



# Efficient Representations of Dynamic Textures

Thanh Tuan Nguyen

## ► To cite this version:

Thanh Tuan Nguyen. Efficient Representations of Dynamic Textures. Signal and Image Processing. Université de Toulon, 2020. English. NNT : 2020TOUL0018 . tel-03408028

**HAL Id: tel-03408028**

**<https://theses.hal.science/tel-03408028>**

Submitted on 28 Oct 2021

**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.

UNIVERSITÉ DE TOULON  
ECOLE DOCTORALE 548 - MER ET SCIENCES  
UMR 7020 - Laboratoire d'Informatique et Systèmes

# THÈSE

pour obtenir le grade de Docteur en Informatique

Présentée et soutenue publiquement par

**Thanh Tuan NGUYEN**

le 27 novembre 2020

## Représentations Efficaces des Textures Dynamiques

### Membres du jury

Lionel FILLATRE	Professeur I3S, Université de Nice Sophia Antipolis	Rapporteur
Antoine MANZANERA	Professeur U2IS, ENSTA Paris, Institut Polytechnique de Paris	Rapporteur
Florence SÈDES	Professeure IRIT, Université Toulouse III-Paul Sabatier	Président du jury
Ngoc Son VU	Maître de Conférences ETIS, École d'ingénieur ENSEA	Examineur
Frédéric BOUCHARA	Maître de Conférences, HDR LIS, Université de Toulon	Directeur de thèse
Thanh Phuong NGUYEN	Maître de Conférences LIS, Université de Toulon	Co-Encadrant

UNIVERSITY OF TOULON  
DOCTORAL SCHOOL 548 - SEA AND SCIENCES  
UMR 7020 - The Computer Science and System Laboratory

# THESIS

for obtaining Doctor of Philosophy in Computer Science

Presented and publicly defended on 27 November 2020 by

**Thanh Tuan NGUYEN**

## Efficient Representations of Dynamic Textures

### Jury members

Lionel FILLATRE	Professor I3S, Université de Nice Sophia Antipolis	Reviewer
Antoine MANZANERA	Professor U2IS, ENSTA Paris, Institut Polytechnique de Paris	Reviewer
Florence SÈDES	Professor IRIT, Université Toulouse III-Paul Sabatier	President of jury
Ngoc Son VU	Associate Professor ETIS, Ecole d'ingénieur ENSEA	Jury member
Frédéric BOUCHARA	Associate Professor (HDR) LIS, Université de Toulon	Supervisor
Thanh Phuong NGUYEN	Associate Professor LIS, Université de Toulon	Co-Supervisor

---

# AUTHOR'S PUBLICATIONS

## Journal publications

- [J1] Thanh Tuan Nguyen, Thanh Phuong Nguyen, and Frédéric Bouchara. Completed statistical adaptive patterns on three orthogonal planes for recognition of dynamic textures and scenes. *Journal of Electronic Imaging*, 27(05):053044, 2018.
- [J2] Thanh Tuan Nguyen, Thanh Phuong Nguyen, and Frédéric Bouchara. Directional dense-trajectory-based patterns for dynamic texture recognition. *IET Computer Vision*, 14(4):162–176, 2020.
- [J3] Thanh Tuan Nguyen, Thanh Phuong Nguyen, and Frédéric Bouchara. Prominent local representation for dynamic textures based on high-order gaussian-gradients. *IEEE Transactions on Multimedia*, in press:1–1, 2020. (in press).
- [J4] Thanh Tuan Nguyen, Thanh Phuong Nguyen, and Frédéric Bouchara. Rubik gaussian-based patterns for dynamic texture classification. *Pattern Recognition Letters*, 135:180–187, 2020.
- [J5] Thanh Tuan Nguyen, Thanh Phuong Nguyen, Frédéric Bouchara, and Xuan Son Nguyen. Momental directional patterns for dynamic texture recognition. *Computer Vision and Image Understanding*, 194:102882, 2020.

## International conference publications

- [C1] Thanh Tuan Nguyen, Thanh Phuong Nguyen, and Frédéric Bouchara. Completed local structure patterns on three orthogonal planes for dynamic texture recognition. In *International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6, 2017.
- [C2] Thanh Tuan Nguyen, Thanh Phuong Nguyen, and Frédéric Bouchara. Smooth-invariant gaussian features for dynamic texture recognition. In *IEEE International Conference on Image Processing (ICIP)*, pages 4400–4404, 2019.
- [C3] Thanh Tuan Nguyen, Thanh Phuong Nguyen, and Frédéric Bouchara. Dynamic texture representation based on hierarchical local patterns. In *Advanced Concepts for Intelligent Vision Systems (ACIVS)*, pages 277–289, 2020.
- [C4] Thanh Tuan Nguyen, Thanh Phuong Nguyen, Frédéric Bouchara, and Xuan Son Nguyen. Directional beams of dense trajectories for dynamic texture recognition. In *Advanced Concepts for Intelligent Vision Systems (ACIVS)*, pages 74–86, 2018.
- [C5] Thanh Tuan Nguyen, Thanh Phuong Nguyen, Frédéric Bouchara, and Ngoc-Son Vu. Volumes of blurred-invariant gaussians for dynamic texture classification. In *Computer Analysis of Images and Patterns (CAIP)*, pages 155–167, 2019.



## Articles submitted and under review

- [S1] Thanh Tuan Nguyen, Thanh Phuong Nguyen, and Frédéric Bouchara. A novel Difference of Derivative Gaussians Kernel for Understanding Dynamic Textures. *Computer Vision and Image Understanding*, 2020 (in review).
- [S2] Thanh Tuan Nguyen, Thanh Phuong Nguyen, and Frédéric Bouchara. Completed Hierarchical Gaussian-filtered Patterns for Dynamic Texture Classification. *Neurocomputing*, 2020 (in minor revision).
- [S3] Thanh Tuan Nguyen, Thanh Phuong Nguyen, and Frédéric Bouchara. Dynamic Texture Representation based on Oriented Magnitudes of Gaussian Gradients. *Journal of Visual Communication and Image Representation*, 2020 (in minor revision).
- [S4] Thanh Tuan Nguyen, Thanh Phuong Nguyen, and Frédéric Bouchara. Dynamic Texture Description Using Adapted Gaussian-based Invariant Features. *Visual Computer*, 2020 (in review).
- [S5] Thanh Tuan Nguyen, Thanh Phuong Nguyen, and Frédéric Bouchara. Representing dynamic textures based on bipolar Gaussian-gradient features. *Multimedia Tools and Applications*, 2020 (in review).

---

# RÉSUMÉ

La représentation des textures dynamiques (TD), considérée comme une séquence de textures en mouvement, est un défi en analyse des vidéos dans des applications diverses de la vision par ordinateur. Cela est en partie causé par la désorientation des mouvements, les impacts négatifs des problèmes bien connus dans la capture des caractéristiques turbulentes: bruit, changements d'environnement, illumination, transformations de similarité, mise en échelles, etc. Rendre les TDs plus “compréhensibles” peut être une des missions importantes dans la mise en œuvre des différents systèmes de vision: surveillance automatique des scènes de trafic, des foules de personnes, des interactions humaines, détection d'objets et d'événements, suivi des mouvements, modélisation d'arrière-plan, etc. Dans le cadre de cette thèse, nous introduisons des solutions significatives afin de traiter les problèmes ci-dessus. Par conséquent, trois approches principales suivantes sont proposées pour le codage efficace des TDs : *i)* à partir de trajectoires denses extraites d'une vidéo donnée; *ii)* basé sur des réponses robustes extraites par des modèles de moment; *iii)* basé sur des résultats filtrés qui sont calculés par des variantes de noyaux de filtrage gaussien. En parallèle, nous proposons également plusieurs opérateurs discriminants pour capturer les caractéristiques spatio-temporelles des codages de TD ci-dessus.

Pour une représentation TD basée sur des trajectoires denses, nous extrayons d'abord des trajectoires denses à partir d'une vidéo donnée. Les points de mouvement le long des trajectoires sont ensuite codés par notre opérateur xLVP, une extension des modèles vectoriels locaux (LVP) dans un contexte de codage complémentaire, afin de capturer des caractéristiques directionnelles basées sur une trajectoire dense pour la représentation efficace de TD.

Pour la description TD basée sur des modèles de moment, motivée par un modèle d'images de moment, nous proposons un nouveau modèle de volumes de moment basé sur des informations statistiques des régions de support sphériques centrées sur un voxel. Deux de ces modèles sont ensuite pris en compte dans l'analyse vidéo pour mettre en évidence des images/volumes de moment. Afin d'encoder les images basées sur le moment, nous nous adressons à l'opérateur CLSP, une variante des modèles binaires locaux terminés (CLBP). De plus, notre opérateur xLDP, une extension des modèles de dérivés locaux (LDP) dans un contexte de codage complémentaire, est introduit pour capturer les caractéristiques spatio-temporelles basés sur les volumes des moments.

Pour la représentation DT basée sur les filtrages Gaussiens, nous étudierons de nombreux types de filtrages dans l'étape de prétraitement d'une vidéo pour mettre en évidence des caractéristiques robustes. Après cette étape, les sorties sont codées par des variantes de LBP pour construire les descripteurs de TD. Plus concrètement, nous exploitons les noyaux gaussiens et des variantes de gradients gaussiens d'ordre élevé pour le filtrage. En particulier, nous introduisons un nouveau noyau de filtrage (DoDG) en tenant compte de la différence des gradients gaussiens, qui permet de mettre en évidence des composants robustes filtrés par DoDG pour construire des descripteurs efficaces en maintenant une petite dimensionalité. Parallèlement aux filtrages gaussiens, certains nouveaux opérateurs sont introduits pour répondre à différents contextes du codage TD local: CAIP, une adaptation de CLBP pour résoudre le problème proche de zéro causé par des caractéristiques bipolaires; LRP, basé sur un concept de cube carré de voisins locaux; CHILOP, une formulation généralisée de CLBP.

Les résultats de reconnaissance TD ont validé que nos propositions fonctionnent de manière significative par rapport à l'état de l'art. Certaines d'entre elles ont des performances très proches des approches d'apprentissage profond. De plus, nos descripteurs qui ont une dimensionalité très petite par rapport à celle des méthodes d'apprentissage profond sont appréciées pour les applications mobiles. Par conséquent, les résultats de nos recherches ont été soumis/publiés dans 05 articles de conférences internationales (publiés), 10 articles de revues (dont 05 ont été publiés, et le reste a été soit en soumission soit en révision mineure/majeure).

---

# ABSTRACT

Representation of dynamic textures (DTs), well-known as a sequence of moving textures, is a challenge in video analysis for various computer vision applications. It is partly due to disorientation of motions, the negative impacts of the well-known issues on capturing turbulent features: noise, changes of environment, illumination, similarity transformations, etc. Making DTs more “understandable” can be one of important missions for vision implementations: visual surveillance of traffic scenes, crowded people, human interaction, detecting objects and events, tracking motion, background subtraction, etc. To this end, we introduce significant solutions in order to deal with above problems. Accordingly, three streams of those are proposed for encoding DTs: *i)* based on dense trajectories extracted from a given video; *ii)* based on robust responses extracted by moment models; *iii)* based on filtered outcomes which are computed by variants of Gaussian-filtering kernels. In parallel, we also propose several discriminative descriptors to capture spatio-temporal features for above DT encodings.

For DT representation based on dense trajectories, we firstly extract dense trajectories from a given video. Motion points along the paths of dense trajectories are then encoded by our xLVP operator, an important extension of Local Vector Patterns (LVP) in a completed encoding context, in order to capture directional dense-trajectory-based features for DT representation.

For DT description based on moment models, motivated by the moment-image model, we propose a novel model of moment volumes based on statistical information of spherical supporting regions centered at a voxel. Two these models are then taken into account video analysis to point out moment-based images/volumes. In order to encode the moment-based images, we address CLSP operator, a variant of completed local binary patterns (CLBP). In the meanwhile, our xLDP, an important extension of Local Derivative Patterns (LDP) in a completed encoding context, is introduced to capture spatio-temporal features of the moment-volume-based outcomes.

For DT representation based on the Gaussian-based filterings, we will investigate many kinds of filterings as pre-processing analysis of a video to point out its filtered outcomes. After that, these outputs are encoded by discriminative operators to structure DT descriptors correspondingly. More concretely, we exploit the Gaussian-based kernel and variants of high-order Gaussian gradients for the filtering analysis. Particularly, we introduce a novel filtering kernel (DoDG) in consideration of the difference of

Gaussian gradients, which allows to point out robust DoDG-filtered components to construct prominent DoDG-based descriptors in small dimension. In parallel to the Gaussian-based filterings, some novel operators will be introduced to meet different contexts of the local DT encoding: CAIP, an adaptation of CLBP to fix the close-to-zero problem caused by separately bipolar features; LRP, based on a concept of a square cube of local neighbors sampled at a center voxel; CHILOP, a generalized formulation of CLBP to adequately investigate local relationships of hierarchical supporting regions.

Experiments for DT recognition have validated that our proposals significantly perform in comparison with state of the art. Some of which have performance being very close to deep-learning approaches, expected as one of appreciated solutions for mobile applications due to their simplicity in computation and their DT descriptors in a small number of bins. Consequently, the results of our researches have been contributed in 05 international conference papers (published), 10 journal articles (05 of which have been published, and the rest have been in either under review or minor revision).

---

# ACKNOWLEDGMENT

Foremost, I would like to express my deep gratitude to my principal supervisors, Assoc. Prof. Frédéric BOUCHARA and Assoc. Prof. Thanh Phuong NGUYEN, who had always been very considerate during the whole period of my research, attempting to lend a hand in every fields as best as they could. They are in rigorous responsibility and always maintain a high standard of research not only in proposing and discussing significant concepts but also in doing experiments, writing and revising my research articles. In addition, they had found a financial resource from a research project to offer me a 1-year working contract which its benefit provided an extra living cost during my study in Laboratoire d'Informatique et des Systèmes (LIS), Université de Toulon, France.

I am very honored and privileged to have the opportunity to work with the fellow members of the Signal-Image-Modeling (SIIM) team and others in the LIS laboratory. Also, I would like to thank the staffs of École Doctorale 548 and others in Université de Toulon, who considerately helped me in processes of official documents related to my PhD course.

I would like to thank the Ministry of Education and Training, Vietnam, which granted me a 4-year scholarship from the 911 project aimed at producing PhDs for universities and junior colleges in a period of 2010-2020. Also, we would like to send many thanks to those in Faculty of IT, HCMC University of Technology and Education, Ho Chi Minh City, Vietnam, who gave us crucial supports in high-performing computer systems for implementing experiments on the large scale datasets.

Finally, I am very grateful to my faithful friends and my pretty family, who gave me the unconditional supports, encouraging me to overcome difficult situations in the duration of my study in Toulon, France.



---

# CONTENTS

<b>Author's publications</b>	<b>i</b>
<b>Résumé</b>	<b>iii</b>
<b>Abstract</b>	<b>v</b>
<b>Acknowledgment</b>	<b>vii</b>
<b>List of Figures</b>	<b>xvi</b>
<b>List of Tables</b>	<b>xviii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Dynamic textures: definition, challenges, and applications . . . . .	1
1.2 An overview of representing DTs based on dense trajectories . . . . .	2
1.3 An overview of representing DTs based on moment-based features . . . . .	3
1.4 An overview of representing DTs based on Gaussian-filtered features . . . . .	4
1.5 Our main contributions . . . . .	5
1.6 Outline of thesis . . . . .	5
<b>2 Literature review</b>	<b>7</b>
2.1 Introduction . . . . .	8
2.2 Optical-flow-based methods . . . . .	8
2.2.1 A brief of optical-flow concept . . . . .	8
2.2.2 Analyzing DTs based on optical flow . . . . .	8
2.3 Model-based methods . . . . .	9
2.3.1 Linear Dynamical Systems (LDS) . . . . .	9
2.3.2 Modeling DTs based on LDS . . . . .	9
2.4 Geometry-based methods . . . . .	9
2.4.1 A brief of fractal analysis . . . . .	9
2.4.2 DT representation based on fractal analysis . . . . .	10
2.5 Learning-based methods . . . . .	10
2.5.1 Deep-learning-based techniques . . . . .	10
2.5.2 Dictionary-learning-based techniques . . . . .	11
2.6 Filter-based methods . . . . .	11
2.6.1 DT description based on learned filters . . . . .	12
2.6.2 DT description based on non-learned filters . . . . .	12
2.7 Local-feature-based methods . . . . .	13
2.7.1 A brief of LBP . . . . .	13



2.7.2	A completed model of LBP (CLBP) . . . . .	14
2.7.3	Completed local structure patterns (CLSP), a variant of CLBP . . . . .	15
2.7.4	LBP-based variants for textural image description . . . . .	16
2.7.5	LBP-based variants for DT representation . . . . .	16
2.8	Datasets and protocols for evaluations of DT recognition . . . . .	16
2.8.1	UCLA dataset . . . . .	17
2.8.2	DynTex dataset . . . . .	17
2.8.3	DynTex++ dataset . . . . .	18
2.8.4	DTDB dataset . . . . .	18
2.9	Classifiers for evaluating DT representation . . . . .	19

### **3 Proposed variants of LBP-based operators 21**

3.1	Introduction . . . . .	21
3.2	Completed Adaptive Patterns (CAIP) . . . . .	22
3.3	Some extensions of Local Derivative Patterns (xLDP) . . . . .	24
3.3.1	Local Derivative Patterns . . . . .	24
3.3.2	Adaptive directional thresholds . . . . .	25
3.3.3	Completed model of LDP . . . . .	26
3.3.4	Assessing our proposed extensions of LDP . . . . .	27
3.4	Some extensions of local vector patterns (xLVP) . . . . .	27
3.4.1	Local Vector Patterns . . . . .	27
3.4.2	Adaptive directional vector thresholds . . . . .	28
3.4.3	A completed model of LVP . . . . .	28
3.5	Local Rubik-based Patterns (LRP) . . . . .	30
3.5.1	Complemented components . . . . .	30
3.5.2	Construction of LRP patterns . . . . .	31
3.6	Completed Hierarchical Local Patterns (CHILOP) . . . . .	32
3.6.1	Construction of CHILOP . . . . .	33
3.6.2	A particular degeneration of CHILOP into CLBP . . . . .	36
3.6.3	Beneficial properties of CHILOP operator . . . . .	36
3.7	Summary . . . . .	36

### **4 Representation based on dense trajectories 37**

4.1	Introduction . . . . .	37
4.2	Dense trajectories . . . . .	38
4.3	Beneficial properties of dense trajectories . . . . .	38
4.3.1	Directional features of a beam trajectory . . . . .	38
4.3.2	Spatio-temporal features of motion points . . . . .	39
4.4	Directional dense trajectory patterns for DT representation . . . . .	40
4.4.1	Proposed DDTP descriptor . . . . .	40
4.4.2	Computational complexity of DDTP descriptor . . . . .	42
4.5	Experiments and evaluations . . . . .	44
4.5.1	Experimental settings . . . . .	44
4.5.2	Experimental results . . . . .	46
4.5.2.1	Recognition on UCLA dataset . . . . .	47
4.5.2.2	Recognition on DynTex dataset . . . . .	49
4.5.2.3	Recognition on DynTex++ dataset . . . . .	50
4.5.3	Global discussion . . . . .	51
4.6	Summary . . . . .	52

<b>5</b>	<b>Representation based on moment models</b>	<b>55</b>
5.1	Introduction . . . . .	55
5.2	Moment models . . . . .	56
5.2.1	Moment images . . . . .	56
5.2.2	A novel moment volumes . . . . .	56
5.2.3	Advantages of moment volume model . . . . .	57
5.3	DT representation based on moment images . . . . .	58
5.4	DT representation based on moment volumes . . . . .	59
5.4.1	Proposed momental directional descriptor . . . . .	59
5.4.2	Enhancing the performance with max-pooling features . . . . .	61
5.5	Experiments and evaluations . . . . .	61
5.5.1	Experimental settings . . . . .	61
5.5.2	Assessment of effectiveness of moment models . . . . .	62
5.5.3	Experimental results of MDP-based descriptors . . . . .	64
5.5.3.1	Recognition on UCLA dataset . . . . .	65
5.5.3.2	Recognition on DynTex dataset . . . . .	67
5.5.3.3	Recognition on Dyntex++ dataset . . . . .	68
5.5.3.4	Assessing the proposed components: Recognition with MDP-B and LDP-TOP . . . . .	69
5.5.3.5	Assessing impact of max-pooling features: Recognition with EMDP descriptor . . . . .	71
5.5.4	Global discussion . . . . .	72
5.6	Summary . . . . .	73
<b>6</b>	<b>Representation based on variants of Gaussian filterings</b>	<b>75</b>
6.1	Introduction . . . . .	76
6.1.1	Motivation . . . . .	76
6.1.2	A brief of our contributions . . . . .	77
6.2	Gaussian-based filtering kernels . . . . .	77
6.2.1	A conventional Gaussian filtering . . . . .	77
6.2.2	Gradients of a Gaussian filtering kernel . . . . .	78
6.3	A novel kernel based on difference of Gaussian gradients . . . . .	78
6.3.1	Definition of a novel DoDG kernel . . . . .	79
6.3.2	Beneficial properties of DoDG compared to DoG . . . . .	80
6.4	Representation based on completed hierarchical Gaussian features . . . . .	81
6.4.1	Construction of Gaussian-filtered CHIOP descriptor . . . . .	81
6.4.2	Experiments and evaluations . . . . .	83
6.4.2.1	Parameters for experimental implementation . . . . .	83
6.4.2.2	Assessments of CHIOP's performances . . . . .	83
6.5	Representation based on RUBik Blurred-Invariant Gaussian features . . . . .	86
6.5.1	Benefits of Gaussian-based filterings . . . . .	86
6.5.2	Construction of RUBIG descriptor . . . . .	87
6.5.3	Experiments and evaluations . . . . .	88
6.5.3.1	Parameters for experimental implementation . . . . .	88
6.5.3.2	Assessments of RUBIG's performances . . . . .	89
6.6	Representation based on Gaussian-filtered CAIP features . . . . .	89
6.6.1	Completed sets of Gaussian-based filtered outcomes . . . . .	90
6.6.2	Beneficial properties of filtered outcomes $\Omega_{\sigma, \sigma'}^{2D/3D}$ . . . . .	92
6.6.3	DT description based on complementary filtered outcomes $\Omega_{\sigma, \sigma'}^{2D/3D}$ . . . . .	93
6.6.4	Experiments and evaluations . . . . .	95
6.6.4.1	Parameters for experimental implementation . . . . .	95

6.6.4.2	Assessments of DoG-based features compared to those of FoSIG and V-BIG . . . . .	95
6.6.4.3	Assessments of LOGIC <sup>2D/3D</sup> 's performances . . . . .	96
6.7	Representation based on oriented magnitudes of Gaussian gradients . . . . .	98
6.7.1	Oriented magnitudes of Gaussian gradients . . . . .	99
6.7.2	DT representation based on oriented magnitudes . . . . .	102
6.7.3	Experiments and evaluations . . . . .	105
6.7.3.1	Parameters for experimental implementation . . . . .	105
6.7.3.2	Assessments of effectiveness of decomposing models . . . . .	106
6.7.3.3	Assessments of MSIOMF <sub><math>\sigma</math></sub> <sup>k,D<sup>4</sup></sup> and MSVOMF <sub><math>\sigma</math></sub> <sup>k,D<sup>4</sup></sup> . . . . .	108
6.8	Representation based on Gaussian-gradient features . . . . .	110
6.8.1	High-order Gaussian-gradient Filtered Components . . . . .	112
6.8.2	DT Representation Based on $\Omega_{\mathcal{H},\sigma}^{2D/3D}$ Components . . . . .	114
6.8.3	Experiments and evaluations . . . . .	116
6.8.3.1	Parameters for experimental implementation . . . . .	116
6.8.3.2	Assessments of High-order Gaussian-gradient Descriptors . . . . .	117
6.8.3.3	Comprehensive Comparison to Non-Gaussian-gradients . . . . .	122
6.9	Representation based on DoDG-filtered features . . . . .	122
6.9.1	Construction of DoDG-filtered descriptors . . . . .	122
6.9.2	Experiments and evaluations . . . . .	125
6.9.2.1	Parameters for experimental implementation . . . . .	125
6.9.2.2	Assessments of DoDG-based descriptors . . . . .	128
6.9.2.3	Comprehensive comparison to DoG-based descriptors . . . . .	129
6.10	Comprehensive evaluations in comparison with existing methods . . . . .	129
6.10.1	Benefits of Gaussian-based filterings . . . . .	130
6.10.1.1	Robustness to the well-known issues of DT description . . . . .	130
6.10.1.2	Rich and discriminative features of Gaussian-gradient-based filterings . . . . .	132
6.10.2	Complexity of our proposed descriptors . . . . .	132
6.10.3	Comprehensive discussions of DT classification on different datasets . . . . .	133
6.10.3.1	Classification on UCLA . . . . .	133
6.10.3.2	Classification on DynTex . . . . .	135
6.10.3.3	Classification on DynTex++ . . . . .	135
6.10.3.4	Classification on DTDB dataset . . . . .	136
6.11	Global discussions . . . . .	138
6.11.1	Further evaluations for Gaussian-gradient-based descriptors . . . . .	138
6.11.2	Evaluating appropriation of our proposals for real applications . . . . .	139
6.12	Summary . . . . .	139
<b>7</b>	<b>Conclusions and perspectives</b> . . . . .	<b>141</b>
7.1	Conclusions . . . . .	141
7.2	Perspectives . . . . .	142

<b>Bibliography</b>	<b>143</b>
---------------------	------------

---

# LIST OF FIGURES

1.1	Several samples of DT sequences. . . . .	1
1.2	Several instances of non-directional, turbulent motions of DTs in a video as well as samples of the video in changes of illumination, contrast, and noise. . . . .	2
1.3	A general framework of encoding a video based on its dense trajectories. . . . .	3
1.4	A proposed framework of encoding a video based on moment-based models. . . . .	3
1.5	A general framework of encoding a video based on filtering. . . . .	4
2.1	An illustration of the architecture of AlexNet [1]. . . . .	10
2.2	A simple model of local neighbors $\{\mathbf{p}_i\}$ for $\mathbf{q}_c$ . . . . .	13
2.3	Computations of LBP-based patterns for an input image with settings of $(P, R) = (8, 1)$ and mappings $u2$ and $riu2$ . . . . .	13
2.4	An simple instance of computing a LBP pattern with $(P, R) = (8, 1)$ . . . . .	13
2.5	Several samples of UCLA at line (a) and DynTex at line (b). . . . .	17
2.6	Several samples of DTDB, line (a): Appearance, line (b): Dynamics. . . . .	18
3.1	An issue example of encoding bipolar-based images in which the fact that $\text{CLBP}_S$ figures out a same pattern for three different local textures is resolved by our $\text{CAIP}_S$ . . . . .	23
3.2	A comparison of patterns structured by CLBP and CAIP operators with settings of $\{(P, R)\} = \{(8, 1)\}$ and $riu2$ mapping to encode a Gaussian-based positive-bipolar image $\mathcal{I} = \mathcal{I}_{\text{DoG}_{\sigma, \sigma'}}^{\text{pos}}$ computed using $\sigma = 0.7$ and $\sigma' = 2\sqrt{5}\sigma$ . . . . .	24
3.3	Model of the first-order LDP patterns of $\mathbf{q}_c$ ( $\mathcal{I}'_{\alpha}(\mathbf{q}_c)$ ) and $\mathbf{p}_i$ ( $\mathcal{I}'_{\alpha}(\mathbf{p}_i)$ ) pixels in directions $\alpha \in \mathcal{D}$ in which $\mathbf{q}_c$ (in red) is the considered point, $\mathbf{p}_i$ is the $i^{\text{th}}$ neighbor of $\mathbf{q}_c$ , and $\mathbf{p}_j$ is the $j^{\text{th}}$ neighbor of $\mathbf{p}_i$ . . . . .	24
3.4	An example of two different local structures (marked in red color) are encoded by the same LDP pattern in concerned direction $\alpha = 0^\circ$ . . . . .	25
3.5	Two different local structures (a) and (b) in Figure 3.4 are encoded by different LDP-D patterns in direction $\alpha = 0^\circ$ . . . . .	25
3.6	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 $(P, R) = (8, 1)$ . . . . .	29
3.7	Computing parts of our framework. (a): A model of encoding feature for a voxel $\mathbf{q}$ (in red) based on its central symmetry voxel $\mathbf{q}_f$ (in blue) on plane image $f$ . (b): A graphical illustration of LRP construction at voxel $\mathbf{q}$ . (c): A calculation of an integrated histogram $\text{DMC}(\cdot)$ for voxels $\{\mathbf{q} \in f_i\}$ along with their symmetry points in images $f_{i-1}$ and $f_{i+1}$ of plane $XY$ in a video. . . . .	32
3.8	An instance of structuring two $\mathcal{L}_H$ patterns of $\mathbf{p}_{i=3}, \mathbf{p}_{i=4} \in \Omega_k$ based on $\{\mathbf{q}_j\}_{j=1}^{P_{k+1}}$ of $\Omega_{k+1}$ , in which $\Omega_k = (8, 1)$ , and $\Omega_{k+1} = (8, 2)$ are two adjacent LBP-based regions, i.e., $P_k = P_{k+1} = 8$ neighbors sampled by $R_k = 1$ , and $R_{k+1} = 2$ . . . . .	33

3.9	An example of CHILOP encoding. Therein, $\mathcal{L}_H, \mathcal{L}_M$ , and $\mathcal{L}_C$ patterns for $\forall \mathbf{p}_i \in \Omega_k$ (in dark blue) are corresponding to lines of (a), (b), and (c), which are structured by exploiting two hierarchical LBP-based supporting regions $\mathcal{D} = \{\Omega_k = (8, 1), \Omega_{k+1} = (8, 2) \text{ (in orange)}\}$ . The corresponding $\mathcal{L}_\nabla$ histograms, i.e., line (d), are formed by using an integration of $\nabla = \{H_{M/C}\}$ . . . . .	34
4.1	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 $B = \{8, 1\}$ . . . . .	39
4.2	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. . . . .	39
4.3	An effective framework for DT representation based on dense trajectories extracted from a video $\mathcal{V}$ . . . . .	40
4.4	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. . . . .	44
4.5	Confusion of DDTP $_{D-M/C}^{L=3}$ on 9-class. . . . .	49
4.6	Confusion of DDTP $_{D-M/C}^{L=3}$ on 8-class. . . . .	49
4.7	Specific rates of DDTP $_{D-M/C}^{L=3}$ on each class of DynTex35. . . . .	50
4.8	Video (a) is confused with (b) in recognition on DynTex35. . . . .	51
4.9	Confusion of DDTP $_{D-M/C}^{L=3}$ on Alpha. . . . .	51
4.10	Confusion of DDTP $_{D-M/C}^{L=3}$ on Beta. . . . .	51
4.11	DDTP $_{D-M/C}^{L=3}$ 's confusion on Gamma. . . . .	51
5.1	A sample of structuring element with $\{(P,R)\}=\{(4,1),(8,2)\}$ [2]. . . . .	57
5.2	A pattern of volume support $\Omega = \{(6, 1)\}$ which has $P_k = 6$ blue neighbors sampled on a sphere with the center of red point and radius $R_k = 1$ . . . . .	57
5.3	Illustration of completed statistical adaptive patterns on three orthogonal planes. . . . .	59
5.4	Illustration of structuring proposed DT descriptor. . . . .	60
5.5	Illustration of constructing EMDP descriptor. . . . .	61
5.6	A sample of computing moment images with supporting region $B = \{(1, 4), (2, 8)\}$ . . . . .	61
5.7	An example of filtering process using two first-order moment volumes (i.e., $m^1$ and $\mu^2$ ) with a supporting element of 3D sphere $\Omega = \{(6, 1)\}$ . Based on frames $f_{i-1}$ and $f_{i+1}$ of a video, frame $f_i$ is filtered to form two corresponding frames $f_{i,m^1}$ and $f_{i,\mu^2}$ . . . . .	62
5.8	Confusion matrix (%) of MMDP $_{D-M/C}$ on 9-class. . . . .	67
5.9	Confusion matrix (%) of MMDP $_{D-M/C}$ on 8-class. . . . .	67
5.10	Specific recognition of MMDP $_{D-M/C}$ on each class of DynTex35. . . . .	68
5.11	Two mutual confused categories in recognition on DynTex35. . . . .	69
5.12	Confusion matrix for MMDP $_{D-M/C}$ on Alpha. . . . .	69
5.13	Confusion matrix for MMDP $_{D-M/C}$ on Beta. . . . .	69
5.14	Confusion matrix for MMDP $_{D-M/C}$ on Gamma. . . . .	69
5.15	Recognition of MMDP $_{D-M/C}$ on specific categories of DynTex++, which the challenging ones are highlighted in red rates. . . . .	71
6.1	Responses of the 2D Gaussian-based filterings with deviations $\sigma = 0.7, \sigma = 1.0$ , and threshold $\varepsilon = 0.25$ for decomposition of $\mathcal{I}_{\text{DoG}_{0.7,1}}$ . . . . .	78
6.2	An instance of the 1 <sup>st</sup> -order 2D/3D Gaussian-gradient filterings with $\sigma = 0.7$ . Therein, (a) is for filtering a still image $\mathcal{I}$ using a 2D gradient kernel, while (b) is for filtering a video $\mathcal{V}$ using a 3D gradient kernel. . . . .	78
6.3	Profile of 1D DoG kernel (a) using a pre-defined pair of standard deviations $(\sigma, \sigma') = (0.7, 1)$ compared to those of 1D DoDG kernels at the first (b) and second (c) orders. . . . .	79

6.4	Instances of 2D Gaussian-based filterings for an given image $\mathcal{I}$ using a pre-defined pair of standard deviations $(\sigma, \sigma') = (0.7, 1)$ . Therein, (a): a DoG-filtered image of the conventional DoG <sup>2D</sup> filtering, (b) and (c): DoDG-based images of odd and even DoDG <sup>2D</sup> filterings respectively. . . . .	80
6.5	Our proposed framework of encoding CHIOP <sup>G<sub>n</sub><sub>V,D</sub></sup> descriptor. . . . .	82
6.6	Prominent performances of our CHIOP for a raw DT description compared to CLBP's [3]. . . . .	85
6.7	Outstanding performances of our CHIOP descriptors in comparison with that of HIOP [C3], FoSIG [C2], and CLBP [3]. . . . .	87
6.8	An instance of 3D Gaussian-based filters. (a) is an input gray-scale frame of a DT video. (b) and (c) are 3D smoothed frames of (a) using $\sigma_1 = 0.5$ and $\sigma_2 = 4$ respectively. (d) denotes the 3D DoG of (b) and (c). . . . .	88
6.9	Illustration of proposed framework for encoding RUBIG descriptor. . . . .	89
6.10	Our proposed framework for structuring an input video $\mathcal{V}$ based on its Gaussian-based filtered outcomes. Therein, the black arrows denote preprocessings using 2D/3D Gaussian-based filtering kernels while the blue ones imply processes of DT encoding. . .	91
6.11	An instance of two Gaussian-based filterings with $\sigma = 0.7$ and $\sigma' = 2\sqrt{5}\sigma$ : (a) for filtering a still image $\mathcal{I}$ using 2D kernels, (b) for filtering a video $\mathcal{V}$ using 3D kernels. . .	92
6.12	An illustration for encoding a Gaussian-based filtered volume $\mathcal{V}_G$ using CAIP operator. .	94
6.13	The performances of our descriptors LOGIC <sup>2D/3D</sup> , which utilize both the adapted CAIP operator and the typical CLBP [3] for DT encoding, are compared to our prior works FoSIG <sup>2D</sup> [C2] and V-BIG <sup>3D</sup> [C5]. . . . .	97
6.14	A proposed framework for encoding a video in general. Therein, the blue arrows denote progresses of pre-processing, the black one is a progress of encoding features of oriented magnitudes. . . . .	98
6.15	A hard-assignment model for decomposing the magnitudes of two Gaussian-gradient images $\mathcal{I}_{0.5}^{\partial x^1}$ and $\mathcal{I}_{0.5}^{\partial y^1}$ into 4 HIOM images subject to a set of 4 equal ranges of direction $\mathcal{D}^4 = \{[0, \pi/2), [\pi/2, \pi), [\pi, 3\pi/2), [3\pi/2, 2\pi)\}$ . . . . .	101
6.16	An instance of 3D Gaussian-gradient filtering and computing the obtained volumes of magnitude features. . . . .	102
6.17	A flowchart of HIOM model subject to direction ranges $d_i = [(i-1)\lambda, i\lambda)$ in $\mathcal{D}^n$ . Therein, the black arrows are noted for pre-processing while the blue ones are for encoding. .	103
6.18	A flowchart of HVOM model subject to direction ranges $d_i = [(i-1)\lambda, i\lambda)$ in $\mathcal{D}^n$ . Therein, the black arrows are pre-processing steps while the blue ones are for encoding. .	105
6.19	Performances (%) on <i>Beta</i> of descriptors based on the 4 <sup>th</sup> -order 3D Gaussian-gradient magnitudes using both decomposing and non-decomposing models. . . . .	106
6.20	Performances on DynTex++ of high-order MSIOMF <sup>k, D<sup>4</sup></sup> <sub><math>\sigma</math></sub> and MSVOMF <sup>k, D<sup>4</sup></sup> <sub><math>\sigma</math></sub> descriptors (represented by 2D-S <sup>k</sup> and 3D-S <sup>k</sup> respectively) are sharply decreased when the higher level of standard deviation $\sigma$ is used for the gradient filterings. . . . .	110
6.21	A comprehensive comparison in pairs of high-order MSIOMF <sup>k, D<sup>4</sup></sup> <sub><math>\sigma=1.3</math></sub> and MSVOMF <sup>k, D<sup>4</sup></sup> <sub><math>\sigma=1.3</math></sub> descriptors. . . . .	112
6.22	Our proposed framework for structuring an input video $\mathcal{V}$ based on Gaussian-gradient filtered components. Therein, the black arrows denote preprocessing steps using Gaussian-gradient kernels, while the blue ones imply processes of DT encoding. . . . .	113
6.23	Illustration of using a simple operator $\Psi = \text{CLBP}_{\{8,1\}}^{\text{riu}2}$ to structure the first-order Gaussian-gradient filtered volume $\mathcal{V}_{z^k}^\sigma$ , which is extracted by convolving a gradient-kernel $G_{\sigma, z^k}^{3D}(x, y, z)$ on the temporal direction $z$ of a given video $\mathcal{V}$ with $\sigma = 0.7$ and $k = 1$ . . . . .	115
6.24	An instance of filtering a video $\mathcal{V}$ using 4 orders (i.e., $k = \{1, 2, 3, 4\}$ ) of a 3D Gaussian-gradient kernel with $\sigma = 0.7$ . Therein, columns (a), (b), and (c) denote Gaussian-gradient filtered outcomes of an input frame $f_{XY}$ . (d) denotes informative magnitudes of the obtained Gaussian-gradients of $f_{XY}$ . . . . .	117

6.25	A sharply decrease of performances of high-order $\text{HoGF}^{2D/3D}$ and zero-order $\text{ZoGF}^{2D/3D}$ on DynTex++ when increasing $\sigma$ from 0.5 to 2 for the Gaussian-gradient filterings. . . . .	120
6.26	Outperformances of $\text{HoGF}^{2D/3D}$ descriptors using asymmetric gradient features compared to those using symmetric features. . . . .	121
6.27	Performances of our high-order $\text{HoGF}^{2D/3D}$ descriptors in comparison with those of non-Gaussian-gradient descriptors, i.e., $\text{ZoGF}^{2D/3D}$ , FoSIG [C2], and V-BIG [C5]. . . .	122
6.28	Our proposed framework for encoding a video $\mathcal{V}$ based on its DoDG-filtered outcomes computed by the novel DoDG filtering kernels. Therein, the black arrows denote pre-processing steps, while the blue ones are for processes of DT encoding. . . . .	123
6.29	Comparing performances of $\text{DoGF}_{\sigma, \sigma'}^{2D/3D}$ with those of the $1^{st}$ -order $\text{DoGF}_{\sigma, \sigma', 1^{st}}^{2D/3D}$ . . . .	129
6.30	Comparison of performances of several local-feature-based descriptors using the same $\text{CLBP}_{8,1}^{riu2}$ for encoding DoG/DoDG-based outcomes. Therein, it should be noted that rates of V-BIG [C5] and FoSIG [C2] are referred to their original works, where Gaussian-blurred outcomes are also addressed for the DT encoding as complemented features in addition to DoGs. . . . .	130
6.31	Noise-instances obtained by using different levels of $\text{SNR}_{dB}$ on a plane-image in a video of UCLA dataset. . . . .	131
6.32	The impacts of Gaussian noise on performances of several proposed descriptors compared to others. . . . .	132
6.33	Confusion matrix (%) of $\text{DoDGF}_{(0.7,1),\{1^{st},2^{nd}\}}^{3D}$ on <i>9-class</i> . . . . .	135
6.34	Confusion matrix (%) of $\text{DoDGF}_{(0.7,1),\{1^{st},2^{nd}\}}^{3D}$ on <i>8-class</i> . . . . .	135
6.35	Confusion matrix (%) of $\text{DoDGF}_{(0.7,1),\{1^{st},2^{nd}\}}^{3D}$ on <i>Beta</i> . . . . .	136
6.36	Confusion matrix (%) of $\text{DoDGF}_{(0.7,1),\{1^{st},2^{nd}\}}^{3D}$ on <i>Gamma</i> . . . . .	136
6.37	The specific results of DT recognition of $\text{DoDGF}_{(0.7,1),\{1^{st},2^{nd}\}}^{3D}$ on each category of DynTex++. The challenging categories are highlighted in red rates. . . . .	136

---

# LIST OF TABLES

2.1	A brief of main properties of DT datasets. . . . .	19
3.1	A brief of our proposed operators for local DT encoding. . . . .	22
4.1	Comparing processing time of encoding two videos in UCLA. . . . .	44
4.2	A comparison of various dimensions of LBP-based descriptors. . . . .	45
4.3	Results (%) on UCLA exploiting DDTP and DDTP-B descriptors. . . . .	45
4.4	Rates (%) on DynTex using DDTP and DDTP-B descriptors. . . . .	46
4.5	Contributions (%) of DBT and TMP of DDTP <sub>D<sub>M</sub>C</sub> descriptor. . . . .	46
4.6	Comparison of recognition rates (%) on UCLA. . . . .	48
4.7	Results (%) on the cropped version of UCLA. . . . .	48
4.8	Comparison of rates (%) on DynTex and DynTex++. . . . .	50
4.9	Rates (%) of DDTP and DDTP-B descriptors on DynTex++. . . . .	52
4.10	Performances (%) on the entire video instead of its dense trajectories. . . . .	52
4.11	Rates (%) of using larger supporting regions and <i>u2</i> mapping. . . . .	53
4.12	Performances (%) on longer dense trajectories on UCLA (50-LOO). . . . .	53
5.1	Several comparative dimensions of LBP-based descriptors for DT recognition. . . . .	63
5.2	Classification rates (%) on DT and scene datasets using single-scale CSAP-TOP <sup><i>riu2</i></sup> and its multi-scale analysis. . . . .	63
5.3	Classification rates (%) on DT and scene datasets using single-scale CSAP-TOP <sup><i>u2</i></sup> and its multi-scale analysis. . . . .	64
5.4	Recognition (%) on “mean” ( $m^1$ ) and “variance” ( $\mu^2$ ) videos. . . . .	65
5.5	Comparison of recognition rates (%) on benchmark DT datasets . . . . .	66
5.6	Classification rates (%) on UCLA using MDP, MDP-B descriptors and their multi-scale settings with mappings of <i>riu2/u2</i> . . . . .	67
5.7	Rates (%) on DynTex using MDP, MDP-B descriptors and their multi-scale settings with mappings of <i>riu2/u2</i> . . . . .	68
5.8	Recognition (%) on DynTex++ using MDP, MDP-B descriptors and their multi-scale settings with mappings of <i>riu2/u2</i> . . . . .	70
5.9	Classification rates (%) of LDP-TOP descriptor and its multi-scale settings with mappings of <i>riu2/u2</i> on DT datasets without applying the proposed moment volume model. . . . .	71
5.10	Recognition rates (%) of EMDP <sub>D<sub>M</sub>C</sub> descriptor and its multi-scale settings with mapping of <i>riu2</i> on DT datasets. . . . .	72
5.11	Contribution of max-pooling features for the performance (%) of descriptors using settings of <sub>D<sub>M</sub>C</sub> , and $\{(P, R)\} = \{(8, 1), (16, 2), (24, 3)\}$ with <i>riu2</i> mapping. . . . .	72
5.12	Recognition rates (%) of MDP descriptors encoded on filtered videos with supporting elements of $\Omega = \{(14, 1), (14, 2)\}$ . . . . .	73



6.1	A brief of proposed descriptors based on Gaussian-based filterings. . . . .	77
6.2	A comparison of various dimensions of LBP-based descriptors. . . . .	84
6.3	Comparison of performances (%) between CHILOP and CLBP [3] in encoding DT features based on the raw plane images of a video using <i>riu2</i> mapping with two popular kinds of incorporation on local supporting regions $\mathcal{D} = \{(8, 1), (8, 2)\}$ . . . . .	84
6.4	Rates (%) of CHILOP $_{\nabla, \mathcal{D}}^{G_{\mathcal{F}}^{2D}}$ in multi-layer of hierarchical regions using settings of Gaussian filtering $\mathcal{F} = \{0.5, 1\}$ and jointing type $\nabla = \{H/M/C\}$ . . . . .	85
6.5	Classification rates (%) on UCLA of CHILOP $_{\nabla, \mathcal{D}}^{G_{\mathcal{F}}^{2D}}$ and CHILOP $_{\nabla, \mathcal{D}}$ descriptors. . . . .	86
6.6	Rates (%) on DynTex and DynTex++ of CHILOP $_{\nabla, \mathcal{D}}^{G_{\mathcal{F}}^{2D}}$ and CHILOP $_{\nabla, \mathcal{D}}$ descriptors. . . . .	86
6.7	Comparison contributions in rates (%) on DynTex++ between components of descriptors FoSIG [C2], V-BIG [C5] and our RUBIG. . . . .	90
6.8	Classification rates (%) on benchmark datasets. . . . .	90
6.9	Performances of each filtered element in $\Omega_{(\sigma_i, \sigma'_i) \in \mathcal{F}}^{2D/3D}$ compared to those of FoSIG $^{2D}$ [C2] and V-BIG $^{3D}$ [C5] on DynTex++ using the same local supporting regions $\{(P, R)\} = \{(8, 1), (8, 2)\}$ . . . . .	95
6.10	Rates (%) of LOGIC $^{2D/3D}$ using CLBP $_{\{(8,1),(8,2)\}}^{riu2}$ [3] instead of CAIP $_{\{(8,1),(8,2)\}}^{riu2}$ . . . . .	96
6.11	Classification rates (%) on DT benchmark datasets of LOGIC $^{2D/3D}$ descriptors. . . . .	97
6.12	Performances (%) of MSVOMF $_{\sigma}^{k, \mathcal{D}^4}$ based on the modified soft-assignment in comparison with SVOMF $_{\sigma}^{k, \mathcal{D}^8}$ based on the basic soft model. . . . .	107
6.13	Classification rates (%) on the challenging schemes of descriptors based on non-oriented-magnitude and IOM/VOM-based features. . . . .	107
6.14	Classification rates (%) on DT benchmark datasets of MSIOMF $_{\sigma}^{k, \mathcal{D}^4}$ descriptor. . . . .	108
6.15	Classification rates (%) on DT benchmark datasets of MSVOMF $_{\sigma}^{k, \mathcal{D}^4}$ descriptor. . . . .	109
6.16	Classification rates (%) on DT benchmark datasets of MSIOMF $_{\{\sigma\}}^{k, \mathcal{D}^4}$ descriptor. . . . .	109
6.17	Classification rates (%) on DT benchmark datasets of MSVOMF $_{\{\sigma\}}^{k, \mathcal{D}^4}$ descriptor. . . . .	110
6.18	Classification rates (%) on DT benchmark datasets of MSIOMF $_{\{\sigma\}}^{\{k\}, \mathcal{D}^4}$ descriptors. . . . .	111
6.19	Classification rates (%) on DT benchmark datasets of MSVOMF $_{\{\sigma\}}^{\{k\}, \mathcal{D}^4}$ descriptors. . . . .	112
6.20	Classification rates (%) on DT benchmark datasets of HoGF $^{2D}$ descriptor. . . . .	118
6.21	Classification rates (%) on DT benchmark datasets of HoGF $^{3D}$ descriptors. . . . .	119
6.22	Classification rates (%) on DT benchmark datasets of multi-order HoGF $^{2D}$ descriptor. . . . .	119
6.23	Classification rates (%) on DT benchmark datasets of multi-order HoGF $^{3D}$ descriptor. . . . .	120
6.24	Contributions of the first-order components in $\Omega_{\{1^{st}\}, \{0.7\}}^{2D/3D}$ . . . . .	121
6.25	Settings for comparison and real implementations. . . . .	122
6.26	Rates (%) on DT benchmark datasets of ZoGF $^{2D}$ descriptor. . . . .	123
6.27	Rates (%) on DT benchmark datasets of ZoGF $^{3D}$ descriptor. . . . .	123
6.28	Rates (%) of DoDGF $_{\sigma, \sigma', \mathcal{F}}^{2D}$ and DoGF $_{\sigma, \sigma'}^{2D}$ descriptors on benchmark datasets. . . . .	126
6.29	Rates (%) of DoDGF $_{\sigma, \sigma', \mathcal{F}}^{3D}$ and DoGF $_{\sigma, \sigma'}^{3D}$ descriptors on benchmark datasets. . . . .	127
6.30	Comparing contributions of DoG and the 1 <sup>st</sup> -order of DoDG. . . . .	129
6.31	Performances on different Gaussian noise subsets: <i>50-4fold</i> and <i>Gamma</i> . . . . .	131
6.32	Comparison of processing time of encoding a $50 \times 50 \times 50$ video in DynTex++. . . . .	133
6.33	Comparison of DT recognition rates (%) on benchmark DT datasets . . . . .	134
6.34	Comparison of rates (%) on two challenging subsets of the large scale DTDB [4] dataset. . . . .	137
6.35	Performances (%) of DoDGF $_{\{(\sigma, \sigma')\}, \mathcal{F}}^{2D/3D}$ in further scale analysis. . . . .	138
6.36	Performances (%) of HoGF $^{2D/3D}$ in further scale analysis. . . . .	139

---

# CHAPTER 1

---

## INTRODUCTION

### Contents

1.1	Dynamic textures: definition, challenges, and applications . . . . .	1
1.2	An overview of representing DTs based on dense trajectories . . . . .	2
1.3	An overview of representing DTs based on moment-based features . . . . .	3
1.4	An overview of representing DTs based on Gaussian-filtered features . . . . .	4
1.5	Our main contributions . . . . .	5
1.6	Outline of thesis . . . . .	5

### 1.1 Dynamic textures: definition, challenges, and applications

Dynamic textures (DTs) are textures repeated in a temporal domain, such as fountain, candle, plant, sea-wave, waterfall, fire, etc. [5] (see Figure 1.1 for several samples of DT videos). Efforts of analysis to make DTs more “understandable” are crucial for important tasks of recognition, segmentation, synthesis, and indexing for retrieval. Those can be primary keys in a large range of real applications in computer vision, such as visual surveillance of traffic scenes [6–8], crowded people [9–12], human interaction [13–16], detecting objects and events [17–21], tracking motion objects [22, 23], background subtraction [24–28], etc. To this end, it could be addressed solutions to solve two main well-known challenges as

- The principal challenges in DT analysis are caused by the wide range of appearances along with non-directional and turbulent motions of DTs. Figure 1.2(b) shows a sample of turbulent motions of DTs in non-direction.
- The negative impacts of the well-known problems: noise, changes of environmental factors and illumination, scales, etc., have also been noticeable causes of precipitating the discrimination power in DT representation. Figure 1.2(d,e,f) shows instances of changes of illumination and contrast.



Figure 1.1: Several samples of DT sequences.

Being aware of the importance of DT representation in computer vision, many works have been proposed to deal with two above challenges by exploiting the advantages of spatio-temporal features as

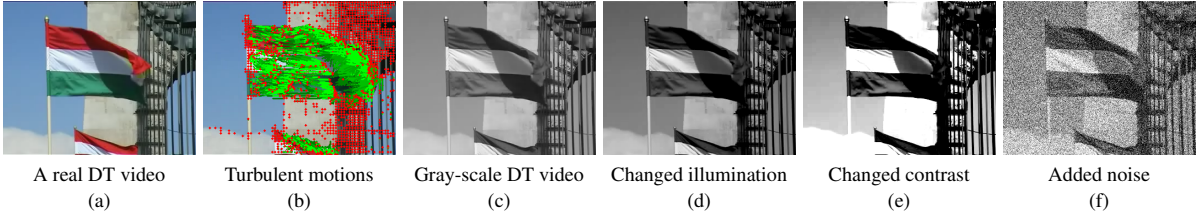


Figure 1.2: Several instances of non-directional, turbulent motions of DTs in a video as well as samples of the video in changes of illumination, contrast, and noise.

well as other properties of DTs. Roughly, these works can be categorized into six major groups:

- *Optical-flow-based*: Using properties of apparent velocities of movements for DT description (see Section 2.2).
- *Model-based*: Using linear dynamical systems to model DT features (see Section 2.3).
- *Geometry-based*: Using fractal methods to analyze DT videos (see Section 2.4).
- *Learning-based*: Using techniques of dictionary learning and convolutional neural networks to learn DT features (see Section 2.5).
- *Filter-based*: Using several filterings to reduce noise before video encoding (see Section 2.6).
- *Local-feature-based*: Using local operators to capture spatio-temporal features (see Section 2.7).

In general, experiments of state of the art in DT recognition have shown that the deep-learning-based ones have often obtained significant performance, but most of them have to take complicated algorithms into account learning a huge number of parameters in deep architectures of neural networks. This is one of crucial barriers in order to bring the deep-learning-based into real applications for mobile devices as well as embedded sensor systems, those which have strictly required tiny resources for their functions. In the meanwhile, the rest approaches have usually addressed in more simplicity but just obtaining at modest levels of performance.

In this thesis, our proposed frameworks could mitigate above shortcomings, i.e., those are in high performance but in low computational complexity, which are expected to be potential for mobile implementations. Indeed, our proposals just utilize simple operators (refer to Chapter 3) to extract spatio-temporal features from two main aspects: based on dense trajectories (refer to Chapter 4) and based on filtered outcomes (refer to Chapters 5 and 6). Hereafter, we will take a general overview of our proposals for a DT video that are based on its dense trajectories and robust filtered outcomes in order to mitigate negative influences of the well-known problems on the DT encoding. After that, we will impress our significant contributions in representing DTs. Also, an outline of the thesis is presented for taking a universal view of the whole organization.

## 1.2 An overview of representing DTs based on dense trajectories

Figure 1.3 graphically illustrates our proposed framework for encoding a video based on its dense trajectories. It could be cleared that our dense-trajectory-based proposal generally includes three main stages to encode a given video  $\mathcal{V}$  as

1. Allocating a set  $\mathcal{T}_L(\mathcal{V})$  of dense trajectories with length of  $L$  which are extracted from  $\mathcal{V}$ .
2. For each dense trajectory  $t_i \in \mathcal{T}$ , a histogram  $h(t_i)$  is computed by using a local operator  $\psi(\cdot)$  to capture features of motion points  $\{\mathbf{p}_{i,j}\}_{j=1}^L$  belonging to the path of  $t_i$ .
3. Structuring a final descriptor for DT representation based on the obtained histograms  $\{h(t_i)\}$ .

It could be seen that we have efficiently exploited the profitable characteristics of both optical-flow-based and local-feature-based methods to boost the discrimination power. Besides, we have also proposed a robust local operator,  $\psi = \text{xLVP}(\cdot)$  (see Section 3.4), in order to encode spatio-temporal features of dense trajectories in more effect. The detail presentation of above stages could be referred to Chapter 4. Experiments in DT recognition have validated the interest of our proposals.

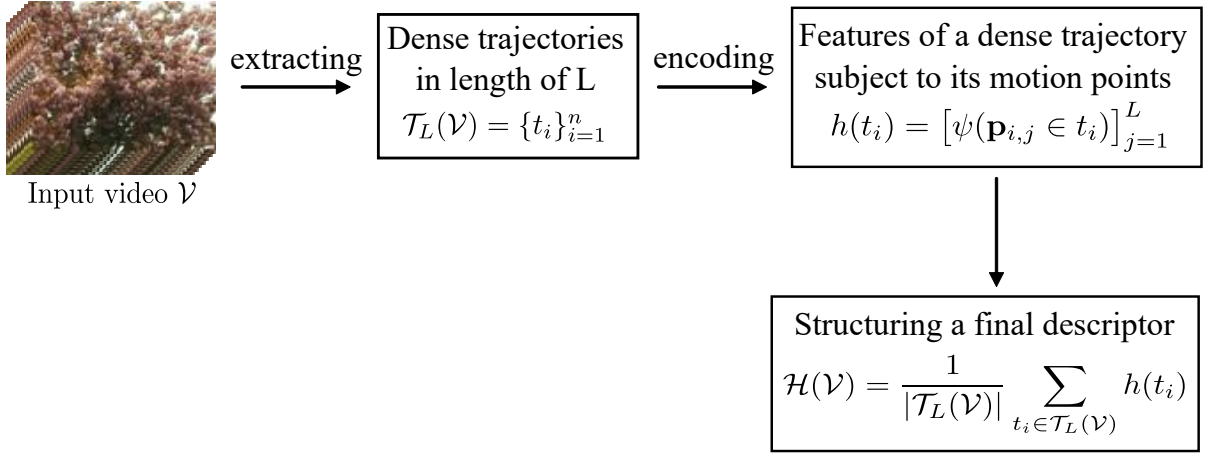


Figure 1.3: A general framework of encoding a video based on its dense trajectories.

### 1.3 An overview of representing DTs based on moment-based features

Motivated by the moment-image model for textural image analysis, introduced by Nguyen *et al.* [2], we take it into account DT representation. Furthermore, we propose a new model of moment volumes which are more adaptive to analyze a video  $\mathcal{V}$ . In general, it can be pointed out that representing DTs based on moment-based models includes three main steps of video analysis as follows.

1. Momental computing models are exploited as a pre-processing in order to compute statistical moments subject to two correspondences of discrete supporting regions, e.g., local circle-based for the moment-image model and local spherical-based for the moment-volume one.
2. To capture local features from the moment-based outcomes  $\{m^r\}$  and  $\{\mu^r\}$  for DT representation, we address CLSP [29] operator (see Section 2.7.3) for encoding the obtained moment-filtered images, while a crucial extension of Local Derivative Patterns [30], named xLDP (see Section 3.3.1), is introduced for encoding the moment-filtered volumes.
3. Structuring a final descriptor based on the obtained histograms  $\{h_m(\cdot), h_\mu(\cdot)\}$ .

The detail presentation of above periods could be referred to Chapter 5. Experiments in DT recognition have validated the better adaptation of our moment-volume model for video analysis in comparison with the moment-image one. Furthermore, our proposed xLDP operator for encoding moment-based volumes has been proved more robustness than its original.

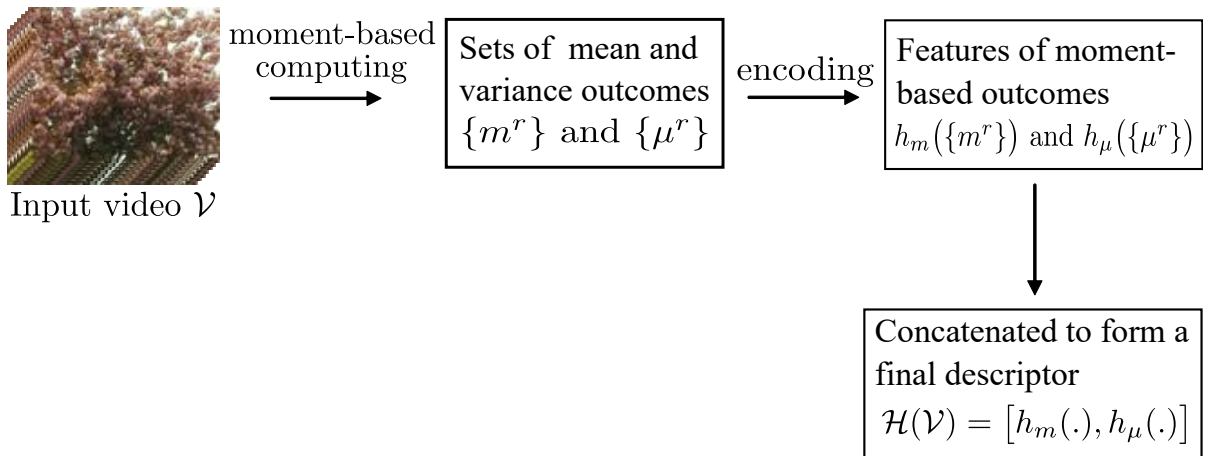


Figure 1.4: A proposed framework of encoding a video based on moment-based models.

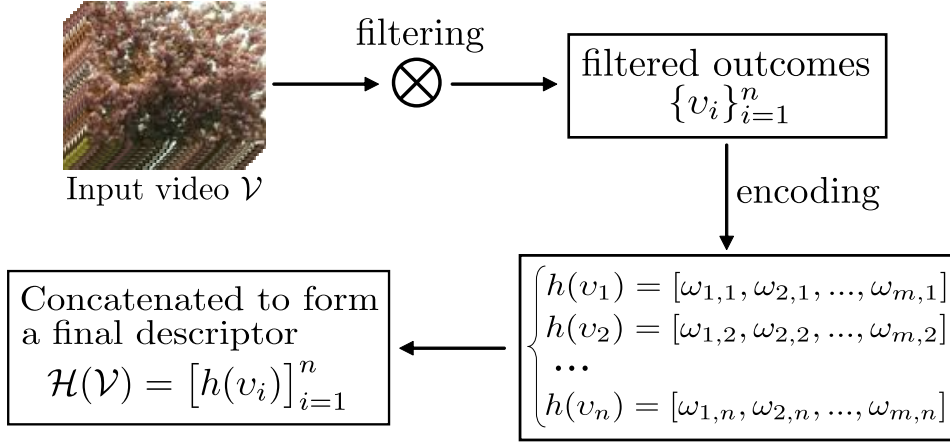


Figure 1.5: A general framework of encoding a video based on filtering.

## 1.4 An overview of representing DTs based on Gaussian-filtered features

Filter-bank approaches, which have been early applied to texture analysis [31], have had promising performance in DT recognition thanks to mitigating the negative influences of noise and other factors on the video encoding. It can be figured out a general diagram for the filtering-based analysis of a given video as shown in Figure 1.5, which includes three main following stages for video representation.

1. Addressing robust filters for filtering a given video  $\mathcal{V}$  to reduce the negative impacts of the problems of DT encoding. This process is applied to the video analysis as a pre-processing in order to point out  $\mathcal{V}$ 's filtered outcomes  $\{v_i\}_{i=1}^n$ .
2. For each filtered outcome  $v_i$ , a histogram  $h(v_i)$  for representing  $v_i$  is formed by using a local operator to capture spatio-temporal features.
3. Structuring a final descriptor for DT representation by simply concatenating the complementary features of all obtained histograms  $\{h(v_i)\}$ .

It can be verified that improvement of the performance majorly depends upon the processes of both filtering and local encoding in the entire framework. Being aware of this substance, we have proposed in this thesis robust filters as well as discriminative local operators as follows.

- For the filtering, we have addressed the Gaussian kernel, its high-order gradients, and other Gaussian-based variants in order to thoroughly investigate various benefits of robust Gaussian-based filtered features for DT representation (refer to Chapter 6).
- For the local operators to capture features from filtered outcomes pointed out by the above filterings, we have inherited and developed some significant ones as follows (refer to Chapter 3 for further expression).
  - CAIP, a crucial adaptation of completed local binary patterns, for efficiently dealing with an issue of close-to-zero pixels caused by the bipolar features of Gaussian-filtered outcomes.
  - LRP, based on local neighbors, which are sampled on a rubik cube centering at a voxel, in order to enrich informative structures for improvement of the discrimination power.
  - CHILOP, exploiting a pairwise of adjacent supporting areas in completed context of analysis to capture hierarchical local patterns for DT representation.

It could be seen that our proposals have taken advantage of both filter-based and local-feature-based properties. Therein, the proposed filters are directly applied to the video filtering. Contrary to several existing methods, ours are non-learned filters which allow to take less the computational cost for the pre-processing, while the obtained responses are robust against the well-known problems in DT representation. On the other hand, our particular operators have assisted to enhance the performance compared to the conventional ones. Indeed, experiments in DT recognition have shown that our proposals have very good performance compared to all non-deep-learning models, while being close to deep-learning approaches. Furthermore, most of them can be potential solutions for mobile applications in practice.

## 1.5 Our main contributions

According to our general concepts for DT representation which are mentioned in Sections 1.2 and 1.4, it could be shortly listed our crucial contributions as follows.

- **Contribution #1:** The novel local operators are proposed to be adopted with different contexts of video encoding: xLVP, xLDP, CAIP, LRP, and CHILOP (refer to Chapter 3). Some of them are applied to representing DTs of raw videos as published in the conference papers [C1, C3]. Some others are used to encode motion points of dense trajectories extracted from a given video (refer to Chapter 4), and filtered responses computed by the filterings (refer to Chapters 5 and 6).
- **Contribution #2:** The thesis introduces a new approach for DT representation by capturing spatio-temporal features of motion points subject to the paths of dense trajectories which are extracted from a given video (refer to Chapter 4). Our contributions have been published in the conference paper [C4] and the journal article [J2].
- **Contribution #3:** Motivated the moment-image model for textural image description, we propose a novel model, named moment volumes, which its responses are computed by local spherical supporting regions instead of the circle-based ones. It have been proved to be more adaptive for video analysis than the moment-image model. In addition, the moment-image model is also investigated on three orthogonal planes of a video to point out mean and variance filtered images for DT representation (refer to Chapter 5). Our contributions have been published in the journal articles [J1, J5].
- **Contribution #4:** Addressing the Gaussian kernel for the video filtering to point out robust Gaussian-filtered outcomes for DT representation. Firstly, we investigate the benefits of single-scale standard deviations, published in the conference papers [C2, C5]. The complementary components of multi-scale Gaussian filterings are then proposed to forcefully capture more rich information. The obtained responses are encoded by our proposed descriptors, LRP (see Section 6.5), CHILOP (see Section 6.4), which the LRP-based results have been published in the journal article [J4] while the CHILOP-based ones have been in minor revision in the journal article [S2]. In another aspect, the difference of Gaussians (DoG) is also conducted in this thesis for DT encoding. Due to the close-to-zero problem caused by decomposing the DoG responses into bipolar components, our CAIP operator is then addressed to deal with that. In addition, the bipolar DoG-filtered features are integrated along with the Gaussian-filtered ones in multi-scale analysis to figure out a discriminative descriptor (see Section 6.6). This result has been under review in the journal article [S4].
- **Contribution #5:** The thesis also takes advantages of partial derivatives of the Gaussian kernel into account the video filterings. We conduct the effect of these filtering kernels in high-order and multi-scale analysis to enrich more robust patterns for DT representation (see Section 6.8). The influential improvement has been published in the journal article [J3]. Especially, we propose a novel filtering kernel based on the difference of high-order Gaussian gradients (DoDG), which allows to point out responses in more robustness (see Section 6.9). Addressing the DoDG responses for local DT encoding allows to structure prominent descriptors in small dimension, which are expected as one of crucial solutions for mobile applications and embedded sensor systems in practice (see Section 6.3). The substantial contribution of DoDG has been under review in the journal article [S1]. In another aspect, the Gaussian gradients are thoroughly discussed in their separately bipolar-based features (under review in the journal article [S5]) and their oriented magnitudes which have been under minor revision in the journal article [S3] (see Section 6.7).

## 1.6 Outline of thesis

The rest of the thesis is organized as follows.

**Chapter 2 Literature review** - In this chapter, we provide a comprehensive overview of state of the art in DT representation, which is categorized into groups of optical-flow-based (see Section 2.2), model-based (see Section 2.3), geometry-based (see Section 2.4), learning-based (see Section 2.5), filter-based

(see Section 2.6), and local-feature-based (see Section 2.7). We also discuss the existing shortcomings of those which should be addressed in further contexts of video analysis for enhancement of describing DTs. Also, benchmark datasets (i.e., UCLA, DynTex, DynTex++, and DTDB) for evaluating the ability of DT descriptors are presented in detail of their properties and protocols as well (see Section 2.8).

**Chapter 3 *Proposed variants of LBP-based operators*** - In this chapter, we propose the novel adaptive descriptors in accordance with the particular contexts of local DT encoding. Therein, CAIP is a crucial adaptation of completed local binary patterns (CLBP) for efficiently dealing with issues of close-to-zero pixels caused by the bipolar features of Gaussian-filtered outcomes (see Section 3.2). xLDP is an important extension of Local Derivative Patterns (LDP) for encoding moment-filtered volumes (see Section 3.3). xLVP is an influential improvement of Local Vector Patterns (LVP) for encoding spatio-temporal features of motion points along the paths of dense trajectories (see Section 3.4). LRP is based on local neighbors sampled on a rubik cube centering at a voxel in order to enrich informative patterns for improvement of the discrimination power (see Section 3.5). CHILOP exploits a pairwise of adjacent supporting areas in completed context of analysis to capture hierarchical local patterns for DT representation (see Section 3.6).

**Chapter 4 *Representation based on dense trajectories*** - In this chapter, we first take a look of the principles of dense trajectories in Section 4.2. We then propose an efficient framework to exploit directional features of beam trajectories and spatio-temporal features of motion points in order to structure a dense-trajectory-based descriptor for a given video (see Sections 4.3, 4.4). Experiments in DT recognition are comprehensively discussed in comparison with state of the art (see Section 4.5).

**Chapter 5 *Representation based on moment models*** - In this chapter, motivated by the moment-image model, we propose a new model of moment volumes, a more appropriate filter for the video filtering (see Section 5.2). After that, we proposed two kinds of DT descriptors: MDP-based utilizes our novel moment volumes and our xLDP operator (see Section 3.3) to encode the obtained filtered volumes in Section 5.3; CSAP-TOP is based on the moment-image model and uses the CLSP-TOP operator (see Section 2.7.3) to encode the obtained filtered images (see Section 5.4). Experiments in DT recognition show that CSAP-TOP has good performance, but not better than the MDP-based descriptors. Because of that, we address the MDP-based descriptors as a crucial solution to thoroughly discuss our proposals in comparison with state of the art (see Section 5.5).

**Chapter 6 *Representation based on variants of Gaussian filterings*** - Motivated by the ascendant of the filtering in denosing for DT representation, we propose in this chapter to exploit several kinds of non-learned filters based on the Gaussian kernel and variants of its partial derivatives. A brief of the Gaussian-based filtering kernels is presented in Section 6.2. A novel filter (named DoDG) is then proposed in consideration of the difference of Gaussian gradients in Section 6.3. After that, robust descriptors and their corresponding performance on recognizing DTs are constructed and experimented as: *i*) representation based on completed hierarchical Gaussian features in Section 6.4, *ii*) representation based on RUBik Blurred-Invariant Gaussian features in Section 6.5, *iii*) representation based on Gaussian-filtered CAIP features in Section 6.6, *iv*) representation based on oriented magnitudes of Gaussian gradients in Section 6.7, *v*) representation based on high-order Gaussian-gradient features in Section 6.8, *vi*) representation based on DoDG-filtered features in Section 6.9. Due to the prominent performance of the DoDG-based descriptors, we address them as one of representative solutions to comprehensively discuss with performance of current approaches in Section 6.10. Finally, a section of global discussion 6.11 will go in some experiments in further contexts as well as will discuss the appreciation of our proposals for mobile applications in practice.

**Chapter 7 *Conclusions and perspectives*** - To come to the end of this thesis, we will restate our proposals, contributions, as well as their advantages and disadvantages. Besides, we also point out several future directions which can be addressed in DT representation for further improvements.

