



HAL
open science

Directional Dense-Trajectory-based Patterns for Dynamic Texture Recognition

Thanh Tuan Nguyen, Thanh Phuong Nguyen, Frédéric Bouchara

► **To cite this version:**

Thanh Tuan Nguyen, Thanh Phuong Nguyen, Frédéric Bouchara. Directional Dense-Trajectory-based Patterns for Dynamic Texture Recognition. IET Computer Vision, In press. hal-02470418

HAL Id: hal-02470418

<https://hal.science/hal-02470418>

Submitted on 7 Feb 2020

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

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

Directional Dense-Trajectory-based Patterns for Dynamic Texture Recognition

THANH TUAN NGUYEN

HCMC University of Technology and Education, Faculty of IT, Ho Chi Minh, Vietnam
Université de Toulon, Aix Marseille Université, CNRS, LIS, Marseille, France

THANH PHUONG NGUYEN *

Université de Toulon, Aix Marseille Université, CNRS, LIS, Marseille, France

FRÉDÉRIC BOUCHARA

Université de Toulon, Aix Marseille Université, CNRS, LIS, Marseille, France

February 7, 2020

Abstract

Representation of dynamic textures (DTs), well-known as a sequence of moving textures, is a challenging problem in video analysis due to disorientation of motion features. Analyzing DTs to make them “understandable” plays an important role in different applications of computer vision. In this paper, an efficient approach for DT description is proposed by addressing the following novel concepts. First, beneficial properties of dense trajectories are exploited for the first time to efficiently describe DTs instead of the whole video. Second, two substantial extensions of Local Vector Pattern operator are introduced to form a completed model which is based on complemented components to enhance its performance in encoding directional features of motion points in a trajectory. Finally, we present a new framework, called Directional Dense Trajectory Patterns, which takes advantage of directional beams of dense trajectories along with spatio-temporal features of their motion points in order to construct dense-trajectory-based descriptors with more robustness. Evaluations of DT recognition on different benchmark datasets (i.e., UCLA, DynTex, and DynTex++) have verified the interest of our proposal.

1 Introduction

Dynamic textures (DTs) are repetition of image textures along a temporal domain [1], such as blowing flag, trees, fire, clouds, waves, foliage, fountain, etc.

*AI Lab, Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam

Analyzing DTs to make them “understandable” is one of crucial principles for computer vision tasks (e.g., segmentation, classification, synthesis) in order to implement applications in real life: fire and smoke detection [2], tracking objects [3, 4, 5, 6, 7], facial expressions [8, 9], action recognition [10, 11]. Different methods have been introduced for video representation to address the major challenges in processing DTs, such as the large region of appearances, illumination, noise, and chaotic motions. In general, it is possible to provisionally group them into six categories as follows.

Optical-flow-based methods: Due to efficiently computing and robustly describing videos in natural way, optical-flow-based approaches are widely utilized not only for DT recognition but also for other problems of computer vision. Peh *et al.* [12] took advantage of magnitude and direction information of the Normal Flow (NF) to deal with spatio-temporal texture depiction while NF and filtering regularity are combined together in [13] to extract six features from a sequence for DT representation. Fazekas *et al.* [14] concentrated on rotation-invariant distortions of local texture images to capture temporal periodicity of moving features. Their experiments verify that complete flow for these distortions outperforms compared to NF in recognition on a large number of classes of datasets with high resolution [15]. According to [8], optical-flow-based techniques in consideration of brightness constancy and local smoothness are not in accordance with capturing chaotic motions in practice. Furthermore, only motion textures are encoded while their apparent characteristics have not been addressed. To mitigate those drawbacks, our previous work utilized dense trajectories extracted from a video to structure directional patterns of motion points using a local-feature-based operator LVP_{full} -TOP in full directions [16].

Model-based methods: Saisan *et al.* [1] proposed Linear Dynamical System (LDS), a foundation of the model-based approaches. Utilizing Gaussian noise conception based on the Hidden Markov Model (HMM), LDS is widely exploited to describe videos for evaluating recognition tasks in computer vision. Chan *et al.* [17] stated a mixture of DT models as an extension of LDS in order to seize motions of objects in sequences. Shortcomings of Principal-Component-Analysis-based (PCA-based) methods, which apply a linear observation function to encode dynamic features of complex motions, are fixed in [18] by exploiting a non-linear observation function with *kernel*-PCA to structure complication of chaotic motions (e.g., turbulent water) and those captured by camera moving (e.g., panning, zooming, and rotations). Based on the similar characteristics of DT mixtures (DTMs) extracted in [17], Mumtaz *et al.* [19] used the hierarchical expectation-maximization algorithm to categorize DTMs into k clusters. The advantages of LDS and a bag of words are taken into account in [20, 21] to depict DTs for classification. Recently, Qiao *et al.* [22] have encoded adjacent spatial voxels using Multivariate Hidden Markov Model and revealed that this model outperforms in comparison with high-order HMM for classifying DTs. Although various efforts have been made for DT recognition, performances of model-based methods are in general less efficient than others’. One of the main reasons is their concentration on spatial-appearance-based features rather than dynamic clues [1] Moreover, the models become more complicated in case of

taking them into account for the dynamical properties [20].

Filter-based methods: In these approaches, filter bank operations are utilized to diminish noise in DT sequences. Arashloo *et al.* [23] addressed Binarized Statistical Image Features on Three Orthogonal Planes (BSIF-TOP) and its multi-scale scheme to encode dynamic patterns in which binary codes are obtained by filtering performances on varied spatio-temporal regions in videos and by binarizing reactions of the filters. In [8], the authors introduced a robust technique, named Directional Number Transitional Graph (DNG), to figure out spatial and temporal directional numbers on the frames of a sequence for the purpose of grouping the video into a 3D grid. Experiments on DT recognition verify that filter-based methods have outperformed on simple motion features (e.g., DTs in UCLA dataset [1]). For more complex DTs, as in DynTex [24] and DynTex++ [25], they are less efficient while learning filters in BSIF-TOP or considering DTs of 3D sub-sequences in DNG takes a long time to process with high computational complexity.

Geometry-based methods: DT features of videos are estimated by fractal analysis techniques in which the information of self-similarities in geometry theory is taken into account video representation to be able to tolerate the environmental changes of sequences. Xu *et al.* [26] proposed a technique of Dynamic Fractal Spectrum (DFS) with two parts integrated into as follows: volumetric DFS considers DT sequences as 3D volumes to seize their statistical self-similarities. The other, called multi-slice DFS, captures fractal patterns repeated on the frames of volumes. Multi-Fractal Spectrum [27] is also introduced in order that SIFT-like features are employed for the fractal processes of DT representation. Then Ji *et al.* [28] used low-pass and high-pass wavelet coefficients along with wavelet leaders to form a wavelet-based MFS descriptor with robust power of discrimination while strongly suffering from the changes of environment. Recently, Spatio-Temporal Lacunarity Spectrum [29] depicts a video based on lacunarity analysis on its DT slices to structure lacunarity-based features. Another work [30] addressed Non-Linear Stationary Subspace Analysis to encode the stationary parts of DT sequences for decreasing the dimension of description. Regarding efficiency on DT classification, geometry-based methods outperform on simple DT datasets (e.g., UCLA) rather than on complex dynamic features in DynTex and DynTex++. Furthermore, some of them are lack of temporal information in the video analysis.

Learning-based methods: Owing to outperforming results of DT recognition, learning-based approaches have recently attracted researchers with promising techniques. Qi *et al.* [31] formed TCoF patterns based on a Convolutional Neural Network (CNN) transfer learning from deep structures in still images for characterizing features in DT recognition. Andrearczyk *et al.* [32] also addressed using two popular CNN architectures (i.e., AlexNet and GoogleNet) to train DT-CNN features based on spatial-temporal frames of three orthogonal planes while Arashloo *et al.* [33] utilized a PCA convolutional network (PCANet) to learn filters on these planes in order to establish a multi-layer convolutional structure, named PCANet-TOP, for DT representation and classification. In the meanwhile, methods of Dictionary Learning have also become

more attractive in which local DT features are figured out by kernel sparse coding to enhance the discriminative power of descriptors. Quan *et al.* [34] introduced a method of sparse coding to learn a dictionary from atoms, known as patches taken from DT sequences, for capturing local DT characteristics. However, because a compulsory requirement is that atoms are in the identical dimension, it is inconvenient to implement in multi-scale resolutions for improvement of the performance. Another effort has been addressed equiangular kernel to advance the effect of learning dictionary process as well as to remedy its high-dimensional problem [35]. Despite achieving significant results in DT recognition, the learning-based methods generally take a long time to capture DT features because of learning algorithms with high computational complexity. Our proposal in this work can obtain competitive classification rates by exploiting dense trajectories of videos along with an efficient operator for encoding local directional features in simple computation.

Local-feature-based methods: In this perception, DT features of videos are mostly captured by Local Binary Pattern (LBP) operator [36] and its LBP-based variants due to their simple and efficient computation. For DT representation, Zhao *et al.* [9] presented two LBP-based operators: Volume LBP (VLBP) for encoding dynamic patterns based on spatio-temporal relations of features on three consecutive frames; LBP on three orthogonal planes (LBP-TOP) for capturing motion and shape cues from these planes. Then, many efforts have been made to advance the discrimination of DT descriptors based on diverse extensions of two above typical operators. Ren *et al.* [37] tried to reduce feature vectors in a reasonable dimension using a technique of learning data-driven LBP structures optimized by a scheme of maximal joint mutual information. The information of local structures and image moments is addressed for the completed scheme [38] on three orthogonal planes to form respectively CLSP-TOP [39] and CSAP-TOP [40] patterns. In the meanwhile, an other combination of Completed Local Binary Count [41] (CLBC) and the concept of VLBP is also exploited to form CVLBC descriptor [42] with more robustness for DT recognition. Tiwari *et al.* [43] introduced Helix Local Binary Patterns (HLBPs) to take the advantages of characteristics in both LBP-TOP and VLBP patterns. Other LBP-based variants for DT representation have been also proposed in recent works, such as WLBPC [44] using Weber’s law to enhance the role of center pixel, EWLSP [45] encoding the information of edge-weighted local structure patterns.

Even though the local-feature-based methods achieve promising DT recognition results, they survive several inherent problems, such as sensitivity to noise, near uniform regions [43, 39], and large dimension [9, 37, 46]. In the meanwhile, our prior effort [16] has attempted to partly deal with restrictions of optical-flow-based methods by taking into account a directional LBP-based operator LVP_{full} -TOP to structure characteristics of dense trajectories in consideration of full directions. However, the important temporal information of motion points in these trajectories has not been exploited as well as directional relationships have not been considered in a completed context of larger local regions. Addressing those obstacles, we indicate the following crucial improvements to enhance the performance compared to our previous work [16]:

- A completed model of Local Vector Pattern (LVP) is introduced to efficiently encode trajectories in a slighter dimension compared to using LVP_{full} -TOP in [16]. In addition, adaptive directional vector thresholds ($DVM_{\alpha,d}(\mathcal{I})$ and $DVC_{\alpha,d}(\mathcal{I})$) have been introduced to address two other components of the completed model (see Fig. 1).
- Exploiting temporal information of motion points in trajectories.
- Directional local relationships are conducted in larger supporting regions. This allows to capture more spatio-temporal information in order to boost the discrimination power.
- A thorough framework for taking beneficial properties of dense trajectories into account DT representation is presented.

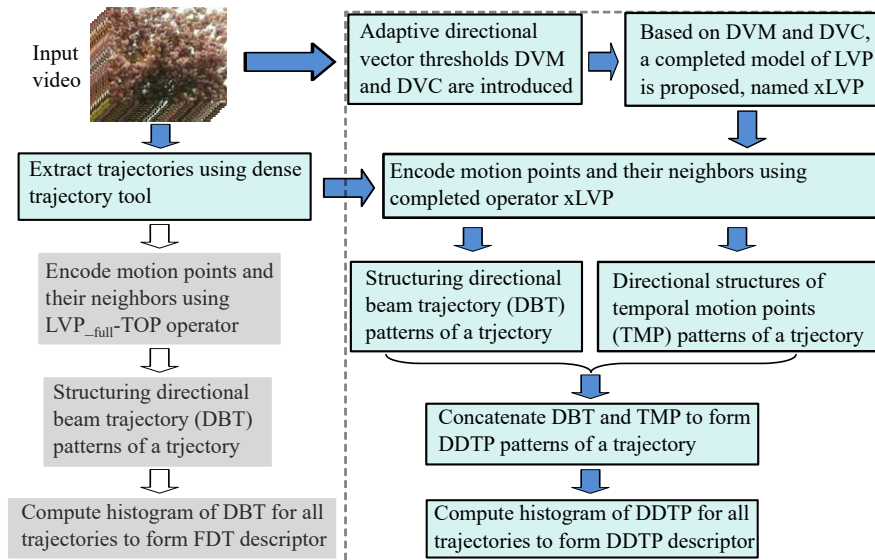


Figure 1: (Best viewed in color) Highlight of this proposal that are presented in blue background (the extension is in the dashed box) compared to our previous work [16], presented in dark background, for encoding a video based on its dense trajectories.

In general, the proposed framework in this work consists of three stages as follows. First, motion points and their paths in a video extracted by using an extracting tool [4]. Second, crucial extensions of LVP [47] operator are proposed by taking advantage of the information of magnitudes and center contrast levels in order to form a completed operator, named xLVP, with outperformance compared to the basic LVP [47]. Third, two important beneficial properties of dense trajectories are exploited: Directional features of beam trajectories, and spatio-temporal features of motion points along with their paths in which their directional relationships are captured by using the robust operator xLVP. Finally,

the obtained histograms are concatenated and normalized to effectively construct DT descriptors, named Directional Dense Trajectory Patterns (DDTP), with more robustness. Consequently, it could be realized that the advantages of both optical-flow-based and local-feature-based methods are consolidated into our approach to improve DT representation. In short, the major contributions of this work can be listed as follows.

- Dense trajectories, extracted from a video, are involved with DT representation for the first time instead of the whole video.
- Profitable characteristics of optical-flow-based and local-feature-based methods are exploited thanks to using a discriminative operator proposed for encoding these dense trajectories.
- A novel operator xLVP is presented to efficiently capture directional information in consideration of an incorporation between beams of dense trajectories and their motion points.
- Two adaptive directional vector thresholds, introduced to make the completed model xLVP, agree with complemented components of magnitudes and center contrast levels.
- An effective framework for DT description has been proposed to form robust DDTP descriptors by taking advantage of properties of dense trajectories.

2 Related work

Taking LBP into account encoding local relationships is one of the most interested approaches in image representation due to its outperformance with simple computation. In this portion, we take a brief review of LBP and its variants for structuring DTs in recognition task. Furthermore, a model of directional LBP-based patterns is recalled in short as well as a technique of extracting dense trajectories from a DT video is also involved with in a summary.

