	99.40	97.61	98.70	98.86	98.86	95.00	96.67	90.74	92.59	92.80	93.94	96.34	97.48
	{1.0}	99.50	**100**	99.50	99.00	98.35	98.55	97.07	98.04	99.43	98.29	96.67	96.67	92.59	92.59	92.80	92.42	96.48	97.17
	{1.3}	**100**	**100**	**100**	**100**	98.40	99.15	98.15	98.59	99.43	99.14	96.67	96.67	93.21	93.83	92.05	95.08	96.27	96.34
	{1.5}	99.00	99.00	99.50	99.50	98.70	98.25	98.37	98.80	**99.71**	99.14	96.67	96.67	92.59	93.21	91.29	91.29	95.77	96.32
	{2.0}	99.50	99.00	99.50	99.50	98.70	98.90	97.07	97.61	98.29	99.14	96.67	96.67	91.98	92.59	92.80	93.56	94.93	95.07
$\{1^{st}, 4^{th}\}$	{0.5}	**100**	**100**	**100**	**100**	97.95	98.95	96.63	97.28	96.57	97.71	96.67	96.67	93.21	93.21	93.56	93.18	96.66	97.08
	{0.7}	**100**	**100**	**100**	**100**	98.30	98.40	99.02	97.50	98.29	**99.71**	96.67	96.67	94.44	94.44	**94.70**	94.32	96.89	97.28
	{1.0}	**100**	**100**	**100**	**100**	98.30	98.25	99.02	99.46	98.29	**99.71**	**98.33**	96.67	92.59	95.06	94.32	94.70	97.08	97.32
	{1.3}	**100**	**100**	**100**	**100**	99.30	98.85	97.93	97.93	98.00	98.86	96.67	96.67	91.98	93.83	93.94	94.32	96.28	97.05
	{1.5}	**100**	**100**	**100**	**100**	97.80	98.55	97.93	97.07	98.86	98.57	96.67	96.67	92.59	93.21	94.32	**95.83**	96.44	96.97
	{2.0}	**100**	**100**	**100**	99.50	98.70	98.65	98.48	98.59	98.57	98.86	96.67	96.67	91.98	94.44	93.94	**95.83**	95.52	95.92
$\{2^{nd}, 3^{rd}\}$	{0.5}	**100**	99.50	99.50	99.50	97.95	99.30	98.26	96.63	98.86	98.00	96.67	96.67	93.83	93.83	92.05	93.18	96.58	97.08
	{0.7}	**100**	**100**	**100**	99.50	98.30	98.35	97.17	98.26	98.57	98.86	96.67	96.67	91.98	93.83	93.94	95.08	**97.21**	**97.59**
	{1.0}	**100**	**100**	**100**	**100**	98.95	97.95	97.50	99.13	99.14	98.57	96.67	96.67	93.21	93.21	**94.70**	95.45	96.93	97.27
	{1.3}	**100**	**100**	**100**	**100**	98.35	99.55	97.50	97.83	97.71	99.14	96.67	96.67	93.21	93.83	92.05	94.70	95.76	96.50
	{1.5}	98.50	99.50	98.50	99.50	99.05	98.25	96.41	97.93	99.43	99.43	96.67	96.67	90.74	91.98	93.56	94.32	95.87	96.25
	{2.0}	**100**	**100**	**100**	99.50	98.05	99.00	98.37	96.85	98.57	99.43	96.67	96.67	91.98	93.21	93.94	94.70	94.75	95.69
$\{2^{nd}, 4^{th}\}$	{0.5}	**100**	99.50	**100**	99.50	96.40	97.90	96.30	97.17	97.43	97.14	**98.33**	96.67	92.59	90.12	91.29	89.77	95.45	96.13
	{0.7}	**100**	**100**	**100**	**100**	97.70	98.85	98.70	98.37	98.57	98.00	96.67	96.67	**95.06**	94.44	93.94	95.08	96.44	97.29
	{1.0}	**100**	**100**	**100**	**100**	99.00	**99.70**	98.26	99.13	97.43	96.86	96.67	96.67	93.21	**96.30**	93.18	95.08	96.32	96.92
	{1.3}	**100**	**100**	**100**	**100**	98.60	98.95	**99.24**	97.72	98.00	98.86	96.67	96.67	91.36	93.83	91.29	94.70	95.96	96.66
	{1.5}	99.50	**100**	99.50	**100**	98.70	98.95	97.17	98.59	99.14	97.14	96.67	96.67	92.59	94.44	93.18	93.94	95.35	96.13
	{2.0}	99.50	**100**	99.50	**100**	98.20	99.15	97.72	97.39	98.00	96.86	96.67	96.67	91.98	94.44	93.18	95.08	94.74	95.13
$\{3^{rd}, 4^{th}\}$	{0.5}	**100**	99.50	**100**	99.50	98.25	98.90	98.04	97.93	97.71	98.86	96.67	96.67	92.59	93.83	91.29	93.18	96.21	97.00
	{0.7}	**100**	**100**	**100**	**100**	98.80	98.50	98.04	**99.57**	98.86	99.14	96.67	96.67	93.83	95.06	93.56	94.32	96.72	97.36
	{1.0}	**100**	**100**	**100**	**100**	99.10	99.50	98.91	98.48	99.14	99.14	96.67	96.67	93.21	93.83	94.32	94.70	96.77	97.07
	{1.3}	**100**	**100**	**100**	**100**	98.95	98.75	99.02	99.35	98.00	98.57	96.67	96.67	93.83	94.44	92.80	94.70	96.14	96.84
	{1.5}	99.50	**100**	99.00	99.50	98.50	98.55	98.59	98.91	98.57	98.57	96.67	96.67	91.98	94.44	93.94	94.70	96.06	96.88
	{2.0}	99.50	**100**	99.50	99.50	98.75	99.60	98.70	98.80	98.00	98.57	96.67	96.67	91.98	93.83	**94.70**	95.45	94.90	95.88

Note: 2D-S and 3D-S are shortened for $\text{MSIOMF}_{\sigma}^{\{k\},\mathcal{D}^4}$ and $\text{MSVOMF}_{\sigma}^{\{k\},\mathcal{D}^4}$ respectively. 50-LOO and 50-4fold denote results on 50-class breakdown using leave-one-out and four cross-fold validation.