---

# CHAPTER 2

---

## LITERATURE REVIEW

### Contents

<b>2.1</b>	<b>Introduction . . . . .</b>	<b>8</b>
<b>2.2</b>	<b>Optical-flow-based methods . . . . .</b>	<b>8</b>
2.2.1	A brief of optical-flow concept . . . . .	8
2.2.2	Analyzing DTs based on optical flow . . . . .	8
<b>2.3</b>	<b>Model-based methods . . . . .</b>	<b>9</b>
2.3.1	Linear Dynamical Systems (LDS) . . . . .	9
2.3.2	Modeling DTs based on LDS . . . . .	9
<b>2.4</b>	<b>Geometry-based methods . . . . .</b>	<b>9</b>
2.4.1	A brief of fractal analysis . . . . .	9
2.4.2	DT representation based on fractal analysis . . . . .	10
<b>2.5</b>	<b>Learning-based methods . . . . .</b>	<b>10</b>
2.5.1	Deep-learning-based techniques . . . . .	10
2.5.2	Dictionary-learning-based techniques . . . . .	11
<b>2.6</b>	<b>Filter-based methods . . . . .</b>	<b>11</b>
2.6.1	DT description based on learned filters . . . . .	12
2.6.2	DT description based on non-learned filters . . . . .	12
<b>2.7</b>	<b>Local-feature-based methods . . . . .</b>	<b>13</b>
2.7.1	A brief of LBP . . . . .	13
2.7.2	A completed model of LBP (CLBP) . . . . .	14
2.7.3	Completed local structure patterns (CLSP), a variant of CLBP . . . . .	15
2.7.4	LBP-based variants for textural image description . . . . .	16
2.7.5	LBP-based variants for DT representation . . . . .	16
<b>2.8</b>	<b>Datasets and protocols for evaluations of DT recognition . . . . .</b>	<b>16</b>
2.8.1	UCLA dataset . . . . .	17
2.8.2	DynTex dataset . . . . .	17
2.8.3	DynTex++ dataset . . . . .	18
2.8.4	DTDB dataset . . . . .	18
<b>2.9</b>	<b>Classifiers for evaluating DT representation . . . . .</b>	<b>19</b>



## 2.1 Introduction

Due to chaotic and turbulent motions of DTs, efforts of analysis to make them more “understandable” are crucial for important tasks of recognition, segmentation, synthesis, and indexing for retrieval. Those are primary keys in a large range of applications in computer vision, such as visual surveillance of traffic scenes, crowded people [9], human interaction [13–16], detecting objects and events [17, 18], tracking motion objects [22], etc. The major challenges in DT analysis are due to the wide range of appearances and non-directional motions of DTs. Many works for DT representation have been raised to deal with the problems by exploiting the advantages of spatio-temporal features and other properties of DTs. In this chapter, we introduce state of the art of approaches for DT representation. In general, a taxonomy of DT recognition methods can be presented in six main categories: optical-flow-based, model-based, learning-based, filter-based, geometry-based, and local-feature-based. Moreover, benchmark datasets along with several popular classifiers, which are used for evaluating the performance of the state-of-the-art proposals, are also taken a look. Hereafter, we take them one by one in more detail of expression.

## 2.2 Optical-flow-based methods

### 2.2.1 A brief of optical-flow concept

Optical flow is the distribution of apparent velocities of movement of brightness patterns in an image [32, 33]. Let  $\mathbf{p}(x, y) \in \mathcal{I}$  be an image pixel at time  $t$ . In a constraint of the brightness constancy, it can generally write the optical flow as

$$\nabla \mathcal{I}(\mathbf{p}, t) \cdot \vec{\mathbf{v}} + \mathcal{I}_t(\mathbf{p}, t) = 0 \quad (2.1)$$

where  $\mathcal{I}_t(\mathbf{p}, t)$  denotes the temporal derivative of  $\mathcal{I}(\mathbf{p}, t)$ ;  $\nabla \mathcal{I}(\mathbf{p}, t)$  is the derivatives of the image at  $\mathcal{I}(\mathbf{p}, t)$  in the corresponding directions  $\mathcal{I}_x$  and  $\mathcal{I}_y$ , i.e.,  $\nabla \mathcal{I}(\mathbf{p}, t) = (\mathcal{I}_x(\mathbf{p}, t), \mathcal{I}_y(\mathbf{p}, t))^\top$ ;  $\vec{\mathbf{v}} = (u, v)^\top$  denotes the 2D velocity;  $\nabla \mathcal{I}(\mathbf{p}, t) \cdot \vec{\mathbf{v}}$  is the usual dot product. As a result, it could be conducted the aperture problem caused by two unknown components of  $\vec{\mathbf{v}}$  in Equation (2.1). To deal with this problem, the optical-flow-based methods have attempted to introduce further constraints for estimating the corresponding flow subject to specific fields.

### 2.2.2 Analyzing DTs based on optical flow

Taking advantage of the efficient computation and video encoding in natural way, optical-flow-based methods for DT representation have obtained remarkable performance [34–36]. To shape and trace the path of a motion in a sequence, Peh *et al.* [34] aggregated spatio-temporal textures formed by magnitudes and directions of the normal flow which are essential to identify motion types. Péteri *et al.* [35] presented a qualitative approach based on the normal vector field and criteria of videos to describe DT features. In another work, these authors combined the normal flow with filtering regularity to capture the revealing properties of DTs [36]. In the meanwhile, Lu *et al.* [37] utilized the velocity and acceleration properties estimated by a structure tensor to form spatio-temporal multi-resolution histogram. As discussed by Rivera *et al.* [38], due to assumption of brightness constancy and local smoothness, the optical-flow-based methods are usually considered as not to be suitable for stochastic DTs in reality. Moreover, just motion features of DTs are encoded while their textures and appearances have not been regarded.

Addressing those problems, we have proposed to take advantage of profitable characteristics of both optical-flow-based and local-feature-based features by *i)* exploiting Features of Directional Trajectory (FDT) in accordance with Motion Angle Patterns (MAP) for addressing local characteristics and angle information of motion points which are along the paths of dense trajectories of a DT sequence [C4]; *ii)* using our discriminative operator, xLVP (see Section 3.4), proposed for capturing local features from motion points along with their dense trajectories extracted from a given video [J2] (see Chapter 4 for further presentation).

## 2.3 Model-based methods

### 2.3.1 Linear Dynamical Systems (LDS)

Doretto *et al.* [39] laid the foundation for model-based methods with a typical model of Linear Dynamical System (LDS). For a given DT video  $\mathcal{V}$  with a set of  $(T+1)$  frames as  $\mathcal{F}_{\mathcal{V}} = \{\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_T\}$ ,  $\mathbf{y}_t \in \mathbb{R}^m$ . In general, the evolution of a LDS is usually presented as

$$\begin{cases} \mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{v}_t \\ \mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \mathbf{w}_t \end{cases} \quad (2.2)$$

where  $\mathbf{y}_t \in \mathbb{R}^m$  and  $\mathbf{x}_t \in \mathbb{R}^n$  denote the observation and its hidden “state” with initial condition  $\mathbf{x}_0 \in \mathbb{R}^n$ ;  $\mathbf{v}_t \in \mathbb{R}^n$  and  $\mathbf{w}_t \in \mathbb{R}^m$  are independent and identically distributed sequences drawn from known distributions;  $\mathbf{A} \in \mathbb{R}^{n \times n}$  and  $\mathbf{C} \in \mathbb{R}^{m \times n}$  are the system parameters of matrices for estimations.

### 2.3.2 Modeling DTs based on LDS

Inspired by the idea of the typical LDS model, many works have taken it into account DT representation for recognition tasks as well as for other problems in computer vision. Saisan *et al.* [5] agreed the “state” noise  $\mathbf{v}_t$  and the observation noise  $\mathbf{w}_t$  with the distributions of zero-mean Gaussian noise levels for representing DTs. Chan *et al.* [40] utilized *kernel*-PCA (Principal Component Analysis) to model the LDS’s observation matrix  $\mathbf{C}$  as a non-linear function to apprehend characteristics of dynamic features in complex motions, such as chaotic motions (e.g., turbulent water) and camera motions (e.g., panning, zooming, and rotations). Later, to capture the motions of objects in sequences, they presented a model of DT mixtures (DTMs) based on the LDS’s concept. The outputs are then fed into an algorithm of hierarchical expectation-maximization (HEM-DTM) in order to categorize DTMs into  $k$  clusters for DT description [41]. Also based on the LDS model, Wang *et al.* [42] made it in accordance with a bag-of-words (BoW) method to extract chaotic features in videos while Ravichandran *et al.* [43] based on bag-of-systems (BoS) to form the corresponding spatio-temporal patterns. To enhance the speed of performing BoS’s codebooks, Mumtaz *et al.* [44] proposed BoS Tree, in which a bottom-up hierarchy is constructed for indexing the codewords. Recently, Wei *et al.* [45] combined the LDS model with the sparse coding technique to develop a joint dictionary learning framework for modeling DT sequences. In terms of efficiency, the model-based methods have usually achieved modest results on DT recognition because their major drawback is that their encoding mostly concentrates on the spatial-appearance-based characteristics of DTs rather than the dynamic-based ones [5]. Furthermore, efforts taking them into account dynamic features can make the models more complex [43].

## 2.4 Geometry-based methods

### 2.4.1 A brief of fractal analysis

Fractal analysis is built on the concept of fractal dimension which is firstly proposed by Mandelbrot [46] as the measurement of power law existing in many natural phenomena. For a non-empty bounded subset  $E \subset \mathbb{R}^n$ , let  $N_r(E)$  be the smallest number of sets of diameter  $r$  that can cover  $E$ . The fractal dimension of  $E$  is defined as the following [47]:

$$\dim(E) = \lim_{r \rightarrow 0} \frac{\log N_r(E)}{-\log r} \quad (2.3)$$

In practical implementations, it can consider the space as a mesh of boxes of size  $r$ , called the  $r$ -mesh boxes, and count these boxes occupied by the point set [48]. Due to this computation, the above fractal formation is located as the *box-counting* dimension. Further transformations as well as specific instances could be referred to [46–48] for more detail.

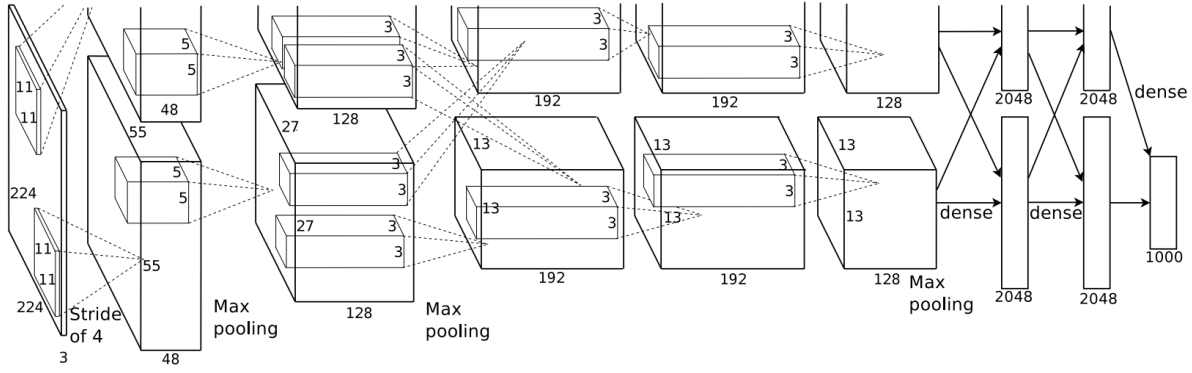


Figure 2.1: An illustration of the architecture of AlexNet [1].

### 2.4.2 DT representation based on fractal analysis

Given a gray-scale DT sequence, Xu *et al.* [49, 50] introduced Dynamic Fractal Spectrum (DFS) based on a fractal analysis of the following integrated measures: pixel intensity, temporal brightness gradient, normal flow, Laplacian, and principal curvature, which are computed subject to a 3D cube centering at a voxel with different values of spatial and temporal radii. An extension of DFS was also proposed in Multi-Fractal Spectrum (MFS) [51] by a combination of capturing stochastic self-similarities and analyzing fractal patterns of DT sequences. However, only spectral information is considered in those works, while spatial domain has been less regarded with. Ji *et al.* [52] addressed this drawback by embedding spatial appearance analysis into MFS in accordance with wavelet coefficients to form Wavelet-based MFS (WMFS) for representing DTs in more effect. In another viewpoint, Quan *et al.* [53] based on the concept of lacunarity, a specialized aspect in fractal geometry for measuring how patterns fill space, in order to propose Spatio-Temporal Lacunarity Spectrum (STLS) descriptor where lacunarity-based features are captured by applying lacunarity analysis to local binary patterns in DT slices. In terms of effectiveness in DT recognition, experiments have shown that the geometry-based methods principally have good performances on simple datasets, e.g., UCLA [5], but not on the more challenging ones, e.g., DynTex [54] and DynTex++ [55]. It may be due to lack of temporal information involved in their encodings.

## 2.5 Learning-based methods

Learning-based methods have been growing into potential approaches as their noteworthy performance in DT recognition. In general, they are usually arranged into two trends: The first one is based on deep learning techniques; the rest is based on dictionary learning. Hereafter, we take a look of these applied to learn features for DT representation.

### 2.5.1 Deep-learning-based techniques

In 1990's, LeCun *et al.* [56, 57] firstly proposed a Convolutional Neural Network (CNN) for hand-written digit recognition. However, until 2012, CNN has been popularized in computer vision when Krizhevsky *et al.* [1] introduced AlexNet which its learning model is very similar to the architecture of LeNet but in deeper, bigger, and featured convolutional layers (see Figure 2.1 for a graphical architecture of AlexNet in general). This popularization is partly thanks to the development of computer hardware architecture with high computational performance. After that, many deep learning models based on CNN's architecture have been proposed to solve different applications in computer vision. The most common ones can be listed such as ZF Net [58], GoogLeNet [59], VGGNet [60], ResNet [61], etc.

For learning DTs, Qi *et al.* [62] adopted AlexNet [1] as a feature extractor to extract mid-level patterns from each frame of a given sequence, and then formed a corresponding DT descriptor by concatenating the first and the second order statistics over the mid-level features, named Transferred ConvNet Features

(TCoF). Andrearczyk *et al.* [63] took AlexNet [1] and GoogLeNet [59] into account video analysis to extract DT features (DT-CNN) from three orthogonal planes of a given video. In the meanwhile, Arashloo *et al.* [64] adopted PCANet [65], a CNN-based model using PCA learned filters for the convolving process, in order to construct a multi-layer convolutional architecture involved with three orthogonal planes of a DT video (PCANet-TOP). Lately, a deep dual descriptor [66] is based on characteristics of “key frames” and “key segments” to learn static and dynamic features. Besides, Hadji *et al.* [4] composed a new challenging large scale dataset, named DTDB (see Section 2.8.4 for its detail expression). They then attempted to implement some deep learning methods for learning DTs on DTDB: Convolutional 3D (C3D) [67], RGB/Flow Stream [68], Marginalized Spatio-temporal Oriented Energy (MSOE) in two learning streams (MSOE-two-Stream) [4].

Although the deep-learning-based approaches have often obtained significant performance for DT recognition, most of them have to take complicated algorithms into account learning a huge number of parameters in deep architectures of neural networks. For instance, as implemented by Hadji *et al.* [4] for DT classification issue on the recent large scale DTDB [4] dataset,  $\sim 80\text{M}$  learned parameters are taken by C3D [67], while  $\sim 88\text{M}$  by Two-Stream [68] and MSOE-two-Stream [4] networks. Besides, DT-CNN [63] is addressed in different sets of parameters to be fed into AlexNet and GoogLeNet frameworks which have  $\sim 61\text{M}$  and  $\sim 6.8\text{M}$  learned parameters respectively for learning DT features on different DT datasets. This is one of crucial barriers in order to bring the deep-learning-based ones into real applications for mobile devices as well as embedded sensor systems, those which have strictly required tiny resources for their functions. In this work, our proposed frameworks could mitigate those shortcomings in low computational complexity, which could be potential for mobile implementations in practice. Indeed, our proposals just utilize simple operators (refer to Chapter 3) to extract spatio-temporal local features for DT representation from two main aspects: based on dense trajectories (refer to Chapter 4) and based on filtered outcomes (refer to Chapters 5 and 6). Experiments have proved that our correspondingly proposed descriptors have small dimension (e.g., HoGF [J3], DoDGF [S1], etc.) while their performance has been close to that of the deep-learning-based approaches.

### 2.5.2 Dictionary-learning-based techniques

Another trend for DT representation is based on dictionary-learning-based techniques using sparse representation to learn DT features. In general, a sparse coding can be briefly presented as follows. Let  $\mathbf{D} \in \mathbb{R}^{n \times K}$  be an over-complete dictionary matrix containing  $K$  atoms  $\{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_K\}$ . It is assumed that a vector  $\mathbf{y} \in \mathbb{R}^n$  can be represented as a sparse linear combination of these atoms: either exactly as  $\mathbf{y} = \mathbf{D}\mathbf{x}$  or approximately as  $\mathbf{y} \approx \mathbf{D}\mathbf{x}$  so that  $\|\mathbf{y} - \mathbf{D}\mathbf{x}\|_p \leq \epsilon$ , where typical norms used for measuring the deviation are the  $l^p$ -norms. Therein,  $\mathbf{x} \in \mathbb{R}^K$  is a coefficient vector of  $\mathbf{y}$ . Motivated by the sparse representation, Quan *et al.* [69] adopted K-SVD [70], an algorithm for training of dictionaries, to model a DT sequence by a set of space-time elements with certain distribution, where local DT features are structured from a dictionary learned via sparse representation from a set of local DT patches of the DT video, known as atoms. However, it is difficult to perform in multi-scale analysis as done in some of the geometry-based approaches [51, 52]. Learning multiple dictionaries with different size of atoms can be possible but it is hard to be efficient in the computing models. On the other side, Quan *et al.* [71] introduced equiangular kernel to learn a dictionary with optimal mutual coherence in computational feasibility. In terms of effectiveness of those in DT representation, experiments have shown that the dictionary-learning-based approaches have performed well in recognizing DTs on simple datasets (e.g., UCLA [5]), but not on the more complex ones (e.g., DynTex [54], DynTex++ [55]). In the meantime, our proposals in simple frameworks can significantly improve the discrimination power of DT descriptors as well as obtain much better rates in DT recognition.

## 2.6 Filter-based methods

As mentioned in Section 1.4, the filtering is one of crucial solutions to reduce noise and other factors which negatively impact on DT representation (see Figure 1.5 for a general view of DT encoding based

on the filtering). In general, the filter-based methods have evinced their efficiency in performance of DT recognition. Experiments have illustrated that the former filter-based approaches have performed well on DT datasets with simple motions (e.g., UCLA [5]), while they either remain several limitations or have not been verified on challenging datasets (e.g., DynTex [54], DynTex++ [55]). Addressing the issue, our proposals in this thesis could thoroughly deal with this negative influence to form local-based descriptors with significant improvement of discrimination power. For taking a look of the filter-based methods, we can shortly arrange the filter-based methods into two categories as follows.

### 2.6.1 DT description based on learned filters

The main perception of this stream is to encode the filtered elements of a given video which are extracted by various learned filters. Arashloo *et al.* [72] exploited the binarized statistical image features (BSIF) [73] to produce learned BSIF filters. To this end, they used a generative model of Independent Component Analysis (ICA) to present a given image patch  $\mathcal{I}$  through a vector  $\mathbf{r}$  of unknown random variables and a feature matrix  $\mathcal{W}$  as follows:  $\mathcal{I} = \mathcal{W}\mathbf{r}$ . The obtained BSIF filters were then applied to the orthogonal plane-images of a given video in order to form spatio-temporal BSIF-TOP descriptor for single-scale analysis of BSIF filters and MBSIF-TOP for the multi-scale one. In another approach, Zhao *et al.* [74] applied the CLBP's concept [3] to encode the filtered responses produced by  $L$  learned filters. Accordingly, let  $\mathbf{W} = [\omega_1, \omega_2, \dots, \omega_L]$  be  $L$  vectorized 3D filters. These 3D filters were learned by following different techniques: Principal component analysis (PCA) in [64, 75], ICA-based in [72, 73], Sparse filtering in [76], K-means clustering in [77]. For a zero-mean vector  $\mathbf{v}_k$  computed by a  $k \times k \times k$  square cube of a video, it could be obtained  $L$  filter responses subject to applying  $\mathbf{W}$  to  $\mathbf{v}_k$  as follows:  $r_k = \mathbf{W}^\top \mathbf{v}_k$ . After that, CLBP-based components could be located for encoding these filtered outputs  $r_k$  in order to structure B3DF descriptors for DT representation.

### 2.6.2 DT description based on non-learned filters

Contrast to the methods based on the learned filters, the main perception of this stream is to encode the filtered elements which are extracted by filterings of non-learning-based filters. It can be effortlessly realized that it can save computational cost due to no learning process related to the filterings. To the best of our knowledge, although there are many efforts using non-learned filters for textural image description (e.g., MRELBP [78], SBP [2], RAMBP [79], etc.), it has been very rarely for representation of DTs until our recent proposals. Specifically, Rivera *et al.* [38] proposed to use a Kirsch compass mask [80] for calculating the spatial directional response of a pre-defined neighborhood in eight different directions. Accordingly, for a given video  $\mathcal{V}$ , each instance of 2D/3D Kirsch mask  $M_k$  is convolved on sub-regions of  $\mathcal{V}$ 's plane-images for the 2D filtering and  $\mathcal{V}$  for the 3D one in order to obtain Kirsch-based responses as follows:  $\mathcal{V}_k = \mathcal{V} * M_k^{2D/3D}$ . These outputs are then adapted to a graph model in order for capturing spatio-temporal features of directional number transitional graph (DNG) for DT description. In this thesis, we will propose to exploit different kinds of potential filterings applied to  $\mathcal{V}$  in order to achieve robust filtered outcomes against the well-known problems for local DT encodings: filtering models of moment images [J1] and volumes [J5] (refer to Chapter 5 for further presentation), filterings based on Gaussian-based kernels [C2, C5, J4, S2, S4] and their derivations [J3, S1, S3, S5] (refer to Chapter 6 for further expression). Experiments in DT recognition issue have validated that our proposed descriptors have very good performance compared to state of the art. Some of them in very simple computation and small dimension have rates being close to those of deep-learning approaches. More significantly, ours could be expected as one of appreciated solutions for mobile applications in practice.

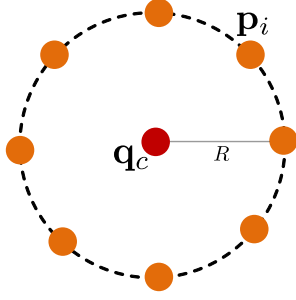


Figure 2.2: A simple model of local neighbors  $\{p_i\}$  for  $q_c$ .

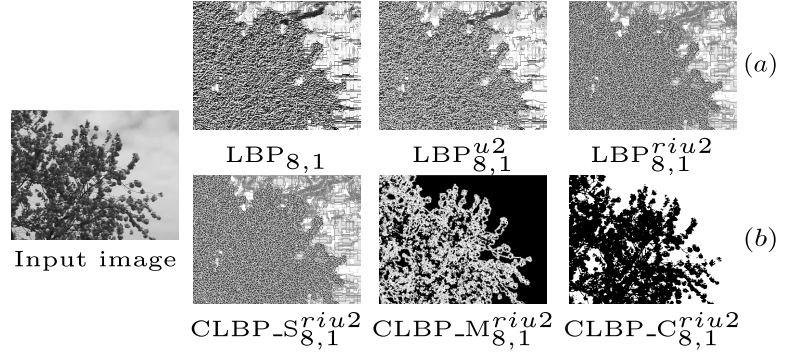


Figure 2.3: Computations of LBP-based patterns for an input image with settings of  $(P, R) = (8, 1)$  and mappings  $u2$  and  $riu2$ .

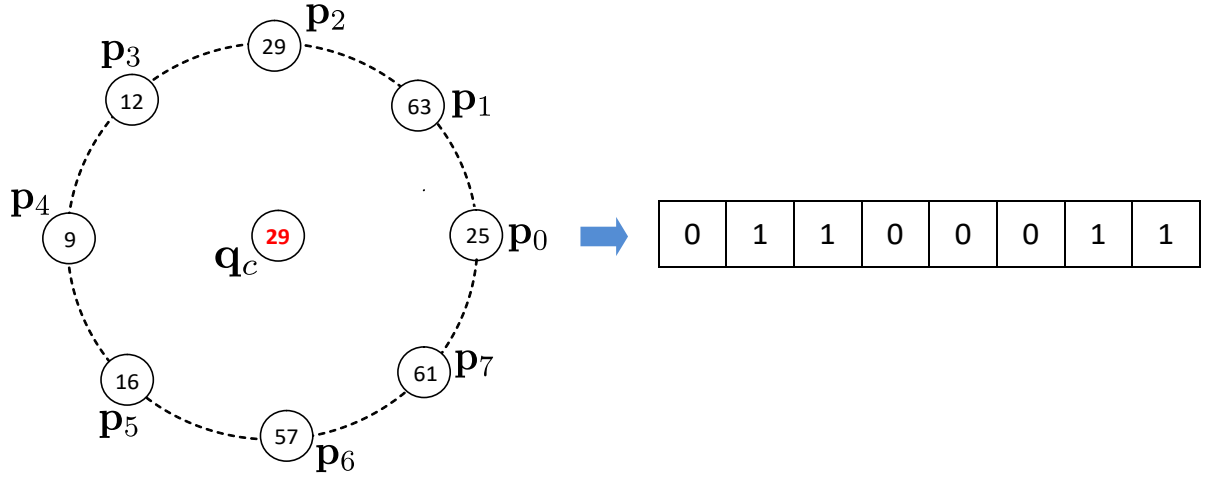


Figure 2.4: An simple instance of computing a LBP pattern with  $(P, R) = (8, 1)$ .

## 2.7 Local-feature-based methods

### 2.7.1 A brief of LBP

An efficient and simple operator LBP is introduced by Ojala *et al.* [81] for encoding a 2D gray-scale image  $\mathcal{I}$ , in which each pixel  $q_c \in \mathcal{I}$  is featured as a string of binary digits by drawing a comparison of the different gray levels between  $q_c$  and its local neighbors  $\{p_i\}$  as follows.

$$\text{LBP}_{P,R}(q_c) = \sum_{i=0}^{P-1} \xi(\mathcal{I}(p_i) - \mathcal{I}(q_c)) \times 2^i \quad (2.4)$$

where  $P$  denotes a number of considered neighbors which can be interpolated on a circle of radius  $R$  and center  $q_c$  as graphically illustrated in Figure 2.2;  $\mathcal{I}(\cdot)$  returns the gray-level of a pixel; and binary thresholding function  $\xi(\cdot)$  is defined as

$$\xi(x) = \begin{cases} 1, & x \geq 0 \\ 0, & \text{otherwise.} \end{cases} \quad (2.5)$$

Figure 2.4 shows an imitation of the particular LBP model in Figure 2.2 for structuring a LBP pattern based on a supporting region  $(P, R) = (8, 1)$ , while Figure 2.3 at line (a) indicates a computation of  $\text{LBP}_{8,1}$  patterns for an input image in reality.

As a result, a LBP code takes  $2^P$  distinct bins to construct a histogram for textural image description. It is impractical for implementation in applications of computer vision due to the curse of large dimen-

sion. Therefore, in real applications, the most popular mappings are usually taken into account to turn it down into a reasonable size as follows.

- In order to reduce the dimension  $2^P$  to  $P(P-1) + 3$  bins, a  $u2$  mapping for uniform patterns ( $\text{LBP}^{u2}$ ) [81] (see Figure 2.3 at line (a)) are derived from the typical LBP codes conditioned by a number of bit-transitions of their binary chains at most 2 as

$$\text{LBP}_{P,R}^{u2}(\mathbf{q}_c) = \{\text{LBP}_{P,R}(\mathbf{q}_c)\} \text{ so that } U(\text{LBP}_{P,R}(\mathbf{q}_c)) \leq 2 \quad (2.6)$$

where  $U$  is a uniformity measure of bit-transitions (1-0 or 0-1) for a LBP pattern and is defined as

$$U(\text{LBP}_{P,R}(\mathbf{q}_c)) = |\xi(\mathcal{I}(\mathbf{p}_{P-1}) - \mathcal{I}(\mathbf{q}_c)) - \xi(\mathcal{I}(\mathbf{p}_0) - \mathcal{I}(\mathbf{q}_c))| + \sum_{i=1}^{P-1} |\xi(\mathcal{I}(\mathbf{p}_i) - \mathcal{I}(\mathbf{q}_c)) - \xi(\mathcal{I}(\mathbf{p}_{i-1}) - \mathcal{I}(\mathbf{q}_c))| \quad (2.7)$$

- An other important mapping to deal with rotation invariant ( $\text{LBP}^{ri}$ ) [81] is stated as

$$\text{LBP}_{P,R}^{ri}(\mathbf{q}_c) = \min_{0 \leq i < P} \{ROR(\text{LBP}_{P,R}(\mathbf{q}_c), i)\} \quad (2.8)$$

where  $ROR(\text{LBP}_{P,R}(\mathbf{q}_c), i)$  calculates the distribution of  $\text{LBP}^{ri}$  by shifting  $i$  times of the  $P$ -bit  $\text{LBP}_{P,R}(\mathbf{q}_c)$  pattern.

- In reality,  $ri$  and  $u2$  mappings are often combined to form patterns of  $riu2$  mapping ( $\text{LBP}^{riu2}$ ) as identified in Equation (2.9). This leads to reduction of dimensional representation from  $2^P$  of the basic LBP to  $P + 2$  distinct values. Figure 2.3 at line (a) shows an instance of  $riu2$  pattern computation.

$$\text{LBP}_{P,R}^{riu2}(\mathbf{q}_c) = \begin{cases} \text{LBP}_{P,R}^{ri}(\mathbf{q}_c) & \text{if } U(\text{LBP}_{P,R}^{ri}(\mathbf{q}_c)) \leq 2 \\ P + 1 & \text{otherwise} \end{cases} \quad (2.9)$$

In addition, inspired by the effectiveness of above mappings, other crucial mappings are suggested to refine these mappings for encoding more textural information. Zhao *et al.* [82] advanced Local Binary Count (LBC), an alternative of  $riu2$  patterns, by considering differences of the higher gray levels between  $P$  neighbors and center pixels correspondingly. On the other hand, Fathi *et al.* [83] extended the basic uniform mapping based on advantages of some non-uniform patterns. Nguyen *et al.* [84] then embedded the underlying mappings and LBC into a general mapping,  $TAP^A$ , for obtaining topological information.

### 2.7.2 A completed model of LBP (CLBP)

For forcefully structuring LBP-based patterns, Guo *et al.* [3] introduced the completed model of LBP (CLBP) including three complementary components for textural image representation as follows:

- First, CLBP\_S is identical to the typical LBP, i.e.,  $\text{CLBP\_S}_{P,R}(\mathbf{q}_c) = \text{LBP}_{P,R}(\mathbf{q}_c)$ .
- Second, CLBP\_M captures magnitude information and is defined as

$$\text{CLBP\_M}_{P,R}(\mathbf{q}_c) = \sum_{i=0}^{P-1} g(|\mathcal{I}(\mathbf{p}_i) - \mathcal{I}(\mathbf{q}_c)|, \tilde{m}_{\mathcal{I}}) \times 2^i \quad (2.10)$$

where  $\tilde{m}_{\mathcal{I}}$  denotes the mean of differences of  $|\mathcal{I}(\mathbf{p}_i) - \mathcal{I}(\mathbf{q}_c)|$  for the whole image  $\mathcal{I}$ , binary function  $g(\cdot)$  is defined as

$$g(x, y) = \begin{cases} 1, & x \geq y \\ 0, & \text{otherwise.} \end{cases} \quad (2.11)$$

- Third, CLBP\_C measures the difference between the gray value of a center pixel  $\mathbf{q}_c$  and the mean of all in image  $\mathcal{I}$ .

$$\text{CLBP\_C}_{P,R}(\mathbf{q}_c) = g(\mathcal{I}(\mathbf{q}_c), \tilde{c}_{\mathcal{I}}) \quad (2.12)$$

where function  $g(\cdot)$  is defined in Equation (2.11),  $\tilde{c}_{\mathcal{I}}$  expresses the average of gray-levels for the whole image  $\mathcal{I}$ .

Figure 2.3 at line (b) shows typical computations of CLBP\_S, CLBP\_M, and CLBP\_C for a given image. In order to enhance the discrimination power, the above complementary components can be integrated together by different ways. Among of them, the description, which is formed by the joint of these with *riu2* mapping (i.e.,  $\text{CLBP}_{S/M/C}^{\text{riu2}}$ ), usually has significant performance compared to the others. It could be referred to comprehensive discussions in [3] for more detail. In our proposed works (e.g., [C5], [C2], [J3], [S3], [S1], and [S5]), CLBP operator is used as a basic solution in order to encode Gaussian-based filtered outcomes for DT representation. For instance, structuring robust descriptors is presented in sections of Chapter 6 such as 6.6, 6.4, 6.7, 6.8, and 6.9. In addition, motivated by CLBP's concept, we have also proposed several significant operators which are defined in Chapter 3.

### 2.7.3 Completed local structure patterns (CLSP), a variant of CLBP

**An adaptive local threshold:** The typical LBP effectively captures the local spatial relations in consideration of differences of gray-levels between a center pixel  $\mathbf{q}_c$  and its neighbors in a given image  $\mathcal{I}$ . However, it also leads to its two well-known restrictions: the sensitivity to noise and near uniform regions because a small change of the center pixel can largely alter the obtained binary pattern. To overcome these issues, Shrivastava *et al.* [29] introduced an adaptive local threshold based on two mean-gray values as follows:

- The first one, named Local Average Difference (LAD), is defined as the mean of local variations of magnitudes around center pixel  $\mathbf{q}_c$  and its  $P$  neighbors  $\{\mathbf{p}_i\}$  as

$$\text{LAD}(\mathbf{q}_c) = \frac{1}{P} \sum_{i=0}^{P-1} |\mathcal{I}(\mathbf{p}_i) - \mathcal{I}(\mathbf{q}_c)| \quad (2.13)$$

where  $\mathcal{I}(\cdot)$  is the gray-scale value of a pixel,  $\mathbf{p}_i$  is the  $i^{\text{th}}$  neighbor of  $\mathbf{q}_c$ ,  $P$  is the number of considered neighbors.

- The second one, called Global Mean Difference (GMD), is calculated by the mean of the absolute differences over the entire image  $\mathcal{I}$  as

$$\text{GMD}(\mathcal{I}) = \frac{1}{P \times N} \sum_{\mathbf{q}_j \in \mathcal{I}} \sum_{i=0}^{P-1} |\mathcal{I}(\mathbf{p}_i) - \mathcal{I}(\mathbf{q}_j)| \quad (2.14)$$

in which  $\mathbf{p}_i$  is the  $i^{\text{th}}$  neighbor of  $\mathbf{q}_j$ ,  $N$  is the total number of pixels in image  $\mathcal{I}$ .

Accordingly, the adaptive local threshold is formed for the center pixel  $\mathbf{q}_c$  as follows.

$$T(\mathbf{q}_c) = \mathcal{I}(\mathbf{q}_c) + \frac{a \times \text{LAD}(\mathbf{q}_c) + b \times \text{GMD}(\mathcal{I})}{a + b} \quad (2.15)$$

where two variables  $a$  and  $b$  are valued at 0 or 1 (i.e.,  $a, b \in \{0, 1\}$ ) to determine which mode of information is exploited. It could be seen that when  $a = b = 0$ , this case is simply identical to LBP.

**A completed model of local structure patterns (CLSP):** Similar to the construction of CLBP for representing textural image  $\mathcal{I}$ , three main complementary components (CLSP\_S, CLSP\_M, and CLSP\_C) are calculated by Shrivastava *et al.* [29] as follows.

- CLSP\_S captures local relationships like CLBP\_S.
- CLSP\_M exploits local variation of magnitudes, which is similar to CLBP\_M but the traditional thresholding is replaced by adaptive thresholding  $T$  as

$$\text{CLSP\_M}_{P,R}(\mathbf{q}_c) = \sum_{i=1}^{P-1} \xi(\mathcal{I}(\mathbf{p}_i) - T(\mathbf{q}_c)) \times 2^i \quad (2.16)$$

where  $\mathbf{p}_i$  is the  $i^{\text{th}}$  neighbor of center pixel  $\mathbf{q}_c$ ,  $P$  is the number of considered neighbors sampled by radius  $R$ ,  $\mathcal{I}(\cdot)$  return the gray-scale value of a pixel,  $\xi(\cdot)$  is defined in Equation (2.5),  $T(\cdot)$  is an



adaptative thresholding calculated by assigning specific values to variables  $a, b$  in Equation (2.15) in order to determine which kind of the relationships are allocated.

- The rest component, CLSP\_C, takes into account information of center pixel  $\mathbf{q}_c$  as

$$\text{CLSP\_C}(\mathbf{q}_c) = \xi(\text{LAVG}_{P,R}(\mathbf{q}_c) - \text{GAVG}(\mathcal{I})) \quad (2.17)$$

where LAVG is the average gray level of  $P$  neighbors around pixel  $\mathbf{q}_c$ , GAVG is the global average gray value of the whole image. Shrivastava *et al.* [29] respectively defined them as follows.

$$\text{LAVG}_{P,R}(\mathbf{q}_c) = \frac{1}{P+1} \left( \mathcal{I}(\mathbf{q}_c) + \sum_{i=0}^{P-1} \mathcal{I}(\mathbf{p}_i) \right) \quad \text{and} \quad \text{GAVG}(\mathcal{I}) = \frac{1}{N} \sum_{j=1}^N \mathcal{I}(\mathbf{q}_j) \quad (2.18)$$

where  $\mathbf{q}_j$  is the  $j^{\text{th}}$  pixel in  $N$  total pixels of image  $\mathcal{I}$ .

Motivated by the incorporation of CLBP's components, the 3D-joint one of these complementary CLSP's components (i.e., CLSP<sub>S/M/C</sub>) is used in our works to structure descriptor CLSP-TOP [C1] based on encoding the raw plane-images of a given video, and CSAP-TOP [J1] based on the plane-images filtered by a model of moment images (see Sections 5.2.1 and 5.3 for further information).

### 2.7.4 LBP-based variants for textural image description

Motivated by LBP and its completed model, many works have been proposed to address issues of textural image representation. A global overview of state-of-the-art methods is particularly presented in [85], where LBP-based variants have been stated as one of important solutions for computer vision applications in practice. This is thanks to the effectiveness of those in structuring textural images with simple computing approaches. Indeed, Liu *et al.* [86] presented a taxonomy of local-feature-based descriptors for texture recognition issue, in which many original LBP-based methods and related extensions are thoroughly implemented. Then evaluations of their ability in noise-resistance and texture classification are comprehensively discussed in their work. Accordingly, the Median Robust Extended Local Binary Pattern (MRELBP) descriptor [78], structured by combining Radial difference (RD), Neighbor (NI), and Center pixel (CI) components, has the best overall performance. In the meanwhile, other LBP-based methods are also considerable for textural image representation: BRINT [87], CINIRD [88], SSLBP [89]. Also, all of those may be potential solutions for video representation in real applications.

### 2.7.5 LBP-based variants for DT representation

Thanks to beneficial properties of LBP-based variants in effectively computing local structures, several works have taken them into account DT representation. Zhao *et al.* [14] introduced volume of LBP-based patterns (VLBP) for analyzing a video in which a voxel is encoded in consideration of its  $3P$  neighbors that are sampled by itself and its two symmetrical voxels in the previous and posterior frames. As the result, it takes up to  $2^{3P+2}$  bins for describing a VLBP pattern, a noticeable restriction for real applications in computer vision. In order to overcome the curse of enormous dimensionality, Zhao *et al.* [14] suggested that a voxel is structured by using its  $P$  neighbors on each orthogonal plane of a video to form LBP-TOP patterns, instead of regarding consecutive frames in the VLBP encoding. The obtained histograms are concatenated and normalized to form the final descriptor with  $3 \times 2^P$  bins. Afterwards, many efforts have principally relied on these encoding concepts to improve the discriminative performance: CVLBC [90] - a combination of CLBC [82] and VLBP; CVLBP [91] - an integration of CLBP [3] and VLBP; CLSP-TOP [C1], CSAP-TOP [J1], and HLBP [92] - dealing with problems of sensitivity to noise and near uniform regions.

## 2.8 Datasets and protocols for evaluations of DT recognition

For DT recognition, the following benchmark datasets are often used to evaluate the ability of our proposed descriptors as well as that of others in state of the art. In this section, we express their properties and protocols one by one. Afterwards, a brief of those is shown in Table 2.1 for a quick reference.



Figure 2.5: Several samples of UCLA at line (a) and DynTex at line (b).

### 2.8.1 UCLA dataset

UCLA [5] consists of 200 videos which are recorded in  $110 \times 160 \times 75$  dimension to capture textural motions such as fountain, waterfall, flower, plant, etc. (see Figure 2.5 at line (a) for several instances of DT videos). The following protocols are usually addressed for DT recognition.

- *50-class breakdown*: Two experimental settings are usually focused on this scheme:  
*Leave-one-out (LOO)*: Following the protocol in [5, 72, 93], just one sample in the scheme is taken out for testing and the rest for training. This trial is performed in repetition for all samples and the final estimation is resulted by the mean of all obtained rates.  
*Four cross-fold validation (4fold)*: As the setting in [49, 72, 92], one-fourth of each class is addressed for testing and the remain for learning. The experiment is repeated four times with distinct test samples for each runtime. The final rate is reported by the average of all repetitions.
- *9-class breakdown*: This scheme is reorganized from the *50-class* model by categorizing its DT sequences into 9 classes named as *boiling water*(8), *fire*(8), *flowers*(12), *fountains*(20), *plants*(108), *sea*(12), *smoke*(4), *water*(12), and *waterfall*(16), where the numbers in parentheses denote quantities for the groups. The experimental setting is followed as that in [43, 49, 55], in which one half of DT sequences in each category is randomly selected for training and the remain for testing. The average of 20 runtimes is reported as the output rate.
- *8-class breakdown*: As the dominant cardinality of the *plants*(108) group in *9-class*, it is eliminated to form a *8-class* scheme with more challenges for DT evaluation. Following [43, 49], the configuration for experiments is set like that 50% of DT sequences, randomly taken out from each class, is utilized for training and the rest for testing. Similar to *9-class*, the trial on this scheme is also run 20 times and the mean of those is reported as the final rate.

### 2.8.2 DynTex dataset

DynTex [54], a challenging dataset for classifying DTs, is broadly used for evaluating abilities of state-of-the-art approaches in DT representation. Basically, it contains 679 videos in AVI format which are recorded in various conditions of environmental changes and fixed at dimension of  $352 \times 288 \times 250$  (see Figure 2.5 at line (b) for some samples of DT videos). DynTex is usually arranged into the following sub-datasets for classification task using the LOO protocol [64, 94, C5]:

- *DynTex35*: It consists of 35 categories which are correspondingly composed from 35 DynTex videos as follows. As the experimental settings in [14, 72, 92, C2, J1], each sequence is split into 8 non-overlapping sub-DTs at random clipping points subject to X, Y, T axes, but not half in those. For example, the clipping points are indicated as in [14], i.e.,  $x = 170, y = 130, t = 100$ . Furthermore, two more sub-videos are also captured by randomly splitting along T axis of the original sequence. As a result, 10 sub-videos for each of 35 videos are addressed in various spatial and temporal dimensions.

- *Alpha*: The subset consists of 60 videos which are labeled in three categories: “grass”, “sea”, and “trees”. Each category contains 20 sequences.
- *Beta*: The subset includes 162 videos grouped into 10 classes: “sea(20)”, “vegetation(20)”, “trees(20)”, “flags(20)”, “calm water(20)”, “fountains(20)”, “traffic(9)”, “smoke(16)”, “escalator(7)”, and “rotation(10)”, where the numbers in the parentheses correspondingly denote quantities of videos in the classes.
- *Gamma*: 264 DynTex videos are arranged into 10 groups: “flowers(29)”, “sea(38)”, “naked trees(25)”, “foliage(35)”, “escalator(7)”, “calm water(30)”, “flags(31)”, “grass(23)”, “traffic(9)”, and “fountains(37)”, where the numbers in the parentheses correspondingly mean quantities of sequences in the groups.

### 2.8.3 DynTex++ dataset

The sequences in DynTex dataset are restructured to form a richer benchmark for DT recognition, named DynTex++ [55]. Accordingly, 345 DynTex’s raw videos are split into sub-sequences with the fixed size of  $50 \times 50 \times 50$  so that they just include the main streams of DTs. The clipped DTs are then filtered by some techniques to expose 3600 sequences, those which are grouped into 36 categories with 100 DTs for each. We follow the same experimental setting in [55, 72, 95] for evaluation. It means that one half of samples from each class is randomly selected for training, and the remain for testing. The experiment is repeated 10 times to report the average performance as the final result.

### 2.8.4 DTDB dataset

DTDB [4] recently is a large scale dataset of DT videos for principally evaluating effectiveness of CNN-based proposals. It consists of over 10000 DT videos with a total of  $\sim 3.5$ M frames captured from different sources: websites, handled cameras, etc. (see Figure 2.6 for some samples). Two challenging schemes are addressed for DT recognition as follows.

- *Appearance* scheme consists of 45 categories, where its DT videos were selected from DTDB so that they mostly focus on features of spatial appearance, i.e., independent of dynamics (see Figure 2.6 at line (a) for some instances).
- *Dynamics* scheme consists of 18 categories. Contrary to *Appearance*, its DT videos, selected from DTDB, just include features of dynamics, i.e., independent of spatial appearance (see Figure 2.6 at line (b) for some instances).

Following the settings set by [4], 70% of samples in each category is randomly selected for training and the rest (30%) for testing. This trial is repeated 10 runtimes and the final result is reported by the average of the achieved rates.

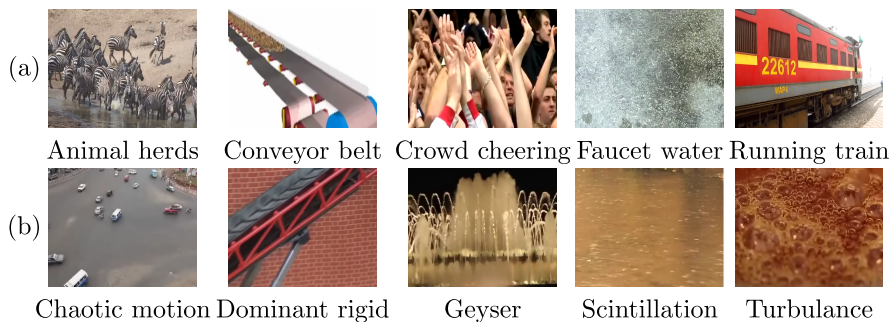


Figure 2.6: Several samples of DTDB, line (a): Appearance, line (b): Dynamics.

Table 2.1: A brief of main properties of DT datasets.

Dataset	Sub-dataset	#Videos	Resolution	#Classes	Protocol
UCLA	50-class	200	$110 \times 160 \times 75$	50	LOO and 4-fold
	9-class	200	$110 \times 160 \times 75$	9	50%/50%
	8-class	92	$110 \times 160 \times 75$	8	50%/50%
DynTex	DynTex35	350	different dimensions	10	LOO
	Alpha	60	$352 \times 288 \times 250$	3	LOO
	Beta	162	$352 \times 288 \times 250$	10	LOO
	Gamma	264	$352 \times 288 \times 250$	10	LOO
DynTex++		3600	$50 \times 50 \times 50$	36	50%/50%
DTDB	Dynamics	> 10000	different dimensions	18	70%/30%
	Appearance	> 9000	different dimensions	45	70%/30%

## 2.9 Classifiers for evaluating DT representation

In order to evaluate the performance of proposed descriptors, most of the approaches for DT representation often address two popular classifiers as follows.

- **K-nearest neighbors (K-NN):** The K-nearest neighbor (K-NN) technique, first introduced by Fix *et al.* [96] is one of the most popular classifier. For evaluating DT description, the most simple variant of K-NN, formed when  $K = 1$  (i.e., 1-NN), is often allocated together the  $\chi^2$  measure to estimate the dissimilarity  $D$  between two histograms. Accordingly, a test sample  $t$  is correctly classified with model  $m$  if it has one of the training samples from a similar class as its nearest neighbor.

$$D(t, m) = \sum_{b=1}^B \frac{(t_b - m_b)^2}{t_b + m_b} \quad (2.19)$$

where  $B$  is the total of bins,  $t_b$  and  $m_b$  are the values of the sample and the model image at the  $b^{th}$  bin respectively.

- **Support vector machines (SVMs):** Given a set of instance-label pairs  $\{\mathbf{x}_i, y_i\}_{i=1}^k$ ,  $\mathbf{x}_i \in \mathbb{R}^n$ ,  $y_i \in \{-1, +1\}$ , SVM, proposed in [97, 98], solves the following unconstrained optimization problem with different loss functions  $\psi(\mathbf{w}; \mathbf{x}_i, y_i)$  in general as

$$\min_{\mathbf{w}} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^k \psi(\mathbf{w}; \mathbf{x}_i, y_i) \quad (2.20)$$

where  $C > 0$  is the regularization parameter; two common loss functions are  $\psi(\mathbf{w}; \mathbf{x}_i, y_i) = \max(1 - y_i \mathbf{w}^T \mathbf{x}_i, 0)$  and  $\psi(\mathbf{w}; \mathbf{x}_i, y_i) = \max(1 - y_i \mathbf{w}^T \mathbf{x}_i, 0)^2$ . For evaluating DT description, we address a linear instance of multi-class SVM classifier which is implemented in the LIBLINEAR<sup>1</sup> library [99]. The default parameters for training and testing stages are located in our experiments.

In general, our experiments using SVM for classifying DTs have obtained better rates than using 1-NN. It could be seen in some of our works [C1, C4, J1, J2]. Therefore, in this thesis, we just mention the results computed by the SVM classifier.

<sup>1</sup><https://www.csie.ntu.edu.tw/~cjlin/liblinear>



---

---

## CHAPTER 3

---

# PROPOSED VARIANTS OF LBP-BASED OPERATORS

### Contents

<b>3.1</b>	<b>Introduction</b>	<b>21</b>
<b>3.2</b>	<b>Completed Adaptive Patterns (CAIP)</b>	<b>22</b>
<b>3.3</b>	<b>Some extensions of Local Derivative Patterns (xLDP)</b>	<b>24</b>
3.3.1	Local Derivative Patterns	24
3.3.2	Adaptive directional thresholds	25
3.3.3	Completed model of LDP	26
3.3.4	Assessing our proposed extensions of LDP	27
<b>3.4</b>	<b>Some extensions of local vector patterns (xLVP)</b>	<b>27</b>
3.4.1	Local Vector Patterns	27
3.4.2	Adaptive directional vector thresholds	28
3.4.3	A completed model of LVP	28
<b>3.5</b>	<b>Local Rubik-based Patterns (LRP)</b>	<b>30</b>
3.5.1	Complemented components	30
3.5.2	Construction of LRP patterns	31
<b>3.6</b>	<b>Completed Hierarchical Local Patterns (CHILOP)</b>	<b>32</b>
3.6.1	Construction of CHILOP	33
3.6.2	A particular degeneration of CHILOP into CLBP	36
3.6.3	Beneficial properties of CHILOP operator	36
<b>3.7</b>	<b>Summary</b>	<b>36</b>

### 3.1 Introduction

As mentioned in Section 2.7, with simple computation, the local-feature-based methods have obtained potential performances in analyzing shapes and motions of DTs in videos. In spite of that, their executions have been restricted due to the negative impacts of the well-known problems: noise, near uniform region, etc. Motivated by the effectiveness and simplicity of LBP and its variants, in this chapter, we propose several powerful operators that are able to more efficiently capture local relationships with more robustness against those problems. Contributions of our proposed operators can be listed as follows.

Table 3.1: A brief of our proposed operators for local DT encoding.

#	Operator	Description	Applied to	Referred to
1	CAIP [S4]	Completed Adaptive Patterns	Encoding bipolar Gaussian-based filtered features (see Section 6.6)	Section 3.2
2	xLDP [J5]	Important extensions of local derivative patterns [30]	Encoding moment-filtered volumes (see Section 5.4)	Section 3.3
3	xLVP [J2]	Important extensions of local vector patterns [100]	Encoding features of dense trajectories (see Section 4.3)	Section 3.4
4	LRP [J4]	Local Rubik-based Patterns	Encoding features of Gaussian-based filtered volumes (see Section 6.5)	Section 3.5
5	CHILOP [S2]	Completed Hierarchical Local Patterns	Encoding features of Gaussian-based filtered elements (see Section 6.4)	Section 3.6

- Proposed CAIP is an adaptive solution of CLBP [3] in order to deal with problems of close-to-zero pixels caused by decomposition of bipolar Gaussian-based outcomes for DT representation [S4].
- Proposed xLDP is a completed model of Local Derivative Patterns (LDP) [30], which its complementary components are based on novel adaptative directional thresholds. xLDP is then addressed to capture local features of moment-filtered volumes [J5].
- Proposed xLVP is a completed model of Local Vector Patterns (LVP) [100], which its complementary components are based on novel adaptive directional vector thresholds. xLVP is then exploited to capture local features of dense trajectories [J2].
- Proposed LRP is inspired by a rubik-based concept in consideration of 6 local sides and 3 orthogonal plane-images which are located around a center voxel. LRP is then used to encode characteristics of voxels in Gaussian-filtered volumes [J4].
- Proposed CHILOP is constructed by addressing different gray levels of two hierarchical neighbors allocated by a center pixel. CHILOP is then used for capturing spatio-temporal features from filtered plane-images computed by a Gaussian filtering kernel [S2].

For convenience, Table 3.1 illustrates a brief of our proposed operators which are exploited to encode local DT features from different aspects of video analysis. Hereunder, we present the mechanism of their construction in detail.

### 3.2 Completed Adaptive Patterns (CAIP)

Utilizing LBP-based variants for exploiting the beneficial properties of bipolar-based images can be in trouble due to the zero-pixel problem. Indeed, all zero-pixels are structured by a  $P$ -bit-1 string when using the conventional LBP operator, which have been addressing the center as a threshold. Figure 3.1 graphically demonstrates a specific example using component  $\text{CLBP}_S$  (i.e., the typical LBP) of CLBP [3]). In order to overcome this problem, we propose in this section a crucial adaptation for CLBP operator so that its adapted version is able to efficiently capture invariant characteristics in the bipolar-based images. The reason that CLBP is addressed for the adaptation is its simplicity and effectiveness as well as one of the most popular LBP-based variants in encoding local features. It is worth noting that this modification can be similarly applied to other LBP-based variants for improving their performances in encoding such bipolar-based images, e.g., CLBC [82], LDP-based [30, J5], LVP-based [100, J2], LRP [J4], etc. Hereunder, we present in detail the procedure to make CLBP more robust against the negative impacts of the zero-bipolar cells.

In order to maintain the advantages of blurred and bipolar-invariant characteristics in  $\Omega_{\sigma, \sigma'}^{2D/3D}$  for DT encoding, a center pixel  $\mathbf{q}$  with its zero-gray-level should be replaced by the mean of its local neighbors  $\{\mathbf{p}_i\}$  taken into account as follows.

$$\tilde{m}_{\mathbf{q}} = \frac{1}{P} \sum_{i=0}^{P-1} \mathcal{I}(\mathbf{p}_i) \quad (3.1)$$

where  $P$  is the number of neighbors,  $\mathcal{I}(\cdot)$  returns the gray-scale of a pixel. It should be noted that this is a

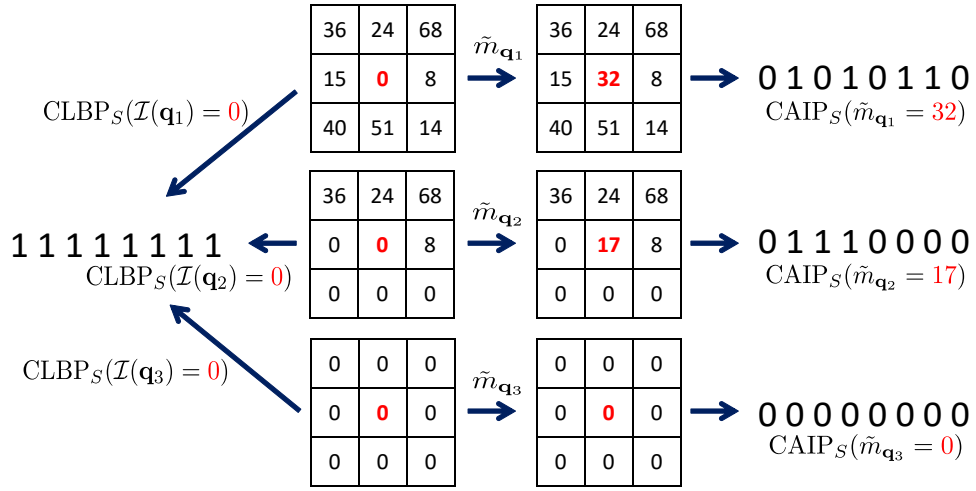


Figure 3.1: An issue example of encoding bipolar-based images in which the fact that CLBP<sub>S</sub> figures out a same pattern for three different local textures is resolved by our CAIP<sub>S</sub>.

little difference compared to the encoding introduced by Jin *et al.* [101], where all center pixels of a still image are involved with this replacement. According to that, the original complementary components CLBP<sub>S</sub>, CLBP<sub>M</sub>, and CLBP<sub>C</sub> for a center pixel  $\mathbf{q}$  are adapted to correspondingly capture Completed Adaptive Patterns (CAIP) with more robustness against the zero-pixel problem. More specifically, the first component, CAIP<sub>S</sub>, is turned as follows.

$$\text{CAIP}_S(\mathbf{q}) = \begin{cases} \text{CLBP}_S(\mathbf{q}), & \text{if } \mathcal{I}(\mathbf{q}) \neq 0 \\ \sum_{i=0}^{P-1} h(\mathcal{I}(\mathbf{p}_i) - \tilde{m}_{\mathbf{q}}) \times 2^i, & \text{if } \mathcal{I}(\mathbf{q}) = 0 \end{cases} \quad (3.2)$$

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

$$h(x) = \begin{cases} 1, & x > 0 \\ 0, & \text{otherwise.} \end{cases} \quad (3.3)$$

It can be verified from Figure 3.1 that our proposed adaptation is crucial for encoding bipolar-invariant features. Indeed, zero-center pixels (i.e.,  $\mathcal{I}(\mathbf{q}_1) = \mathcal{I}(\mathbf{q}_2) = \mathcal{I}(\mathbf{q}_3) = 0$ ) with different types of their local neighbors are structured to more discriminative patterns using CAIP<sub>S</sub> operator. In the meanwhile, all of them are featured by only one pattern when using the typical CLBP<sub>S</sub> (see Figure 3.1). It should be noted that the thresholding function  $h(\cdot)$  is defined in a little difference from that of LBP-based variants (i.e., Equation (2.5)) in order to eliminate meaningless encoding of zero-center textural pixels with their local zero-area neighbors, as in case of structuring a pattern for  $\mathbf{q}_3$  shown in Figure 3.1.

Similarly, the second (i.e., CAIP<sub>M</sub>) and the last (i.e., CAIP<sub>C</sub>) components are respectively adapted with  $\tilde{m}_{\mathbf{q}}$  as

$$\text{CAIP}_M(\mathbf{q}) = \begin{cases} \text{CLBP}_M(\mathbf{q}), & \text{if } \mathcal{I}(\mathbf{q}) \neq 0 \\ \sum_{i=0}^{P-1} g(|\mathcal{I}(\mathbf{p}_i) - \tilde{m}_{\mathbf{q}}|, \tilde{m}_{\mathcal{I}}) \times 2^i, & \text{if } \mathcal{I}(\mathbf{q}) = 0 \end{cases} \quad (3.4)$$

and

$$\text{CAIP}_C(\mathbf{q}) = \begin{cases} \text{CLBP}_C(\mathbf{q}), & \text{if } \mathcal{I}(\mathbf{q}) \neq 0 \\ g(\tilde{m}_{\mathbf{q}}, \tilde{c}_{\mathcal{I}}), & \text{if } \mathcal{I}(\mathbf{q}) = 0 \end{cases} \quad (3.5)$$

where binary function  $g(\cdot)$  is defined in Equation (2.11).

Since those components (i.e., CAIP<sub>S</sub>, CAIP<sub>M</sub>, CAIP<sub>C</sub>) are complementary, they should be integrated in different ways to form histograms with more robustness. Among of them, the 3D joint setting of



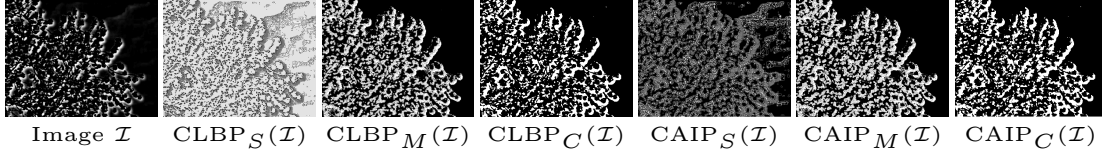


Figure 3.2: A comparison of patterns structured by CLBP and CAIP operators with settings of  $\{(P, R)\} = \{(8, 1)\}$  and  $riu2$  mapping to encode a Gaussian-based positive-bipolar image  $\mathcal{I} = \mathcal{I}_{\text{DoG}_{\sigma, \sigma'}}^{\text{pos}}$  computed using  $\sigma = 0.7$  and  $\sigma' = 2\sqrt{5}\sigma$ .

those components is addressed in our work due to its outperformance. From now on, CAIP is implicated for this joint in the default circumstance.

In short, based on the adaptive modification, our CAIP has two following beneficial properties to improve the discrimination power compared to CLBP [3] and other LBP-based variants:

- In addition to inheriting the benefits of CLBP, CAIP is enhanced its performance thanks to efficiently dealing with the problem of encoding the zero-centered bipolar cells in Gaussian-based filtered outcomes of  $\Omega_{\sigma, \sigma'}^{2D/3D}$ . Figure 3.2 shows the effectiveness of our CAIP.
- In order to preserve blurred and bipolar-invariant features being useful for DT representation, only zero-pixels  $\mathbf{q}$  in the bipolar images are replaced by their  $\tilde{m}_{\mathbf{q}}$  instead of doing that for all pixels as done by Jin *et al.* [101]. This allows to reduce noise and to carry out near uniform regional problems (see Figure 3.1).

### 3.3 Some extensions of Local Derivative Patterns (xLDP)

The typical LDP operator has been initially proposed for face recognition [30] by exploiting local derivative direction variations and then successfully applied to other applications, such as action recognition [13]. We adopt in this section for the first time this operator to capture shape and motion cues for DT description. Moreover, we also propose two following important extensions of LDP operator to improve its discriminative power: adaptative directional thresholds and completed model of LDP.

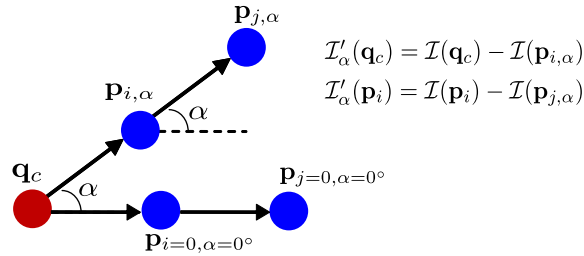


Figure 3.3: Model of the first-order LDP patterns of  $\mathbf{q}_c$  ( $\mathcal{I}'_{\alpha}(\mathbf{q}_c)$ ) and  $\mathbf{p}_i$  ( $\mathcal{I}'_{\alpha}(\mathbf{p}_i)$ ) pixels in directions  $\alpha \in \mathcal{D}$  in which  $\mathbf{q}_c$  (in red) is the considered point,  $\mathbf{p}_i$  is the  $i^{\text{th}}$  neighbor of  $\mathbf{q}_c$ , and  $\mathbf{p}_j$  is the  $j^{\text{th}}$  neighbor of  $\mathbf{p}_i$ .

#### 3.3.1 Local Derivative Patterns

Zhang *et al.* [30] introduced Local Derivative Patterns (LDPs), a directional extension of LBP, by taking into account local high-order derivative variations based on considering a pixel and its neighbors in different directions to capture more robust features.

The first-order LDP at a pixel for a set of considered directions  $\mathcal{D}$  is defined as follows.

$$\mathcal{I}'_{\alpha}(\mathbf{q}_c) = \mathcal{I}(\mathbf{q}_c) - \mathcal{I}(\mathbf{p}_{i,\alpha}) \quad (3.6)$$

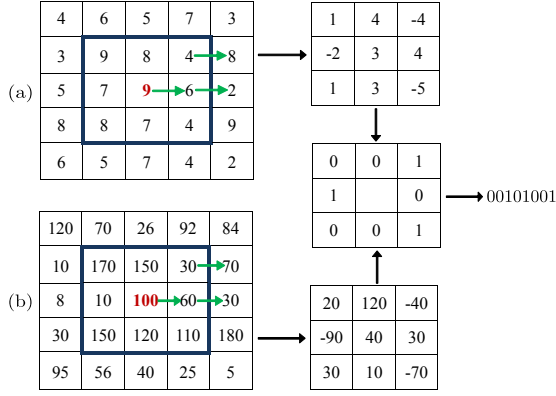


Figure 3.4: An example of two different local structures (marked in red color) are encoded by the same LDP pattern in concerned direction  $\alpha = 0^\circ$ .

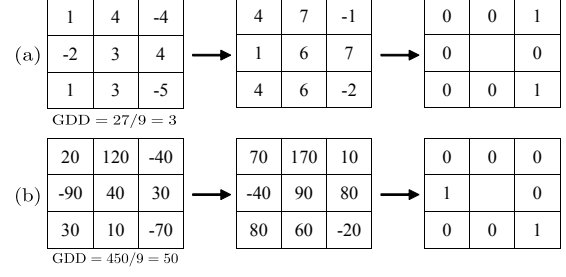


Figure 3.5: Two different local structures (a) and (b) in Figure 3.4 are encoded by different LDP-D patterns in direction  $\alpha = 0^\circ$ .

where  $\mathbf{p}_{i,\alpha}$  is the  $i^{th}$  neighbor of a center point  $\mathbf{q}_c$  in a concerning direction  $\alpha$ ,  $\mathcal{I}(\cdot)$  is gray-scale image level of a pixel. Figure 3.3 graphically illustrates the regular computation of the first-order LDP patterns corresponding to directions  $\alpha \in \mathcal{D}$ . In general, the  $n^{th}$ -order LDP is defined as follows, for the center pixel  $\mathbf{q}_c$  and its  $P$  neighbors circled with radius  $R$ .

$$\text{LDP}_{P,R,\alpha}^n(\mathbf{q}_c) = \{f(I_\alpha^{n-1}(\mathbf{q}_c), I_\alpha^{n-1}(\mathbf{p}_i))\}_{1 \leq i \leq P} \quad (3.7)$$

where  $I_\alpha^{n-1}(\cdot)$  means the  $(n-1)^{th}$ -order derivative in direction  $\alpha$  at a pixel,  $\mathbf{p}_i$  is the  $i^{th}$  neighbor of the center point  $\mathbf{q}_c$ , and function  $f(\cdot)$  is defined as

$$f(x, y) = \begin{cases} 1, & \text{if } x \times y \leq 0 \\ 0, & \text{otherwise.} \end{cases} \quad (3.8)$$

The detail of other LDP's formulations as well as samples of its calculation is discussed in [30]. In practice, four directions are often considered, i.e.,  $\alpha \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ , to capture directional mutual relations of pixels [13, 30]. In case of inspecting the first-order derivative variations in all of directions, LDP is simply identical to the basic LBP.

### 3.3.2 Adaptive directional thresholds

Similar to a well-known restriction of the typical LBP, LDP is not occasionally able to judge different structure patterns because its encoding is still thresholded by the center with around neighbors. It can be observed in Figure 3.4 that two different local structures (a) and (b) are figured out by the same pattern. In order to handle this issue, we propose to define three following adaptative thresholds<sup>1</sup> for LDP operator. The key idea for that is the consideration of the first-order LDP. These thresholds will be then exploited in Section 3.3.3 to construct the completed model of LDP.

Firstly, Global Directional Difference (GDD) of an image texture is calculated as the mean of absolute directional differences on the entire of concerned directions.

$$\text{GDD}(\mathcal{I}) = \frac{1}{|\mathcal{D}| \times \mathcal{N}} \left( \sum_{\mathbf{q}_j \in \mathcal{I}} |\mathcal{I}'_\alpha(\mathbf{q}_j)| \right) \Big|_{\alpha \in \mathcal{D}} \quad (3.9)$$

where  $\mathcal{N} = (\mathcal{W} - 2) \times (\mathcal{H} - 2)$ ,  $\mathcal{W}$  and  $\mathcal{H}$  are width and height dimensions of 2D image  $\mathcal{I}$  respectively,  $|\mathcal{D}|$  is the total of considered directions,  $\mathcal{I}'_\alpha(\cdot)$  is the first-order local derivative pattern of a pixel in regarding direction  $\alpha$ .

<sup>1</sup>Contrary to the two last thresholds, the first one is empirically proposed without depending on  $\alpha$  because this leads to more robust and stable results.

Secondly, to capture the information of Directional Magnitudes ( $DM_\alpha$ ) for each direction  $\alpha$ , we compute the mean of absolute multiplication of directional differences on the whole image as follows.

$$DM_\alpha(\mathcal{I}) = \frac{1}{\mathcal{N} \times P} \left( \sum_{i=0}^{P-1} |\mathcal{I}'_\alpha(\mathbf{q}_j) \times \mathcal{I}'_\alpha(\mathbf{p}_i)| \right) \Big|_{\mathbf{q}_j \in \mathcal{I}} \quad (3.10)$$

in which  $\mathbf{p}_i$  is the  $i^{th}$  neighbor of current pixel  $\mathbf{q}_j$  of image  $\mathcal{I}$ ,  $P$  is the number of considered neighbors.

Thirdly, the Directional Center ( $DC_\alpha$ ) threshold is defined as the average of directional centered differences on the whole image.

$$DC_\alpha(\mathcal{I}) = \frac{1}{\mathcal{N}} \sum_{\mathbf{q}_j \in \mathcal{I}} |\mathcal{I}'_\alpha(\mathbf{q}_j)| \quad (3.11)$$

### 3.3.3 Completed model of LDP

Guo *et al.* [3] showed that considering local variations of magnitudes together with the typical LBP makes the descriptor more robust and discriminant because they are complementary. Inspired by this idea, we introduce in this portion, a completed model of the second order LDP using adaptative thresholds, which are presented in Section 3.3.2. Similar to [3], it also consists of three following complementary components.

First, we propose LDP-D operator as the first component in order to capture the second-order local derivative patterns adjusted by an adaptive thresholding GDD (see Equation (3.9)) as follows.

$$LDP-D_{P,R,\alpha}(\mathbf{q}_c) = \sum_{i=0}^{P-1} \psi(\mathcal{I}'_\alpha(\mathbf{q}_c), \mathcal{I}'_\alpha(\mathbf{p}_i), GDD(\mathcal{I})) \times 2^i \quad (3.12)$$

where  $\mathbf{p}_i$  is the  $i^{th}$  neighbor of the center pixel  $\mathbf{q}_c$  in accordance with direction  $\alpha$ ,  $P$  is number of considered neighbors circled by radius  $R$ , and function  $\psi(\cdot)$  is estimated as

$$\psi(x, y, z) = \begin{cases} 1, & \text{if } (x + z) \times (y + z) \leq 0 \\ 0, & \text{otherwise.} \end{cases} \quad (3.13)$$

In assumption of just considering one direction of  $\alpha = 0^\circ$  (i.e.,  $|\mathcal{D}| = 1$ ), in contrast to the basic LDP, the proposed operator LDP-D is able to differentiate the local structures (a) and (b) as detailed in Figure 3.5.

Second, LDP-M component exploits the information of magnitudes in a direction  $\alpha$  by using adaptative threshold  $DM_\alpha$  (see Equation (3.10)) and is formed as

$$LDP-M_{P,R,\alpha}(\mathbf{q}_c) = \sum_{i=0}^{P-1} h(\mathcal{I}'_\alpha(\mathbf{q}_c), \mathcal{I}'_\alpha(\mathbf{p}_i), DM_\alpha(\mathcal{I})) \times 2^i \quad (3.14)$$

where  $h(\cdot)$  is defined as

$$h(x, y, z) = \begin{cases} 1, & \text{if } |x \times y| \geq z \\ 0, & \text{otherwise.} \end{cases} \quad (3.15)$$

Third, LDP-C regards to the directional contrast of a center against the mean of directional differences on the whole image.

$$LDP-C_\alpha(\mathbf{q}_c) = s(\mathcal{I}'_\alpha(\mathbf{q}_c) - DC_\alpha(\mathcal{I})) \quad (3.16)$$

in which  $s(\cdot)$  is defined by Equation (2.5).

Three above complements (abbreviated to  $LDP_D$ ,  $LDP_M$ , and  $LDP_C$ ) should be combined in different ways to produce extended LDP operator, named xLDP, for investigation to find out an enhanced operator LDP for encoding DT features. An instance of those is  $xLDP = LDP_{D-M/C}$ , in which the signals of “-” and “/” in the style  $D-M/C$  mean that histograms obtained by the corresponding components are concatenated and jointed respectively. It should be noted that our operator can be also generated in high-order derivative patterns ( $xLDP^n$ ) by exploiting the  $n^{th}$ -order directional LDPs ( $n > 2$ ) [30] for calculation of the proposed components above.