2.1 A brief review of LBP

Ojala *et al.* [36] introduced LBP, a well-known operator of effective computation in still images, to encode a local textural feature as a binary chain code. Specifically, given a center pixel \mathbf{q}_c of a 2D texture image \mathcal{I} , binary codes of LBP for \mathbf{q}_c are defined as follows.

$$\text{LBP}_{P,R}(\mathbf{q}_c) = \sum_{i=0}^{P-1} s(\mathcal{I}(\mathbf{p}_i) - \mathcal{I}(\mathbf{q}_c))2^i \quad (1)$$

where \mathbf{p}_i is the i^{th} surrounding neighbor of \mathbf{q}_c , P is a number of neighbors interpolated on a circle of radius R centered at \mathbf{q}_c , and function $s(\cdot)$ is defined

as

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

As the result of that, a texture image is formed by a histogram of 2^P distinct values. Because of the large dimension of the basic LBP, in practice, two most popular mappings are usually taken into account to turn it down into a reasonable size as follows. Uniform patterns (LBP^{u2}) [36] with $P(P-1)+3$ bins are derived from the typical LBP codes conditioned by number of bit-transitions (1-0 or 0-1) of their binary chains at most 2. The other important mapping to deal with rotation invariant (LBP^{ri}) [36] is stated as

$$\text{LBP}_{P,R}^{ri}(\mathbf{q}_c) = \min_{0 \leq i < P} \{ROR(\text{LBP}_{P,R}(\mathbf{q}_c), i)\} \quad (3)$$

where $ROR(\text{LBP}, i)$ calculates the distribution of LBP^{ri} by shifting i times of the P -bit basic LBP. In real applications, ri and $u2$ mappings are often combined to form patterns of $riu2$ mapping (LBP^{riu2}). This leads to reduction of dimensional representation from 2^P of the basic LBP to $P+2$ distinct values, in which P is the number of considered neighbors. Inspired by the effectiveness of above mappings, other crucial mappings are suggested to refine these mappings for encoding more textural information. Zhao *et al.* [41] advanced Local Binary Count (LBC), an alternative of uniform patterns, by considering differences of the higher gray levels between P neighbors and center pixel. On the other hand, Fathi *et al.* [48] extended the basic uniform mapping based on advantages of some non-uniform patterns. Nguyen *et al.* [49] then embedded the underlying mappings and LBC into a general mapping, named TAP^A , to capture topological attribute patterns.

2.2 LBP-based variants for dynamic textures

Taking advantage of LBP operator in still image processing, various LBP-based variants have been proposed for video representation. At first, an extension of the typical LBP to DT description, VLBP [9] encodes a voxel based on three center points corresponding to P neighbors on three consecutive frames which are located at the same spatial coordinate of the centers. Accordingly, these $3P+2$ neighbors are thresholded by the second center pixel to form a $(3P+2)$ -bit binary code which figures out local features and motion cues surrounding this voxel. Because this encoding shapes a descriptor with a very large dimension of 2^{3P+2} bins, it is restricted for implementation in reality. To overcome this problem, Zhao *et al.* [9] introduced LBP-TOP in which the basic LBP is considered on three orthogonal planes of a video. The final DT representation is constructed by concatenating the sub-descriptors computed on these planes. Thereafter, other approaches based on the perceptions of two above encoding models to enhance the discrimination power of descriptors. CVLBC [42] is combined by CLBC [41] and VLBP while Tiwari *et al.* [46] proposed CVLBP operator by taking advantage of the ideas of CLBP [38] and VLBP. CLSP-TOP

[39] addresses local/global information, while CSAP-TOP [40] captures DT features on moment images. Both advanced features of LBP-TOP and VLBP are utilized in [43] to structured HLBP patterns.

2.3 Local Vector Patterns

Fan *et al.* [47] proposed Local Vector Pattern (LVP) operator for image description by regarding a pairwise of directional vectors in order to remedy the remaining shortcomings of local pattern representation. Let \mathcal{I} denote a 2D image. The first-order derivative of a center pixel \mathbf{q}_c conducted by a direction α is computed as

$$\mathcal{I}'_{\alpha,d}(\mathbf{q}_c) = \mathcal{I}(\mathbf{q}_{\alpha,d}) - \mathcal{I}(\mathbf{q}_c) \quad (4)$$

in which $\mathbf{q}_{\alpha,d}$ is an adjacent neighbor sampled by direction α and a distance d from the considered pixel \mathbf{q}_c , $\mathcal{I}(\cdot)$ returns the gray-scale image value of a pixel. The first-order LVP of \mathbf{q}_c is defined as a P -bit binary chain by concerning it with P local directional relations in a couple of directions ($\alpha, \alpha + 45^\circ$) and formed as follows.

$$\text{LVP}_{P,R,\alpha,d}(\mathbf{q}_c) = \left\{ f(\mathcal{I}'_{\alpha,d}(\mathbf{q}_c), \mathcal{I}'_{\alpha+45^\circ,d}(\mathbf{q}_c), \mathcal{I}'_{\alpha,d}(\mathbf{p}_i), \mathcal{I}'_{\alpha+45^\circ,d}(\mathbf{p}_i)) \right\}_{i=0}^{P-1} \quad (5)$$

where $\{\mathbf{p}_i\}$ denotes P neighbors of \mathbf{q}_c , $d \in \{1, 2, 3\}$ presents the distance of the considered pixel with its contiguous points, and $f(\cdot)$, a function of Comparative Space Transform (CST), is defined as

$$f(x, y, z, t) = \begin{cases} 1, & \text{if } t - \frac{y * z}{x} \geq 0 \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

Other formulations of LVP along with samples of encoding LVP-based patterns for texture images are clearly discussed in [47]. In practice, four possible directions are often employed in real applications, i.e., $\alpha = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$, to enrich discriminative information of descriptors [47, 50, 51].

2.4 Dense trajectories

Wang *et al.* [4] introduced an efficient technique for extracting dense trajectories in videos based on a dense optical flow field to locate and track the paths of motion points. In particular, let $\mathbf{q}_f = (x_f, y_f)$ denote a motion point at the f^{th} frame with corresponding coordinates of x_f and y_f . Its displacement at the $(f+1)^{th}$ frame is interpolated by addressing the polynomial expansion algorithm for two-frame motion estimation [52] along with an optical flow $\omega_f = (u_f, v_f)$, which is known as a median filter. Therein, u_f and v_f mean the horizontal and vertical optical flow components. The inferred position of \mathbf{q}_f in the posterior frame, i.e., $\mathbf{q}_{f+1} = (x_{f+1}, y_{f+1})$, is tracked as

$$\mathbf{q}_{f+1} = \mathbf{q}_f + (M * \omega_f)|_{(\bar{x}_f, \bar{y}_f)} \quad (7)$$

in which (\bar{x}_f, \bar{y}_f) refers to the rounded position value of \mathbf{q}_f , M is a median filter kernel of 3×3 pixels. According to that, a dense trajectory with length of L can be structured by a concatenation of the motion point \mathbf{q}_f and its displacements inferred through L consecutive frames, i.e., $\{\mathbf{q}_f, \mathbf{q}_{f+1}, \dots, \mathbf{q}_{f+L-1}\}$. In our framework, we use the version 1.2 of dense trajectories as a tool to extract motion paths of dynamic features for DT description.

3 Important extensions of local vector patterns

The basic LVP operator [47] has been originally introduced to exploit the directional information of texture image patterns in high-order derivative spaces for face recognition. It is then interested in utilizing for other applications in computer vision, such as action recognition [51], image retrieval [50]. For DT description, we get involved with this operator for the first time in order to encode directional vector structures of motion points along their dense trajectories which are extracted from a DT sequence. Due to being a derivation of the LBP concept in textural image representation, the basic LVP operator has existed the internal limitations of LBP, such as sensitivity to noise, illumination, and near uniform images. To mitigate those problems, we hereafter propose two following important extensions of LVP in order to enhance its discrimination for DT recognition task: adaptive directional vector thresholds and a completed model of LVP.

3.1 Adaptive directional vector thresholds

Motivated by the first-order concept of LVP, we define hereunder two adaptive vector thresholds to apply for two corresponding components that are defined in below section to capture magnitude information and directional centered contrast patterns. First, to exploit the information of Directional Vector Magnitudes (DVM) for each direction α , we calculate the mean of absolute CST on the whole image as follows.

$$\text{DVM}_{\alpha,d}(\mathcal{I}) = \frac{\sum_{\mathbf{q} \in \mathcal{I}} \sum_{i=0}^{P-1} \left| \mathcal{I}'_{\beta,d}(\mathbf{p}_i) - \frac{\mathcal{I}'_{\beta,d}(\mathbf{q})}{\mathcal{I}'_{\alpha,d}(\mathbf{q})} * \mathcal{I}'_{\alpha,d}(\mathbf{p}_i) \right|}{\mathcal{N} * P} \quad (8)$$

in which $\mathcal{I}'_{\alpha,d}(\cdot)$ is the first-order derivative of a pixel in concerned direction α and distance d ; $\beta = \alpha + 45^\circ$; \mathbf{p}_i denotes the i^{th} neighbor of the current pixel \mathbf{q} in an image \mathcal{I} ; P is the number of considered neighbors; $\mathcal{N} = (\mathcal{W} - 2) * (\mathcal{H} - 2)$ where \mathcal{W} and \mathcal{H} are the width and height dimensions of 2D image \mathcal{I} respectively.

Second, a Directional Vector Center (DVC) threshold is defined as absolute multiplication of directional differences which are averaged on the whole image as follows.

$$\text{DVC}_{\alpha,d}(\mathcal{I}) = \frac{1}{\mathcal{N}} \sum_{\mathbf{q} \in \mathcal{I}} \left| \mathcal{I}'_{\alpha,d}(\mathbf{q}) * \mathcal{I}'_{\beta,d}(\mathbf{q}) \right| \quad (9)$$

where each pixel $\mathbf{q} \in \mathcal{I}$ is addressed in a pair of concerned directions (α, β) to form first-order derivatives correspondingly.

3.2 A completed model of LVP

Guo *et al.* [38] indicated that the integration of complementary components: local variations of magnitudes, centered contrast levels, and along with the typical LBP, leads to structuring effectively a descriptor with more robust and discriminative power. Inspired by this concept, we propose in this section, a completed model of the first-order LVP using the adaptive thresholds which are defined in Section 3.1. In essence, it is an integration of three following parts:

The first component is proposed to compute local vector patterns in each direction of $\alpha \in \Phi$ for a motion point \mathbf{q}_c as follows.

$$\text{LVP-D}_{P,R,\alpha,d}(\mathbf{q}_c) = \sum_{i=0}^{P-1} h(\mathcal{I}'_{\alpha,d}(\mathbf{q}_c), \mathcal{I}'_{\beta,d}(\mathbf{q}_c), \mathcal{I}'_{\alpha,d}(\mathbf{p}_i), \mathcal{I}'_{\beta,d}(\mathbf{p}_i)) 2^i \quad (10)$$

in which P is the number of considered neighbors sampled on a circle of radius R centered at \mathbf{q}_c , $\beta = \alpha + 45^\circ$, and function $h(\cdot)$ is defined as

$$h(x, y, u, v) = \begin{cases} 1, & \text{if } v \geq u * \frac{y}{x} \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

The fact that each LVP-D pattern is similar to the basic LVP [47], except that it is separately encoded in a binary string for each concerned direction instead of the combination of all into one long pattern for the whole directions as the typical LVP (see Fig. 2 for an example of this computation). Indeed, it is possible to utilize popular mappings (e.g., *u2*, *riu2*) for dimensional reduction.

The second, called LVP-M, captures magnitude variations of a motion point \mathbf{q}_c according to directions of Φ as follows:

$$\text{LVP-M}_{P,R,\alpha,d}(\mathbf{q}_c) = \sum_{i=0}^{P-1} \psi(\mathcal{I}'_{\alpha,d}(\mathbf{q}_c), \mathcal{I}'_{\beta,d}(\mathbf{q}_c), \mathcal{I}'_{\alpha,d}(\mathbf{p}_i), \mathcal{I}'_{\beta,d}(\mathbf{p}_i), \text{DVM}_{\alpha,d}(\mathcal{I})) 2^i \quad (12)$$

where function $\psi(\cdot)$ is defined as

$$\psi(x, y, u, v, t) = \begin{cases} 1, & \text{if } |v - u * \frac{y}{x}| \geq t \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

Third, LVP-C regards to the contrast level of \mathbf{q}_c in a direction α against the mean of directional differences on the whole image.

$$\text{LVP-C}_{\alpha,d}(\mathbf{q}_c) = s(\mathcal{I}'_{\alpha,d}(\mathbf{q}_c) - \text{DVC}_{\alpha,d}(\mathcal{I})) \quad (14)$$

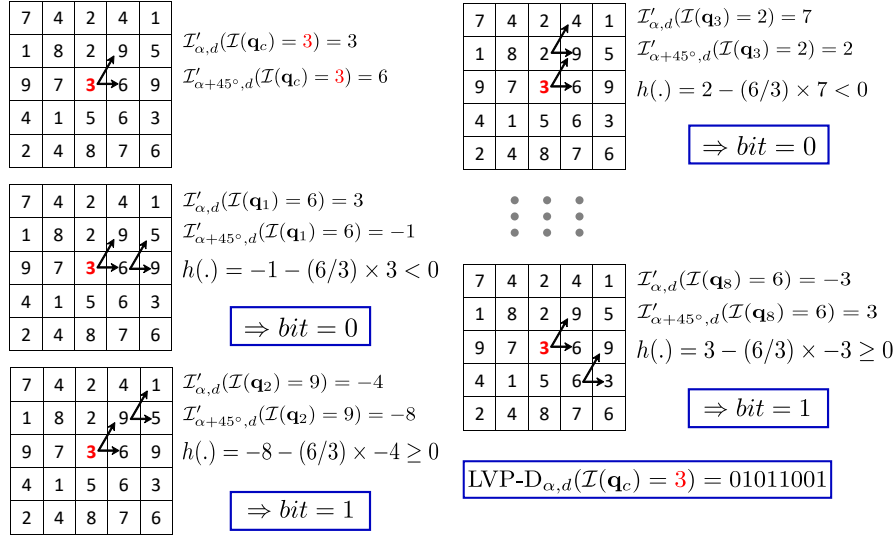


Figure 2: (Best viewed in color) Computing the first-order LVP-D binary pattern for a dynamic point $\mathcal{I}(\mathbf{q}_c) = 3$ (in red) with $\alpha = 0^\circ$, $d = 1$, and $(P, R) = (8, 1)$.

in which $s(\cdot)$ is defined by Equation (2).

These components (respectively abbreviated to LVP_D , LVP_M , and LVP_C) are supplementary to enrich more discriminative information. Therefore, they should be integrated together into different ways to enhance the discrimination power. Each integration makes a corresponding extended LVP operator, named xLVP in general. For example, $xLVP = LVP_{D-M/C}$ means that probability distributions structured by LVP_D , LVP_M , and LVP_C are respectively concatenated and jointed corresponding to the signals of “-” and “/” in style “ $_{D-M/C}$ ”. It should be noted that our xLVP operator can be also inferred to n^{th} -order derivative ($n > 1$) to capture high-order directional patterns ($xLVP^n$), as similarly as generated in [47].

Our xLVP operator takes into account several following properties to improve the performance in comparison with the basic LVP [47]:

- Based on complementary components, the xLVP operator is able to forcefully capture directional relationships in various contexts of local regions. In the meanwhile, LVP just considers one scale for computing local features.
- For each concerned direction, a directional pattern of the components is encoded in a separative binary string of 8 bits. In contrast to the basic LVP, its binary outputs are concatenated to form a long chain for all considered directions, e.g., a 32-bit string for the first-order LVP in four directions.

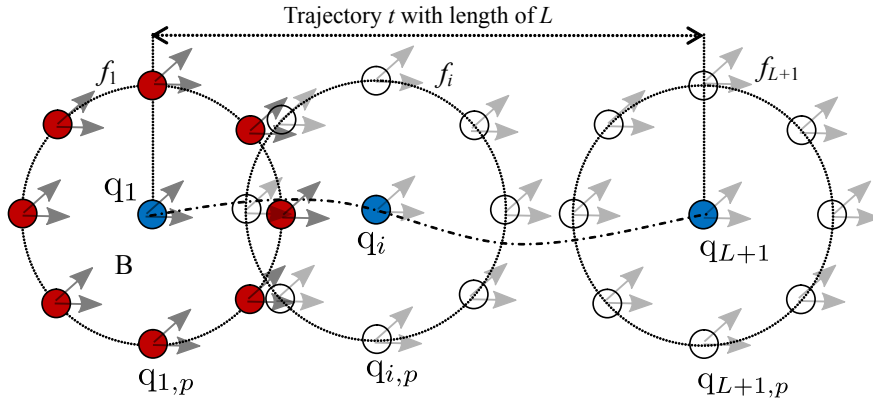


Figure 3: (Best viewed in color) A general model for encoding DBT patterns in which dense trajectory t with length of L is structured by $L + 1$ blue motion points located in consecutive frames along with their neighbors in different colors situated in a vicinity $B = \{8, 1\}$.