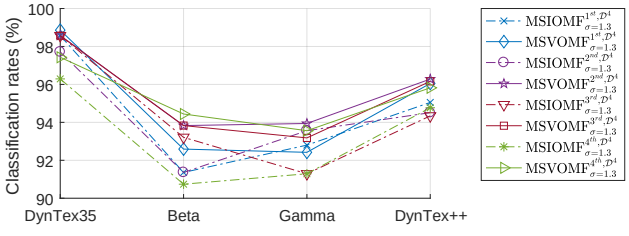

Fig. 10. (Best viewed in color) A comprehensive comparison in pairs of high-order $\text{MSIOMF}_{\sigma=1.3}^{k,\mathcal{D}^4}$ and $\text{MSVOMF}_{\sigma=1.3}^{k,\mathcal{D}^4}$ descriptors.

is 99.57% obtained by $\text{MSVOMF}_{\{0.7,1.0\}}^{4^{th}}$, $\text{MSVOMF}_{\{0.7\}}^{\{1^{st},2^{nd}\}}$, and $\text{MSVOMF}_{\{0.7\}}^{\{3^{rd},4^{th}\}}$ (see Tables 7 and 8). It is noteworthy that FD-MAP [22] (99.35%, 99.57%), DNGP [21] (99.6%, 99.4%), and CVLBC [61] (99.2%, 99.02%) nearly have the same our abilities on these two breakdowns. However, CVLBC and FD-MAP are inferior to ours in classifying DTs on *50-LOO* and *50-4fold* of UCLA (see Table 10), as well as not better than ours on subsets of DynTex and on DynTex++ (see Table 11). Moreover, CVLBC and DNGP have not been verified on other challenging subsets, i.e., *Alpha*, *Beta*, and *Gamma*. In respect of comparing with Gaussian-based descriptors (i.e., V-BIG [39] and FoSIG [40]), our proposal has prominent results (see Table 10). This has proved the interest of oriented magnitudes instead of purely exploiting Gaussian-filtered features for DT representation.

Classification on DynTex: It can be verified from Table 11 that in general, our MSIOMF and MSVOMF descriptors mostly have the best performances in comparison with all non-deep-learning approaches. Specifically, our MSVOMF descriptor just reaches at 99.71% rate of DT recognition on *DynTex35* due to a mutual confusion between two classes of very similar DT motions, as highlighted in red in Fig. 13. This result is just a little lower than CSAP-TOP's [38] (100%). However, beside a larger dimension (13200 bins), CSAP-TOP is also not better than ours on the other sub-sets of this schema (i.e., *Alpha*, *Beta*, and *Gamma*), as well as on UCLA (see Table 10). In terms of classifying DTs on other challenging schemes, due to two confusions in Fig. 14, our MSVOMF obtains 96.67% on *Alpha*, about 3.3% lower than V-BIG [39] with rate of 100%. However, in the other schemes, V-BIG [39] does not perform in stability (see Tables 10 and 11). In the meanwhile, performances of 96.3% on *Beta* and 95.08% on *Gamma* are the very good rates in comparison with all shallow methods (see Tables 10 and 11). It is noteworthy that recently, two local-feature-based methods RUBIG (95.68%) [49] and MEMDP (96.91%) [50] are

Table 9. Classification rates (%) on DT benchmark datasets of multi-order MSIOMF$^{\{k\},\mathcal{D}^4}_{\{\sigma\}}$ and MSVOMF$^{\{k\},\mathcal{D}^4}_{\{\sigma\}}$ descriptors.

Dataset		UCLA								DynTex								DynTex++	
Sub-set		50-LOO		50-4fold		9-class		8-class		DynTex35		Alpha		Beta		Gamma			
Multi-order	$\{\sigma_i\}$	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S	2D-S	3D-S
$\{1^{st}, 2^{nd}\}$	{0.5, 0.7}	**100**	**100**	**100**	**100**	98.55	97.70	96.20	97.07	98.00	99.43	96.67	96.67	93.83	95.06	93.94	94.32	97.46	97.57
	{0.5, 1.0}	**100**	**100**	**100**	**100**	99.00	98.15	98.59	98.70	99.14	99.14	96.67	96.67	**95.68**	94.44	94.70	93.56	97.29	97.73
	{0.7, 1.0}	**100**	99.50	**100**	**100**	98.30	99.40	99.02	99.02	98.86	98.86	96.67	96.67	93.21	95.06	95.45	95.45	97.29	97.42
	{1.0, 1.3}	**100**	**100**	**100**	**100**	98.00	99.00	96.74	97.83	99.43	**99.71**	96.67	96.67	92.59	93.21	95.08	95.45	97.32	97.61
	{1.0, 1.5}	99.50	99.50	99.00	99.50	98.00	98.05	97.28	97.17	**99.71**	98.86	96.67	96.67	93.21	93.83	95.08	95.45	97.22	97.26
	{1.5, 2.0}	**100**	**100**	**100**	99.50	98.40	99.30	98.70	98.26	99.43	99.43	96.67	96.67	93.21	92.59	94.70	**95.83**	96.31	96.71
$\{1^{st}, 3^{rd}\}$	{0.5, 0.7}	99.00	99.50	99.50	99.50	98.95	98.65	97.50	96.96	99.14	99.14	95.00	96.67	91.98	95.06	93.56	92.42	97.44	97.40
	{0.5, 1.0}	**100**	99.50	99.50	99.50	**99.45**	97.95	96.74	97.83	99.43	99.14	96.67	96.67	95.06	94.44	92.05	92.42	97.19	97.43
	{0.7, 1.0}	99.50	99.50	99.50	99.50	98.20	98.95	98.59	99.13	**99.71**	98.86	96.67	96.67	91.36	93.83	91.67	92.80	97.10	97.27
	{1.0, 1.3}	**100**	**100**	99.50	**100**	98.10	98.75	97.93	98.37	99.43	99.43	96.67	96.67	92.59	94.44	93.18	93.56	97.22	97.17
	{1.0, 1.5}	99.00	99.50	99.50	99.50	98.45	98.70	97.61	97.39	**99.71**	**99.71**	96.67	96.67	93.21	92.59	91.67	91.67	96.91	97.15
	{1.5, 2.0}	99.50	99.50	99.50	99.50	98.20	98.65	96.30	97.07	98.86	99.43	96.67	96.67	90.74	92.59	91.67	92.05	95.68	96.07
$\{1^{st}, 4^{th}\}$	{0.5, 0.7}	**100**	**100**	**100**	**100**	98.70	98.35	97.61	96.74	97.14	98.29	96.67	96.67	93.21	95.06	**95.83**	93.56	97.19	97.36
	{0.5, 1.0}	**100**	**100**	**100**	**100**	98.45	98.80	97.83	97.93	98.00	99.43	**98.33**	96.67	94.44	95.68	95.08	93.94	97.26	97.76
	{0.7, 1.0}	**100**	**100**	**100**	**100**	99.20	99.35	98.80	**99.35**	98.00	**99.71**	96.67	96.67	94.44	**96.30**	95.45	95.08	**97.57**	**97.87**
	{1.0, 1.3}	**100**	**100**	**100**	**100**	97.75	98.80	97.61	97.17	98.86	**99.71**	96.67	96.67	91.98	95.06	95.08	95.45	97.26	97.29
	{1.0, 1.5}	**100**	**100**	**100**	**100**	99.30	**99.55**	**99.13**	98.04	98.86	**99.71**	96.67	96.67	92.59	93.21	93.56	95.08	97.22	97.44
	{1.5, 2.0}	**100**	**100**	**100**	99.50	99.00	98.75	97.17	99.13	98.86	99.14	96.67	96.67	91.98	93.21	93.56	95.45	96.75	96.88
$\{2^{nd}, 3^{rd}\}$	{0.5, 0.7}	**100**	99.50	**100**	99.50	96.85	98.25	97.72	96.52	98.86	99.43	96.67	96.67	93.21	95.06	93.94	93.94	97.23	97.73
	{0.5, 1.0}	**100**	99.50	**100**	99.50	98.70	98.75	96.30	98.59	99.43	**99.71**	96.67	96.67	94.44	93.83	93.18	93.56	97.35	97.64
	{0.7, 1.0}	**100**	**100**	**100**	99.50	98.95	98.45	98.37	97.39	98.86	99.14	96.67	96.67	92.59	94.44	94.70	94.32	97.37	97.52
	{1.0, 1.3}	**100**	**100**	**100**	**100**	98.70	99.30	98.04	97.83	98.86	98.86	96.67	96.67	93.83	93.83	94.70	95.08	97.17	97.32
	{1.0, 1.5}	99.50	**100**	99.00	**100**	98.70	98.30	97.39	98.70	99.43	99.43	96.67	96.67	92.59	92.59	95.08	**95.83**	97.20	97.35
	{1.5, 2.0}	99.50	99.50	99.50	99.50	98.25	99.35	98.59	98.04	98.86	99.43	96.67	96.67	90.74	93.21	93.94	95.08	96.06	96.53
$\{2^{nd}, 4^{th}\}$	{0.5, 0.7}	**100**	**100**	**100**	**100**	98.20	97.50	97.28	97.83	98.00	98.57	96.67	96.67	91.36	92.59	94.70	94.70	96.78	97.17
	{0.5, 1.0}	**100**	**100**	**100**	**100**	97.50	97.70	97.61	96.41	98.00	97.71	96.67	96.67	92.59	92.59	94.32	93.94	96.95	97.16
	{0.7, 1.0}	**100**	**100**	**100**	**100**	98.95	98.65	97.50	99.02	98.57	97.71	96.67	96.67	94.44	95.06	94.32	94.70	96.67	97.36
	{1.0, 1.3}	**100**	**100**	**100**	**100**	98.35	98.95	98.04	98.48	97.71	97.71	96.67	96.67	93.21	95.06	92.80	95.08	96.89	97.08
	{1.0, 1.5}	**100**	**100**	**100**	**100**	98.70	98.75	98.59	98.59	98.86	97.14	96.67	96.67	92.59	93.21	93.18	95.08	96.56	96.89
	{1.5, 2.0}	99.50	**100**	99.50	**100**	98.25	99.40	96.20	98.15	99.14	97.43	96.67	96.67	91.36	94.44	93.18	95.08	95.65	96.49
$\{3^{rd}, 4^{th}\}$	{0.5, 0.7}	**100**	99.50	**100**	99.50	98.50	98.95	97.17	97.28	98.57	99.14	96.67	96.67	92.59	95.06	95.08	93.94	97.29	97.47
	{0.5, 1.0}	**100**	**100**	**100**	**100**	97.35	98.70	98.48	98.15	98.86	**99.71**	96.67	96.67	94.44	95.68	94.70	93.94	97.26	97.43
	{0.7, 1.0}	**100**	**100**	**100**	**100**	98.95	98.85	97.61	98.91	99.14	99.43	96.67	96.67	93.83	95.68	94.70	94.32	97.12	97.78
	{1.0, 1.3}	**100**	**100**	**100**	**100**	98.45	99.30	98.15	98.80	99.14	99.43	96.67	96.67	93.83	95.68	94.32	94.70	97.27	97.32
	{1.0, 1.5}	**100**	**100**	99.50	**100**	98.70	98.80	96.74	98.70	99.14	**99.71**	96.67	96.67	92.59	95.06	94.70	95.45	96.98	97.31
	{1.5, 2.0}	99.50	**100**	99.50	99.50	98.75	98.35	98.80	**99.35**	97.43	99.43	96.67	96.67	91.98	94.44	94.32	**95.83**	96.16	96.69