Our xLDP is different from the typical LDP [30] in several properties to enhance the performance:

- The xLDP operator considers local structures in diversity of directional relations based on 3 complemented components, in contrast to LDP with only in consideration of local derivative patterns.
- Our operator is more insensitive to noise when exploiting adaptative directional thresholds (see an instance of encoding patterns in Figures 3.4 and 3.5).
- To encode a local structure in each direction, LDPs are separately computed by using the corresponding components. In the meanwhile, the basic LDP encodes a pixel in a long binary chain for all concerned directions, e.g., a string of 32 bits for four 8-bit LDPs.
- Thanks to structuring patterns in separative strings of binary codes, two popular mappings of  $riu2$  and  $u2$  for the processing of description can be utilized to advance the performance of descriptor with practical dimension. In contrast, LDPs are calculated on subregions of an image texture with various parameters of histogram bins.

### 3.3.4 Assessing our proposed extensions of LDP

In order to evaluate the proposed complementary components for LDP operator, we also implement the basic LDP [30] for DT description based on the filtered videos captured by the proposed model of  $r$ -order moment volumes. For a center pixel  $\mathbf{q}_c$  and its  $P$  considered neighbors sampled by a circle with radius  $R$ , the second-order typical local derivative pattern (LDP) of  $\mathbf{q}_c$  in direction  $\alpha$ , named  $LDP_{P,R,\alpha}$ , is defined as

$$LDP_{P,R,\alpha}(\mathbf{q}_c) = \sum_{i=0}^{P-1} f(\mathcal{I}'_{\alpha}(\mathbf{q}_c), \mathcal{I}'_{\alpha}(\mathbf{p}_i)) \times 2^i \quad (3.17)$$

where the function  $f(\cdot)$  is defined by Equation (3.8). Actually, this operator is the same LDP-D without exploiting the adaptative threshold of GDD.

## 3.4 Some extensions of local vector patterns (xLVP)

The basic LVP operator [100] 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 [102], image retrieval [103]. 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.

### 3.4.1 Local Vector Patterns

Fan *et al.* [100] 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

$\alpha$  is computed as

$$\mathcal{I}'_{\alpha,d}(\mathbf{q}_c) = \mathcal{I}(\mathbf{q}_{\alpha,d}) - \mathcal{I}(\mathbf{q}_c) \quad (3.18)$$

in which  $\mathbf{q}_{\alpha,d}$  is an adjacent neighbor sampled by direction  $\alpha$  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.

$$\text{LVP}_{P,R,\alpha,d}(\mathbf{q}_c) = \sum_{i=0}^{P-1} f(\mathcal{I}'_{\alpha,d}(\mathbf{q}_c), \mathcal{I}'_{\alpha+45^\circ,d}(\mathbf{q}_c), \mathcal{I}'_{\alpha,d}(\mathbf{p}_i), \mathcal{I}'_{\alpha+45^\circ,d}(\mathbf{p}_i)) \times 2^i \quad (3.19)$$

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

$$f(x, y, z, t) = \begin{cases} 1, & \text{if } t - \frac{y \times z}{x} \geq 0 \\ 0, & \text{otherwise.} \end{cases} \quad (3.20)$$

Other formulations of LVP along with samples of encoding LVP-based patterns for texture images are clearly discussed in [100]. 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 [100, 102, 103].

### 3.4.2 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  $\alpha$ , we calculate the mean of absolute CST on the whole image as

$$\text{DVM}_{\alpha,d}(\mathcal{I}) = \frac{1}{\mathcal{N} \times P} \sum_{\mathbf{q} \in \mathcal{I}} \sum_{i=0}^{P-1} \left| \mathcal{I}'_{\beta,d}(\mathbf{p}_i) - \frac{\mathcal{I}'_{\beta,d}(\mathbf{q})}{\mathcal{I}'_{\alpha,d}(\mathbf{q})} \times \mathcal{I}'_{\alpha,d}(\mathbf{p}_i) \right| \quad (3.21)$$

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

$$\text{DVC}_{\alpha,d}(\mathcal{I}) = \frac{1}{\mathcal{N}} \sum_{\mathbf{q} \in \mathcal{I}} |\mathcal{I}'_{\alpha,d}(\mathbf{q}) \times \mathcal{I}'_{\beta,d}(\mathbf{q})| \quad (3.22)$$

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

### 3.4.3 A completed model of LVP

Guo *et al.* [3] 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 3.4.2. In essence, it is an integration of three following parts:

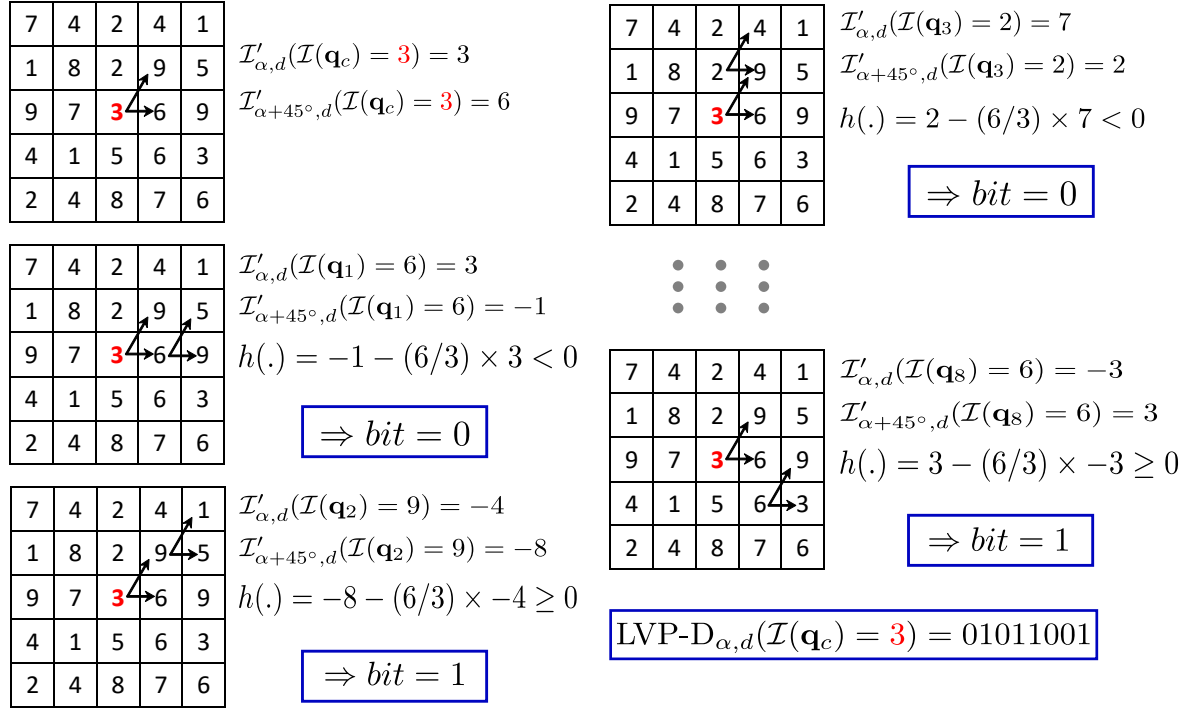


Figure 3.6: Computing the first-order LVP-D binary pattern for a dynamic point  $\mathcal{I}(\mathbf{q}_c) = 3$  (in red) with  $\alpha = 0^\circ$ ,  $d = 1$ , and  $(P, R) = (8, 1)$ .

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.

$$\text{LVP-D}_{P,R,\alpha,d}(\mathbf{q}_c) = \sum_{i=0}^{P-1} h(\mathcal{I}'_{\alpha,d}(\mathbf{q}_c), \mathcal{I}'_{\beta,d}(\mathbf{q}_c), \mathcal{I}'_{\alpha,d}(\mathbf{p}_i), \mathcal{I}'_{\beta,d}(\mathbf{p}_i)) \times 2^i \quad (3.23)$$

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

$$h(x, y, u, v) = \begin{cases} 1, & \text{if } v \geq u \times \frac{y}{x} \\ 0, & \text{otherwise.} \end{cases} \quad (3.24)$$

The fact that each LVP-D pattern is similar to the basic LVP [100], 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 Figure 3.6 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  $\Phi$  as follows:

$$\text{LVP-M}_{P,R,\alpha,d}(\mathbf{q}_c) = \sum_{i=0}^{P-1} \psi(\mathcal{I}'_{\alpha,d}(\mathbf{q}_c), \mathcal{I}'_{\beta,d}(\mathbf{q}_c), \mathcal{I}'_{\alpha,d}(\mathbf{p}_i), \mathcal{I}'_{\beta,d}(\mathbf{p}_i), \text{DVM}_{\alpha,d}(\mathcal{I})) \times 2^i \quad (3.25)$$

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

$$\psi(x, y, u, v, t) = \begin{cases} 1, & \text{if } |v - u \times \frac{y}{x}| \geq t \\ 0, & \text{otherwise.} \end{cases} \quad (3.26)$$

### 3.5. LOCAL RUBIK-BASED PATTERNS (LRP)

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

$$\text{LVP-C}_{\alpha,d}(\mathbf{q}_c) = s(\mathcal{I}'_{\alpha,d}(\mathbf{q}_c) - \text{DVC}_{\alpha,d}(\mathcal{I})) \quad (3.27)$$

in which  $s(\cdot)$  is defined by Equation (2.5).

These components (respectively abbreviated to  $\text{LVP}_D$ ,  $\text{LVP}_M$ , and  $\text{LVP}_C$ ) 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,  $\text{xLVP} = \text{LVP}_{D\_M/C}$  means that probability distributions structured by  $\text{LVP}_D$ ,  $\text{LVP}_M$ , and  $\text{LVP}_C$  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  $n^{\text{th}}$ -order derivative ( $n > 1$ ) to capture high-order directional patterns ( $\text{xLVP}^n$ ), as similarly as generated in [100].

Our xLVP operator takes into account several following properties to improve the performance in comparison with the basic LVP [100]:

- 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.
- Due to encoding directional features in separative 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 [100].

## 3.5 Local Rubik-based Patterns (LRP)

### 3.5.1 Complemented components

Motivated by the conception of complemented components in [3, 90, 91], three prominent components are proposed to address forceful discrimination of local textural features by adapting the concept to the supporting region constructed from 6 sides of a rubik cube and by introducing new concepts of encoding and thresholding dedicated to this neighborhood configuration for three completed components (see Equations (3.29), (3.31), (3.32), and (3.34)). Accordingly, let  $\mathbf{q}$  be a voxel in a video  $\mathcal{V}$ ;  $\mathbf{q}_f$  be its projection on a plane-image  $f \in \mathcal{V}$  (see Figure 3.7(a) for a graphical illustration). Figure 3.7(b) presents our neighborhood supporting region which is constructed from 6 sides of a rubik cube centered at the voxel together with 3 orthogonal planes passing through this voxel. The first component captures the differences between the mean gray-level center points (i.e.,  $\mathbf{q}$ ,  $\mathbf{q}_f$ ) and each of  $\{\mathbf{p}_{i,f}\}$  local neighbors of  $\mathbf{q}_f$  as follows.

$$D_{P,R,f}(\mathbf{q}, \mathbf{q}_f) = \sum_{i=0}^{P-1} s(\mathcal{I}(\mathbf{p}_{i,f}), \mathcal{I}(\mathbf{q}_f), \mathcal{I}(\mathbf{q})) \times 2^i \quad (3.28)$$

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

$$s(x, y, z) = \begin{cases} 1, & x \geq \frac{y+z}{2} \\ 0, & \text{otherwise.} \end{cases} \quad (3.29)$$

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

$$M_{P,R,f}(\mathbf{q}, \mathbf{q}_f) = \sum_{i=0}^{P-1} h(\mathcal{I}(\mathbf{p}_{i,f}), \mathcal{I}(\mathbf{q}_f), \mathcal{I}(\mathbf{q}), \bar{m}_f) \times 2^i \quad (3.30)$$

in which  $\bar{m}_f$  and function  $h(\cdot)$  are defined in (3.31) and (3.32) respectively,  $\mathcal{N}$  means the quantity of pixels  $\{\mathbf{q}_j\}$  in current image  $f$ .

$$\bar{m}_f = \frac{1}{P \times \mathcal{N}} \sum_{j=0}^{\mathcal{N}} \sum_{i=0}^{P-1} \left( \mathcal{I}(\mathbf{p}_{i,f}) - \frac{\mathcal{I}(\mathbf{q}_{j,f}) + \mathcal{I}(\mathbf{q})}{2} \right) \quad (3.31)$$

$$h(x, y, z, t) = \begin{cases} 1, & x - \frac{y+z}{2} \geq t \\ 0, & \text{otherwise.} \end{cases} \quad (3.32)$$

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

$$C_{P,R,f}(\mathbf{q}, \mathbf{q}_f) = g(\mathcal{I}(\mathbf{q}_f) + \mathcal{I}(\mathbf{q}) - \bar{c}_f) \quad (3.33)$$

where  $g(\cdot)$  is identical to Equation (2.5) and  $\bar{c}_f$  is computed as

$$\bar{c}_f = \frac{1}{\mathcal{N}} \sum_{j=0}^{\mathcal{N}} \left( \mathcal{I}(\mathbf{q}_{j,f}) + \mathcal{I}(\mathbf{q}) \right) \quad (3.34)$$

Those components are complementary [3]. Therefore, their integration is recommended in order to improve the discriminant power. Let  $\text{DMC}_{P,R,\Omega}(\cdot)$  be an integration  $\Omega$  of the complemented components (i.e.,  $D_{P,R,f}(\cdot)$ ,  $M_{P,R,f}(\cdot)$ ,  $C_{P,R,f}(\cdot)$ ) subject to each voxel. For instance,  $\text{DMC}_{P,R,\Omega}(\mathbf{q}, \mathbf{q}_{f_{i-1}})$  computes  $D_{P,R,f_{i-1}}(\mathbf{q}, \mathbf{q}_{f_{i-1}})$ ,  $M_{P,R,f_{i-1}}(\mathbf{q}, \mathbf{q}_{f_{i-1}})$ , and  $C_{P,R,f_{i-1}}(\mathbf{q}, \mathbf{q}_{f_{i-1}})$  based on  $\mathbf{q}$ 's central symmetry voxel  $\mathbf{q}_{f_{i-1}}$  at image  $f_{i-1}$  in plane  $XY$  (see Figure 3.7(c) for a sample of this computation). Those are then integrated into different ways  $\Omega$  to form space-completed patterns. Therein,  $\Omega = \{D\_M/C, D\_M/C, \text{etc.}\}$  where signs “-” and “/” mean operations of concatenating and jointing probability distributions of the components respectively, e.g., “ $D\_M/C$ ” indicates that a joint histogram of  $M(\cdot)$  and  $C(\cdot)$  is concatenated to that of  $D(\cdot)$ .

### 3.5.2 Construction of LRP patterns

Based on the concept of complemented model in the previous section, we introduce hereafter the novel LRP operator. For a video  $\mathcal{V}$ , let a center voxel  $\mathbf{q} \in \mathcal{V}$  be an intersection point of orthogonal plane images  $f_i \in XY$ ,  $f_j \in XT$ , and  $f_k \in YT$  where  $\{XY, XT, YT\}$  are planes of  $\mathcal{V}$ . A rubik cube  $\Gamma$  of  $\mathbf{q}$  is addressed in consideration of the previous and posterior plane-images of  $f_i$ ,  $f_j$ , and  $f_k$  respectively (i.e.,  $f_{i-1}, f_{i+1}$  for  $XY$ ,  $f_{j-1}, f_{j+1}$  for  $XT$ ,  $f_{k-1}, f_{k+1}$  for  $YT$ , see Figure 3.7(b) for a graphical instance). A local rubik-based pattern for  $\mathbf{q}$  is structured by integrating complementary components computed on 6 sides and 3 orthogonal plane-images of rubik cube  $\Gamma$  as follows.

$$\text{LRP}_{\Gamma,\Omega}(\mathbf{q}) = \biguplus_{f \in \mathcal{F}} [\text{DMC}_{P,R,\Omega}(\mathbf{q}, \mathbf{q}_f)] \quad (3.35)$$

in which  $\mathcal{F} = \{f_{i-1}, f_i, f_{i+1}, f_{j-1}, f_j, f_{j+1}, f_{k-1}, f_k, f_{k+1}\}$  is a set of 6 sides and 3 orthogonal plane-images of rubik cube  $\Gamma$ ,  $\mathbf{q}_f$  is the central symmetry voxel of  $\mathbf{q}$  that is orthogonally projected on plane-image  $f$  (see Figure 3.7(a) for an instance of a projection of  $\mathbf{q}$ );  $\biguplus$  denotes a concatenating function of histograms.

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



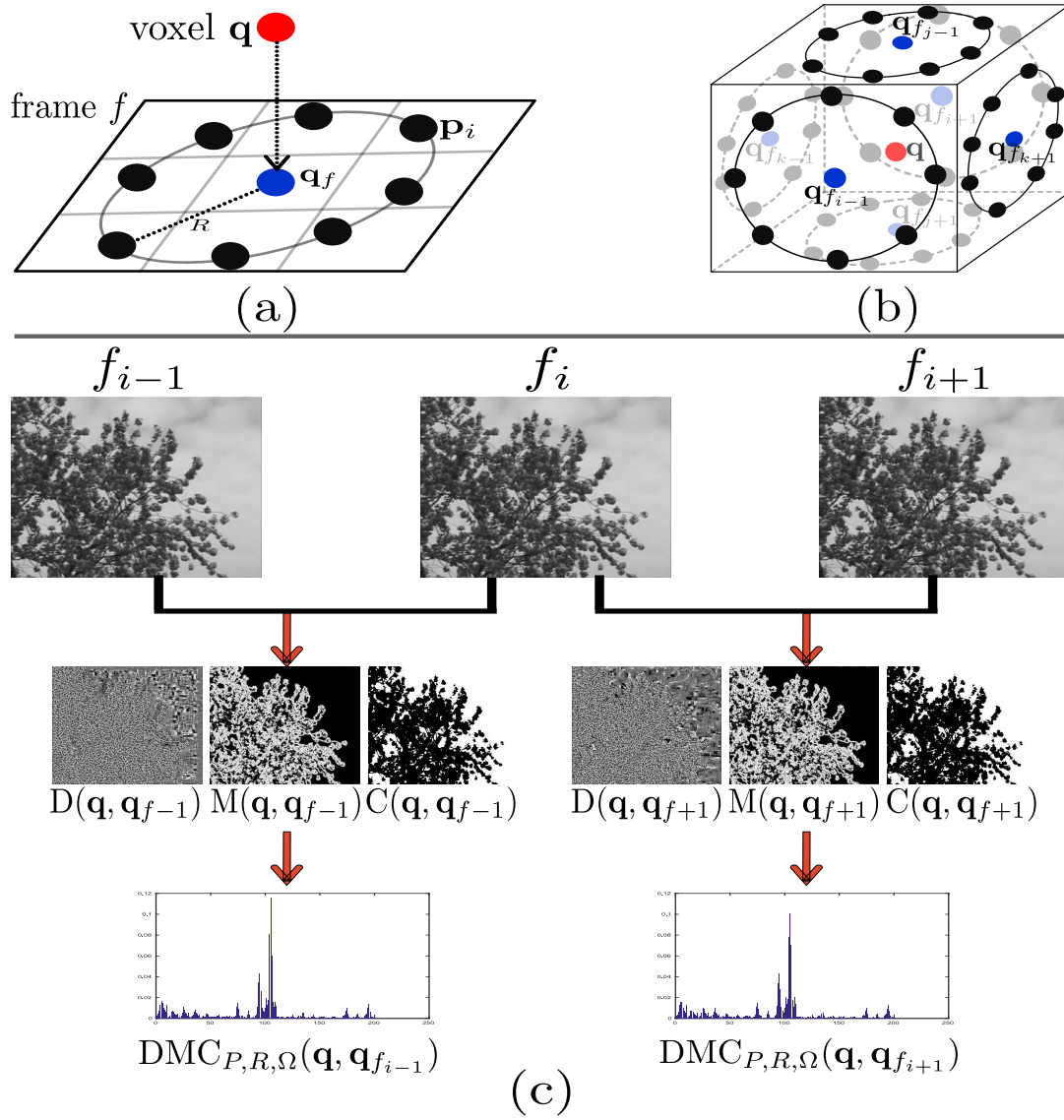


Figure 3.7: Computing parts of our framework. (a): A model of encoding feature for a voxel  $q$  (in red) based on its central symmetry voxel  $q_f$  (in blue) on plane image  $f$ . (b): A graphical illustration of LRP construction at voxel  $q$ . (c): A calculation of an integrated histogram  $DMC(\cdot)$  for voxels  $\{q \in f_i\}$  along with their symmetry points in images  $f_{i-1}$  and  $f_{i+1}$  of plane  $XY$  in a video.

- LRP structures a voxel in consideration of rich spatio-temporal relationships extracted from 6 sides of the rubik cube (see Figure 3.7(b)) while other LBP-based variants mostly based on three orthogonal planes for DT representation [92, J1].
- Discriminative information of a center voxel is embedded into encoding side patterns against near-uniform regions.
- Based on a block shape, LRP is more suitable for encoding DT videos than LBP-based variants which are separately applied to still images of the planes in videos.
- By addressing previous and posterior plane-images, LRP can capture changes of a voxel in global spatio-temporal appearances. In the meanwhile, VLBP for structuring temporal appearances in plane  $XY$ , and LBP-TOP for addressing local orthogonal patterns [14].

### 3.6 Completed Hierarchical Local Patterns (CHILOP)

In this section, we propose a novel, efficient operator, named CHILOP, to adequately capture local relationships in multi-layer hierarchical supporting regions against problems of sensitivity to noise and near uniform regions (see Section 3.6.1). It should be noted that our CHILOP is a generalized concept

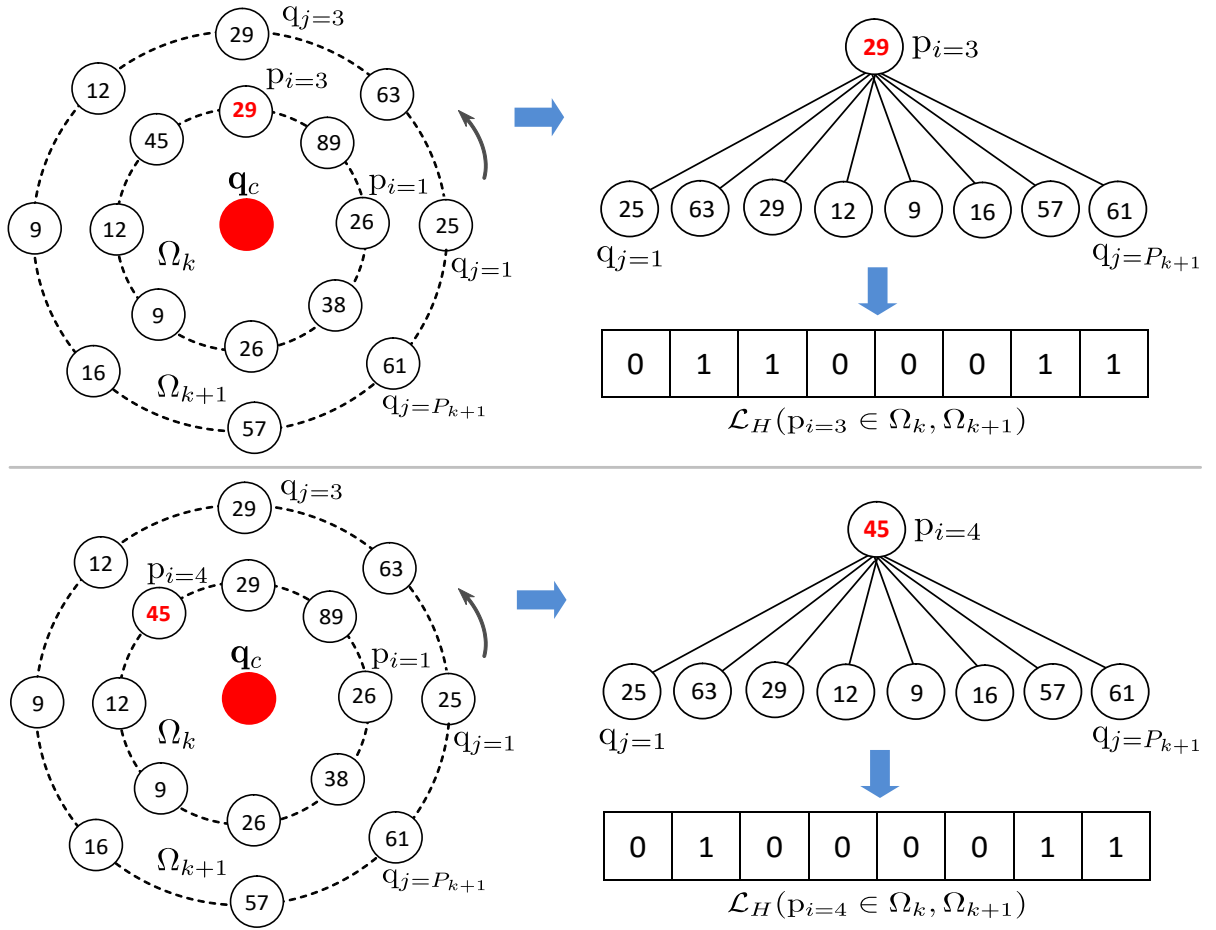


Figure 3.8: An instance of structuring two  $\mathcal{L}_H$  patterns of  $p_{i=3}, p_{i=4} \in \Omega_k$  based on  $\{q_j\}_{j=1}^{P_{k+1}}$  of  $\Omega_{k+1}$ , in which  $\Omega_k = (8, 1)$ , and  $\Omega_{k+1} = (8, 2)$  are two adjacent LBP-based regions, i.e.,  $P_k = P_{k+1} = 8$  neighbors sampled by  $R_k = 1$ , and  $R_{k+1} = 2$ .

of CLBP [3], one of the most popular LBP-based variants for textural image description. Furthermore, it is possible to take advantage of the CHILOP approach for other directional and non-directional LBP-based methods in order to improve their performances, such as CLBC [82], LDP-based [30, J5], LVP-based [100, J2], LRP [J4], etc. Due to being a local encoding operator, our CHILOP may be limited by influences of environmental changes and illumination. To deal with those, we introduce an effective framework in which CHILOP is exploited on Gaussian-filtered images in order to make the output patterns more discriminative against those negative impacts (see Section 6.4.1). Hereafter, we express above processes in detail.

### 3.6.1 Construction of CHILOP

As mentioned in Section 2.7, with simple computation, local-feature-based methods have obtained promising performances in analyzing shapes and motions of DTs in videos. In spite of that, their executions have been restricted due to the negative impacts of changes of environmental elements, illumination and noise, near uniform regional problems. In this section, we propose a novel operator that is able to efficiently capture local relationships with more robustness against those problems. To this end, in our prior work [C3], a pairwise of adjacent supporting areas is exploited to capture Hierarchical Local Patterns (HILOP) for DT representation. However, it has achieved moderate results due to lack of forceful informative appearances. Addressing those crucial omissions, we make an important extension of HILOP in consideration of a completely analyzing context in order to structure Completed HILOP Patterns (i.e., CHILOP) with more discrimination against problems of sensitivity to noise, near uniform regions. Sim-

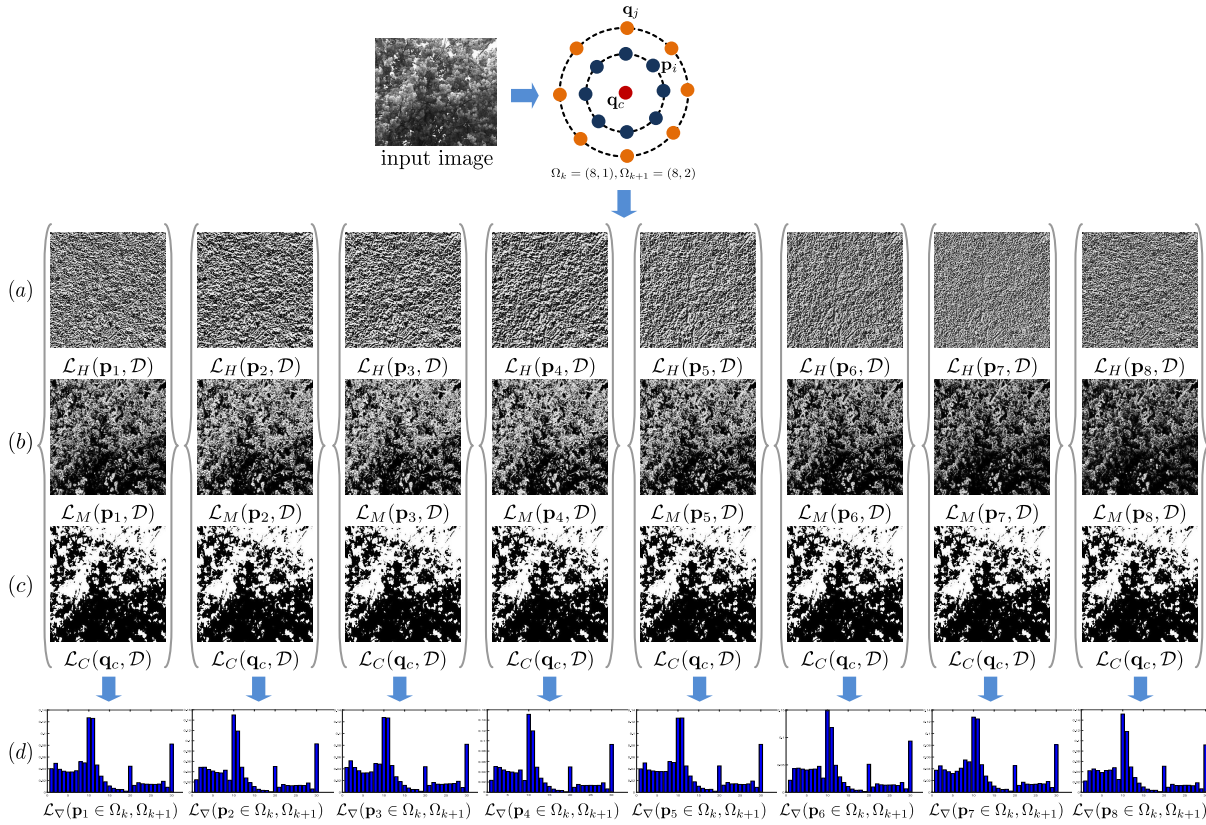


Figure 3.9: An example of CHILOP encoding. Therein,  $\mathcal{L}_H$ ,  $\mathcal{L}_M$ , and  $\mathcal{L}_C$  patterns for  $\forall \mathbf{p}_i \in \Omega_k$  (in dark blue) are corresponding to lines of (a), (b), and (c), which are structured by exploiting two hierarchical LBP-based supporting regions  $\mathcal{D} = \{\Omega_k = (8, 1), \Omega_{k+1} = (8, 2)\}$ . The corresponding  $\mathcal{L}_\nabla$  histograms, i.e., line (d), are formed by using an integration of  $\nabla = \{H\_M/C\}$ .

ilar to CLBP [3], our proposed operator CHILOP also consists of three complementary components, but in a generally novel concept of investigating local relationships in which CLBP-based patterns can be deduced from CHILOP in a specific condition of degeneration (see Section 3.6.2). In other words, our CHILOP is more general than CLBP in local encoding. Hereunder, we express the proposed foundation of our CHILOP in detail.

Let  $\Omega_k = \{\mathbf{p}_i\}_{i=1}^{P_k}$  and  $\Omega_{k+1} = \{\mathbf{q}_j\}_{j=1}^{P_{k+1}}$ , ( $k, P_k, P_{k+1} \in \mathbb{Z}^+$ ), be two adjacent hierarchies of supporting regions of a pixel  $\mathbf{q}_c$  in an image  $\mathcal{I}$ , so that  $\Omega_k \cap \Omega_{k+1} = \emptyset$  and  $P_k \leq P_{k+1}$ , in which  $P_k$ ,  $P_{k+1}$  denote numbers of  $\mathbf{q}_c$ 's neighbors determined in hierarchical regions  $\Omega_k$ ,  $\Omega_{k+1}$  respectively. For each neighbor  $\mathbf{p}_i \in \Omega_k$ , three kinds of hierarchical local relationships are addressed as follows.

First,  $\mathbf{p}_i$ 's hierarchic pattern  $\mathcal{L}_H$  is encoded as a binary string of  $P_{k+1}$  bits by considering the difference of  $\mathbf{p}_i$ 's gray-value with that of all  $\mathbf{q}_j \in \Omega_{k+1}$  as follows.

$$\mathcal{L}_H(\mathbf{p}_i \in \Omega_k, \Omega_{k+1}) = \sum_{j=1}^{P_{k+1}} \xi(\mathcal{I}(\mathbf{q}_j) - \mathcal{I}(\mathbf{p}_i)) \times 2^{j-1} \quad (3.36)$$

in which  $\mathcal{I}(\cdot)$  returns the gray-value of a pixel,  $\xi(\cdot)$  is defined in Equation (2.5). It can be seen that when  $\Omega_k = \{\mathbf{q}_c\}$ , and  $P_{k+1} > 1$ ,  $\mathcal{L}_H$  is equivalent to the basic LBP [81]. Figure 3.8 particularly shows an instance of this computation for  $\mathbf{p}_{i=3}, \mathbf{p}_{i=4} \in \Omega_k$  using two adjacent LBP-based supporting regions  $\Omega_k = (8, 1)$  and  $\Omega_{k+1} = (8, 2)$ . In the meanwhile, Figure 3.9(a) illustrates visual samples of  $\mathcal{L}_H$  patterns that are encoded for  $\forall \mathbf{p}_i \in \Omega_k$  in an input image. In a former work [C3], we used the  $\mathcal{L}_H$  patterns for video description and obtained good performance of DT recognition. This leads to a motivation for a completed model of hierarchical features by defining two complementary components  $\mathcal{L}_M$  and  $\mathcal{L}_C$  as presented below.

Second, in consideration of the intensity of hierarchic properties on the entire image  $\mathcal{I}$ , the hierarchical magnitude information  $\mathcal{L}_M$  of a pixel  $\mathbf{p}_i \in \Omega_k$  is formulated as

$$\mathcal{L}_M(\mathbf{p}_i \in \Omega_k, \Omega_{k+1}) = \sum_{j=1}^{P_{k+1}} \xi(|\mathcal{I}(\mathbf{q}_j) - \mathcal{I}(\mathbf{p}_i)| - \tilde{\mathbf{m}}_i) \times 2^{j-1} \quad (3.37)$$

where  $\tilde{\mathbf{m}}_i$  is the average of gray-value differences of  $\mathbf{p}_i$  versus all  $\mathbf{q}_j \in \Omega_{k+1}$  computed on the whole image  $\mathcal{I}$  as

$$\tilde{\mathbf{m}}_i = \frac{1}{\mathcal{N} \times P_{k+1}} \sum_{\mathbf{p}_i \in \mathcal{I}} \sum_{j=1}^{P_{k+1}} (|\mathcal{I}(\mathbf{q}_j) - \mathcal{I}(\mathbf{p}_i)|) \quad (3.38)$$

in which  $\mathcal{N}$  denotes the total of  $\mathbf{p}_i$  pixels determined by the center  $\mathbf{q}_c$  for the whole image  $\mathcal{I}$  (see Figure 3.9(b) for a visual look of encoding  $\mathcal{L}_M$  features).

Third, the remain is to measure the gray-scale central difference  $\mathcal{L}_C$  in which a consolidation of the gray-level average  $\tilde{\gamma}_{\Omega_k, \Omega_{k+1}}$  of two hierarchical regions along with the center pixel  $\mathbf{q}_c$  is addressed as a threshold in order to compare with  $\tilde{\mathbf{c}}_{\mathcal{I}}$ , the average of all pixels in image  $\mathcal{I}$ . Figure 3.9(c) shows an instance of the  $\mathcal{L}_C$  encoding in which its achieved patterns are used for computing jointed histograms subject to the computation of  $\{\mathbf{p}_i\}$  in image  $\mathcal{I}$ .

$$\mathcal{L}_C(\mathbf{q}_c, \Omega_k, \Omega_{k+1}) = \xi\left(\frac{\tilde{\gamma}_{\Omega_k, \Omega_{k+1}} + \mathcal{I}(\mathbf{q}_c)}{2} - \tilde{\mathbf{c}}_{\mathcal{I}}\right) \quad (3.39)$$

in which  $\tilde{\gamma}_{\Omega_k, \Omega_{k+1}}$  is defined as

$$\tilde{\gamma}_{\Omega_k, \Omega_{k+1}} = \frac{1}{P_k + P_{k+1}} \left( \sum_{\mathbf{p}_i \in \Omega_k} \mathcal{I}(\mathbf{p}_i) + \sum_{\mathbf{q}_j \in \Omega_{k+1}} \mathcal{I}(\mathbf{q}_j) \right) \quad (3.40)$$

Similar to CLBP [3], the proposed patterns  $\mathcal{L}_H$ ,  $\mathcal{L}_M$ , and  $\mathcal{L}_C$  can be also considered as complementary components. Therefore, they can be integrated in several ways  $\nabla$  to enhance the discrimination power, such as  $\nabla = \{H/M/C, H\_M/C, \text{etc.}\}$ , where the signal of “ $H/M/C$ ” denotes a 3D jointing histogram of  $\mathcal{L}_H$ ,  $\mathcal{L}_M$ , and  $\mathcal{L}_C$ , while the signal of “ $H\_M/C$ ” means that a histogram of  $\mathcal{L}_H$  is concatenated with a 2D jointing of  $\mathcal{L}_M$  and  $\mathcal{L}_C$  (see Figure 3.9(d) for an example of calculating a “ $H\_M/C$ ” histogram). From now on,  $\mathcal{L}_{\nabla}$  is denoted as an integration of these complementary components in general.

By taking all neighbors  $\mathbf{p}_i$  of  $\Omega_k$  into account the  $\mathcal{L}_{\nabla}$  encoding, two-hierarchical feature of  $\mathbf{q}_c$  is formed in completed consideration of a pairwise of hierarchical supporting areas  $\Omega_k$  and  $\Omega_{k+1}$  as follows.

$$\Gamma_{\nabla, \Omega_k, \Omega_{k+1}}(\mathbf{q}_c) = [\mathcal{L}_{\nabla}(\mathbf{p}_i \in \Omega_k, \Omega_{k+1})]_{i=1}^{P_k} \quad (3.41)$$

It should be noted that structuring  $\mathcal{L}_H$  patterns of  $\Gamma(\cdot)$  is absolutely different from capturing difference-based patterns introduced in [88], i.e., RD-LBP and AD-LBP. More specifically, in our proposal, all of  $\mathbf{q}_j \in \Omega_{k+1}$  are thresholded with each of  $\mathbf{p}_i \in \Omega_k$  to be able to figure out  $P_k$  patterns. In contrast to that, RD-LBP [88] is formed by comparing a pairwise of  $(\mathbf{q}_j, \mathbf{p}_j)$  in parallel to achieve only one pattern, while AD-LBP [88] is computed by addressing the differences of pixels in the same regions.

In order to forcefully enrich discriminative information, we address the function  $\Gamma(\cdot)$  for multi-regional analysis to capture more useful properties in the further regions. According to that, let  $\mathcal{D} = \{\Omega_1, \Omega_2, \dots, \Omega_l\}$  be a set of adjacent hierarchical areas extracted from a given pixel  $\mathbf{q}_c \in \mathcal{I}$  so that each pairwise of which is separative, i.e.,  $\Omega_k \cap \Omega_{k+1} = \emptyset, \forall k$ . Completed Hierarchical Local Pattern (CHILOP)<sup>2</sup> of  $\mathbf{q}_c$  is structured as follows.

$$\text{CHILOP}_{\nabla, \mathcal{D}}(\mathbf{q}_c) = [\Gamma_{\nabla, \Omega_k, \Omega_{k+1}}(\mathbf{q}_c)]_{k=1}^l \quad (3.42)$$

<sup>2</sup>A simple MATLAB code for structuring CHILOP patterns is available at <http://tpnguyen.univ-tln.fr/download/MATCodeCHILOP>

### 3.6.2 A particular degeneration of CHILOP into CLBP

It can be realized that our CHILOP operator in a single scale, proposed in Section 3.6.1, is more general in a completed context of local encoding than CLBP [3]. Indeed, when  $\Omega_k = \{\mathbf{q}_c\}$ , and  $P_{k+1} > 1$ , it can be simply proved from Equations (3.36) and (2.4) that the component  $\mathcal{L}_H$  is identical to the basic LBP [81], i.e., the  $\text{CLBP}_S$  component of CLBP. Similarly, the function  $\mathcal{L}_M$  is degenerated into  $\text{CLBP}_M$  due to a replacement of  $\Omega_k = \{\mathbf{q}_c\}$  in Equation (3.37) to deduce (2.10). In the meanwhile, the component  $\mathcal{L}_C$  is embedded more the gray-scale information of local points in hierarchical supporting regions, but in general, it can be considered as a equivalent function of  $\text{CLBP}_C$  (see Equations (3.39) and (2.12)). As the result of those, beside inheriting the advantages of CLBP, our CHILOP features are enriched more hierarchical structures to improve the discrimination against problems of sensitivity to noise and near uniform patterns (see Table 6.3 and Figure 6.6 for a practical comparison of their performances).

### 3.6.3 Beneficial properties of CHILOP operator

Based on the generally novel concept presented in Section 3.6.1, our CHILOP has the following beneficial properties to enhance the discrimination power in comparison with our prior operator HILOP [C3], as well as with CLBP [3], and other LBP-based variants:

- Instead of considering the difference of a center pixel and its local neighbors as conducted in LBP-based variants, CHILOP addresses a pairwise of adjacent supporting regions in order to capture hierarchical characteristics with more robustness against problems of noise and near uniform patterns.
- Beside structuring hierarchical features by considering relationships of two adjacent regional hierarchies, CHILOP operator is able to forcefully capture more informative discrimination thanks to two more complementary components  $\mathcal{L}_M$  and  $\mathcal{L}_C$ . In the meanwhile, just one kind of hierarchical features is exploited in HILOP [C3] (see Figure 6.7 for a particular comparison of their performances).
- The concept of hierarchical structures in CHILOP can be extended to other directional/non-directional LBP-based variants in order to enhance their executions, e.g., CLBC [82], LDP-based [30, J5], LVP-based [100, J2], LRP [J4], etc.
- In general, the complexity of our proposed operator has the same order as that of CLBP. Indeed, let  $\Theta$  be the complexity of CLBP, CHILOP's is about  $\Theta \times P_k$ , in which  $P_k$  denotes the number of local neighbors of the inner hierarchical area involved with, i.e.,  $\Omega_k$ .

## 3.7 Summary

In this chapter, we have proposed five types of local-based descriptors for DT encoding: CAIP [S4], xLDP [J5], xLVP [J2], LRP [J4], and CHILOP [S2], which are robust against the conventional problems of sensitivity to noise and near uniform patterns. Moreover, their computation is simple due to being inherited from the advantage of LBP-based variants. Those operators are then exploited to encode local DT features from different aspects of video analysis as presented in the following Chapters 5, 4, 6, in which experimental results for DT recognition task have substantiated that our proposals are the same order of computational cost compared to their originals and the popular one CLBP [3], but much better performance in DT representation, e.g., our xLDP versus its original LDP [30], xLVP versus LVP [100], CAIP versus CLBP [3], etc.

---

# CHAPTER 4

---

## REPRESENTATION BASED ON DENSE TRAJECTORIES

### Contents

<b>4.1</b>	<b>Introduction</b>	<b>37</b>
<b>4.2</b>	<b>Dense trajectories</b>	<b>38</b>
<b>4.3</b>	<b>Beneficial properties of dense trajectories</b>	<b>38</b>
4.3.1	Directional features of a beam trajectory	38
4.3.2	Spatio-temporal features of motion points	39
<b>4.4</b>	<b>Directional dense trajectory patterns for DT representation</b>	<b>40</b>
4.4.1	Proposed DDTP descriptor	40
4.4.2	Computational complexity of DDTP descriptor	42
<b>4.5</b>	<b>Experiments and evaluations</b>	<b>44</b>
4.5.1	Experimental settings	44
4.5.2	Experimental results	46
4.5.3	Global discussion	51
<b>4.6</b>	<b>Summary</b>	<b>52</b>

### 4.1 Introduction

In this chapter, we propose an efficient framework for DT representation to exploit spatio-temporal features of their dense trajectories extracted from a given video. In general, the proposed framework consists of three stages as follows. Firstly, motion points and their paths in a video extracted by using an extracting tool [104]. Secondly, our proposed xLVP operator, a crucial completed model of LVP [100] operator as presented in Section 3.4, is taken into account to encode the obtained trajectories. Thirdly, 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 output 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 [C4, J2].

- Exploiting LVP to capture directional features of beam trajectories [C4].
- Addressing xLVP for both beam trajectories and spatio-temporal features of motion points [J2]. It means that the profitable characteristics of optical-flow-based and local-feature-based methods are combined for DT representation in more effect.

## 4.2 Dense trajectories

Wang *et al.* [104] 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 [105] 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

$$\mathbf{q}_{f+1} = \mathbf{q}_f + (M \times \omega_f)|_{(\bar{x}_f, \bar{y}_f)} \quad (4.1)$$

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  $3 \times 3$  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<sup>1</sup> as a tool to extract motion paths of dynamic features for DT description.

## 4.3 Beneficial properties of dense trajectories

Dense trajectories, introduced by Wang *et al.* [104], 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 [104, 106], object segmentation [107], 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.

### 4.3.1 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 Figure 4.1 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  $\Phi$ . 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.

$$\text{DBT}_{L, \Phi, d}(t) = \left[ \sum_{i=1}^{L+1} H_{\mathbf{q}_i}(\text{xLVP}_{P, R, \Phi, d}(\mathbf{q}_{i, f_i})), \biguplus_{\mathbf{p}_j \in B} \left[ \sum_{i=1}^{L+1} H_{\mathbf{p}_j}(\text{xLVP}_{P, R, \Phi, d}(\mathbf{p}_{j, f_i})) \right] \right] \quad (4.2)$$

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  $\Phi$ ;  $\mathbf{p}_j$

<sup>1</sup>[http://lear.inrialpes.fr/people/wang/dense\\_trajectories](http://lear.inrialpes.fr/people/wang/dense_trajectories)



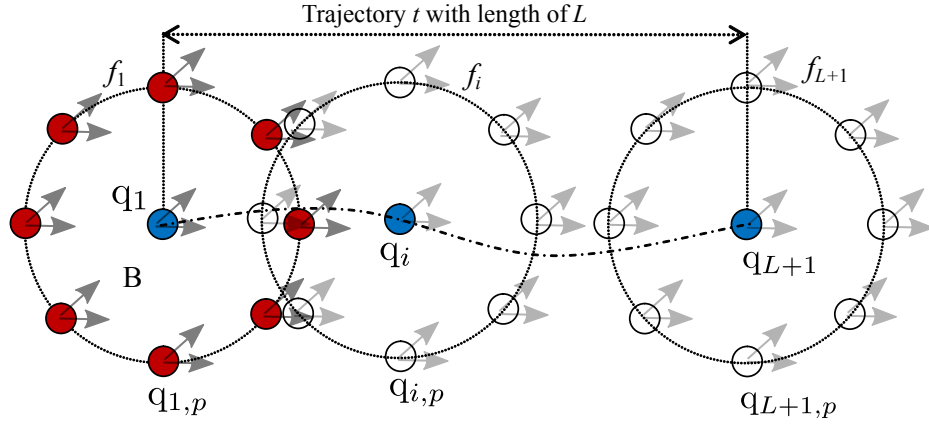


Figure 4.1: A general model for encoding DBT patterns in which dense trajectory  $t$  with length of  $L$  is structured by  $L + 1$  blue motion points located in consecutive frames along with their neighbors in different colors situated in a vicinity  $B = \{8, 1\}$ .

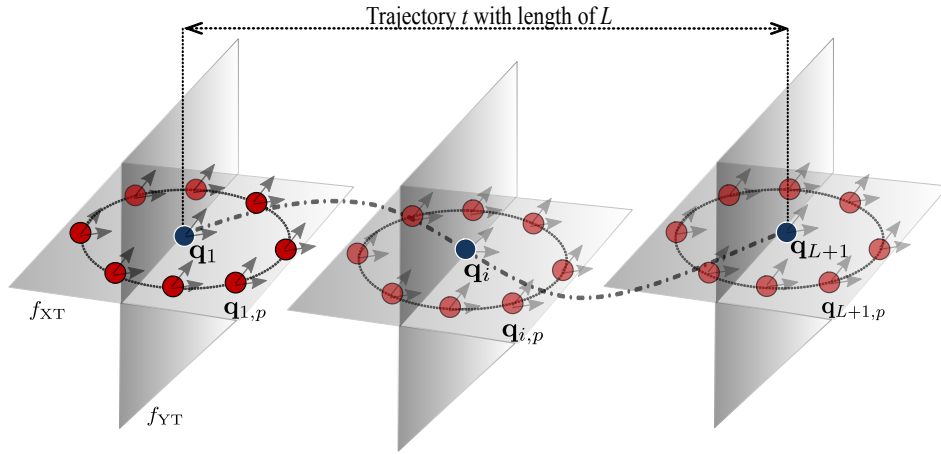


Figure 4.2: A typical TMP model in which directional temporal information of motion points (in blue) are encoded along their trajectory  $t$  with length of  $L$  by exploiting directional relations of those with their local neighbors  $P = 8$  (in red) sampled by a circle of radius  $R = 1$  on XT and YT planes.

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;  $\uplus$  denotes a concatenating function for the obtained histograms  $H_{\mathbf{p}_j}(\cdot)$ .

### 4.3.2 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 [14, 92, C1], 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 Figure 4.2, in order to form directional structures of temporal motion points (TMP) as

$$\text{TMP}_{L,\Phi,d}(t) = \left[ H_{XT}(\text{xLVP}_{P,R,\Phi,d}(\mathbf{q}_i)), H_{YT}(\text{xLVP}_{P,R,\Phi,d}(\mathbf{q}_i)) \right]_{i=1}^{L+1} \quad (4.3)$$



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  $\Phi$ ;  $H_{XT}(\cdot)$  and  $H_{YT}(\cdot)$  are histograms of motion point  $\mathbf{q}_i$  calculated for the corresponding planes.

#### 4.4 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 Figure 4.3. According to that, dense trajectories are extracted at first using the tool introduced in [104]. We then apply our extended operator  $\text{xLVP}$  on those to capture their directional motion cues through encoding patterns of directional beam trajectories, as proposed in Section 4.3.1. 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 4.3.2. 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.

##### 4.4.1 Proposed DDTP descriptor

Let  $\mathcal{T} = \{t_1, t_2, \dots, t_n\}$  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  $\Phi$  using the completed model  $\text{xLVP}$ . Parallel to this encoding, TMP patterns are also structured by addressing  $\text{xLVP}$  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

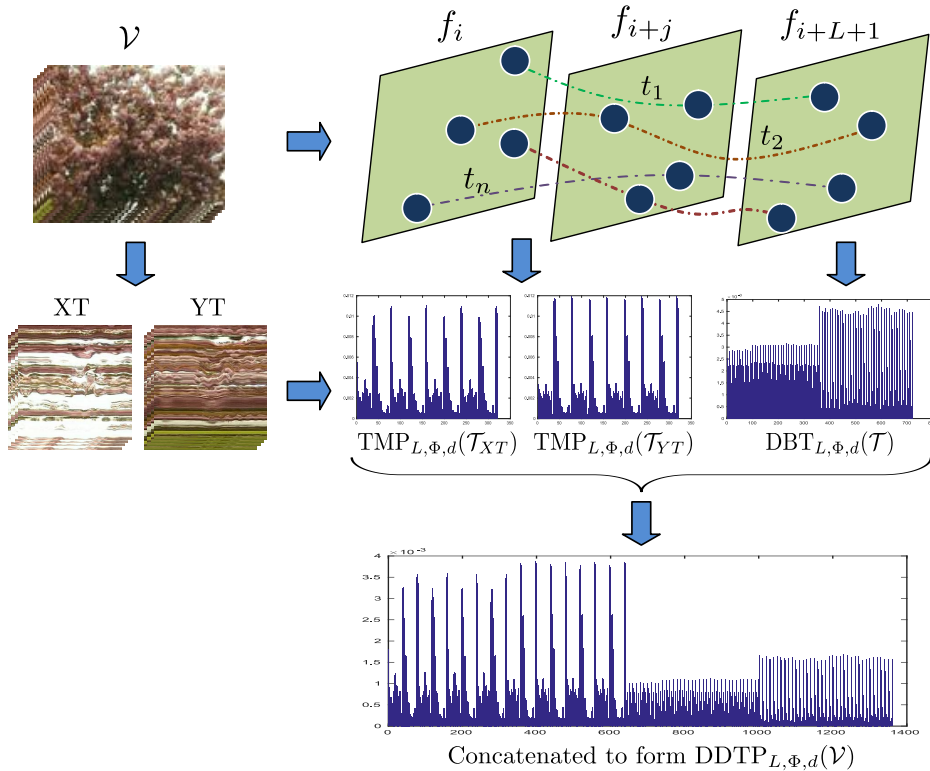


Figure 4.3: An effective framework for DT representation based on dense trajectories extracted from a video  $\mathcal{V}$ .

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

$$\text{DDTP}_{L,\Phi,d}(\mathcal{V}) = \frac{1}{|\mathcal{T}|} \sum_{t_i \in \mathcal{T}} [\text{DBT}_{L,\Phi,d}(t_i), \text{TMP}_{L,\Phi,d}(t_i)] \quad (4.4)$$

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,  $\text{DDTP}_{D-M/C}$  indicates that it is structured by  $\text{xLVP} = \text{LVP}_{D-M/C}$  (see Section 3.4.3 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.,  $\text{LVP}_D$ ), is concerned by addressing the same implementation above.

$$\text{DDTP-B}_{L,\Phi,d}(\mathcal{V}) = \frac{1}{|\mathcal{T}|} \sum_{t_i \in \mathcal{T}} [\text{DBT-B}_{L,\Phi,d}(t_i), \text{TMP-B}_{L,\Phi,d}(t_i)] \quad (4.5)$$

where DBT-B and TMP-B are respectively computed as similarly as in Equations (4.2) and (4.3) but only  $\text{LVP}_D$  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.

$$\text{xLVP-TOP}_{\Phi,d}(\mathcal{V}) = [\text{xLVP}_{P,R,\Phi,d}(\mathcal{V}_{XY}), \text{xLVP}_{P,R,\Phi,d}(\mathcal{V}_{XT}), \text{xLVP}_{P,R,\Phi,d}(\mathcal{V}_{YT})] \quad (4.6)$$

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 [100] on planes of  $\{XY, XT, YT\}$  as

$$\text{LVP-TOP}_{\Phi,d}(\mathcal{V}) = [\text{LVP}_{P,R,\Phi,d}(\mathcal{V}_{XY}), \text{LVP}_{P,R,\Phi,d}(\mathcal{V}_{XT}), \text{LVP}_{P,R,\Phi,d}(\mathcal{V}_{YT})] \quad (4.7)$$

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 3.4.3) 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  $\text{LVP}_D$  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 [108] 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 4.5 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.

#### 4.4.2 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 [104]. The computational cost of this extraction  $Q_{\mathcal{T}}$  can be referred to [104] 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  $\text{xLVP}_{XY}$  (the two other components have the same 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 (3.21) and (3.22) in Section 3.4.2 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 3.4.3, our xLVP consists of three complementary components:  $\text{LVP}_D$ ,  $\text{LVP}_M$ , and  $\text{LVP}_C$ . Based on Equations (3.23), (3.25), and (3.27), 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  $Q_{\text{LVP}_D}$ ,  $Q_{\text{LVP}_M}$ , and  $Q_{\text{LVP}_C}$ , i.e.,  $\mathcal{O}(P \times \mathcal{H} \times \mathcal{W})$ . Therefore, the complexity for extraction of  $\text{xLVP}_{XY}$  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  $\Phi$ .
- *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.

$$Q_{\text{DDTP}} = Q_{\mathcal{T}} + Q_{\text{vMP}} + Q_{\text{xLVP}} + Q_{\text{DBT}} + Q_{\text{TMP}} \quad (4.8)$$

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 4.5.1) are much smaller than the others, they can be also ignored. Consequently,  $Q_{\text{DDTP}}$  could be approximated by Equation (4.9), 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  $P = 16$ .

$$Q_{\text{DDTP}} \approx \max(Q_{\text{vMP}}, Q_{\text{xLVP}}, Q_{\text{DBT}}, Q_{\text{TMP}}) \approx \mathcal{O}(P \times \mathcal{H} \times \mathcal{W} \times F) \quad (4.9)$$

---

**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  $\mathcal{H} \times \mathcal{W} \times F$ , 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        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     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     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     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 vMP( $\mathbf{q}$ ) > 0 then
30         Structuring DBT and TMP features based on xLVP $_{XY}$ , xLVP $_{XT}$ , 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];
    