- Due to encoding directional features in separate chains of binary codes, it is possible to take advantage of two popular mappings of $riu2$ and $u2$ in order to enhance the discriminative power of descriptor with a reasonable dimension. In contrast, the conventional LVPs are calculated on sub-regions of a texture image and the obtained spatial histograms are adopted into equal interval by using a method of uniform quantization [47].

4 Beneficial properties of dense trajectories

Dense trajectories, introduced in [4], are traces of dense motion points which are tracked through in a certain number of frames based on the information of their displacements in a video. Exploiting robust properties of these complex motions, dense-trajectory-based methods are interested in analyzing videos for action recognition [4, 53], object segmentation [54], etc. In our framework, we take this approach for the first time into account DT representation by concerning motion of dynamic textures in consideration of different local directions to address two important properties: directional beams of dense trajectories and spatio-temporal characteristics of motion points along their paths. Hereunder, we present in detail a novel concept for embedding dense trajectories in accordance with the completed model xLVP to figure out directional trajectory-based patterns with more discrimination. In the other hand, the advantages of both optical-flow-based and local-feature-based techniques are wedged into our proposed framework for DT representation.

4.1 Directional features of a beam trajectory

Let $t = \{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_L, \mathbf{q}_{L+1}\}$ be a dense trajectory with length of L which is structured by motion point \mathbf{q}_1 and its inferred derivations (i.e., $\{\mathbf{q}_2, \dots, \mathbf{q}_L, \mathbf{q}_{L+1}\}$) through $L + 1$ consecutive frames $\{f_1, f_2, \dots, f_L, f_{L+1}\}$. We address directional movements of each motion point $\mathbf{q}_i \in t$ and its local neighbors sampled by a vicinity of B (see Fig. 3 for a graphical illustration) to estimate dynamic features for chaotic motions as well as their spatial characteristics along trajectory t using the completed operator xLVP in directions of Φ . The obtained histograms are then concatenated to form directional beam trajectory (DBT) patterns of t , efficiently describing the directional moving cues of beams of dynamic points.

$$\text{DBT}_{L,\Phi,d}(t) = \left[\sum_{i=1}^{L+1} H_{\mathbf{q}_i}(\text{xLVP}_{P,R,\Phi,d}(\mathbf{q}_i, f_i)), \right. \\ \left. \biguplus_{\mathbf{p}_j \in B} \left[\sum_{i=1}^{L+1} H_{\mathbf{p}_j}(\text{xLVP}_{P,R,\Phi,d}(\mathbf{p}_j, f_i)) \right] \right] \quad (15)$$

in which $\text{xLVP}(\cdot)$ means completed local vector pattern of a pixel at a frame in consideration of its local neighbors P sampled by a circle of radius R with a given distance d and concerned directions Φ ; \mathbf{p}_j refers to the j^{th} neighbor of motion point \mathbf{q}_i in supporting region B at frame f_i ; $H_{\mathbf{q}_i}(\cdot)$ and $H_{\mathbf{p}_j}(\cdot)$ are probability distributions of \mathbf{q}_i and its neighbors respectively; \biguplus denotes a concatenating function for the obtained histograms $H_{\mathbf{p}_j}(\cdot)$.

4.2 Spatio-temporal features of motion points

The spatio-temporal information of a voxel in a DT video is crucial in analysis to make it more “understandable” as exploited in [9, 39, 43], in which the authors determined the shape and motion cues based on three orthogonal planes. In this section, we take this concept into account motion points of dense trajectory t to boost the performance of DT descriptor. Because of the fact that the spatial information of those along t has been involved in the DBT model, we just address the temporal features in consideration of those on XT and YT planes using the completed operator xLVP. To be in accordance with encoding of DBT features of t with length of L , the obtained probability distributions should be concatenated through their trajectory t , as graphically demonstrated in Fig. 4, in order to form directional structures of temporal motion points (TMP) as

$$\text{TMP}_{L,\Phi,d}(t) = \left[H_{XT}(\text{xLVP}_{P,R,\Phi,d}(\mathbf{q}_i)), \right. \\ \left. H_{YT}(\text{xLVP}_{P,R,\Phi,d}(\mathbf{q}_i)) \right]_{i=1}^{L+1} \quad (16)$$

where $\text{xLVP}(\cdot)$ denotes completed local vector pattern of a pixel computed by considering its local neighbors P interpolated by a circle of radius R with a given distance d in concerned directions Φ ; $H_{XT}(\cdot)$ and $H_{YT}(\cdot)$ are histograms of motion point \mathbf{q}_i calculated for the corresponding planes.

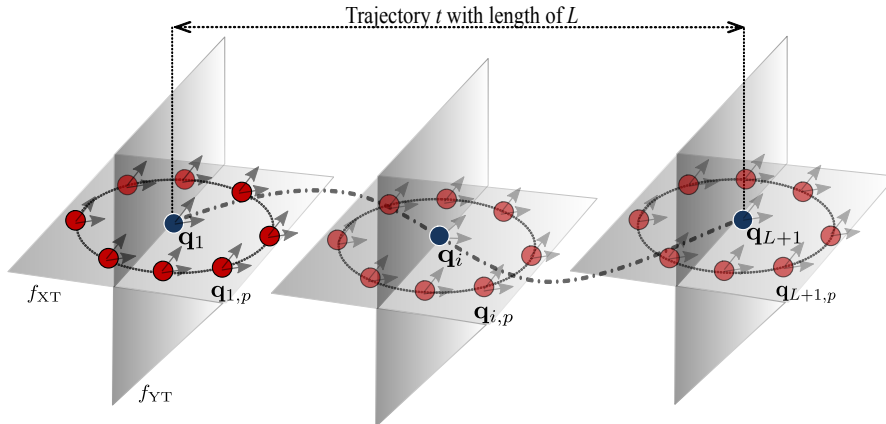


Figure 4: (Best viewed in color) A typical TMP model in which directional temporal information of motion points (in blue) are encoded along their trajectory t with length of L by exploiting directional relations of those with their local neighbors $P = 8$ (in red) sampled by a circle of radius $R = 1$ on XT and YT planes.

5 Directional dense trajectory patterns for DT representation

In this section, we introduce an efficient framework for DT representation, called Directional Dense Trajectory Patterns (DDTP), in which DT features of a video are effectively encoded just using dense trajectories instead of the whole video. On the other hand, our perception is to take advantage of two important properties of directional dense trajectories for constructing robust descriptors for DT recognition, as graphically illustrated in Fig. 5. According to that, dense trajectories are extracted at first using the tool introduced in [4]. We then apply our extended operator xLVP on those to capture their directional motion cues through encoding patterns of directional beam trajectories, as proposed in Section 4.1. This completed operator is also implemented for capturing spatio-temporal structures of motion points along their trajectories based on analysis of the planes, as presented in Section 4.2. Lastly, the obtained probability distributions of two above components calculated for the whole dense trajectories are concatenated and normalized to enhance the performance. Also in this section, the computational complexity of DDTP is discussed thoroughly for potential applications in practice. Those above processes are detailed hereafter.

5.1 Proposed DDTP descriptor

Let $\mathcal{T} = \{t_1, t_2, \dots, t_m\}$ denote a set of dense trajectories with the same length of L which are extracted from a video \mathcal{V} . DBT patterns of each $t_i \in \mathcal{T}$ are then encoded in consideration of its motion points along the path in directions Φ

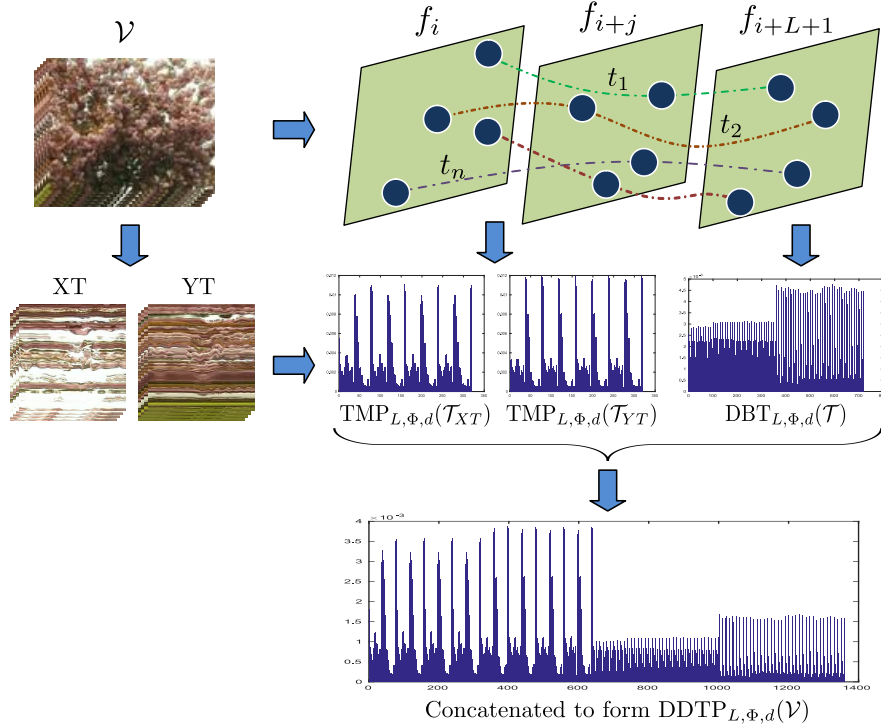


Figure 5: (Best viewed in color) An effective framework for DT representation based on dense trajectories extracted from a video \mathcal{V} .

using the completed model xLVP. Parallel to this encoding, TMP patterns are also structured by addressing xLVP with the directions for the corresponding motion points of trajectory t_i based on analysis of those on the temporal planes of \mathcal{V} (i.e., XT , YT). To form a robust and discriminative descriptor for DT recognition, we concatenate and normalize DBT and TMP features that are computed for all of trajectories in \mathcal{T} as

$$DDTP_{L,\Phi,d}(\mathcal{V}) = \frac{1}{|\mathcal{T}|} \sum_{t_i \in \mathcal{T}} [DBT_{L,\Phi,d}(t_i), TMP_{L,\Phi,d}(t_i)] \quad (17)$$

in which $|\mathcal{T}|$ denotes the total of dense trajectories. From now on, we imply a specific DDTP descriptor in agreement with an integration way of completed operator xLVP. For instance, $DDTP_{D-M/C}$ indicates that it is structured by $xLVP = LVP_{D-M/C}$ (see Section 3.2 for a detail of this integration).

In order to verify the prominent contribution of our completed operator xLVP, a basic descriptor DDTP-B which is based on the first-order LVP (i.e.,

LVP_D), is concerned by addressing the same implementation above.

$$\text{DDTP-B}_{L,\Phi,d}(\mathcal{V}) = \frac{1}{|\mathcal{T}|} \sum_{t_i \in \mathcal{T}} [\text{DBT-B}_{L,\Phi,d}(t_i), \text{TMP-B}_{L,\Phi,d}(t_i)] \quad (18)$$

where DBT-B and TMP-B are respectively computed as similarly as in Equations (15) and (16) but only LVP_D is used instead of xLVP.

To evaluate the expected effectiveness of exploiting beneficial properties of dense trajectories for DT description in contrast to using the whole video, xLVP is taken into account structuring dynamic features on three orthogonal planes $\{XY, XT, YT\}$ to form another DT descriptor, named xLVP-TOP as follows.

$$\begin{aligned} \text{xLVP-TOP}_{\Phi,d}(\mathcal{V}) = & [\text{xLVP}_{P,R,\Phi,d}(\mathcal{V}_{XY}), \\ & \text{xLVP}_{P,R,\Phi,d}(\mathcal{V}_{XT}), \\ & \text{xLVP}_{P,R,\Phi,d}(\mathcal{V}_{YT})] \end{aligned} \quad (19)$$

On the other hand, for assessing our crucial extended model of LVP, we have also experimented on DT recognition using LVP-TOP descriptor formed by the basic LVP operator [47] on planes of $\{XY, XT, YT\}$ as

$$\begin{aligned} \text{LVP-TOP}_{\Phi,d}(\mathcal{V}) = & [\text{LVP}_{P,R,\Phi,d}(\mathcal{V}_{XY}), \\ & \text{LVP}_{P,R,\Phi,d}(\mathcal{V}_{XT}), \\ & \text{LVP}_{P,R,\Phi,d}(\mathcal{V}_{YT})] \end{aligned} \quad (20)$$

where $\text{LVP}_{P,R,\Phi,d}(\cdot)$ is a probability distribution. It is actually dealt with as similarly as LVP-D's (see Section 3.2) to take advantage of the popular mappings in dimensional reduction.

In order to reduce the size of DDTP descriptors, two popular mappings are utilized: *riu2* giving $l_{riu2} = (P + 2)$ and *u2* giving $l_{u2} = (P(P - 1) + 3)$ distinct bins for each pattern of a pixel, where P is a number of local neighbors taken into account. Particularly, dimension of DDTP descriptors directly relies on the integration of complementary components in specific ways to form xLVP for computing DBT and TMP features. For example, $\text{DDTP}_{D-M/C}$ has the total bins of two following components: $\text{DBT}_{D-M/C}$ and $\text{TMP}_{D-M/C}$ with $3k(|B| + 1)$ and $6k(L + 1)$ dimensions respectively, in which $|B|$ means the cardinality of local neighbors sampled around a motion point for encoding directional beams of trajectories with the same length of L , $k = l_{riu2}/u2 \times |\Phi|$ is the dimension of a pattern encoded by the completed operator $\text{xLVP} = \text{LVP}_{D-M/C}$ with *riu2/u2* mappings in consideration of a number of concerning directions $|\Phi|$. As the result of those, the final size of $\text{DDTP}_{D-M/C}$ is $3k(|B| + 2L + 3)$ bins. Similarly, dimension of $\text{xLVP-TOP}_{D-M/C}$ descriptor is $9k$ bins; of the original LVP-TOP is $3k$; and of DDTP-B is the one-third of $\text{DDTP}_{D-M/C}$'s in this case since only LVP_D is involved with.

In order to effectively form DDTP descriptor, Algorithm 1 presents our idea for its construction based on a mechanism of *shared features*, in which xLVP

features of each frame are calculated for only one time and are used effectively for constructing DDTP description of all trajectories passing through this frame. It is proposed by addressing three main following steps:

1. Labeling all motion points of trajectories with mapping volume vMP.
2. Constructing xLVP features of the considered video.
3. Calculating DDTP of each trajectory from the labels of its motion points (vMP) and xLVP features.

Moreover, we also take advantage of multi-scale analysis [55] to improve the discriminative power of DDTB descriptors, in which our xLVP is exploited for many of different $\{(P, R)\}$ situations in order to forcefully capture directional relationships in further local regions. The obtained histograms are then concatenated and normalized to structure multi-scale DT representation.

Our proposed DDTP descriptor has more robust and discriminative power based on the following prominent properties:

- Incorporation between DBT and TMP features makes DDTP descriptors more discriminative for DT recognition (see Table 4 for contributions of each of them).
- The advantages of both optical-flow-based and local-feature-based methods are embedded into DDTP descriptors thanks to utilizing xLVP for encoding dense trajectories.
- Using dense trajectories extracted from a video allows to efficiently analyze chaotic motions of moving DTs in the sequence, an interested alternative for DT representation.