Note: 2D-S and 3D-S are shortened for MSIOMF$^{\{k\},\mathcal{D}^4}_{\{\sigma\}}$ and MSVOMF$^{\{k\},\mathcal{D}^4}_{\{\sigma\}}$ respectively. 50-LOO and 50-4fold denote results on 50-class breakdown using leave-one-out and four cross-fold validation.

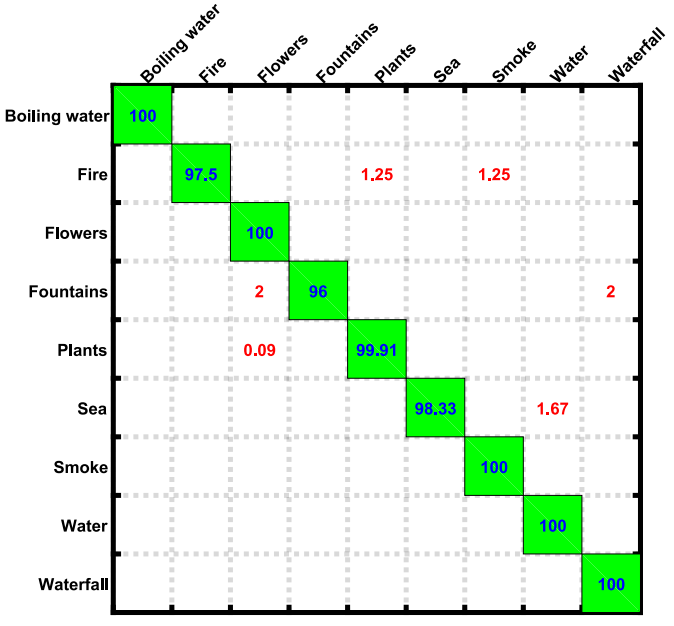

Fig. 11. Confusion matrix (%) for MSVOMF$^{\{1^{st},4^{th}\},\mathcal{D}^4}_{\{0.7,1.0\}}$ on *9-class*.

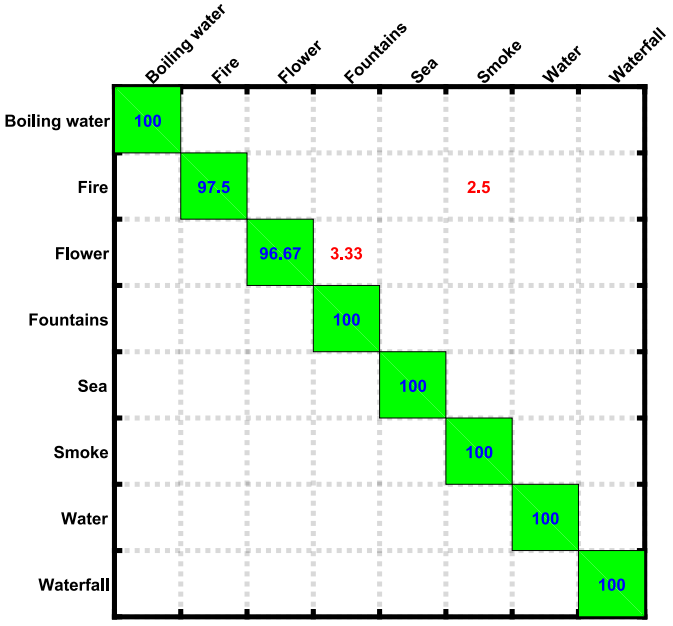

Fig. 12. Confusion matrix (%) for MSVOMF$^{\{1^{st},4^{th}\},\mathcal{D}^4}_{\{0.7,1.0\}}$ on *8-class*.