```

---

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 (4.9) 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 6.32 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 6.32 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 4.7 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 4.1: Comparing processing time of encoding two videos in UCLA.

Sample video	Resolution	$L$	Level of turbulence	Number of trajectories	Runtime (s)
	$110 \times 160 \times 75$ (original)	3	A single candle flame	3674	$\approx 8.7$
	$48 \times 48 \times 75$ (cropped)	3	A single candle flame	1507	$\approx 2.6$
	$110 \times 160 \times 75$ (original)	3	All leaf vibrations	25562	$\approx 35.3$
	$48 \times 48 \times 75$ (cropped)	3	All leaf vibrations	2134	$\approx 3.1$

## 4.5 Experiments and evaluations

In this section, comprehensive evaluations of the proposed framework on the benchmark DT datasets (i.e., UCLA [5], DynTex [54], and DynTex++ [55]) are specifically expressed by following experimental protocols and parameter settings for implementation. The obtained recognition rates are then evaluated in comparison with those of the state-of-the-art methods.

### 4.5.1 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 [104], 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 4.4 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 4.4(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  $min\_distance = 1$ .

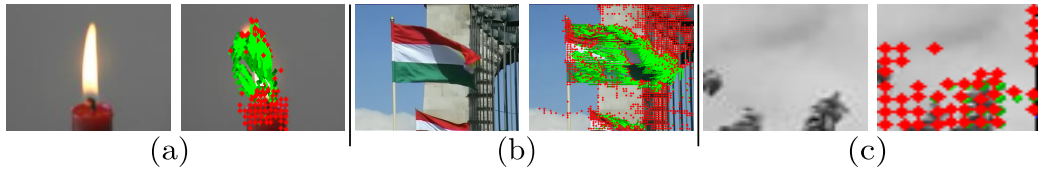


Figure 4.4: Samples (a), (b), (c) of dense trajectories extracted from the corresponding videos in UCLA, DynTex, and DynTex++ datasets respectively in which green lines show paths of motion points through the consecutive frames.

*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 Figure 4.1 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  $\Omega$ . For different types of DDTP descriptor, structured subject to integrating complemented components in xLVP operator, we address three descriptors for experiments

Table 4.2: A comparison of various dimensions of LBP-based descriptors.

Method	Dimensions	$P = 8$	$P = 16$	$P = 24$
LBP-TOP <sup>riu2</sup> [14]	$3(P(P-1)+3)$	177	729	1665
VLBP [14]	$2^{3P+2}$	-	-	-
CVLBP [91]	$3 \times 2^{3P+2}$	-	-	-
HLBP [92]	$6 \times 2^P$	1536	-	-
CLSP-TOP <sup>riu2</sup> [C1]	$6(P+2)^2$	600	1944	4,056
WLBPC [109]	$6 \times 2^P$	1536	-	-
MEWLSP [95]	$6 \times 2^P$	1536	-	-
CVLBC [90]	$2(3P+3)^2$	1458	5202	11125
CSAP-TOP <sup>riu2</sup> [J1]	$12(P+2)^2$	1200	3888	8112
FDT <sup>u2</sup> [C4]	$216P((P-1)+3)$	12744	-	-
FD-MAP <sup>u2</sup> <sub>L=2</sub> [C4]	$216P((P-1)+3)+16$	12760	-	-
<b>DDTP<sup>riu2</sup><sub>D_M</sub></b>	$8(P+7)(P+2)$	1200	3312	6448
<b>DDTP<sup>riu2</sup><sub>D_M.C</sub></b>	$8(P+7)(P+3)$	1320	3496	6696
<b>DDTP<sup>riu2</sup><sub>D_M/C</sub></b>	$12(P+7)(P+2)$	1800	4968	9672
<b>DDTP-B<sup>riu2</sup></b>	$4(P+7)(P+2)$	600	1656	3224
<b>xLVP-TOP<sup>riu2</sup><sub>D_M</sub></b>	$24(P+2)$	240	432	624
<b>xLVP-TOP<sup>riu2</sup><sub>D_M.C</sub></b>	$24(P+3)$	264	456	648
<b>xLVP-TOP<sup>riu2</sup><sub>D_M/C</sub></b>	$36(P+2)$	360	648	936
<b>LVP-TOP<sup>riu2</sup></b>	$12(P+2)$	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 4.3: Results (%) on UCLA exploiting DDTP and DDTP-B descriptors.

Scheme	50-LOO				50-4fold				9-class				8-class			
$\{(P, R)\}_L^{riu2}$	$D_M$	$D_{M.C}$	$D_{M/C}$	$\sim B$	$D_M$	$D_{M.C}$	$D_{M/C}$	$\sim B$	$D_M$	$D_{M.C}$	$D_{M/C}$	$\sim B$	$D_M$	$D_{M.C}$	$D_{M/C}$	$\sim B$
$\{(8, 1)\}_{L=2}^{riu2}$	97.00	97.50	99.00	98.50	94.00	96.00	99.00	98.00	98.60	98.10	98.10	<b>97.90</b>	96.20	96.85	97.28	94.24
$\{(16, 2)\}_{L=2}^{riu2}$	99.50	<b>100</b>	<b>99.50</b>	95.00	<b>100</b>	<b>100</b>	99.50	94.50	97.40	96.60	97.90	95.80	96.09	95.76	96.43	95.33
$\{(8, 1), (16, 2)\}_{L=2}^{riu2}$	<b>100</b>	<b>100</b>	99.00	<b>99.50</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>99.00</b>	98.35	98.25	98.50	97.85	97.28	96.96	97.50	<b>97.61</b>
$\{(8, 1)\}_{L=3}^{riu2}$	94.00	94.00	99.00	98.50	95.50	95.50	99.00	98.50	98.10	<b>98.55</b>	98.30	97.45	96.52	97.17	95.33	95.22
$\{(16, 2)\}_{L=3}^{riu2}$	<b>100</b>	<b>100</b>	<b>99.50</b>	96.50	<b>100</b>	<b>100</b>	99.50	98.50	97.50	97.60	96.65	95.90	97.07	<b>98.15</b>	96.74	93.15
$\{(8, 1), (16, 2)\}_{L=3}^{riu2}$	<b>100</b>	<b>100</b>	99.00	<b>99.50</b>	<b>100</b>	<b>100</b>	99.50	98.50	<b>98.60</b>	97.95	<b>98.75</b>	96.15	<b>97.72</b>	98.04	<b>98.04</b>	96.30

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 xLVP.  $\sim B$  means the DDTP-B descriptor.

on DT classification, i.e., DDTP<sub>D\_M</sub>, DDTP<sub>D\_M/C</sub>, and DDTP<sub>D\_M.C</sub> (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<sub>D\_M</sub> of  $24l_{riu2/u2}$  bins, xLVP-TOP<sub>D\_M/C</sub> of  $36l_{riu2/u2}$ , and xLVP-TOP<sub>D\_M.C</sub> 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 6.2 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.

Table 4.4: Rates (%) on DynTex using DDTP and DDTP-B descriptors.

Scheme	DynTex35				Alpha				Beta				Gamma			
	$D_M$	$D_{M,C}$	$D_{M/C}$	$\sim B$	$D_M$	$D_{M,C}$	$D_{M/C}$	$\sim B$	$D_M$	$D_{M,C}$	$D_{M/C}$	$\sim B$	$D_M$	$D_{M,C}$	$D_{M/C}$	$\sim B$
$\{(P, R)\}_{L=2}^{riu2}$	98.57	98.29	98.00	98.00	<b>98.33</b>	<b>98.33</b>	<b>98.33</b>	<b>98.33</b>	90.12	91.36	93.21	87.04	88.64	87.88	90.53	88.64
$\{(16, 2)\}_{L=2}^{riu2}$	<b>99.43</b>	<b>99.43</b>	<b>99.43</b>	<b>100</b>	96.67	96.67	96.67	93.33	91.36	90.74	91.98	87.65	90.53	<b>91.67</b>	89.77	87.88
$\{(8, 1), (16, 2)\}_{L=2}^{riu2}$	<b>99.43</b>	98.86	99.14	99.14	96.67	96.67	96.67	<b>98.33</b>	91.36	<b>91.98</b>	92.59	88.27	<b>92.80</b>	<b>91.67</b>	<b>91.29</b>	87.88
$\{(8, 1)\}_{L=3}^{riu2}$	98.00	98.00	98.57	98.29	<b>98.33</b>	<b>98.33</b>	<b>98.33</b>	<b>98.33</b>	89.51	91.36	<b>94.44</b>	<b>88.89</b>	88.26	89.02	90.15	<b>89.77</b>
$\{(16, 2)\}_{L=3}^{riu2}$	<b>99.43</b>	<b>99.43</b>	<b>99.71</b>	<b>100</b>	96.67	96.67	96.67	93.33	91.36	<b>91.98</b>	93.21	<b>88.89</b>	90.53	90.53	89.77	85.98
$\{(8, 1), (16, 2)\}_{L=3}^{riu2}$	<b>99.43</b>	<b>99.43</b>	<b>99.71</b>	98.86	96.67	96.67	96.67	<b>98.33</b>	<b>91.98</b>	<b>91.98</b>	93.83	88.27	92.42	90.91	<b>91.29</b>	88.60

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 4.5: Contributions (%) of DBT and TMP of DDTP $_{D_{M,C}}$  descriptor.

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

## 4.5.2 Experimental results

Evaluations of our framework for DT recognition on various benchmark datasets (UCLA, DynTex, and DynTex++) are specifically expressed in Tables 4.3, 4.4, and 4.9 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.

- As expected in Section 4.4.1, the incorporation between spatio-temporal of motion points (TMP) and directional features of beam trajectories (DBT) has boosted the performance in comparison with FDT [C4], in which motion points of dense trajectories along with their local neighbors are encoded to form directional beams of features (see Tables 4.6 and 4.8). Table 4.5 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 6.2).
- As mentioned in Section 3.4.3, 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 4.3, 4.8, and 4.9). It has verified the contributions of our important extensions to form the completed xLVP operator compared to the basic LVP [100].
- 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 4.3, 4.4, and 4.9).
- It is in accordance with our analysis in Section 4.4.1 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 4.3, 4.4, and 4.9). 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 [100] (see Section 4.4.1) 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 4.10). 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 4.6). 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.,  $\{(8, 1), (16, 2)\}^{riu2}$ .

#### 4.5.2.1 Recognition on UCLA dataset

It can be observed from Tables 4.3 and 4.6 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  $L = \{2, 3\}$ ,  $\text{DDTP}_{D-M}^{L=\{2,3\}}$  and  $\text{DDTP}_{D-M/C}^{L=\{2,3\}}$  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,  $\text{DDTP}_{D-M/C}^{L=3}$  descriptor gains 99% and 99.5% respectively, the highest compared to all LBP-based variants (see Table 4.6). Those performances are the same FDT's [C4], but in a half smaller dimension, i.e., 6768 versus over 13000 bins (see Table 6.2). On the other hand, DDTP-B using the setting of  $\{(8, 1), (16, 2)\}^{riu2}_{L=3}$  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  $\text{DDTP}_{D-M/C}^{L=3}$  has critical performances with 98.75% and 98.04% respectively, the highest in comparison with the LBP-based variants (see Table 4.6), except CVLBC [90] 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 4.8). In our previous work, FDT [C4] encoding motions of DTs along their trajectory is just better than  $\text{DDTP}_{D-M/C}^{L=3}$  on *8-class* with rate of 99.57%, but in about twice larger dimension. Furthermore, it should be noted that DT-CNN [63] 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 Figure 4.5,  $\text{DDTP}_{D-M/C}^{L=3}$  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 Figure 4.6).

In addition, it should be noted that several existing methods [92, 95, 112, J1] 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  $\text{DDTP}_{D-M/C}^{L=3}$  descriptor, as indicated in Table 4.7. Accordingly, its performance is noticeably reduced in comparison with those done on  $110 \times 160 \times 75$  videos (see Tables 4.3 and 4.7). 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 6.32 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.



Table 4.6: Comparison of recognition rates (%) on UCLA.

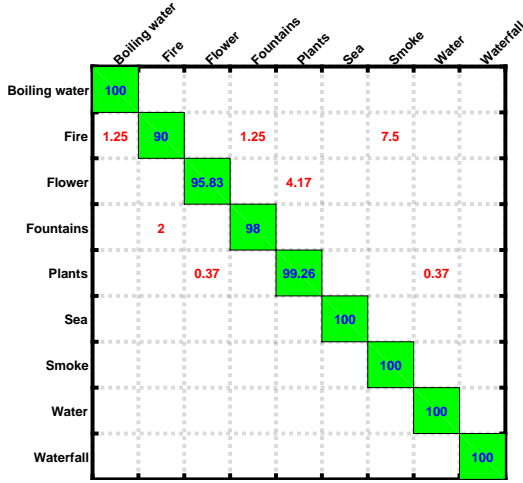
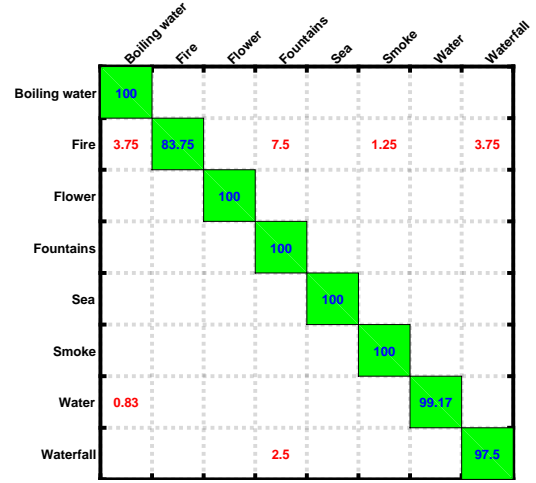
Group	Encoding method	50-LOO	50-4fold	9-class	8-class
A	FDT [C4]	98.50	99.00	97.70	99.35
	FD-MAP [C4]	99.50	99.00	99.35	<b>99.57</b>
	<b>DDTP</b> $_{D\_M}\{(8, 1), (16, 2)\}_{L=3}^{riu2}$	<b>100</b>	<b>100</b>	98.60	97.72
	<b>DDTP</b> $_{D\_M\_C}\{(8, 1), (16, 2)\}_{L=3}^{riu2}$	<b>100</b>	<b>100</b>	97.95	98.04
	<b>DDTP</b> $_{D\_M/C}\{(8, 1), (16, 2)\}_{L=3}^{riu2}$	99.00	99.50	98.75	98.04
	<b>DDTP-B</b> $\{(8, 1), (16, 2)\}_{L=3}^{riu2}$	99.50	98.50	96.15	96.30
B	AR-LDS [5]	89.90 <sup>N</sup>	-	-	-
	Chaotic vector [42]	-	-	85.10 <sup>N</sup>	85.00 <sup>N</sup>
	Diffusion-based model [110]	-	98.50 <sup>N</sup>	97.80 <sup>N</sup>	96.22 <sup>N</sup>
C	3D-OTF [51]	-	87.10	97.23	99.50
	DFS [50]	-	<b>100</b>	97.50	99.20
	STLS [53]	-	99.50	97.40	99.50
D	MBSIF-TOP [72]	99.50 <sup>N</sup>	-	-	-
	DNGP [38]	-	-	<b>99.60</b>	99.40
E	VLBP [14]	-	89.50 <sup>N</sup>	96.30 <sup>N</sup>	91.96 <sup>N</sup>
	LBP-TOP [14]	-	94.50 <sup>N</sup>	96.00 <sup>N</sup>	93.67 <sup>N</sup>
	CVLBP [91]	-	93.00 <sup>N</sup>	96.90 <sup>N</sup>	95.65 <sup>N</sup>
	HLBP [92]	95.00 <sup>N</sup>	95.00 <sup>N</sup>	98.35 <sup>N</sup>	97.50 <sup>N</sup>
	CLSP-TOP [C1]	99.00 <sup>N</sup>	99.00 <sup>N</sup>	98.60 <sup>N</sup>	97.72 <sup>N</sup>
	MEWLSP [95]	96.50 <sup>N</sup>	96.50 <sup>N</sup>	98.55 <sup>N</sup>	98.04 <sup>N</sup>
	WLBPC [109]	-	96.50 <sup>N</sup>	97.17 <sup>N</sup>	97.61 <sup>N</sup>
	CVLBC [90]	98.50 <sup>N</sup>	99.00 <sup>N</sup>	99.20 <sup>N</sup>	99.02 <sup>N</sup>
	CSAP-TOP [J1]	99.50	99.50	96.80	95.98
F	DL-PEGASOS [55]	-	97.50	95.60	-
	PI-LBP+super hist [111]	-	<b>100</b> <sup>N</sup>	98.20 <sup>N</sup>	-
	Orthogonal Tensor DL [69]	-	99.80	98.20	99.50
	Randomized neural network [112]	-	97.05 <sup>N</sup>	98.54 <sup>N</sup>	97.74 <sup>N</sup>
	PCANet-TOP [64]	99.50*	-	-	-
	DT-CNN-AlexNet [63]	-	99.50*	98.05*	98.48*
	DT-CNN-GoogleNet [63]	-	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 4.7: Results (%) on the cropped version of UCLA.

DDTP $_{D\_M/C}^{L=3}$	50-LOO			50-4fold			9-class			8-class		
$\{(P, R)\}_{L=3}^{riu2}$	$D\_M$	$D\_M\_C$	$D\_M/C$	$D\_M$	$D\_M\_C$	$D\_M/C$	$D\_M$	$D\_M\_C$	$D\_M/C$	$D\_M$	$D\_M\_C$	$D\_M/C$
$\{(8, 1)\}_{L=3}^{riu2}$	95.50	96.00	<b>96.00</b>	<b>97.00</b>	97.00	<b>97.00</b>	<b>95.00</b>	<b>95.40</b>	<b>96.45</b>	93.37	<b>95.87</b>	<b>94.89</b>
$\{(16, 2)\}_{L=3}^{riu2}$	93.50	96.00	94.00	<b>97.00</b>	<b>97.50</b>	96.00	92.50	92.80	94.95	92.72	91.41	92.72
$\{(8, 1), (16, 2)\}_{L=3}^{riu2}$	<b>96.50</b>	<b>96.50</b>	<b>96.00</b>	96.50	97.00	96.50	94.15	95.05	95.75	<b>94.46</b>	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.


 Figure 4.5: Confusion of  $\text{DDTP}_{D-M/C}^{L=3}$  on 9-class.

 Figure 4.6: Confusion of  $\text{DDTP}_{D-M/C}^{L=3}$  on 8-class.

#### 4.5.2.2 Recognition on DynTex dataset

It can be verified from Tables 4.4 and 4.8 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 [C4]. 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 [100]. Hereafter, we detail evaluations on each subset.

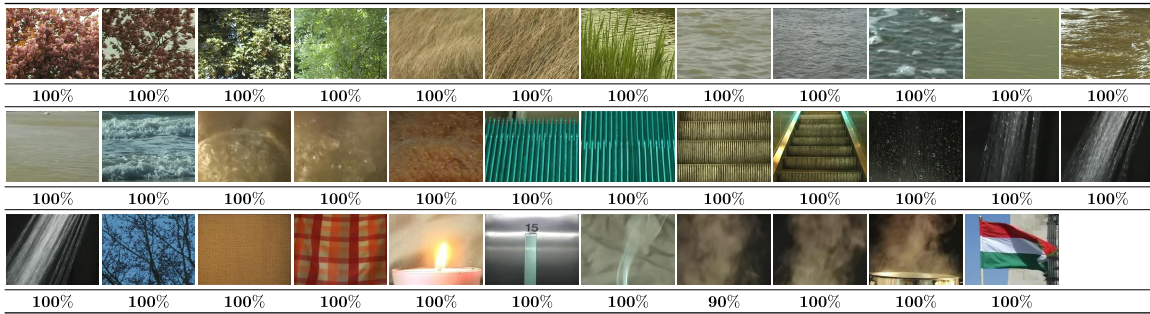
For DT recognition on *DynTex35*,  $\text{DDTP}_{D-M/C}^{L=3}$  descriptor with 6768 bins achieves 99.71%, a little lower than CSAP-TOP's [J1] (100%) with 13200 bins. It is due to the very similar motions of DTs in videos as shown in Figure 4.8(a) and Figure 4.8(b). Figure 4.7 expresses specific rates of each category. In the meanwhile, FD-MAP and FDT descriptors in our previous work [C4] just obtain rate of 98.86%. It is because only appearances of trajectories are involved with. The LBP-based method MEWLSP [95] also has the same our ability. However, it has not been verified on other challenging subsets, i.e., *Alpha*, *Beta*, and *Gamma* (see Table 4.8).

In respect of DT classification on other challenging subsets,  $\text{DDTP}^{L=\{2,3\}}$  descriptors obtain 98.33% on *Alpha* using  $\{(P, R)\} = \{(8, 1)\}$  of *riu2* mapping for both length of trajectories  $L = \{2, 3\}$  (see Table 4.4), but not outperform on *Beta* and *Gamma* in comparison with other parameters. For the setting of comparison,  $\text{DDTP}_{D-M/C}^{L=3}$  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 Figure 4.9). 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 Figures 4.10 and 4.11 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  $\text{DDTP}_{D-M/C}^{L=3}$  is the same those of CSAP-TOP [J1], FD-MAP [C4], and FDT [C4] (see Table 4.8), but in much smaller dimension, i.e., 6768 versus over 13000 bins of them (see Table 6.2). In the scenarios,  $\text{DDTP-B}$  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 [62], D3 [66], DT-CNN [63], obtain the best performances (see Table 4.8). However, they take a huge cost of computation as well as different parameters for learning DT features on each benchmark dataset.

Table 4.8: Comparison of rates (%) on DynTex and DynTex++.

Group	Encoding method	DynTex35	Alpha	Beta	Gamma	DynTex++
A	FDT [C4]	98.86	98.33	93.21	91.67	95.31
	FD-MAP [C4]	98.86	98.33	92.59	91.67	95.69
	$\text{DDTP}_{D.M}\{(8,1), (16,2)\}_{L=3}^{riu2}$	99.43	96.67	91.98	92.42	94.62
	$\text{DDTP}_{D.M.C}\{(8,1), (16,2)\}_{L=3}^{riu2}$	99.43	96.67	91.98	90.91	94.69
	$\text{DDTP}_{D.M/C}\{(8,1), (16,2)\}_{L=3}^{riu2}$	99.71	96.67	93.83	91.29	95.09
	$\text{DDTP-B}\{(8,1), (16,2)\}_{L=3}^{riu2}$	98.86	98.33	88.27	88.60	90.98
B	Diffusion-based model [110]	-	-	-	-	93.80 <sup>N</sup>
C	3D-OTF [51]	96.70	83.61	73.22	72.53	89.17
	DFS [50]	97.16	85.24	76.93	74.82	91.70
	2D+T [94]	-	85.00	67.00	63.00	-
	STLS [53]	98.20	89.40	80.80	79.80	94.50
D	MBSIF-TOP [72]	98.61 <sup>N</sup>	90.00 <sup>N</sup>	90.70 <sup>N</sup>	91.30 <sup>N</sup>	97.12 <sup>N</sup>
	DNGP [38]	-	-	-	-	93.80
E	VLBP [14]	81.14 <sup>N</sup>	-	-	-	94.98 <sup>N</sup>
	LBP-TOP [14]	92.45 <sup>N</sup>	98.33	88.89	84.85 <sup>N</sup>	94.05 <sup>N</sup>
	DDLBP with MJMI [113]	-	-	-	-	95.80
	CVLBP [91]	85.14 <sup>N</sup>	-	-	-	-
	HLBP [92]	98.57 <sup>N</sup>	-	-	-	96.28 <sup>N</sup>
	CLSP-TOP [C1]	98.29 <sup>N</sup>	95.00 <sup>N</sup>	91.98 <sup>N</sup>	91.29 <sup>N</sup>	95.50 <sup>N</sup>
	MEWLSP [95]	99.71 <sup>N</sup>	-	-	-	98.48 <sup>N</sup>
	WLBPC [109]	-	-	-	-	95.01 <sup>N</sup>
	CVLBC [90]	98.86 <sup>N</sup>	-	-	-	91.31 <sup>N</sup>
	CSAP-TOP [J1]	<b>100</b>	96.67	92.59	90.53	-
F	DL-PEGASOS [55]	-	-	-	-	63.70
	PCA-cLBP/PI/PD-LBP [111]	-	-	-	-	92.40
	Orthogonal Tensor DL [69]	-	87.80	76.70	74.80	94.70
	Equiangular Kernel DL [71]	-	88.80	77.40	75.60	93.40
	Randomized neural network [112]	-	-	-	-	96.51 <sup>N</sup>
	st-TCof [62]	-	<b>100*</b>	<b>100*</b>	98.11*	-
	PCANet-TOP [64]	-	96.67*	90.74*	89.39*	-
	D3 [66]	-	<b>100*</b>	<b>100*</b>	98.11*	-
	DT-CNN-AlexNet [63]	-	<b>100*</b>	99.38*	<b>99.62*</b>	98.18*
	DT-CNN-GoogleNet [63]	-	<b>100*</b>	<b>100*</b>	<b>99.62*</b>	<b>98.58*</b>

Note: “-” means “not available”. Superscript “\*” indicates result using deep learning algorithms. “N” is rate with 1-NN classifier. Group A denotes *optical-flow-based methods*, B: *model-based*, C: *geometry-based*, D: *filter-based*, E: *local-feature-based*, F: *learning-based*.

Figure 4.7: Specific rates of  $\text{DDTP}_{D-M/C}^{L=3}$  on each class of *DynTex35*.

#### 4.5.2.3 Recognition on DynTex++ dataset

Recognition results of our proposed framework with different settings are presented in Table 4.9. It can be observed from the table that  $\text{DDTP-B}^{L=\{2,3\}}$  descriptors with the setting of  $\{(8,1), (16,2)\}^{riu2}$

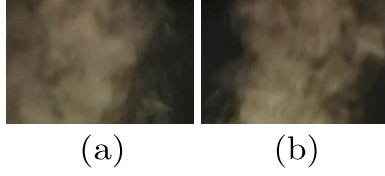


Figure 4.8: Video (a) is confused with (b) in recognition on *DynTex35*.

	Sea(20)	Grass(20)	Trees(20)
Sea(20)	20		
Grass(20)		19	1
Trees(20)		1	19

Figure 4.9: Confusion of  $\text{DDTP}_{D-M/C}^{L=3}$  on *Alpha*.

	Sea(20)	Vegetation(20)	Trees(20)	Flags(20)	Calm water(20)	Fountains(20)	Smoke(16)	Escalator(7)	Traffic(9)	Rotation(10)
Sea(20)	20									
Vegetation(20)		19				1				
Trees(20)			19			1				
Flags(20)				20						
Calm water(20)	1				19					
Fountains(20)						19				1
Smoke(16)							16			
Escalator(7)		1		1	1			4		
Traffic(9)									9	
Rotation(10)		1	1							7

Figure 4.10: Confusion of  $\text{DDTP}_{D-M/C}^{L=3}$  on *Beta*.

	Flowers(29)	Sea(38)	Naked trees(25)	Foliage(35)	Escalator(7)	Calm water(30)	Flags(31)	Grass(23)	Traffic(9)	Fountains(37)
Flowers(29)	27									
Sea(38)		38								
Naked trees(25)			23	1						
Foliage(35)				34						
Escalator(7)					5					
Calm water(30)						24				
Flags(31)							27			
Grass(23)								23		
Traffic(9)									9	
Fountains(37)										32

Figure 4.11:  $\text{DDTP}_{D-M/C}^{L=3}$ 's confusion on *Gamma*.

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 [100]. In terms of the settings chosen for comparison, the proposed descriptor  $\text{DDTP}_{D-M/C}^{L=3}$  achieves rate of 95.09%, the competitive performance compared to most of the existing methods (see Table 4.8). More specifically, only LBP-based approach MEWLSP [95] gains 98.48%, but as mentioned above, it is not better than ours on UCLA (see Table 4.6) as well as has not been validated on the challenging subsets of DynTex, i.e., *Alpha*, *Beta*, and *Gamma*. In the meanwhile, FDT [C4] and FD-MAP [C4], which are based on directional beams of dense trajectories for DT representation, obtain rates of 95.31% and 95.69% respectively, just a little higher than ours. Nevertheless, their dimensions are about twofold (see Table 6.2). DT-CNN [63] obtains 98.18% for the AlexNet framework, 98.58% for the GoogleNet framework (see Table 4.8). 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.

### 4.5.3 Global discussion

Beside particular evaluations on different benchmark DT datasets in Section 4.5.2, several general findings can be derived as follows.

- For DT representation, it can be validated from experimental results in Tables 4.3, 4.4, and 4.10 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 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

Table 4.9: Rates (%) of DDTP and DDTP-B descriptors on DynTex++.

$\{(P, R)\}_L^{riu2}$	$D\_M$	$D\_M\_C$	$D\_M/C$	DDTP-B
$\{(8, 1)\}_{L=2}^{riu2}$	93.85	94.01	94.14	87.10
$\{(16, 2)\}_{L=2}^{riu2}$	93.53	<b>94.92</b>	94.16	86.65
$\{(8, 1), (16, 2)\}_{L=2}^{riu2}$	<b>94.75</b>	<b>94.92</b>	<b>95.04</b>	<b>91.00</b>
$\{(8, 1)\}_{L=3}^{riu2}$	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	<b>95.09</b>	90.98

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

Dataset	UCLA (50-LOO)				DynTex35			
$\{(P, R)\}_L^{riu2}$	$D\_M$	$D\_M\_C$	$D\_M/C$	LVP-TOP	$D\_M$	$D\_M\_C$	$D\_M/C$	LVP-TOP
$\{(8, 1)\}_{L=2}^{riu2}$	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.

properties of a sequence, xLVP-TOP also significantly outperforms the basic LVP [100] applied on three orthogonal planes (see Table 4.10).

- 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 6.2 and 4.10).
- 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 4.3, 4.4, and 4.11). 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 6.2).
- Addressing  $u2$  mapping (e.g.,  $\{(8, 1)\}_{L=\{2,3\}}^{u2}$ ) for structuring DDTP features points out much larger dimension (see Section 4.5.1) while its performance is not improved as expected (see Table 4.11).
- 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 4.12 for that). This may be due to the short “living” time of turbulent motions in a video.

## 4.6 Summary

In this chapter, we have proposed an efficient framework for DT description 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. The experimental results have validated two following important contributions as follows. Firstly, taking dense trajectories into account DT representation is an interested alternative beside investigating the entire properties of a DT video. Secondly, 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 [100] do. In addition, evaluations have also verified that xLVP operator

Table 4.11: 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	<b>99.71</b>	96.86
$\{(24, 3)\}_{L=3}^{riu2}$	93.00	97.00	98.50	83.00	<b>99.14</b>	<b>99.43</b>	<b>99.71</b>	96.86
$\{(8, 1), (16, 2), (24, 3)\}_{L=2}^{riu2}$	<b>100</b>	99.50	<b>99.50</b>	97.50	<b>99.14</b>	<b>99.43</b>	99.43	<b>100</b>
$\{(8, 1), (16, 2), (24, 3)\}_{L=3}^{riu2}$	99.50	<b>100</b>	<b>99.50</b>	<b>99.50</b>	<b>99.14</b>	99.14	<b>99.71</b>	99.43
$\{(8, 1)\}_{L=2}^{u2}$	99.50	99.50	<b>99.50</b>	99.00	98.00	97.71	98.00	95.43
$\{(8, 1)\}_{L=3}^{u2}$	99.50	99.50	<b>99.50</b>	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 4.12: Performances (%) on longer dense trajectories on UCLA (50-LOO).

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

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.

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 [2, J5], Gaussian-based kernels [C2, C5, J3], can mitigate the negative impacts of illumination and noise on encoding dense trajectories.



---

---

# CHAPTER 5

---

## REPRESENTATION BASED ON MOMENT MODELS

### Contents

<b>5.1</b>	<b>Introduction</b>	<b>55</b>
<b>5.2</b>	<b>Moment models</b>	<b>56</b>
5.2.1	Moment images	56
5.2.2	A novel moment volumes	56
5.2.3	Advantages of moment volume model	57
<b>5.3</b>	<b>DT representation based on moment images</b>	<b>58</b>
<b>5.4</b>	<b>DT representation based on moment volumes</b>	<b>59</b>
5.4.1	Proposed momental directional descriptor	59
5.4.2	Enhancing the performance with max-pooling features	61
<b>5.5</b>	<b>Experiments and evaluations</b>	<b>61</b>
5.5.1	Experimental settings	61
5.5.2	Assessment of effectiveness of moment models	62
5.5.3	Experimental results of MDP-based descriptors	64
5.5.4	Global discussion	72
<b>5.6</b>	<b>Summary</b>	<b>73</b>

### 5.1 Introduction

In this chapter, we propose to represent DTs based on features of filtered elements computed by two filtering models: moment images and moment volumes. In general, the proposed framework mainly includes three following stages. Firstly, two moment models are taken into account video analysis to point out moment-filtered images/volumes correspondingly. Secondly, operator CLSP [29] is used to capture local features of the moment-filtered images while our proposed xLDP operator, an extended operator of Local Derivative Patterns [30] (see Section 3.3), is for capturing local derivative features of the moment-filtered volumes. Finally, the obtained histograms are concatenated and normalized to form robust descriptors of CSAP-TOP (see Section 5.3) and MDP-based features (see Section 5.4). To verify our works, we have experimented on benchmark DT datasets (UCLA [5], DynTex [54], DynTex++ [55]) for the recognition task. Experimental results which are thoroughly discussed in Section 5.5 have indicated that our framework outperforms significantly compared to the existing approaches, especially,



the performance of the MDP-based descriptors. Consequently, the major contributions of this work can be listed as follows.

- Representing DTs by extracting CLSP-based features from the moment-filtered images [J1].
- Representing DTs based on the novel moment-filtered volumes and the proposed xLDP [J5].

## 5.2 Moment models

Taking into account the advantages of filter-bank approaches as well as motivated by a filtering model of moment images [2], we propose in this section a new concept of moment volumes as a filtering technique in which different order moments of a sequence are calculated from a pre-defined element of spherical supporting volumes. In our framework for DT representation, this operation is regarded as a pre-processing with a low cost of computation to enrich robustness and discrimination for local DT features. Hereunder, we firstly take a look of the filtering model of moment images in Section 5.2.1. We then introduce in Section 5.2.2 the novel model of moment volumes which is stated more adaptive for video analysis due to its principle of enriching statistical features for a voxel instead for a pixel as done in the moment images.

### 5.2.1 Moment images

Nguyen *et al.* [2] presented a model of moment images, also known as a pre-processing step of image texture classification, in which still images are filtered by exploiting a LBP-based filter with a pre-defined supporting regional element. Encoding based on the filtered images points out local relationships with more stable textural structures against changes of environment. Two types of local statistical moments are produced as follows. First, the  $r$ -order moment image calculates the statistic distribution around a pixel  $\mathbf{q}_c$  as

$$m_{(\mathcal{I},B)}^r(\mathbf{q}_c) = \frac{1}{|B|} \sum_{\mathbf{p}_i \in B} \left( \mathcal{I}(\mathbf{q}_c + \mathbf{p}_i) \right)^r \quad (5.1)$$

in which  $\mathcal{I}$  means a 2D gray-scale image texture,  $\mathbf{q}_c$  is a center pixel (i.e.  $\mathbf{q}_c \in \mathcal{I}$ ),  $B$  is a supporting regional element consisting of points sampled by one or more concentric circles of the center  $\mathbf{q}_c$  with different radii  $R$ , i.e.,  $\{(P, R)\}$  (see Figure 5.1),  $|B|$  is the cardinality of  $B$ .

Second, the  $r$ -order centered moment image ( $r > 1$ ) defines the statistic distribution around a pixel  $\mathbf{q}_c$  as follows.

$$\mu_{(\mathcal{I},B)}^r(\mathbf{q}_c) = \frac{1}{|B|} \sum_{\mathbf{p}_i \in B} \left( \mathcal{I}(\mathbf{q}_c + \mathbf{p}_i) - m_{(\mathcal{I},B)}^1(\mathbf{q}_c) \right)^r \quad (5.2)$$

where  $m_{(\mathcal{I},B)}^1(\mathbf{q}_c)$  denotes the mean value (1-order moment) formed around pixel  $\mathbf{q}_c$ . Empirically, Nguyen *et al.* [2] have also shown that working on a series of moment images of different orders brings more textural information because the regional gray distribution is better captured using different statistical moments.

### 5.2.2 A novel moment volumes

The model of moment images has just considered spatial relations of a center pixel with its neighbors for image texture classification. To be in accordance with video representation, we hereafter propose a new local statistical model, called moment volumes as an extension of moment images, based on statistical moments calculated from a pre-defined spherical support. Similar to [2], our idea is motivated from filter-bank approaches to exploit more rich and robust information of shape and motion cues of DT videos by addressing different statistic distributions.

Let  $\mathcal{V}$  denote a 3D gray-scale level of a video and  $\mathbf{q}_c$  an arbitrary voxel of  $\mathcal{V}$ . Let  $\Omega = \{S_1, S_2, \dots, S_n\}$  be a local supporting volume as union of discrete spheres, centered at the same spatial coordinate, for calculating the statistic distributions at each voxel of  $\mathcal{V}$ . Each single spheric structuring

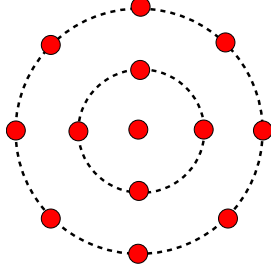


Figure 5.1: A sample of structuring element with  $\{(P,R)\}=\{(4,1),(8,2)\}$  [2].

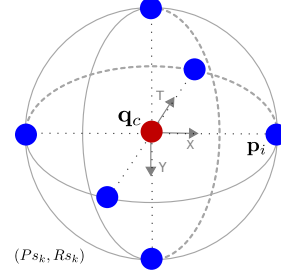


Figure 5.2: A pattern of volume support  $\Omega = \{(6, 1)\}$  which has  $P_k = 6$  blue neighbors sampled on a sphere with the center of red point and radius  $R_k = 1$ .

support  $S_k = (P_k, R_k)$  expresses that  $P_k$  neighbors are located on a sphere with radius  $R_k$ . In order to compute local statistic distribution at dynamic voxel  $\mathbf{q}_c$ , it is simply to settle the center of  $\Omega$  at  $\mathbf{q}_c$  and then to determine its neighbors defined by  $\Omega$ . To simplify the presentation, we adopt hereafter an assumption that coordinate of  $\mathbf{q}_c$  is  $(0, 0, 0)$ , it is possible to situate  $P_k$  neighbors on the sphere  $S_k$  in two following configurations:

- First, six points are placed on the endings of its orthogonal diameters, i.e.,  $\{(0, 0, R_k), (0, 0, -R_k), (-R_k, 0, 0), (R_k, 0, 0), (0, -R_k, 0), (0, R_k, 0)\}$ ,
- Second, in addition to the above set, this also consists of eight radial points. Each of which is located on the center of each one-eighth of the sphere  $S_k$ , i.e., its coordinate can be referred as one of different instances of  $(\pm R_k/\sqrt{3}, \pm R_k/\sqrt{3}, \pm R_k/\sqrt{3})$ . As the result, there are 14 neighbors in this configuration which can be considered for each supporting volume.

A sample of  $S_k = (6, 1)$  for the center  $\mathbf{q}_c$  can be formed by  $P_k = 6$  local neighbors on a sphere of  $R_k = 1$  as graphically illustrated in Figure 5.2. On the other hand, a local supporting volume may be unions of different discrete spheres. For example,  $\Omega = \{(6, 1), (14, 2)\}$  consists of two spheric structuring supports.

Given a pre-defined supporting volume  $\Omega$ , we propose to consider two following statistic distributions. Firstly, the  $r$ -order moment volume of association between video  $\mathcal{V}$  and the local supporting volume  $\Omega$  is defined as follows.

$$m_{\mathcal{V},\Omega}^r(\mathbf{q}_c) = \frac{1}{|\Omega|} \sum_{i=1}^{|\Omega|} \left( g(\mathbf{q}_c + \mathbf{p}_i) \right)^r \quad (5.3)$$

in which  $\mathbf{q}_c \in \mathcal{V}$  is the current voxel with its surrounding neighbors  $\mathbf{p}_i \in \Omega$ , the volume element  $\Omega$  can be structured by one or more spheres with the same center dynamic voxel and different radii,  $|\Omega|$  is the total of considered neighbors. Function  $g(\cdot)$  returns the gray level value of a voxel. Secondly, the  $r$ -order centered moment volume ( $r > 1$ ) can also be defined as

$$\mu_{\mathcal{V},\Omega}^r(\mathbf{q}_c) = \frac{1}{|\Omega|} \sum_{i=1}^{|\Omega|} \left( g(\mathbf{q}_c + \mathbf{p}_i) - m_{\mathcal{V},\Omega}^1(\mathbf{q}_c) \right)^r \quad (5.4)$$

where  $m_{\mathcal{V},\Omega}^1(\mathbf{q}_c)$  is the 1-order moment volume at the dynamic voxel  $\mathbf{q}_c$ . In practice, our model particularly considers two following types of moment volumes: the mean  $m^1$  and the variance  $\mu^2$  that are complementary and exploit well shape and motion cues of DT videos.

### 5.2.3 Advantages of moment volume model

By addressing different statistic distributions calculated from a pre-defined structuring volume, the proposed model of moment volumes has several following beneficial properties.

- *Insensitivity to noise:* Considering local statistic distributions (mean and variance) calculated from neighbors allows moment volumes to be more robust against noise than the raw video because the proposed model works like a low-pass filter which is able to eliminate dynamic voxels with intensely high frequency corresponding to noise.
- *Invariance to rotation:* Our model is independent on angle changes of frames in DT sequences because the pre-defined supporting region for calculating volume of moments is a union of discrete spheres, which is isotropic and so on discards all orientation information. Therefore, the moment volumes are invariant against rotation.
- *Information richness:* The concept of moment volume, which exploits textural information about local structures, allows to capture more global information. In addition, taking into account the advantages of filter-bank methods, our model permits to obtain more numerous types of local structures by using various moment distributions with different elements of the structuring volume. In practice, two order moments “mean” and “variance” are complementary, so these convey richer information than the original video.
- *Low computational cost:* Concerning the computational complexity, as filtered sequences are calculated on a pre-defined structuring volume, their calculation is simple and efficient along with the same computing cost like the typical LBP operator. Our algorithm in raw MATLAB code runs impressively fast on a Linux laptop of CPU Intel Core i7 1.9 Ghz with 4G RAM. It just takes less than 0.11s to handle a video with dimension of  $48 \times 48 \times 75$  for a 3D spherical supporting volume of  $\Omega = \{(6, 1)\}$  (see Figure 5.2).

### 5.3 DT representation based on moment images

Inspired by the concepts of CLSP [29] (see Section 2.7.3) and moment-image model, SBP [2], (see Section 5.2.1), we propose in this section an effective model of Completed Statistical Adaptive Patterns (CSAP) for DT description. Accordingly, as presented in Section 5.2.1, the model of moment images points out mean and variance of an input image. To be compliant with analysis of a given video  $\mathcal{V}$ , firstly, it is split into sets of plane-images  $\{f_{XY}, f_{XT}, f_{YT}\}$  subject to its orthogonal planes  $\{XY, XT, YT\}$ . For a set of plane-image  $f_{XY}$ , CLSP patterns are then computed for its plane-images as

$$\text{CSAP}_{P,R,m_1,\mu_2}(f_{XY}) = \frac{1}{|f_{XY}|} \sum_{\mathcal{I} \in f_{XY}} [\text{CLSP}_{P,R}(\mathcal{I}_{m_1}), \text{CLSP}_{P,R}(\mathcal{I}_{\mu_2})] \quad (5.5)$$

where  $|f_{XY}|$  denote the quantity of plane-images in  $f_{XY}$ ,  $\mathcal{I}_{m_1}$  and  $\mathcal{I}_{\mu_2}$  are mean and variance of  $\mathcal{I}$ . This computation is similarly exploited for the rest sets  $f_{XT}$  and  $f_{YT}$  to obtain corresponding histograms  $\text{CSAP}_{P,R,m_1,\mu_2}(f_{XT})$  and  $\text{CSAP}_{P,R,m_1,\mu_2}(f_{YT})$ . Consequently, A spatio-temporal CSAP-based descriptor based on three orthogonal planes of  $\mathcal{V}$  is constructed by simply concatenating and normalizing the obtained histograms as follows. Figure 5.3 perceptibly illustrates how to construct CSAP-TOP descriptor for DT representation.

$$\text{CSAP-TOP} = [\text{CSAP}_{P,R,m_1,\mu_2}(f_{XY}), \text{CSAP}_{P,R,m_1,\mu_2}(f_{XT}), \text{CSAP}_{P,R,m_1,\mu_2}(f_{YT})] \quad (5.6)$$

We now discuss how to utilize CSAP for an efficient description of DTs. As we have pointed out in Section 2.7, LBP-based approaches have been successfully applied to numerous works. Generally speaking, there are two main encoding mechanisms to transpose LBP-based variants from still images to dynamic texture: VLBP [14] that considers three neighboring circles from three consecutive frames, LBP-TOP [14] that addresses three circles extracted from three orthogonal planes. It should be noted that VLBP generates very higher dimensional feature vector and in the meanwhile it is less performance compared to LBP-TOP. Therefore, we investigate in the following CSAP on three orthogonal planes to form CSAP-TOP representation of sequences.

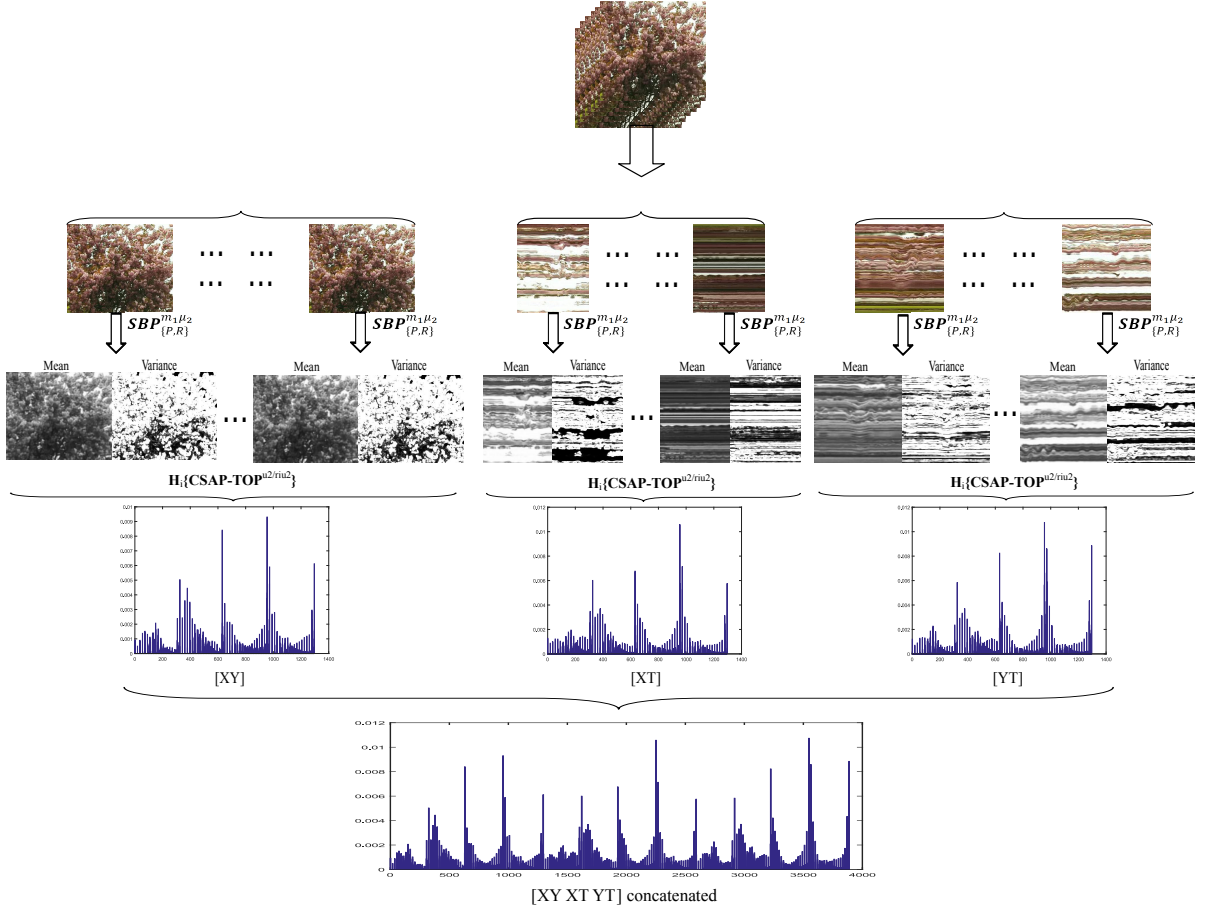


Figure 5.3: Illustration of completed statistical adaptive patterns on three orthogonal planes.

## 5.4 DT representation based on moment volumes

### 5.4.1 Proposed momental directional descriptor

In this section, we propose a new approach, named Momental Directional Patterns (MDP), to efficiently capture directional DT patterns from filtered videos obtained by the  $r$ -order moment volume model. Our idea is to take into account the advantages of filter bank approaches and a complementary LBP-based variant allowing to obtain more textural information in DT videos. We then consider our extended xLDP operator, presented in Section 3.3, on a series of volume moments which are introduced in Section 5.2 to result in Momental Directional Patterns for DT representation. Let us recall that the extended operator xLDP is introduced to work in still images. For that reason, in order to take it into account describing shape and motion cues of a DT video, we adopt the idea of [14] to address xLDP on three orthogonal planes of moment volumes.

Let  $\mathcal{V}$  denote a video and  $\mathcal{D}$  be a set of considered directions. The  $r$ -order moment volumes with supporting region  $\Omega$  are utilized to point out filtered sequences, i.e., mean ( $m^r$ ) and variance ( $\mu^r$ ) videos. DT characteristics in each of these are then encoded by exploiting the proposed operator xLDP with directions  $\alpha \in \mathcal{D}$  on three orthogonal planes XY, XT, YT of these moment volumes to compute the corresponding probability distributions, as graphically demonstrated in Figure 5.4. The obtained histograms are concatenated and normalized to form the final descriptor of video  $\mathcal{V}$  as follows.

$$\text{MDP}_{\Omega, \mathcal{D}}(\mathcal{V}) = [\text{xLDP}_{P, R, \mathcal{D}}(m^r_{XY}), \text{xLDP}_{P, R, \mathcal{D}}(m^r_{YT}), \text{xLDP}_{P, R, \mathcal{D}}(m^r_{XT}), \text{xLDP}_{P, R, \mathcal{D}}(\mu^r_{XY}), \text{xLDP}_{P, R, \mathcal{D}}(\mu^r_{XT}), \text{xLDP}_{P, R, \mathcal{D}}(\mu^r_{YT})] \quad (5.7)$$

From now on, we use the combination way of the extended xLDP operator to denote the corresponding

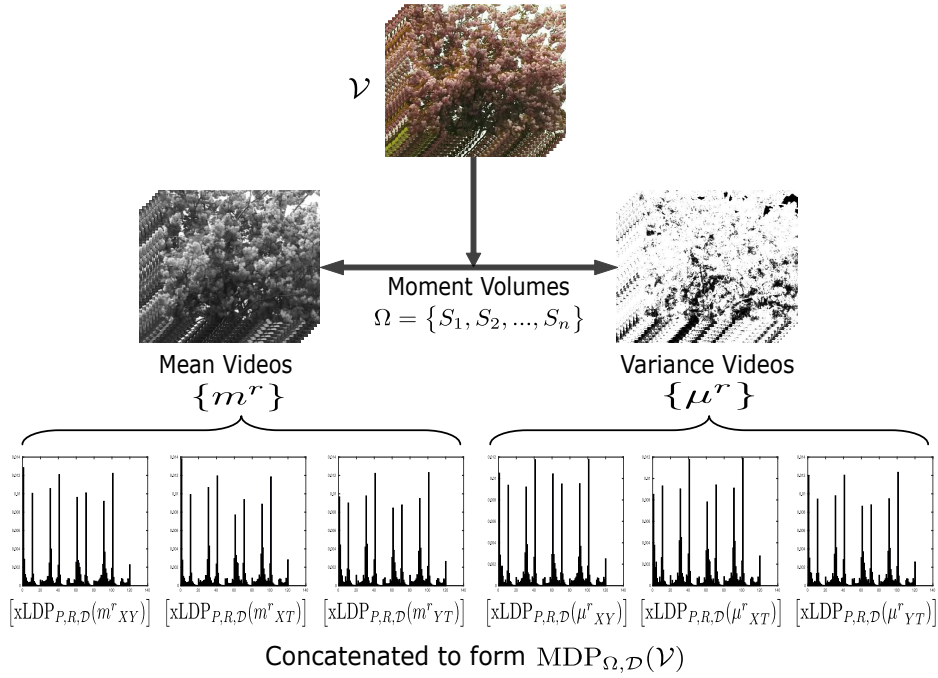


Figure 5.4: Illustration of structuring proposed DT descriptor.

descriptor MDP. For example,  $MDP_{D-M/C}$  means that it is based on the extended operator  $xLDP = LDP_D \cdot LDP_M / LDP_C$ , which is the concatenation between  $LDP_D$  and the joint of two components  $LDP_M$  and  $LDP_C$ .

In order to evaluate the contribution of the proposed extensions of LDP operator, a basic descriptor MDP-B, which is based on the second-order LDPs, is also considered by using the similar construction.

$$MDP-B_{\Omega,D}(\mathcal{V}) = [LDP_{P,R,D}(m^r_{XY}), LDP_{P,R,D}(m^r_{YT}), LDP_{P,R,D}(m^r_{XT}), \\ LDP_{P,R,D}(\mu^r_{XY}), LDP_{P,R,D}(\mu^r_{XT}), LDP_{P,R,D}(\mu^r_{YT})] \quad (5.8)$$

On the other hand, to verify the eminent contribution of our model of moment volumes, we also structure LDP-TOP patterns to depict the original DT sequence  $\mathcal{V}$  with non-supporting volume elements. These patterns are encoded by the typical second-order LDP operator on three orthogonal planes.

$$LDP-TOP_D(\mathcal{V}) = [LDP_{P,R,D}(\mathcal{V}_{XY}), LDP_{P,R,D}(\mathcal{V}_{XT}), LDP_{P,R,D}(\mathcal{V}_{YT})] \quad (5.9)$$

Two possible mappings can be taken into account for encoding DT features in order to reduce the dimension of representation:  $riu2$  and  $u2$  giving  $L_{riu2} = (P + 2)$  and  $L_{u2} = (P(P - 1) + 3)$  distinct values for each pixel pattern respectively, in which  $P$  is the considered neighbors. Particularly, the size of MDP descriptor depends on the combination ways of complemented components to form xLDP. For instance, descriptor  $MDP_{D-M/C}$ , computed by a style of  $xLDP = LDP_D \cdot LDP_M / LDP_C$  with  $3 \times |\mathcal{D}| \times L_{riu2/u2}$  bins, has dimension of  $9 \times |\mathcal{D}| \times L_{riu2/u2}(|m^r| + |\mu^r|)$  for  $riu2$  and  $u2$  mappings. Therein,  $|\mathcal{D}|$  denotes the number of concerned directions.  $|m^r|$  and  $|\mu^r|$  explain the quantity of “mean” and “variance” videos filtered by the  $r$ -order moment volume model. Towards the MDP-B and LDP-TOP descriptors, their dimensions are respectively fixed as  $3 \times |\mathcal{D}| \times L_{riu2/u2}(|m^r| + |\mu^r|)$  and  $3 \times |\mathcal{D}| \times L_{riu2/u2}$  bins corresponding to the mappings.

Furthermore, we also take advantage of the multi-scale performance [108] to enhance the discriminative power of DT descriptors. According to that, the proposed operators are utilized to calculate concerning probability distributions with different samples of neighbors  $\{(P, R)\}$ . The output histograms are then concatenated and normalized to produce multi-scale descriptors MMDP, MMDP-B, and MLDP-TOP.

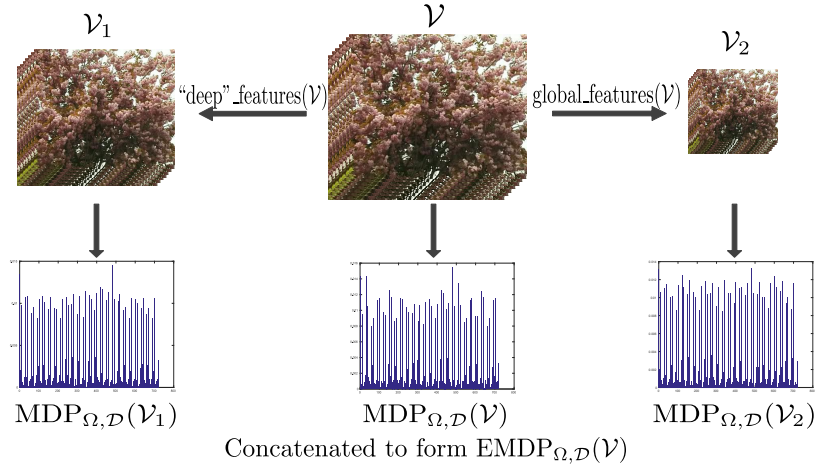


Figure 5.5: Illustration of constructing EMDP descriptor.

## 5.4.2 Enhancing the performance with max-pooling features

Inspired by CNNs [1, 56], we exploit the stage of max-pooling to obtain more intensity of global characteristics and “deep” patterns for DT representation (hereunder referred as max-pooling features). Accordingly, for a filtering window with size of  $\omega \times \omega$ , the max-pooling process is taken into account to analyze a video  $\mathcal{V}$  by striding the filter at 1 for calculating  $\mathcal{V}_1$  of “deep” features and at  $\omega$  for capturing  $\mathcal{V}_2$  of global characteristics. Then MDPs of the obtained sequences are computed and concatenated with those of  $\mathcal{V}$  to form an enhanced MDP (EMDP) descriptor as

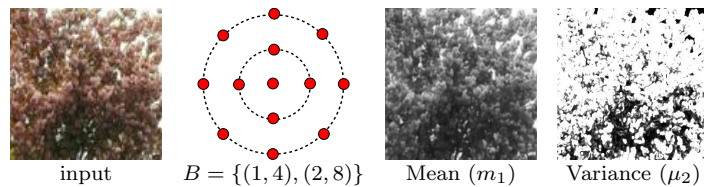
$$\text{EMDP}_{\Omega, \mathcal{D}}(\mathcal{V}) = [\text{MDP}_{\Omega, \mathcal{D}}(\mathcal{V}), \text{MDP}_{\Omega, \mathcal{D}}(\mathcal{V}_1), \text{MDP}_{\Omega, \mathcal{D}}(\mathcal{V}_2)] \quad (5.10)$$

Figure 5.5 graphically demonstrates an example of this computation. Similarity to MDP operator, EMDP is also considered in multi-scale regions to capture the further local features for structuring a more robust descriptor MEMDP.

## 5.5 Experiments and evaluations

### 5.5.1 Experimental settings

**Settings for moment images:** To be compliant with the LBP representation, we have chosen structuring elements as unions of discrete circles:  $B = \{(R_i, P_i)\}$  where  $P_i$  is a number of neighbors sampled with radius  $R_i$  of the  $i^{th}$  structuring element. Specifically, we have evaluated different instances as follows:  $\{(1, 4), (2, 8)\}$ ,  $\{(1, 5), (2, 8)\}$ ,  $\{(1, 5), (2, 10)\}$ ,  $\{(1, 6), (2, 10)\}$ ,  $\{(1, 6), (2, 12)\}$ ,  $\{(1, 8), (2, 16)\}$ . Figure 5.6 shows an example of computing moment images with supporting region  $B = \{(1, 4), (2, 8)\}$ . In the next sections, we only mention experiments using structuring element  $\{(1, 6), (2, 12)\}$  due to its outperformance on the various dynamic texture and scene datasets.


 Figure 5.6: A sample of computing moment images with supporting region  $B = \{(1, 4), (2, 8)\}$ .

**Settings for moment volumes:** Since encoding DTs on the high-order moment volumes results out DT descriptor with a large dimension, it should be considered in this work two first orders of moment volume model to calculate mean ( $m^1$ ) and variance ( $\mu^2$ ) sequences, i.e.,  $|m^1| =$



$|\mu^2| = 1$ . Structuring volume elements adopted to this filtering process are a set of supporting 3D spheres as  $\mathcal{S} = \{\Omega_1, \Omega_2, \dots, \Omega_m\}$ . Particularly, we have experimented on various elements of  $\{\{(6, 1)\}, \{(14, 1)\}, \{(6, 1), (6, 2)\}, \{(6, 1), (14, 2)\}, \{(14, 1)\}, (14, 2)\}\}$ . In the coming sections, we only present experiments using supporting volume of  $\Omega = \{(6, 1)\}$  owing to its better performance on the different DT datasets. An instance of filtering process exploiting this structuring element in two first-order moment volumes (i.e.,  $m^1$  and  $\mu^2$ ) is graphically illustrated in Figure 5.7.

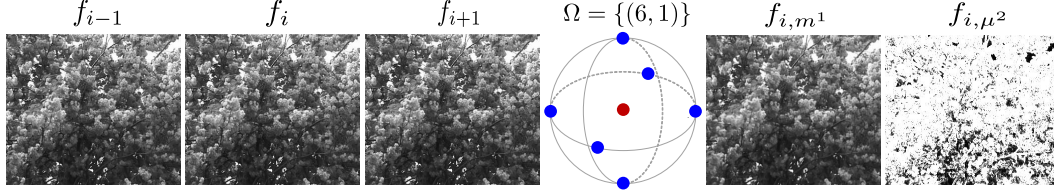


Figure 5.7: An example of filtering process using two first-order moment volumes (i.e.,  $m^1$  and  $\mu^2$ ) with a supporting element of 3D sphere  $\Omega = \{(6, 1)\}$ . Based on frames  $f_{i-1}$  and  $f_{i+1}$  of a video, frame  $f_i$  is filtered to form two corresponding frames  $f_{i,m^1}$  and  $f_{i,\mu^2}$ .

**Settings of moment-image-based descriptors:** For CSAP-TOP descriptor, two possible mappings can be used in our framework: *riu2* giving a descriptor of  $12(P+2)^2$  dimensions, *u2* giving a descriptor of  $12(P(P-1)+3)^2$  dimensions, where  $P$  is the number of considered neighbors. Table 5.1 illustrates the size of CSAP-TOP descriptor with *riu2* mapping compared to other LBP-based methods. It can be seen from this table that it is possible to take into account the advantage of multi-scale analysis [108] in order to improve the recognition accuracy, in which a computation of multiple operators with various parameters ( $P, R$ ) figures our corresponding histograms. These outputs are then normalized and concatenated to form multi-scale representation MCSAP-TOP. Our experiments indicate that the proposed framework points out better results with *riu2* mapping than *u2*. In particular, using single scale configuration gives good results but multi-scale is recommended since its performance on most of the DT datasets is still improved (see Tables 5.2, 5.3). In this case, the CLSP’s parameters give the best results with  $a = 0; b = 1$ . Regarding the neighborhood configuration, the best parameter setting is chosen as follows to compare with existing methods: *riu2* mapping with multi-scale  $\{(P, R)\} = \{(8, 1), (16, 2), (24, 3)\}$ .

**Settings of moment-volume-based descriptors:** Based on the two first-order filtered sequences to structure DT descriptors in justifiable dimension, we compute MDP, MDP-B, and LDP-TOP descriptors in 4 directions of  $\mathcal{D} = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ . For MDP descriptor, formed by the extended xLDP operator, three kinds of integrating complementary components can be experimented as  $\{\text{MDP}_{D-M}, \text{MDP}_{D-M/C}, \text{MDP}_{D-M.C}\}$  (hereunder called MDP descriptors for all) corresponding to dimensions of  $\{48L_{riu2/u2}, 72L_{riu2/u2}, 48(L_{riu2/u2} + 2)\}$  with *riu2* and *u2* mappings respectively. In respect of MDP-B and LDP-TOP descriptors, their lengths in this case are  $24L_{riu2/u2}$  and  $12L_{riu2/u2}$ . Several particular dimensions of these descriptors of *riu2* mapping can be seen in Table 5.1, in which it is possible for our operators to compute multi-scale descriptors for capturing more robust structural relations while retaining their sizes in reasonable dimensions compared to other LBP-based methods. Similarity to the settings for encoding MDP, descriptor EMDP is extra enhanced with the enhanced features computed from max-pooling videos which are formed with the *vl\_nnpool()* function<sup>1</sup> using the default parameters except Square filter =  $2 \times 2$ , Stride = 1 for “deep” features, and Stride = 2 for global characteristics.

### 5.5.2 Assessment of effectiveness of moment models

Addressing the settings settled in Section 5.5.1, we have experimented two descriptors based on the models of moment images (CSAP) and moment volumes (MDP) for DT recognition issue on benchmark

<sup>1</sup>[http://www.vlfeat.org/matconvnet/mfiles/vl\\_nnpool](http://www.vlfeat.org/matconvnet/mfiles/vl_nnpool)

Table 5.1: Several comparative dimensions of LBP-based descriptors for DT recognition.

Method	Dimensions	$P = 4$	$P = 8$	$P = 16$	$P = 24$
LBP-TOP <sup>u2</sup> [14]	$3 \times (P(P - 1) + 3)$	45	177	729	1665
VLBP [14]	$2^{3P+2}$	16384	-	-	-
CVLBP [91]	$3 \times 2^{3P+2}$	49152	-	-	-
HLBP [92]	$6 \times 2^P$	96	1536	-	-
CLSP-TOP <sup>riu2</sup> [C1]	$6(P + 2)^2$	216	600	1944	4056
WLBP [109]	$6 \times 2^P$	96	1536	-	-
MEWLSP [95]	$6 \times 2^P$	96	1536	-	-
CVLBC [90]	$2(3P + 3)^2$	392	1458	5202	11125
<b>CSAP-TOP<sup>riu2</sup></b>	$12(P + 2)^2$	-	1200	3888	8112
<b>MDP<sup>riu2</sup><sub>D-M</sub></b>	$48(P + 2)$	-	480	864	-
<b>MDP<sup>riu2</sup><sub>D-M-C</sub></b>	$48(P + 4)$	-	576	960	-
<b>MDP<sup>riu2</sup><sub>D-M/C</sub></b>	$72(P + 2)$	-	720	1296	-
<b>MDP-B<sup>riu2</sup></b>	$24(P + 2)$	-	240	432	-
<b>LDP-TOP<sup>riu2</sup></b>	$12(P + 2)$	-	120	216	-

Note:  $P$  is the considered neighbors. “-” means that the corresponding setting is either not reported or not implemented in practice due to its huge dimension. *riu2* and *u2* are two popular mappings for LBP-based variants. MDP-B and MDP descriptors are structured in 4 directions on two first-order filtered videos (also the settings for comparison their performance with the state-of-the-art in DT recognition).

 Table 5.2: Classification rates (%) on DT and scene datasets using single-scale CSAP-TOP<sup>riu2</sup> and its multi-scale analysis.

Datasets	UCLA				DynTex			
P, R, a, b	50-LOO	50-4fold	9-class	8-class	DynTex35	Alpha	Beta	Gamma
8, 1, 1, 0	96.50	97.00	97.05	96.52	98.29	<b>96.67</b>	93.21	92.80
8, 1, 0, 1	99.50	97.50	96.40	96.20	98.86	95.00	92.59	90.15
8, 1, 1, 1	98.50	98.50	97.10	95.98	98.29	93.33	90.74	91.29
16, 2, 1, 0	97.50	98.50	95.60	96.41	98.86	<b>96.67</b>	<b>93.21</b>	<b>94.70</b>
16, 2, 0, 1	<b>99.50</b>	99.00	96.65	95.22	99.43	<b>96.67</b>	<b>93.21</b>	91.29
16, 2, 1, 1	99.00	98.00	6.95	94.89	98.86	<b>96.67</b>	91.36	91.29
24, 3, 1, 0	<b>99.50</b>	<b>99.50</b>	95.25	94.35	99.71	<b>96.67</b>	92.59	92.80
24, 3, 0, 1	99.00	<b>99.50</b>	94.95	94.35	99.14	<b>96.67</b>	91.98	91.66
24, 3, 1, 1	<b>99.50</b>	<b>99.50</b>	96.00	95.65	<b>100</b>	<b>96.67</b>	90.12	91.29
$\{(4, 1), (8, 2), (12, 3)\}, 1, 0$	99.00	98.00	96.50	94.13	99.43	<b>96.67</b>	<b>93.83</b>	<b>94.70</b>
$\{(4, 1), (8, 2), (12, 3)\}, 0, 1$	<b>99.50</b>	<b>99.50</b>	96.80	94.13	99.71	<b>96.67</b>	92.59	92.05
$\{(4, 1), (8, 2), (12, 3)\}, 1, 1$	99.00	<b>99.50</b>	<b>97.50</b>	95.33	99.71	<b>96.67</b>	91.36	92.42
$\{(8, 1), (16, 2), (24, 3)\}, 1, 0$	<b>99.50</b>	<b>99.50</b>	96.15	95.11	99.71	<b>96.67</b>	<b>93.83</b>	93.18
$\{(8, 1), (16, 2), (24, 3)\}, 0, 1$	<b>99.50</b>	<b>99.50</b>	96.80	95.98	<b>100</b>	<b>96.67</b>	92.59	90.53
$\{(8, 1), (16, 2), (24, 3)\}, 1, 1$	<b>99.50</b>	<b>99.50</b>	96.75	<b>97.83</b>	<b>100</b>	95.00	89.51	91.29

Note: 50-LOO and 50-4fold denote results on 50-class breakdown using leave-one-out and four cross-fold validation respectively.

datasets UCLA [5] and DynTex [54]. Accordingly, experimental results of CSAP-TOP<sup>riu2/u2</sup> descriptors are shown on Tables 5.2 and 5.3 in their single-scale and multi-scale analyses using mostly complete kinds of encoding settings and their combinations, while performances of the MDP-based descriptors are mainly presented in Tables 5.6 and 5.7 in both single-scale and multi-scale analysis of local supporting regions.

In general, it can be verified from these tables that the MDP-based descriptors have significantly better performances compared to the CSAP-based ones. This certainly is thanks to addressing the xLDP



Table 5.3: Classification rates (%) on DT and scene datasets using single-scale CSAP-TOP<sup>u2</sup> and its multi-scale analysis.

Datasets	UCLA				DynTex			
P, R, a, b	50-LOO	50-4fold	9-class	8-class	DynTex35	Alpha	Beta	Gamma
4, 2, 1, 0	99.50	99.50	96.35	96.74	99.71	95.00	92.59	92.42
4, 2, 0, 1	99.50	99.50	95.80	94.24	98.86	<b>96.67</b>	90.74	92.42
4, 2, 1, 1	<b>100</b>	<b>100</b>	96.45	95.33	98.86	95.00	88.89	91.29
7, 2, 1, 0	99.00	99.50	96.75	94.02	99.14	95.00	92.59	93.18
7, 2, 0, 1	99.50	99.50	<b>98.25</b>	95.65	99.71	93.33	91.36	94.32
7, 2, 1, 1	99.00	99.50	97.55	96.30	<b>100</b>	95.00	92.59	<b>95.08</b>
$\{(4, 2), (5, 3)\}, 1, 0$	99.00	99.00	96.50	92.72	99.43	<b>96.67</b>	92.59	93.56
$\{(4, 2), (5, 3)\}, 0, 1$	99.00	99.00	97.35	94.35	99.43	93.33	93.21	93.18
$\{(4, 2), (5, 3)\}, 1, 1$	99.00	99.00	96.80	94.90	99.43	95.00	90.74	91.67
$\{(7, 2), (7, 3)\}, 1, 0$	98.50	99.00	96.65	94.89	99.71	95.00	92.59	93.94
$\{(7, 2), (7, 3)\}, 0, 1$	99.00	99.00	96.45	94.78	99.43	93.33	<b>93.83</b>	94.32
$\{(7, 2), (7, 3)\}, 1, 1$	99.00	99.00	97.05	<b>96.86</b>	<b>100</b>	95.00	93.21	94.70

Note: 50-LOO and 50-4fold denote results on 50-class breakdown using leave-one-out and four cross-fold validation respectively.

operator to forcefully extract the MDP-based features from the moment-filtered volumes for DT representation. This have substantiated the prominent contribution our proposals. Hence, in the rest of this Chapter, we just discuss the performance of the MDP-based descriptors in comprehensive comparison with other existing methods.