5.2 Computational complexity of DDTP descriptor

In order to estimate the computational complexity of our DDTP descriptor, we present a simple algorithm to encode DDTP patterns, as generally shown in Algorithm 1. Accordingly, it takes five steps to handle a video \mathcal{V} of $\mathcal{H} \times \mathcal{W} \times F$ dimension as follows.

- *Step 1:* Dense trajectories \mathcal{T} with length of L are extracted by exploiting a tool introduced in [4]. The computational cost of this extraction $Q_{\mathcal{T}}$ can be referred to [4] for more detail.
- *Step 2:* A mapping volume vMP is used to signed which motion points belong to which trajectory $t \in \mathcal{T}$. The complexity is estimated as $Q_{\text{vMP}} = \mathcal{O}(L \times |\mathcal{T}|)$.
- *Step 3:* xLVP features are calculated from collection of slices of \mathcal{V} in three orthogonal planes XY , XT , and YT . Let us consider plane XY concerning component xLVP_{XY} (the two other components have the same

Algorithm 1: Encoding DDTP patterns

```
1 Input: A video  $\mathcal{V}$  of  $\mathcal{H} \times \mathcal{W} \times F$  dimension, length of trajectory  $L$ ,  
   number of neighbors  $P$ , directions  $\Phi$ .  
2 Output: DDTP descriptor.  
3 Step 1: Extraction of trajectories. %%%%  
4 Extracting dense trajectories  $\mathcal{T}$  from video  $\mathcal{V}$  subject to  $L$ .  
5 Step 2: Labeling of motion points. %%%%  
6 Initialize vMP of size  $\mathcal{H} \times \mathcal{W} \times F$ ,  $\text{vMP}(\mathbf{q}) = 0 \forall \mathbf{q}$ .  
7 for  $t=1:|\mathcal{T}|$  do  
8   for  $i=1:L+1$  do  
9      $\mathbf{q}_i = i^{\text{th}}$  motion point of trajectory  $\mathcal{T}(t)$ ;  
10     $\text{vMP}(\mathbf{q}_i) = t$ ;  
11  end for  
12 end for  
13 Step 3: Extraction of xLVP features. %%%%  
14 for  $f=1:F$  do  
15    $\mathcal{I}_f$ : slice of  $\mathcal{V}$  at frame  $f$  in plane  $XY$ ;  
16    $\text{xLVP}_{XY}(f) = \{\text{LVP-D}(\mathcal{I}_f), \text{LVP-M}(\mathcal{I}_f), \text{LVP-C}(\mathcal{I}_f)\}$ ;  
17 end for  
18 for  $y=1:H$  do  
19    $\mathcal{I}_y$ : slice of  $\mathcal{V}$  at ordinate  $y$  in plane  $XT$ ;  
20    $\text{xLVP}_{XT}(y) = \{\text{LVP-D}(\mathcal{I}_y), \text{LVP-M}(\mathcal{I}_y), \text{LVP-C}(\mathcal{I}_y)\}$ ;  
21 end for  
22 for  $x=1:W$  do  
23    $\mathcal{I}_x$ : slice of  $\mathcal{V}$  at abscissa  $x$  in plane  $YT$ ;  
24    $\text{xLVP}_{YT}(x) = \{\text{LVP-D}(\mathcal{I}_x), \text{LVP-M}(\mathcal{I}_x), \text{LVP-C}(\mathcal{I}_x)\}$ ;  
25 end for  
26 Step 4: Construction of DBT and TMP %%%%  
27 for each  $\mathbf{q} \in \text{vMP}$  do  
28   Check  $\mathbf{q}$  is motion point. %%%%  
29   if  $\text{vMP}(\mathbf{q}) > 0$  then  
30     Structuring DBT and TMP features based on  $\text{xLVP}_{XY}$ ,  
      $\text{xLVP}_{XT}$ ,  $\text{xLVP}_{YT}$  for motion points  $\mathbf{q}$  in the trajectory  
      $t = \text{vMP}(\mathbf{q})$ .  
31   end if  
32 end for  
33 Step 5: Construction of DDTP. %%%%  
34 Concatenate to form DDTP = [DBT, TMP];
```

complexity by using similar arguments). We consider now the complexity to calculate xLVP features for each input plane-image \mathcal{I}_f of $\mathcal{H} \times \mathcal{W}$ dimension, it can be deduced from Equations (8) and (9) in Section 3.1 that our proposed directional thresholds DVM and DVC have computational costs of $Q_{\text{DVM}} = \mathcal{O}(P \times \mathcal{H} \times \mathcal{W})$ and $Q_{\text{DVC}} = \mathcal{O}(\mathcal{H} \times \mathcal{W})$ respectively, where P is the number of considered neighbors for encoding xLVP. As mentioned in Section 3.2, our xLVP consists of three complementary components: LVP_D , LVP_M , and LVP_C . Based on Equations (10), (12), and (14), their computation costs are respectively estimated as $Q_{\text{LVP}_D} = \mathcal{O}(P \times \mathcal{H} \times \mathcal{W})$, $Q_{\text{LVP}_M} = \mathcal{O}(P \times \mathcal{H} \times \mathcal{W}) + Q_{\text{DVM}}$, and $Q_{\text{LVP}_C} = \mathcal{O}(\mathcal{H} \times \mathcal{W}) + Q_{\text{DVC}}$. Since these components are computed independently, the complexity of $\text{xLVP}(\mathcal{I})$ can be approximately estimated as the maximum of Q_{LVP_D} , Q_{LVP_M} , and Q_{LVP_C} , i.e., $\mathcal{O}(P \times \mathcal{H} \times \mathcal{W})$. Therefore, the complexity for extraction of xLVP_{XY} component on XY plane is $\mathcal{O}(P \times \mathcal{H} \times \mathcal{W} \times F)$ because there are F considered slices. By applying similar arguments on two other components calculated on planes YT and XT , we deduce that the complexity of this step is $Q_{\text{xLVP}} = \mathcal{O}(P \times \mathcal{H} \times \mathcal{W} \times F)$.

- *Step 4:* Based on the mapping volume vMP, DBT and TMP features are structured by using xLVP patterns for motion points in the same trajectory. The complexities of these processes are estimated as $Q_{\text{DBT}} = \mathcal{O}(P \times L \times |\Phi| \times \mathcal{H} \times \mathcal{W} \times F)$ for encoding DBT features and $Q_{\text{TMP}} = \mathcal{O}(L \times |\Phi| \times \mathcal{H} \times \mathcal{W} \times F)$ for TMP, in which $|\Phi|$ denotes the cardinality of directions Φ .
- *Step 5:* Finally, DDTP descriptor is formed by concatenating DBT and TMP features. The complexity of this concatenation is $\mathcal{O}(1)$.

Therefore, the complexity of our proposed descriptor can be generally estimated as follows.

$$Q_{\text{DDTP}} = Q_{\mathcal{T}} + Q_{\text{vMP}} + Q_{\text{xLVP}} + Q_{\text{DBT}} + Q_{\text{TMP}} \quad (21)$$



In order to concentrate on the computational cost of our proposed DDTP descriptor based on a given collection of dense trajectories, we disregard $Q_{\mathcal{T}}$. In addition, since parameters L and $|\Phi|$ (e.g., $L \in \{2, 3\}$ and $|\Phi| = 4$ as valued in Section 6.1) are much smaller than the others, they can be also ignored. Consequently, Q_{DDTP} could be approximated by Equation (22), which shows that the construction of DDTP descriptor from dense trajectories has linear complexity with respect to the number of voxels of an input video since P can be considered as a constant, i.e., $P = 8$ or $P = 16$.

$$\begin{aligned} Q_{\text{DDTP}} &\approx \max(Q_{\text{vMP}}, Q_{\text{xLVP}}, Q_{\text{DBT}}, Q_{\text{TMP}}) \\ &\approx \mathcal{O}(P \times \mathcal{H} \times \mathcal{W} \times F) \end{aligned} \quad (22)$$

In terms of processing time, the consumption mainly depends on the turbulent level of DTs in a video, i.e., the more turbulence the video has, the larger motion points are signed in mapping volume vMP (see lines 4-12 of Algorithm

1), and then the heavier computation of DBT and TMP is (see lines 27-31 of Algorithm 1). However, it can be verified from Equation (22) that our proposal principally depends on the dimension of a given video, not on the number of its trajectories. Indeed, in consideration of videos with the same dimension but levels of turbulence in high difference, we address two particular videos of UCLA in both original and cropped versions for an instance of runtime estimation. Table 1 illustrates the consumption of encoding $DDTP_{D-M/C}$ descriptors with settings of $L = 3$, $P = 8$, and $|\Phi| = 4$. It can be seen from Table 1 that the higher turbulent video needs more processing time. In addition, using the cropped version can save the runtime, but it negatively impacts the performances for DT recognition (see Table 7 for instances). It is worth noting that a raw MATLAB code of our algorithm is run on a 64-bit Linux desktop of CPU Core i7 3.4GHz, 16G RAM.

Table 1: Comparing processing time of encoding two videos in UCLA.

Sample video	Resolution	L	Level of turbulence	#Traj.	Runtime
	$110 \times 160 \times 75$ (orig.)	3	A single candle flame	3674	$\approx 8.7s$
	$48 \times 48 \times 75$ (crop)	3	A single candle flame	1507	$\approx 2.6s$
	$110 \times 160 \times 75$ (orig.)	3	All leaf vibrations	25562	$\approx 35.3s$
	$48 \times 48 \times 75$ (crop)	3	All leaf vibrations	2134	$\approx 3.1s$

6 Experiments

In this section, comprehensive evaluations of the proposed framework on the benchmark DT datasets (i.e., UCLA [1], DynTex [24], and DynTex++ [25]) are specifically expressed by following experimental protocols and parameter settings for implementation. In order to classify DTs, we addressed two following popular classifiers: *i) Support Vector Machines (SVM)* - We use a linear SVM with the default parameters implemented in library LIBLINEAR [56]. *ii) k-nearest neighbors (k-NN)* - To be comparable with performances of existing approaches [57, 58, 39, 43], we also employ k-nearest neighbors in simplicity with $k = 1$ (i.e., 1-NN), in which chi-square (χ^2) is used for dissimilarity measure. The obtained recognition rates are then evaluated in comparison with those of the state-of-the-art methods.

6.1 Experimental settings

Settings for extracting dense trajectories: Due to the short “living” time of most of turbulent dynamic points in DT videos, lengths of dense trajectories $L \in \{2, 3\}$ should be addressed in our experiments. We utilize a tool, introduced in [4], for extracting these trajectories from a DT sequence. Since the default

settings of this tool are set for mainly achieving motions of human actions, to be in accordance with the particular DT characteristics, we make a change of rejecting trajectory parameter $min_var = 5 \times 10^{-5}$ in order to acquire “weak” trajectories of chaotic motion points. Figure 6 graphically illustrates several samples of dense trajectories extracted from the corresponding sequences using the customized settings. Empirically, for datasets (like DynTex++) which are built by splitting from other original videos, some of cropped sequences point out a number of trajectories that are not sufficient for DT representation (see Figure 6(c)). In this case, a few tracking parameters should be addressed in lower levels to boost the quantity of trajectories in our framework as $quality = 10^{-8}$ and $min_distance = 1$.

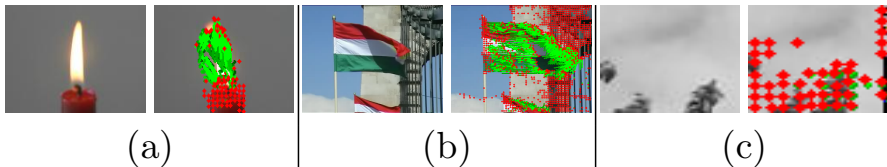


Figure 6: (Best viewed in color) Samples (a), (b), (c) of dense trajectories extracted from the corresponding videos in UCLA, DynTex, and DynTex++ datasets respectively in which green lines show paths of motion points through the consecutive frames.

Parameter settings for structuring descriptors: The first-order xLVP operator (i.e., $d = 1$) is used to structure local vector patterns of dynamic features in four directions of $\Phi = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$, i.e., $|\Phi| = 4$. To be compliant with the LBP-based concept, it is possible to conduct different supporting regions $\Omega = \{B_i\}$ for encoding directional beams of dense trajectories DBT, where each $B_i = \{P_{B_i}, R_{B_i}\}$ denotes P_{B_i} neighbors circled by radius R_{B_i} . In our experiments, we address $\Omega = \{\{8, 1\}, \{16, 2\}\}$ (see Fig. 3 for an instance of $B_i = \{8, 1\}$, i.e., $|B_i| = 8$, which is taken into account.) To be in accordance with the DBT calculation, locating local neighbors $\{(P, R)\}$ for computing TMP on the temporal planes should be agreed with the way of addressing Ω . For different types of DDTP descriptor, structured subject to integrating complemented components in xLVP operator, we address three descriptors for experiments on DT classification, i.e., $DDTP_{D_M}$, $DDTP_{D_M/C}$, and $DDTP_{D_M/C}$ (hereafter generally named DDTP descriptors). Their dimensions are respectively $8\eta l_{riu2/u2}$, $12\eta l_{riu2/u2}$, and $8\eta(l_{riu2/u2} + 1)$ with $riu2/u2$ mappings, where $\eta = |B_i| + 2L + 3$. Similarly, we have various xLVP-TOP descriptors as follows: xLVP-TOP $_{D_M}$ of $24l_{riu2/u2}$ bins, xLVP-TOP $_{D_M/C}$ of $36l_{riu2/u2}$, and xLVP-TOP $_{D_M/C}$ of $24(l_{riu2/u2} + 1)$. In terms of DDTP-B, and LVP-TOP descriptors, they are structured by $4\eta l_{riu2/u2}$ and $12l_{riu2/u2}$ distinct values in this case. Table 2 details some specific dimensions of these descriptors of $riu2$ mapping. It is verified from this table that multi-scale analysis is able to be regarded for our completed operator xLVP to capture more robust directional relationships in larger supporting regions while the dimension is still moderate

compared to other LBP-based methods.

6.2 Datasets and experimental protocols

In this section, we firstly detail features and protocols of benchmark DT datasets for verifying our framework in recognition issue. Their main properties are then summarized in Table 3 at a glance.

6.2.1 UCLA dataset

It consists of 50 categories with 200 different DT videos [1], corresponding to four sequences per class, which capture chaotic motions, such as fire, boiling water, fountain, etc. Each original sequence is recorded in 75 frames with dimension of 110×160 . Several samples of this dataset are shown in above row of Fig. 7. The categories are arranged in various ways to compose challenging subsets as follows.

- *50-class*: Usually, 50 categories are addressed for DT recognition in two experimental protocols as follows.

Leave-one-out (LOO) [23, 44, 20]: Only one sequence of a class is taken out for testing. The training set is addressed by taking the remain of this class along with all videos of other classes. The final rate is averaged by repeating this trial for all samples of the dataset.

4-fold cross validation [39, 43, 18]: Three videos of each class are picked out for learning and the remaining one for testing. This process is repeated four times with different testing samples. Then the average of these repetitions forms the final result.

- *9-class*: Original 50 categories are rearranged to form 9 classes with more challenge for DT recognition: “boiling water”(8), “fire”(8), “flowers”(12), “fountains”(20), “plants”(108), “sea”(12), “smoke”(4), “water”(12), and “waterfall”(16), where numbers in parentheses indicate quantities of sequences in the corresponding group [20, 26]. Following protocols in [39, 25], a half of sequences in each category is randomly selected for training, the rest for testing. The average rate of 20 runtimes reports the final result.
- *8-class*: Due to the dominant quantity of sequences in “plants” group, it is eliminated to form 8-class scheme with more challenges for recognition task. Similarity to [20, 39], a half of sequences in each category is randomly selected for training and the remaining for testing. The final evaluation of recognition is estimated by the mean of rates in 20 runtimes.