Table 10. Comparison of recognition rates (%) on UCLA.

Group	Encoding method	50-LOO	50-4fold	9-class	8-class
A	FDT [22]	98.50	99.00	97.70	99.35
	FD-MAP [22]	**99.50**	99.00	99.35	**99.57**
	DDTP [64]	99.00	99.50	98.75	98.04
B	AR-LDS [15]	89.90^{N}	-	-	-
	Chaotic vector [26]	-	-	85.10^{N}	85.00^{N}
C	3D-OTF [11]	-	87.10	97.23	99.50
	DFS [65]	-	**100**	97.50	99.20
	STLS [13]	-	99.50	97.40	99.50
D	MBSIF-TOP [35]	**99.50**^{N}	-	-	-
	DNGP [21]	-	-	**99.60**	99.40
	B3DF_SMC [36]	99.50^{N}	99.50^{N}	98.85^{N}	98.15^{N}
E	VLBP [42]	-	89.50^{N}	96.30^{N}	91.96^{N}
	LBP-TOP [42]	-	94.50^{N}	96.00^{N}	93.67^{N}
	CVLBP [44]	-	93.00^{N}	96.90^{N}	95.65^{N}
	HLBP [45]	95.00^{N}	95.00^{N}	98.35^{N}	97.50^{N}
	CLSP-TOP [46]	99.00^{N}	99.00^{N}	98.60^{N}	97.72^{N}
	MEWLSP [58]	96.50^{N}	96.50^{N}	98.55^{N}	98.04^{N}
	WLBPC [60]	-	96.50^{N}	97.17^{N}	97.61^{N}
	CVLBC [61]	98.50^{N}	99.00^{N}	99.20^{N}	99.02^{N}
	CSAP-TOP [38]	99.50	99.50	96.80	95.98
	FoSIG [40]	99.50	**100**	98.95	98.59
	V-BIG [39]	99.50	99.50	97.95	97.50
	HILOP [66]	99.50	99.50	97.80	96.30
	$MMDP_{D\_M/C}$ [50]	**100**	**100**	98.70	98.70
	$MEMDP_{D\_M/C}$ [50]	**100**	**100**	98.90	98.70
	RUBIG [49]	**100**	**100**	99.20	99.13
	Our MSIOMF$^{\{1^{st},2^{nd}\},\mathcal{D}^4}_{\{0.5,1.0\}}$	**100**	**100**	99.00	98.59
	Our MSVOMF$^{\{1^{st},4^{th}\},\mathcal{D}^4}_{\{0.7,1.0\}}$	**100**	**100**	99.35	99.35
F	DL-PEGASOS [17]	-	97.50	95.60	-
	PI-LBP+super hist [48]	-	**100**^{N}	98.20^{N}	-
	Orthogonal Tensor DL [33]	-	99.80	98.20	99.50
	PCANet-TOP [31]	**99.50***	-	-	-
	DT-CNN-AlexNet [30]	-	99.50*	98.05*	98.48*
	DT-CNN-GoogleNet [30]	-	99.50*	98.35*	99.02*

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

Table 11. Comparison of rates (%) on DynTex and DynTex++.

Group	Encoding method	Dyn35	Alpha	Beta	Gamma	Dyn++
A	FDT [22]	98.86	98.33	93.21	91.67	95.31
	FD-MAP [22]	98.86	98.33	92.59	91.67	95.69
	DDTP [64]	99.71	96.67	93.83	91.29	95.09
C	3D-OTF [11]	96.70	83.61	73.22	72.53	89.17
	DFS [65]	97.16	85.24	76.93	74.82	91.70
	2D+T [57]	-	85.00	67.00	63.00	-
	STLS [13]	98.20	89.40	80.80	79.80	94.50
D	MBSIF-TOP [35]	98.61^{N}	90.00^{N}	90.70^{N}	91.30^{N}	97.12^{N}
	DNGP [21]	-	-	-	-	93.80
	B3DF_SMC [36]	99.71^{N}	95.00^{N}	90.12^{N}	90.91^{N}	95.58^{N}
E	VLBP [42]	81.14^{N}	-	-	-	94.98^{N}
	LBP-TOP [42]	92.45^{N}	98.33	88.89	84.85^{N}	94.05^{N}
	DDLBP with MJMI [47]	-	-	-	-	95.80
	CVLBP [44]	85.14^{N}	-	-	-	-
	HLBP [45]	98.57^{N}	-	-	-	96.28^{N}
	CLSP-TOP [46]	98.29^{N}	95.00^{N}	91.98^{N}	91.29^{N}	95.50^{N}
	MEWLSP [58]	99.71^{N}	-	-	-	98.48^{N}
	WLBPC [60]	-	-	-	-	95.01^{N}
	CVLBC [61]	98.86^{N}	-	-	-	91.31^{N}
	CSAP-TOP [38]	**100**	96.67	92.59	90.53	-
	FoSIG [40]	99.14	96.67	92.59	92.42	95.99
	V-BIG [39]	99.43	**100**	95.06	94.32	96.65
	HILOP [66]	99.71	96.67	91.36	92.05	96.21
	$MMDP_{D\_M/C}$ [50]	99.43	98.33	96.91	92.05	95.86
	$MEMDP_{D\_M/C}$ [50]	99.71	96.67	96.91	93.94	96.03
	RUBIG [49]	98.86	**100**	95.68	93.56	97.08
	Our MSIOMF$^{\{1^{st},2^{nd}\},\mathcal{D}^4}_{\{0.5,1.0\}}$	99.14	96.67	95.68	94.70	97.29
	Our MSVOMF$^{\{1^{st},4^{th}\},\mathcal{D}^4}_{\{0.7,1.0\}}$	99.71	96.67	96.30	95.08	97.87
F	DL-PEGASOS [17]	-	-	-	-	63.70
	PCA-cLBP/PI/PD-LBP [48]	-	-	-	-	92.40
	Orthogonal Tensor DL [33]	-	87.80	76.70	74.80	94.70
	Equiangular Kernel DL [34]	-	88.80	77.40	75.60	93.40
	st-TCoF [29]	-	**100***	**100***	98.11*	-
	PCANet-TOP [31]	-	96.67*	90.74*	89.39*	-
	D3 [32]	-	**100***	**100***	98.11*	-
	DT-CNN-AlexNet [30]	-	**100***	99.38*	**99.62***	98.18*
	DT-CNN-GoogleNet [30]	-	**100***	**100***	**99.62***	**98.58***

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

in the nearly same order with MSVOMF on *Beta* but they are not on the others (see Tables 10 and 11). In addition, the highest rates of our proposed descriptors are 98.33% on *Alpha* and 95.83% on *Gamma* (see Tables 6, 8, and 9). Also recommended for mobile applications, MSIOMF$^{\{1^{st},2^{nd}\}}_{\{0.5,1.0\}}$ obtains the promising rates, 99.14% on *DynTex35* and 95.68% on *Beta* just in a small dimension of 2880 bins.