### 5.5.3 Experimental results of MDP-based descriptors

Performances on different benchmark DT datasets (UCLA [5], DynTex [54], DynTex++ [55]) of our framework, in which the proposed operators along with *riu2* and *u2* mappings are utilized to encode filtered videos in single-scale and multi-scale analyses for DT description, are detailed in corresponding Tables 5.6, 5.7, and 5.8 respectively. Based on the experimental results, we could make some crucial statements as follows.

First, as mentioned in Sections 5.2.2 and 5.2.3, exploiting moment volumes makes DT representation more insensitive to noise and illumination. Our experiments have verified that the DT descriptors MDP and MDP-B, computed on the filtered videos, have outstanding performance in comparison to the LDP-TOP’s, encoded on the raw DT sequence with non-supporting volumes (see Tables 5.6, 5.7, 5.8 for MDP, MDP-B, and Table 5.9 for LDP-TOP descriptor). In this regard, two first-order filtered (“mean” and “variance”) videos have notably contributed to the performance of the proposed descriptors (see Table 5.4). Second, it is in accordance with our evaluation in Section 3.3.4 that the combination of complemented components comprises additional discriminant information. As expected, all MDP descriptors outperforms significantly compared to MDP-B with a single complementary element (see Tables 5.6, 5.7, 5.8). Third, MDP descriptors exploiting the factor of directional center contrast (the component  $LDP_C$  of an extended operator xLDP) are often more informative than others. Therein, jointing with this factor makes those more robust to noise (see Tables 5.6, 5.8). Fourth, MDP descriptors with *riu2* mapping not only have tiny dimension but also deal with more efficiently than *u2*. Fifth, it is consistent with our analysis in Section 5.4.1 that multi-scale encoding allows to capture more local directional structures in larger regions. More specifically, multi-scale descriptors of *riu2* mapping ( $\{(P, R)\}^{riu2}$ ) are more efficient than single-scale. Therein, 2-scale (e.g.,  $\{(8, 1), (16, 2)\}$ ) achieves good results but the performance of 3-scale, i.e.,  $\{(8, 1), (16, 2), (24, 3)\}$  seems more “stable” on most of the benchmark DT datasets thanks to considering spatial features on the broad locality. Consequently, it should be recommended for implementation in practice, and also be the setting chosen for comparing with the state-of-the-art performances.

Table 5.4: Recognition (%) on “mean” ( $m^1$ ) and “variance” ( $\mu^2$ ) videos.

Dataset	50-LOO (UCLA)			Beta (DynTex)			DynTex++		
Descriptor	$m^1$	$\mu^2$	$\{m^1, \mu^2\}$	$m^1$	$\mu^2$	$\{m^1, \mu^2\}$	$m^1$	$\mu^2$	$\{m^1, \mu^2\}$
<b>MDP</b> <sub><math>D\_M</math></sub>	<b>99.50</b>	99.50	<b>100</b>	<b>96.30</b>	95.06	97.53	94.87	<b>94.89</b>	95.58
<b>MDP</b> <sub><math>D\_M\_C</math></sub>	<b>99.50</b>	99.50	<b>100</b>	<b>96.30</b>	<b>95.68</b>	97.53	<b>94.88</b>	94.68	95.70
<b>MDP</b> <sub><math>D\_M/C</math></sub>	<b>99.50</b>	<b>100</b>	<b>100</b>	95.68	<b>95.68</b>	<b>96.91</b>	94.41	<b>94.89</b>	<b>95.86</b>
<b>MDP-B</b>	98.00	98.00	99.50	88.89	89.51	88.27	93.98	94.02	95.82

Note:  $D\_M$ ,  $D\_M\_C$ , and  $D\_M/C$  are different integrations of complemented components of the extended operator xLDP to form the corresponding MDP descriptors. 50-LOO means results on 50-class breakdown using leave-one-out validation.

Furthermore, the MDP-B, which is based on the basic LDP (see Sections 3.3.4 and 5.5.3.4), has not performed as efficiently as MDP descriptors structured by the extended LDP operator. MDP also outperforms in comparison with LDP-TOP using the same configuration. These facts prove the effectiveness of our proposed components: the extensions of LDP operator and the model of moment volumes. However, it should be noted that MDP-B also obtains promising results compared to existing LBP-based methods thanks to the contribution of the  $r$ -order moment volume model.

In aspect of comparison with the existing approaches, our proposed method with a simple encoding technique conducts outstandingly in DT recognition issue compared to LBP-based variants for DT representation. In addition, its ability is the same as that of deep-learning-based frameworks in several circumstances (see Table 5.6). Hereafter, comprehensive evaluations of our proposal on different DT datasets are expressed clearly, in which if MDP descriptors are not specified their implemented configurations in detail, the default setting is mentioned for them, i.e.,  $\{(8, 1), (16, 2), (24, 3)\}^{riu2}$ .

### 5.5.3.1 Recognition on UCLA dataset

It can be verified in Tables 5.6 and 6.33 that the proposed method obtains the best recognition rates of 100% for both 50-LOO and 50-4fold schemes compared to the state-of-the-art results. For 9-class and 8-class scenarios, our proposal also acquires competitive performances. Hereafter, estimations on each of UCLA’s sub-datasets are detailed specifically.

*50-class:* It can be realized in Table 6.33 that MMDP <sub>$D\_M\_C$</sub>  and MMDP <sub>$D\_M$</sub>  achieve good results with 100% and 99.5% on 50-LOO and 50-4fold scenarios respectively. In aspect of the chosen comparing setting (see Section 5.5.3), MMDP <sub>$D\_M/C$</sub>  with only 3,888 bins outperforms with rate of 100% on both scenarios. It is the best performance in comparison to all existing methods including deep-learning-based approaches PCANet-TOP [64] and DT-CNN [63]. The filter-based method, MBSIF-TOP [72], achieves rate of 99.5% using a 7-scale descriptor of larger dimension (5,376 bins). Utilizing multi-fractal analysis to measure spatio-temporal features, DFS [50] obtains the same ours (100%) on 50-4fold scheme but it has not dealt with well on other challenging DT datasets (e.g., DynTex). Similarly, PI/PD-LBP variants [111] structure DT descriptors with grand dimensions using complicated learning procedures, and they have not been tested on DynTex.

*9-class:* In this scheme, MMDP <sub>$D\_M$</sub>  with rate of 98.90% is the best performance compared to other MDP descriptors. In the meanwhile, accuracies of MMDP <sub>$D\_M\_C$</sub>  and MMDP <sub>$D\_M/C$</sub>  are 98.35% and 98.70%, slightly lower rates of 99.20%, 99.35%, and 99.60% which are reported by CVLBC [90], FD-MAP [C4], and DNGP [38] respectively. However, CVLBC and FD-MAP is not better than ours on other scenarios (except 8-class) of UCLA dataset while DNGP has a complex representation. It should be noted that our method outperforms lightly compared to DT-CNN’s [63], 98.05% for AlexNet and 98.35% for GoogleNet deep learning framework. Specific recognition rate on each category in Figure 5.8 illustrates that MMDP <sub>$D\_M/C$</sub>  has mainly confused sequences of “Fire” with “Plants”, “Water” with “Waterfall”, and “Smoke” with “Water”. The reason for that may be the similar properties of those.

Table 5.5: Comparison of recognition rates (%) on benchmark DT datasets

Category	Dataset Encoding method	UCLA				DynTex				
		50-LOO	50-4fold	9-class	8-class	Dyn35	Alpha	Beta	Gamma	Dyn++
A	FDT [C4]	98.50	99.00	97.70	99.35	98.86	98.33	93.21	91.67	95.31
	FD-MAP [C4]	99.50	99.00	99.35	<b>99.57</b>	98.86	98.33	92.59	91.67	95.69
B	AR-LDS [5]	89.90 <sup>N</sup>	-	-	-	-	-	-	-	-
	KDT-MD [40]	-	97.50	-	-	-	-	-	-	-
	NLDR [43]	-	-	-	80.00	-	-	-	-	-
	Chaotic vector [42]	-	-	85.10 <sup>N</sup>	85.00 <sup>N</sup>	-	-	-	-	-
C	3D-OTF [51]	-	87.10	97.23	99.50	96.70	83.61	73.22	72.53	89.17
	WMFS [52]	-	-	97.11	96.96	-	-	-	-	-
	NLSSA [114]	-	-	-	-	-	-	-	-	92.40
	DFS [50]	-	<b>100</b>	97.50	99.20	97.16	85.24	76.93	74.82	91.70
	2D+T [94]	-	-	-	-	-	85.00	67.00	63.00	-
	STLS [53]	-	99.50	97.40	99.50	98.20	89.40	80.80	79.80	94.50
D	MBSIF-TOP [72]	99.50 <sup>N</sup>	-	-	-	98.61 <sup>N</sup>	90.00 <sup>N</sup>	90.70 <sup>N</sup>	91.30 <sup>N</sup>	97.12 <sup>N</sup>
	DNGP [38]	-	-	<b>99.60</b>	99.40	-	-	-	-	93.80
E	VLBP [14]	-	89.50 <sup>N</sup>	96.30 <sup>N</sup>	91.96 <sup>N</sup>	81.14 <sup>N</sup>	-	-	-	94.98 <sup>N</sup>
	LBP-TOP [14]	-	94.50 <sup>N</sup>	96.00 <sup>N</sup>	93.67 <sup>N</sup>	92.45 <sup>N</sup>	98.33	88.89	84.85 <sup>N</sup>	94.05 <sup>N</sup>
	DDLBP with MJMI [113]	-	-	-	-	-	-	-	-	95.80
	CVLBP [91]	-	93.00 <sup>N</sup>	96.90 <sup>N</sup>	95.65 <sup>N</sup>	85.14 <sup>N</sup>	-	-	-	-
	HLBP [92]	95.00 <sup>N</sup>	95.00 <sup>N</sup>	98.35 <sup>N</sup>	97.50 <sup>N</sup>	98.57 <sup>N</sup>	-	-	-	96.28 <sup>N</sup>
	CLSP-TOP [C1]	99.00 <sup>N</sup>	99.00 <sup>N</sup>	98.60 <sup>N</sup>	97.72 <sup>N</sup>	98.29 <sup>N</sup>	95.00 <sup>N</sup>	91.98 <sup>N</sup>	91.29 <sup>N</sup>	95.50 <sup>N</sup>
	MEWLSP [95]	96.50 <sup>N</sup>	96.50 <sup>N</sup>	98.55 <sup>N</sup>	98.04 <sup>N</sup>	<b>99.71<sup>N</sup></b>	-	-	-	98.48 <sup>N</sup>
	WLBPC [109]	-	96.50 <sup>N</sup>	97.17 <sup>N</sup>	97.61 <sup>N</sup>	-	-	-	-	95.01 <sup>N</sup>
	CVLBC [90]	98.50 <sup>N</sup>	99.00 <sup>N</sup>	99.20 <sup>N</sup>	99.02 <sup>N</sup>	98.86 <sup>N</sup>	-	-	-	91.31 <sup>N</sup>
	CSAP-TOP [J1]	99.50	99.50	96.80	95.98	<b>100</b>	96.67	92.59	90.53	-
	MMDP <sub>D,M</sub> [J5]	<b>100</b>	99.50	98.90	98.15	99.43	98.33	97.53	92.42	95.58
	MMDP <sub>D,M,C</sub> [J5]	<b>100</b>	99.50	98.35	98.59	99.43	98.33	97.53	92.42	95.70
	MMDP <sub>D,M/C</sub> [J5]	<b>100</b>	<b>100</b>	98.70	98.70	99.43	98.33	96.91	92.05	95.86
	MEMDP <sub>D,M/C</sub> [J5]	<b>100</b>	<b>100</b>	98.90	98.70	<b>99.71</b>	96.67	96.91	93.94	96.03
	MMDP-B [J5]	99.50	98.50	98.05	97.61	98.86	96.67	88.27	93.18	95.82
	MLDP-TOP [J5]	97.00	97.00	96.50	96.09	98.86	96.67	88.89	92.80	94.02
F	DL-PEGASOS [55]	-	97.50	95.60	-	-	-	-	-	63.70
	PI-LBP+super hist [111]	-	<b>100<sup>N</sup></b>	98.20 <sup>N</sup>	-	-	-	-	-	-
	PD-LBP+super hist [111]	-	<b>100<sup>N</sup></b>	98.10 <sup>N</sup>	-	-	-	-	-	-
	PCA-cLBP/PI-LBP/PD-LBP [111]	-	-	-	-	-	-	-	-	92.40
	Orthogonal Tensor DL [69]	-	99.80	98.20	99.50	-	87.80	76.70	74.80	94.70
	Equiangular Kernel DL [71]	-	-	-	-	-	88.80	77.40	75.60	93.40
	st-TCof [62]	-	-	-	-	-	<b>100*</b>	<b>100*</b>	98.11*	-
	PCANet-TOP [64]	99.50*	-	-	-	-	96.67*	90.74*	89.39*	-
	D3 [66]	-	-	-	-	-	<b>100*</b>	<b>100*</b>	98.11*	-
	DT-CNN-AlexNet [63]	-	99.50*	98.05*	98.48*	-	<b>100*</b>	99.38*	<b>99.62*</b>	98.18*
	DT-CNN-GoogleNet [63]	-	99.50*	98.35*	99.02*	-	<b>100*</b>	<b>100*</b>	<b>99.62*</b>	<b>98.58*</b>

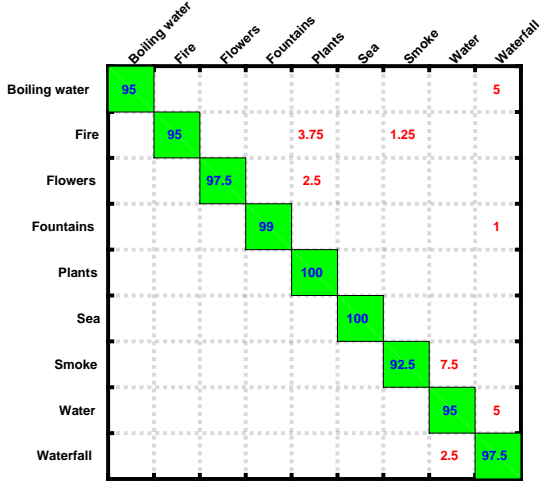
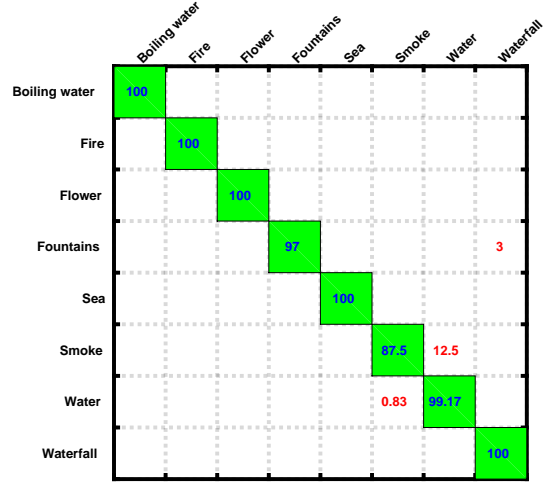
Note: “-” means “not available”. Superscript “\*” indicates results using deep learning algorithms. “N” indicates rates with 1-NN classifier. 50-LOO and 50-4fold denote results on 50-class breakdown using leave-one-out and four cross-fold validation respectively. Dyn35 and Dyn++ are abbreviated for DynTex35 and DynTex++ respectively. Group A is *optical-flow-based methods*, B: *model-based*, C: *geometry-based*, D: *filter-based*, E: *local-feature-based*, F: *learning-based*.

*8-class*: Obtaining rate of 98.7% with MMDP<sub>D,M/C</sub> in the more challenging scheme (see Table 5.6), it is interesting to note that the ability of our method is nearly the same as DT-CNN’s [63] utilizing deep-learning-based frameworks: AlexNet (98.48%) and GoogleNet (99.02%). It can be also observed in Table 6.33 that our method has the best performance among LBP-based methods, excluding CVLBC [90]. As mentioned above, it does not handle well on other schemes and has not been verified on the more challenging subsets of DynTex (i.e., Alpha, Beta, Gamma). Other non-LBP-based approaches, like Orthogonal Tensor DL (99.50%) [69], STLS (99.5%) [53], DNGP (99.4%) [38], DFS (99.2%) [50], 3D-OTF (99.5%) [51], FDT (99.35%) [C4], FD-MAP (99.57%) [C4], deal with more effectively than ours but their drawbacks are either sophisticated computation (e.g., Orthogonal Tensor DL, DNGP) or inefficient operation on other DT datasets (e.g., Orthogonal Tensor DL, DFS, 3D-OTF, STLS, FDT, FD-

Table 5.6: Classification rates (%) on UCLA using MDP, MDP-B descriptors and their multi-scale settings with mappings of  $riu2/u2$ .

Scheme	50-LOO				50-4fold				9-class				8-class			
	D_M	D_M.C	D_M/C	MDP-B	D_M	D_M.C	D_M/C	MDP-B	D_M	D_M.C	D_M/C	MDP-B	D_M	D_M.C	D_M/C	MDP-B
$\{(P, R)\}^{riu2/u2}$																
$\{(8, 1)\}^{riu2}$	98.00	98.00	98.50	96.00	97.50	97.50	98.50	96.00	97.60	98.60	98.40	94.50	95.33	96.85	96.41	94.89
$\{(16, 2)\}^{riu2}$	99.50	99.00	99.50	98.50	99.00	99.00	<b>100</b>	98.50	97.70	97.85	97.90	96.10	96.63	95.33	96.74	96.20
$\{(24, 3)\}^{riu2}$	99.50	99.50	97.00	98.50	<b>100</b>	<b>100</b>	97.50	98.00	96.85	98.25	97.45	95.50	96.96	97.17	97.39	95.54
$\{(8, 1), (16, 2)\}^{riu2}$	99.50	99.50	<b>100</b>	98.00	99.00	99.00	<b>100</b>	98.00	98.45	<b>99.00</b>	98.20	96.45	97.71	97.71	97.07	95.22
$\{(8, 1), (24, 3)\}^{riu2}$	<b>100</b>	99.50	<b>100</b>	98.00	99.50	99.50	<b>100</b>	97.50	98.20	98.65	98.40	96.55	97.83	97.50	98.15	97.28
$\{(16, 2), (24, 3)\}^{riu2}$	<b>100</b>	<b>100</b>	<b>100</b>	99.00	<b>100</b>	<b>100</b>	<b>100</b>	98.50	98.10	98.05	98.55	96.40	97.61	97.50	98.40	96.41
$\{(8, 1), (16, 2), (24, 3)\}^{riu2}$	<b>100</b>	<b>100</b>	<b>100</b>	<b>99.50</b>	99.50	99.50	<b>100</b>	98.50	<b>98.90</b>	98.35	<b>98.70</b>	<b>98.05</b>	98.15	<b>98.59</b>	<b>98.70</b>	97.61
$\{(8, 1)\}^{u2}$	99.00	99.00	99.00	98.00	99.00	99.00	99.00	97.50	98.60	98.25	97.35	97.65	<b>98.80</b>	98.37	97.93	95.00
$\{(16, 2)\}^{u2}$	99.50	99.50	99.50	99.00	99.50	99.50	99.50	98.00	96.95	98.00	97.30	95.65	96.96	97.50	96.52	<b>98.80</b>
$\{(24, 3)\}^{u2}$	99.50	99.50	-	<b>99.50</b>	99.50	99.50	-	<b>99.00</b>	96.40	96.60	-	94.65	97.07	96.10	-	95.54

Note: D\_M, D\_M.C, and D\_M/C are different integrations of complemented components of the extended operator xLDP to form the corresponding MDP descriptors. 50-LOO and 50-4fold denote results on 50-class breakdown using leave-one-out and four cross-fold validation respectively. “-” means that the corresponding MDP is not implemented due to the problem of large dimension.


 Figure 5.8: Confusion matrix (%) of  $MMDP_{D_M/C}$  on 9-class.

 Figure 5.9: Confusion matrix (%) of  $MMDP_{D_M/C}$  on 8-class.

MAP). The confusion matrix of each class in Figure 5.9 indicates that  $MMDP_{D_M/C}$  has principally confused the properties of “Smoke” sequences with “Water” due to their alike features.

### 5.5.3.2 Recognition on DynTex dataset

Tables 5.7 and 6.33 indicate that our method obtains the best results compared to existing LBP-based methods and other non-deep-learning techniques on this scheme. Specific evaluations on each of DynTex’s variants are expressed in detail as follows.

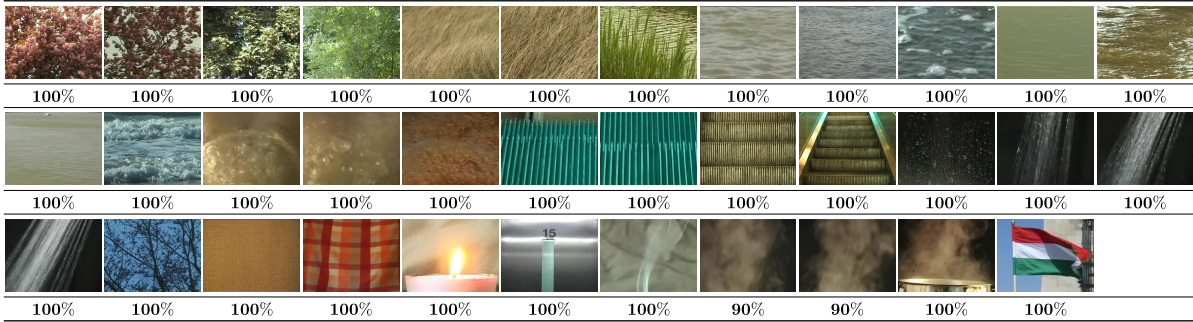
**DynTex35:** It can be observed in Table 5.7 that the highest rate of recognition on this scenario is 100% reported by  $MDP_{D_M}^{u2}(24, 3)$  and  $MDP_{D_M.C}^{u2}(24, 3)$ . In the meanwhile,  $MMDP_{D_M}$ ,  $MMDP_{D_M.C}$ , and  $MMDP_{D_M/C}$  result out lightly lower rate of 99.43%. This is because of the similarity of features in two classes c and d, as shown in Figure 5.11, that they are not able to differentiate. The detail of classification rate of  $MMDP_{D_M/C}$  is exposed in Figure 5.10. CVLBC [90] obtains accuracy of 99.71% on this scheme (see Table 6.33), slightly higher than our  $MMDP$  descriptors’ but it has not verified on other challenging variants of DynTex (i.e., Alpha, Beta, Gamma).

**Alpha:** In this scheme,  $MMDP_{D_M}$  and  $MMDP_{D_M.C}$  with rate of 98.33% (see Table 5.7) outperform compared to that of  $MMDP_{D_M/C}$  with 96.67% due to the confusion of two DT sequences (see Figure 5.12). Those results are also the best in comparison with all existing methods excluding deep-learning-based approaches st-TCof [62], DT-CNN [63], and D3 [66].

Table 5.7: Rates (%) on DynTex using MDP, MDP-B descriptors and their multi-scale settings with mappings of  $riu2/u2$ .

Scheme	DynTex35				Alpha				Beta				Gamma			
	D.M	D.M.C	D.M/C	MDP-B	D.M	D.M.C	D.M/C	MDP-B	D.M	D.M.C	D.M/C	MDP-B	D.M	D.M.C	D.M/C	MDP-B
$\{(P, R)\}^{riu2/u2}$																
$\{(8, 1)\}^{riu2}$	96.86	96.57	97.43	97.43	95.00	93.33	95.00	<b>96.67</b>	94.44	95.06	95.68	90.12	92.42	92.05	92.80	91.29
$\{(16, 2)\}^{riu2}$	98.00	97.71	98.86	98.57	<b>98.83</b>	<b>98.83</b>	<b>98.83</b>	<b>96.67</b>	95.68	95.68	96.30	90.74	<b>93.18</b>	92.05	91.67	<b>93.94</b>
$\{(24, 3)\}^{riu2}$	99.43	99.43	<b>99.43</b>	99.14	<b>98.83</b>	<b>98.83</b>	96.67	<b>96.67</b>	96.91	96.91	<b>96.91</b>	88.89	<b>93.18</b>	92.80	93.18	90.15
$\{(8, 1), (16, 2)\}^{riu2}$	97.71	98.00	98.86	98.86	<b>98.33</b>	<b>98.33</b>	<b>98.33</b>	<b>96.67</b>	95.06	95.06	96.30	90.74	92.80	92.42	92.05	92.05
$\{(8, 1), (24, 3)\}^{riu2}$	99.43	99.43	<b>99.43</b>	98.86	<b>98.33</b>	<b>98.33</b>	96.67	<b>96.67</b>	96.91	97.53	<b>96.91</b>	89.51	<b>93.18</b>	92.80	91.67	90.91
$\{(16, 2), (24, 3)\}^{riu2}$	99.43	99.43	<b>99.43</b>	98.57	<b>98.33</b>	<b>98.33</b>	96.67	<b>96.67</b>	96.91	<b>98.15</b>	96.30	88.89	92.42	92.42	92.80	<b>93.94</b>
$\{(8, 1), (16, 2), (24, 3)\}^{riu2}$	99.43	99.43	<b>99.43</b>	98.86	<b>98.33</b>	<b>98.33</b>	96.67	<b>96.67</b>	<b>97.53</b>	97.53	<b>96.91</b>	88.27	92.42	92.42	92.05	93.18
$\{(8, 1)\}^{u2}$	97.14	97.14	98.00	98.57	95.00	95.00	95.00	<b>96.67</b>	92.59	93.83	93.21	90.12	92.80	92.42	92.80	89.77
$\{(16, 2)\}^{u2}$	98.86	99.14	<b>99.43</b>	99.14	96.67	96.67	96.67	<b>96.67</b>	93.83	94.44	95.06	91.36	<b>93.18</b>	92.80	<b>94.68</b>	91.67
$\{(24, 3)\}^{u2}$	<b>100</b>	<b>100</b>	-	<b>99.43</b>	96.67	96.67	-	95.00	93.83	93.83	-	<b>92.59</b>	<b>93.18</b>	<b>93.18</b>	-	90.91

Note: D.M, D.M.C, and D.M/C are different integrations of complemented components to form the corresponding MDP descriptors. “-” denotes that the corresponding MDP is not implemented due to the problem of large dimension.

Figure 5.10: Specific recognition of  $MMDP_{D.M/C}$  on each class of DynTex35.

**Beta:** It can be realized in Tables 5.7 and 6.33 that our MDP descriptors have the best performance compared to all non-deep-learning-based methods. More specifically,  $MMDP_{D.M.C}$  of  $(16, 2)(24, 3)^{riu2}$  gains the highest rate of 98.15%, slightly better than  $MMDP_{D.M}$  and  $MMDP_{D.M/C}$  with (96.91%) and (97.53%) respectively. Those performances are much better than PCANet-TOP’s [64] and about 1% to 3% lower than st-TCofF’s [62], DT-CNN’s [63], and D3 [66], in which exploiting complicated learning algorithms along with tremendous dimension of DT representation while those are crucial to ensure feasible implementations in practice. The confusion matrix of  $MMDP_{D.M/C}$  in Figure 5.13 indicates that it has mostly confused “Rotation” sequences with “Vegetation” and “Trees”.

**Gamma:** In this scenario, rate of 94.68% is the best recognition pointed out by  $MDP_{D.M/C}^{u2}(16, 2)$  while multi-scale MMDP also obtains good results from 92% to 93%. Towards the setting chosen for comparison,  $MMDP_{D.M/C}$  achieves rate of 92.05%, better than all existing methods excepting LBP-TOP’s implemented in [62] and that of deep-learning-based approaches. In order to address which categories have enforced the misunderstanding of  $MMDP_{D.M/C}$  for the improvement work, the confusion matrix is figured out as in Figure 5.14. According to that, mutual confusion between sequences of “Fountains” and “Calm water” should be concentrated on for perspectives.

### 5.5.3.3 Recognition on Dyntex++ dataset

It can be observed from Table 6.33 and 5.8 that MDP descriptors have performed well in comparison to the existing approaches. Specifically, the best recognition rate on this scheme is 96.51% (see Table 5.8) reported by  $MDP_{D.M.C}^{u2}(8, 1)$ . The descriptors of  $MMDP_{D.M}$ ,  $MMDP_{D.M.C}$ , and  $MMDP_{D.M/C}$  obtain 95.58%, 95.7%, and 95.86% respectively, those which are the highest rates compared to the existing methods using SVM algorithm for classification. In aspect of the comparing setting, the performance of  $MMDP_{D.M/C}$  is nearly the same MBSIF-TOP’s (97.12%) [72] with 8-scale descriptor formed by 8 learned filters, and about 3% lower than DT-CNN’s (98.18%) [63] using deep learning techniques of AlexNet for learning DT features. The LBP-based method, MEWLSP [95], acquires the highest recog-



Figure 5.11: Two mutual confused categories in recognition on DynTex35.

	Sea(20)	Grass(20)	Trees(20)
Sea(20)	20		
Grass(20)		19	1
Trees(20)		1	19

Figure 5.12: Confusion matrix for  $\text{MMDP}_{D\_M/C}$  on Alpha.

	Sea(20)	Vegetation(20)	Trees(20)	Flags(20)	Calm water(20)	Fountains(20)	Smoke(16)	Escalator(7)	Traffic(9)	Rotation(10)
Sea(20)	20									
Vegetation(20)		20								
Trees(20)			20							
Flags(20)				19					1	
Calm water(20)	1				19					
Fountains(20)						20				
Smoke(16)							16			
Escalator(7)								6		1
Traffic(9)									9	
Rotation(10)		1	1							8

Figure 5.13: Confusion matrix for  $\text{MMDP}_{D\_M/C}$  on Beta.

	Flowers(29)	Sea(38)	Naked trees(25)	Foliage(35)	Escalator(7)	Calm water(30)	Flags(31)	Grass(23)	Traffic(9)	Fountains(37)
Flowers(29)	26		2						1	
Sea(38)		38								
Naked trees(25)	1		23	1						
Foliage(35)	1			34						
Escalator(7)	1				5		1			
Calm water(30)						26			4	
Flags(31)			1				29		1	
Grass(23)			1		1			21		
Traffic(9)									9	
Fountains(37)			1		4					32

Figure 5.14: Confusion matrix for  $\text{MMDP}_{D\_M/C}$  on Gamma.

nition rate of 98.48% on this scheme, even better than DT-CNN's (98.18%) [63]. However, it does not outperform on UCLA dataset compared to ours as well as has not been justified on other challenging DynTex variants (i.e., Alpha, Beta, Gamma). Another sophisticated method utilizing deep learning framework of GoogleNet [63] has prominent classification rate but it takes a long time to handle DT features with a huge complicated computation while these costs are crucial in real-time applications of computer vision. Accuracies of  $\text{MMDP}_{D\_M/C}$  on each categories are detailed in Figure 5.15. Accordingly, our descriptor outperforms on most of categories, only five of them (highlighted in red rates) are really challenges for the future work.

#### 5.5.3.4 Assessing the proposed components: Recognition with MDP-B and LDP-TOP

We address in this section some experiments for verifying our proposed components. Two following descriptors (see also Section 5.4.1 for more details) are considered: i) LDP-TOP that applies directly the second-order LDP operator on three orthogonal planes of raw videos; ii) MDP-B has the same architecture as that of MDP descriptors but on the contrary it is based only on LDP operator. It is evident that the comparisons between LDP-TOP and MDP-B, between MDP-B and MDP, allow to highlight respectively the contribution of moment volumes, and that of the extended operator xLDP.

It could be seen from Tables 5.6, 5.7, 6.33, 5.8, 5.9 that MDP descriptors are more efficient and “stable” than MDP-B and LDP-TOP ones. Table 6.33 shows that our proposals permit to prominently improve MDP's performance compared to the straightforward LDP-TOP version on most of DT datasets (e.g., up to 8.64% on Beta dataset). It also outperforms in comparison with MDP-B on various datasets (e.g., up to 9.26% on Beta).

Moreover, the execution of LDP-TOP is impaired in comparison to MDP-B's on most of the DT datasets (see also Table 6.33) due to non-supporting volume taken into account. This fact proves that

Table 5.8: Recognition (%) on DynTex++ using MDP, MDP-B descriptors and their multi-scale settings with mappings of  $riu2/u2$ .

Dataset	DynTex++			
$\{(P, R)\}^{riu2/u2}$	D_M	D_M_C	D_M/C	MDP-B
$\{(8, 1)\}^{riu2}$	93.93	94.28	94.52	92.71
$\{(16, 2)\}^{riu2}$	95.27	94.70	95.18	94.25
$\{(24, 3)\}^{riu2}$	93.92	94.09	93.71	92.16
$\{(8, 1), (16, 2)\}^{riu2}$	95.47	95.59	95.56	95.38
$\{(8, 1), (24, 3)\}^{riu2}$	94.92	95.10	94.88	94.92
$\{(16, 2), (24, 3)\}^{riu2}$	95.37	94.85	95.11	95.07
$\{(8, 1), (16, 2), (24, 3)\}^{riu2}$	95.58	95.70	95.86	95.82
$\{(8, 1)\}^{u2}$	95.97	<b>96.51</b>	<b>96.18</b>	<b>96.51</b>
$\{(16, 2)\}^{u2}$	<b>96.37</b>	96.28	95.92	96.39
$\{(24, 3)\}^{u2}$	95.72	95.68	-	94.79

Note: D\_M, D\_M\_C, and D\_M/C are different integrations of complemented components of the extended operator xLDP to form the corresponding MDP descriptors. “-” denotes that the corresponding MDP is not implemented due to the problem of large dimension.

considering moment volumes inspite of raw videos allows to capture more robust and discriminative features to enhance the performance of DT descriptors.

In the meanwhile, with the same configuration, MDP-B fails behind MDP descriptors on most of DT datasets because the typical second-order LDP is used instead of our extended operator xLDP (see Section 3.3.4). This shows the important contribution of two proposed extensions for LDP operator to make DT descriptors more robust and discriminative. However, it should be noted that MDP-B’s performance produces competitive results that are still comparable with the existing methods in several circumstances thanks to the collaboration of the filtered videos figured out by the proposed model of  $r$ -order moment volumes.

Because of those, the below evaluations mainly focus on the performance of MDP-B compared to the existing approaches.

**UCLA:** The performance of MMDP-B with multi-scale setting of  $\{(8, 1), (16, 2), (24, 3)\}^{riu2}$  acquires recognition rates of 99.5%, 98.5%, 98.05%, and 97.61% for 50-LOO, 50-4fold, 9-class, and 8-class scenarios respectively, those which are comparable to the LBP-based methods (see Table 6.33). In 50-LOO and 50-4fold schemes, the results of LDP-TOP $^{u2}$ (16, 2) are also promising with rates of 99% and 99.5% (see Table 5.9).

**DynTex:** In this scheme, MMDP-B and MLDP-TOP with comparing configuration just break down on Beta with classification rate of 88.27% and 88.89% respectively while they and their other settings perform well on other variants of DynTex dataset (see Table 5.7). More specifically, the best recognition rates on DynTex35 is 99.43% resulted by MDP-B $^{u2}$ (24, 3), LDP-TOP $^{u2}$ (24, 3), and 99.14% reported by MDP-B $^{riu2}$ (24, 3) with only 624 dimensions. Towards the comparing setting, MMDP-B and MLDP-TOP achieve rate of 98.86% on DynTex35, the best classification among the LBP-based variants except MEWLSP’s [95] (99.71%) (see Table 6.33). Although not better than the ability of MDP on Beta, MDP-B obtains comparable rates against those of all existing techniques excepting deep learning methods, i.e., st-TCof [62], D3 [66], DT-CNN [63]. Furthermore, it is interesting to note that the operation of MMDP-B is slightly better than MMDP’s in verifying on Gamma scheme with 93.18% in contrast to 92.05% of MMDP.







































											
100%	100%	96.80%	100%	92.20%	97.20%	99.80%	100%	91.00%	98.40%	94.80%	89.60%
											
98.00%	99.20%	100%	100%	99.20%	97.20%	100%	95.20%	99.00%	100%	99.20%	94.40%
											
98.40%	99.80%	93.20%	100%	86.60%	98.00%	98.20%	81.40%	95.40%	76.40%	96.20%	86.00%

Figure 5.15: Recognition of  $\text{MMDP}_{D-M/C}$  on specific categories of DynTex++, which the challenging ones are highlighted in red rates.

**DynTex++:** Utilizing complicated learning algorithms, DT-CNN [63] outperforms dominantly on this scenario (98.58%). The MMDP-B descriptor of  $\{(8, 1), (16, 2), (24, 3)\}^{riu2}$  with only size of 1,350 bins gains the promising results with rate of 95.82%, lightly better than that of  $\text{MMDP}_{D-M}$  and  $\text{MMDP}_{D-M/C}$ . Thanks to exploiting spatio-temporal information of the moment volumes,  $\text{MDP-B}^{u2}(8, 1)$  resulted out the highest rate of 96.51%, just about 2% lower than DT-CNN’s [63].

Table 5.9: Classification rates (%) of LDP-TOP descriptor and its multi-scale settings with mappings of  $riu2/u2$  on DT datasets without applying the proposed moment volume model.

Dataset	UCLA				DynTex				
$\{(P, R)\}^{riu2/u2}$	50-LOO	50-4fold	9-class	8-class	Dyn35	Alpha	Beta	Gamma	Dyn++
$\{(8, 1)\}^{riu2}$	93.00	96.00	96.30	96.09	96.00	<b>98.33</b>	87.04	87.12	89.82
$\{(16, 2)\}^{riu2}$	96.50	98.00	96.55	<b>96.74</b>	97.14	96.67	<b>90.74</b>	89.39	91.02
$\{(24, 3)\}^{riu2}$	86.00	92.50	93.40	93.48	97.43	96.67	86.42	88.26	87.01
$\{(8, 1), (16, 2)\}^{riu2}$	97.50	97.00	96.75	95.98	97.71	96.67	89.51	<b>92.05</b>	93.61
$\{(8, 1), (24, 3)\}^{riu2}$	95.50	96.00	96.85	92.72	97.71	96.67	88.27	90.53	92.84
$\{(16, 2), (24, 3)\}^{riu2}$	95.00	96.50	96.25	95.33	98.57	96.67	87.65	<b>92.05</b>	92.52
$\{(8, 1), (16, 2), (24, 3)\}^{riu2}$	97.00	97.00	96.50	96.09	98.86	96.67	88.89	92.80	94.02
$\{(8, 1)\}^{u2}$	97.00	97.50	96.40	95.54	97.71	95.00	<b>90.74</b>	91.29	95.31
$\{(16, 2)\}^{u2}$	<b>99.00</b>	<b>99.50</b>	<b>96.90</b>	96.41	98.86	96.67	88.27	90.91	<b>95.86</b>
$\{(24, 3)\}^{u2}$	92.00	95.50	92.65	95.00	<b>99.43</b>	93.33	90.12	90.53	93.26

Note: 50-LOO and 50-4fold mean rates on 50-class breakdown using leave-one-out and four cross-fold validation respectively. Dyn35 and Dyn++ are shortened for DynTex35 and DynTex++ datasets.

### 5.5.3.5 Assessing impact of max-pooling features: Recognition with EMDP descriptor

We conduct in this section several experiments for investigating the impact of max-pooling features on encoding MDP patterns. As validated in Section 5.5.3 that the configurations of  $riu2$  mapping and  $D-M/C$  integration reported the best performance, we just address these settings to compute EMDP descriptor.

It could be verified from Tables 5.6, 5.7, 5.8, 5.10 that EMDP descriptor is more discriminative than MDP thanks to the contribution of max-pooling features. Specifically, the performance of its single-scale variants has significantly improved in the recognition issue of 50-class schemes in the UCLA dataset. For instance, with  $\{(P, R)\} = \{(8, 1)\}$  of  $riu2$  mapping, EMDP obtains 1.5% better than that of MDP (see Tables 5.6, 5.10). In the setting chosen for comparison with the state of the art (i.e.,  $\{(8, 1), (16, 2), (24, 3)\}^{riu2}$ ), EMDP outperforms about 0.3% compared to MDP (99.43%) on Dyn-



Table 5.10: Recognition rates (%) of EMDP<sub>D-M/C</sub> descriptor and its multi-scale settings with mapping of *riu2* on DT datasets.

Dataset	UCLA				DynTex				
	50-LOO	50-4fold	9-class	8-class	Dyn35	Alpha	Beta	Gamma	Dyn++
$\{(P, R)\}$									
$\{(8, 1)\}$	99.50	98.50	98.40	97.07	97.71	95.00	95.68	92.80	95.17
$\{(16, 2)\}$	<b>100</b>	100	97.15	97.07	99.14	<b>98.33</b>	96.91	93.18	95.27
$\{(24, 3)\}$	99.50	99.50	98.25	98.04	<b>99.71</b>	95.00	96.91	93.56	94.67
$\{(8, 1), (16, 2)\}$	<b>100</b>	<b>100</b>	97.90	97.61	99.43	96.67	96.91	93.18	95.90
$\{(8, 1), (24, 3)\}$	<b>100</b>	<b>100</b>	98.55	98.26	99.43	96.67	96.91	93.56	95.66
$\{(16, 2), (24, 3)\}$	<b>100</b>	99.50	97.05	97.17	<b>99.71</b>	96.67	<b>97.53</b>	93.18	95.68
$\{(8, 1), (16, 2), (24, 3)\}$	<b>100</b>	<b>100</b>	<b>98.90</b>	<b>98.70</b>	<b>99.71</b>	96.67	96.91	<b>93.94</b>	<b>96.03</b>

Note: 50-LOO and 50-4fold denote rates on 50-class breakdown using leave-one-out and four cross-fold validation respectively. Dyn35 and Dyn++ are shortened for DynTex35 and DynTex++.

Tex35. Particularly, it gains 93.94% rate of recognition on the complicated dataset, Gamma, about 2% higher than MDP's. In terms of classification on DynTex++, the operation of EMDP looks more "stable" and achieves a little better rate with 96.03% in comparison to those of MDP with 95.86% (see Tables 5.8, 5.10).

In general, it is validated that the impact of the max-pooling features is positive in enhancing the performance of the proposed descriptors. Table 5.11 indicates the important contribution of these enhanced features in shallow analysis. It can be realized from this Table that it is possible to take advantage of these in the deeper max-pooling layers as well as to combine this computation with other advance components of CNNs in the further context.

Table 5.11: Contribution of max-pooling features for the performance (%) of descriptors using settings of  $D_{-M/C}$ , and  $\{(P, R)\} = \{(8, 1), (16, 2), (24, 3)\}$  with *riu2* mapping.

Descriptors	DynTex35	Gamma	DynTex++
MMDP	99.43	92.05	95.86
MMDP + "deep" features	99.71	91.30	95.85
MMDP + global features	99.14	<b>93.94</b>	95.34
MMDP + "deep" and global features (e.g., MEMDP)	<b>99.71</b>	<b>93.94</b>	<b>96.03</b>

#### 5.5.4 Global discussion

Based on the above experimental results on different benchmark DT datasets, it can be derived several general findings as follows.

- The proposed moment volume model can be judged as a filter bank approach for pre-processing techniques since its principle is a local filter in which its operator is inherited from the basic LBP concept with low computing costs (see Section 5.2.2 and 5.2.3) to exploit robust and discriminant features of DT videos. Outputs of this process, i.e., "mean" and "variance" videos, are regarded as complementary parts to boost the discriminative power of DT representation (see Table 5.4).
- Considering larger supporting volumes to construct moment volumes can be lead to outputs of blurred videos. This induces that encoding on these videos of our proposed operators reduces their performance due to the increase of noise patterns structured from the near uniform voxels. It can be seen from Tables 5.7, 5.8, 5.12 that the performance of DT descriptors are affected significantly by blurred videos dealt with by the model of two first-order moment volumes with large supporting regions  $\Omega = \{(14, 1), (14, 2)\}$ . Moreover, bigger elements of supporting volumes also increase the time cost of filtering voxel features without enhancing the operation of recognition as expected. In practice, the setting of regional volume  $\Omega = \{(6, 1)\}$  should be empirically recommended for

Table 5.12: Recognition rates (%) of MDP descriptors encoded on filtered videos with supporting elements of  $\Omega = \{(14, 1), (14, 2)\}$ .

Dataset	Beta (DynTex)			DynTex++		
$\{(P, R)\}^{riu2/u2}$	D_M	D_M_C	D_M/C	D_M	D_M_C	D_M/C
$\{(8, 1)\}^{riu2}$	93.21	93.21	93.83	92.74	93.44	93.76
$\{(16, 2)\}^{riu2}$	92.59	92.59	<b>95.06</b>	93.88	94.24	93.92
$\{(24, 3)\}^{riu2}$	<b>95.06</b>	94.44	93.21	94.04	93.96	93.07
$\{(8, 1), (16, 2)\}^{riu2}$	93.21	93.21	94.44	94.61	94.60	<b>94.82</b>
$\{(8, 1), (24, 3)\}^{riu2}$	93.83	93.83	94.44	94.27	94.54	94.58
$\{(16, 2), (24, 3)\}^{riu2}$	94.44	<b>95.06</b>	94.44	94.49	94.36	94.62
$\{(8, 1), (16, 2), (24, 3)\}^{riu2}$	94.44	93.83	94.44	<b>95.27</b>	<b>94.85</b>	94.70

Note: D\_M, D\_M\_C, and D\_M/C are different integrations of complemented components to form the corresponding MDP descriptors.

the proposed model of  $r$ -order moment volumes.

- Two proposed extensions for LDP operator resulting in the extended operator xLDP make our descriptor MDP even more robust and discriminative than the straightforward version MDP-B, which is based on LDP, in spite of the fact that this simple descriptor is also very competitive compared to the state-of-the-art results.
- MDP descriptors, based on the configuration of  $\{(8, 1), (16, 2), (24, 3)\}^{riu2}$ , have more substantial performance compared to others thanks to more relationships of local directional structures involved in.
- Directional complement of center contrast level LDP<sub>C</sub> has a trivial impact on improving the performance of DT descriptor in our framework (see Tables 5.6, 5.7, 5.8). Concatenating it to form the corresponding descriptor would just grow up 2 bins for each concerned direction, i.e.,  $L_{riu2/u2} + 2$ , while that would be double size in case of jointing, i.e.,  $2L_{riu2/u2}$ . Therefore, it is possible to make a trade-off between accuracy of recognition and the computing consumption in particular applications.

## 5.6 Summary

In this chapter, we have presented effective descriptors for DT representation, which are based on the filtering models of moment images/volumes. Therein, the filtered elements, computed by the moment-volume-based model, have been proved empirically more robustness to noise that allows to capture the proposed xLDP patterns to form the MDP-based descriptors with higher performance in comparison with addressing operator CLSP [29] for those based on the model of moment images. Due to turbulent motions of DTs, full directions should be addressed for the future works to entirely investigate the relations of local informative directions for an image texture. Furthermore, in consideration of treating the large dimension problem, encoding DT features with  $n$ -order MDP <sup>$n$</sup>  ( $n \geq 3$ ) operator on filtered sequences figured out by high-order moment volumes can obtain more robust spatio-temporal relationships to boost the discriminative power of DT description.



---



---

# CHAPTER 6

---

## REPRESENTATION BASED ON VARIANTS OF GAUSSIAN FILTERINGS

### Contents

<b>6.1</b>	<b>Introduction</b>	<b>76</b>
6.1.1	Motivation	76
6.1.2	A brief of our contributions	77
<b>6.2</b>	<b>Gaussian-based filtering kernels</b>	<b>77</b>
6.2.1	A conventional Gaussian filtering	77
6.2.2	Gradients of a Gaussian filtering kernel	78
<b>6.3</b>	<b>A novel kernel based on difference of Gaussian gradients</b>	<b>78</b>
6.3.1	Definition of a novel DoDG kernel	79
6.3.2	Beneficial properties of DoDG compared to DoG	80
<b>6.4</b>	<b>Representation based on completed hierarchical Gaussian features</b>	<b>81</b>
6.4.1	Construction of Gaussian-filtered CHILOP descriptor	81
6.4.2	Experiments and evaluations	83
<b>6.5</b>	<b>Representation based on RUBik Blurred-Invariant Gaussian features</b>	<b>86</b>
6.5.1	Benefits of Gaussian-based filterings	86
6.5.2	Construction of RUBIG descriptor	87
6.5.3	Experiments and evaluations	88
<b>6.6</b>	<b>Representation based on Gaussian-filtered CAIP features</b>	<b>89</b>
6.6.1	Completed sets of Gaussian-based filtered outcomes	90
6.6.2	Beneficial properties of filtered outcomes $\Omega_{\sigma, \sigma'}^{2D/3D}$	92
6.6.3	DT description based on complementary filtered outcomes $\Omega_{\sigma, \sigma'}^{2D/3D}$	93
6.6.4	Experiments and evaluations	95
<b>6.7</b>	<b>Representation based on oriented magnitudes of Gaussian gradients</b>	<b>98</b>
6.7.1	Oriented magnitudes of Gaussian gradients	99
6.7.2	DT representation based on oriented magnitudes	102
6.7.3	Experiments and evaluations	105
<b>6.8</b>	<b>Representation based on Gaussian-gradient features</b>	<b>110</b>
6.8.1	High-order Gaussian-gradient Filtered Components	112
6.8.2	DT Representation Based on $\Omega_{\mathcal{H}, \sigma}^{2D/3D}$ Components	114

6.8.3	Experiments and evaluations . . . . .	116
<b>6.9</b>	<b>Representation based on DoDG-filtered features . . . . .</b>	<b>122</b>
6.9.1	Construction of DoDG-filtered descriptors . . . . .	122
6.9.2	Experiments and evaluations . . . . .	125
<b>6.10</b>	<b>Comprehensive evaluations in comparison with existing methods . . . . .</b>	<b>129</b>
6.10.1	Benefits of Gaussian-based filterings . . . . .	130
6.10.2	Complexity of our proposed descriptors . . . . .	132
6.10.3	Comprehensive discussions of DT classification on different datasets . . . . .	133
<b>6.11</b>	<b>Global discussions . . . . .</b>	<b>138</b>
6.11.1	Further evaluations for Gaussian-gradient-based descriptors . . . . .	138
6.11.2	Evaluating appropriation of our proposals for real applications . . . . .	139
<b>6.12</b>	<b>Summary . . . . .</b>	<b>139</b>

---

## 6.1 Introduction

### 6.1.1 Motivation

Since early years of 90s, filter-bank approach has been addressed for texture analysis [31]. Recently, its denoising benefits has been consolidated in the LBP-based encoding for an effective description of texture images [2, 78]. Also mentioned in Section 2.6, many recent works [72, 74, 93, 115] have been proposed to take advantage of the filter-bank in order to reduce noise for DT representation. Inspired by the general framework that is based on filterings to mitigate the negative impacts of the well-known issues on DT encoding, in the previous Chapter 5, we have developed the discriminative MDP-based descriptors using the novel model of moment volumes as a filter to extract the robust responses. Moreover, also discussed in Section 2.7, the LBP-based methods have been potential solutions in encoding local patterns for DT representation thanks to their simplicity and effectiveness of computations. Motivated by those benefits, we propose efficient frameworks in which several robust filtering kernels are taken into account the video analysis as a pre-processing stage to point out filtered outcomes for local DT encoding. Accordingly, two main stages of the general framework in Figure 1.5 are addressed in this chapter as follows.

- For the filterings, different variants of Gaussian-based kernels have been investigated and thoroughly evaluated their ability in noise reduction: the typical Gaussian kernel, Difference of Gaussians (DoG), Gaussian gradient kernels, and especially a novel filtering kernel based on Difference of Derivative Gaussians (DoDG) (see Section 6.3.1 for its definition). Also, the influences of Gaussian-based filterings on DT representation are discussed thoroughly in multi-scale of standard deviations as well as multi-order of the Gaussian gradients. Furthermore, the Gaussian-based magnitude features and their oriented properties are also exploited in diverse aspects to provide more rich informative patterns for DT representation.
- For the local DT encoding, we utilize our proposed operators, e.g., CAIP (see Section 3.2), LRP (see Section 3.5), and CHILOP (see Section 3.6), in order to extract spatio-temporal features from the obtained filtering responses. In addition, to concentrate on evaluating how well the filtering executions of our proposals are working, we just use the basic operator CLBP [3] to encode the obtained outcomes for DT representation.

As a result, local discriminative descriptors are correspondingly constructed for DT recognition issues. Experiments have validated their significant results compared to state of the art. Among of them, with high performance in small dimension, the Gaussian-gradient-based descriptors (e.g., HoGF [J3] and DoDGF [S1]) are expected as one of appreciated solutions for mobile applications and embedded sensor systems which have demanded the restricted resources to execute their functions. In short, the Gaussian-based kernels along with the local operators, which are addressed in below sections to deal with the well-known issues of DT representation, can be outlined in Table 6.1 as follows.

Table 6.1: A brief of proposed descriptors based on Gaussian-based filterings.

#	Proposed descriptor	Filtering kernel	Local operator	Referred to
1	CHILOP [S2]	Typical Gaussian kernel	Proposed CHILOP (see Section 3.6)	Section 6.4
2	RUBIG [J4]	Typical Gaussian kernel, DoG	Proposed LRP (see Section 3.5)	Section 6.5
3	LOGIC [S4]	Typical Gaussian kernel, DoG	Proposed CAIP (see Section 3.2)	Section 6.6
4	SIOMF/SVOMF [S3]	Kernels of Gaussian gradients	CLBP [3], the popular operator	Section 6.7
5	HoGF [J3]	Kernels of Gaussian gradients	CLBP [3], the popular operator	Section 6.8
6	DoDGF [S1]	Novel DoDG (see Section 6.3.1)	CLBP [3], the popular operator	Section 6.9

## 6.1.2 A brief of our contributions

Efficiently, we have taken the Gaussian kernel, different variants of its partial derivatives along with our proposed local operators into account the video analysis for DT representation. Our significant contributions can be listed in short as follows.

- Representing DTs based on spatio-temporal features extracted from typical 2D/3D Gaussian-filtered outcomes using the basic CLBP [3] operator [C2, C5].
- Local Gaussian-based invariant characteristics for DT representation [S4]
- Completed hierarchical Gaussian-filtered patterns for DT classification [S2]
- Prominent local representation for DTs based on high-order Gaussian-gradients [J3]
- Representing DTs based on oriented magnitudes and separately bipolar-filtered features of Gaussian gradients [S3, S5]
- A novel difference of derivative Gaussians kernel for understanding DTs [S1]

## 6.2 Gaussian-based filtering kernels

### 6.2.1 A conventional Gaussian filtering

A conventional Gaussian filtering is a process of convolving a Gaussian kernel on a spatial domain. It should be in accordance with the regulation of a Gaussian distribution. Accordingly, let  $\gamma_n = \{x_i\}_{i=1}^n$  denote  $n$  spacial axes. A  $n$ -dimensional Gaussian filtering kernel is defined in general as

$$G_{\sigma}^n(\gamma_n) = \frac{1}{(\sigma\sqrt{2\pi})^n} \exp\left(-\frac{x_1^2 + x_2^2 + \dots + x_n^2}{2\sigma^2}\right) \quad (6.1)$$

where  $\sigma \in \mathbb{R}^+$  denotes a predefined standard deviation. Figure 6.1 at (a) and (b) shows an instance of the 2D Gaussian filtering of a given textural image with standard deviations  $\sigma \in \{0.7, 1.0\}$ . This typical Gaussian filtering is addressed as a preprocessing stage of the constructions of our proposed descriptors: LOGIC [S4] (see Section 6.6), CHILOP [S2] (see Section 6.4), and RUBIG [J4] (see Section 6.5).

According to the above definition of Gaussian filtering, the difference of two Gaussian filterings with  $\sigma$  and  $\sigma'$ , ( $\sigma < \sigma'$ ), is formulated as

$$\text{DoG}_{\sigma, \sigma'}^n(\gamma_n) = G_{\sigma}^n(\gamma_n) - G_{\sigma'}^n(\gamma_n) \quad (6.2)$$

Figure 6.1 (c) shows an example of a 2D DoG filtering of two responses of 2D Gaussian filterings with  $\sigma = 0.7$  and  $\sigma' = 1.0$ . Moreover, in order to obtain more filtered outcomes for DT encoding, Vu *et al.* [116] proposed to decompose a 2D DoG-filtered image  $\mathcal{I}_{bf}$  into two following bipolar-based images  $\mathcal{I}^+$  and  $\mathcal{I}^-$  as

$$\mathcal{I}_{bf}^+(\mathbf{q}) = \begin{cases} \mathcal{I}_{bf}(\mathbf{q}), & \text{if } \mathcal{I}_{bf}(\mathbf{q}) \geq \varepsilon \\ 0, & \text{otherwise.} \end{cases} \quad \text{and} \quad \mathcal{I}_{bf}^-(\mathbf{q}) = \begin{cases} |\mathcal{I}_{bf}(\mathbf{q})|, & \text{if } \mathcal{I}_{bf}(\mathbf{q}) \leq -\varepsilon \\ 0, & \text{otherwise.} \end{cases} \quad (6.3)$$

where  $\varepsilon$  is a micro-valued threshold to eliminate meaningless textural pixels caused by the closed-to-zero pixels of the corresponding DoG response. Figure 6.1 at (d) and (e) shows an example of decomposing a DoG-filtered image into two bipolar-based images with  $\varepsilon = 0.25$ .

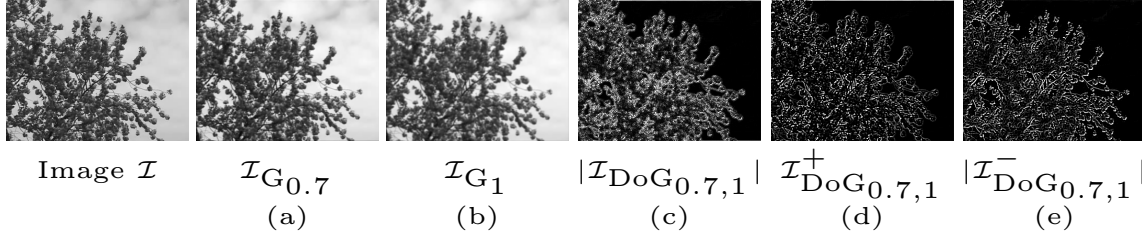


Figure 6.1: Responses of the 2D Gaussian-based filterings with deviations  $\sigma = 0.7$ ,  $\sigma = 1.0$ , and threshold  $\varepsilon = 0.25$  for decomposition of  $\mathcal{I}_{\text{DoG}_{0.7,1}}$ .

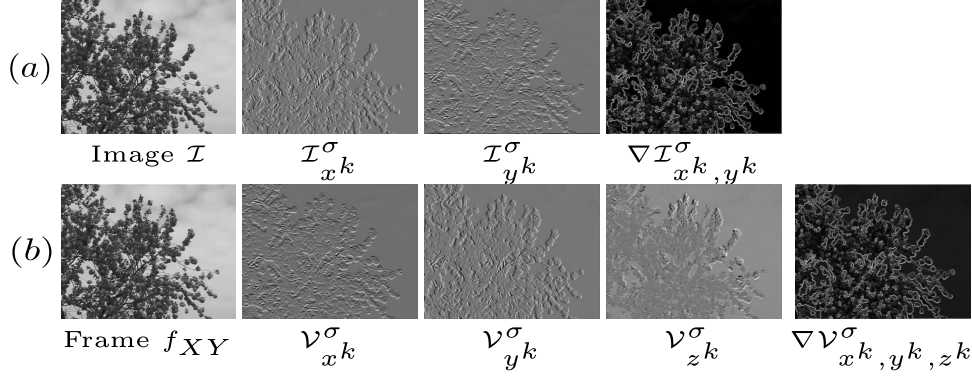


Figure 6.2: An instance of the 1<sup>st</sup>-order 2D/3D Gaussian-gradient filterings with  $\sigma = 0.7$ . Therein, (a) is for filtering a still image  $\mathcal{I}$  using a 2D gradient kernel, while (b) is for filtering a video  $\mathcal{V}$  using a 3D gradient kernel.

### 6.2.2 Gradients of a Gaussian filtering kernel

According to Equation 6.1, it could be conduced that a  $k$ -order partial derivative of  $G_\sigma^n(\gamma_n)$  with respect to a direction  $x_i \in \gamma_n$  is formed as

$$G_{\sigma, \partial x_i^k}^n(\gamma_n) = \frac{\partial^k G_\sigma^n(\gamma_n)}{\partial x_i^k} \quad (6.4)$$

in which “ $\partial$ ” denotes a gradient operation. Due to Equations 6.4 and 6.1, it can be seen that the Gaussian-gradient filtering points out  $n$  filtered outcomes subject to the partial derivative of each direction, while only one is done by the non-Gaussian-gradient filtering. In addition, the filtering in high-orders of Gaussian gradients could respond more robust filtered elements for DT representation. Figure 6.2 shows an instance of the 1<sup>st</sup>-order 2D/3D Gaussian-gradient filterings with  $\sigma = 0.7$ . This 2D/3D filtering is taken into account video analysis as a preprocessing stage of the constructions of our proposed descriptors: SIOMF/SVOMF [S3] (see Section 6.7) and HoGF [J3] (see Section 6.8).

## 6.3 A novel kernel based on difference of Gaussian gradients

The well-known DoG filtering kernel was exploited as a pre-processing stage in FoSIG [C2], V-BIG [C5] to reduce the negative impacts of the noise issues on DT representation. However, its performance is not as good as expected due to a lack of complementary filtered components involved in the DT encoding, i.e., only one DoG-filtered outcome (see Figure 6.4 line (a)) obtained by a DoG filtering operation with a pre-defined pair of standard deviations. To deal with this shortcoming, we hereafter introduce a novel DoDG filtering kernel with simple computation based on the difference of high-order Gaussian gradients in order to efficiently maintain invariant spatial features as well as forcefully capture discriminative information on various robust filtered outcomes for DT understanding.

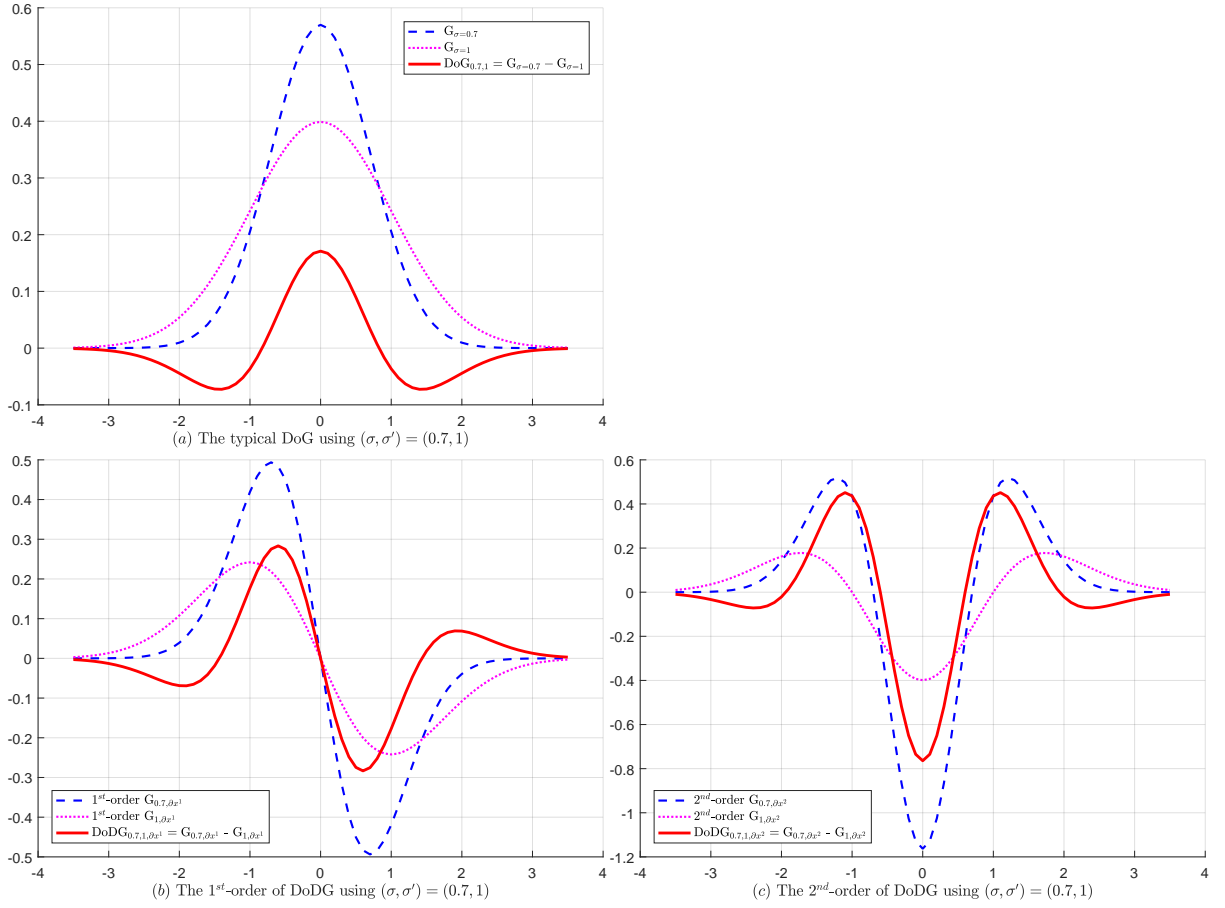


Figure 6.3: Profile of 1D DoG kernel (a) using a pre-defined pair of standard deviations  $(\sigma, \sigma') = (0.7, 1)$  compared to those of 1D DoG kernels at the first (b) and second (c) orders.

### 6.3.1 Definition of a novel DoDG kernel

Let  $(\sigma, \sigma')$  denote a pre-defined pair of standard deviations, so that  $0 < \sigma < \sigma'$ . Based on high-order gradients of a Gaussian kernel formulated as in Equation (6.4), a  $k$ -order filtering kernel of DoDG for a direction  $x_i \in \gamma_n$ , named  $\text{DoDG}_{\sigma,\sigma',\partial x_i^k}^n(\gamma_n)$ , is defined as the difference of two  $k$ -order Gaussian gradients corresponding to  $\sigma$  and  $\sigma'$  as follows.

$$\text{DoDG}_{\sigma,\sigma',\partial x_i^k}^n(\gamma_n) = G_{\sigma,\partial x_i^k}^n(\gamma_n) - G_{\sigma',\partial x_i^k}^n(\gamma_n) \quad (6.5)$$

Figure 6.3 at (b) and (c) respectively shows plots of the densities of  $\text{DoDG}^{1D}$  kernel in the first ( $k = 1$ ) and second ( $k = 2$ ) orders of  $(\sigma, \sigma') = (0.7, 1)$ . Appreciably, it can be deduced in general that the DoDG kernels for the spatial domain  $\gamma_n = \{x_i\}_{i=1}^n$  as

$$\begin{cases} \text{DoDG}_{\sigma,\sigma',\partial x_1^k}^n(\gamma_n) = G_{\sigma,\partial x_1^k}^n(\gamma_n) - G_{\sigma',\partial x_1^k}^n(\gamma_n) \\ \text{DoDG}_{\sigma,\sigma',\partial x_2^k}^n(\gamma_n) = G_{\sigma,\partial x_2^k}^n(\gamma_n) - G_{\sigma',\partial x_2^k}^n(\gamma_n) \\ \vdots \\ \text{DoDG}_{\sigma,\sigma',\partial x_n^k}^n(\gamma_n) = G_{\sigma,\partial x_n^k}^n(\gamma_n) - G_{\sigma',\partial x_n^k}^n(\gamma_n) \end{cases} \quad (6.6)$$

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



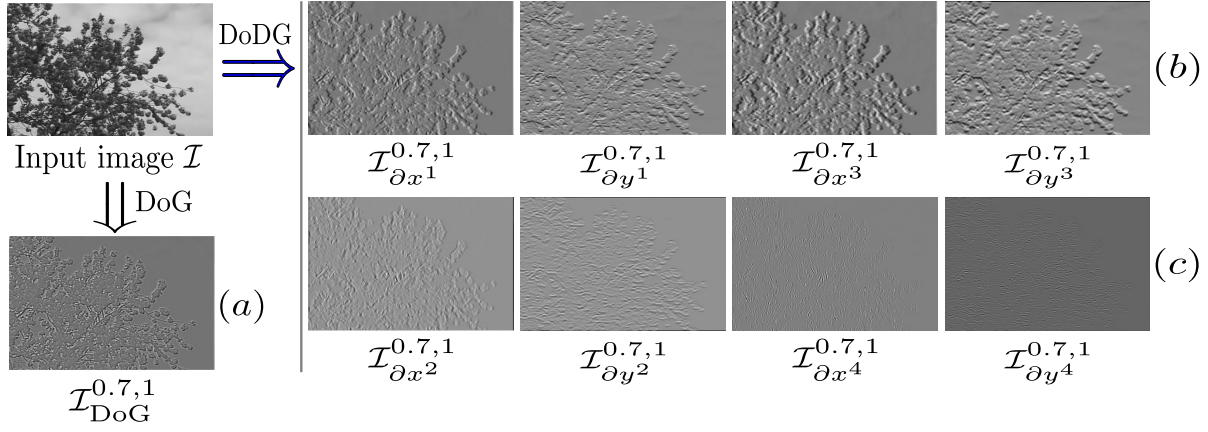


Figure 6.4: Instances of 2D Gaussian-based filterings for an given image  $\mathcal{I}$  using a pre-defined pair of standard deviations  $(\sigma, \sigma') = (0.7, 1)$ . Therein, (a): a DoG-filtered image of the conventional DoG<sup>2D</sup> filtering, (b) and (c): DoDG-based images of odd and even DoDG<sup>2D</sup> filterings respectively.

### 6.3.2 Beneficial properties of DoDG compared to DoG

Hereafter, we point out some beneficial properties of DoDG kernels for DT representation. For the simplicity of presentation, let us consider  $k$ -order DoDG kernels in 1D space, which their profiles are shown in Figure 6.3.

- When  $k$  is odd (see Figure 6.3(b)), the response of a DoDG kernel is semi-symmetric since  $\text{DoDG}_{\sigma, \sigma'}(x) = -\text{DoDG}_{\sigma, \sigma'}(-x)$ .
- When  $k$  is even, the response of a DoDG kernel is symmetric (see Figure 6.3(c)) since  $\text{DoDG}_{\sigma, \sigma'}(x) = \text{DoDG}_{\sigma, \sigma'}(-x)$ . Its response is somewhat similar to that of the DoG kernel (also see Figure 6.3(a)).
- Similar to the DoG-filtered outcome, our DoDG-filtered ones are also robust against changes of scales, illumination, and contrast by addressing the difference of two filtering scales.
- Being a Gaussian-based kernel, the DoDG kernel is naturally robust against noise.

Accordingly, DoDG kernels can be structured into 2 groups: odd and even order kernels. It is evident that those two groups are complementary since they exploit local features in a totally different way. A combination of those, which allows to take into account both symmetric and asymmetric features, enhances informative richness and discriminative power.

On the other hand, since the  $G_{\sigma, \partial x_i^k}^n$  filtering kernel has separable and linear properties, the computational complexity of our  $\text{DoDG}_{\sigma, \sigma', \partial x_i^k}^n$  is also inherited from those advantages. Those allow us to compute DoDG<sup>1</sup> in different partial derivatives to forcefully consider DoDG-filtered features in multi-scale analysis of higher orders. Figure 6.4 in lines (b) and (c) shows DoDG-filtered images obtained by using the DoDG<sup>2D</sup> filtering kernel with  $(\sigma, \sigma') = (0.7, 1)$  in four levels of partial derivatives, i.e.,  $k \in \{1, 2, 3, 4\}$ .