6.2.2 DynTex dataset

It is a collection of more than 650 high-quality DT sequences in AVI format which are recorded in various conditions of environment [24]. Following the



Figure 7: Samples of UCLA (above row) and DynTex (bottom row).

works in [23, 43, 9], the version of $352 \times 288 \times 250$ videos is addressed in our experiments with LOO protocol for evaluating the performance of our proposal. (see Fig. 7 for some DT samples). There are 4 challenging subsets which are composed from the original sequences for DT recognition task as follows.

- *DynTex35* is arranged from 35 videos in order to form corresponding 35 categories as follows. Each of sequence is randomly split into 8 non-overlapping sub-videos so that cutting points are not in half of the X, Y, and T axes. In addition, two more sub-sequences are also pointed out by cutting along the temporal axis of the original sequence. Consequently, there are 10 sub-DTs with different spatial-temporal dimension for each sequence [23, 43, 9].
- *Alpha* includes 60 videos equally divided into three categories: “sea”, “grass”, and “trees”, with 20 sequences in each of them.
- *Beta* contains 162 sequences grouped into 10 classes with different numbers of sequences: “sea”, “vegetation”, “trees”, “flags”, “calm water”, “fountains”, “smoke”, “escalator”, “traffic”, and “rotation”.
- *Gamma* comprises 10 categories with 264 videos in total: “flowers”, “sea”, “naked trees”, “foliage”, “escalator”, “calm water”, “flags”, “grass”, “traffic”, and “fountains”. Each of which includes a sample of diverse sequences.

6.2.3 DynTex++ dataset

From more than 650 sequences of the original DynTex, Ghanem *et al.* [25] filtered 345 raw videos to build DynTex++ so that the filtered videos only contain the main DTs, not consist of other DT features such as panning, zooming, and dynamic background. After applying some techniques of preprocessing to the selected videos, they are divided into 36 categories where each of which has 100 sequences with fixed dimension of $50 \times 50 \times 50$, i.e., 3600 videos in total.

As the settings in [23, 25], the training set is formed by randomly selecting a half of samples from each group and the rest for testing. The final evaluation is taken by the average of 10 repetitions of this trial.

6.3 Experimental results

Evaluations of our framework for DT recognition on various benchmark datasets (UCLA, DynTex, and DynTex++) are specifically expressed in Tables 5, 6, and 10 respectively, in which descriptors DDTP and DDTP-B are formed by corresponding operators xLVP and LVP_D using *riu2* mapping for dense trajectories with length $L = \{2, 3\}$. It can be verified from those tables that addressing dense trajectories for DT description is a significant alternative beside considering DT appearances in temporal aspects of a video as in the existing methods. Based on the experimental results, several critical assessments could be derived from as follows.

- It can be verified from Tables 5, 6, and 10 that our proposed descriptors have much better results in classifying DTs when using SVM classifier compared to 1-NN. Therefore, SVM should be recommended for our below evaluations as well as for applications in practice. From now on, if no classifier is explicitly indicated for the DT recognition issue, the mentioned rates are based on SVM.
- As expected in Section 5.1, the incorporation between spatio-temporal of motion points (TMP) and directional features of beam trajectories (DBT) has boosted the performance in comparison with FDT [16], in which motion points of dense trajectories along with their local neighbors are encoded to form directional beams of features (see Tables 8 and 9). Table 4 expresses contributions of these components making DDTP descriptors more discriminative. Furthermore, our descriptors have dimension at least a half slighter than FDT’s (see Table 2).
- As mentioned in Section 3.2, the integration of complemented components additionally produces more informative discrimination for encoding dense trajectories. In fact, most of DDTP descriptors outperform significantly in comparison with DDTP-B, just utilizing one complemented factor (see Tables 5, 9, and 10). It has verified the contributions of our important extensions to form the completed xLVP operator compared to the basic LVP [47].
- Taking directional vector center contrast, i.e., LVP-C, into account structuring DDTP descriptors is frequently more robust than others. Therein, the jointing with this component seems to point out descriptors with more “stable” performance (see Tables 5, 6, and 10).
- It is in accordance with our analysis in Section 5.1 that capturing directional features of dense trajectories in multi-scale local regions of their motion points is more effective than single-scale. Therein, the 2-scale

D_{-M_C} descriptor of *riu2* mapping with length of trajectories $L = 3$, i.e., $\{(8, 1), (16, 2)\}_{L=3}^{riu2}$, obtains more “stable” on most of the benchmark datasets (see Tables 5, 6, and 10). Therefore, it should be suggested for applications in practice, and also be the setting selected for comparing with performances of state of the art.

- In most of circumstances, the performance of DDTP-B based on the typical LVP [47] (see Section 5.1) is not better than DDTP’s computed by the extended operator xLVP. Moreover, xLVP-TOP also outperforms compared to LVP-TOP in consideration of each voxel on three orthogonal plans of a video instead of its dense trajectories (see Table 11). These facts prove the interest of our proposed components: completed operator xLVP with two adaptative directional vector thresholds (i.e., DVM, DVC) and dense-trajectory-based features for DT representation.

In terms of comparison with the state-of-the-art methods, our proposed framework for encoding dense trajectories using completed model xLVP produces discriminative descriptors for DT recognition task compared to LBP-based variants and others in several circumstances. Furthermore, their performances are nearly the same those of deep-learning-based approaches on UCLA dataset (see Table 8). Hereinafter, comprehensive estimations of our proposal on various benchmark datasets are presented in detail, in which if DDTP descriptors are not explicit in their implemented settings, the default configuration is indicated for them, i.e., $\{(8, 1), (16, 2)\}^{riu2}$.

6.3.1 Recognition on UCLA dataset

It can be observed from Tables 5 and 8 that our proposed descriptors have significant performances of DT recognition on UCLA compared to those of state-of-the-art methods, including deep learning techniques in several circumstances, which are expressed in detail as follows.

In scenario of DT classification on *50-class*, by addressing trajectories of $L = \{2, 3\}$, $DDTP_{D-M}^{L=\{2,3\}}$ and $DDTP_{D-M-C}^{L=\{2,3\}}$ have reported rates of 100% on both *50-LOO* and *50-4fold* schemes, the best performances compared to all existing methods, including deep-learning approaches. In the meanwhile, with the setting for comparison, $DDTP_{D-M/C}^{L=3}$ descriptor gains 99% and 99.5% respectively, the highest compared to all LBP-based variants (see Table 8). Those performances are the same FDT’s [16], but in a half smaller dimension, i.e., 6768 versus over 13000 bins (see Table 2). On the other hand, DDTP-B using the setting of $\{(8, 1), (16, 2)\}_{L=3}^{riu2}$ also obtains competitive results with rates of 99.5% and 98.5% in comparison with those of the local-feature-based methods. Above facts have validated that utilizing dense trajectories along with the completed model of LVP for encoding directional features of motion points figures out discriminative descriptors in DT recognition task.

In terms of evaluations on *9-class* and *8-class*, our descriptor $DDTP_{D-M/C}^{L=3}$ has critical performances with 98.75% and 98.04% respectively, the highest in comparison with the LBP-based variants (see Table 8), except CVLBC [42] with

rates of 99.20% and 99.02%. However, it is not better than ours on DynTex35 and DynTex++ datasets as well as not been verified on the challenging subsets of DynTex, i.e., *Alpha*, *Beta*, and *Gamma* (see Table 9). In our previous work, FDT [16] encoding motions of DTs along their trajectory is just better than DDTP $_{D-M/C}^{L=3}$ on *8-class* with rate of 99.57%, but in about twice larger dimension. Furthermore, it should be noted that DT-CNN [32] only outperforms ours on *8-class* with rates of 98.48% for framework AlexNet and 99.02% for GoogleNet. For improvement in further contexts, we concentrate on which videos have been confused with others. On scheme *9-class*, it can be observed from Fig. 8, DDTP $_{D-M/C}^{L=3}$ has mainly confused the motions of DTs in “Fire” sequences with those in “Smoke”; and the properties of trajectories in “Flowers” with those in “Plants”. The confusion on scheme *8-class* principally falls in the turbulent properties of “Fire” videos with those of “Fountains” and “Waterfall” (see Fig. 9).

In addition, it should be noted that several existing methods [43, 45, 58, 40] have experimented DT classification on the short version of UCLA with videos of $48 \times 48 \times 75$ dimension. Addressing those for our proposal, we achieved some results for DDTP $_{D-M/C}^{L=3}$ descriptor, as indicated in Table 7. Accordingly, its performance is noticeably reduced in comparison with those done on $110 \times 160 \times 75$ videos (see Tables 5 and 7). It could be lack of spatio-temporal information due to less dense trajectories that are extracted from the cropped version. However, the speed of encoding is much faster thanks to a sharp reduction of turbulence in the cropped version (see Table 1 for a comparison of time consumption). Therefore, a trade-off between the high rates and the processing time should be discreetly considered for real implementations.

6.3.2 Recognition on DynTex dataset

It can be verified from Tables 6 and 9 that the proposed framework outperforms significantly compared to most of the state-of-the-art methods. In general, DDTP descriptors with at least a half smaller dimension are more robust than our previous work FDT [16]. It is thanks to exploiting spatio-temporal features of motion points along their trajectories which are encoded by the completed LVP model rather the typical LVP [47]. Hereafter, we detail evaluations on each subset.

For DT recognition on *DynTex35*, DDTP $_{D-M/C}^{L=3}$ descriptor with 6768 bins achieves 99.71%, a little lower than CSAP-TOP’s [40] (100%) with 13200 bins. It is due to the very similar motions of DTs in videos as shown in Fig. 11(a) and Fig. 11(b). Figure 10 expresses specific rates of each category. In the meanwhile, FD-MAP and FDT descriptors in our previous work [16] just obtain rate of 98.86%. It is because only appearances of trajectories are involved with. The LBP-based method MEWLSP [45] also has the same our ability. However, it has not been verified on other challenging subsets, i.e., *Alpha*, *Beta*, and *Gamma* (see Table 9).

In respect of DT classification on other challenging subsets, DDTP $^{L=\{2,3\}}$ descriptors obtain 98.33% on *Alpha* using $\{(P, R)\} = \{(8, 1)\}$ of *riu2* mapping

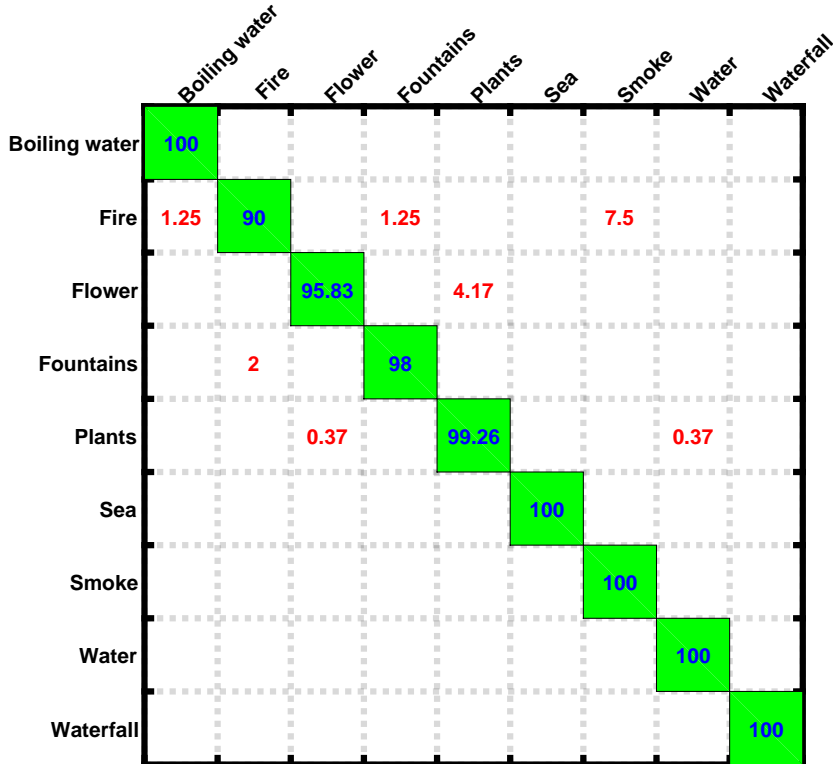


Figure 8: Confusion matrix of $DDTP_{D-M/C}^{L=3}$ on 9-class.

for both length of trajectories $L = \{2, 3\}$ (see Table 6), but not outperform on *Beta* and *Gamma* in comparison with other parameters. For the setting of comparison, $DDTP_{D-M/C}^{L=3}$ achieves a little lower rate of 96.67% on *Alpha* due to the mutual confusion between turbulent motions of DTs in “Trees” and those in “Grass” sequences (see Fig. 12). In the meantime, its performances on *Beta* and *Gamma* are 93.83% and 91.29%. Its modest results are caused by cases of confusion shown in Fig 13 and 14 respectively, where motions in “Escalator” and “Rotation” are confused with others in DT recognition on *Beta* while those in “Calm water” and “Fountains” on *Gamma*. In general, our performance is nearly the best results on these challenges compared to most of the existing approaches, except deep learning methods. Moreover, the execution of $DDTP_{D-M/C}^{L=3}$ is the same those of CSAP-TOP [40], FD-MAP [16], and FDT [16] (see Table 9), but in much smaller dimension, i.e., 6768 versus over 13000 bins of them (see Table 2). In the scenarios, $DDTP-B$ with the setting of $\{(8, 1), (16, 2)\}_{L=3}^{riu2}$ also gains significant rate of 98.33% on *Alpha*, but faulting on the remains since just directional features of the typical LVP are exploited. The deep learning methods, i.e., st-TCoF [31], D3 [60], DT-CNN [32], obtain the

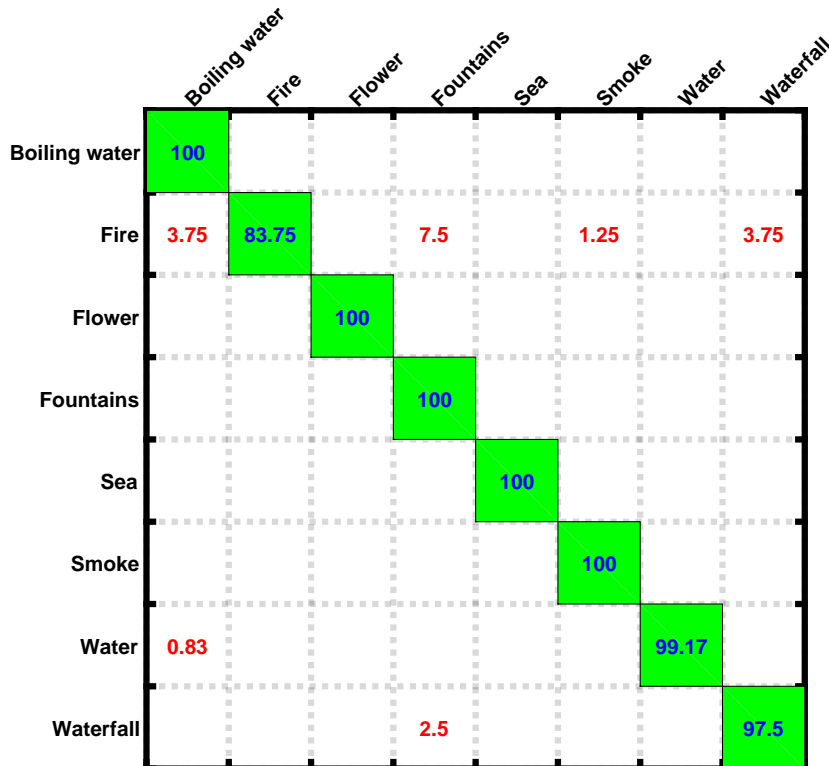


Figure 9: Confusion matrix of $DDTP_{D-M/C}^{L=3}$ on 8-class.

best performances (see Table 9). However, they take a huge cost of computation as well as different parameters for learning DT features on each benchmark dataset.