Classification on DynTex++: It can be seen from Table 9 that the multi-analysis of deviations and gradients could significantly improve the performance of our proposed descriptors in DT classification on DynTex++, mostly about 97% compared to the other analyses (see Tables 6, 7, and 8). With respect to settings chosen for comparison, our MSIOMF and MSVOMF descriptors achieve the highest rates in comparison with the shallow methods, except MEWLSP [58] with less about 0.6% (see Table 11). Nevertheless, MEWLSP has not been verified on the challenging subsets of DynTex (i.e., *Alpha*, *Beta*, and *Gamma*), as well as not better than ours on UCLA (see Table 10). For improvement in further context, the challenging categories in red rates in Fig. 17 should be concentrated on.

4.7. Comprehensive comparison to deep-learning methods

Classification on UCLA: It can be seen from Table 10 that our shallow framework has good performances in understanding turbulent motions of DTs in UCLA videos compared to deep-learning methods. For instance, MSVOMF$^{\{1^{st},4^{th}\}}_{\{0.7,1.0\}}$ obtains rates of 100% for *50-4fold*; 99.35% for both *9-class* and *8-class*. These are about 0.5∼1% better than rates of DT-CNN [30] that utilizes GoogleNet learning framework to achieve rates of 99.5%, 98.35%, and 99.02% respectively (see Table 10).

Classification on DynTex: On the challenging subsets of DynTex, the deep-learning techniques [29, 30, 32] have shown their effectiveness in learning features of DTs (see Table 11). However, they take a tremendous number of parameters for complicated learning algorithms. In the meanwhile, just using a shallow analysis, our proposal also has results being close to those of the deep-learning methods. More particularly, our MSVOMF$^{\{1^{st},4^{th}\}}_{\{0.7,1.0\}}$ obtains 96.67% on *Alpha*, 3.3% lower than st-TCoF's [29], D3's [32], and DT-CNN's [30]. This is due to just two confusions between "Grass" and "Tree" (see Fig. 14). In regard to classifying DTs on *Beta* and *Gamma*, our proposed framework also obtains promising rates, 96.30% on *Beta* by MSVOMF$^{\{1^{st},4^{th}\}}_{\{0.7,1.0\}}$, MSVOMF$^{\{4^{th}\}}_{\{1.0,1.3\}}$, and MSVOMF$^{\{2^{nd},4^{th}\}}_{\{1.0\}}$ (see

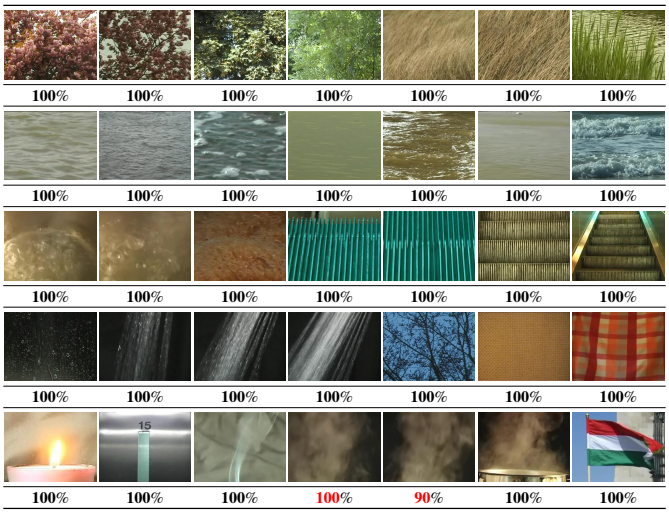

Fig. 13. (Best viewed in color) Classification rates of $\text{MSVOMF}^{\{1^{st},4^{th}\}}_{\{0.7,1.0\}}$ **on specific categories of** *DynTex35*.

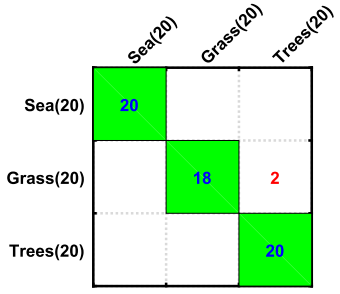

Fig. 14. Confusions of $\text{MSVOMF}^{\{1^{st},4^{th}\}}_{\{0.7,1.0\}}$ **on** *Alpha*.

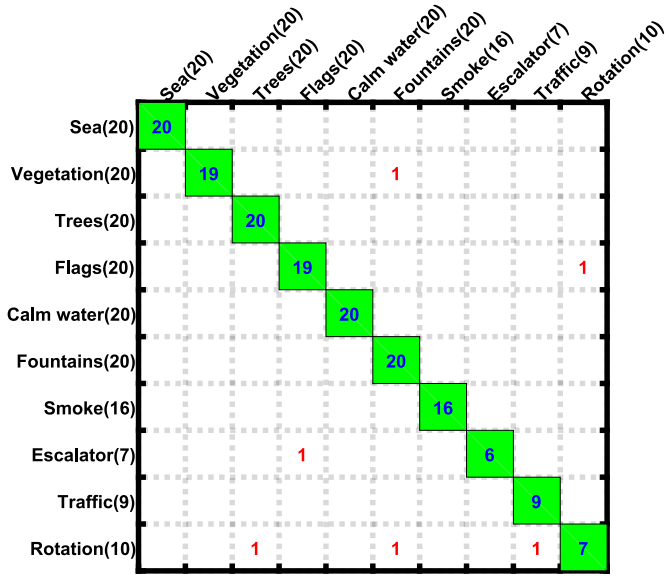

Fig. 15. Confusions of $\text{MSVOMF}^{\{1^{st},4^{th}\}}_{\{0.7,1.0\}}$ **on** *Beta*.

Tables 7, 8, and 9); while 95.83% on *Gamma* by many of the SIOMF and SVOMF descriptors (see Tables 8 and 9). In terms of settings chosen for comparison, the obtained performances are 96.30% on *Beta* and 95.08%, a little lower on *Gamma*.