In addition, it is worth noting that the conventional DoG kernel can be also conducted as a degeneration of our novel DoDG kernel at the zero-order (i.e.,  $k = 0$ ). It means that Equation (6.5) can be rewritten for the band-pass filter DoG as

$$\text{DoDG}_{\sigma, \sigma', \partial x_i^k}^n(\gamma_n) = G_{\sigma, \partial x_i^k}^n(\gamma_n) - G_{\sigma', \partial x_i^k}^n(\gamma_n) \quad (6.7)$$

Consequently, it could be pointed out several crucial statements making a better execution of DoDG in noise reduction for understanding DTs compared to that of DoG as follows.

<sup>1</sup>A simple MATLAB code for high-order 2D/3D DoDG filtering kernels is available at <http://tpnguyen.univ-tln.fr/download/MATCodeDoDG>

- For filtering processes, each spatial domain in  $\gamma_n$  is often truncated by a scale range of  $[-3\sigma, 3\sigma]$  for the convolving operation to optimally capturing the energy of Gaussian distribution. Figure 6.3 illustrates a graphical view of exploiting both DoG and DoDG kernels to filter an image with a specific pair of standard deviations  $(\sigma, \sigma') = (0.7, 1)$ . Accordingly, it can be visually realized that our DoDG has figured out less zero-bipolar features than DoG, those which make the encoding more sensitive to noise caused by the closed-to-zero pixels of the filtered outcomes.
- Our DoDG has pointed out more diversity of bipolar filtered-image partitions than DoG (see Figure 6.3), allowing to capturing forceful features for DT representation.
- Also, conducted from Figure 6.3(b) and (c), our DoDG could maintains invariant spatial information in better stable frequencies thanks to an adaptive conservation of DoDG's distribution in accordance with that of the concerning Gaussian gradients. In the meanwhile, it is not for DoG since the subtraction of non-Gaussian-gradient filterings is agreed with an approximation of the Laplacian of Gaussian (LoG) (see Figure 6.3(a)).
- Furthermore, it can be verified from Equations (6.2) and (6.6) that for a pre-defined pair of  $(\sigma, \sigma')$  taken into account a filtering process, our novel DoDG kernel could figure out more complementary filtered outcomes than the only one done by the DoG kernel (see Figure 6.4 for an instance of these filterings). This allows to forcefully investigate DoDG-filtered features for further enhancement.

In order to validate above advantageous points, both DoG and DoDG are addressed for video analysis as a pre-processing step to handle the well-known issues of DT description (see Section 6.9). After that, the obtained results in DT recognition are thoroughly discussed in Sections 6.9.2.2 and 6.9.2.3.

## 6.4 Representation based on completed hierarchical Gaussian features

### 6.4.1 Construction of Gaussian-filtered CHILOP descriptor

In this section, a simple framework for video representation is introduced by taking our CHILOP operator into account efficiently shape and motion cues of DTs, as graphically illustrated in Figure 6.5. Accordingly, the proposed framework takes the following steps for adeptly analyzing an input video  $\mathcal{V}$ : First, video  $\mathcal{V}$  is splitted into sets of plane images  $\{f_{XY}\}$ ,  $\{f_{XT}\}$ , and  $\{f_{YT}\}$  subject to its three orthogonal planes  $\{XY, XT, YT\}$ . Secondly, CHILOP is used for resolving plane images in order to completely capture hierarchical spatio-temporal characteristics based on a set of multi-layer supporting regions  $\mathcal{D}$ . Similar to other LBP-based variants, the performance of CHILOP may be reduced due to the negative impacts of changes of environmental elements, illumination and noise. To deal with them, inspired by our prior work of Gaussian filtering in [C2], we take a  $n$ -dimensional Gaussian kernel into account filtering plane images as a pre-processing step. According to that, let  $\mathcal{F} = \{\sigma_1, \sigma_2, \dots, \sigma_m\}$  be a set of pre-defined standard deviations. For each plane image  $\mathcal{I}_{XY} \in f_{XY}$ ,  $\mathcal{I}_{XT} \in f_{XT}$ , and  $\mathcal{I}_{YT} \in f_{YT}$ , the corresponding Gaussian-filtered images are figured out as

$$\begin{cases} \mathcal{I}_{XY}^{\mathcal{G}_{\sigma_i}^n} = \mathcal{G}_{\sigma_i}^n(\varphi_n) * \mathcal{I}_{XY} \\ \mathcal{I}_{XT}^{\mathcal{G}_{\sigma_i}^n} = \mathcal{G}_{\sigma_i}^n(\varphi_n) * \mathcal{I}_{XT} \\ \mathcal{I}_{YT}^{\mathcal{G}_{\sigma_i}^n} = \mathcal{G}_{\sigma_i}^n(\varphi_n) * \mathcal{I}_{YT} \end{cases} \quad (6.8)$$

where  $\sigma_i \in \mathcal{F}$ , “\*” denotes a convolutional operator. Therefore, instead of analyzing the raw plane images of  $\{f_{XY}\}$ ,  $\{f_{XT}\}$ , and  $\{f_{YT}\}$ , our CHILOP is exploited for their Gaussian-filtered images in order to structure Completed Hierarchical Local Gaussian-filtered Patterns ( $\text{CHILOP}_{\nabla, \mathcal{D}}^{\mathcal{G}_{\mathcal{F}}^n}$ ) with more discriminative power. Finally, the obtained histograms are concatenated and normalized to produce a robust descriptor for DT representation as

$$\text{CHILOP}_{\nabla, \mathcal{D}}^{\mathcal{G}_{\mathcal{F}}^n}(\mathcal{V}) = \left[ \Psi_{\nabla, \mathcal{D}}(f_{XY}^{\mathcal{G}_{\sigma_i}^n}), \Psi_{\nabla, \mathcal{D}}(f_{XT}^{\mathcal{G}_{\sigma_i}^n}), \Psi_{\nabla, \mathcal{D}}(f_{YT}^{\mathcal{G}_{\sigma_i}^n}) \right]_{i=1}^{|\mathcal{F}|} \quad (6.9)$$

where  $\Psi$  stands for our CHILOP operator,  $|\mathcal{F}|$  is the cardinality of standard deviations in  $\mathcal{F}$  involved with the Gaussian filterings. In the meanwhile,  $\{f_{XY}^{\mathcal{G}_{\sigma_i}^n}\}$ ,  $\{f_{XT}^{\mathcal{G}_{\sigma_i}^n}\}$ , and  $\{f_{YT}^{\mathcal{G}_{\sigma_i}^n}\}$  are sets of Gaussian-filtered

images corresponding to results of utilizing Equation (6.8) on the raw plane images of  $\{f_{XY}\}$ ,  $\{f_{XT}\}$ , and  $\{f_{YT}\}$  respectively.

Thanks to above construction, our proposed descriptor  $\text{CHIOP}_{\nabla, \mathcal{D}}^{\mathcal{G}_{\mathcal{F}}^n}$  has the following beneficial properties in order to enhance the performance:

- $\text{CHIOP}_{\nabla, \mathcal{D}}^{\mathcal{G}_{\mathcal{F}}^n}$  descriptor is structured by the proposed CHIOP operator allowing to adequately describe shape and motion cues of DTs in consideration of a complete context of hierarchical local regions. In the meanwhile, just hierarchical features are involved with our prior descriptor HILOP [C3] (see Figure 6.7 for a specific comparison of their performances).
- Taking multi-layer supporting regions  $\mathcal{D}$  into account the CHIOP encoding, the further hierarchical features of  $\text{CHIOP}_{\nabla, \mathcal{D}}^{\mathcal{G}_{\mathcal{F}}^n}$  are boosted for improvement of their discriminative information (see Table 6.4).
- CHIOP features captured from the filtered plane images of  $\{f^{\mathcal{G}_{\sigma_i}}\}$  are more insensitive to noise compared to those encoded from the raw images (see Tables 6.3, 6.5, and 6.6). It should be noted that the Gaussian filtering kernel in this work is directly convoluted on the whole images, contrariwise to [108] in which the filtering is calculated on neighborhoods of a pixel in different local areas for description of textural images.
- Constructing  $\text{CHIOP}_{\nabla, \mathcal{D}}^{\mathcal{G}_{\mathcal{F}}^n}$  descriptor based on multi-scale of Gaussian-filtered plane images allows to forcefully structure spatio-temporal patterns with more robustness against the well-known problems: changes of environmental elements, illumination and noise, etc. In the meanwhile, FoSIG [C2] exploits CLBP [3] for encoding Gaussian-filtered plane images, but in single-scale analysis of Gaussian filtering kernel as well as lack of hierarchical local properties.

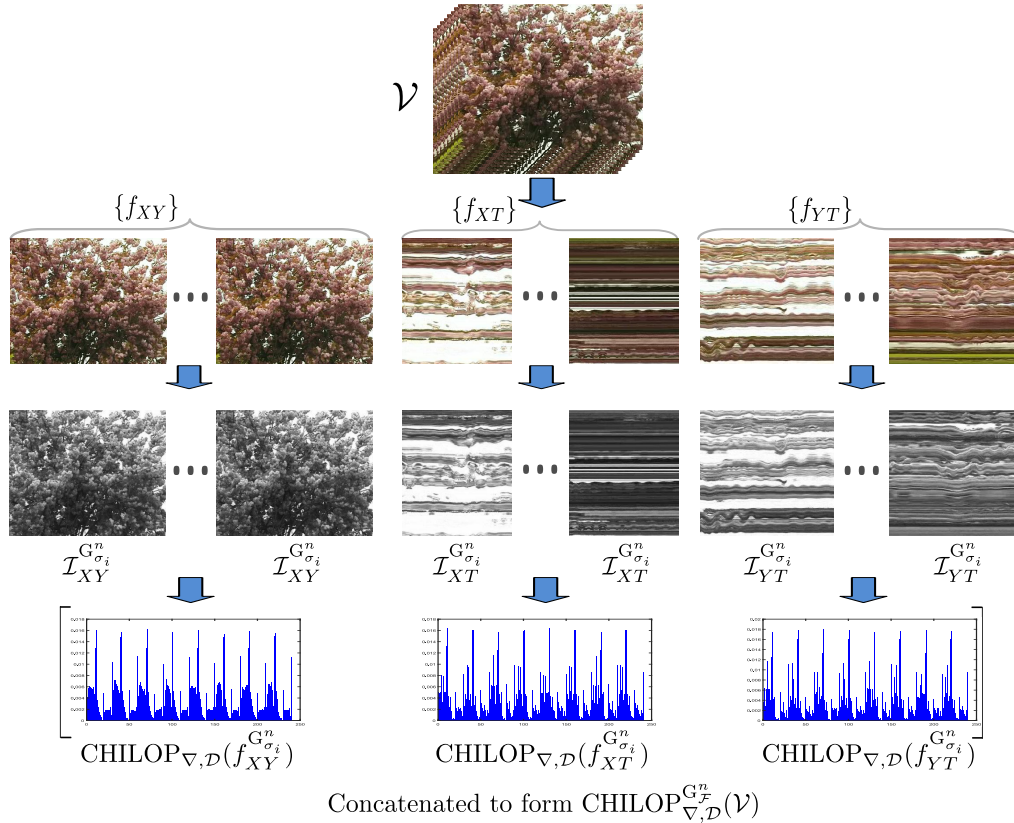


Figure 6.5: Our proposed framework of encoding  $\text{CHIOP}_{\nabla, \mathcal{D}}^{\mathcal{G}_{\mathcal{F}}^n}$  descriptor.

## 6.4.2 Experiments and evaluations

### 6.4.2.1 Parameters for experimental implementation

**Settings for the Gaussian filtering:** In order to figure out filtered images, we conduct a 2D Gaussian filtering kernel, i.e.,  $G_{\sigma}^{2D}(x, y)$  in which  $x, y \in [-3\sigma, 3\sigma]$ . Empirically, we address a set of pre-defined standard derivations,  $\mathcal{F} = \{0.5, 0.7, 1.0, 1.3, 1.5\}$ , for an investigation of the Gaussian filtering in multi-scale analysis in order to productively reduce the negative impacts of environmental changes, illumination and noise on the CHILOP encoding.

**Settings for CHILOP $_{\nabla, \mathcal{D}}^{G_{\mathcal{F}}^{2D}}$  descriptor:** In conformity with the LBP-based encoding, each supporting region  $\Omega_k \in \mathcal{D}$  is located by  $P_k$  local neighbors that are interpolated on a circle of radius  $R_k$  centered at  $\mathbf{q}_c$ , i.e.,  $\Omega_k = (P_k, R_k)$ . Specifically, we address  $\mathcal{D} = \{(8, 1), (8, 2), (8, 3), (8, 4)\}$  (i.e.,  $P_k = 8, \forall k$ ) for a further investigation of hierarchical areas. For a single plane image, to structure CHILOP $_{\nabla, \mathcal{D}}^{G_{\mathcal{F}}^{2D}}$  descriptor in reasonable dimension for DT classification issue, the *riu2* mapping is addressed for two complementary components of CHILOP operator, i.e.,  $\mathcal{L}_H(\cdot)$  and  $\mathcal{L}_M(\cdot)$  patterns. The obtained features are then jointed in two ways of  $\nabla = \{H\_M/C, H\_M/C\}$  with  $t_{H\_M/C} = 3P_k(P_k + 2)$  bins and  $t_{H/M/C} = 2P_k(P_k + 2)^2$  correspondingly. In order to construct a video descriptor, we address CHILOP operator on its three orthogonal planes, the obtained histograms are then concatenated to form a final representation. As the result of above those, the conclusive dimension of CHILOP $_{\nabla, \mathcal{D}}^{G_{\mathcal{F}}^{2D}}$  for a video description is subject to a number of Gaussian filtering scales (i.e.,  $|\mathcal{F}|$ ) and hierarchical regions (i.e.,  $|\mathcal{D}|$ ) that are currently involved in a CHILOP encoding on plane images of the video. That means  $3 \times |\mathcal{F}| \times (|\mathcal{D}| - 1) \times t_{\nabla}$  bins for a concerned integration in  $\nabla$ . Therein,  $t_{\nabla} \in \{t_{H\_M/C}, t_{H/M/C}\}$ . For instance, in order to encode a video based on a two-adjacent-hierarchical supporting region  $\mathcal{D} = \{(8, 1), (8, 2)\}$  along with a single-scale Gaussian filtering (i.e.,  $|\mathcal{F}| = 1$  and  $|\mathcal{D}| = 2$ ), it takes 720 and 4800 dimensions for  $H\_M/C$  and  $H/M/C$  descriptors respectively (see Table 6.2 for comparison with other LBP-based descriptors). Furthermore, for a strict assessment of our CHILOP's outperformance in comparison with that of the basic CLBP [3], it is better to take both of them into account extracting DT features from the raw plane images of a video  $\mathcal{V}$  (i.e.,  $\{f_{XY}\}$ ,  $\{f_{XT}\}$ , and  $\{f_{YT}\}$ ) in order to structure two corresponding descriptors CHILOP $_{\nabla, \mathcal{D}}(\mathcal{V})$  and CLBP $_{P, R}(\mathcal{V})$  with the same *riu2* mapping for both, while integrating techniques of  $\{S\_M/C, S/M/C\}$  for the CLBP patterns (refer to specific settings in Table 6.3).

### 6.4.2.2 Assessments of CHILOP's performances

We thoroughly discuss the effectiveness of CHILOP operator in encoding hierarchical spatio-temporal patterns for DT representation based on both raw features and Gaussian-filtered properties of plane images in a video. According to that, two corresponding descriptors CHILOP $_{\nabla, \mathcal{D}}$  and CHILOP $_{\nabla, \mathcal{D}}^{G_{\mathcal{F}}^{2D}}$  are constructed using the parameters designated in Section 6.4.2.1. Experiments for DT classification on benchmark datasets have verified that both of them significantly outperform compared to the basic CLBP [3]. It is thanks to a completed consideration of CHILOP in multi-hierarchical supporting areas. Furthermore, taking advantage of Gaussian-filtered characteristics, the performance of CHILOP $_{\nabla, \mathcal{D}}^{G_{\mathcal{F}}^{2D}}$  is better and more "stable" than that of CHILOP $_{\nabla, \mathcal{D}}$ . Hereafter, the effectiveness of CHILOP is assessed in detail as follows.

- As expected in Sections 3.6.1 and 3.6.3, encoding shape and motion cues of DTs in consideration of local relationships on hierarchical supporting regions has figured out robust descriptors with promising discrimination. Indeed, Figure 6.6 indicates that the performance lines of CHILOP $_{\nabla, \mathcal{D}}$  for a raw DT description are over those of CLBP (see Table 6.3 for their specific rates on several schemes). That means the general operator CHILOP has more discriminative power than its degeneration, i.e., CLBP (see Section 3.6.2).
- Taking multi-layer analysis into account structuring in higher-hierarchical supporting areas is able to capture more forceful patterns for enhancing the performance. Absolutely, experimental results

Table 6.2: A comparison of various dimensions of LBP-based descriptors.

Method	#bins	$P = 4$	$P = 8$	$P = 16$	$P = 24$
LBP-TOP <sup>riu2</sup> [14]	$3(P(P-1)+3)$	-	177	729	1665
VLBP [14]	$2^{3P+2}$	16384	-	-	-
CVLBP [91]	$3 \times 2^{3P+2}$	32768	-	-	-
HLBP [92]	$6 \times 2^P$	-	1536	-	-
WLBPC [109]	$6 \times 2^P$	-	1536	-	-
MEWLSP [95]	$6 \times 2^P$	-	1536	-	-
CVLBC [90]	$2(3P+3)^2$	-	1458	5202	11125
CLSP-TOP <sup>riu2</sup> [C1]	$6(P+2)^2$	-	600	1944	4056
CSAP-TOP <sup>riu2</sup> [J1]	$12(P+2)^2$	-	1200	3888	8112
FDT <sup>u2</sup> [C4]	$216P((P-1)+3)$	-	12744	-	-
FD-MAP <sup>u2</sup> <sub>L=2</sub> [C4]	$216P((P-1)+3)+16$	-	12760	-	-
HILOP [C3]	$3P(P(P-1)+3)$	-	1416	-	-
FoSIG [C2]	$12(P+2)^2$	-	1200	-	-
V-BIG [C5]	$12(P+2)^2$	-	1200	-	-
CHILOP <sub>H/M/C</sub> [S2]	$2P(P+2)^2$	-	4800	-	-
DDTP <sup>riu2</sup> <sub>D-M/C</sub> [J2]	$12(P+7)(P+2)$	-	1800	4968	9672
RUBIG [J4]	$36(P+2)^2$	-	3600	-	-
VOM-based [S3]	$72(P+2)$	-	720	-	-
IOM-based [S3]	$216(P+2)$	-	2160	-	-
MDP <sup>riu2</sup> <sub>D-M/C</sub> [J5]	$72(P+2)$	-	720	1296	1872
HoGF <sup>2D</sup> [J3]	$36(P+2)^2$	-	3600	-	-
HoGF <sup>3D</sup> [J3]	$48(P+2)^2$	-	4800	-	-
DoDGF <sup>2D</sup> [S1]	$24(P+2)^2$	-	2400	-	-
DoDGF <sup>3D</sup> [S1]	$36(P+2)^2$	-	3600	-	-

Note:  $P$  denotes the concerned neighbors. “-” means either “not available” or not been implemented experimentally. Dimension of all above descriptors is referred to their basic parameters used for encoding a given video. For DDTP<sup>riu2</sup><sub>D-M/C</sub> [J2], a directional beam of  $|B| = P = 8$  neighbors in consideration of dense trajectories with length  $L = 2$ .

Table 6.3: Comparison of performances (%) between CHILOP and CLBP [3] in encoding DT features based on the raw plane images of a video using *riu2* mapping with two popular kinds of incorporation on local supporting regions  $\mathcal{D} = \{(8, 1), (8, 2)\}$ .

Descriptor	9-class	8-class	DynTex35	Beta	DynTex++
CLBP <sub>S-M/C</sub> [3]	94.60	95.43	97.14	88.27	89.32
<b>Our</b> CHILOP <sub>H-M/C</sub>	97.85	98.48	98.86	93.83	94.39
CLBP <sub>S/M/C</sub> [3]	97.80	96.09	99.43	90.74	95.59
<b>Our</b> CHILOP <sub>H/M/C</sub>	<b>98.50</b>	<b>99.02</b>	<b>99.71</b>	<b>95.68</b>	<b>96.54</b>

Note: “ $S_{-M/C}$ ” and “ $S_{/M/C}$ ” respectively denote 2D and 3D jointing histograms of CLBP’s components, i.e., CLBP<sub>S</sub>, CLBP<sub>M</sub>, and CLBP<sub>C</sub>.

in Table 6.4 have verified that CHILOP obtains the best execution when being exploited on three adjacent regional hierarchies of a pixel, i.e.,  $\mathcal{D} = \{(8, 1), (8, 2), (8, 3)\}$ . In the meanwhile, on the larger hierarchical local regions, it can face with the negative influence of turbulent motions of DTs due to a decline of their spatio-temporal textural features in further areas.

- Thanks to considering hierarchical supporting areas in a completed context of local encoding (see Section 3.6.1), our CHILOP operator significantly outperforms compared to HILOP [C3] in which only one kind of hierarchical patterns is involved in (see Figure 6.7).
- It can be verified from Tables 6.5 and 6.6 that the CHILOP’s components ( $\mathcal{L}_H$ ,  $\mathcal{L}_M$ , and  $\mathcal{L}_C$ ) in the 3D-joint way (i.e., using incorporation of  $\nabla = \{H/M/C\}$ ) significantly improve the discrimination power of the corresponding obtained descriptor. This is in accordance with other local completed operators based on complementary components in their construction such as CLBP [3], LRP [J4], CLBC [82], etc.

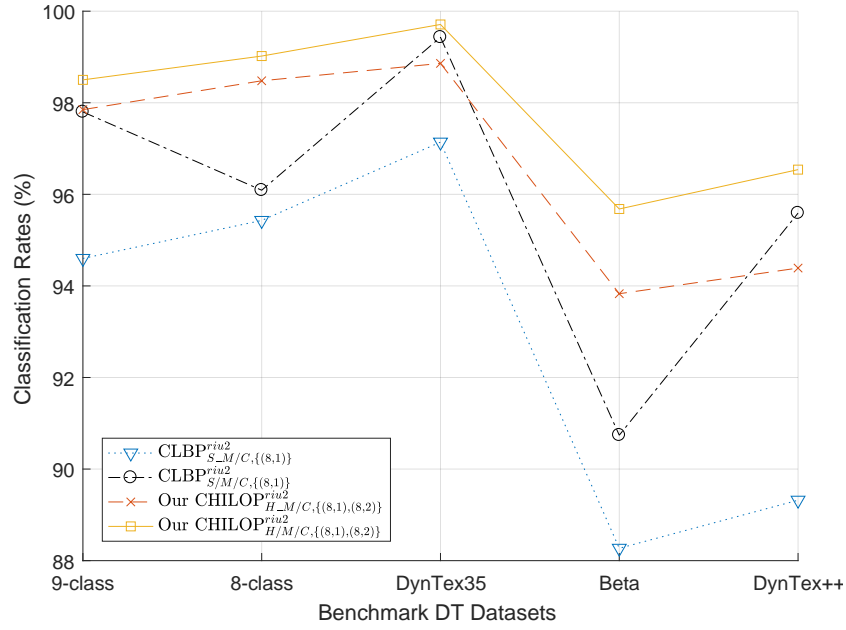


Figure 6.6: Prominent performances of our CHILOP for a raw DT description compared to CLBP’s [3].

Table 6.4: Rates (%) of  $\text{CHILOP}_{\nabla, \mathcal{D}}^{\mathcal{G}_{\mathcal{F}}^{2D}}$  in multi-layer of hierarchical regions using settings of Gaussian filtering  $\mathcal{F} = \{0.5, 1\}$  and jointing type  $\nabla = \{H/M/C\}$ .

$\mathcal{D} = \{(P, \{R\})\}$	UCLA				DynTex				Dyn++
	50-LOO	50-4fold	9-class	8-class	Dyn35	Alpha	Beta	Gamma	
$\{(8, \{1, 2\})\}$	<b>100</b>	<b>100</b>	98.65	98.70	99.43	<b>96.67</b>	93.83	94.32	97.67
$\{(8, \{1, 2, 3\})\}$	<b>100</b>	<b>100</b>	<b>99.45</b>	<b>99.02</b>	<b>99.71</b>	<b>96.67</b>	95.68	<b>94.70</b>	<b>98.06</b>
$\{(8, \{1, 2, 3, 4\})\}$	<b>100</b>	<b>100</b>	98.50	98.26	<b>99.71</b>	95.00	<b>96.91</b>	93.94	97.83

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

- As expected in Section 6.4.1, it can be verified that addressing the Gaussian filtering in the proposed framework makes the obtained descriptors more robust against the negative influences of environmental changes, illumination and noise. Indeed, following the best number of hierarchical regions, i.e.,  $\mathcal{D} = \{(8, 1), (8, 2), (8, 3)\}$  (see Table 6.4), we conduct an analysis of Gaussian filtering in constructing  $\text{CHILOP}_{\nabla, \mathcal{D}}^{\mathcal{G}_{\mathcal{F}}^{2D}}$  descriptors. Experimental results in Tables 6.5 and 6.6 have validated that  $\text{CHILOP}_{\nabla, \mathcal{D}}^{\mathcal{G}_{\mathcal{F}}^{2D}}$  has better and more “stable” performance in comparison with that of  $\text{CHILOP}_{\nabla, \mathcal{D}}$  in which CHILOP operator is exploited for capturing DT features on the raw plane images. It should be noted that the Gaussian filtering is also addressed in [C2] to form FoSIG descriptor, but its ability is just at a modest level due to lack of hierarchical information of DTs (see Figure 6.7).
- It can be seen from Tables 6.5 and 6.6 that the multi-scale Gaussian filtered encoding allows to capture more scale-information in order to enhance the performance. Therein, incorporation of two Gaussian filtering scales of  $\mathcal{F} = \{0.5, 1\}$  has figured out DT features with more discrimination than the others.
- Furthermore, the experiments have also indicated that CHILOP can decently resist changes of environmental elements, illumination and noise when encoding spatio-temporal features on the raw plane images for DT representation (see Tables 6.5, 6.6, and Figure 6.7).

Based on above assessments, we found out the best configurations for our proposed framework in encoding spatio-temporal patterns for  $\text{CHILOP}_{\nabla, \mathcal{D}}^{\mathcal{G}_{\mathcal{F}}^{2D}}$  and  $\text{CHILOP}_{\nabla, \mathcal{D}}$  descriptors as follows:  $\nabla = \{H/M/C\}$ ,  $\mathcal{F} = \{0.5, 1\}$ , and  $\mathcal{D} = \{(8, 1), (8, 2), (8, 3)\}$ . In general, the performance of our proposed

Table 6.5: Classification rates (%) on UCLA of  $\text{CHIOP}_{\nabla, \mathcal{D}}^{\mathcal{G}_{\mathcal{F}}^{2D}}$  and  $\text{CHIOP}_{\nabla, \mathcal{D}}$  descriptors.

	50-LOO		50-4fold		9-class		8-class	
$\mathcal{F} = \{\sigma_i\}$	$H\_M/C$	$H/M/C$	$H\_M/C$	$H/M/C$	$H\_M/C$	$H/M/C$	$H\_M/C$	$H/M/C$
$\{\emptyset\}$	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	98.40	98.10	96.30	98.26
$\{0.5\}$	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	98.65	98.05	97.72	96.52
$\{0.7\}$	<b>100</b>	<b>100</b>	99.50	<b>100</b>	97.90	98.80	98.15	97.93
$\{1.0\}$	99.50	<b>100</b>	99.50	<b>100</b>	<b>99.45</b>	98.80	98.59	98.70
$\{1.3\}$	99.50	<b>100</b>	99.50	<b>100</b>	98.45	98.20	<b>99.13</b>	96.74
$\{1.5\}$	99.50	<b>100</b>	99.50	<b>100</b>	99.25	97.85	98.59	97.93
$\{0.5, 0.7\}$	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	98.75	99.20	98.26	98.26
$\{0.5, 1.0\}$	99.50	<b>100</b>	99.50	<b>100</b>	98.35	<b>99.45</b>	98.37	<b>99.02</b>
$\{0.5, 1.3\}$	99.50	<b>100</b>	99.50	<b>100</b>	99.10	99.35	99.02	98.70
$\{0.5, 1.5\}$	99.50	<b>100</b>	99.50	<b>100</b>	98.40	99.30	98.70	98.26

Note: 50-LOO and 50-4fold denote results on 50-class breakdown using leave-one-out and four cross-fold validation.  $\{\emptyset\}$  indicates rates of  $\text{CHIOP}_{\nabla, \mathcal{D}}$  without Gaussian filtering involved in the DT encoding.

Table 6.6: Rates (%) on DynTex and DynTex++ of  $\text{CHIOP}_{\nabla, \mathcal{D}}^{\mathcal{G}_{\mathcal{F}}^{2D}}$  and  $\text{CHIOP}_{\nabla, \mathcal{D}}$  descriptors.

	DynTex35		Alpha		Beta		Gamma		DynTex++	
$\mathcal{F} = \{\sigma_i\}$	$H\_M/C$	$H/M/C$	$H\_M/C$	$H/M/C$	$H\_M/C$	$H/M/C$	$H\_M/C$	$H/M/C$	$H\_M/C$	$H/M/C$
$\{\emptyset\}$	97.43	<b>99.71</b>	<b>96.67</b>	<b>96.67</b>	<b>95.06</b>	95.68	91.67	93.56	95.51	96.62
$\{0.5\}$	97.71	99.43	95.00	<b>96.67</b>	91.36	95.68	92.05	93.56	97.13	96.34
$\{0.7\}$	98.29	99.14	95.00	<b>96.67</b>	91.36	95.06	90.91	94.32	94.65	96.29
$\{1.0\}$	<b>98.86</b>	99.43	95.00	<b>96.67</b>	90.74	93.83	89.39	<b>95.08</b>	96.28	93.10
$\{1.3\}$	98.57	99.43	95.00	95.00	90.74	94.44	89.39	93.94	93.79	95.54
$\{1.5\}$	98.29	99.43	95.00	95.00	90.74	93.83	90.15	93.56	93.16	94.83
$\{0.5, 0.7\}$	98.00	99.43	<b>96.67</b>	<b>96.67</b>	91.98	95.06	91.29	93.94	97.39	96.89
$\{0.5, 1.0\}$	<b>98.86</b>	<b>99.71</b>	95.00	<b>96.67</b>	91.98	95.68	90.91	94.70	<b>98.53</b>	<b>98.06</b>
$\{0.5, 1.3\}$	98.57	<b>99.71</b>	95.00	<b>96.67</b>	92.59	<b>96.30</b>	92.05	94.70	97.67	96.58
$\{0.5, 1.5\}$	98.29	<b>99.71</b>	95.00	<b>96.67</b>	92.59	95.68	<b>92.42</b>	94.70	97.53	96.49

Note:  $\{\emptyset\}$  indicates rates of  $\text{CHIOP}_{\nabla, \mathcal{D}}$  without Gaussian filtering involved in the DT encoding.

descriptors is mostly more efficient than that of all non-deep-learning methods (see Table 6.33). In terms of comparison with deep-learning approaches, our proposal is better than those in DT recognition on UCLA and nearly the same ability on DynTex++, but not on DynTex. Hereunder, we evaluate in detail the effectiveness of  $\text{CHIOP}_{\nabla, \mathcal{D}}^{\mathcal{G}_{\mathcal{F}}^{2D}}$  and  $\text{CHIOP}_{\nabla, \mathcal{D}}$  for DT classification on specific datasets, in which if particular settings for  $\text{CHIOP}_{\nabla, \mathcal{D}}^{\mathcal{G}_{\mathcal{F}}^{2D}}$  and  $\text{CHIOP}_{\nabla, \mathcal{D}}$  are not explicitly indicated, the best configurations are addressed for comprehensive evaluations in comparison with state of the art.

## 6.5 Representation based on RUBik Blurred-Invariant Gaussian features

### 6.5.1 Benefits of Gaussian-based filterings

Filter-bank approach, which has been early applied for texture analysis since years of 90s [31], was also considered for DT representation in recent works [72, 93, 115, C2, C5]. Moreover, filter-bank and LBP-based approaches have been also addressed together in [2] for an effective texture representation. Inspired from this approach, we address Gaussian-based filters to overcome well-known issues in DT description: the influence of noise, changes of environments, scales and illumination, etc. Indeed, two complementary families of filtering are taken into account for this purpose. First, Gaussian filters  $\mathcal{G}_{\sigma}^n$  are used to produce blurred volumes  $\mathcal{V}^G$  which are more robust against noise. Second, DoG filters are addressed to figure out invariant volumes  $\mathcal{V}^{DoG}$  which is robust against changes of illuminations and scales. It should be noted that Gaussian distribution has been also used in a totally different way in [117]



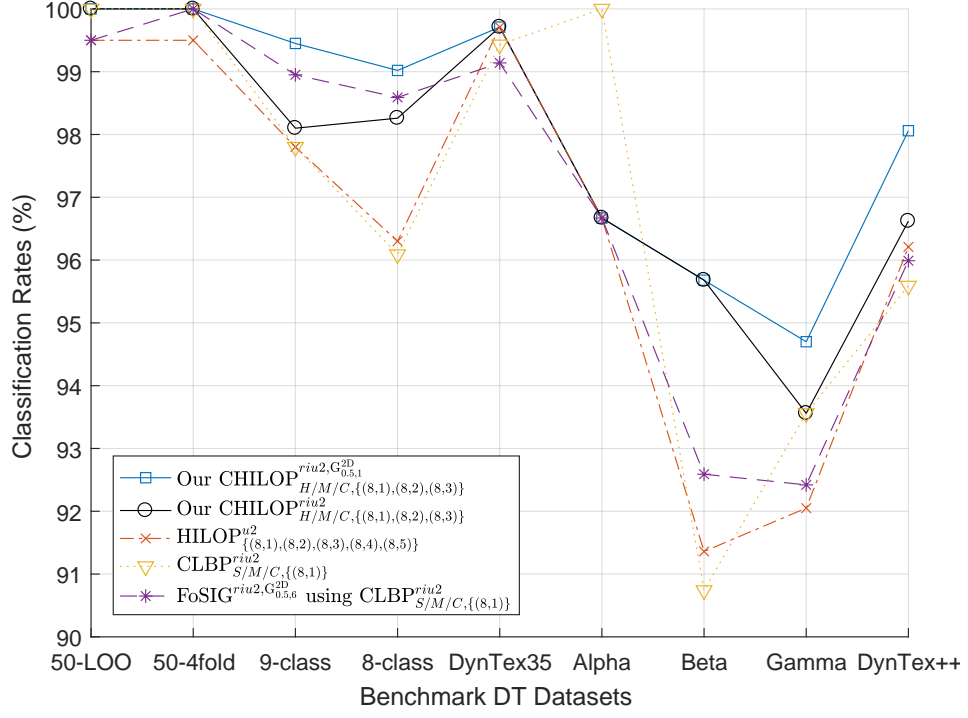


Figure 6.7: Outstanding performances of our CHILOP descriptors in comparison with that of HILOP [C3], FoSIG [C2], and CLBP [3].

to simulate image texture by stationary Gaussian random fields. We point out hereafter the following beneficial properties of our approach inheriting from these Gaussian-based filters.

- *Robustness to changes of illumination, scales, and environment:* Gaussian-based filtered volumes  $\mathcal{V}_{\sigma_i, \sigma'_i}^{DoG}$  are invariant sequences against illumination thanks to exploiting various scales of Gaussian filtering kernels. In addition, the receptive  $\mathcal{V}_{\sigma_i, \sigma'_i}^{DoG}$  volumes, formed by two different Gaussian kernels, allow to capture features with more robustness to the major remaining problems of DT description: illumination, scale, and environmental changes.
- *Robustness to noise:* Instead of extracting features from a raw video  $\mathcal{V}$ , its Gaussian-based filtered volumes  $\mathcal{V}^G$  allow to capture local features with more intensity to noise. On the other hand, DoG features are also exploited in our proposal to make descriptor more robust against changes of environment and illumination.
- *Forceful incisive elements:* Well-known as an approximation of Laplacian of Gaussian (LoG),  $\mathcal{V}_{\sigma_i, \sigma'_i}^{DoG}$  sequences provide beneficial receptive clues for feature encoding. Meantime,  $\mathcal{V}_{\sigma_i}^G$  volumes produce robust blurred features for the description. Consequently, the performance of DT recognition is enhanced thanks to these supplementary filtered volumes (see Table 6.7 for their contributions).

### 6.5.2 Construction of RUBIG descriptor

As a derivation of the LBP-based computation, encoding rubik-based patterns can be faced with sensitivity to noise and illumination problems. To treat those, Gaussian-based filtering kernels in Equations (6.1) and (6.2) are addressed as a pre-processing step to reduce the negative impacts of environmental changes on DT representation. It should be noted that Gaussian filter has been addressed together with LBP operator in [108]. However, it employed a 2D Gaussian kernel to analyze neighborhoods at different area scales of a pixel for texture description, while Nguyen *et al.* [C2] utilized it for capturing spatio-temporal features from filtered images of planes in a video. Accordingly, for a video  $\mathcal{V}$  along with pre-defined couples of standard deviations  $\Lambda = \{(\sigma_i, \sigma'_i)\}_{i=1}^m$ , a series of volumes of blurred Gaussian



features  $\mathcal{V}_{\sigma_i}^G$  and the difference of Gaussians  $\mathcal{V}_{\sigma_i, \sigma'_i}^{DoG}$  are computed as follows. Figure 6.8 shows several samples of this filtering.

$$\mathcal{V}_{\sigma_i}^G = G_{\sigma_i}^n(\varphi_n) * \mathcal{V} \quad , \quad \mathcal{V}_{\sigma_i, \sigma'_i}^{DoG} = |\text{DoG}_{\sigma_i, \sigma'_i}^n(\varphi_n)| * \mathcal{V} \quad (6.10)$$

where “\*” is a convolving operator, and  $\sigma_i < \sigma'_i$ . We then utilize the proposed LRP operator for each filtered volume to capture RUBik Blurred-Invariant Gaussian (RUBIG) features for DT description (see Figure 6.9 for a graphical illustration of this construction). The obtained histograms are then normalized and concatenated to form a discriminative descriptor.

$$\text{RUBIG}_{\Gamma, \Omega, \Lambda}(\mathcal{V}) = [\text{LRP}_{\Gamma, \Omega}(\mathcal{V}_{\sigma_i}^G), \text{LRP}_{\Gamma, \Omega}(\mathcal{V}_{\sigma_i, \sigma'_i}^{DoG})]_{i=1}^m \quad (6.11)$$

Our RUBIG is based on two important properties to boost its performance compared to that of V-BIG [C5] (see Table 6.7 for a specific performing comparison): *i*) RUBIG is enriched by rich spatio-temporal features thanks to our novel, discriminative operator LRP. *ii*) RUBIG can be better resistant to the illumination and noise since its blurred-invariant features are encoded from multi-scale Gaussian-based volumes. Besides the beneficial properties inheriting from Gaussian-based filtering (see Section 6.5.1), our RUBIG has also following properties.

- *Multi-scale and rich spatio-temporal features:* RUBIG is concerned with analysis of rich spatio-temporal features to form an effective descriptor that is more discriminative than CLBP features of V-BIG. Moreover, it is enriched by robust clues based on various scales of Gaussian kernels taken into account the filtering, while V-BIG is lack of multi-scale analysis due to just a single-scale involved in.
- *Informative voxel discrimination:* Shape and motion cues are jointly structured thanks to voxels in a DT video enriched by discriminative information with 3D Gaussian kernels. In the meanwhile, FoSIG [C2] just captures spatio-temporal features of voxels on 2D Gaussian filtered images of the planes in the video.

### 6.5.3 Experiments and evaluations

#### 6.5.3.1 Parameters for experimental implementation

The 3-dimensional Gaussian-based kernels are exploited to capture volumes of blurred-invariant features, where the kernel width of each axis is traditionally truncated to  $[-3\sigma, 3\sigma]$  ( $\sigma$  is the standard deviation of Gaussian distribution) for optimally capturing the energy of Gaussian distribution. We then consider a set of couples of standard deviations  $\Lambda = \{(\sigma_i, \sigma'_i)\}_{i=1}^m = \{(0.5, 6), (0.75, 5), (1, 4)\}$  (i.e.,  $m = 3$ ) in order to compute DoG together with Gaussian-filtered outcomes. In brief, for each couple  $(\sigma_i, \sigma'_i)$ , two following outcomes  $\mathcal{V}_{\sigma_i}^G$ , and  $\mathcal{V}_{\sigma_i, \sigma'_i}^{DoG}$  are produced and then are encoded by our LRP operator in the next step. It should be noted that the large scale ratios between two scales of each couple of standard deviations are taken into account. Our idea is to highlight the invariant features of DoG outcome

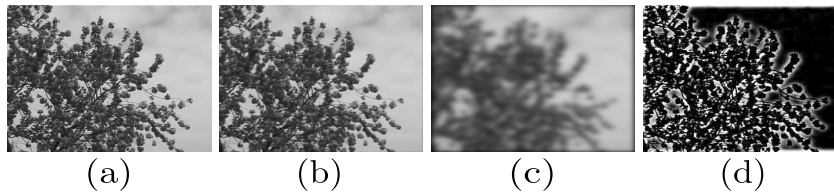


Figure 6.8: An instance of 3D Gaussian-based filters. (a) is an input gray-scale frame of a DT video. (b) and (c) are 3D smoothed frames of (a) using  $\sigma_1 = 0.5$  and  $\sigma_2 = 4$  respectively. (d) denotes the 3D DoG of (b) and (c).

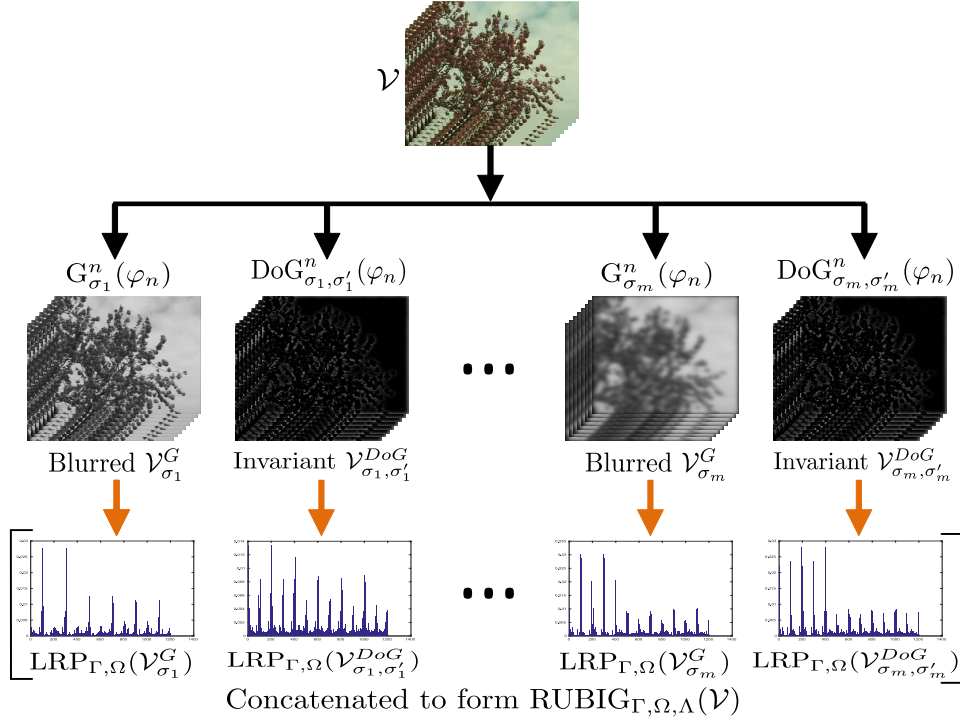


Figure 6.9: Illustration of proposed framework for encoding RUBIG descriptor.

extracted from two different scales of Gaussian filtering. Empirically, the more two standard deviations are different, the more DoG outcome contains rich, discriminative, and robust features for LRP operator. Therefore, this concept justifies the large scale ratios of standard deviations between two scales in our model. For DT representation, LRP features are extracted from the filtered volumes by utilizing parameters of *riu2* mapping,  $\{(P, R)\} = \{(8, 1)\}$  for single-scale relationships, and  $\{(8, 1), (8, 2)\}$  for multi-scale in further local regions. The achieved components are integrated in two investigations of  $\Omega = \{D_{M/C}, D_{M/C}\}$  to form corresponding RUBIG descriptors with dimensions of 540 and 3600 bins respectively (see Table 6.2 for comparison with other LBP-based descriptors).

### 6.5.3.2 Assessments of RUBIG's performances

Specific experimental results of our descriptor RUBIG on benchmark datasets are shown in Table 6.8 with the highest rates in bold. It should be noted that only results of the setting of  $D_{M/C}$  are reported due to its high performance. As expected, it can be verified from Tables 6.7, 6.8 that RUBIG outperforms compared to those of FoSIG and V-BIG thanks to the crucial contributions of the proposed operator LRP utilized for capturing rich spatio-temporal patterns in the Gaussian-based filtered volumes. The experiments have also validated that RUBIG's performance becomes more "stable" in consideration of various scales of Gaussian-based kernels (see Table 6.8). In general, our framework performs very well in comparison with the state-of-the-art approaches, including deep-learning-based methods in several circumstances (see Table 6.33). Due to these recognition rates on most of DT datasets, the settings of  $D_{M/C}$  and  $\{(0.5, 6), (0.75, 5), (1, 4)\}$  for the multi-scale LRP encoding are addressed for comparison (see Table 6.8).

## 6.6 Representation based on Gaussian-filtered CAIP features

Hereunder, we present in general our proposed framework, as illustrated in Figure 6.10. Briefly, we investigate two crucial types of DT representation which are correspondingly based on  $2D/3D$  Gaussian-based filtering kernels in order to take advantage of  $\text{LOGIC}^{2D/3D}$  properties against the negative influences on DT encoding. To this end, first, the  $2D/3D$  Gaussian-based kernels are involved in pre-processing an input video  $\mathcal{V}$  (see Section 6.6.1) to figure out completed sets of  $2D/3D$  Gaussian-based

Table 6.7: Comparison contributions in rates (%) on DynTex++ between components of descriptors FoSIG [C2], V-BIG [C5] and our RUBIG.

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

Table 6.8: Classification rates (%) on benchmark datasets.

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

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

filtered outcomes  $\Omega_{\sigma, \sigma'}^{2D/3D}$  with more insensitivity to noise and bipolar-invariant characteristics. Second, in order to preserve advantages of these bipolar-invariant features for DT encoding, we proposed in Section 3.2 an important modification of CLBP operator to be agreed with the particular characteristics of  $\Omega_{\sigma, \sigma'}^{2D/3D}$  outcomes where the typical CLBP inefficiently reacts to noise and near uniform regional problems caused by their zero-gray-scale bipolar cells. As the result of that, Completed Adaptive Patterns (CAIP) with more discrimination power are pointed out to be able to handle those problems for improving the performance. Indeed, our experiments for DT classification have validated the significant contribution of this encoding adaptation compared to that of CLBP and other LBP-based approaches for DT description (see Section 6.6.4). Finally, forceful and discriminative descriptors LOGIC<sup>2D/3D</sup> are formed by correspondingly utilizing CAIP operator to encode blurred and bipolar-invariant features of  $\Omega_{\sigma, \sigma'}^{2D/3D}$  filtered outcomes that are addressed in different scales of Gaussian-based filtering kernels (see Section 6.6.3). Hereafter, we express above processes in detail.

### 6.6.1 Completed sets of Gaussian-based filtered outcomes

Motivated by filter-bank approaches [2, 72, 74], the 2D/3D Gaussian-based filtering kernels are addressed in our framework to overcome the well-known problems in DT representation. In two our prior works, descriptor FoSIG [C2] is involved with smooth-invariant features of  $G_{\sigma}^{2D}$  and  $DoG_{\sigma, \sigma'}^{2D}$  filtered images figured out by using the 2D Gaussian-based filtering kernels, while V-BIG [C5] takes the 3D Gaussian-based filters into account filtering a video to obtain smooth-invariant volumes. However, their  $DoG_{\sigma, \sigma'}^{2D/3D}$  filtered supplements have remained some noise caused by the features closed to zero-gray-values that negatively impact on LBP-based encoding. Furthermore, the crucial properties of Gaussian bipolar derived from these DoGs have not been exploited to enrich more spatio-temporal characteristics for DT description. Addressing the missing leverages, we present in this section completed sets of filtered outcomes using the 2D/3D kernels in Equations (6.1) and (6.2) as follows.

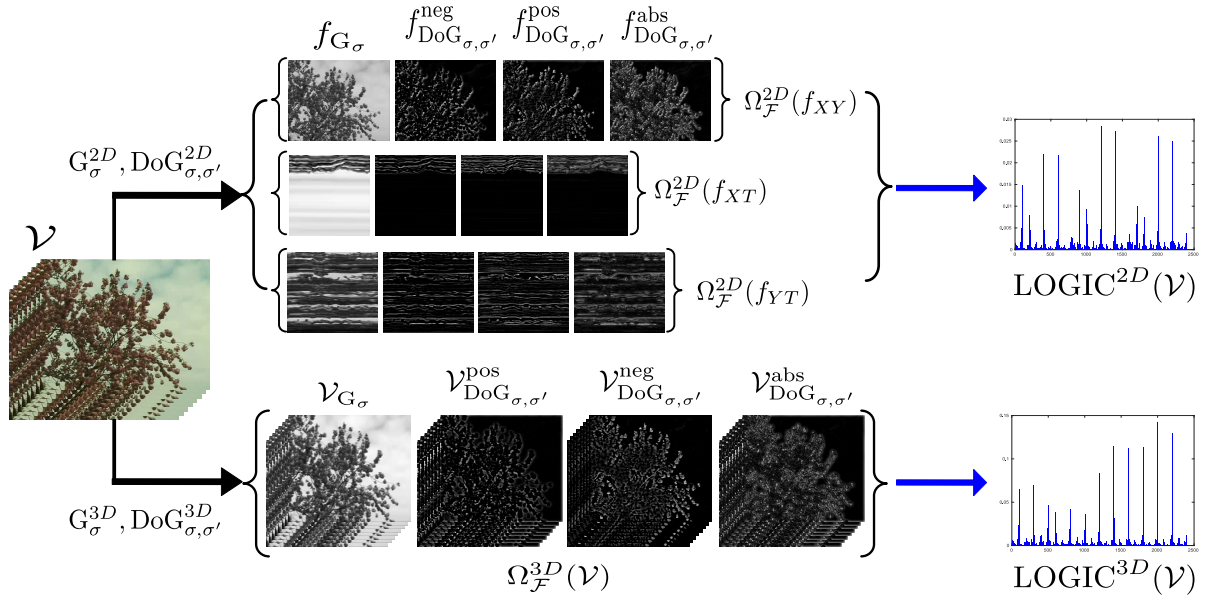


Figure 6.10: Our proposed framework for structuring an input video  $\mathcal{V}$  based on its Gaussian-based filtered outcomes. Therein, the black arrows denote preprocessings using  $2D/3D$  Gaussian-based filtering kernels while the blue ones imply processes of DT encoding.

Given a gray-scale image  $\mathcal{I}$  (correspondingly a video  $\mathcal{V}$ ) and a pair of pre-defined standard deviations  $(\sigma, \sigma')$ , two Gaussian-based kernels  $G_{\sigma}^{2D/3D}$  and  $\text{DoG}_{\sigma, \sigma'}^{2D/3D}$  are taken into account for a preprocessing analysis to produce the following blurred ( $\mathcal{I}_{G_{\sigma}}$ ) and invariant ( $\mathcal{I}_{\text{DoG}_{\sigma, \sigma'}}$ ) filtered images as

$$\mathcal{I}_{G_{\sigma}} = G_{\sigma}^{2D}(x, y) * \mathcal{I} \quad \text{and} \quad \mathcal{I}_{\text{DoG}_{\sigma, \sigma'}} = \text{DoG}_{\sigma, \sigma'}^{2D}(x, y) * \mathcal{I} \quad (6.12)$$

in which  $x$  and  $y$  denote the spatial coordinates,  $\sigma < \sigma'$ , “ $*$ ” means a convolution operator. Correspondingly,  $\mathcal{V}_{G_{\sigma}^{3D}}$  and  $\mathcal{V}_{\text{DoG}_{\sigma, \sigma'}^{3D}}$  filtered volumes are also formed as follows.

$$\mathcal{V}_{G_{\sigma}} = G_{\sigma}^{3D}(x, y, t) * \mathcal{V} \quad \text{and} \quad \mathcal{V}_{\text{DoG}_{\sigma, \sigma'}} = \text{DoG}_{\sigma, \sigma'}^{3D}(x, y, t) * \mathcal{V} \quad (6.13)$$

where  $t$  indicates the temporal coordinate.

Motivated by the concept of biologically-inspired filtering, introduced by Vu *et al.* [116], positive-bipolar  $\mathcal{I}_{\text{DoG}_{\sigma, \sigma'}^{\text{pos}}}$ , negative-bipolar  $\mathcal{I}_{\text{DoG}_{\sigma, \sigma'}^{\text{neg}}}$  and invariant-absolute  $\mathcal{I}_{\text{DoG}_{\sigma, \sigma'}^{\text{abs}}}$  filtered images are generated by addressing bipolar cells which are derived from  $\mathcal{I}_{\text{DoG}_{\sigma, \sigma'}}$  as follows.

$$\begin{aligned} \mathcal{I}_{\text{DoG}_{\sigma, \sigma'}^{\text{pos}}}(\mathbf{q}) &= \begin{cases} \mathcal{I}_{\text{DoG}_{\sigma, \sigma'}}(\mathbf{q}), & \text{if } \mathcal{I}_{\text{DoG}_{\sigma, \sigma'}}(\mathbf{q}) \geq \varepsilon \\ 0, & \text{otherwise.} \end{cases} \\ \mathcal{I}_{\text{DoG}_{\sigma, \sigma'}^{\text{neg}}}(\mathbf{q}) &= \begin{cases} |\mathcal{I}_{\text{DoG}_{\sigma, \sigma'}}(\mathbf{q})|, & \text{if } \mathcal{I}_{\text{DoG}_{\sigma, \sigma'}}(\mathbf{q}) \leq -\varepsilon \\ 0, & \text{otherwise.} \end{cases} \\ \mathcal{I}_{\text{DoG}_{\sigma, \sigma'}^{\text{abs}}}(\mathbf{q}) &= \mathcal{I}_{\text{DoG}_{\sigma, \sigma'}^{\text{pos}}}(\mathbf{q}) + \mathcal{I}_{\text{DoG}_{\sigma, \sigma'}^{\text{neg}}}(\mathbf{q}) \end{aligned} \quad (6.14)$$

where  $\varepsilon$  is a pre-defined threshold in order to prevent from taking closed-to-zero areas into account feature encoding because of the uniform regional problem; function  $\mathcal{I}_{\text{DoG}_{\sigma, \sigma'}}(\mathbf{q})$  returns a gray-filtered value of a pixel  $\mathbf{q}$  in DoG-filtered image  $\mathcal{I}_{\text{DoG}_{\sigma, \sigma'}}$  (see Figure 6.11(a) for an example of this  $2D$  filtering).

Similarly, we take this partition into account for  $\mathcal{V}_{\text{DoG}_{\sigma, \sigma'}}$  analysis in order to produce bipolar and

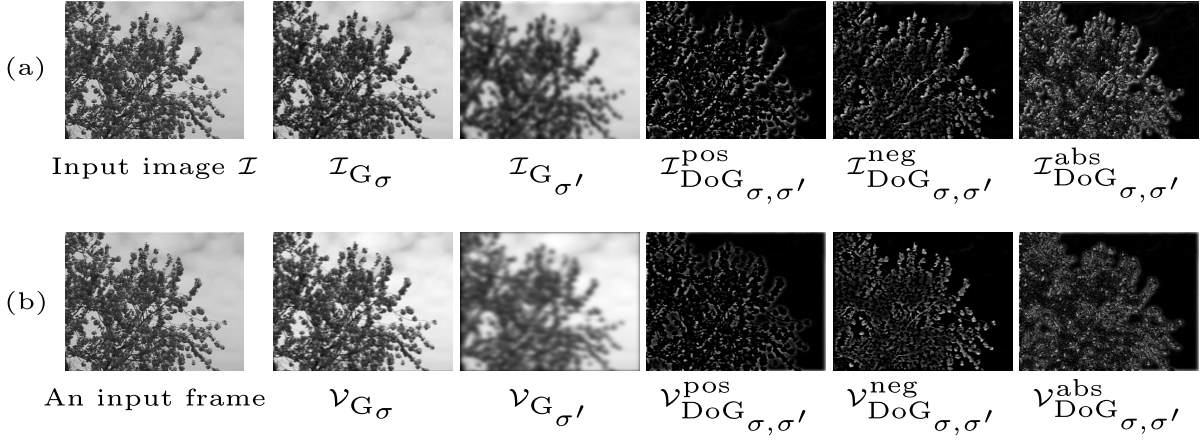


Figure 6.11: An instance of two Gaussian-based filterings with  $\sigma = 0.7$  and  $\sigma' = 2\sqrt{5}\sigma$ : (a) for filtering a still image  $\mathcal{I}$  using  $2D$  kernels, (b) for filtering a video  $\mathcal{V}$  using  $3D$  kernels.

invariant-absolute filtered volumes as follows.

$$\begin{aligned}
 \mathcal{V}_{\text{DoG}_{\sigma,\sigma'}^{\text{pos}}}(\mathbf{p}) &= \begin{cases} \mathcal{V}_{\text{DoG}_{\sigma,\sigma'}}(\mathbf{p}), & \text{if } \mathcal{V}_{\text{DoG}_{\sigma,\sigma'}}(\mathbf{p}) \geq \varepsilon \\ 0, & \text{otherwise.} \end{cases} \\
 \mathcal{V}_{\text{DoG}_{\sigma,\sigma'}^{\text{neg}}}(\mathbf{p}) &= \begin{cases} |\mathcal{V}_{\text{DoG}_{\sigma,\sigma'}}(\mathbf{p})|, & \text{if } \mathcal{V}_{\text{DoG}_{\sigma,\sigma'}}(\mathbf{p}) \leq -\varepsilon \\ 0, & \text{otherwise.} \end{cases} \\
 \mathcal{V}_{\text{DoG}_{\sigma,\sigma'}^{\text{abs}}}(\mathbf{p}) &= \mathcal{V}_{\text{DoG}_{\sigma,\sigma'}^{\text{pos}}}(\mathbf{p}) + \mathcal{V}_{\text{DoG}_{\sigma,\sigma'}^{\text{neg}}}(\mathbf{p})
 \end{aligned} \tag{6.15}$$

in which  $\mathcal{V}_{\text{DoG}_{\sigma,\sigma'}}(\mathbf{p})$  returns a gray-filtered value of a voxel  $\mathbf{p}$  in DoG-filtered volume  $\mathcal{V}_{\text{DoG}_{\sigma,\sigma'}}$  (see Figure 6.11(b) for an instance of this  $3D$  Gaussian-based filtering).

As the result of above those, for the input image  $\mathcal{I}$  (correspondingly the video  $\mathcal{V}$ ), a completed set of supplementary  $2D$  ( $3D$ ) Gaussian-based filtered images (volumes) with a pair of standard deviations  $(\sigma, \sigma')$  is formed in order to forcefully structure discriminative features for DT representation as

$$\begin{aligned}
 \Omega_{\sigma,\sigma'}^{2D}(\mathcal{I}) &= \{\mathcal{I}_{G_\sigma}, \mathcal{I}_{\text{DoG}_{\sigma,\sigma'}^{\text{pos}}}, \mathcal{I}_{\text{DoG}_{\sigma,\sigma'}^{\text{neg}}}, \mathcal{I}_{\text{DoG}_{\sigma,\sigma'}^{\text{abs}}}\} \\
 \Omega_{\sigma,\sigma'}^{3D}(\mathcal{V}) &= \{\mathcal{V}_{G_\sigma}, \mathcal{V}_{\text{DoG}_{\sigma,\sigma'}^{\text{pos}}}, \mathcal{V}_{\text{DoG}_{\sigma,\sigma'}^{\text{neg}}}, \mathcal{V}_{\text{DoG}_{\sigma,\sigma'}^{\text{abs}}}\}
 \end{aligned} \tag{6.16}$$

It should be noted that  $\mathcal{I}_{G_\sigma}$  and  $\mathcal{I}_{\text{DoG}_{\sigma,\sigma'}^{\text{abs}}}$  filtered images have not been exploited in the prior work [116] while their contributions are significant in describing blurred-invariant features of DT motions. Indeed, our experiments for DT classification task have validated their positive influences (see Table 6.11 for specific instances). Furthermore, also based on the concept of bipolar cells, we investigate the impacts of elements in  $\Omega_{\sigma,\sigma'}^{3D}$  on DT representation, in which  $\Omega_{\sigma,\sigma'}^{3D}$  is formed by using the  $3D$  Gaussian-based kernels for video filtering. For convenience in presentation,  $\Omega_{\sigma,\sigma'}^{2D/3D}$  is henceforward an abbreviation of filtered images  $\Omega_{\sigma,\sigma'}^{2D}$  and filtered volumes  $\Omega_{\sigma,\sigma'}^{3D}$  in general.

### 6.6.2 Beneficial properties of filtered outcomes $\Omega_{\sigma,\sigma'}^{2D/3D}$

Addressing the  $2D/3D$  Gaussian-based filtering kernels with the meaningful threshold  $\varepsilon$  for denoising has figured out the completed sets of filtered outcomes  $\Omega_{\sigma,\sigma'}^{2D/3D}$  for DT representation (see Section 6.6.1). It should be noted that those sets contain complementary components. Hereafter, we discuss advantages of our proposed framework inheriting from these filterings:

- *Robustness to illumination and changes of environmental factors:* All filtered outcomes in  $\Omega_{\sigma,\sigma'}^{2D/3D}$  are invariant against changes of environment and illumination thanks to utilizing different

scales of Gaussian filtering kernels. Furthermore, the DoG-filtered outcomes, i.e.,  $\mathcal{I}_{\text{DoG}_{\sigma,\sigma'}}^{\text{abs}}$  and  $\mathcal{V}_{\text{DoG}_{\sigma,\sigma'}}^{\text{abs}}$ , allow to exploit more receptive features in order to enrich discriminative information of shape and motion cues for DT representation (see Table 6.11).

- *Insensitivity to noise:* Instead of encoding features from a raw video  $\mathcal{V}$ , taking advantage of its  $\Omega_{\sigma,\sigma'}^{2D/3D}$  outcomes grants our framework to capture local patterns with more robustness to noise. It should be noted that the 2D Gaussian kernel is used by Mäenpää *et al.* [108] to consider neighborhood areas of a pixel in different scales for textural image analysis, while our prior works FoSIG [C2] and V-BIG [C5] have also exploited the 2D/3D Gaussian kernels for capturing spatio-temporal characteristics from filtered images and volumes. Different from them, in addition to taking smooth-invariant Gaussian features into account DT representation (as done in FoSIG and V-BIG), an augmentation of bipolar outcomes is related to in this work, i.e.,  $\mathcal{I}_{\text{DoG}_{\sigma,\sigma'}}^{\text{pos}}$ ,  $\mathcal{I}_{\text{DoG}_{\sigma,\sigma'}}^{\text{neg}}$ ,  $\mathcal{V}_{\text{DoG}_{\sigma,\sigma'}}^{\text{pos}}$ , and  $\mathcal{V}_{\text{DoG}_{\sigma,\sigma'}}^{\text{neg}}$ . This makes the obtained patterns more robustness against the major remaining problems of DT description (see Table 6.11 and Figure 6.13). Furthermore,  $\text{DoG}^{2D/3D}$  outcomes are denoised by using the threshold  $\varepsilon$  to form  $\mathcal{I}_{\text{DoG}_{\sigma,\sigma'}}^{\text{abs}}$  and  $\mathcal{V}_{\text{DoG}_{\sigma,\sigma'}}^{\text{abs}}$ . This allows to efficiently boost the performance compared to taking the raw properties of them into account DT analysis, also as asserted for textural image description [116].
- *Forceful discriminating properties:* Addressing the completed sets of  $\Omega_{\sigma,\sigma'}^{2D/3D}$  for encoding DTs figures out robust descriptors of multi-filtered-features dealing with the major remaining problems of DT description. Indeed, the well-known DoG is considered as an approximation of Laplacian of Gaussian (LoG). Therefore, its variants  $\mathcal{I}_{\text{DoG}_{\sigma,\sigma'}}^{\text{pos}}$ ,  $\mathcal{I}_{\text{DoG}_{\sigma,\sigma'}}^{\text{abs}}$  and  $\mathcal{I}_{\text{DoG}_{\sigma,\sigma'}}^{\text{neg}}$  (resp.  $\mathcal{V}_{\text{DoG}_{\sigma,\sigma'}}^{\text{pos}}$ ,  $\mathcal{V}_{\text{DoG}_{\sigma,\sigma'}}^{\text{abs}}$ , and  $\mathcal{V}_{\text{DoG}_{\sigma,\sigma'}}^{\text{neg}}$ ) provide critical reception of shape and motion clues for feature encoding. In addition, the positive-bipolar and negative-bipolar characteristics have proved their important contributions in order to boost the performance of DT classification (see Table 6.11 for a specific contribution of each in  $\Omega_{\sigma,\sigma'}^{2D/3D}$ ).

### 6.6.3 DT description based on complementary filtered outcomes $\Omega_{\sigma,\sigma'}^{2D/3D}$

As mentioned in Sections 6.6.1 and 6.6.2, the completed sets of filtered outcomes  $\Omega_{\sigma,\sigma'}^{2D/3D}$  are robust against factors which negatively impact on DT representation. Moreover, they consist of complementary components. In this section, we take advantage of these  $\Omega_{\sigma,\sigma'}^{2D/3D}$  properties along with the completed adaptive operator CAIP in order to efficiently capture blurred and bipolar-invariant characteristics for boosting the discrimination power. Accordingly, given a video  $\mathcal{V}$  and a pre-defined set of pair of standard deviations  $\mathcal{F} = \{(\sigma_i, \sigma'_i)\}_{i=1}^m$ , in which  $m \in \mathbb{Z}^+$  is the cardinality of  $\mathcal{F}$ , we hereunder investigate two appreciable types of DT descriptions relying upon two corresponding completed sets of multi-scale 2D/3D Gaussian-based filtered supporting outcomes (i.e.,  $\Omega_{\mathcal{F}}^{2D}$  and  $\Omega_{\mathcal{F}}^{3D}$ ) in order to construct the following robust descriptors with significant performances in DT classification task.

**Proposed LOGIC<sup>2D</sup> descriptor:** In order to address the completed set of multi-scale 2D Gaussian-based filtered images  $\Omega_{\mathcal{F}}^{2D}$ , video  $\mathcal{V}$  is firstly split into separative collections of plane images  $f_{XY}$ ,  $f_{XT}$ , and  $f_{YT}$  subject to its three orthogonal planes  $\{XY, XT, YT\}$  (see Figure 6.10 for a graphical illustration). For each of plane-image collections, its 2D Gaussian-based filtered outcomes are computed by using Equations (6.12) and (6.14) to form corresponding completed sets, i.e.,  $\Omega_{\mathcal{F}}^{2D}(f_{XY})$ ,  $\Omega_{\mathcal{F}}^{2D}(f_{XT})$ , and  $\Omega_{\mathcal{F}}^{2D}(f_{YT})$ . The proposed CAIP operator is then utilized for these sets to efficiently capture blurred and bipolar-invariant characteristics  $\Gamma$  of spatio-temporal appearances. For instance, with each plane-image  $f \in f_{XY}$ , properties of  $\Gamma$  are structured as follows.

$$\Gamma(f, \Omega_{\mathcal{F}}^{2D}(f)) = \biguplus_{i=1}^m \left[ \text{CAIP}(f_{G_{\sigma_i}}), \text{CAIP}(f_{\text{DoG}_{\sigma_i, \sigma'_i}}^{\text{pos}}), \text{CAIP}(f_{\text{DoG}_{\sigma_i, \sigma'_i}}^{\text{neg}}), \text{CAIP}(f_{\text{DoG}_{\sigma_i, \sigma'_i}}^{\text{abs}}) \right] \quad (6.17)$$

where  $\biguplus$  stands for an operation to concatenate the obtained histograms  $\text{CAIP}(\cdot)$  of the corresponding Gaussian-filtered images  $\Omega_{\mathcal{F}}^{2D}(f)$  in order to form a terminal histogram  $\Gamma(f, \Omega_{\mathcal{F}}^{2D}(f))$  for the input plane-

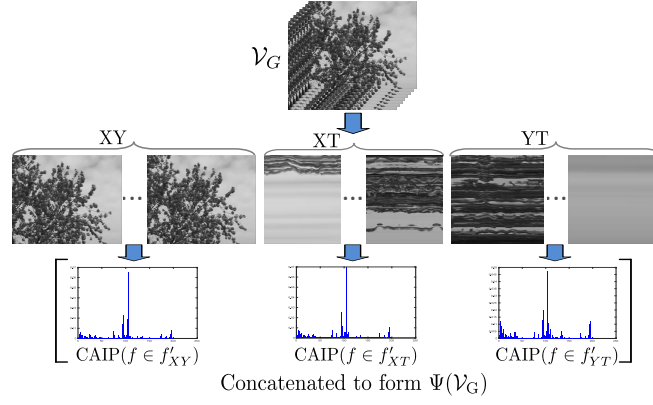


Figure 6.12: An illustration for encoding a Gaussian-based filtered volume  $\mathcal{V}_G$  using CAIP operator.

image  $f$ . Accordingly, it could deduce a description  $\Upsilon(f_{XY})$  for all plane-images of  $f_{XY}$  as

$$\Upsilon(f_{XY}) = \frac{1}{\mathcal{N}_{XY}} \sum_{f \in f_{XY}} \Gamma(f, \Omega_{\mathcal{F}}^{2D}(f)) \quad (6.18)$$

where  $\mathcal{N}_{XY}$  denotes a total of plane-images in  $f_{XY}$ . This computation is applied to the rest collections of plane-images,  $f_{XT}$  and  $f_{YT}$ , for  $\Upsilon(f_{XT})$  and  $\Upsilon(f_{YT})$  respectively. Finally, the obtained probability distributions are concatenated and normalized to construct a robust descriptor of Local 2D Gaussian-based Invariant Characteristics (LOGIC<sup>2D</sup>) as

$$\text{LOGIC}^{2D}(\mathcal{V}) = [\Upsilon(f_{XY}), \Upsilon(f_{XT}), \Upsilon(f_{YT})] \quad (6.19)$$

**Proposed LOGIC<sup>3D</sup> descriptor:** Similarly, the Gaussian-based filtered volumes  $\Omega_{\mathcal{F}}^{3D}(\mathcal{V})$  are addressed by allocating Equations (6.13) and (6.15) for analyzing the input video  $\mathcal{V}$  (see Figure 6.10 for a graphical illustration). Our CAIP operator is then nominated for each plane of a filtered volume  $\mathcal{V}_G \in \Omega_{\mathcal{F}}^{3D}(\mathcal{V})$  in order to effectively extract spatio-temporal DT features from blurred and bipolar-invariant properties of  $\mathcal{V}_G$  as follows.

$$\Psi(\mathcal{V}_G) = \left[ \frac{1}{|f'_{XY}|} \sum_{f \in f'_{XY}} \text{CAIP}(f), \frac{1}{|f'_{XT}|} \sum_{f \in f'_{XT}} \text{CAIP}(f), \frac{1}{|f'_{YT}|} \sum_{f \in f'_{YT}} \text{CAIP}(f) \right] \quad (6.20)$$

where  $|f'_{XY}|$ ,  $|f'_{XT}|$ , and  $|f'_{YT}|$  are respectively the cardinality of plane-image collections  $f'_{XY}$ ,  $f'_{XT}$ , and  $f'_{YT}$  that are separated subject to three orthogonal planes of  $\mathcal{V}_G$  (see Figure 6.12 for a graphical instance of this encoding). Finally, the obtained histograms are concatenated and normalized in order to structure a forceful descriptor of Local 3D Gaussian-based Invariant Characteristics (LOGIC<sup>3D</sup>) as follows.

$$\text{LOGIC}^{3D}(\mathcal{V}, \Omega_{\mathcal{F}}^{3D}(\mathcal{V})) = \biguplus_{i=1}^m \left[ \Psi(\mathcal{V}_{G_{\sigma_i}}), \Psi(\mathcal{V}_{\text{DoG}_{\sigma_i, \sigma'_i}}^{\text{pos}}), \Psi(\mathcal{V}_{\text{DoG}_{\sigma_i, \sigma'_i}}^{\text{neg}}), \Psi(\mathcal{V}_{\text{DoG}_{\sigma_i, \sigma'_i}}^{\text{abs}}) \right] \quad (6.21)$$

where  $\biguplus$  stands for an operation to concatenate the obtained histograms  $\Psi(\cdot)$  of the corresponding Gaussian-filtered volumes in  $\Omega_{\mathcal{F}}^{3D}(\mathcal{V})$ .

For a convenience in presentation, LOGIC<sup>2D</sup> and LOGIC<sup>3D</sup> can be henceforward abbreviated as LOGIC<sup>2D/3D</sup> in general. In terms of comparison to our prior works (i.e., FoSIG [C2], V-BIG [C5]), the proposed descriptors LOGIC<sup>2D/3D</sup> are based on the following advantages to enhance the performance:

- A meaningful threshold  $\varepsilon > 0$  is used to reduce noise caused by the closed-to-zero areas in DoG-filtered outcomes. This makes their invariant features more robust against illumination and environmental changes compared to using the raw DoG characteristics in FoSIG and V-BIG (see Table 6.9 for a comprehensive evaluations).

Table 6.9: Performances of each filtered element in  $\Omega_{(\sigma_i, \sigma'_i) \in \mathcal{F}}^{2D/3D}$  compared to those of FoSIG<sup>2D</sup> [C2] and V-BIG<sup>3D</sup> [C5] on DynTex++ using the same local supporting regions  $\{(P, R)\} = \{(8, 1), (8, 2)\}$ .