6.3.3 Recognition on DynTex++ dataset

Recognition results of our proposed framework with different settings are presented in Table 10. It can be observed from the table that $DDTP-B^{L=\{2,3\}}$ descriptors with the setting of $\{(8, 1), (16, 2)\}^{riu2}$ just obtain 91% for length of dense trajectories $L = 2$ and 90.98% for $L = 3$, about 4% lower than those of DDTP descriptors with the same parameters. This has proved the importance of the completed model xLVP for encoding directional characteristics of dense trajectories compared to the basic LVP [47]. In terms of the settings chosen for comparison, the proposed descriptor $DDTP_{D-M/C}^{L=3}$ achieves rate of 95.09%, the competitive performance compared to most of the existing methods (see Table 9). More specifically, only LBP-based approach MEWLSP [45] gains 98.48%, but as mentioned above, it is not better than ours on UCLA (see Table 8) as well as has not been validated on the challenging subsets of DynTex, i.e., *Alpha*, *Beta*,

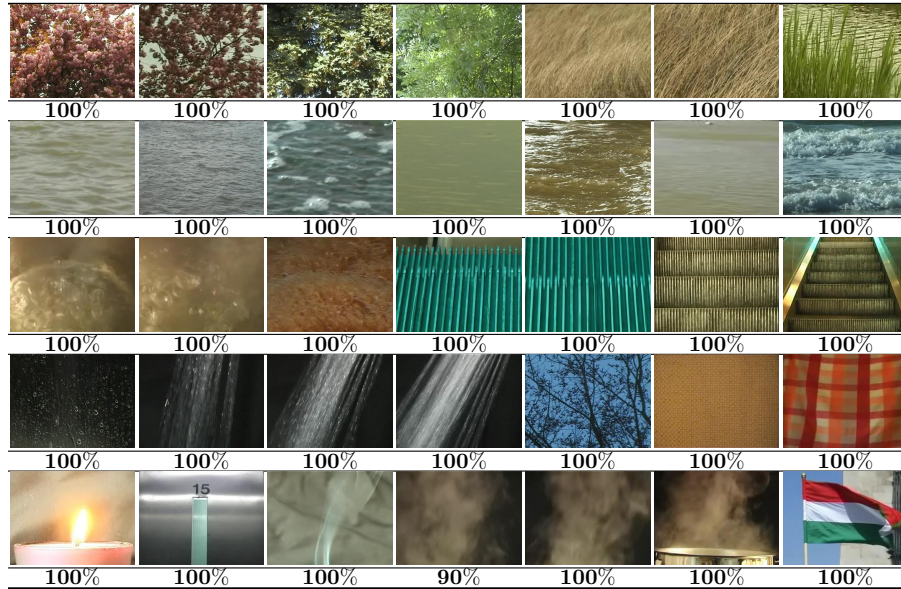


Figure 10: Specific rate of $DDTP_{D.M/C}^{L=3}$ on each class of *DynTex35*.

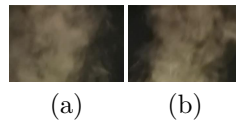


Figure 11: Video (a) is confused with (b) in recognition on *DynTex35*.

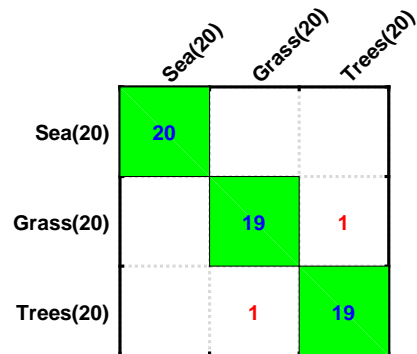


Figure 12: Confusion matrix of $DDTP_{D.M/C}^{L=3}$ on *Alpha*.

and *Gamma*. In the meanwhile, FDT [16] and FD-MAP [16], which are based on directional beams of dense trajectories for DT representation, obtain rates

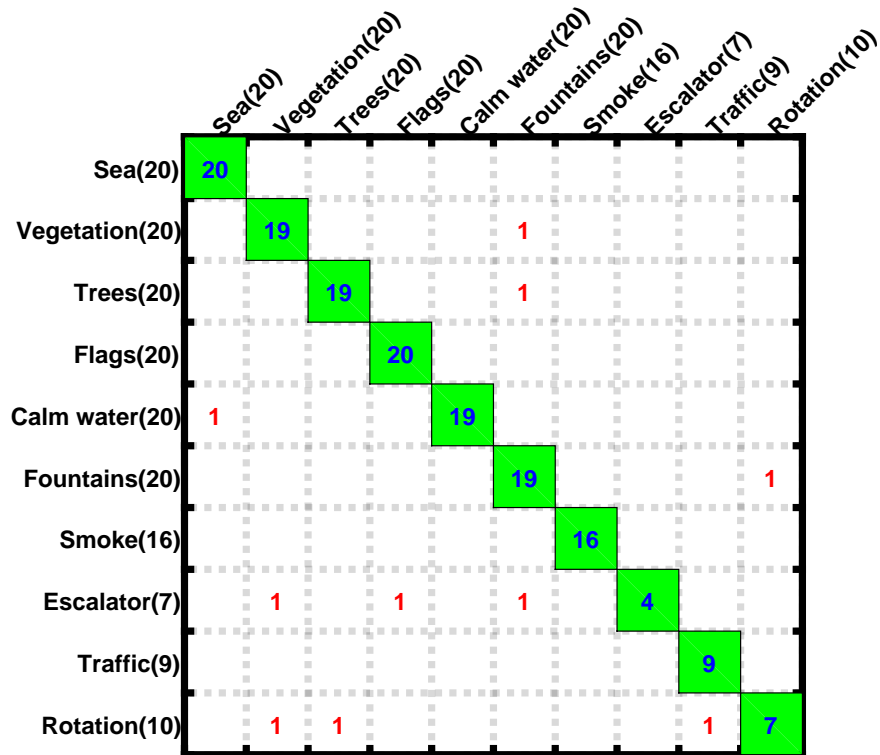


Figure 13: Confusion matrix of $DDTP_{D-M/C}^{L=3}$ on *Beta*.

of 95.31% and 95.69% respectively, just a little higher than ours. Nevertheless, their dimensions are about twofold (see Table 2). DT-CNN [32] obtains 98.18% for the AlexNet framework, 98.58% for the GoogleNet framework (see Table 9). However, it takes a long time to learn features for deep layers along with a huge complicated computation, which may be especially limited in implementations for mobile devices.

6.4 Global discussion

Beside particular evaluations on different benchmark DT datasets in Section 6.3, several general findings can be derived as follows.

- For DT representation, it can be validated from experimental results in Tables 5, 6, and 11 that encoding DTs based on dense trajectories of a video has structured descriptors with more robustness compared to that based on three orthogonal planes of the sequence. That means our xLVP

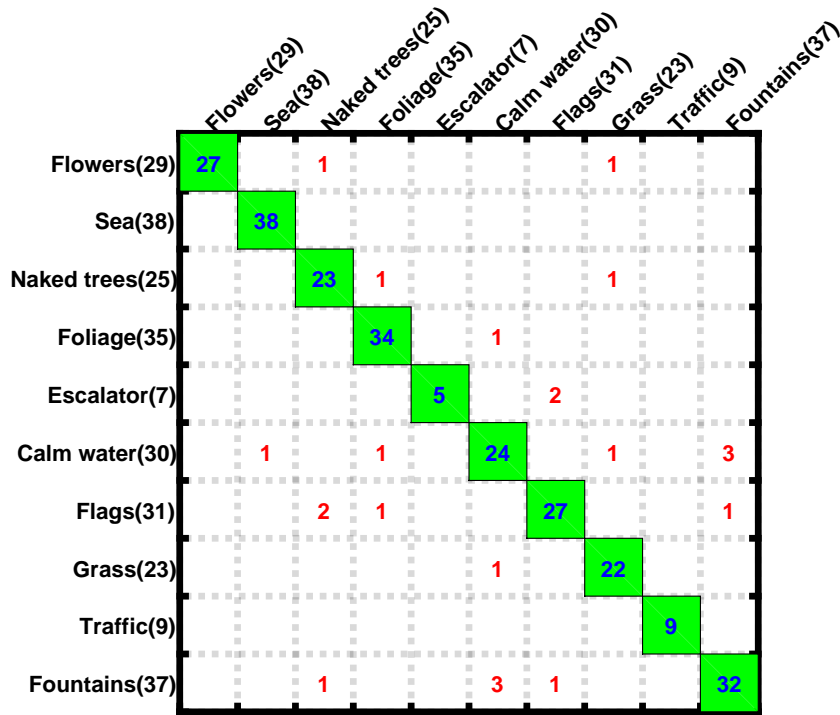


Figure 14: Confusion matrix of $DDTP_{D-M/C}^{L=3}$ on *Gamma*.

operator could be suitable for capturing directional features of dense trajectories instead of for investigating the whole video. It should be noted that in case of focusing on the entire properties of a sequence, xLVP-TOP also significantly outperforms the basic LVP [47] applied on three orthogonal planes (see Table 11).

- xLVP-TOP can be also considered as an alternative solution for encoding DT videos in practice since its performance is reasonable with tiny dimension as well as more outstanding in comparison with the basic LVP-TOP (see Tables 2 and 11).
- Expanding supporting regions for encoding dense trajectories is not a strong recommendation due to lack of concerned spatio-temporal information of directional beams. Indeed, with $\Omega = \{24, 3\}$ and single-scale settings of $\{(24, 3)\}_{L=\{2,3\}}^{riu2}$, the performances of corresponding DDTP descriptors dramatically drop on UCLA (*50-LOO*) and *DynTex35* datasets compared to those of others (see Tables 5, 6, and 12). In the meantime, DDTP descriptors with 3-scale setting of $\{(8, 1), (16, 2), (24, 3)\}_{L=\{2,3\}}^{riu2}$ are

just nearly the same performance as those of 2-scale, i.e., $\{(8, 1), (16, 2)\}_{L=\{2,3\}}^{riu2}$, but in much larger dimension (see Table 2).

- Addressing $u2$ mapping (e.g., $\{(8, 1)\}_{L=\{2,3\}}^{u2}$) for structuring DDTP features points out much larger dimension (see Section 6.1) while its performance is not improved as expected (see Table 12).
- In addition, taking into account motion points in longer dense trajectories enlarges the dimension of proposed descriptors while their performances are not enhanced (see Table 13 for that). This may be due to the short “*living*” time of turbulent motions in a video.

7 Conclusions

In this paper, an efficient framework for DT description has been proposed by incorporating advantages of optical-flow-based and local-feature-based techniques in order to figure out robust descriptors for DT recognition task. Specifically, beams of dense trajectories, extracted from a DT video, are completely investigated in both spatial and temporal changes of motion points. Directional features of them are encoded by xLVP, the crucial extensions of LVP, allowing to capture more forceful local vector relationships. Experiments have validated two following important contributions. First, taking dense trajectories into account DT representation is an interested alternative beside investigating the entire properties of a DT video. Second, based on motion points along their dense trajectories, the completed model xLVP could point out directional patterns with more discriminative power rather than the basic LVP [47] do. In addition, evaluations have also verified that xLVP operator is preferred to encode dense trajectories rather than to consider each voxel on three orthogonal planes of a sequence.

For the further future works, the high-order xLVP can be utilized to contemplate the potential properties of larger local vector structures on movement of these motion points. In order to deal with the curse of large dimension, xLVP can be considered in full directions to seize the entire local directional relations. In addition, exploiting filtering techniques e.g., moment models [62, 63], Gaussian-based kernels [64, 65], can mitigate the negative impacts of illumination and noise on encoding dense trajectories.

Acknowledgment

We would like to express our sincere appreciation for the insightful and valuable comments of the editors and reviewers. Those allow us to clarify and improve the presentation of this work.

References

- [1] Saisan, P., Doretto, G., Wu, Y.N., Soatto, S. ‘Dynamic texture recognition’. In: CVPR. (, 2001. pp. 58–63
- [2] Barmpoutis, P., Dimitropoulos, K., Grammalidis, N. ‘Smoke detection using spatio-temporal analysis, motion modeling and dynamic texture recognition’. In: EUSIPCO. (, 2014. pp. 1078–1082
- [3] Nguyen, T.P., Manzanera, A., Garrigues, M., Vu, N.S.: ‘Spatial motion patterns: Action models from semi-dense trajectories’, *IJPRAI*, 2014, **28**, (7)
- [4] Wang, H., Kläser, A., Schmid, C., Liu, C.: ‘Dense trajectories and motion boundary descriptors for action recognition’, *IJCV*, 2013, **103**, (1), pp. 60–79
- [5] Garrigues, M., Manzanera, A., Bernard, T.M.: ‘Video extruder: a semi-dense point tracker for extracting beams of trajectories in real time’, *J Real-Time IP*, 2016, **11**, (4), pp. 785–798
- [6] Elafi, I., Jedra, M., Zahid, N.: ‘Tracking objects with co-occurrence matrix and particle filter in infrared video sequences’, *IET Computer Vision*, 2018, **12**, (5), pp. 634–639
- [7] Yin, X., Liu, G.: ‘Effective appearance model update strategy in object tracking’, *IET Computer Vision*, 2019, **13**, (6), pp. 531–541
- [8] Rivera, A.R., Chae, O.: ‘Spatiotemporal directional number transitional graph for dynamic texture recognition’, *IEEE Trans PAMI*, 2015, **37**, (10), pp. 2146–2152
- [9] Zhao, G., Pietikäinen, M.: ‘Dynamic texture recognition using local binary patterns with an application to facial expressions’, *IEEE Trans PAMI*, 2007, **29**, (6), pp. 915–928
- [10] Phan, H., Vu, N., Nguyen, V., Quoy, M.: ‘Action recognition based on motion of oriented magnitude patterns and feature selection’, *IET Computer Vision*, 2018, **12**, (5), pp. 735–743
- [11] Ding, W., Liu, K., Chen, H., Tang, F.: ‘Human action recognition using similarity degree between postures and spectral learning’, *IET Computer Vision*, 2018, **12**, (1), pp. 110–117
- [12] Peh, C.H., Cheong, L.F.: ‘Synergizing spatial and temporal texture’, *IEEE Trans IP*, 2002, **11**, (10), pp. 1179–1191
- [13] Péteri, R., Chetverikov, D. ‘Dynamic texture recognition using normal flow and texture regularity’. In: Marques, J.S., de la Blanca, N.P., Pina, P., editors. IbPRIA. vol. 3523 of *LNCS*. (, 2005. pp. 223–230