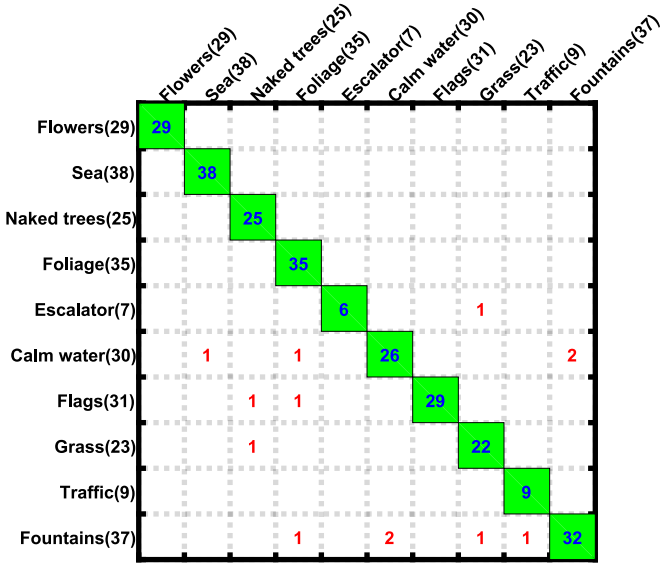

Fig. 16. Confusions of $\text{MSVOMF}^{\{1^{st},4^{th}\}}_{\{0.7,1.0\}}$ **on** *Gamma*.

Those are due to confusions of similar DT motions in categories shown in Fig. 15 for *Beta* and Fig. 16 for *Gamma*.

Classification on DynTex++: Just using a simple framework, performances of our proposed descriptors are nearly the same as those of deep-learning methods. Indeed, it can be verified from Table 11 that results of deep model DT-CNN [30] are 98.18% with AlexNet framework and 98.58% with GoogleNet. These are just 0.3∼0.6% better than our $\text{MSVOMF}^{\{1^{st},4^{th}\}}_{\{0.7,1.0\}}$ with 97.87%. It should be noted that AlexNet and GoogleNet used complicated algorithms and ∼61M and ∼6.8M learned parameters respectively for learning DT features on different datasets. For further improvement, the challenging categories expressed in red rates in Fig. 17 should be addressed in future works.

5. Global discussions

Beside the comprehensive evaluations are thoroughly discussed in Section 4.5, it can be derived further statements and experimental results of $\text{MSIOMF}^{\{k\},\mathcal{D}^n}_{\{\sigma\}}$ and $\text{MSVOMF}^{\{k\},\mathcal{D}^n}_{\{\sigma\}}$ as:

- It should be noted that high-order oriented magnitudes extracted by a direction range $\mathcal{D}^2 = \{[0,\pi),[\pi,2\pi)\}$ could make the corresponding descriptors (i.e., $\text{MSIOMF}^{\{k\},\mathcal{D}^2}_{\{\sigma\}}$ and $\text{MSVOMF}^{\{k\},\mathcal{D}^2}_{\{\sigma\}}$) be in inferior performances (see Table 12) due to lack of complements of micro-oriented information. In spite of that, $\text{MSIOMF}^{\{1^{st},2^{nd}\},\mathcal{D}^2}_{\{0.5,1.0\}}$, just 1400 bins, obtains noticeable rates on *Beta* (95.68%) and *Gamma* (95.98%). It may be a potential solution for mobile applications on edge devices having limited resources.
- It can be verified from Table 12 that addressing the decomposing models in smaller angles (e.g., $\mathcal{D}^8$) makes a sharp increase of dimensions while the performance seems not to be improved, except a little of $\text{MSVOMF}^{\{1^{st},4^{th}\},\mathcal{D}^8}_{\{0.7,1.0\}}$ on *Gamma*. This can be due to the weakness of appearance features caused by the smallness of direction ranges.

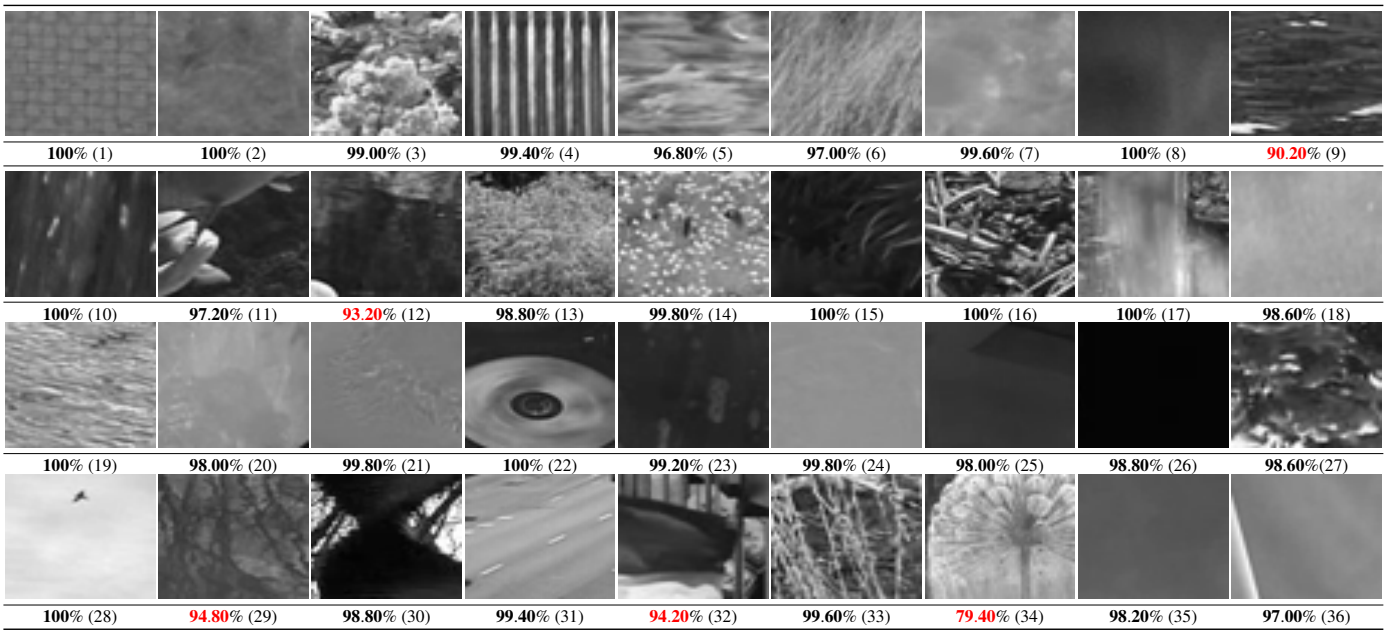

Fig. 17. (Best viewed in color) Specific rates of $\text{MSVOMF}^{\{1^{st},4^{th}\}}_{\{0.7,1.0\}}$ **on each category of DynTex++. The challenging categories are in red rates. Therein, the numbers in the parentheses denote the numbered class labels correspondingly**

Table 12. Rates (%) of MSIOMF and MSVOMF in different ranges $\mathcal{D}^n$ on challenging datasets using the settings chosen for comparison.