Descriptor	LOGIC <sup>2D</sup>					LOGIC <sup>3D</sup>					FoSIG <sup>2D</sup> [C2]		V-BIG <sup>3D</sup> [C5]	
$\mathcal{F} = \{(\sigma_i, \sigma_i \times k), k = 2\sqrt{5}\}$	$\mathcal{I}_G$	$\mathcal{I}_{DoG}^{abs}$	$\mathcal{I}_{DoG}^{pos}$	$\mathcal{I}_{DoG}^{neg}$	$\mathcal{I}_{DoG}^{pos} + \mathcal{I}_{DoG}^{neg}$	$\mathcal{V}_G$	$\mathcal{V}_{DoG}^{abs}$	$\mathcal{V}_{DoG}^{pos}$	$\mathcal{V}_{DoG}^{neg}$	$\mathcal{V}_{DoG}^{pos} + \mathcal{V}_{DoG}^{neg}$	$\mathcal{I}_G^*$	$\mathcal{I}_{DoG}^*$	$\mathcal{V}_G^*$	$\mathcal{V}_{DoG}^*$
$\{(0.7, 0.7k)\}$	95.73	94.42	94.24	93.87	95.32	95.01	94.06	92.63	94.74	94.96	<b>95.73</b>	<b>93.22</b>	<b>95.01</b>	<b>94.26</b>
$\{(1, k)\}$	94.48	94.67	86.61	94.63	95.19	94.06	93.24	94.58	93.59	96.08	94.48	91.78	94.06	93.33
$\{(1.3, 1.3k)\}$	93.64	93.07	91.98	92.17	93.51	92.19	94.53	93.66	91.99	95.12	93.64	90.11	92.19	92.48
$\{(0.7, 0.7k), (1, k)\}$	98.30	98.28	97.90	98.19	98.61	<b>95.98</b>	<b>95.77</b>	<b>95.83</b>	<b>95.61</b>	95.85	-	-	-	-
$\{(1, k), (1.3, 1.3k)\}$	97.96	98.15	97.36	97.94	98.36	94.79	95.57	95.75	94.76	96.12	-	-	-	-
$\{(0.7, 0.7k), (1, k), (1.3, 1.3k)\}$	<b>98.46</b>	<b>98.58</b>	<b>98.09</b>	<b>98.71</b>	<b>98.66</b>	95.41	95.47	95.49	95.34	<b>96.41</b>	-	-	-	-

Note: “-” means “not available”.  $\mathcal{I}_G^*$ ,  $\mathcal{I}_{DoG}^*$ ,  $\mathcal{V}_G^*$ , and  $\mathcal{V}_{DoG}^*$  respectively denote filtered elements of FoSIG [C2] and V-BIG [C5] addressed to be accordance with standard deviations  $\mathcal{F}$  and local supporting regions  $\{(P, R)\}$ .

- Multi-scale 2D/3D Gaussian-based kernels are exploited to forcefully capture blurred-invariant features for DT representation. In the meanwhile, FoSIG and V-BIG are lack of informative scales due to only a single Gaussian-based filtering kernel involved in (see Tables 6.9 and 6.11).
- Besides taking advantage of the smooth-invariant properties in  $G_{\sigma}^{2D/3D}$  and  $DoG_{\sigma, \sigma'}^{2D/3D}$ , our proposed descriptors LOGIC<sup>2D/3D</sup> are also enriched more spatio-temporal features of the positive-bipolar and negative-bipolar cells which have been derived from  $DoG^{2D/3D}$  (see Table 6.11).

## 6.6.4 Experiments and evaluations

### 6.6.4.1 Parameters for experimental implementation

**Settings for the Gaussian-based filterings:** We address multi-scale Gaussian-based filtering kernels  $\mathcal{F} = \{(\sigma_i, \sigma'_i)\}_{i=1}^m$  of spatio-temporal coordinates  $x, y, t \in [-3\sigma, 3\sigma]$ . Therein,  $\sigma_i$  and  $\sigma'_i$  are standard deviations conditioned by  $\sigma'_i = k \times \sigma_i$  so that  $\sigma_i < \sigma'_i$ .  $k \in \mathbb{R}^+$  is a pre-defined fitting coefficient which should be valued so that  $k > 1$  and  $k \times \sigma_i \leq 6$  since the informative appearance of DTs is sharply diminished when  $\sigma'_i$  is closed to  $\sigma_i$  or  $\sigma'_i > 6$ . As a result, we empirically investigate  $\sigma_i = 0.7$ ,  $\sigma_{i+1} = \sigma_i + 0.3$ , and  $k = 2\sqrt{5}$  in three consecutive scales so that the obtained descriptors are still in a reasonable dimension. That means  $\mathcal{F} = \{(0.7, 0.7 \times k), (1, k), (1.3, 1.3 \times k)\}$ . In order to eliminate useless regions for local encoding, the meaningful threshold  $\varepsilon$  should be set to 0.15, as empirically reported by Vu *et al.* [116].

**Settings for structuring LOGIC<sup>2D/3D</sup> descriptors:** The proposed operator CAIP is exploited using the joint parameters of *riu2* mapping with  $2(P+2)^2$  bins, where  $P$  is a number of considered neighbors. A multi-scale analysis using two scales of local neighbors  $\{(P, R)\} = \{(8, 1), (8, 2)\}$  is addressed for the DT encoding in order to capture more forceful information in larger regions. Because three planes XY, XT, and YT are addressed for each scale of local supports, it takes  $3 \times 2 \times 2 \times (8+2)^2 = 1200$  bins to structure a Gaussian-based filtered outcome in  $\Omega_{(\sigma_i, \sigma'_i) \in \mathcal{F}}^{2D/3D}$ . That means  $1200 \times m \times t$  bins for the final dimension, in which  $m = |\mathcal{F}| = 3$  denotes the number of Gaussian-based scales,  $t \in \mathbb{Z}^+$  denotes the quantity of outcomes in  $\Omega_{(\sigma_i, \sigma'_i) \in \mathcal{F}}^{2D/3D}$  involved with, i.e.,  $t \in \{1, 2, 3, 4\}$  (see Equation (6.16)). Table 6.2 shows a comprehensive comparison with other LBP-based descriptors.

### 6.6.4.2 Assessments of DoG-based features compared to those of FoSIG and V-BIG

For an objective comparison of performances on DynTex++ in particular, the settings for structuring FoSIG [C2] and V-BIG [C5] descriptors should be addressed to be accordance with standard deviations  $\mathcal{F} = \{(0.7, 0.7 \times k), (1, k), (1.3, 1.3 \times k)\}$  and local supporting regions  $\{(P, R)\} = \{(8, 1), (8, 2)\}$ . We arm at presenting the performances of separated elements in  $\Omega_{\sigma, \sigma'}^{2D/3D}$  to facilitate comparison with FoSIG and V-BIG, as well as to evaluate the crucial contributions of those in enhancing the discrimination of LOGIC<sup>2D/3D</sup> descriptors. It can be verified in Table 6.9 that the descriptors, based on blurred



Table 6.10: Rates (%) of LOGIC<sup>2D/3D</sup> using CLBP<sup>riu2</sup><sub>{(8,1),(8,2)}</sub> [3] instead of CAIP<sup>riu2</sup><sub>{(8,1),(8,2)}</sub>.

Dataset	UCLA				DynTex				Dyn++
Descriptor	50-LOO	50-4fold	9-class	8-class	Dyn35	Alpha	Beta	Gamma	
LOGIC <sup>2D</sup> <sub>CLBP</sub>	100	100	98.45	97.39	99.71	98.33	91.98	93.18	96.29
LOGIC <sup>3D</sup> <sub>CLBP</sub>	100	100	97.90	97.83	99.43	100	93.83	93.56	96.77

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

and bipolar-invariant characteristics, have significant improvement compared to the initial DoG features employed in FoSIG and V-BIG. Indeed, these advancements can be thoroughly discussed in detail as follows.

Regarding the single-scale of  $\mathcal{F}$ , the performance of  $\mathcal{I}_{\text{DoG}}^{\text{abs}}$  element is more stable and improved about 1.5~3% compared to  $\mathcal{I}_{\text{DoG}}^*$  of FoSIG. In the meanwhile,  $\mathcal{V}_{\text{DoG}}^{\text{abs}}$  is just more stable than that of V-BIG. This obtained enhancement is thanks to that our CAIP is agreed with encoding bipolar-invariant features in  $\Omega_{\sigma,\sigma'}^{2D/3D}$ . It should be noted that  $\mathcal{I}_{\text{DoG}}^{\text{abs}}$  and  $\mathcal{V}_{\text{DoG}}^{\text{abs}}$  in  $\Omega_{\sigma,\sigma'}^{2D/3D}$  respectively are DoGs of FoSIG and V-BIG which are thresholded by the meaningful level  $\varepsilon$ . In addition, integrating the positive and negative features of DoG also increases the discrimination power in general (see columns of  $\mathcal{I}_{\text{DoG}}^{\text{pos}} + \mathcal{I}_{\text{DoG}}^{\text{neg}}$  and  $\mathcal{V}_{\text{DoG}}^{\text{pos}} + \mathcal{V}_{\text{DoG}}^{\text{neg}}$  in Table 6.9).

Using multi-scale analysis of  $\mathcal{F}$ , the DoG-based components obtain outstanding performances: increasing by about 3% (up to over 98%) for the 2D filtered outcomes, and by about 1% for the 3D ones (see the last 3 rows in Table 6.9). Therein, those of the 3-scale 2D setting give the best rates, i.e.,  $\mathcal{I}_{\text{DoG}}^{\text{abs}}$  (98.58%),  $\mathcal{I}_{\text{DoG}}^{\text{pos}}$  (98.09%), and  $\mathcal{I}_{\text{DoG}}^{\text{neg}}$  (98.71%). In the meanwhile, those of 2-scale of  $\{(0.7, 0.7 \times k), (1, 1 \times k)\}$  also have considerable performances for both 2D and 3D filtered outcomes. Therefore, they can be implemented for mobile applications in practice thanks to their tiny dimension, i.e., 3600 bins for the 3-scale and 2400 bins for the 2-scale. Besides, the multi-scale settings of blurred features significantly improve the performance compared to the single-scale ones of FoSIG and V-BIG which have been lacking of the scale-information involved with the DT encoding (see results in a pair of columns  $\mathcal{I}_{\text{G}}$  and  $\mathcal{I}_{\text{G}}^*$ ,  $\mathcal{V}_{\text{G}}$  and  $\mathcal{V}_{\text{G}}^*$  in Table 6.9).

Furthermore, the experimental results in Table 6.9 have indicated that the larger value of standard deviation  $\sigma$  is taken into account the filterings, the less appearance information can be captured for DT representation. This is in accordance with all filtered elements in  $\Omega^{2D/3D}$  as well as those of FoSIG and V-BIG. Consequently, in practice, these Gaussian-based filterings should be addressed by  $(\sigma, \sigma')$  in reasonable constraints.

#### 6.6.4.3 Assessments of LOGIC<sup>2D/3D</sup>'s performances

Results of our proposed LOGIC<sup>2D/3D</sup> descriptors for DT classification on benchmark datasets are detailed in Table 6.11, in which the highest rates are in bold. Based on the experimental results, it could be pointed out the following crucial statements.

First, it can be verified that taking the blurred and bipolar-invariant Gaussian-based features into account capturing shape and motion clues for DT representation has been authenticated the prominent effectiveness, as mentioned in Sections 6.6.1, 6.6.2, and 6.6.3. Indeed, Table 6.11 illustrates that DT encoding based on Gaussian-based filtered complements  $\Omega_{\mathcal{F}}^{2D/3D}$  has correspondingly figured out robust descriptors LOGIC<sup>2D/3D</sup> with outperformances compared to all non-deep-learning methods. In comparison with those of deep-learning techniques, our LOGIC<sup>2D</sup> descriptor outperforms on most of circumstances, except recognizing DT on *Beta* and *Gamma* schemes (see Tables 6.11 and 6.33). In the meanwhile, LOGIC<sup>3D</sup> has also resulted out the promising classification rates.

Second, it can be observed from Table 6.11 that the performance of LOGIC<sup>2D</sup> descriptor is better

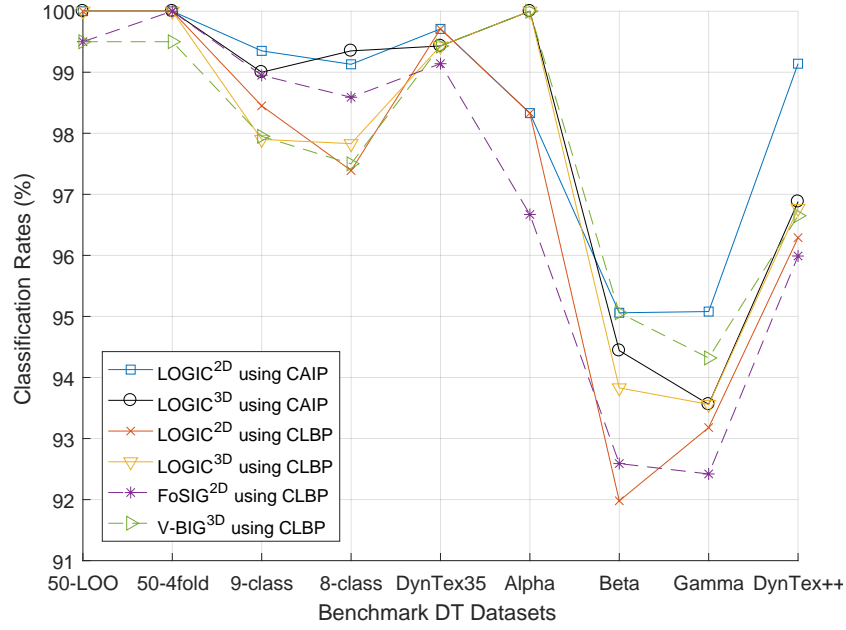


Figure 6.13: The performances of our descriptors  $\text{LOGIC}^{2D/3D}$ , which utilize both the adapted CAIP operator and the typical CLBP [3] for DT encoding, are compared to our prior works  $\text{FoSIG}^{2D}$  [C2] and  $\text{V-BIG}^{3D}$  [C5].

Table 6.11: Classification rates (%) on DT benchmark datasets of  $\text{LOGIC}^{2D/3D}$  descriptors.

Dataset		UCLA				DynTex				Dyn++
Descriptor	Gaussian-based Filtered Complement(s)	50-LOO	50-4fold	9-class	8-class	Dyn35	Alpha	Beta	Gamma	
$\text{LOGIC}^{2D}$	$f_{G_{\mathcal{F}}}$	100	100	97.85	98.59	98.29	100	94.44	93.56	98.46
	$f_{\text{DoG}_{\mathcal{F}}}^{\text{pos}}$	100	100	98.50	98.15	98.57	100	94.44	91.29	98.09
	$f_{\text{DoG}_{\mathcal{F}}}^{\text{neg}}$	100	100	99.25	99.13	98.57	98.33	<b>95.68</b>	91.67	98.71
	$f_{\text{DoG}_{\mathcal{F}}}^{\text{abs}}$	100	100	99.20	97.39	99.14	98.33	93.21	91.29	98.58
	$f_{\text{DoG}_{\mathcal{F}}}^{\text{pos}} + f_{\text{DoG}_{\mathcal{F}}}^{\text{neg}}$	100	100	98.50	98.80	99.14	98.33	95.06	93.94	98.66
	$f_{G_{\mathcal{F}}} + f_{\text{DoG}_{\mathcal{F}}}^{\text{abs}}$	100	100	98.55	98.70	98.86	98.33	95.06	92.80	98.91
	$f_{G_{\mathcal{F}}} + f_{\text{DoG}_{\mathcal{F}}}^{\text{pos}} + f_{\text{DoG}_{\mathcal{F}}}^{\text{neg}}$	100	100	98.85	97.39	99.14	98.33	94.44	<b>95.45</b>	99.02
	$f_{\text{DoG}_{\mathcal{F}}}^{\text{pos}} + f_{\text{DoG}_{\mathcal{F}}}^{\text{neg}} + f_{\text{DoG}_{\mathcal{F}}}^{\text{abs}}$	100	100	98.95	97.83	99.43	98.33	95.06	93.56	99.02
	$\Omega_{\mathcal{F}}^{2D} = f_{G_{\mathcal{F}}} + f_{\text{DoG}_{\mathcal{F}}}^{\text{pos}} + f_{\text{DoG}_{\mathcal{F}}}^{\text{neg}} + f_{\text{DoG}_{\mathcal{F}}}^{\text{abs}}$	100	100	<b>99.35</b>	99.13	<b>99.71</b>	98.33	95.06	95.08	<b>99.14</b>
$\text{LOGIC}^{3D}$	$\mathcal{V}_{G_{\mathcal{F}}}$	100	100	98.60	96.74	98.57	100	92.59	93.56	95.41
	$\mathcal{V}_{\text{DoG}_{\mathcal{F}}}^{\text{pos}}$	100	100	97.35	97.28	99.43	100	91.98	92.80	95.49
	$\mathcal{V}_{\text{DoG}_{\mathcal{F}}}^{\text{neg}}$	100	100	99.15	98.37	98.57	100	94.44	92.05	95.34
	$\mathcal{V}_{\text{DoG}_{\mathcal{F}}}^{\text{abs}}$	99.00	100	98.75	96.52	98.57	95.00	91.36	91.29	95.47
	$\mathcal{V}_{\text{DoG}_{\mathcal{F}}}^{\text{pos}} + \mathcal{V}_{\text{DoG}_{\mathcal{F}}}^{\text{neg}}$	100	100	99.15	98.48	<b>99.71</b>	100	93.21	92.05	96.41
	$\mathcal{V}_{G_{\mathcal{F}}} + \mathcal{V}_{\text{DoG}_{\mathcal{F}}}^{\text{abs}}$	100	100	97.75	97.61	98.86	98.33	93.21	92.80	96.37
	$\mathcal{V}_{G_{\mathcal{F}}} + \mathcal{V}_{\text{DoG}_{\mathcal{F}}}^{\text{pos}} + \mathcal{V}_{\text{DoG}_{\mathcal{F}}}^{\text{neg}}$	100	100	98.65	98.70	<b>99.71</b>	100	93.83	93.56	96.71
	$\mathcal{V}_{\text{DoG}_{\mathcal{F}}}^{\text{pos}} + \mathcal{V}_{\text{DoG}_{\mathcal{F}}}^{\text{neg}} + \mathcal{V}_{\text{DoG}_{\mathcal{F}}}^{\text{abs}}$	100	100	99.10	97.07	99.43	100	92.59	91.67	96.39
	$\Omega_{\mathcal{F}}^{3D} = \mathcal{V}_{G_{\mathcal{F}}} + \mathcal{V}_{\text{DoG}_{\mathcal{F}}}^{\text{pos}} + \mathcal{V}_{\text{DoG}_{\mathcal{F}}}^{\text{neg}} + \mathcal{V}_{\text{DoG}_{\mathcal{F}}}^{\text{abs}}$	100	100	99.00	<b>99.35</b>	99.43	100	94.44	93.56	96.88

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

than that of  $\text{LOGIC}^{3D}$  in general. It means that the Gaussian-based filtering kernels exploited on spatial-temporal plane-images (i.e., separative images subject to  $\{XY, XT, YT\}$  planes of a video) allow to efficiently deal with the negative impacts of illumination and noise on DT encoding rather than using these kernels in volume filtering structures.

Third, as mentioned in Section 3.2, the problems of near uniform regions and sensitivity to noise caused by the zero-center bipolar cells have been efficaciously carried out by our adapted CAIP operator to boost the performance. Actually, it can be verified from Figure 6.13 and Table 6.10 that the  $\text{LOGIC}^{2D/3D}$  descriptors, using CAIP to capture shape and motion cues of DTs in filtered complements  $\Omega_{\mathcal{F}}^{2D/3D}$ , are significantly enhanced in comparison with using the typical CLBP [3] on  $\Omega_{\mathcal{F}}^{2D/3D}$ .

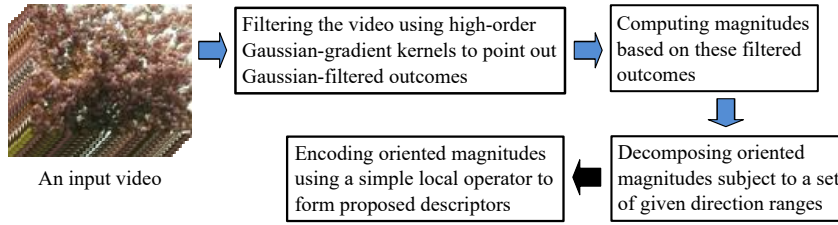


Figure 6.14: A proposed framework for encoding a video in general. Therein, the blue arrows denote progresses of pre-processing, the black one is a progress of encoding features of oriented magnitudes.

Fourth, it can be seen from Figure 6.13 that the  $\text{LOGIC}^{2D/3D}$  descriptors outperform in general compared to  $\text{FoSIG}^{2D}$  [C2] and  $\text{V-BIG}^{3D}$  [C5]. It is thanks to taking advantage of the adapted CAIP operator and the  $2D/3D$  Gaussian-based kernels in multi-scale analysis. In addition, with the similar complemented supplements, i.e.,  $f_{G_F} + f_{\text{DoG}_F}^{\text{abs}}$ ,  $\text{LOGIC}^{2D}$  also has better performance than that of  $\text{FoSIG}^{2D}$  and  $\text{V-BIG}^{3D}$  in most cases of DT classification (see Tables 6.11, 6.33, and Figure 6.13). It again validates the important contribution of our CAIP modification for DT encoding.

Finally, each of filtered outcomes in  $\Omega_F^{2D/3D}$ , as well as integrating them into different ways has figured out corresponding descriptors with competitive discrimination powers (see Table 6.11 for a specific comparison of their performances). Among of them, the integrating instances of all, i.e.,  $\text{LOGIC}^{2D/3D}$  using all complements in  $\Omega_F^{2D/3D}$  for DT representation, have more “stable” executions and higher accuracies. This is thanks to exploiting benefits of the blurred and bipolar-invariant features which are extracted from the Gaussian-based filtered outcomes in the completed context.

In terms of comparison with the state-of-the-art methods, we nominate descriptor  $\text{LOGIC}^{2D}$  since its performance is better than that of  $\text{LOGIC}^{3D}$ . From now on, if  $\text{LOGIC}^{2D/3D}$  descriptors are not specified in detail of which kinds of complements in  $\Omega_F^{2D/3D}$  are implemented, the default integration is all their filtered complements involved with. Hereafter, comprehensive evaluations of the proposed framework on different DT datasets are discussed in a global comparison with the existing approaches.

## 6.7 Representation based on oriented magnitudes of Gaussian gradients

In our prior works [C2, C5, J4], we have indicated that taking Gaussian-based filtering kernels into account DT representation could improve the discrimination power of local DT encoding. This is thanks to mitigating the negative impacts the typical problems on DT encoding. However, the achieved improvements are still at a moderate level since those problems may be not dealt with thoroughly. Instead of exploiting Gaussian-based filtered features as in [C2, C5], motivated from HoG descriptor [118] where oriented information has been successfully exploited for representation of local features, we propose in this work an efficient framework for DT representation based on high-order oriented magnitudes that are decomposed from Gaussian-gradient outcomes as graphically illustrated in Figure 6.14. Accordingly, high-order Gaussian-gradient kernels are used to filter a given video for noise reduction. Magnitude features are then extracted from the gradient-filtered outcomes. Different decomposing models of hard and soft-based assignments are then addressed to separate these obtained magnitude features into oriented magnitudes subject to a given orientation range (see Section 6.7.1). Finally, robust descriptors are structured by using a simple local operator to encode the oriented magnitudes (see Section 6.7.2). Experiments for DT classification have validated the good performance of oriented magnitudes compared to Gaussian-based filtered features in [C2, C5] (see Section 6.7.3.3). Hereafter, we express above proposed processes in detail.

### 6.7.1 Oriented magnitudes of Gaussian gradients

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

$$\begin{cases} \mathcal{I}_\sigma^{\partial x^k}(\mathbf{q}) = G_{\sigma, \partial x^k}^{2D}(x, y) * \mathcal{I}(\mathbf{q}) \\ \mathcal{I}_\sigma^{\partial y^k}(\mathbf{q}) = G_{\sigma, \partial y^k}^{2D}(x, y) * \mathcal{I}(\mathbf{q}) \end{cases} \quad (6.22)$$

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

$$\begin{cases} \mathcal{V}_\sigma^{\partial x^k}(\mathbf{u}) = G_{\sigma, \partial x^k}^{3D}(x, y, z) * \mathcal{V}(\mathbf{u}) \\ \mathcal{V}_\sigma^{\partial y^k}(\mathbf{u}) = G_{\sigma, \partial y^k}^{3D}(x, y, z) * \mathcal{V}(\mathbf{u}) \\ \mathcal{V}_\sigma^{\partial z^k}(\mathbf{u}) = G_{\sigma, \partial z^k}^{3D}(x, y, z) * \mathcal{V}(\mathbf{u}) \end{cases} \quad (6.23)$$

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

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

**Quantification strategies:** In consideration of an uniform quantification of an oriented feature  $f$ , which is defined at an arbitrary pixel  $\mathbf{q}$  as  $f(\mathbf{q})$ , into  $n$  bins, it can be decomposed into two components: orientation  $\bar{f}(\mathbf{q}) \in [0, 2\pi)$  and magnitude  $\|f(\mathbf{q})\|$ . Let us suppose that  $(i-1)\lambda \leq \bar{f}(\mathbf{q}) < i\lambda$ , where  $i \in \{1, 2, \dots, n\}$  and  $\lambda = \frac{2\pi}{n}$ . Traditional methods address two possible strategies for decomposition of  $f$  into  $n$  bins:  $\{m_i\}_{i=1}^n$ .

- **Hard assignment:**  $f(\mathbf{q})$  is totally assigned to bin  $m_i$  with value  $\|f(\mathbf{q})\|$ .
- **Soft assignment:**  $f(\mathbf{q})$  is partially assigned to bin  $m_i$  with value  $\frac{i\lambda - \bar{f}(\mathbf{q})}{\lambda} \|f(\mathbf{q})\|$  and to bin  $m_{i+1}$  with value  $\frac{\bar{f}(\mathbf{q}) - (i-1)\lambda}{\lambda} \|f(\mathbf{q})\|$ , where  $m_{n+1} \equiv m_1$ .

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

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

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

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

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

suppose that  $\theta_{\sigma}^{\partial x^k, \partial y^k}(\mathbf{q}) \in d$ . Accordingly, a feature of Image of Oriented Magnitudes (IOM) could be quantified by the hard-assignment principle as

$$\text{HIOM}_{\sigma,i}^{\partial x^k, \partial y^k}(\mathbf{q}) = \|\nabla \mathcal{I}_{\sigma}^{\partial x^k, \partial y^k}(\mathbf{q})\|, \text{ so that } \theta_{\sigma}^{\partial x^k, \partial y^k}(\mathbf{q}) \in d \quad (6.24)$$

by the soft-assignment as

$$\begin{cases} \text{SIOM}_{\sigma,i}^{\partial x^k, \partial y^k}(\mathbf{q}) = \|\nabla \mathcal{I}_{\sigma}^{\partial x^k, \partial y^k}(\mathbf{q})\| \times \frac{\beta - \theta_{\sigma}^{\partial x^k, \partial y^k}(\mathbf{q})}{\beta - \alpha} \\ \text{SIOM}_{\sigma,i+1}^{\partial x^k, \partial y^k}(\mathbf{q}) = \|\nabla \mathcal{I}_{\sigma}^{\partial x^k, \partial y^k}(\mathbf{q})\| \times \frac{\theta_{\sigma}^{\partial x^k, \partial y^k}(\mathbf{q}) - \alpha}{\beta - \alpha} \end{cases} \quad (6.25)$$

and by the modified soft-assignment as

$$\begin{cases} \text{pMSIOM}_{\sigma,i}^{\partial x^k, \partial y^k}(\mathbf{q}) = \|\nabla \mathcal{I}_{\sigma}^{\partial x^k, \partial y^k}(\mathbf{q})\| \times \frac{\beta - \theta_{\sigma}^{\partial x^k, \partial y^k}(\mathbf{q})}{\beta - \alpha} \\ \text{nMSIOM}_{\sigma,i+1}^{\partial x^k, \partial y^k}(\mathbf{q}) = \|\nabla \mathcal{I}_{\sigma}^{\partial x^k, \partial y^k}(\mathbf{q})\| \times \frac{\theta_{\sigma}^{\partial x^k, \partial y^k}(\mathbf{q}) - \alpha}{\beta - \alpha} \end{cases} \quad (6.26)$$

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

$$\|\nabla \mathcal{I}_{\sigma}^{\partial x^k, \partial y^k}(\mathbf{q})\| = \sqrt{(\mathcal{I}_{\sigma}^{x^k}(\mathbf{q}))^2 + (\mathcal{I}_{\sigma}^{y^k}(\mathbf{q}))^2} \quad (6.27)$$

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

$$\theta_{\sigma}^{\partial x^k, \partial y^k}(\mathbf{q}) = \arctan(\mathcal{I}_{\sigma}^{\partial y^k}(\mathbf{q}) / \mathcal{I}_{\sigma}^{\partial x^k}(\mathbf{q})) \quad (6.28)$$

Figure 6.15 graphically illustrates an instance of decomposing the magnitudes of two Gaussian-gradient images  $\mathcal{I}_{0.5}^{\partial x^1}$  and  $\mathcal{I}_{0.5}^{\partial y^1}$  in order to obtain 4 HIOM images subject to a set of 4 equal ranges of direction  $\mathcal{D}^4 = \{[0, \pi/2), [\pi/2, \pi), [\pi, 3\pi/2), [3\pi/2, 2\pi)\}$ .

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

$$\begin{cases} \text{HVOM}_{\sigma,i}^{\partial x^k, \partial y^k}(\mathbf{u}) = \|\nabla \mathcal{V}_{\sigma}^{\partial x^k, \partial y^k}(\mathbf{u})\|, \text{ so that } \phi_{\sigma}^{\partial x^k, \partial y^k}(\mathbf{u}) \in d \\ \text{HVOM}_{\sigma,i}^{\partial y^k, \partial z^k}(\mathbf{u}) = \|\nabla \mathcal{V}_{\sigma}^{\partial y^k, \partial z^k}(\mathbf{u})\|, \text{ so that } \phi_{\sigma}^{\partial y^k, \partial z^k}(\mathbf{u}) \in d \\ \text{HVOM}_{\sigma,i}^{\partial z^k, \partial x^k}(\mathbf{u}) = \|\nabla \mathcal{V}_{\sigma}^{\partial z^k, \partial x^k}(\mathbf{u})\|, \text{ so that } \phi_{\sigma}^{\partial z^k, \partial x^k}(\mathbf{u}) \in d \end{cases} \quad (6.29)$$

by the soft-assignment as

$$\begin{cases} \text{SVOM}_{\sigma,i}^{\partial x^k, \partial y^k}(\mathbf{u}) = \|\nabla \mathcal{V}_{\sigma}^{\partial x^k, \partial y^k}(\mathbf{u})\| \times \frac{\beta - \phi_{\sigma}^{\partial x^k, \partial y^k}(\mathbf{u})}{\beta - \alpha} \\ \text{SVOM}_{\sigma,i+1}^{\partial x^k, \partial y^k}(\mathbf{u}) = \|\nabla \mathcal{V}_{\sigma}^{\partial x^k, \partial y^k}(\mathbf{u})\| \times \frac{\phi_{\sigma}^{\partial x^k, \partial y^k}(\mathbf{u}) - \alpha}{\beta - \alpha} \\ \text{SVOM}_{\sigma,i}^{\partial y^k, \partial z^k}(\mathbf{u}) = \|\nabla \mathcal{V}_{\sigma}^{\partial y^k, \partial z^k}(\mathbf{u})\| \times \frac{\beta - \phi_{\sigma}^{\partial y^k, \partial z^k}(\mathbf{u})}{\beta - \alpha} \\ \text{SVOM}_{\sigma,i+1}^{\partial y^k, \partial z^k}(\mathbf{u}) = \|\nabla \mathcal{V}_{\sigma}^{\partial y^k, \partial z^k}(\mathbf{u})\| \times \frac{\phi_{\sigma}^{\partial y^k, \partial z^k}(\mathbf{u}) - \alpha}{\beta - \alpha} \\ \text{SVOM}_{\sigma,i}^{\partial z^k, \partial x^k}(\mathbf{u}) = \|\nabla \mathcal{V}_{\sigma}^{\partial z^k, \partial x^k}(\mathbf{u})\| \times \frac{\beta - \phi_{\sigma}^{\partial z^k, \partial x^k}(\mathbf{u})}{\beta - \alpha} \\ \text{SVOM}_{\sigma,i+1}^{\partial z^k, \partial x^k}(\mathbf{u}) = \|\nabla \mathcal{V}_{\sigma}^{\partial z^k, \partial x^k}(\mathbf{u})\| \times \frac{\phi_{\sigma}^{\partial z^k, \partial x^k}(\mathbf{u}) - \alpha}{\beta - \alpha} \end{cases} \quad (6.30)$$

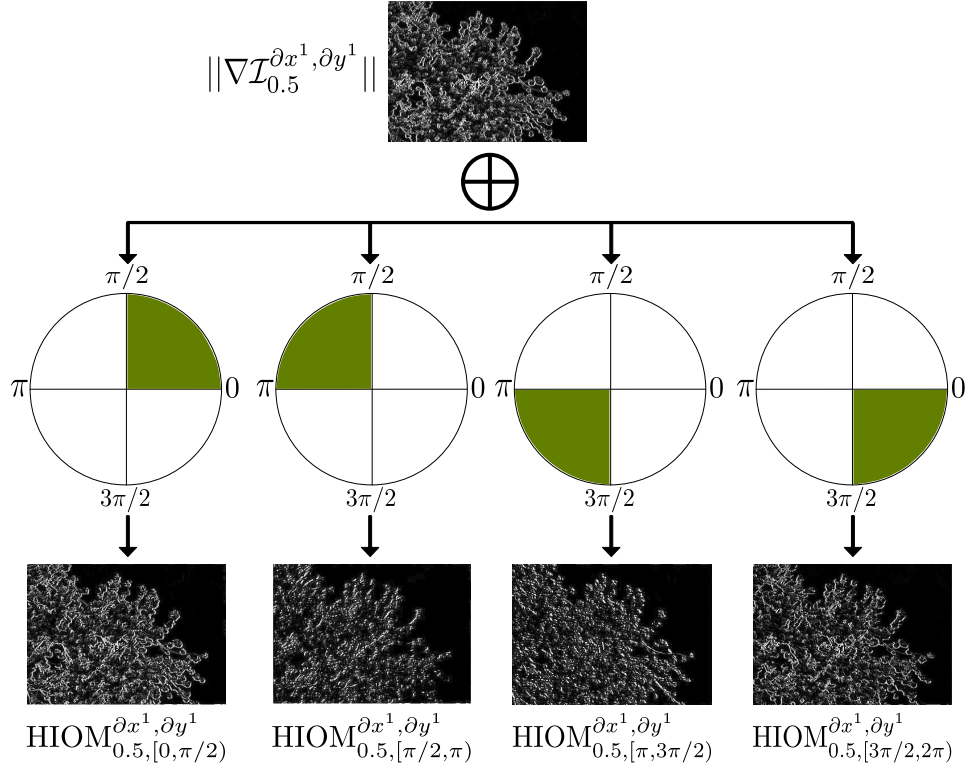


Figure 6.15: A hard-assignment model for decomposing the magnitudes of two Gaussian-gradient images  $\mathcal{I}_{0.5}^{\partial x^1}$  and  $\mathcal{I}_{0.5}^{\partial y^1}$  into 4 HIOM images subject to a set of 4 equal ranges of direction  $\mathcal{D}^4 = \{[0, \pi/2), [\pi/2, \pi), [\pi, 3\pi/2), [3\pi/2, 2\pi)\}$ .

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

In the meanwhile, a feature of VOM can be quantified to two bins by the modified soft-assignment as

$$\begin{cases} \text{pMSVOM}_{\sigma, i}^{\partial x^k, \partial y^k}(\mathbf{u}) = \|\nabla \mathcal{V}_{\sigma}^{\partial x^k, \partial y^k}(\mathbf{u})\| \times \frac{\beta - \phi_{\sigma}^{\partial x^k, \partial y^k}(\mathbf{u})}{\beta - \alpha} \\ \text{nMSVOM}_{\sigma, i+1}^{\partial x^k, \partial y^k}(\mathbf{u}) = \|\nabla \mathcal{V}_{\sigma}^{\partial x^k, \partial y^k}(\mathbf{u})\| \times \frac{\phi_{\sigma}^{\partial x^k, \partial y^k}(\mathbf{u}) - \alpha}{\beta - \alpha} \\ \text{pMSVOM}_{\sigma, i}^{\partial y^k, \partial z^k}(\mathbf{u}) = \|\nabla \mathcal{V}_{\sigma}^{\partial y^k, \partial z^k}(\mathbf{u})\| \times \frac{\beta - \phi_{\sigma}^{\partial y^k, \partial z^k}(\mathbf{u})}{\beta - \alpha} \\ \text{nMSVOM}_{\sigma, i+1}^{\partial y^k, \partial z^k}(\mathbf{u}) = \|\nabla \mathcal{V}_{\sigma}^{\partial y^k, \partial z^k}(\mathbf{u})\| \times \frac{\phi_{\sigma}^{\partial y^k, \partial z^k}(\mathbf{u}) - \alpha}{\beta - \alpha} \\ \text{pMSVOM}_{\sigma, i}^{\partial z^k, \partial x^k}(\mathbf{u}) = \|\nabla \mathcal{V}_{\sigma}^{\partial z^k, \partial x^k}(\mathbf{u})\| \times \frac{\beta - \phi_{\sigma}^{\partial z^k, \partial x^k}(\mathbf{u})}{\beta - \alpha} \\ \text{nMSVOM}_{\sigma, i+1}^{\partial z^k, \partial x^k}(\mathbf{u}) = \|\nabla \mathcal{V}_{\sigma}^{\partial z^k, \partial x^k}(\mathbf{u})\| \times \frac{\phi_{\sigma}^{\partial z^k, \partial x^k}(\mathbf{u}) - \alpha}{\beta - \alpha} \end{cases} \quad (6.31)$$

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

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

$$\begin{cases} \|\nabla \mathcal{V}_{\sigma}^{\partial x^k, \partial y^k}(\mathbf{u})\| = \sqrt{(\mathcal{V}_{\sigma}^{x^k}(\mathbf{u}))^2 + (\mathcal{V}_{\sigma}^{y^k}(\mathbf{u}))^2} \\ \|\nabla \mathcal{V}_{\sigma}^{\partial y^k, \partial z^k}(\mathbf{u})\| = \sqrt{(\mathcal{V}_{\sigma}^{y^k}(\mathbf{u}))^2 + (\mathcal{V}_{\sigma}^{z^k}(\mathbf{u}))^2} \\ \|\nabla \mathcal{V}_{\sigma}^{\partial z^k, \partial x^k}(\mathbf{u})\| = \sqrt{(\mathcal{V}_{\sigma}^{z^k}(\mathbf{u}))^2 + (\mathcal{V}_{\sigma}^{x^k}(\mathbf{u}))^2} \end{cases} \quad (6.32)$$

Figure 6.16 shows an example of computing magnitude volumes of Gaussian gradients. Gradient direc-

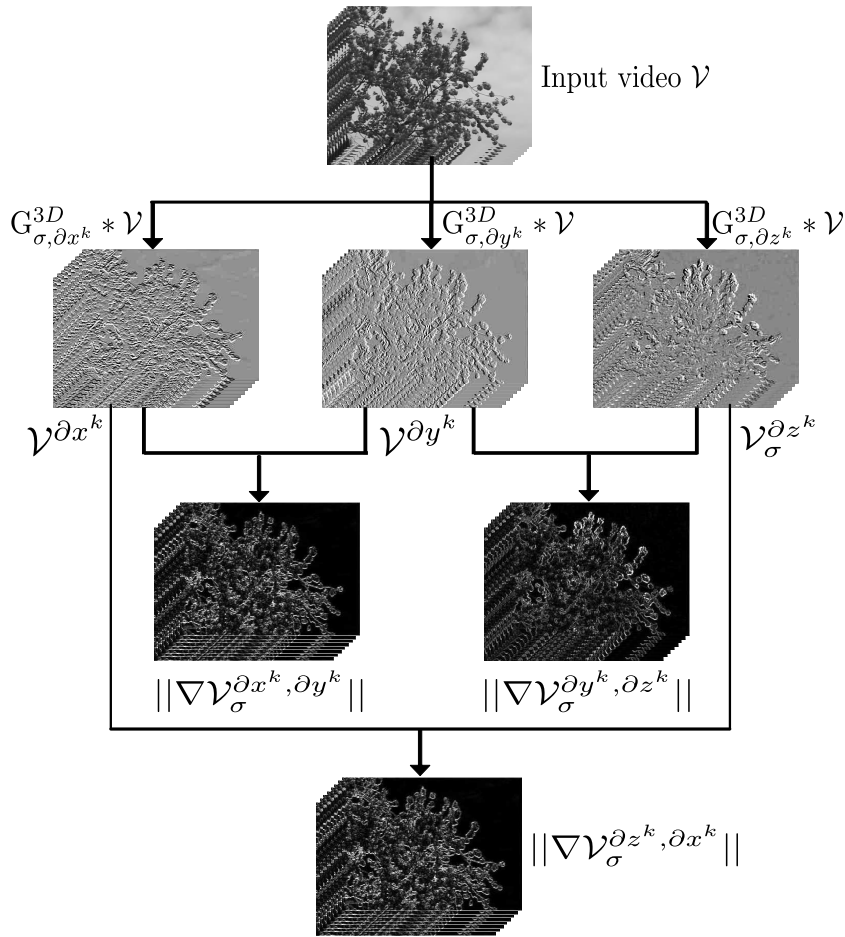


Figure 6.16: An instance of 3D Gaussian-gradient filtering and computing the obtained volumes of magnitude features.

tions  $\phi_{\sigma}^{\partial x^k, \partial y^k}(\mathbf{u})$ ,  $\phi_{\sigma}^{\partial y^k, \partial z^k}(\mathbf{u})$ , and  $\phi_{\sigma}^{\partial z^k, \partial x^k}(\mathbf{u})$  are inferred as

$$\begin{cases} \phi_{\sigma}^{\partial x^k, \partial y^k}(\mathbf{u}) = \arctan(\mathcal{V}_{\sigma}^{\partial y^k}(\mathbf{u})/\mathcal{V}_{\sigma}^{\partial x^k}(\mathbf{u})) \\ \phi_{\sigma}^{\partial y^k, \partial z^k}(\mathbf{u}) = \arctan(\mathcal{V}_{\sigma}^{\partial z^k}(\mathbf{u})/\mathcal{V}_{\sigma}^{\partial y^k}(\mathbf{u})) \\ \phi_{\sigma}^{\partial z^k, \partial x^k}(\mathbf{u}) = \arctan(\mathcal{V}_{\sigma}^{\partial x^k}(\mathbf{u})/\mathcal{V}_{\sigma}^{\partial z^k}(\mathbf{u})) \end{cases} \quad (6.33)$$

Figure 6.18 graphically illustrates a general model of decomposing a volume of magnitude features.

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

### 6.7.2 DT representation based on oriented magnitudes

In order to generally investigate oriented magnitudes for DT representation, we address the IOM and VOM computations in  $n$  ( $n \in \mathbb{Z}^+$ ) equal ranges of direction as  $\mathcal{D}^n = \{[(i-1)\lambda, i\lambda)\}_{i=1}^n$ , where  $\lambda = \frac{2\pi}{n}$  denotes an angle coefficient for decomposing the  $k$ -order image/volume magnitudes. For example, with respect to  $\lambda = \pi/2$ , we have  $n = 4$  direction ranges in equality (i.e.,  $\mathcal{D}^4 = \{[0, \pi/2), [\pi/2, \pi), [\pi, 3\pi/2), [3\pi/2, 2\pi)\}$ ) that are respectively used to decompose a magnitude image  $||\nabla \mathcal{I}_{\sigma}^{\partial x^k, \partial y^k}||$ , as shown in Figure 6.15 for an operation of the hard-assignment decomposition.

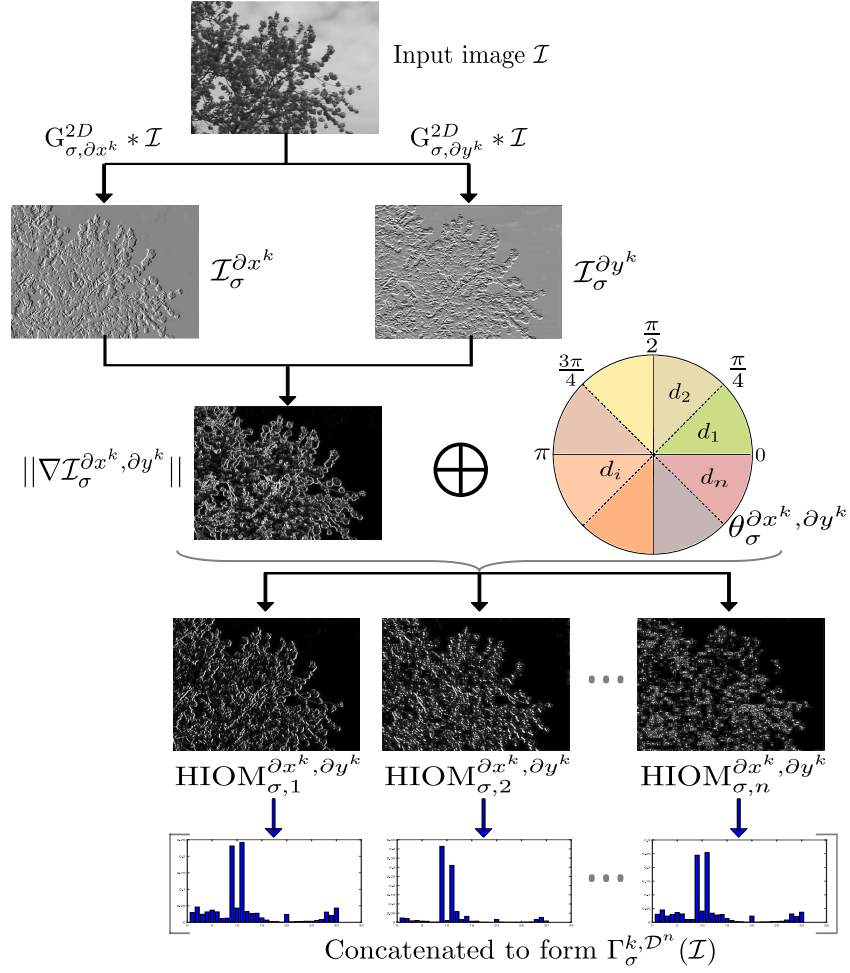


Figure 6.17: A flowchart of HIOM model subject to direction ranges  $d_i = [(i-1)\lambda, i\lambda)$  in  $\mathcal{D}^n$ . Therein, the black arrows are noted for pre-processing while the blue ones are for encoding.

Hereunder, we propose robust descriptors structured corresponding to the IOM-based and VOM-based outcomes.

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

$$\Gamma_{\sigma}^{k, \mathcal{D}^n}(f_{XY}) = \frac{1}{\mathcal{N}} \sum_{\mathcal{I} \in f_{XY}} \left[ \xi(\text{HIOM}_{\sigma,1}^{\partial x^k, \partial y^k}(\mathcal{I})), \dots, \xi(\text{HIOM}_{\sigma,n}^{\partial x^k, \partial y^k}(\mathcal{I})) \right] \quad (6.34)$$

and

$$\Upsilon_{\sigma}^{k, \mathcal{D}^n}(f_{XY}) = \frac{1}{\mathcal{N}} \sum_{\mathcal{I} \in f_{XY}} \left[ \xi(\text{SIOM}_{\sigma,1}^{\partial x^k, \partial y^k}(\mathcal{I})), \xi(\text{SIOM}_{\sigma,2}^{\partial x^k, \partial y^k}(\mathcal{I})), \dots, \xi(\text{SIOM}_{\sigma,n}^{\partial x^k, \partial y^k}(\mathcal{I})) \right] \quad (6.35)$$

and

$$\Omega_{\sigma}^{k, \mathcal{D}^n}(f_{XY}) = \frac{1}{\mathcal{N}} \sum_{\mathcal{I} \in f_{XY}} \left[ \xi(\text{pMSIOM}_{\sigma,1}^{\partial x^k, \partial y^k}(\mathcal{I})), \xi(\text{nMSIOM}_{\sigma,1}^{\partial x^k, \partial y^k}(\mathcal{I})), \dots, \xi(\text{pMSIOM}_{\sigma,n}^{\partial x^k, \partial y^k}(\mathcal{I})), \xi(\text{nMSIOM}_{\sigma,n}^{\partial x^k, \partial y^k}(\mathcal{I})) \right] \quad (6.36)$$

in which  $\mathcal{N}$  means a number of plane-images in  $f_{XY}$ ,  $\xi(\cdot)$  denotes a simple function using a local operator (e.g., LBP [81], CLBP [3], etc.) in order to figure out the corresponding histograms. Figure



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

$$\text{HIOMF}_{\sigma}^{k,D^n}(\mathcal{V}) = [\Gamma_{\sigma}^{k,D^n}(f_{XY}), \Gamma_{\sigma}^{k,D^n}(f_{XT}), \Gamma_{\sigma}^{k,D^n}(f_{YT})] \quad (6.37)$$

and

$$\text{SIOMF}_{\sigma}^{k,D^n}(\mathcal{V}) = [\Upsilon_{\sigma}^{k,D^n}(f_{XY}), \Upsilon_{\sigma}^{k,D^n}(f_{XT}), \Upsilon_{\sigma}^{k,D^n}(f_{YT})] \quad (6.38)$$

and

$$\text{MSIOMF}_{\sigma}^{k,D^n}(\mathcal{V}) = [\Omega_{\sigma}^{k,D^n}(f_{XY}), \Omega_{\sigma}^{k,D^n}(f_{XT}), \Omega_{\sigma}^{k,D^n}(f_{YT})] \quad (6.39)$$

**Proposed VOM-based descriptors:** As mentioned in Section 6.7.1 for the hard decomposition (refer to Equation 6.29), three filtered volumes of oriented magnitudes are pointed out corresponding to three pairs of spacial domains convolved on a given video  $\mathcal{V}$ . Those volumes are taken into account local analysis to construct a robust descriptor as follows. For an obtained volume  $\text{HVOM}_{\sigma,i}^{\partial x^k, \partial y^k}$ , ( $i \in \{1, 2, \dots, n\}$ ), it is firstly split into collections of filtered plane-images ( $f'_{XY}$ ,  $f'_{XT}$ , and  $f'_{YT}$ ) subject to its three orthogonal planes  $\{XY, XT, YT\}$ . The simple operator  $\xi(\cdot)$  is then utilized to capture local spatio-temporal features of DTs as

$$\Psi(\text{HVOM}_{\sigma,i}^{\partial x^k, \partial y^k}) = \left[ \frac{\sum_{\mathcal{I} \in f'_{XY}} \xi(\mathcal{I})}{\mathcal{N}_{XY}}, \frac{\sum_{\mathcal{I} \in f'_{XT}} \xi(\mathcal{I})}{\mathcal{N}_{XT}}, \frac{\sum_{\mathcal{I} \in f'_{YT}} \xi(\mathcal{I})}{\mathcal{N}_{YT}} \right] \quad (6.40)$$

in which  $\mathcal{N}_{XY}$ ,  $\mathcal{N}_{XT}$ , and  $\mathcal{N}_{YT}$  are numbers of plane-images  $f'_{XY}$ ,  $f'_{XT}$ , and  $f'_{YT}$  of  $\text{HVOM}_{\sigma,i}^{\partial x^k, \partial y^k}$  respectively. Figure 6.18 illustrates a graphical view of encoding a HVOM volume. This encoding is similarly deployed for the remaining volumes  $\text{HVOM}_{\sigma,i}^{\partial y^k, \partial z^k}$  and  $\text{HVOM}_{\sigma,i}^{\partial z^k, \partial x^k}$ . As the result, a discriminative descriptor based on the  $k$ -order HVOM features is constructed by concatenating these obtained histograms as

$$\text{HVOMF}_{\sigma}^{k,D^n}(\mathcal{V}) = \biguplus \left[ \Psi(\text{HVOM}_{\sigma,i}^{\partial x^k, \partial y^k}), \Psi(\text{HVOM}_{\sigma,i}^{\partial y^k, \partial z^k}), \Psi(\text{HVOM}_{\sigma,i}^{\partial z^k, \partial x^k}) \right]_{i=1}^n \quad (6.41)$$

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

Similarly, this HVOMF encoding could be applied to 3 SVOM (resp. 6 MSVOM) outcomes extracted by the soft decomposition (refer to Equation 6.30) subject to the direction range  $\mathcal{D}^n$ . Accordingly, other robust descriptors based on the  $k$ -order SVOM (resp. MSVOM) features are formed by concatenating the corresponding histograms as

$$\text{SVOMF}_{\sigma}^{k,D^n}(\mathcal{V}) = \biguplus \left[ \Psi(\text{SVOM}_{\sigma,i}^{\partial x^k, \partial y^k}), \Psi(\text{SVOM}_{\sigma,i}^{\partial y^k, \partial z^k}), \Psi(\text{SVOM}_{\sigma,i}^{\partial z^k, \partial x^k}) \right]_{i=1}^n \quad (6.42)$$

and

$$\begin{aligned} \text{MSVOMF}_{\sigma}^{k,D^n}(\mathcal{V}) = \biguplus \left[ \Psi(\text{pMSVOM}_{\sigma,i}^{\partial x^k, \partial y^k}), \Psi(\text{nMSVOM}_{\sigma,i}^{\partial x^k, \partial y^k}), \Psi(\text{pMSVOM}_{\sigma,i}^{\partial y^k, \partial z^k}), \right. \\ \left. \Psi(\text{nMSVOM}_{\sigma,i}^{\partial y^k, \partial z^k}), \Psi(\text{pMSVOM}_{\sigma,i}^{\partial z^k, \partial x^k}), \Psi(\text{nMSVOM}_{\sigma,i}^{\partial z^k, \partial x^k}) \right]_{i=1}^n \end{aligned} \quad (6.43)$$

Our proposed IOM/VOM-based descriptors take the following benefits to improve the performance compared to other local Gaussian-based descriptors (also see Sections 6.7.3.2, 6.7.3.3 for a comprehensive evaluation):

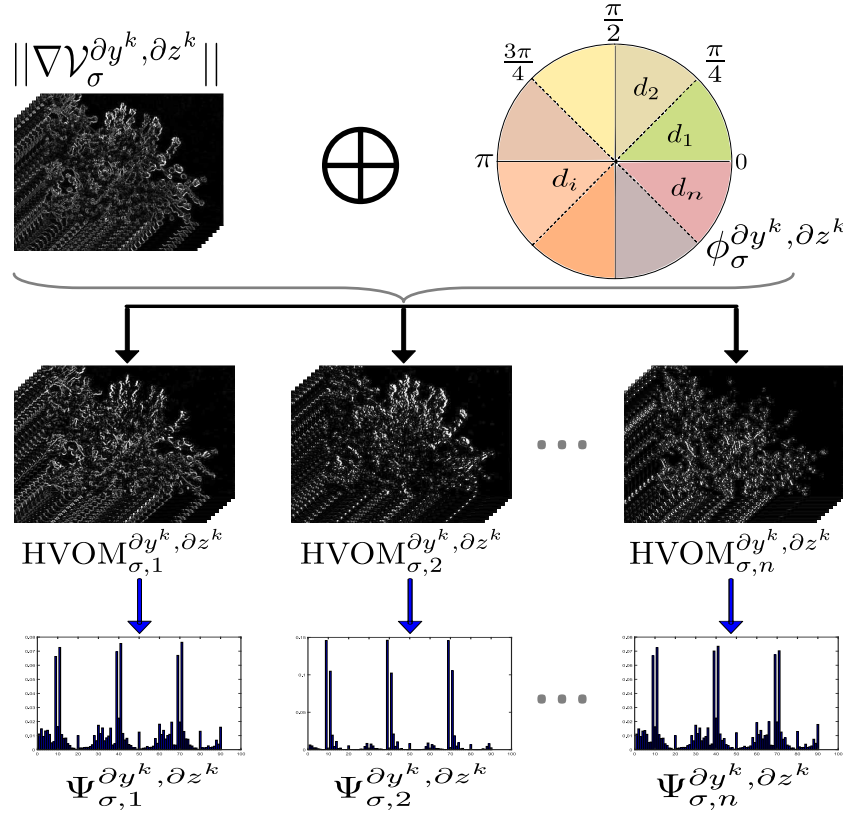


Figure 6.18: A flowchart of HVOM model subject to direction ranges  $d_i = [(i-1)\lambda, i\lambda)$  in  $\mathcal{D}^n$ . Therein, the black arrows are pre-processing steps while the blue ones are for encoding.

- Different from exploiting Gaussian-based filtered features to construct local descriptors FoSIG [C2] and V-BIG [C5], in this work, the high-order oriented magnitudes are taken into account DT representation. Thanks to the decomposing models presented in Section 6.7.1, the magnitudes of Gaussian-gradient-filtered outcomes are addressed in diversity of invariant features to enhance the robustness against the well-known issues in more effect. In the mean while, exploiting oriented features makes those outcomes still more discriminative for texture description.
- The Gaussian-gradient filterings allow to produce more filtered outcomes for the DT encoding. In the meanwhile, just one DoG-based element is used in FoSIG [C2] and V-BIG [C5] due to taking the Different of Gaussians (DoG) kernel into account the filterings.
- To enhance the discrimination power, it is possible to address the IOM/VOM-based descriptors for a multi-analysis of high-orders along with different Gaussian filtering scales, while keeping their representation in reasonable dimensions thanks to the tiny size of single-scale ones (see Table 6.2). In the meantime, just single-scale of Gaussian filtering is addressed in FoSIG [C2] and V-BIG [C5].
- It should be noted that the 2D-magnitude information (i.e., non-decomposition applied to) is also exploited in [119] for structuring textual images. However, taking it into account DT representation is not more adaptive than taking its oriented properties (see Table 6.13 for a fact of this statement). It has proved the interest of our proposed framework.

### 6.7.3 Experiments and evaluations

#### 6.7.3.1 Parameters for experimental implementation

**For computing high-order oriented magnitudes:** We investigate 2D/3D Gaussian filtering kernels in high-order gradients of  $k \in \{1, 2, 3, 4\}$ . Therein, standard deviation  $\sigma \in \{0.5, 0.7, 1, 1.3, 1.5, 2\}$  and spatio-temporal coordinates of convolution  $x, y, z \in [-3\sigma, 3\sigma]$  could be empirically conducted for

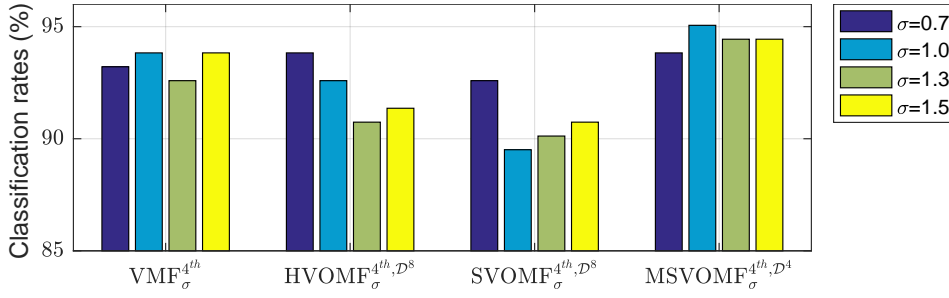


Figure 6.19: Performances (%) on *Beta* of descriptors based on the 4<sup>th</sup>-order 3D Gaussian-gradient magnitudes using both decomposing and non-decomposing models.

each Gaussian-gradient kernel in order to compute corresponding filtered outcomes. With respect to addressing direction ranges for decomposing these obtained results to achieve IOM-based images and VOM-based volumes, it can take into account various numbers of equal direction ranges, e.g.,  $n \in \{4, 6, 8\}$  corresponding to  $\lambda \in \{\pi/2, \pi/3, \pi/4\}$  respectively. Furthermore, as mentioned in Section 6.7.1 (refer to Equations. (6.24), (6.25), (6.26), (6.29), (6.30), (6.31)), the modified soft-assignment decomposition has produced a double number of oriented magnitude outcomes than the others. To take an objective evaluation in effectiveness of these decomposing models, we address  $n = 8$  (i.e.,  $\mathcal{D}^8$ ) for the traditional models (i.e., hard and soft) and  $n = 4$  (i.e.,  $\mathcal{D}^4$ ) for the modified one in order to obtain the same numbers of outcomes. This could be appropriate since for a direction range  $[0, \pi/2)$ , the soft model and its modified version respectively decompose a magnitude image into 2 SIOMs (refer to Equation (6.25)) and 4 MSIOMs (refer to Equation (6.26)) by adopting the pixels which their gradient directions are close to  $\pi/4$ . It is nearly the same that the hard model is addressed in two ranges  $[0, \pi/4)$  and  $[\pi/4, \pi/2)$  to obtain 2 HIOMs (refer to Equation (6.24)) correspondingly.

**For structuring IOM-based and VOM-based descriptors:** In order to encode the obtained outcomes of oriented magnitudes, we use a simple operator CLBP [3], one of the most popular local operator, with *riu2* mapping and local supporting region  $\{(P, R)\} = \{(8, 1)\}$ , i.e.,  $\xi = \text{CLBP}_{8,1}^{riu2}$ . To structure our proposed descriptors in reasonable dimension, the integration of “S.M/C” should be utilized for jointing CLBP’s components. That means it generally needs  $\Omega = 3(P + 2) \times 3 \times |\nabla|$  bins for representing the oriented magnitudes decomposed by a direction range, in which  $|\nabla|$  denotes a number of Gaussian-gradient magnitudes fed into a decomposing model. As a result, the final dimension to describe a DT video is subject to which the decomposing model is taken into account. For instance, using  $\mathcal{D}^8$  for the traditional decomposition (i.e.,  $n = 8$ ), dimension of single-scale  $\text{HIOMF}_\sigma^{k, \mathcal{D}^8}$  (i.e.,  $|\nabla| = 1$ ) is  $\Omega \times 8 = 720$  bins, while that of single-scale  $\text{HVOMF}_\sigma^{k, \mathcal{D}^8}$  (i.e.,  $|\nabla| = 3$ ) is  $\Omega \times 8 = 2160$  bins. Those are the same bins for  $\text{SIOMF}_\sigma^{k, \mathcal{D}^8}$  and  $\text{SVOMF}_\sigma^{k, \mathcal{D}^8}$  respectively. Due to addressing  $\mathcal{D}^4$  for the modified soft-assignment, the dimensions in single-scale analysis of  $\text{MSIOMF}_\sigma^{k, \mathcal{D}^4}$  and  $\text{MSVOMF}_\sigma^{k, \mathcal{D}^4}$  is also the same as those above, i.e.,  $\Omega \times 2 \times 4 = 720$  and  $\Omega \times 2 \times 4 = 2160$  bins respectively. Table 6.2 shows the dimensions of our descriptors in comparison with those of current local methods. Due to these tiny bins, it is possible to take advantage of the IOM/VOM-based outcomes in multi-oriented magnitudes by addressing multi-scale of standard deviations and multi-order of Gaussian-gradient kernels in order to enrich more discriminative information for improvement of their performances.

### 6.7.3.2 Assessments of effectiveness of decomposing models

As mentioned in Sections 6.7.1 and 6.7.2, corresponding to the decomposing models, we address the proposed IOM/VOM-based descriptors for DT classification task on the challenging schemes, i.e., *Beta*, *Gamma*, and DynTex++. In addition, for an objective comparison, we also take non-oriented Gaussian-gradient magnitudes into account DT representation with the same encoding parameters presented in Section 6.7.3.1 (i.e.,  $\xi = \text{CLBP}_{8,1}^{riu2}$ ) in order to construct corresponding descriptors of image/volume non-oriented magnitude features ( $\text{IMF}_\sigma^k$  and  $\text{VMF}_\sigma^k$ ). Experimental results in Tables 6.12 and 6.13 have

Table 6.12: Performances (%) of  $\text{MSVOMF}_{\sigma}^{k, \mathcal{D}^4}$  based on the modified soft-assignment in comparison with  $\text{SVOMF}_{\sigma}^{k, \mathcal{D}^8}$  based on the basic soft model.

Scheme		Beta		Gamma		DynTex++	
Order	$\{\sigma_i\}$	SVOM	MSVOM	SVOM	MSVOM	SVOM	MSVOM
$1^{st}$	$\{0.7\}$	90.74	93.21	91.29	<b>94.32</b>	96.66	<b>97.01</b>
	$\{1.0\}$	90.12	92.59	92.80	93.18	95.57	96.76
	$\{1.3\}$	89.51	92.59	92.42	92.42	95.47	96.05
	$\{1.5\}$	90.74	91.98	93.18	92.05	95.08	95.85
$2^{nd}$	$\{0.7\}$	91.36	94.44	93.56	93.18	<b>96.77</b>	96.82
	$\{1.0\}$	90.74	<b>95.06</b>	<b>94.32</b>	93.56	96.09	96.23
	$\{1.3\}$	91.98	93.83	89.77	93.94	94.88	96.28
	$\{1.5\}$	91.36	93.21	93.18	93.18	95.40	95.93
$3^{rd}$	$\{0.7\}$	<b>92.59</b>	92.59	93.94	93.18	96.23	96.81
	$\{1.0\}$	91.36	92.59	92.80	91.29	95.04	96.18
	$\{1.3\}$	91.36	93.83	92.05	93.18	94.63	96.16
	$\{1.5\}$	88.89	93.83	90.53	91.67	93.79	95.66
$4^{th}$	$\{0.7\}$	<b>92.59</b>	93.83	93.94	93.94	95.99	96.07
	$\{1.0\}$	89.51	<b>95.06</b>	90.91	<b>94.32</b>	95.46	96.57
	$\{1.3\}$	90.12	94.44	92.42	93.56	94.37	95.82
	$\{1.5\}$	90.74	94.44	93.18	<b>94.32</b>	94.44	95.62

Note: SVOM and MSVOM stand for  $\text{SVOMF}_{\sigma}^{k, \mathcal{D}^8}$  and  $\text{MSVOMF}_{\sigma}^{k, \mathcal{D}^4}$ .

Table 6.13: Classification rates (%) on the challenging schemes of descriptors based on non-oriented-magnitude and IOM/VOM-based features.

Scheme		Beta						DynTex++					
Order	$\{\sigma_i\}$	2D-H	2D-S	3D-H	3D-S	IMF	VMF	2D-H	2D-S	3D-H	3D-S	IMF	VMF
$1^{st}$	$\{0.7\}$	91.36	90.74	90.74	93.21	91.36	93.83	<b>95.77</b>	<b>96.08</b>	<b>97.13</b>	<b>97.01</b>	<b>87.99</b>	<b>93.68</b>
	$\{1.0\}$	91.36	91.36	91.98	92.59	91.98	93.21	94.72	95.73	96.18	96.76	88.92	93.19
	$\{1.3\}$	91.98	91.36	91.98	92.59	89.51	93.83	94.61	95.05	96.05	96.05	85.51	91.09
	$\{1.5\}$	89.51	91.36	91.36	91.98	91.36	92.59	93.90	94.98	95.51	95.85	86.96	91.10
$2^{nd}$	$\{0.7\}$	91.36	<b>93.83</b>	91.36	94.44	91.36	<b>94.44</b>	95.66	95.76	96.51	96.82	85.73	93.09
	$\{1.0\}$	93.21	93.21	92.59	<b>95.06</b>	<b>92.59</b>	91.98	94.88	95.39	96.44	96.23	86.03	92.10
	$\{1.3\}$	91.36	91.36	91.36	93.83	88.27	90.74	94.10	94.51	95.31	96.28	84.76	92.17
	$\{1.5\}$	90.74	92.59	93.21	93.21	90.74	92.59	94.19	94.07	95.14	95.93	83.51	91.35
$3^{rd}$	$\{0.7\}$	89.51	89.51	91.98	92.59	89.51	93.83	95.54	95.67	96.51	96.81	85.49	92.57
	$\{1.0\}$	91.36	92.59	93.21	92.59	88.89	93.83	93.52	95.34	95.82	96.18	85.71	91.88
	$\{1.3\}$	<b>95.06</b>	93.21	<b>95.06</b>	93.83	88.27	93.21	93.88	94.34	95.27	96.16	83.84	92.31
	$\{1.5\}$	90.74	91.98	93.21	93.83	90.12	90.74	94.20	94.38	94.83	95.66	85.00	91.26
$4^{th}$	$\{0.7\}$	92.59	<b>93.83</b>	93.83	93.83	90.12	93.21	94.81	95.02	96.39	96.07	85.62	93.07
	$\{1.0\}$	90.74	91.36	92.59	<b>95.06</b>	88.89	93.83	94.27	95.22	95.55	96.57	85.46	92.47
	$\{1.3\}$	90.12	90.74	90.74	94.44	89.51	92.59	93.58	94.77	95.56	95.82	86.73	<b>93.68</b>
	$\{1.5\}$	89.51	91.98	91.36	94.44	89.51	93.83	92.72	93.90	94.89	95.62	84.19	91.09

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

shown rates of these descriptors in various scale analyses. Based on those, it could be pointed out two crucial statements as follows.

- In general, it can be seen from Tables 6.12 and 6.13 that the ability of the basic soft-assignment does not perform well in decomposing Gaussian-gradient magnitudes for DT encoding compared to the hard one, even being inferior to the non-decomposing model (i.e., exploiting IMF and VMF features of non-oriented magnitudes) in some cases, e.g., DT recognition on *Beta* as shown in

Figure 6.19. It may be due to the intensified textural appearances caused by quantizing oriented magnitudes in adjacent orientation ranges instead of softly separating as in our modified model.

- As expected, our modified soft-assignment has much improved the performance compared to its original model (see Table 6.12). Furthermore, its discriminative power is significantly better than that of the non-decomposing and hard ones (see Table 6.13). This is thanks to the adjusted voting strategy as proposed in Section 6.7.1, which appropriately adopt the magnitude features subject to a given direction range to obtain filtered outcomes in more robustness for DT encoding (refer to Equations (6.26) and (6.31) for detail).

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

Table 6.14: Classification rates (%) on DT benchmark datasets of  $\text{MSIOMF}_{\sigma}^{k, \mathcal{D}^4}$  descriptor.

Dataset		UCLA				DynTex				DynTex++
Sub-set		50-LOO	50-4fold	9-class	8-class	DynTex35	Alpha	Beta	Gamma	
1 <sup>st</sup>	{0.5}	99.50	99.00	98.90	96.74	98.29	96.67	90.74	92.42	95.90
	{0.7}	99.00	99.00	<b>99.70</b>	96.63	98.29	95.00	90.74	93.94	<b>96.08</b>
	{1.0}	99.50	<b>100</b>	99.30	97.50	98.86	96.67	91.36	92.80	95.73
	{1.3}	98.50	98.50	99.05	95.11	98.57	96.67	91.36	92.80	95.05
	{1.5}	99.00	99.50	96.85	96.85	99.14	96.67	91.36	92.05	94.98
	{2.0}	<b>100</b>	<b>100</b>	98.75	96.74	98.57	<b>98.33</b>	91.98	91.67	93.77
2 <sup>nd</sup>	{0.5}	<b>100</b>	<b>100</b>	97.15	95.87	97.71	96.67	91.36	89.77	94.49
	{0.7}	<b>100</b>	<b>100</b>	98.90	97.28	98.00	96.67	93.83	93.56	95.76
	{1.0}	99.50	99.00	98.60	98.49	98.57	96.67	93.21	93.18	95.39
	{1.3}	99.50	99.50	99.25	98.15	97.71	96.67	91.36	93.56	94.51
	{1.5}	99.00	99.00	98.10	99.02	98.86	96.67	92.59	92.80	94.07
	{2.0}	99.00	99.00	98.60	97.07	97.71	96.67	91.36	93.18	93.12
3 <sup>rd</sup>	{0.5}	99.50	<b>100</b>	99.10	97.61	98.29	96.67	<b>95.06</b>	92.80	95.22
	{0.7}	99.00	99.50	98.40	97.72	98.86	96.67	89.51	<b>93.94</b>	95.67
	{1.0}	<b>100</b>	<b>100</b>	98.30	99.13	<b>99.71</b>	96.67	92.59	93.18	95.34
	{1.3}	<b>100</b>	<b>100</b>	98.45	94.67	98.57	96.67	93.21	91.29	94.34
	{1.5}	99.00	99.00	98.55	96.30	98.86	96.67	91.98	91.29	94.38
	{2.0}	99.50	99.50	98.70	98.49	98.00	96.67	93.21	92.05	92.89
4 <sup>th</sup>	{0.5}	<b>100</b>	<b>100</b>	96.35	96.96	96.29	96.67	91.36	90.53	94.35
	{0.7}	99.00	99.50	97.95	98.04	98.29	96.67	93.83	93.18	95.02
	{1.0}	99.50	<b>100</b>	98.65	98.80	92.86	96.67	91.36	90.53	95.22
	{1.3}	99.00	99.00	98.55	97.83	96.29	96.67	90.74	91.29	94.77
	{1.5}	99.50	99.50	98.45	<b>99.35</b>	98.00	96.67	91.98	92.42	93.90
	{2.0}	99.50	99.50	98.50	98.59	93.71	96.67	91.36	92.42	92.53

Note: 50-LOO and 50-4fold are results on 50-class using leave-one-out and four cross-fold validation.

### 6.7.3.3 Assessments of $\text{MSIOMF}_{\sigma}^{k, \mathcal{D}^4}$ and $\text{MSVOMF}_{\sigma}^{k, \mathcal{D}^4}$

We thoroughly discuss the significant effectiveness of taking high-order oriented magnitudes into account DT representation in comparison with other Gaussian-based filtered features. Based on the experimental results in Tables 6.14, 6.15, 6.16, 6.17, 6.18, and 6.19, it could be stated the following crucial assessments:

- Firstly, to prove the validation of our proposal, we have also implemented other local DT descriptors, named  $\text{IMF}_{\sigma}^k$  and  $\text{VMF}_{\sigma}^k$ , that are correspondingly based on the 2D/3D non-oriented magnitudes of Gaussian gradients (i.e., non-decomposing models involved in). It can be seen from Table 6.13 that  $\text{IMF}_{\sigma}^k$  and  $\text{VMF}_{\sigma}^k$  are not generally efficient compared to taking advantage of their oriented ones.
- Decomposing the Gaussian-gradient filtered outcomes in the same ranges of direction, the obtained MSVOM features are more discriminative than the MSIOM ones (see Figure 6.21 for a graphical

Table 6.15: Classification rates (%) on DT benchmark datasets of  $\text{MSVOMF}_{\sigma}^{k, \mathcal{D}^4}$  descriptor.

Dataset		UCLA				DynTex				DynTex++
Sub-set		50-LOO	50-4fold	9-class	8-class	DynTex35	Alpha	Beta	Gamma	
1 <sup>st</sup>	{0.5}	<b>100</b>	99.50	98.55	98.80	95.43	96.67	<b>95.68</b>	93.94	97.12
	{0.7}	99.50	99.50	99.60	97.50	96.57	96.67	93.21	94.32	97.01
	{1.0}	99.50	99.50	98.75	97.17	98.29	96.67	92.59	93.18	96.76
	{1.3}	99.00	99.50	97.65	97.39	98.86	96.67	92.59	92.42	96.05
	{1.5}	99.00	99.50	98.70	98.04	98.86	96.67	91.98	92.05	95.85
	{2.0}	99.50	99.50	99.35	98.04	98.86	96.67	91.98	93.18	94.01
2 <sup>nd</sup>	{0.5}	99.00	99.50	98.75	97.83	98.00	<b>98.33</b>	90.12	88.64	95.76
	{0.7}	<b>100</b>	<b>100</b>	99.40	98.04	97.14	96.67	94.44	93.18	96.82
	{1.0}	<b>100</b>	<b>100</b>	99.00	97.39	97.71	96.67	95.06	93.56	96.23
	{1.3}	<b>100</b>	<b>100</b>	98.70	97.07	98.57	96.67	93.83	93.94	96.28
	{1.5}	99.00	99.00	99.35	98.04	97.43	96.67	93.21	93.18	95.93
	{2.0}	<b>100</b>	<b>100</b>	98.50	97.93	97.71	96.67	92.59	95.08	93.89
3 <sup>rd</sup>	{0.5}	99.50	99.00	99.15	98.04	98.29	<b>98.33</b>	92.59	91.29	<b>97.13</b>
	{0.7}	99.50	99.50	98.90	97.39	98.86	96.67	92.59	93.18	96.81
	{1.0}	<b>100</b>	99.50	98.45	97.50	<b>99.43</b>	96.67	92.59	91.29	96.18
	{1.3}	<b>100</b>	<b>100</b>	99.05	96.74	98.57	96.67	93.83	93.18	96.16
	{1.5}	99.00	99.50	98.40	97.17	<b>99.43</b>	96.67	93.83	91.67	95.66
	{2.0}	98.50	99.50	98.45	96.20	98.86	96.67	93.21	93.18	93.79
4 <sup>th</sup>	{0.5}	99.50	99.50	97.80	97.07	96.29	96.67	90.12	89.39	94.34
	{0.7}	<b>100</b>	<b>100</b>	98.65	98.70	98.86	96.67	93.83	93.94	96.07
	{1.0}	<b>100</b>	<b>100</b>	98.85	98.04	97.14	96.67	95.06	94.32	96.57
	{1.3}	<b>100</b>	<b>100</b>	97.85	99.02	97.43	96.67	94.44	93.56	95.82
	{1.5}	99.50	99.50	<b>99.80</b>	98.49	96.57	96.67	94.44	94.32	95.62
	{2.0}	<b>100</b>	<b>100</b>	99.20	<b>99.24</b>	97.43	96.67	95.06	<b>95.45</b>	94.39

Note: 50-LOO and 50-4fold are results on 50-class using leave-one-out and four cross-fold validation.

 Table 6.16: Classification rates (%) on DT benchmark datasets of  $\text{MSIOMF}_{\{\sigma\}}^{k, \mathcal{D}^4}$  descriptor.

Dataset		UCLA				DynTex				DynTex++
Sub-set		50-LOO	50-4fold	9-class	8-class	DynTex35	Alpha	Beta	Gamma	
1 <sup>st</sup>	{0.5, 0.7}	99.00	99.50	97.95	95.22	98.57	95.00	93.83	93.18	93.77
	{0.5, 1.0}	99.50	99.50	99.20	<b>99.02</b>	99.14	96.67	93.83	92.42	96.67
	{0.7, 1.0}	99.00	99.50	99.25	97.72	99.14	95.00	92.59	92.80	96.72
2 <sup>nd</sup>	{0.5, 0.7}	<b>100</b>	<b>100</b>	98.85	96.52	98.57	96.67	93.83	93.56	96.66
	{0.5, 1.0}	<b>100</b>	<b>100</b>	98.45	97.28	97.14	96.67	91.98	93.94	96.63
	{0.7, 1.0}	<b>100</b>	<b>100</b>	99.15	97.17	98.86	96.67	93.83	92.80	96.45
3 <sup>rd</sup>	{0.5, 0.7}	99.50	99.50	98.90	97.39	98.57	96.67	92.59	93.56	96.69
	{0.5, 1.0}	<b>100</b>	<b>100</b>	99.25	97.61	99.43	96.67	<b>94.44</b>	93.56	<b>96.72</b>
	{0.7, 1.0}	99.50	99.50	98.65	97.83	99.14	96.67	91.98	92.80	96.36
4 <sup>th</sup>	{0.5, 0.7}	<b>100</b>	<b>100</b>	97.20	97.50	97.14	96.67	91.98	<b>94.32</b>	96.29
	{0.5, 1.0}	<b>100</b>	<b>100</b>	98.90	98.48	96.86	96.67	93.21	93.18	94.47
	{0.7, 1.0}	99.50	99.50	98.20	97.39	98.29	96.67	93.83	92.42	96.19

Note: 50-LOO and 50-4fold are results on 50-class breakdown using leave-one-out and four cross-fold validation.

view of those in settings of  $\mathcal{D}^4$  and  $\sigma = 1.3$ , see Tables 6.15 and 6.15 for other circumstances in general). This is because there are complements from the intensification of pairs of gradients in the MSVOM decomposition.