- [14] Fazekas, S., Chetverikov, D. ‘Dynamic texture recognition using optical flow features and temporal periodicity’. In: *CBMI.* (, 2007. pp. 25–32
- [15] Fazekas, S., Chetverikov, D.: ‘Analysis and performance evaluation of optical flow features for dynamic texture recognition’, *Sig Proc: Image Comm*, 2007, **22**, (7-8), pp. 680–691
- [16] Nguyen, T.T., Nguyen, T.P., Bouchara, F., Nguyen, X.S. ‘Directional beams of dense trajectories for dynamic texture recognition’. In: Blanc.Talon, J., Helbert, D., Philips, W., Popescu, D., Scheunders, P., editors. *ACIVS.* (, 2018. pp. 74–86
- [17] Chan, A.B., Vasconcelos, N.: ‘Modeling, clustering, and segmenting video with mixtures of dynamic textures’, *IEEE Trans PAMI*, 2008, **30**, (5), pp. 909–926
- [18] B..Chan, A.B., Vasconcelos, N. ‘Classifying video with kernel dynamic textures’. In: *CVPR.* (, 2007. pp. 1–6
- [19] Mumtaz, A., Coviello, E., Lanckriet, G.R.G., Chan, A.B.: ‘Clustering dynamic textures with the hierarchical EM algorithm for modeling video’, *IEEE Trans PAMI*, 2013, **35**, (7), pp. 1606–1621
- [20] Ravichandran, A., Chaudhry, R., Vidal, R. ‘View-invariant dynamic texture recognition using a bag of dynamical systems’. In: *CVPR.* (, 2009. pp. 1651–1657
- [21] Wang, Y., Hu, S.: ‘Chaotic features for dynamic textures recognition’, *Soft Computing*, 2016, **20**, (5), pp. 1977–1989
- [22] Qiao, Y., Xing, Z.: ‘Dynamic texture classification using multivariate hidden markov model’, *IEICE Transactions*, 2018, **101-A**, (1), pp. 302–305
- [23] Arashloo, S.R., Kittler, J.: ‘Dynamic texture recognition using multiscale binarized statistical image features’, *IEEE Trans Multimedia*, 2014, **16**, (8), pp. 2099–2109
- [24] Péteri, R., Fazekas, S., Huiskes, M.J.: ‘Dyntex: A comprehensive database of dynamic textures’, *Pattern Recognition Letters*, 2010, **31**, (12), pp. 1627–1632
- [25] Ghanem, B., Ahuja, N. ‘Maximum margin distance learning for dynamic texture recognition’. In: Daniilidis, K., Maragos, P., Paragios, N., editors. *ECCV.* vol. 6312 of *LNCS.* (, 2010. pp. 223–236
- [26] Xu, Y., Quan, Y., Zhang, Z., Ling, H., Ji, H.: ‘Classifying dynamic textures via spatiotemporal fractal analysis’, *Pattern Recognition*, 2015, **48**, (10), pp. 3239–3248
- [27] Xu, Y., Huang, S.B., Ji, H., Fermüller, C.: ‘Scale-space texture description on sift-like textons’, *CVIU*, 2012, **116**, (9), pp. 999–1013

- [28] Ji, H., Yang, X., Ling, H., Xu, Y.: ‘Wavelet domain multifractal analysis for static and dynamic texture classification’, *IEEE Trans IP*, 2013, **22**, (1), pp. 286–299
- [29] Quan, Y., Sun, Y., Xu, Y.: ‘Spatiotemporal lacunarity spectrum for dynamic texture classification’, *CVIU*, 2017, **165**, pp. 85–96
- [30] Baktashmotlagh, M., Harandi, M.T., A., C..Lovell, B.C., Salzmann, M.: ‘Discriminative non-linear stationary subspace analysis for video classification’, *IEEE Trans PAMI*, 2014, **36**, (12), pp. 2353–2366
- [31] Qi, X., Li, C.G., Zhao, G., Hong, X., Pietikainen, M.: ‘Dynamic texture and scene classification by transferring deep image features’, *Neurocomputing*, 2016, **171**, pp. 1230 – 1241
- [32] Andrearczyk, V., Whelan, P.F.: ‘Convolutional neural network on three orthogonal planes for dynamic texture classification’, *Pattern Recognition*, 2018, **76**, pp. 36 – 49
- [33] Arashloo, S.R., Amirani, M.C., Noroozi, A.: ‘Dynamic texture representation using a deep multi-scale convolutional network’, *JVCIR*, 2017, **43**, pp. 89 – 97
- [34] Quan, Y., Huang, Y., Ji, H. ‘Dynamic texture recognition via orthogonal tensor dictionary learning’. In: ICCV. (, 2015. pp. 73–81
- [35] Quan, Y., Bao, C., Ji, H. ‘Equiangular kernel dictionary learning with applications to dynamic texture analysis’. In: CVPR. (, 2016. pp. 308–316
- [36] Ojala, T., Pietikäinen, M., Mäenpää, T.: ‘Multiresolution gray-scale and rotation invariant texture classification with local binary patterns’, *IEEE Trans PAMI*, 2002, **24**, (7), pp. 971–987
- [37] Ren, J., Jiang, X., Yuan, J., Wang, G.: ‘Optimizing LBP structure for visual recognition using binary quadratic programming’, *IEEE Signal Processing Letters*, 2014, **21**, (11), pp. 1346–1350
- [38] Guo, Z., Zhang, L., Zhang, D.: ‘A completed modeling of local binary pattern operator for texture classification’, *IEEE Trans IP*, 2010, **19**, (6), pp. 1657–1663
- [39] Nguyen, T.T., Nguyen, T.P., Bouchara, F. ‘Completed local structure patterns on three orthogonal planes for dynamic texture recognition’. In: IPTA. (, 2017. pp. 1–6
- [40] Nguyen, T.T., Nguyen, T.P., Bouchara, F.: ‘Completed statistical adaptive patterns on three orthogonal planes for recognition of dynamic textures and scenes’, *J Electronic Imaging*, 2018, **27**, (05), pp. 053044

- [41] Zhao, Y., Huang, D.S., Jia, W.: ‘Completed Local Binary Count for Rotation Invariant Texture Classification’, *IEEE Trans IP*, 2012, **21**, (10), pp. 4492–4497
- [42] Zhao, X., Lin, Y., Heikkilä, J.: ‘Dynamic texture recognition using volume local binary count patterns with an application to 2d face spoofing detection’, *IEEE Trans Multimedia*, 2018, **20**, (3), pp. 552–566
- [43] Tiwari, D., Tyagi, V.: ‘A novel scheme based on local binary pattern for dynamic texture recognition’, *CVIU*, 2016, **150**, pp. 58–65
- [44] Tiwari, D., Tyagi, V.: ‘Improved weber’s law based local binary pattern for dynamic texture recognition’, *Multimedia Tools Appl*, 2017, **76**, (5), pp. 6623–6640
- [45] Tiwari, D., Tyagi, V.: ‘Dynamic texture recognition using multiresolution edge-weighted local structure pattern’, *Computers & Electrical Engineering*, 2017, **62**, pp. 485–498
- [46] Tiwari, D., Tyagi, V.: ‘Dynamic texture recognition based on completed volume local binary pattern’, *MSSP*, 2016, **27**, (2), pp. 563–575
- [47] Fan, K., Hung, T.: ‘A novel local pattern descriptor - local vector pattern in high-order derivative space for face recognition’, *IEEE Trans IP*, 2014, **23**, (7), pp. 2877–2891
- [48] Fathi, A., Naghsh.Nilchi, A.R.: ‘Noise Tolerant Local Binary Pattern Operator for Efficient Texture Analysis’, *Pattern Recognition Letters*, 2012, **33**, (9), pp. 1093–1100
- [49] Nguyen, T.P., Manzanera, A., Kropatsch, W.G., N’Guyen, X.S.: ‘Topological attribute patterns for texture recognition’, *Pattern Recognition Letters*, 2016, **80**, pp. 91–97
- [50] Naik, J.B., Srinivasarao, C., Kande, G.B.: ‘Local vector pattern with global index angles for a content-based image retrieval system’, *JASIST*, 2017, **68**, (12), pp. 2755–2770
- [51] Nguyen, X.S., Mouaddib, A.I., Nguyen, T.P., Jeanpierre, L.: ‘Action recognition in depth videos using hierarchical gaussian descriptor’, *Multimedia Tools and Applications*, 2018,
- [52] Farnebäck, G. ‘Two-frame motion estimation based on polynomial expansion’. In: SCIA. (, 2003. pp. 363–370
- [53] Mukherjee, S., Singh, K.K.: ‘Human action and event recognition using a novel descriptor based on improved dense trajectories’, *Multimedia Tools Appl*, 2018, **77**, (11), pp. 13661–13678
- [54] Chen, L., Shen, J., Wang, W., Ni, B.: ‘Video object segmentation via dense trajectories’, *IEEE Trans Multimedia*, 2015, **17**, (12), pp. 2225–2234

- [55] Mäenpää, T., Pietikäinen, M. ‘Multi-scale binary patterns for texture analysis’. In: SCIA. (, 2003. pp. 885–892
- [56] Fan, R., Chang, K., Hsieh, C., Wang, X., Lin, C.: ‘LIBLINEAR: A library for large linear classification’, *JMLR*, 2008, **9**, pp. 1871–1874
- [57] Ribas, L.C., Gonçalves, W.N., Bruno, O.M.: ‘Dynamic texture analysis with diffusion in networks’, *Digital Signal Processing*, 2019, **92**, pp. 109–126
- [58] de Mesquita.Sá.Junior, J.J., Ribas, L.C., Bruno, O.M.: ‘Randomized neural network based signature for dynamic texture classification’, *Expert Syst Appl*, 2019, **135**, pp. 194–200
- [59] Ren, J., Jiang, X., Yuan, J. ‘Dynamic texture recognition using enhanced LBP features’. In: ICASSP. (, 2013. pp. 2400–2404
- [60] Hong, S., Ryu, J., Im, W., Yang, H.S.: ‘D3: recognizing dynamic scenes with deep dual descriptor based on key frames and key segments’, *Neurocomputing*, 2018, **273**, pp. 611–621
- [61] Dubois, S., Péteri, R., Ménard, M.: ‘Characterization and recognition of dynamic textures based on the 2d+t curvelet transform’, *Signal, Image and Video Processing*, 2015, **9**, (4), pp. 819–830
- [62] Nguyen, T.P., Vu, N.S., Manzanera, A.: ‘Statistical binary patterns for rotational invariant texture classification’, *Neurocomputing*, 2016, **173**, pp. 1565–1577
- [63] Nguyen, T.T., Nguyen, T.P., Bouchara, F., Nguyen, X.S.: ‘Momental directional patterns for dynamic texture recognition’, *CVIU*, 2020,
- [64] Nguyen, T.T., Nguyen, T.P., Bouchara, F., Vu, N. ‘Volumes of blurred-invariant gaussians for dynamic texture classification’. In: Vento, M., Percannella, G., editors. CAIP. (, 2019. pp. 155–167
- [65] Nguyen, T.T., Nguyen, T.P., Bouchara, F. ‘Smooth-invariant gaussian features for dynamic texture recognition’. In: ICIP. (, 2019. pp. 4400–4404

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

Method	Dimensions	$P = 8$	$P = 16$	$P = 24$
LBP-TOP ^{u2} [9]	$3(P(P - 1) + 3)$	177	729	1665
VLBP [9]	2^{3P+2}	-	-	-
CVLBP [46]	$3 \times 2^{3P+2}$	-	-	-
HLBP [43]	6×2^P	1536	-	-
CLSP-TOP ^{riu2} [39]	$6(P + 2)^2$	600	1944	4,056
WLBPC [44]	6×2^P	1536	-	-
MEWLSP [45]	6×2^P	1536	-	-
CVLBC [42]	$2(3P + 3)^2$	1458	5202	11125
CSAP-TOP ^{riu2} [40]	$12(P + 2)^2$	1200	3888	8112
FDT ^{u2} [16]	$216P((P - 1) + 3)$	12744	-	-
FD-MAP ^{u2} _{L=2} [16]	$216P((P - 1) + 3) + 16$	12760	-	-
DDTP ^{riu2} _{D-M}	$8(P + 7)(P + 2)$	1200	3312	6448
DDTP ^{riu2} _{D-M-C}	$8(P + 7)(P + 3)$	1320	3496	6696
DDTP ^{riu2} _{D-M/C}	$12(P + 7)(P + 2)$	1800	4968	9672
DDTP-B ^{riu2}	$4(P + 7)(P + 2)$	600	1656	3224
xLVP-TOP ^{riu2} _{D-M}	$24(P + 2)$	240	432	624
xLVP-TOP ^{riu2} _{D-M-C}	$24(P + 3)$	264	456	648
xLVP-TOP ^{riu2} _{D-M/C}	$36(P + 2)$	360	648	936
LVP-TOP ^{riu2}	$12(P + 2)$	120	216	312

Note: P is the concerned neighbors. DDTP, and DDTP-B encode dense trajectories with the length of $L = 2$. All our descriptors are computed by completed operator xLVP in 4 directions with *riu2* mapping (also the settings for comparison their performance with the existing methods).

Table 3: A summary of main properties of DT datasets and protocols.

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. 50%/50% denotes a protocol of taking randomly 50% samples for training and the remain (50%) for testing.

Table 4: Contributions (%) of DBT and TMP of DDTP_{D_MC} descriptor.

Dataset	UCLA (50-LOO)			DynTex35		
	DBT	TMP	DDTP	DBT	TMP	DDTP
$\{(P, R)\}_{L=2}^{riu2}$						
$\{(8, 1)\}_{L=2}^{riu2}$	99.00	90.50	97.50	98.57	96.57	98.00
$\{(16, 2)\}_{L=2}^{riu2}$	99.00	97.50	100	98.86	99.14	99.43
$\{(8, 1), (16, 2)\}_{L=2}^{riu2}$	99.50	97.50	100	98.57	98.29	99.43

Table 5: Results (%) on UCLA exploiting DDTP and DDTP-B descriptors.

Classifier	Scheme	50-LOO				50-4fold				9-class				8-class			
		D _M	D _{M,C}	D _{M/C}	~B	D _M	D _{M,C}	D _{M/C}	~B	D _M	D _{M,C}	D _{M/C}	~B	D _M	D _{M,C}	D _{M/C}	~B
SVM	$\{(8, 1)\}_{L=2}^{riu2}$	97.00	97.50	99.00	98.50	94.00	96.00	99.00	98.00	98.60	98.10	98.10	97.90	96.20	96.85	97.28	94.24
	$\{(16, 2)\}_{L=2}^{riu2}$	99.50	100	99.50	95.00	100	100	99.50	94.50	97.40	96.60	97.90	95.80	96.09	95.76	96.43	95.33
	$\{(8, 1), (16, 2)\}_{L=2}^{riu2}$	100	100	99.00	99.50	100	100	100	99.00	98.35	98.25	98.50	97.85	97.28	96.96	97.50	97.61
	$\{(8, 1)\}_{L=3}^{riu2}$	94.00	94.00	99.00	98.50	95.50	95.50	99.00	98.50	98.10	98.55	98.30	97.45	96.52	97.17	95.33	95.22
	$\{(16, 2)\}_{L=3}^{riu2}$	100	100	99.50	96.50	100	100	99.50	98.50	97.50	97.60	96.65	95.90	97.07	98.15	96.74	93.15
1-NN	$\{(8, 1), (16, 2)\}_{L=3}^{riu2}$	100	100	99.00	99.50	100	100	99.50	98.50	98.60	97.95	98.75	96.15	97.72	98.04	98.04	96.30
	$\{(8, 1)\}_{L=2}^{riu2}$	98.50	98.50	98.50	98.00	99.00	99.00	98.50	98.00	96.20	96.75	95.15	96.90	94.67	93.70	97.07	95.54
	$\{(16, 2)\}_{L=2}^{riu2}$	99.00	99.00	98.50	99.00	99.00	99.00	98.50	99.00	93.55	97.45	95.15	94.80	94.13	96.20	95.87	96.30
	$\{(8, 1), (16, 2)\}_{L=2}^{riu2}$	99.00	98.50	98.50	99.00	99.00	99.00	98.50	99.00	96.10	95.40	96.30	96.20	95.54	96.52	93.70	95.22
	$\{(8, 1)\}_{L=3}^{riu2}$	98.00	98.00	98.50	98.00	98.50	98.50	98.50	98.00	96.55	95.75	96.35	96.95	95.76	97.07	95.87	96.74
	$\{(16, 2)\}_{L=3}^{riu2}$	99.00	99.00	98.50	99.00	99.00	99.00	98.50	99.00	96.00	96.45	95.60	96.20	94.89	96.52	93.37	94.02
	$\{(8, 1), (16, 2)\}_{L=3}^{riu2}$	99.00	99.00	98.50	99.00	99.00	99.00	98.50	99.00	96.30	95.15	93.85	95.75	96.30	96.73	96.09	97.39

Note: 50-LOO and 50-4fold mean recognition rates on 50-class scenario using leave-one-out and four cross-fold validation respectively. D_M, D_{M,C}, and D_{M/C} are different instances of DDTP descriptors formed by integrating the corresponding components of completed operator xLNP. ~B means the DDTP-B descriptor.