Range	Descriptor	#bins	Dyn35	Alpha	Beta	Gamma	Dyn++
$\mathcal{D}^2$	$\text{MSIOMF}^{\{1^{st},2^{nd}\}}_{\{0.5,1.0\}}$	1400	97.43	**96.67**	95.68	**95.83**	96.96
	$\text{MSVOMF}^{\{1^{st},4^{th}\}}_{\{0.7,1.0\}}$	4320	98.29	**96.67**	93.83	94.70	97.86
$\mathcal{D}^4$	$\text{MSIOMF}^{\{1^{st},2^{nd}\}}_{\{0.5,1.0\}}$	2880	99.14	**96.67**	95.68	94.70	97.29
	$\text{MSVOMF}^{\{1^{st},4^{th}\}}_{\{0.7,1.0\}}$	8640	**99.71**	**96.67**	**96.30**	95.08	**97.87**
$\mathcal{D}^8$	$\text{MSIOMF}^{\{1^{st},2^{nd}\}}_{\{0.5,1.0\}}$	5760	97.71	**96.67**	95.68	93.18	97.32
	$\text{MSVOMF}^{\{1^{st},4^{th}\}}_{\{0.7,1.0\}}$	17280	98.57	**96.67**	94.44	95.45	97.43

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

Recently, the deep-learning trend has become one of the main streams of computer vision community. Deep-learning methods have often achieved good results in recognizing DTs on challenging schemes (see Table 11). Nevertheless, they have spent much computational cost in learning millions of parameters through deep neural networks (DNN). In addition, because the inference phase is based on a huge volume of learned parameters, a deployment of DNN models on edge devices is challenging. For instance, it takes ~61M parameters for AlexNet [67] and ~6.8M for GoogleNet [68] implemented in DT-CNN [30] for DT representation. As a result, it is restricted to deploy the deep-learning for real applications in mobile devices as well as embedded sensor systems due to a strict requirement of tiny resources for their executions. Contrary to the complicated models of the deep-learning-based methods, our proposed framework have obtained the competitive performances but just using shallow analysis, expected to be one of potential solutions for mobile implementations. Indeed, just utilizing a simple operator to capture the IOM/VOM-based features from the Gaussian-gradient magnitudes, our MSIOMF and MSVOMF descriptors have the significant performance compared to that of all non-deep-learning methods, while being close to that of the deep-learning ones (see Tables 10 and 11). Furthermore, CLBP [37] at the period of local encoding could be replaced by other robust operators (e.g., CLBC [52], MRELBP [69], LRP [49], LDP-based [70, 50], LVP-based [71, 64], etc.) in order to investigate the IOM/VOM-based features in different circumstances for potential enhancements.

In respect of real-world applications of our proposed framework, it can be used for early-warning fire monitoring systems or for computing devices of ubiquitous smart home [72] in order to detect fire-flame as investigated in former work [73]. It is thanks to the effectiveness of the proposed IOM/VOM-based features in the shallow analysis. In addition, ours may be considered in other applications based on DT analysis as done in several former works: facial expression [42, 74, 75], segmentation [76, 77, 78, 79], lipreading [80], iris recognition [81], etc.

6. Conclusions

In this paper, we have proposed a simple and efficient framework in which the high-order oriented magnitudes of Gaussian gradients are exploited for DT representation. Accordingly, the decomposing models of hard and soft-based assignments have been investigated in different direction ranges (i.e., $\mathcal{D}^2$, $\mathcal{D}^4$, $\mathcal{D}^8$) for extracting IOM/VOM-based features from the Gaussian-gradient magnitudes. Therein, the modified soft-assignment model of $\mathcal{D}^4$ has pointed out the best performances. The experimental results for DT classification issue have validated that local descriptors $\text{MSIOMF}^{\{k\},\mathcal{D}^4}_{\{\sigma\}}$ and $\text{MSVOMF}^{\{k\},\mathcal{D}^4}_{\{\sigma\}}$ based on these extracted features have significant enhancement of discrimination power in comparison with the Gaussian-based descriptors (i.e., FoSIG [40] and V-BIG [39]) as well as the others in state-of-the-art methods. Also, those have confirmed the

interest of our approach based on the oriented magnitudes of Gaussian gradients rather than based on the non-oriented ones.

For perspectives, the problems of zero-pixels/voxels in the IOM/VOM-based outcomes, which are caused by the decomposing models, can negatively affect the discriminative power when using CLBP [37] for the local encoding stage. To overcome those, CLBP can be modified in the future work to adapt this encoding context. Moreover, in case of treating the curse of expansive dimension, it is able to address a multi-scale analysis of supporting regions (e.g., $\{(P,R)\} = \{(8,1),(8,2)\}$) to explore more local relationships of IOM/VOM-based features in larger neighborhoods for further improvements.

Acknowledgment

We would like to express our deep gratitude to the reviewers and editors, 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, Ho Chi Minh City, Vietnam, who gave us several crucial supports.

References

[1] S. Ghodsi, H. Mohammadzade, E. Korki, Simultaneous joint and object trajectory templates for human activity recognition from 3-d data, J. Visual Communication and Image Representation 55 (2018) 729–741.

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

[3] N. Ç. Kiliboz, U. Güdükbay, A hand gesture recognition technique for human-computer interaction, J. Visual Communication and Image Representation 28 (2015) 97–104.

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

[5] S. Tian, L. Zou, C. Fan, L. Chen, Weighted correlation filters guidance with spatial-temporal attention for online multi-object tracking, J. Visual Communication and Image Representation 63 (2019).

[6] D. Jeyabharathi, Dejey, Cut set-based dynamic key frame selection and adaptive layer-based background modeling for background subtraction, J. Visual Communication and Image Representation 55 (2018) 434–446.

[7] G. Srivastava, R. Srivastava, Salient object detection using background subtraction, gabor filters, objectness and minimum directional backgroundness, J. Visual Communication and Image Representation 62 (2019) 330–339.

[8] A. Dehghan, M. Shah, Binary quadratic programing for online tracking of hundreds of people in extremely crowded scenes, IEEE Trans. Pattern Anal. Mach. Intell. 40 (2018) 568–581.

[9] Z. Q. H. Al-Zaydi, D. L. Ndzi, Y. Yang, L. Kamarudin, An adaptive people counting system with dynamic features selection and occlusion handling, J. Visual Communication and Image Representation 39 (2016) 218–225.

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

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

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

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

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

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

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

[17] B. Ghanem, N. Ahuja, Maximum margin distance learning for dynamic texture recognition, in: K. Daniilidis, P. Maragos, N. Paragios (Eds.), ECCV, volume 6312 of *LNCS*, 2010, pp. 223–236.

[18] R. Péteri, D. Chetverikov, Dynamic texture recognition using normal flow and texture regularity, in: J. S. Marques, N. P. de la Blanca, P. Pina (Eds.), IbPRIA, volume 3523 of *LNCS*, 2005, pp. 223–230.