Table 6: Rates (%) on DynTex using DDTP and DDTP-B descriptors.

Classifier	Scheme	DynTex35				Alpha				Beta				Gamma			
		D_M	$D_{M,C}$	$D_{M/C}$	$\sim B$	D_M	$D_{M,C}$	$D_{M/C}$	$\sim B$	D_M	$D_{M,C}$	$D_{M/C}$	$\sim B$	D_M	$D_{M,C}$	$D_{M/C}$	$\sim B$
SVM	$\{(P,R)\}_L^{riu2}$	98.57	98.29	98.00	98.00	98.33	98.33	98.33	98.33	90.12	91.36	93.21	87.04	88.64	87.88	90.53	88.64
	$\{(16,2)\}_L^{riu2}$	99.43	99.43	99.43	100	96.67	96.67	96.67	93.33	91.36	90.74	91.98	87.65	90.53	91.67	89.77	87.88
	$\{(8,1),(16,2)\}_L^{riu2}$	99.43	98.86	99.14	99.14	96.67	96.67	96.67	98.33	91.36	91.98	92.59	88.27	92.80	91.67	91.29	87.88
	$\{(8,1)\}_{L=3}^{riu2}$	98.00	98.00	98.57	98.29	98.33	98.33	98.33	98.33	89.51	91.36	94.44	88.89	88.26	89.02	90.15	89.77
	$\{(16,2)\}_{L=3}^{riu2}$	99.43	99.43	99.71	100	96.67	96.67	96.67	93.33	91.36	91.98	93.21	88.89	90.53	90.53	89.77	85.98
1-NN	$\{(8,1)\}_L^{riu2}$	96.29	96.57	96.29	96.29	88.33	88.33	91.67	86.67	79.63	79.63	80.25	81.48	76.14	77.65	78.41	74.62
	$\{(16,2)\}_L^{riu2}$	96.86	96.29	96.57	96.29	91.67	91.67	91.67	90.00	82.72	82.72	83.95	79.01	79.92	79.17	82.58	75.38
	$\{(8,1),(16,2)\}_L^{riu2}$	96.29	96.29	96.57	96.00	93.33	91.67	91.67	91.67	83.33	88.33	88.33	80.25	79.54	79.54	81.82	76.52
	$\{(8,1)\}_{L=3}^{riu2}$	96.57	96.57	96.29	96.00	90.00	90.00	91.67	85.00	80.86	80.25	79.63	80.86	76.14	77.65	78.03	74.24
	$\{(16,2)\}_{L=3}^{riu2}$	96.86	96.57	96.86	96.29	91.67	91.67	91.67	90.00	84.57	84.57	83.33	79.63	79.54	79.54	82.95	75.00
	$\{(8,1),(16,2)\}_{L=3}^{riu2}$	96.86	96.86	96.86	96.57	93.33	91.67	91.67	90.67	82.72	83.33	83.33	80.86	79.17	79.54	82.20	75.52

Note: D_M , $D_{M,C}$, and $D_{M/C}$ are different ways of integrating components of xLVP operator to compute the corresponding DDTP descriptors. $\sim B$ means the DDTP-B descriptor.

Table 7: Results (%) on the cropped version of UCLA.

DDTP $_{D_{M/C}}^{L=3}$	50-LOO			50-4fold			9-class			8-class		
$\{(P,R)\}_L^{riu2}$	D_M	$D_{M,C}$	$D_{M/C}$	D_M	$D_{M,C}$	$D_{M/C}$	D_M	$D_{M,C}$	$D_{M/C}$	D_M	$D_{M,C}$	$D_{M/C}$
$\{(8,1)\}_L^{riu2}$	95.50	96.00	96.00	97.00	97.00	97.00	95.00	95.40	96.45	93.37	95.87	94.89
$\{(16,2)\}_L^{riu2}$	93.50	96.00	94.00	97.00	97.50	96.00	92.50	92.80	94.95	92.72	91.41	92.72
$\{(8,1),(16,2)\}_{L=3}^{riu2}$	96.50	96.50	96.00	96.50	97.00	96.50	94.15	95.05	95.75	94.46	94.13	93.80

Note: 50-LOO and 50-4fold mean recognition rates on 50-class scenario using leave-one-out and four cross-fold validation respectively. D_M , $D_{M,C}$, and $D_{M/C}$ are different instances of DDTP $_{D_{M/C}}^{L=3}$ formed by integrating the corresponding components of xLVP.

Table 8: Comparison of recognition rates (%) on UCLA.

Group	Encoding method	50-LOO	50-4fold	9-class	8-class
A	FDT [16]	98.50	99.00	97.70	99.35
	FD-MAP [16]	99.50	99.00	99.35	99.57
	DDTP _{<i>D.M</i>} {(8, 1), (16, 2)} _{<i>L=3</i>} ^{<i>riu2</i>}	100	100	98.60	97.72
	DDTP _{<i>D.M.C</i>} {(8, 1), (16, 2)} _{<i>L=3</i>} ^{<i>riu2</i>}	100	100	97.95	98.04
	DDTP _{<i>D.M/C</i>} {(8, 1), (16, 2)} _{<i>L=3</i>} ^{<i>riu2</i>}	99.00	99.50	98.75	98.04
B	DDTP-B {(8, 1), (16, 2)} _{<i>L=3</i>} ^{<i>riu2</i>}	99.50	98.50	96.15	96.30
	AR-LDS [1]	89.90 ^N	-	-	-
	Chaotic vector [21]	-	-	85.10 ^N	85.00 ^N
C	Diffusion-based model [57]	-	98.50 ^N	97.80 ^N	96.22 ^N
	3D-OTF [27]	-	87.10	97.23	99.50
	DFS [26]	-	100	97.50	99.20
D	STLS [29]	-	99.50	97.40	99.50
	MBSIF-TOP [23]	99.50 ^N	-	-	-
E	DNGP [8]	-	-	99.60	99.40
	VLBP [9]	-	89.50 ^N	96.30 ^N	91.96 ^N
	LBP-TOP [9]	-	94.50 ^N	96.00 ^N	93.67 ^N
	CVLBP [46]	-	93.00 ^N	96.90 ^N	95.65 ^N
	HLBP [43]	95.00 ^N	95.00 ^N	98.35 ^N	97.50 ^N
	CLSP-TOP [39]	99.00 ^N	99.00 ^N	98.60 ^N	97.72 ^N
	MEWLSP [45]	96.50 ^N	96.50 ^N	98.55 ^N	98.04 ^N
	WLBPC [44]	-	96.50 ^N	97.17 ^N	97.61 ^N
	CVLBC [42]	98.50 ^N	99.00 ^N	99.20 ^N	99.02 ^N
CSAP-TOP [40]	99.50	99.50	96.80	95.98	
F	DL-PEGASOS [25]	-	97.50	95.60	-
	PI-LBP+super hist [59]	-	100 ^N	98.20 ^N	-
	Orthogonal Tensor DL [34]	-	99.80	98.20	99.50
	Randomized neural network [58]	-	97.05 ^N	98.54 ^N	97.74 ^N
	PCANet-TOP [33]	99.50*	-	-	-
	DT-CNN-AlexNet [32]	-	99.50*	98.05*	98.48*
DT-CNN-GoogleNet [32]	-	99.50*	98.35*	99.02*	

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

Table 9: Comparison of rates (%) on DynTex and DynTex++.

Group	Encoding method	Dyn35	Alpha	Beta	Gamma	Dyn++
A	FDT [16]	98.86	98.33	93.21	91.67	95.31
	FD-MAP [16]	98.86	98.33	92.59	91.67	95.69
	DDTP _{<i>D-M</i>} {(8, 1), (16, 2)} _{<i>L=3</i>} ^{<i>riu2</i>}	99.43	96.67	91.98	92.42	94.62
	DDTP _{<i>D-M-C</i>} {(8, 1), (16, 2)} _{<i>L=3</i>} ^{<i>riu2</i>}	99.43	96.67	91.98	90.91	94.69
	DDTP _{<i>D-M/C</i>} {(8, 1), (16, 2)} _{<i>L=3</i>} ^{<i>riu2</i>}	99.71	96.67	93.83	91.29	95.09
	DDTP-B {(8, 1), (16, 2)} _{<i>L=3</i>} ^{<i>riu2</i>}	98.86	98.33	88.27	88.60	90.98
B	Diffusion-based model [57]	-	-	-	-	93.80 ^N
C	3D-OTF [27]	96.70	83.61	73.22	72.53	89.17
	DFS [26]	97.16	85.24	76.93	74.82	91.70
	2D+T [61]	-	85.00	67.00	63.00	-
	STLS [29]	98.20	89.40	80.80	79.80	94.50
D	MBSIF-TOP [23]	98.61 ^N	90.00 ^N	90.70 ^N	91.30 ^N	97.12 ^N
	DNGP [8]	-	-	-	-	93.80
E	VLBP [9]	81.14 ^N	-	-	-	94.98 ^N
	LBP-TOP [9]	92.45 ^N	98.33	88.89	84.85 ^N	94.05 ^N
	DDLBP with MJMI [37]	-	-	-	-	95.80
	CVLBP [46]	85.14 ^N	-	-	-	-
	HLBP [43]	98.57 ^N	-	-	-	96.28 ^N
	CLSP-TOP [39]	98.29 ^N	95.00 ^N	91.98 ^N	91.29 ^N	95.50 ^N
	MEWLSP [45]	99.71 ^N	-	-	-	98.48 ^N
	WLBPC [44]	-	-	-	-	95.01 ^N
	CVLBC [42]	98.86 ^N	-	-	-	91.31 ^N
CSAP-TOP [40]	100	96.67	92.59	90.53	-	
F	DL-PEGASOS [25]	-	-	-	-	63.70
	PCA-cLBP/PI/PD-LBP [59]	-	-	-	-	92.40
	Orthogonal Tensor DL [34]	-	87.80	76.70	74.80	94.70
	Equiangular Kernel DL [35]	-	88.80	77.40	75.60	93.40
	Randomized neural network [58]	-	-	-	-	96.51 ^N
	st-TCoF [31]	-	100*	100*	98.11*	-
	PCANet-TOP [33]	-	96.67*	90.74*	89.39*	-
	D3 [60]	-	100*	100*	98.11*	-
	DT-CNN-AlexNet [32]	-	100*	99.38*	99.62*	98.18*
	DT-CNN-GoogleNet [32]	-	100*	100*	99.62*	98.58*

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

Table 10: Rates (%) of DDTP and DDTP-B descriptors on DynTex++.

Classifier	$\{(P, R)\}_L^{riu2}$	D_M	D_{M_C}	$D_{M/C}$	DDTP-B
SVM	$\{(8, 1)\}_{L=2}^{riu2}$	93.85	94.01	94.14	87.10
	$\{(16, 2)\}_{L=2}^{riu2}$	93.53	94.92	94.16	86.65
	$\{(8, 1), (16, 2)\}_{L=2}^{riu2}$	94.75	94.92	95.04	91.00
	$\{(8, 1)\}_{L=3}^{riu2}$	93.28	93.92	94.27	87.69
	$\{(16, 2)\}_{L=3}^{riu2}$	93.32	94.69	93.76	87.28
	$\{(8, 1), (16, 2)\}_{L=3}^{riu2}$	94.62	94.69	95.09	90.98
1-NN	$\{(8, 1)\}_{L=2}^{riu2}$	91.14	91.47	89.63	89.49
	$\{(16, 2)\}_{L=2}^{riu2}$	90.64	90.72	88.45	88.12
	$\{(8, 1), (16, 2)\}_{L=2}^{riu2}$	91.63	91.73	89.47	89.33
	$\{(8, 1)\}_{L=3}^{riu2}$	90.91	91.33	90.08	89.12
	$\{(16, 2)\}_{L=3}^{riu2}$	90.71	90.89	88.12	87.23
	$\{(8, 1), (16, 2)\}_{L=3}^{riu2}$	91.43	91.35	89.24	89.33

Note: D_M , D_{M_C} , and $D_{M/C}$ are different instances of DDTP descriptors formed by integrating the corresponding components of xLVP operator.

Table 11: Performances (%) on the entire video instead of its dense trajectories.

Dataset	UCLA (50-LOO)				DynTex35			
	D_M	D_{M_C}	$D_{M/C}$	LVP-TOP	D_M	D_{M_C}	$D_{M/C}$	LVP-TOP
$\{(P, R)\}_L^{riu2}$								
$\{(8, 1)\}_{L=2}^{riu2}$	98.00	99.00	99.50	94.00	97.71	97.14	94.29	97.71
$\{(16, 2)\}_{L=2}^{riu2}$	97.00	98.50	99.50	95.00	98.86	98.57	97.71	98.86
$\{(8, 1), (16, 2)\}_{L=2}^{riu2}$	96.50	94.00	98.00	97.00	97.71	98.29	97.14	99.14

Note: D_M , D_{M_C} , and $D_{M/C}$ are different instances of xLVP-TOP descriptors subject to the way of integrating complementary components of xLVP operator.

Table 12: Rates (%) of using larger supporting regions and $u2$ mapping.

Dataset	UCLA (50-LOO)				DynTex35			
	D_M	D_{M_C}	$D_{M/C}$	$\sim B$	D_M	D_{M_C}	$D_{M/C}$	$\sim B$
$\{(P, R)\}_L^{riu2/u2}$								
$\{(24, 3)\}_{L=2}^{riu2}$	95.50	97.50	97.00	79.00	98.86	99.14	99.71	96.86
$\{(24, 3)\}_{L=3}^{riu2}$	93.00	97.00	98.50	83.00	99.14	99.43	99.71	96.86
$\{(8, 1), (16, 2), (24, 3)\}_{L=2}^{riu2}$	100	99.50	99.50	97.50	99.14	99.43	99.43	100
$\{(8, 1), (16, 2), (24, 3)\}_{L=3}^{riu2}$	99.50	100	99.50	99.50	99.14	99.14	99.71	99.43
$\{(8, 1)\}_{L=2}^{u2}$	99.50	99.50	99.50	99.00	98.00	97.71	98.00	95.43
$\{(8, 1)\}_{L=3}^{u2}$	99.50	99.50	99.50	99.00	98.29	98.57	98.00	97.14

Note: D_M , D_{M_C} , and $D_{M/C}$ are different instances of DDTP descriptors subject to the way of integrating complementary components of xLVP operator. $\sim B$ means the DDTP-B descriptor.

Table 13: Performances (%) on longer dense trajectories on UCLA (50-LOO).

Dataset	$L = 5$				$L = 7$			
	D_M	D_{M_C}	$D_{M/C}$	$\sim B$	D_M	D_{M_C}	$D_{M/C}$	$\sim B$
$\{(P, R)\}^{riu2}$								
$\{(8, 1)\}^{riu2}$	96.50	95.50	99.00	97.50	95.00	93.50	98.00	96.50
$\{(16, 2)\}^{riu2}$	100	100	99.50	95.00	99.50	100	99.00	96.00
$\{(8, 1), (16, 2)\}^{riu2}$	99.50	99.50	99.50	98.50	99.50	99.50	98.50	98.50

Note: D_M , D_{M_C} , and $D_{M/C}$ are different instances of DDTP descriptors subject to the way of integrating complementary components of xLVP operator. $\sim B$ means the DDTP-B descriptor.