[19] R. Péteri, D. Chetverikov, Qualitative characterization of dynamic textures for video retrieval, in: K. W. Wojciechowski, B. Smolka, H. Palus, R. Kozera, W. Skarbek, L. Noakes (Eds.), ICCVG, volume 32 of *Computational Imaging and Vision*, 2004, pp. 33–38.

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

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

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

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

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

[25] 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 (2013) 1606–1621.

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

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

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

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

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

[31] Dynamic texture representation using a deep multi-scale convolutional network, J. Visual Communication and Image Representation 43 (2017) 89 – 97.

[32] 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.

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

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

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

[36] 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 (2019) 1694–1708.

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

[38] 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 (2018) 053044.

[39] 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.

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

[41] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE

Trans. PAMI 24 (2002) 971–987.
[42] G. Zhao, M. Pietikäinen, Dynamic texture recognition using local binary patterns with an application to facial expressions, IEEE Trans. PAMI 29 (2007) 915–928.
[43] G. Zhao, T. Ahonen, J. Matas, M. Pietikäinen, Rotation-invariant image and video description with local binary pattern features, IEEE Trans. IP 21 (2012) 1465–1477.
[44] D. Tiwari, V. Tyagi, Dynamic texture recognition based on completed volume local binary pattern, MSSP 27 (2016) 563–575.
[45] D. Tiwari, V. Tyagi, A novel scheme based on local binary pattern for dynamic texture recognition, CVIU 150 (2016) 58–65.
[46] 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.
[47] J. Ren, X. Jiang, J. Yuan, G. Wang, Optimizing LBP structure for visual recognition using binary quadratic programming, IEEE Signal Process. Lett. 21 (2014) 1346–1350.
[48] J. Ren, X. Jiang, J. Yuan, Dynamic texture recognition using enhanced LBP features, in: ICASSP, 2013, pp. 2400–2404.
[49] T. T. Nguyen, T. P. Nguyen, F. Bouchara, Rubik gaussian-based patterns for dynamic texture classification, Pattern Recognition Letters 135 (2020) 180–187.
[50] T. T. Nguyen, T. P. Nguyen, F. Bouchara, X. S. Nguyen, Momental directional patterns for dynamic texture recognition, CVIU 194 (2020) 102882.
[51] T. P. Nguyen, A. Manzanera, W. G. Kropatsch, X. S. N'Guyen, Topological attribute patterns for texture recognition, Pattern Recognition Letters 80 (2016) 91–97.
[52] Y. Zhao, D.-S. Huang, W. Jia, Completed Local Binary Count for Rotation Invariant Texture Classification, IEEE Trans. IP 21 (2012) 4492–4497.
[53] A. K. Jain, F. Farrokhnia, Unsupervised texture segmentation using gabor filters, Pattern Recognition 24 (1991) 1167–1186.
[54] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: CVPR, 2005, pp. 886–893.
[55] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2004) 91–110.
[56] T. Song, L. Luo, C. Gao, G. Zhang, Texture representation using local binary encoding across scales, frequency bands and image domains, in: ICIP, 2019, pp. 4405–4409.
[57] 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 (2015) 819–830.
[58] D. Tiwari, V. Tyagi, Dynamic texture recognition using multiresolution edge-weighted local structure pattern, Computers & Electrical Engineering 62 (2017) 485–498.
[59] R. Fan, K. Chang, C. Hsieh, X. Wang, C. Lin, LIBLINEAR: A library for large linear classification, JMLR 9 (2008) 1871–1874.
[60] D. Tiwari, V. Tyagi, Improved weber's law based local binary pattern for dynamic texture recognition, Multimedia Tools Appl. 76 (2017) 6623–6640.
[61] 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 (2018) 552–566.
[62] G. M. Amdahl, Validity of the single processor approach to achieving large-scale computing capabilities, in: AFIPS, volume 30, 1967, pp. 483–485.
[63] N. J. Gunther, Scalability—A Quantitative Approach, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 41–69.
[64] T. T. Nguyen, T. P. Nguyen, F. Bouchara, Directional dense-trajectory-based patterns for dynamic texture recognition, IET Computer Vision 14 (2020) 162–176.
[65] Y. Xu, Y. Quan, Z. Zhang, H. Ling, H. Ji, Classifying dynamic textures via spatiotemporal fractal analysis, Pattern Recognition 48 (2015) 3239–3248.
[66] T. T. Nguyen, T. P. Nguyen, F. Bouchara, Dynamic texture representation based on hierarchical local patterns, in: ACIVS, 2020, pp. 277–289.
[67] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Conference on Neural Information Processing Systems (NIPS), 2012, pp. 1106–1114.
[68] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9.
[69] 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 (2016) 1368–1381.
[70] 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 (2010) 533–544.
[71] K. Fan, T. Hung, A novel local pattern descriptor - local vector pattern in high-order derivative space for face recognition, IEEE Trans. IP 23 (2014) 2877–2891.
[72] Y. Lai, C. Lai, Y. Huang, H. Chao, Multi-appliance recognition system with hybrid SVM/GMM classifier in ubiquitous smart home, Inf. Sci. 230 (2013) 39–55.
[73] K. Dimitropoulos, P. Barmpoutis, N. Grammalidis, Spatio-temporal flame modeling and dynamic texture analysis for automatic video-based fire detection, IEEE Trans. Circuits Syst. Video Technol. 25 (2015) 339–351.
[74] X. Huang, G. Zhao, W. Zheng, M. Pietikäinen, Spatiotemporal local monogenic binary patterns for facial expression recognition, IEEE Signal Process. Lett. 19 (2012) 243–246.
[75] R. Shao, X. Lan, P. C. Yuen, Joint discriminative learning of deep dynamic textures for 3d mask face anti-spoofing, IEEE Transactions on Information Forensics and Security 14 (2019) 923–938.
[76] W. N. Gonçalves, O. M. Bruno, Dynamic texture segmentation based on deterministic partially self-avoiding walks, Comput. Vis. Image Underst. 117 (2013) 1163–1174.
[77] W. N. Gonçalves, O. M. Bruno, Dynamic texture analysis and segmentation using deterministic partially self-avoiding walks, Expert Syst. Appl. 40 (2013) 4283–4300.
[78] J. Chen, G. Zhao, M. Salo, E. Rahtu, M. Pietikäinen, Automatic dynamic texture segmentation using local descriptors and optical flow, IEEE Transactions on Image Processing 22 (2013) 326–339.
[79] J. Chen, G. Zhao, M. Pietikäinen, Unsupervised dynamic texture segmentation using local descriptors in volumes, in: ICPR, IEEE Computer Society, 2012, pp. 3622–3625.
[80] G. Zhao, M. Barnard, M. Pietikäinen, Lipreading with local spatiotemporal descriptors, IEEE Transactions on Multimedia 11 (2009) 1254–1265.
[81] V. de Melo Langoni, A. Gonzaga, Evaluating dynamic texture descriptors to recognize human iris in video image sequence, Pattern Anal. Appl. 23 (2020) 771–784.