- The higher level of standard deviation  $\sigma$  is used for the Gaussian-gradient filterings, the less robustness of our  $\text{MSIOMF}_{\sigma}^{k, \mathcal{D}^4}$  and  $\text{MSVOMF}_{\sigma}^{k, \mathcal{D}^4}$  descriptors is mostly achieved. Absolutely, it can be verified in Figure 6.20 that with an increase of  $\sigma$  from 0.5 to 2, their performances on DynTex++ dataset are decreased about from 1% to 3% in general. This is due to lack of appearance features caused by the Gaussian-gradient filterings with large levels of  $\sigma$ . Hence, we mainly present results based on  $\sigma \in \{0.5, 0.7, 1\}$  in the rest of this section.
- Instead of exploiting Gaussian-based filtered characteristics as done in FoSIG [C2] and V-BIG [C5], taking the high-order oriented magnitudes into account DT representation has significantly

Table 6.17: Classification rates (%) on DT benchmark datasets of  $\text{MSVOMF}_{\{\sigma\}}^{k, \mathcal{D}^4}$  descriptor.

Dataset		UCLA				DynTex				DynTex++
Sub-set		50-LOO	50-4fold	9-class	8-class	DynTex35	Alpha	Beta	Gamma	
1 <sup>st</sup>	{0.5, 0.7}	99.50	<b>100</b>	98.85	96.41	98.00	96.67	95.06	93.18	97.36
	{0.5, 1.0}	99.50	99.50	98.80	97.50	99.43	96.67	94.44	93.56	97.28
	{0.7, 1.0}	99.50	99.50	98.10	98.48	98.86	96.67	93.83	93.18	96.78
2 <sup>nd</sup>	{0.5, 0.7}	<b>100</b>	<b>100</b>	97.75	97.50	98.86	96.67	93.83	93.94	<b>97.37</b>
	{0.5, 1.0}	<b>100</b>	<b>100</b>	98.50	97.39	98.00	96.67	93.21	93.56	97.08
	{0.7, 1.0}	<b>100</b>	<b>100</b>	98.65	97.39	97.43	96.67	93.83	93.56	97.08
3 <sup>rd</sup>	{0.5, 0.7}	99.50	99.50	98.70	98.37	99.14	96.67	92.59	93.18	97.32
	{0.5, 1.0}	<b>100</b>	99.50	98.15	97.72	<b>99.71</b>	96.67	93.21	91.67	97.06
	{0.7, 1.0}	99.50	99.50	98.70	97.28	99.43	96.67	92.59	92.04	97.25
4 <sup>th</sup>	{0.5, 0.7}	<b>100</b>	<b>100</b>	98.30	97.50	97.43	96.67	91.36	92.80	96.80
	{0.5, 1.0}	<b>100</b>	<b>100</b>	98.40	97.72	98.29	96.67	93.21	93.56	96.88
	{0.7, 1.0}	<b>100</b>	<b>100</b>	99.05	<b>99.57</b>	97.71	96.67	94.44	94.70	96.93

Note: 50-LOO and 50-4fold are results on 50-class breakdown using leave-one-out and four cross-fold validation.

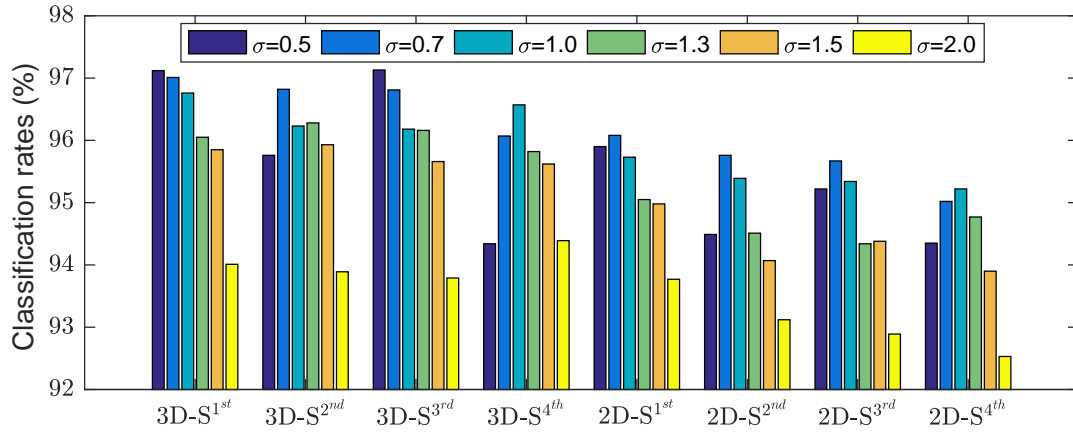


Figure 6.20: Performances on DynTex++ of high-order  $\text{MSIOMF}_{\sigma}^{k, \mathcal{D}^4}$  and  $\text{MSVOMF}_{\sigma}^{k, \mathcal{D}^4}$  descriptors (represented by  $2\text{D-S}^k$  and  $3\text{D-S}^k$  respectively) are sharply decreased when the higher level of standard deviation  $\sigma$  is used for the gradient filterings.

improved the discrimination power (see Table 6.33).

- It can be found out that for the challenging datasets (i.e., *DynTex35*, *Beta*, *Gamma*), the proposed descriptors with the odd derivatives often give better effectiveness of DT classification (see Tables 6.14, 6.15, 6.16, and 6.17). Therefore, they should be nominated for applications in practice.
- As expected in Section 6.7.2, the multi-analysis has significantly improved the discrimination power. Indeed, it can be seen from Tables 6.14, 6.15, 6.16, and 6.17, that using 2-scale of Gaussian filterings with different standard deviations, the abilities of  $\text{MSIOMF}_{\{\sigma\}}^{k, \mathcal{D}^4}$  and  $\text{MSVOMF}_{\{\sigma\}}^{k, \mathcal{D}^4}$  are enhanced and more “stable” than those of the single-scale. Also, the 2-order descriptors are better than the single-order ones (see Tables 6.14, 6.15, 6.18, and 6.19). Furthermore, an incorporation of 2-scale and 2-order features points out the best (see Tables 6.18 and 6.19).

Consequently, based on the effectiveness of  $\text{MSIOMF}_{\{\sigma\}}^{\{k\}, \mathcal{D}^4}$  and  $\text{MSVOMF}_{\{\sigma\}}^{\{k\}, \mathcal{D}^4}$  in classifying DTs, the settings of those:  $\text{MSIOMF}_{\{0.5, 1.0\}}^{\{1^{st}, 2^{nd}\}, \mathcal{D}^4}$  and  $\text{MSVOMF}_{\{0.7, 1.0\}}^{\{1^{st}, 4^{th}\}, \mathcal{D}^4}$  should be recommended for real applications as well as for comprehensive comparison with recent methods due to their best performances.

## 6.8 Representation based on Gaussian-gradient features

We hereunder express our prominent framework for DT description, as graphically illustrated in Figure 6.22. The main idea is to extract robust features of high-order 2D/3D Gaussian-gradient filtering

outcomes which are robust against noise, changes of environment, and illumination. To this end, we first compute high-order partial derivatives of the 2D/3D Gaussian filtering kernels subject to spatio-temporal directions (see Section 6.8.1). For a given video  $\mathcal{V}$ , Gaussian-gradient-based components are computed to overcome the negative impacts on DT representation (see Section 6.10.1 for their comprehensive evaluations). As the result, completed sets  $\Omega_{\mathcal{H},\sigma}^{2D/3D}$  are formed by adding Gaussian-gradient filtered elements along with their calculated magnitudes (see Section 6.8.1). In parallel, we also exploit multi-scale analysis of 2D/3D Gaussian-gradient kernels with a set of various standard deviations  $\mathcal{F}$  to forcefully investigate scale-gradient characteristics (see Section 6.8.2). Finally, robust HoGF<sup>2D/3D</sup> descriptors are introduced by addressing the typical operator CLBP on the set of complementary components  $\Omega_{\mathcal{H},\mathcal{F}}^{2D/3D}$  (see Section 6.8.2). Hereafter, we present above processes in detail.

 Table 6.18: Classification rates (%) on DT benchmark datasets of MSIOMF <sub>$\{\sigma\}$</sub>  <sup>$\{k\},\mathcal{D}^4$</sup>  descriptors.

Dataset		UCLA				DynTex				Dyn++
Sub-set		50-LOO	50-4fold	9-class	8-class	DynTex35	Alpha	Beta	Gamma	
{1 <sup>st</sup> , 2 <sup>nd</sup> }	{0.5}	<b>100</b>	<b>100</b>	97.65	97.61	97.71	96.67	92.59	93.56	96.61
	{0.7}	<b>100</b>	<b>100</b>	97.95	96.74	98.57	96.67	93.21	94.70	97.04
	{1.0}	99.50	99.50	<b>99.20</b>	97.72	98.86	96.67	93.21	94.70	97.07
{1 <sup>st</sup> , 3 <sup>rd</sup> }	{0.5}	<b>100</b>	<b>100</b>	99.15	98.15	98.86	96.67	93.83	92.42	96.68
	{0.7}	99.50	99.50	99.10	97.61	98.86	95.00	90.74	92.80	96.34
	{1.0}	99.50	99.50	98.35	97.07	99.43	96.67	92.59	92.80	96.48
{1 <sup>st</sup> , 4 <sup>th</sup> }	{0.5}	<b>100</b>	<b>100</b>	97.95	96.63	96.57	96.67	93.21	93.56	96.66
	{0.7}	<b>100</b>	<b>100</b>	98.30	<b>99.02</b>	98.29	96.67	94.44	94.70	96.89
	{1.0}	<b>100</b>	<b>100</b>	98.30	<b>99.02</b>	98.29	<b>98.33</b>	92.59	94.32	97.08
{2 <sup>nd</sup> , 3 <sup>rd</sup> }	{0.5}	<b>100</b>	99.50	97.95	98.26	98.86	96.67	93.83	92.05	96.58
	{0.7}	<b>100</b>	<b>100</b>	98.30	97.17	98.57	96.67	91.98	93.94	97.21
	{1.0}	<b>100</b>	<b>100</b>	98.95	97.50	99.14	96.67	93.21	94.70	96.93
{2 <sup>nd</sup> , 4 <sup>th</sup> }	{0.5}	<b>100</b>	<b>100</b>	96.40	96.30	97.43	<b>98.33</b>	92.59	91.29	95.45
	{0.7}	<b>100</b>	<b>100</b>	97.70	98.70	98.57	96.67	95.06	93.94	96.44
	{1.0}	<b>100</b>	<b>100</b>	99.00	98.26	97.43	96.67	93.21	93.18	96.32
{3 <sup>rd</sup> , 4 <sup>th</sup> }	{0.5}	<b>100</b>	<b>100</b>	98.25	98.04	97.71	96.67	92.59	91.29	96.21
	{0.7}	<b>100</b>	<b>100</b>	98.80	98.04	98.86	96.67	93.83	93.56	96.72
	{1.0}	<b>100</b>	<b>100</b>	99.10	98.91	99.14	96.67	93.21	94.32	96.77
{1 <sup>st</sup> , 2 <sup>nd</sup> }	{0.5, 0.7}	<b>100</b>	<b>100</b>	98.55	96.20	98.00	96.67	93.83	93.94	97.46
	{0.5, 1.0}	<b>100</b>	<b>100</b>	99.00	98.59	99.14	96.67	<b>95.68</b>	94.70	97.29
	{0.7, 1.0}	<b>100</b>	<b>100</b>	98.30	<b>99.02</b>	98.86	96.67	93.21	95.45	97.29
{1 <sup>st</sup> , 3 <sup>rd</sup> }	{0.5, 0.7}	99.00	99.50	98.95	97.50	99.14	95.00	91.98	93.56	97.44
	{0.5, 1.0}	<b>100</b>	99.50	<b>99.45</b>	96.74	99.43	96.67	95.06	92.05	97.19
	{0.7, 1.0}	99.50	99.50	98.20	98.59	<b>99.71</b>	96.67	91.36	91.67	97.10
{1 <sup>st</sup> , 4 <sup>th</sup> }	{0.5, 0.7}	<b>100</b>	<b>100</b>	98.70	97.61	97.14	96.67	93.21	<b>95.83</b>	97.19
	{0.5, 1.0}	<b>100</b>	<b>100</b>	98.45	97.83	98.00	<b>98.33</b>	94.44	95.08	97.26
	{0.7, 1.0}	<b>100</b>	<b>100</b>	99.20	98.80	98.00	96.67	94.44	95.45	<b>97.57</b>
{2 <sup>nd</sup> , 3 <sup>rd</sup> }	{0.5, 0.7}	<b>100</b>	<b>100</b>	96.85	97.72	98.86	96.67	93.21	93.94	97.23
	{0.5, 1.0}	<b>100</b>	<b>100</b>	98.70	96.30	99.43	96.67	94.44	93.18	97.35
	{0.7, 1.0}	<b>100</b>	<b>100</b>	98.95	98.37	98.86	96.67	92.59	94.70	97.37
{2 <sup>nd</sup> , 4 <sup>th</sup> }	{0.5, 0.7}	<b>100</b>	<b>100</b>	98.20	97.28	98.00	96.67	91.36	94.70	96.78
	{0.5, 1.0}	<b>100</b>	<b>100</b>	97.50	97.61	98.00	96.67	92.59	94.32	96.95
	{0.7, 1.0}	<b>100</b>	<b>100</b>	98.95	97.50	98.57	96.67	94.44	94.32	96.67
{3 <sup>rd</sup> , 4 <sup>th</sup> }	{0.5, 0.7}	<b>100</b>	<b>100</b>	98.50	97.17	98.57	96.67	92.59	95.08	97.29
	{0.5, 1.0}	<b>100</b>	<b>100</b>	97.35	98.48	98.86	96.67	94.44	94.70	97.26
	{0.7, 1.0}	<b>100</b>	<b>100</b>	98.95	97.61	99.14	96.67	93.83	94.70	97.12

Note: 50-LOO and 50-4fold are results on 50-class breakdown using leave-one-out and four cross-fold validation. Dyn++ stands for DynTex++.



Table 6.19: Classification rates (%) on DT benchmark datasets of  $\text{MSVOMF}_{\{\sigma\}}^{k, \mathcal{D}^4}$  descriptors.

Dataset		UCLA				DynTex				DynTex++
Sub-set		50-LOO	50-4fold	9-class	8-class	DynTex35	Alpha	Beta	Gamma	
$\{1^{st}, 2^{nd}\}$	{0.5}	<b>100</b>	<b>100</b>	98.35	96.63	98.57	96.67	95.68	93.18	97.07
	{0.7}	<b>100</b>	<b>100</b>	<b>99.70</b>	<b>99.57</b>	98.86	96.67	95.06	95.08	97.18
	{1.0}	99.50	99.50	98.25	98.15	98.57	96.67	94.44	95.08	97.28
$\{1^{st}, 3^{rd}\}$	{0.5}	99.50	<b>100</b>	98.45	98.59	<b>99.71</b>	96.67	95.06	92.05	97.25
	{0.7}	99.50	99.50	99.40	98.70	98.86	96.67	92.59	93.94	97.48
	{1.0}	<b>100</b>	99.00	98.55	98.04	98.29	96.67	92.59	92.42	97.17
$\{1^{st}, 4^{th}\}$	{0.5}	<b>100</b>	<b>100</b>	98.95	97.28	97.71	96.67	93.21	93.18	97.08
	{0.7}	<b>100</b>	<b>100</b>	98.40	97.50	<b>99.71</b>	96.67	94.44	94.32	97.28
	{1.0}	<b>100</b>	<b>100</b>	98.25	99.46	<b>99.71</b>	96.67	95.06	94.70	97.32
$\{2^{nd}, 3^{rd}\}$	{0.5}	99.50	99.50	99.30	96.63	98.00	96.67	93.83	93.18	97.08
	{0.7}	<b>100</b>	99.50	98.35	98.26	98.86	96.67	93.83	95.08	97.59
	{1.0}	<b>100</b>	<b>100</b>	97.95	99.13	98.57	96.67	93.21	<b>95.45</b>	97.27
$\{2^{nd}, 4^{th}\}$	{0.5}	99.50	99.50	97.90	97.17	97.14	96.67	90.12	89.77	96.13
	{0.7}	<b>100</b>	<b>100</b>	98.85	98.37	98.00	96.67	94.44	95.08	97.29
	{1.0}	<b>100</b>	<b>100</b>	<b>99.70</b>	99.13	96.86	96.67	<b>96.30</b>	95.08	96.92
$\{3^{rd}, 4^{th}\}$	{0.5}	99.50	99.50	98.90	97.93	98.86	96.67	93.83	93.18	97.00
	{0.7}	<b>100</b>	<b>100</b>	98.50	<b>99.57</b>	99.14	96.67	95.06	94.32	97.36
	{1.0}	<b>100</b>	<b>100</b>	99.50	98.48	99.14	96.67	93.83	94.70	97.07
$\{1^{st}, 2^{nd}\}$	{0.5, 0.7}	<b>100</b>	<b>100</b>	97.70	97.07	99.43	96.67	95.06	94.32	97.57
	{0.5, 1.0}	<b>100</b>	<b>100</b>	98.15	98.70	99.14	96.67	94.44	93.56	97.73
	{0.7, 1.0}	99.50	<b>100</b>	99.40	99.02	98.86	96.67	95.06	<b>95.45</b>	97.42
$\{1^{st}, 3^{rd}\}$	{0.5, 0.7}	99.50	99.50	98.65	96.96	99.14	96.67	95.06	92.42	97.40
	{0.5, 1.0}	99.50	99.50	97.95	97.83	99.14	96.67	94.44	92.42	97.43
	{0.7, 1.0}	99.50	99.50	98.95	99.13	98.86	96.67	93.83	92.80	97.27
$\{1^{st}, 4^{th}\}$	{0.5, 0.7}	<b>100</b>	<b>100</b>	98.35	96.74	98.29	96.67	95.06	93.56	97.36
	{0.5, 1.0}	<b>100</b>	<b>100</b>	98.80	97.93	99.43	96.67	95.68	93.94	97.76
	{0.7, 1.0}	<b>100</b>	<b>100</b>	99.35	99.35	<b>99.71</b>	96.67	<b>96.30</b>	95.08	<b>97.87</b>
$\{2^{nd}, 3^{rd}\}$	{0.5, 0.7}	99.50	99.50	98.25	96.52	99.43	96.67	95.06	93.94	97.73
	{0.5, 1.0}	99.50	99.50	98.75	98.59	<b>99.71</b>	96.67	93.83	93.56	97.64
	{0.7, 1.0}	<b>100</b>	99.50	98.45	97.39	99.14	96.67	94.44	94.32	97.52
$\{2^{nd}, 4^{th}\}$	{0.5, 0.7}	<b>100</b>	<b>100</b>	97.50	97.83	98.57	96.67	92.59	94.70	97.17
	{0.5, 1.0}	<b>100</b>	<b>100</b>	97.70	96.41	97.71	96.67	92.59	93.94	97.16
	{0.7, 1.0}	<b>100</b>	<b>100</b>	98.65	99.02	97.71	96.67	95.06	94.70	97.36
$\{3^{rd}, 4^{th}\}$	{0.5, 0.7}	99.50	99.50	98.95	97.28	99.14	96.67	95.06	93.94	97.47
	{0.5, 1.0}	<b>100</b>	<b>100</b>	98.70	98.15	<b>99.71</b>	96.67	95.68	93.94	97.43
	{0.7, 1.0}	<b>100</b>	<b>100</b>	98.85	98.91	99.43	96.67	95.68	94.32	97.78

Note: 50-LOO and 50-4fold denote results on 50-class breakdown using leave-one-out and four cross-fold validation.

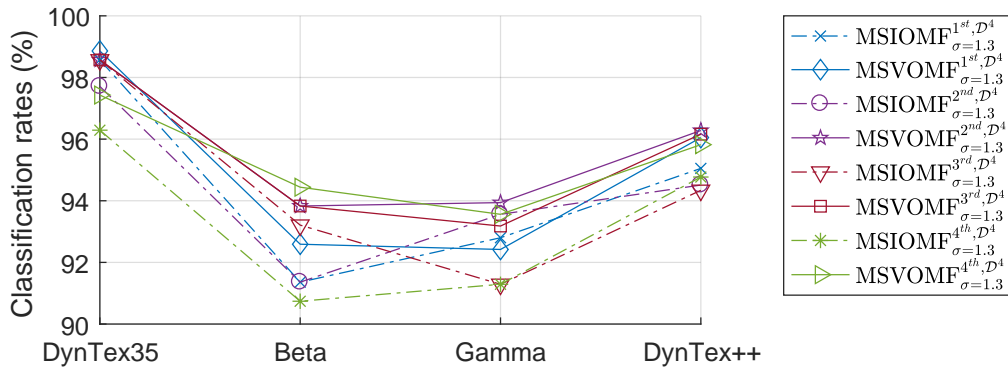


Figure 6.21: A comprehensive comparison in pairs of high-order  $\text{MSIOMF}_{\sigma=1.3}^{k, \mathcal{D}^4}$  and  $\text{MSVOMF}_{\sigma=1.3}^{k, \mathcal{D}^4}$  descriptors.

### 6.8.1 High-order Gaussian-gradient Filtered Components

Motivated by the meaningful contributions of filter-bank methods [2, C2, C5], in this work, we exploit the robust and discriminative outcomes of high-order partial derivatives of 2D/3D Gaussian filtering

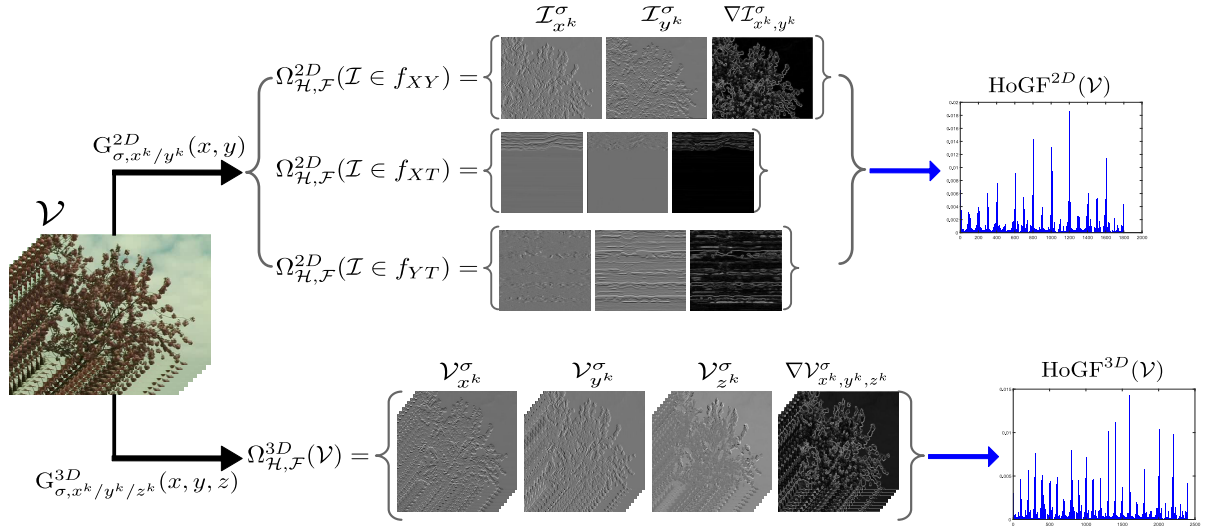


Figure 6.22: Our proposed framework for structuring an input video  $\mathcal{V}$  based on Gaussian-gradient filtered components. Therein, the black arrows denote preprocessing steps using Gaussian-gradient kernels, while the blue ones imply processes of DT encoding.

kernels on struggling against the well-known issues in DT representation. It is noticed that the filtering is also applicable for the higher  $n$ -dimensional Gaussian kernels ( $n > 3$ ). In the prior works, FoSIG [C2] and V-BIG [C5] descriptors were introduced by taking advantage of Gaussian-based features to improve the discrimination of DT encoding. Despite achieving considerable results, those issues have been dealt with incompletely. By exploiting high-order 2D/3D Gaussian-gradient filtered components  $\Omega_{\mathcal{H}, \sigma}^{2D/3D}$  instead of the non-derivative ones as done in [C2, C5], we propose  $\text{HoGF}^{2D/3D}$  descriptors with the performances improved clearly (see Section 6.8.2). Hereafter, we evidently present how to form completed sets  $\Omega_{\mathcal{H}, \sigma}^{2D/3D}$  for DT representation.

According to Equation (6.4), the  $k^{\text{th}}$ -order gradients ( $k \in \mathbb{Z}$  and  $k > 0$ ) of a 2D/3D Gaussian filtering kernel subject to spatial coordinates (i.e.,  $\{x, y\}$  for the 2D kernel and  $\{x, y, z\}$  for the 3D) are computed as follows.

$$\begin{cases} G_{\sigma, x^k}^{2D}(x, y) = \frac{\partial^k G_{\sigma}^{2D}(x, y)}{\partial x^k} \\ G_{\sigma, y^k}^{2D}(x, y) = \frac{\partial^k G_{\sigma}^{2D}(x, y)}{\partial y^k} \end{cases} \quad (6.44) \quad \begin{cases} G_{\sigma, x^k}^{3D}(x, y, z) = \frac{\partial^k G_{\sigma}^{3D}(x, y, z)}{\partial x^k} \\ G_{\sigma, y^k}^{3D}(x, y, z) = \frac{\partial^k G_{\sigma}^{3D}(x, y, z)}{\partial y^k} \\ G_{\sigma, z^k}^{3D}(x, y, z) = \frac{\partial^k G_{\sigma}^{3D}(x, y, z)}{\partial z^k} \end{cases} \quad (6.45)$$

where  $x, y, z$  denote the spatial and temporal axes.

Given a gray-scale image  $\mathcal{I}$  (correspondingly a video  $\mathcal{V}$ ) and a pre-defined standard deviation  $\sigma$ , two 2D Gaussian-gradient filtering kernels  $G_{\sigma, x^k}^{2D}$  and  $G_{\sigma, y^k}^{2D}$  are taken into account as a pre-processing analysis to produce the following  $k^{\text{th}}$ -order gradient-filtered images as Equation (6.46)

$$\begin{cases} \mathcal{I}_{x^k}^\sigma = |G_{\sigma, x^k}^{2D}(x, y) * \mathcal{I}| \\ \mathcal{I}_{y^k}^\sigma = |G_{\sigma, y^k}^{2D}(x, y) * \mathcal{I}| \end{cases} \quad (6.46) \quad \begin{cases} \mathcal{V}_{x^k}^\sigma = |G_{\sigma, x^k}^{3D}(x, y, z) * \mathcal{V}| \\ \mathcal{V}_{y^k}^\sigma = |G_{\sigma, y^k}^{3D}(x, y, z) * \mathcal{V}| \\ \mathcal{V}_{z^k}^\sigma = |G_{\sigma, z^k}^{3D}(x, y, z) * \mathcal{V}| \end{cases} \quad (6.47)$$

where “ $*$ ” is a convoluting operator. Figure 6.2(a) shows an instance of this 2D filtering. Correspondingly, the  $k^{\text{th}}$ -order gradient-filtered volumes are also figured out by convoluting 3D kernels  $G_{\sigma, x^k}^{3D}$ ,  $G_{\sigma, y^k}^{3D}$ , and  $G_{\sigma, z^k}^{3D}$  on video  $\mathcal{V}$  as Equation (6.46). Figure 6.2(b) shows an example of this computational 3D filtering.

Furthermore, in order to forcefully capture more intensive features for DT encoding, the magnitude

property  $\nabla \mathcal{I}$  (correspondingly  $\nabla \mathcal{V}$ ) of these 2D (3D) Gaussian-gradient filtered images (volumes) is taken into account the video analysis. They are calculated as follows. It can be observed in Figure 6.2 for a specific instance of this filtering computation.

$$\nabla \mathcal{I}_{x^k, y^k}^\sigma = \sqrt{(\mathcal{I}_{x^k}^\sigma)^2 + (\mathcal{I}_{y^k}^\sigma)^2} \quad , \quad \nabla \mathcal{V}_{x^k, y^k, z^k}^\sigma = \sqrt{(\mathcal{V}_{x^k}^\sigma)^2 + (\mathcal{V}_{y^k}^\sigma)^2 + (\mathcal{V}_{z^k}^\sigma)^2} \quad (6.48)$$

As the result of those, for an input image  $\mathcal{I}$  (correspondingly an input video  $\mathcal{V}$ ), a completed set  $\Omega_{\mathcal{H}, \sigma}^{2D}$  ( $\Omega_{\mathcal{H}, \sigma}^{3D}$ ) of high-order 2D (3D) Gaussian-gradient complemented components is constructed as follows in order to completely investigate the robust and discriminative features for DT representation against the well-known encoding problems: changes of environmental elements, illumination, and noise.

$$\Omega_{\mathcal{H}, \sigma}^{2D} = \{\mathcal{I}_{x^k}^\sigma, \mathcal{I}_{y^k}^\sigma, \nabla \mathcal{I}_{x^k, y^k}^\sigma\}_{k=1}^m \quad , \quad \Omega_{\mathcal{H}, \sigma}^{3D} = \{\mathcal{V}_{x^k}^\sigma, \mathcal{V}_{y^k}^\sigma, \mathcal{V}_{z^k}^\sigma, \nabla \mathcal{V}_{x^k, y^k, z^k}^\sigma\}_{k=1}^m \quad (6.49)$$

where  $m = |\mathcal{H}|$  is a positive integer that denotes a number of computing derivations involved with a particular DT encoding. For example,  $m = 2$  means that two different orders of derivative operations are taken into account the current computation.

It should be noted that the 2D Gaussian-gradient filtered components subject to separative directions (i.e.,  $\mathcal{I}_{x^k}^\sigma$  and  $\mathcal{I}_{y^k}^\sigma$ ) are not exploited in [119], where only the intensive magnitude information is addressed for encoding textual images. In the meanwhile, the  $\mathcal{I}_{x^k}^\sigma$  and  $\mathcal{I}_{y^k}^\sigma$  properties are significant in improving the performance. Indeed, our experiments for DT classification have verified their productive contributions in enriching more gradient-filtered patterns for DT description (see Table 6.24 for a certain confirmation). In order to be convenient for further presentation,  $\Omega_{\mathcal{H}, \sigma}^{2D/3D}$  is henceforward an abbreviation of high-order Gaussian-gradient filtered images  $\Omega_{\mathcal{H}, \sigma}^{2D}$  and volumes  $\Omega_{\mathcal{H}, \sigma}^{3D}$  in general.

### 6.8.2 DT Representation Based on $\Omega_{\mathcal{H}, \sigma}^{2D/3D}$ Components

As revealed in Section 6.8.1, the filtered elements in  $\Omega_{\mathcal{H}, \sigma}^{2D/3D}$  are robust against the well-known shortcomings of DT representation. They also are complementary to each other in enriching shape and motion clues. In this section, we take advantage of these beneficial properties along with multi-scale Gaussian-based filtering analysis in order to effectively capture high-order Gaussian-gradient features for enhancing the discriminative ability. Appropriately, let  $\mathcal{V}$  denote an input video;  $\mathcal{F} = \{\sigma_i\}_{i=1}^l$  be a set of pre-defined standard deviations, where  $l \in \mathbb{Z}^+$  indicates a number of Gaussian filtering scales involved with. To be compliant with types of  $\Omega_{\mathcal{H}, \mathcal{F}}^{2D/3D}$  components, we hereafter conduct two corresponding DT descriptions subject to the completed supporting components of multi-scale high-order 2D/3D Gaussian-gradients (i.e.,  $\Omega_{\mathcal{H}, \mathcal{F}}^{2D}$  and  $\Omega_{\mathcal{H}, \mathcal{F}}^{3D}$ ). According to that, two significant descriptors with high performances are constructed as follows.

**Proposed HoGF<sup>2D</sup> descriptor:** To be in accordance with structuring the completed set of high-order 2D Gaussian-gradient components  $\Omega_{\mathcal{H}, \mathcal{F}}^{2D}$ , we firstly partition the video  $\mathcal{V}$  into separated collections of plane-images  $\{f_{XY}, f_{XT}, f_{YT}\}$  subject to its three orthogonal planes  $\{XY, XT, YT\}$  (see Figure 6.22 for a visual demonstration). After that, Equations (6.46) and (6.48) are used in order to compute high-order 2D Gaussian-gradient filtered components based on these plane-image collections, i.e., correspondingly  $\Omega_{\mathcal{H}, \mathcal{F}}^{2D}(f_{XY})$ ,  $\Omega_{\mathcal{H}, \mathcal{F}}^{2D}(f_{XT})$ , and  $\Omega_{\mathcal{H}, \mathcal{F}}^{2D}(f_{YT})$ . A simple operator  $\Psi(\cdot)$  is then utilized to locally analyze the obtained components in order to efficiently capture high-order 2D Gaussian-gradient features  $\Gamma(\cdot)$  of multi-scale spatio-temporal appearances. For example, with respect to a plane-image  $\mathcal{I} \in f_{XY}$ , properties of  $\Gamma(\cdot)$  are structured as

$$\Gamma(\mathcal{I}, \Omega_{\mathcal{H}, \mathcal{F}}^{2D}(\mathcal{I})) = \biguplus_{k=1}^m \left[ \Psi(\mathcal{I}_{x^k}^{\sigma_i}), \Psi(\mathcal{I}_{y^k}^{\sigma_i}), \Psi(\nabla \mathcal{I}_{x^k, y^k}^{\sigma_i}) \right]_{i=1}^l \quad (6.50)$$

where  $\biguplus$  indicates an operation of concatenating 2D Gaussian-gradient filtered features addressed by  $m = |\mathcal{H}|$  different orders of partial derivatives with respect to the spatial domain  $\{x, y\}$ ,  $\Psi(\cdot)$  denotes

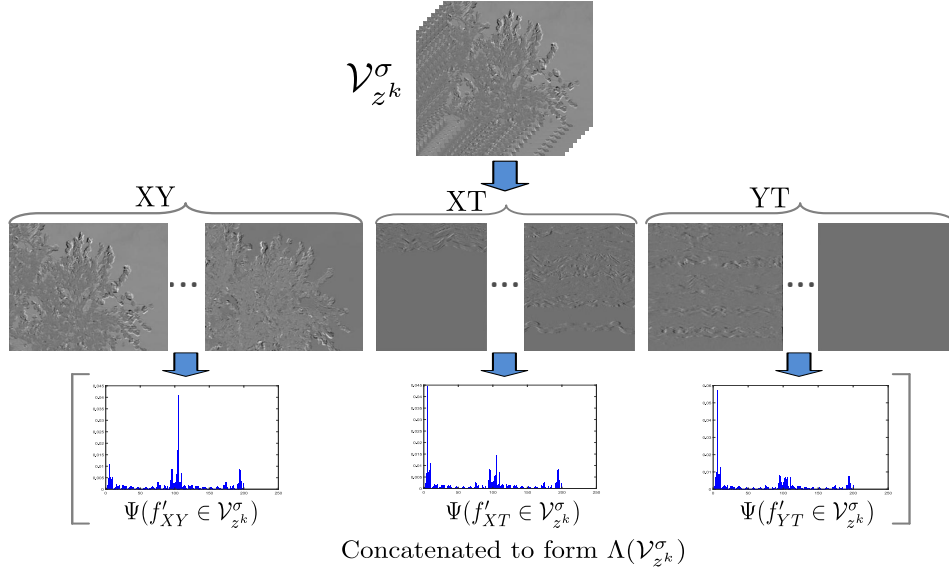


Figure 6.23: Illustration of using a simple operator  $\Psi = \text{CLBP}_{\{8,1\}}^{\text{riu2}}$  to structure the first-order Gaussian-gradient filtered volume  $\mathcal{V}_{z^k}^\sigma$ , which is extracted by convolving a gradient-kernel  $\mathcal{G}_{\sigma, z^k}^{3D}(x, y, z)$  on the temporal direction  $z$  of a given video  $\mathcal{V}$  with  $\sigma = 0.7$  and  $k = 1$ .

a local operator involved in the encoding, e.g., LBP [81], CLBP [3], etc. Finally, the achieved probability distributions are concatenated and normalized to construct a forceful descriptor of High-order 2D Gaussian-gradient-based Features ( $\text{HoGF}^{2D}$ ) as follows.

$$\text{HoGF}^{2D}(\mathcal{V}) = \left[ \Gamma(f_{XY}, \Omega_{\mathcal{H}, \mathcal{F}}^{2D}(f_{XY})), \Gamma(f_{XT}, \Omega_{\mathcal{H}, \mathcal{F}}^{2D}(f_{XT})), \Gamma(f_{YT}, \Omega_{\mathcal{H}, \mathcal{F}}^{2D}(f_{YT})) \right] \quad (6.51)$$

**Proposed  $\text{HoGF}^{3D}$  descriptor:** Similar to computing on  $\Omega_{\mathcal{H}, \mathcal{F}}^{2D}$ , we firstly address Equations (6.47) and (6.48) for preprocessing video  $\mathcal{V}$  to calculate Gaussian-gradient volumes  $\Omega_{\mathcal{H}, \mathcal{F}}^{3D}$ . For each volume  $\mathcal{V}_G \in \Omega_{\mathcal{H}, \mathcal{F}}^{3D}$ , let  $f'_{XY}$ ,  $f'_{XT}$ , and  $f'_{YT}$  be collections of plane-images separated subject to three orthogonal planes of  $\mathcal{V}_G$ . The complementary filtered component is then encoded by applying the simple operator  $\Psi(\cdot)$  to efficiently extract shape and motion cues of DTs as

$$\Lambda(\mathcal{V}_G) = \left[ \frac{\sum_{\mathcal{I} \in f'_{XY}} \Psi(\mathcal{I})}{|f'_{XY}|}, \frac{\sum_{\mathcal{I} \in f'_{XT}} \Psi(\mathcal{I})}{|f'_{XT}|}, \frac{\sum_{\mathcal{I} \in f'_{YT}} \Psi(\mathcal{I})}{|f'_{YT}|} \right] \quad (6.52)$$

where  $|f'_{XY}|$ ,  $|f'_{XT}|$ , and  $|f'_{YT}|$  denote the number of plane-image collections  $f'_{XY}$ ,  $f'_{XT}$ , and  $f'_{YT}$  respectively. Figure 6.23 shows an instance of a visual computation for the first-order Gaussian-gradient filtered volume  $\mathcal{V}_{z^1}^{0.7}$  using  $\Psi = \text{CLBP}_{\{8,1\}}^{\text{riu2}}$ . Finally, the obtained histograms are concatenated and normalized to construct a robust descriptor of High-order 3D Gaussian-gradient-based Features ( $\text{HoGF}^{3D}$ ) as

$$\text{HoGF}^{3D}(\mathcal{V}, \Omega_{\mathcal{H}, \mathcal{F}}^{3D}(\mathcal{V})) = \left[ \biguplus_{k=1}^m \left[ \Lambda(\mathcal{V}_{x^k}^{\sigma_i}), \Lambda(\mathcal{V}_{y^k}^{\sigma_i}), \Lambda(\mathcal{V}_{z^k}^{\sigma_i}), \Lambda(\nabla \mathcal{V}_{x^k, y^k, z^k}^{\sigma_i}) \right] \right]_{i=1}^l \quad (6.53)$$

where  $\biguplus$  indicates a concatenation of different 3D Gaussian-gradient filtered features addressed by  $m = |\mathcal{H}|$  different orders of partial derivatives in respect of the spatial and temporal directions  $\{x, y, z\}$ . For a convenience in further presentation,  $\text{HoGF}^{2D/3D}$  could be denoted as an abbreviation of the proposed DT descriptors in general.

Furthermore, in order to evaluate the performance of our proposed framework, we also investigate 2D/3D Gaussian filtering kernels without derivatives for a comprehensive comparison in DT representation. According to structuring the  $\text{HoGF}^{2D}$  descriptor, the Zero-order 2D Gaussian-gradient Features

(ZoGF<sup>2D</sup>) are captured as

$$\text{ZoGF}^{2D}(\mathcal{V}) = \left[ \Psi(f_{XY}^{\sigma_i}), \Psi(f_{XT}^{\sigma_i}), \Psi(f_{YT}^{\sigma_i}) \right]_{i=1}^l \quad (6.54)$$

in which  $f_{XY}^{\sigma_i}$ ,  $f_{XT}^{\sigma_i}$ , and  $f_{YT}^{\sigma_i}$  are sets of filtered plane-images of  $\mathcal{V}$  that are computed by using a 2D non-Gaussian-gradient filtering kernel with  $\sigma_i$ . Similarly, ZoGF<sup>3D</sup> for 3D non-Gaussian-gradient Features is formed as follows.

$$\text{ZoGF}^{3D}(\mathcal{V}) = \left[ \Lambda(\mathcal{V}_G^{\sigma_i}) \right]_{i=1}^l \quad (6.55)$$

where  $\mathcal{V}_G^{\sigma_i}$  is a filtered volume extracted by convolving on  $\mathcal{V}$  a 3D non-Gaussian-gradient filtering kernel with  $\sigma_i$ . In fact, these zero-order descriptors ZoGF<sup>2D/3D</sup> are respectively crucial extensions in multi-scale filtering analysis for FoSIG [C2] and V-BIG [C5]. It should be noted that the DoG (difference of Gaussians) characteristics in FoSIG and V-BIG are not regarded in ZoGF<sup>2D/3D</sup> due to an objective comparison with HoGF<sup>2D/3D</sup> in DT classification performances.

In terms of comparison with the abilities of non-Gaussian-gradient descriptors (i.e., ZoGF<sup>2D/3D</sup>, FoSIG [C2], and V-BIG [C5]), our descriptors HoGF<sup>2D/3D</sup> have the following advantageous properties to enhance the discrimination power.

- The concept of Gaussian-gradient kernels allows HoGF to address more complementary components than the non-Gaussian-gradient descriptors have done. Specifically, while HoGF takes into account three components for HoGF<sup>2D</sup> or four for HoGF<sup>3D</sup> due to Equation (6.49), the non-Gaussian-gradient methods consider at most two complementary components because they have dealt with only Gaussian-based blurred/invariant features.
- Beside informative magnitudes (i.e.,  $\nabla \mathcal{I}_{x^k, y^k}^{\mathcal{F}}$  and  $\nabla \mathcal{V}_{x^k, y^k, z^k}^{\mathcal{F}}$ ) exploited for DT encoding, the proposed HoGF<sup>2D/3D</sup> descriptors can exploit both symmetric and asymmetric features by addressing even and odd orders in the 2D/3D Gaussian-gradient filterings which extract local features in a totally different way (see Fig. 6.26 for particular performances of these features). Then combining those of the even and odd orders allows to point out more complementary filtered features.
- The discrimination power is enhanced thanks to a feature-concatenated operation of coherent Gaussian-gradient patterns which are extracted from complementary filtered elements in  $\Omega_{\mathcal{H}, \mathcal{F}}^{2D/3D}$  (see Figure 6.27 for evaluations).
- Different deviations of 2D/3D Gaussian-gradient filtering kernels are utilized to diversely capture Gaussian-gradient-based features in a multi-scale analysis for DT representation. In the meantime, FoSIG [C2] and V-BIG [C5] are lack of scale-pattern information due to only a single Gaussian-based kernel involved with.

### 6.8.3 Experiments and evaluations

#### 6.8.3.1 Parameters for experimental implementation

**Settings for Gaussian-gradient filterings:** In order to calculate high-order Gaussian-gradient filtered images/volumes (i.e.,  $\Omega_{\mathcal{H}, \mathcal{F}}^{2D/3D}$ ), we empirically investigate a relevant set of standard deviations  $\mathcal{F} = \{0.5, 0.7, 1, 1.5, 2\}$  with spatial and temporal directions  $x, y, z \in [-3\sigma_i, 3\sigma_i]$ , such that  $\sigma_i \in \mathcal{F}$ . In terms of computing partial derivatives subject to the  $x, y, z$  coordinates, we conduct Gaussian-gradients from the 1<sup>st</sup> to 4<sup>th</sup>-order, i.e.,  $\mathcal{H} = \{1^{st}, 2^{nd}, 3^{rd}, 4^{th}\}$ . Accordingly, different coefficients of a 2D/3D Gaussian-gradient kernel are respectively formed for the separable filtering convolutions on along a direction  $t \in \{x, y, z\}$  as follows.

$$\left\{ -\frac{t}{\sigma_i^2}, \frac{t^2 - \sigma_i^2}{\sigma_i^4}, -\frac{t^3 - 3t\sigma_i^2}{\sigma_i^6}, \frac{t^4 - 6t^2\sigma_i^2 + 3\sigma_i^4}{\sigma_i^8} \right\} \quad (6.56)$$

where  $\sigma_i \in \mathcal{F}$ . Figure 6.24 shows an instance of filtered results using different orders of a 3D Gaussian-gradient with  $\sigma = 0.7$ .

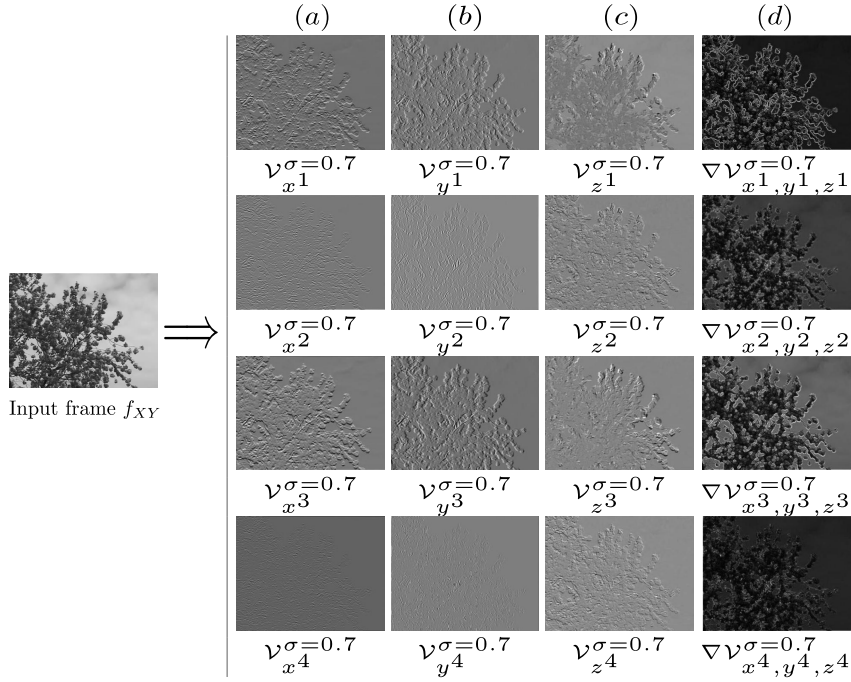


Figure 6.24: An instance of filtering a video  $\mathcal{V}$  using 4 orders (i.e.,  $k = \{1, 2, 3, 4\}$ ) of a 3D Gaussian-gradient kernel with  $\sigma = 0.7$ . Therein, columns (a), (b), and (c) denote Gaussian-gradient filtered outcomes of an input frame  $f_{XY}$ . (d) denotes informative magnitudes of the obtained Gaussian-gradients of  $f_{XY}$ .

**Settings for structuring  $\text{HoGF}^{2D/3D}$  descriptors:** In order to figure out  $\text{HoGF}^{2D/3D}$  patterns based on the filtered components  $\Omega_{\mathcal{H}, \mathcal{F}}^{2D/3D}$ , we exploit the simple and popular operator CLBP using 3D joint settings of *riu2* mapping with  $2(P+2)^2$  bins, i.e.,  $\Psi = \text{CLBP}_{P,R}^{\text{riu2}}$ . Therein, local supporting regions can be addressed as  $\{(P, R)\} = \{(8, 1), (8, 2)\}$  for capturing more forceful information in larger regions, where  $P \in \mathbb{Z}^+$  is a number of considered neighbors involved with. As mentioned in Section 6.8.2, due to three planes  $\{XY, XT, YT\}$  used for each scale of CLBP computation, it takes  $3 \times 2 \times 2 \times (8+2)^2 = 1200$  bins in order to structure a Gaussian-gradient filtered component in  $\Omega_{\mathcal{H}, \mathcal{F}}^{2D/3D}$ . Accordingly, the dimensions of our  $\text{HoGF}^{2D/3D}$  descriptors are then subject to a number of high-order derivatives along with a quantity of Gaussian filtering scales taken into account the DT encoding. That means  $1200 \times m \times |\mathcal{F}| \times |\Omega_{\mathcal{H}, \mathcal{F}}^{2D/3D}|$  bins in general, where  $l = |\mathcal{F}|$  and  $|\Omega_{\mathcal{H}, \mathcal{F}}^{2D/3D}|$  respectively denote the cardinality of standard deviations and of complementary components involved with the filtering. For instance, in respect of one filtered component in  $\Omega_{\mathcal{H}, \mathcal{F}}^{2D/3D}$  that is computed by using the first-order Gaussian-gradient (i.e.,  $m = 1$ ) with a single-scale of Gaussian filtering (i.e.,  $|\mathcal{F}| = 1$ ), the final dimensions are  $1200 \times 3 = 3600$  bins for  $\text{HoGF}^{2D}$  descriptor, and  $1200 \times 4 = 4800$  bins for  $\text{HoGF}^{3D}$  (see Table 6.2 for comparison with other LBP-based descriptors).

**Settings for structuring  $\text{ZoGF}^{2D/3D}$  descriptors:** To be objective in comparison with our  $\text{HoGF}^{2D/3D}$ , parameters for implementing the zero-order  $\text{ZoGF}^{2D/3D}$  descriptors should be in accordance with those of  $\text{HoGF}^{2D/3D}$  for DT representation, i.e.,  $\Psi = \text{CLBP}_{P,R}^{\text{riu2}}$  with local supporting regions  $\{(P, R)\} = \{(8, 1), (8, 2)\}$ , and  $\mathcal{F} = \{0.5, 0.7, 1, 1.5, 2\}$  for the convolved directions  $x, y, z \in [-3\sigma_i, 3\sigma_i]$ . Accordingly, it takes 1200 bins for single-scale  $\text{ZoGF}^{2D/3D}$  to describe dynamic patterns.

### 6.8.3.2 Assessments of High-order Gaussian-gradient Descriptors

We thoroughly discuss the significant effectiveness of the high-order Gaussian-gradient features in comparison with the non-Gaussian-gradient characteristics. Accordingly, the descriptors  $\text{HoGF}^{2D/3D}$

Table 6.20: Classification rates (%) on DT benchmark datasets of HoGF<sup>2D</sup> descriptor.

Dataset		UCLA				DynTex				DynTex++
Order	$\{\sigma_i\}$	50-LOO	50-4fold	9-class	8-class	DynTex35	Alpha	Beta	Gamma	
1 <sup>st</sup>	{0.5}	<b>100</b>	<b>100</b>	99.25	98.91	97.71	96.67	95.68	95.08	96.89
	{0.7}	<b>100</b>	<b>100</b>	98.95	97.61	99.14	98.33	95.06	94.70	96.99
	{1.0}	<b>100</b>	<b>100</b>	99.50	98.70	98.86	98.33	94.44	95.08	96.43
	{0.5, 0.7}	<b>100</b>	<b>100</b>	98.55	98.70	98.86	96.67	95.68	95.45	<b>97.64</b>
	{0.7, 1.0}	<b>100</b>	<b>100</b>	98.10	98.15	98.86	98.33	94.44	95.45	97.01
	{0.5, 1.0}	<b>100</b>	<b>100</b>	99.15	<b>99.13</b>	98.86	96.67	95.68	96.21	97.44
2 <sup>nd</sup>	{0.5}	<b>100</b>	<b>100</b>	99.05	97.39	98.86	<b>100</b>	93.21	92.80	96.37
	{0.7}	99.50	99.50	98.25	97.93	98.57	98.33	<b>95.68</b>	93.56	96.98
	{1.0}	<b>100</b>	<b>100</b>	98.55	97.83	98.86	<b>100</b>	95.06	93.94	96.67
	{0.5, 0.7}	<b>100</b>	<b>100</b>	98.35	97.83	99.14	98.33	<b>95.68</b>	93.56	97.20
	{0.7, 1.0}	<b>100</b>	<b>100</b>	98.75	98.70	99.43	<b>100</b>	95.06	94.70	97.10
	{0.5, 1.0}	<b>100</b>	<b>100</b>	98.55	98.70	<b>100</b>	<b>100</b>	95.06	94.32	97.12
3 <sup>rd</sup>	{0.5}	<b>100</b>	<b>100</b>	99.05	98.80	99.14	<b>100</b>	<b>95.68</b>	<b>96.97</b>	96.71
	{0.7}	<b>100</b>	<b>100</b>	98.40	98.26	98.57	98.33	95.06	95.08	96.57
	{1.0}	<b>100</b>	<b>100</b>	97.65	98.26	99.14	98.33	95.06	93.94	96.56
	{0.5, 0.7}	<b>100</b>	<b>100</b>	98.85	97.72	98.86	98.83	94.44	<b>96.97</b>	97.04
	{0.7, 1.0}	<b>100</b>	<b>100</b>	<b>99.60</b>	97.50	99.14	98.33	<b>95.68</b>	95.08	96.86
	{0.5, 1.0}	<b>100</b>	<b>100</b>	98.90	97.83	99.14	98.33	96.30	95.45	97.06
4 <sup>th</sup>	{0.5}	<b>100</b>	<b>100</b>	98.10	98.04	99.14	<b>100</b>	91.36	93.18	96.71
	{0.7}	99.50	99.50	97.90	98.26	99.14	96.67	93.83	93.18	95.95
	{1.0}	99.50	99.50	97.85	95.87	99.14	96.67	95.68	92.80	95.53
	{0.5, 0.7}	<b>100</b>	<b>100</b>	98.85	97.50	99.43	<b>100</b>	94.44	93.18	97.09
	{0.7, 1.0}	99.50	99.50	98.85	96.85	99.43	96.67	93.83	93.18	96.29
	{0.5, 1.0}	99.50	99.50	97.85	96.41	99.14	<b>100</b>	95.06	93.94	97.18

Note: 50-LOO and 50-4fold denote results on 50-class breakdown using leave-one-out and four cross-fold validation.

and ZoGF<sup>2D/3D</sup> are constructed using the parameters nominated in Section 6.8.3.1. In general, experimental results for DT classification on benchmark datasets have verified that spatio-temporal patterns of HoGF<sup>2D/3D</sup> have much better discrimination power compared to those of ZoGF<sup>2D/3D</sup>, especially, performances on the challenging datasets, i.e., *Beta*, *Gamma*, and DynTex++. It has proved the advantages of Gaussian-gradients in outstandingly resisting the well-known problems of DT encoding: changes of environmental factors, illumination, and noise. As deliberated in Section 6.8.2, it could be asserted the following crucial statements based on the experimental results:

First, the higher value of standard deviation  $\sigma$  is taken into account a Gaussian-gradient filtering, the less discrimination power of our HoGF<sup>2D/3D</sup> descriptors is obtained. Indeed, it can be verified in Figure 6.25 that with an increase of  $\sigma$  from 0.5 to 2, their performance on DynTex++ dataset is decreased about from 1% to 3% in general. This is due to the reductive appearance information caused by the gradient filterings with the large values of  $\sigma$ . Furthermore, Figure 6.25 also indicates the significant and “stable” operation of our HoGF<sup>2D/3D</sup> at levels of  $\sigma \in [0.5, 1]$ . Besides, the reduction is similarly adapted for ZoGF<sup>2D/3D</sup> descriptors (see the 0<sup>th</sup>-order label in Figure 6.25). Therefore, from now on, we mostly report evaluations in consideration of the filterings with standard deviations in that range, i.e.,  $\mathcal{F} = \{0.5, 0.7, 1\}$  (see Tables 6.20, 6.21, 6.22, and 6.23).

Second, as pointed out in Section 6.8.2, taking advantage of Gaussian-gradient features in  $\Omega_{\mathcal{H},\mathcal{F}}^{2D/3D}$  for DT representation, our proposed HoGF<sup>2D/3D</sup> descriptors obtain significant rates in classifying DTs (see Tables 6.20, 6.21, 6.22, and 6.23). This is thanks to the crucial contribution of each component in  $\Omega_{\mathcal{H},\mathcal{F}}^{2D/3D}$  (see Table 6.24). Furthermore, the filterings using 3D Gaussian-gradient kernels allow to enrich more spacial information for voxels of a video  $\mathcal{V}$ , i.e., complementary filtered volumes in  $\Omega_{\mathcal{H},\mathcal{F}}^{3D}$ . Therefore, capturing spatio-temporal patterns based on these volumes have structured HoGF<sup>3D</sup> descriptor with more stable performances in general, contrary to encoding HoGF<sup>2D</sup> from filtered elements in  $\Omega_{\mathcal{H},\mathcal{F}}^{2D}$  computed by convolving 2D filtering kernels on  $\mathcal{V}$ ’s plane-images (see Tables 6.20, 6.21, 6.22, and

Table 6.21: Classification rates (%) on DT benchmark datasets of HoGF<sup>3D</sup> descriptors.

Dataset		UCLA				DynTex				DynTex++
Order	$\{\sigma_i\}$	50-LOO	50-4fold	9-class	8-class	DynTex35	Alpha	Beta	Gamma	
1 <sup>st</sup>	{0.5}	<b>100</b>	<b>100</b>	98.30	98.59	98.57	98.33	96.91	96.21	97.38
	{0.7}	<b>100</b>	<b>100</b>	98.70	99.13	99.14	98.33	96.91	96.21	97.64
	{1.0}	<b>100</b>	<b>100</b>	98.65	98.91	98.86	98.33	96.91	<b>96.97</b>	97.23
	{0.5, 0.7}	<b>100</b>	<b>100</b>	98.90	97.39	98.86	98.33	96.91	96.59	97.63
	{0.7, 1.0}	<b>100</b>	<b>100</b>	98.95	98.04	98.86	98.33	<b>97.53</b>	95.83	97.26
	{0.5, 1.0}	<b>100</b>	<b>100</b>	<b>99.65</b>	97.93	99.43	98.33	96.91	96.59	<b>97.84</b>
2 <sup>nd</sup>	{0.5}	<b>100</b>	<b>100</b>	98.65	98.80	99.43	<b>100</b>	92.59	93.56	96.44
	{0.7}	<b>100</b>	<b>100</b>	99.30	<b>99.46</b>	98.29	<b>100</b>	96.30	94.70	97.32
	{1.0}	<b>100</b>	<b>100</b>	99.00	99.02	98.86	<b>100</b>	95.06	95.08	97.13
	{0.5, 0.7}	<b>100</b>	<b>100</b>	99.45	99.24	99.71	<b>100</b>	96.91	93.94	97.79
	{0.7, 1.0}	<b>100</b>	<b>100</b>	98.45	98.80	99.14	<b>100</b>	96.30	94.70	97.26
	{0.5, 1.0}	<b>100</b>	<b>100</b>	98.75	98.48	<b>100</b>	<b>100</b>	96.30	95.08	97.64
3 <sup>rd</sup>	{0.5}	<b>100</b>	<b>100</b>	99.40	98.70	99.14	98.83	96.30	96.21	97.18
	{0.7}	<b>100</b>	<b>100</b>	98.55	97.83	98.57	98.83	96.30	95.08	97.02
	{1.0}	<b>100</b>	<b>100</b>	98.85	98.59	99.14	98.83	<b>97.53</b>	95.45	96.96
	{0.5, 0.7}	<b>100</b>	<b>100</b>	98.70	97.61	98.86	98.83	96.91	96.21	97.30
	{0.7, 1.0}	<b>100</b>	<b>100</b>	98.05	98.04	99.14	98.83	<b>97.53</b>	95.83	97.34
	{0.5, 1.0}	<b>100</b>	<b>100</b>	99.60	97.83	99.14	98.83	96.30	96.21	97.34
4 <sup>th</sup>	{0.5}	<b>100</b>	<b>100</b>	98.55	97.61	99.14	<b>100</b>	92.59	93.94	96.84
	{0.7}	99.50	<b>100</b>	98.95	97.93	98.86	96.67	95.06	93.18	96.23
	{1.0}	99.50	99.50	98.20	97.50	99.14	98.33	96.30	93.56	96.07
	{0.5, 0.7}	<b>100</b>	<b>100</b>	99.65	96.74	99.43	<b>100</b>	95.68	93.94	97.62
	{0.7, 1.0}	99.50	<b>100</b>	98.90	97.07	99.43	96.67	96.30	94.70	96.89
	{0.5, 1.0}	<b>100</b>	<b>100</b>	99.60	96.20	99.43	<b>100</b>	95.69	93.94	97.34

Note: 50-LOO and 50-4fold denote results on 50-class breakdown using leave-one-out and four cross-fold validation.

 Table 6.22: Classification rates (%) on DT benchmark datasets of multi-order HoGF<sup>2D</sup> descriptor.

Dataset		UCLA				DynTex				DynTex++
Order	$\{\sigma_i\}$	50-LOO	50-4fold	9-class	8-class	DynTex35	Alpha	Beta	Gamma	
{1 <sup>st</sup> , 2 <sup>nd</sup> }	{0.5}	<b>100</b>	<b>100</b>	98.30	96.41	98.86	96.67	96.91	93.94	97.47
	{0.7}	<b>100</b>	99.50	98.00	95.76	99.43	<b>100</b>	95.06	95.83	97.43
	{1.0}	<b>100</b>	<b>100</b>	99.30	99.02	99.71	<b>100</b>	96.91	95.08	97.39
{1 <sup>st</sup> , 3 <sup>rd</sup> }	{0.5}	<b>100</b>	<b>100</b>	99.15	97.39	99.14	98.33	96.30	96.21	97.38
	{0.7}	<b>100</b>	<b>100</b>	99.00	96.96	99.14	98.33	95.06	94.32	97.37
	{1.0}	<b>100</b>	<b>100</b>	<b>99.45</b>	96.09	99.14	<b>100</b>	95.06	95.08	97.22
{1 <sup>st</sup> , 4 <sup>th</sup> }	{0.5}	<b>100</b>	<b>100</b>	98.35	97.39	98.57	<b>100</b>	95.68	93.94	<b>97.58</b>
	{0.7}	<b>100</b>	<b>100</b>	98.75	98.04	98.86	96.67	94.44	95.08	97.39
	{1.0}	99.50	99.50	98.40	96.41	98.86	98.33	<b>97.53</b>	95.83	97.01
{2 <sup>nd</sup> , 3 <sup>rd</sup> }	{0.5}	<b>100</b>	<b>100</b>	98.90	96.96	<b>100</b>	<b>100</b>	96.30	94.70	97.37
	{0.7}	<b>100</b>	<b>100</b>	98.45	96.85	99.14	<b>100</b>	95.68	95.45	97.31
	{1.0}	<b>100</b>	<b>100</b>	99.20	98.91	99.71	<b>100</b>	<b>97.53</b>	96.59	97.19
{2 <sup>nd</sup> , 4 <sup>th</sup> }	{0.5}	<b>100</b>	<b>100</b>	97.15	97.83	99.43	<b>100</b>	92.59	92.80	96.99
	{0.7}	<b>100</b>	<b>100</b>	98.75	<b>99.13</b>	98.86	98.33	93.83	93.18	96.77
	{1.0}	99.50	99.50	98.70	95.89	99.71	98.33	95.06	93.94	96.71
{3 <sup>rd</sup> , 4 <sup>th</sup> }	{0.5}	<b>100</b>	<b>100</b>	99.15	96.52	99.71	<b>100</b>	96.30	95.08	97.24
	{0.7}	<b>100</b>	<b>100</b>	98.65	96.96	99.43	96.67	95.68	95.08	97.34
	{1.0}	99.50	99.50	98.10	96.96	99.43	98.33	<b>97.53</b>	<b>97.35</b>	97.07

Note: 50-LOO and 50-4fold denote results on 50-class breakdown using leave-one-out and four cross-fold validation.

6.23). This is also in accordance with the statements asserted in the prior work [C5], in which V-BIG [C5], exploiting 3D Gaussian-based kernels, outperforms FoSIG [C2] with the 2D kernels involved in classifying DTs on the challenging dataset, i.e., DynTex and DynTex++ (see Section 6.8.3.3 for further evaluations).



Table 6.23: Classification rates (%) on DT benchmark datasets of multi-order HoGF<sup>3D</sup> descriptor.

Dataset		UCLA				DynTex				DynTex++
Order	$\{\sigma_i\}$	50-LOO	50-4fold	9-class	8-class	DynTex35	Alpha	Beta	Gamma	
$\{1^{st}, 2^{nd}\}$	$\{0.5\}$	<b>100</b>	<b>100</b>	99.15	99.13	99.14	98.33	96.91	95.08	<b>97.74</b>
	$\{0.7\}$	<b>100</b>	<b>100</b>	97.20	98.70	99.14	<b>100</b>	96.91	96.21	97.71
	$\{1.0\}$	<b>100</b>	<b>100</b>	<b>99.60</b>	97.50	99.71	<b>100</b>	96.91	96.59	97.34
$\{1^{st}, 3^{rd}\}$	$\{0.5\}$	<b>100</b>	<b>100</b>	97.80	98.70	99.14	98.33	96.30	96.59	97.67
	$\{0.7\}$	<b>100</b>	<b>100</b>	98.60	99.35	98.57	98.33	96.91	96.21	97.57
	$\{1.0\}$	<b>100</b>	<b>100</b>	99.40	99.13	99.14	98.33	96.30	97.53	96.98
$\{1^{st}, 4^{th}\}$	$\{0.5\}$	<b>100</b>	<b>100</b>	98.70	98.26	98.86	<b>100</b>	96.91	95.45	97.39
	$\{0.7\}$	<b>100</b>	<b>100</b>	98.70	97.50	99.43	98.33	96.30	94.70	<b>97.74</b>
	$\{1.0\}$	<b>100</b>	<b>100</b>	99.35	97.83	98.86	98.33	<b>98.15</b>	<b>97.73</b>	97.28
$\{2^{nd}, 3^{rd}\}$	$\{0.5\}$	<b>100</b>	<b>100</b>	98.30	97.07	99.71	98.33	95.06	93.94	97.40
	$\{0.7\}$	<b>100</b>	<b>100</b>	99.05	99.02	98.86	98.33	96.91	96.21	97.73
	$\{1.0\}$	<b>100</b>	<b>100</b>	99.55	99.02	99.71	<b>100</b>	97.53	96.59	97.39
$\{2^{nd}, 4^{th}\}$	$\{0.5\}$	<b>100</b>	<b>100</b>	99.00	96.52	99.71	<b>100</b>	92.59	93.18	96.88
	$\{0.7\}$	<b>100</b>	<b>100</b>	99.05	98.37	99.43	98.33	95.06	93.18	97.08
	$\{1.0\}$	<b>100</b>	<b>100</b>	98.80	96.85	99.71	98.33	96.91	95.83	97.28
$\{3^{rd}, 4^{th}\}$	$\{0.5\}$	<b>100</b>	<b>100</b>	98.25	97.83	99.71	<b>100</b>	95.68	94.32	97.31
	$\{0.7\}$	<b>100</b>	<b>100</b>	99.50	97.83	99.71	98.33	96.91	95.08	97.65
	$\{1.0\}$	<b>100</b>	<b>100</b>	99.25	<b>99.57</b>	99.43	98.33	<b>98.15</b>	97.53	97.63

Note: 50-LOO and 50-4fold denote results on 50-class breakdown using leave-one-out and four cross-fold validation.

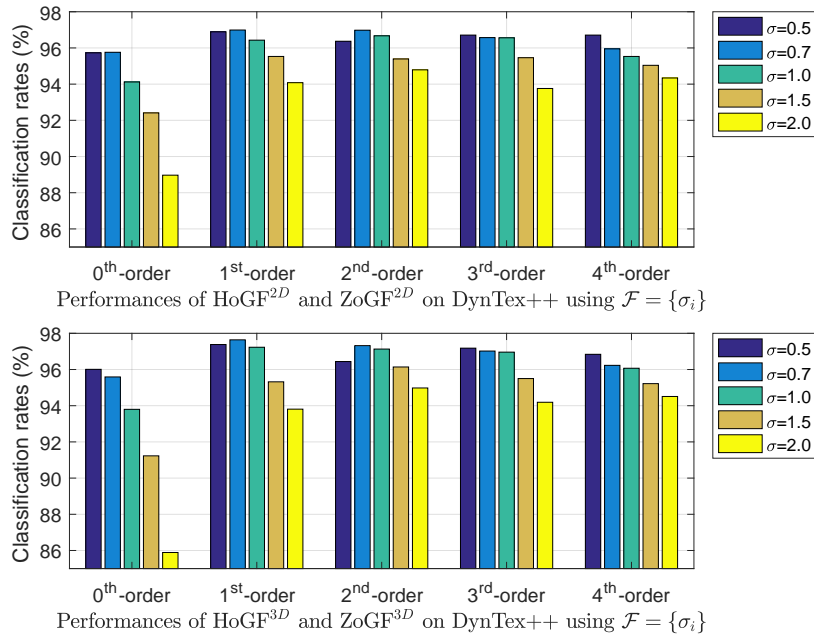


Figure 6.25: A sharply decrease of performances of high-order HoGF<sup>2D/3D</sup> and zero-order ZoGF<sup>2D/3D</sup> on DynTex++ when increasing  $\sigma$  from 0.5 to 2 for the Gaussian-gradient filterings.

Third, DT asymmetric gradient features extracted from the odd derivative functions mostly have more forceful robustness in enhancing the performance compared to symmetric patterns derived from the even derivations. It could be thanks to the homogeneous distributions of filtered results. Indeed, experimental results have verified that in general, HoGF<sup>2D/3D</sup> using the 1<sup>st</sup> and 3<sup>rd</sup> partial derivatives obtain better and “stable” performances compared to those exploiting the 2<sup>nd</sup> and 4<sup>th</sup> ones, particularly, the performances on the challenging schemes, i.e., *Beta* and *Gamma* (see Tables 6.20 and 6.21). This assessment is also agreed with the abilities of the zero-order descriptor ZoGF<sup>2D/3D</sup> since they are also based on an even derivative. Also, it should be noted that the 2<sup>nd</sup> one points out significant rates on UCLA as well as on *DynTex35* and *Alpha* schemes but not “stably” on *Beta*, *Gamma*, and DynTex++. Moreover, Figure

Table 6.24: Contributions of the first-order components in  $\Omega_{\{1^{st}\},\{0.7\}}^{2D/3D}$ .

	UCLA				DynTex				
$\Omega_{\{1^{st}\},\{0.7\}}^{2D/3D}$	50-LOO	50-4fold	9-class	8-class	Dyn35	Alpha	Beta	Gamma	Dyn++
$\mathcal{I}_{x^1}^{\sigma=0.7}$	<b>100</b>	<b>100</b>	<b>99.20</b>	96.20	98.00	98.33	93.21	91.29	95.64
$\mathcal{I}_{y^1}^{\sigma=0.7}$	<b>100</b>	<b>100</b>	98.10	97.83	97.14	<b>100</b>	93.21	94.32	95.74
$\nabla \mathcal{I}_{x^1,y^1}^{\sigma=0.7}$	<b>100</b>	<b>100</b>	97.75	97.50	96.29	95.00	91.98	92.42	95.22
$\Omega_{\{1^{st}\},\{0.7\}}^{2D}$	<b>100</b>	<b>100</b>	98.95	97.61	<b>99.14</b>	98.33	95.06	94.70	96.99
$\mathcal{V}_{x^1}^{\sigma=0.7}$	<b>100</b>	<b>100</b>	98.25	97.83	96.29	96.67	93.83	93.94	94.55
$\mathcal{V}_{y^1}^{\sigma=0.7}$	99.50	99.50	98.50	98.80	98.86	96.67	92.59	92.42	94.02
$\mathcal{V}_{z^1}^{\sigma=0.7}$	97.50	97.00	98.85	98.70	97.71	98.33	95.06	91.29	95.29
$\nabla \mathcal{V}_{x^1,y^1,z^1}^{\sigma=0.7}$	<b>100</b>	<b>100</b>	98.95	97.61	96.57	95.00	93.83	92.05	95.13
$\Omega_{\{1^{st}\},\{0.7\}}^{3D}$	<b>100</b>	<b>100</b>	98.70	<b>99.13</b>	<b>99.14</b>	98.33	<b>96.91</b>	<b>96.21</b>	<b>97.64</b>

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

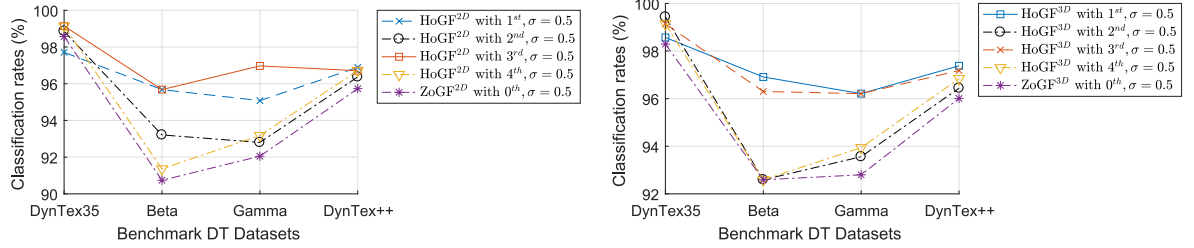


Figure 6.26: Outperformances of  $\text{HoGF}^{2D/3D}$  descriptors using asymmetric gradient features compared to those using symmetric features.

6.26 shows the outstanding performances of the odd  $\text{HoGF}^{2D/3D}$  descriptors compared to the even ones, especially, from 2% to 4% rates of improvement on challenging datasets, i.e., *Beta* and *Gamma*.

Fourth, it can be visually seen from Figure 6.24 that the higher orders of derivatives are taken into account, the larger informative appearances have been lost. It is due to the increase of amplitude frequencies in the higher orders of differentiation. This negatively affects the understanding of spatial features in DT encoding. Indeed, our experimental results have agreed with that (see Table 6.21). For instance, using  $\sigma = 1$  for the filtering, classification rate of the 1<sup>st</sup>-order  $\text{HoGF}^{3D}$  descriptor has been reduced from 96.97% to 95.45% on *Gamma++*, while from 95.08% to 93.56% for the 2<sup>nd</sup>-order one.

Fifth, based on the experimental results, it can be generally validated that the multi-scale analysis of different standard deviations  $\mathcal{F} = \{\sigma_i\}$  allows to adequately capture scale-informative features in order to enhance the discrimination power for DT representation (see Tables 6.20 and 6.21). In spite of that, the enhancements are just at modest levels and not stable as well, while the dimension of  $\text{HoGF}^{2D/3D}$  increases by 2 times. For instance, the first-order descriptor  $\text{HoGF}_{1^{st}}^{2D}$  obtains 95.08% and 94.70% using single-scale  $\sigma = 0.5$  and  $\sigma = 0.7$  on *Gamma* respectively. In the meanwhile, a little better rate of 95.45% is achieved by integrating these scales, i.e.,  $\{0.5, 0.7\}$ .

Finally, the spatio-temporal characteristics in  $\text{HoGF}^{2D/3D}$  could be enhanced in more robustness and operative stability thanks to multi-order of derivatives taken into account the DT encoding, as asserted in Section 6.8.1. In fact, Tables 6.22 and 6.23 shows the better performances of our  $\text{HoGF}^{2D/3D}$  descriptors when exploiting multi-order Gaussian-gradient features. Among of them, the discriminative power of  $\text{HoGF}^{2D}$  is improved significantly (see Tables 6.20, 6.21, 6.22, and 6.23).

In short, based on above thorough assessments, we point out in Table 6.25 several advantageous settings of our proposed  $\text{HoGF}^{2D/3D}$  descriptors that are also recommended for comparing to state of the art, as well as implementing applications in practice. Accordingly, it can be considered for either using the single-order  $\text{HoGF}^{2D/3D}$  descriptors to meet demands of lower dimension, or the 2-order ones for strict requirements of high precision on challenging datasets. Hereunder, we assess in detail

Table 6.25: Settings for comparison and real implementations.

Setting	$\{\sigma_i\}$	Single-order	Multi-order	#Dimension	Reference
HoGF <sup>2D</sup>	{0.5}	3 <sup>rd</sup> -order	-	3600 bins	Table 6.20
HoGF <sup>2D</sup>	{1.0}	-	{1 <sup>st</sup> , 2 <sup>nd</sup> }	7200 bins	Table 6.22
HoGF <sup>2D</sup>	{1.0}	-	{2 <sup>nd</sup> , 3 <sup>rd</sup> }	7200 bins	Table 6.22
HoGF <sup>3D</sup>	{0.5}	3 <sup>rd</sup> -order	-	4800 bins	Table 6.21
HoGF <sup>3D</sup>	{1.0}	-	{3 <sup>rd</sup> , 4 <sup>th</sup> }	9600 bins	Table 6.23
HoGF <sup>3D</sup>	{1.0}	-	{2 <sup>nd</sup> , 3 <sup>rd</sup> }	9600 bins	Table 6.23

Note: “-” means “not available”.

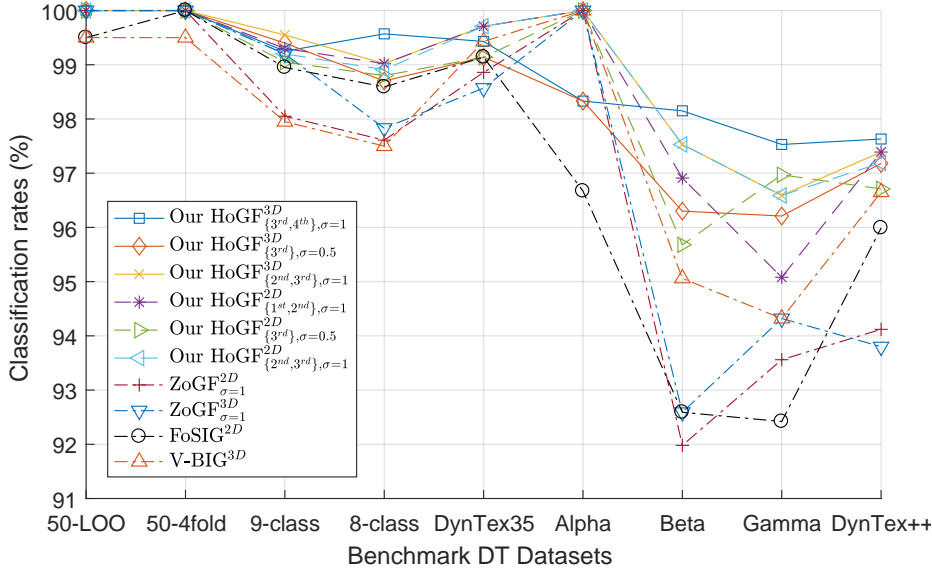


Figure 6.27: Performances of our high-order HoGF<sup>2D/3D</sup> descriptors in comparison with those of non-Gaussian-gradient descriptors, i.e., ZoGF<sup>2D/3D</sup>, FoSIG [C2], and V-BIG [C5].

the outstanding performances of HoGF<sup>2D/3D</sup> compared to ZoGF<sup>2D/3D</sup>, descriptors without Gaussian-gradient features involved in. After that, particular evaluations classifying DTs on each dataset are discussed in a global comparison with the existing methods.

### 6.8.3.3 Comprehensive Comparison to Non-Gaussian-gradients

As mentioned in Section 6.8.2, our HoGF<sup>2D/3D</sup> descriptors have prominent abilities in classifying DT videos compared to the zero-order ZoGF<sup>2D/3D</sup> and other Gaussian-based DT descriptors, i.e., V-BIG [C5] and FoSIG [C2] (see Figure 6.27). It is thanks to capturing spatio-temporal features in the high-order Gaussian-gradient components  $\Omega_{\mathcal{H}, \mathcal{F}}^{2D/3D}$  instead of those without derivatives taken into account. Indeed, it can be observed from Tables 6.20, 6.21, 6.26, and 6.27 that using the same settings of standard deviations  $\mathcal{F}$  and supporting regions  $\{(P, R)\}$  (see Section 6.8.3.1), the single-order HoGF<sup>2D/3D</sup> descriptors obtain significantly higher rates compared to ZoGF<sup>2D/3D</sup> in both single-scale and multi-scale configurations on challenging datasets, i.e., *Beta*, *Gamma*, and *DynTex++* (see Figure 6.26).

## 6.9 Representation based on DoDG-filtered features

### 6.9.1 Construction of DoDG-filtered descriptors

Our proposal is graphically illustrated as Figure 6.28. In general, it takes two major steps to structure a given video  $\mathcal{V}$ : i) a novel filtering for an efficient reduction of the negative impacts of the problems on DT representation; ii) a local DT encoding of the obtained filtered-outcomes in simplicity of computa-

Table 6.26: Rates (%) on DT benchmark datasets of ZoGF<sup>2D</sup> descriptor.

Dataset	UCLA				DynTex				DynTex++
	50-LOO	50-4fold	9-class	8-class	DynTex35	Alpha	Beta	Gamma	
$\{\sigma_i\}$									
$\{0.5\}$	<b>100</b>	<b>100</b>	<b>98.85</b>	98.15	98.57	<b>100</b>	90.74	92.05	95.73
$\{0.7\}$	<b>100</b>	<b>100</b>	98.05	97.17	98.57	<b>100</b>	<b>91.98</b>	93.18	95.76
$\{1.0\}$	<b>100</b>	<b>100</b>	98.05	97.61	98.86	<b>100</b>	<b>91.98</b>	<b>93.56</b>	94.12
$\{0.5, 0.7\}$	<b>100</b>	<b>100</b>	98.20	<b>98.80</b>	98.86	<b>100</b>	91.36	92.05	<b>96.16</b>
$\{0.7, 1.0\}$	<b>100</b>	<b>100</b>	98.65	96.41	98.57	<b>100</b>	<b>91.98</b>	92.05	95.72
$\{0.5, 1.0\}$	<b>100</b>	<b>100</b>	97.65	97.17	<b>99.14</b>	<b>100</b>	90.74	91.67	95.90

Note: 50-LOO and 50-4fold denote results on 50-class using leave-one-out and four cross-fold validation.

 Table 6.27: Rates (%) on DT benchmark datasets of ZoGF<sup>3D</sup> descriptor.

Dataset	UCLA				DynTex				DynTex++
	50-LOO	50-4fold	9-class	8-class	DynTex35	Alpha	Beta	Gamma	
$\{\sigma_i\}$									
$\{0.5\}$	<b>100</b>	<b>100</b>	99.10	97.93	98.29	<b>100</b>	92.59	92.80	96.01
$\{0.7\}$	<b>100</b>	<b>100</b>	98.75	97.61	98.29	<b>100</b>	92.59	93.56	95.59
$\{1.0\}$	<b>100</b>	<b>100</b>	<b>99.20</b>	97.83	98.57	<b>100</b>	92.59	<b>94.32</b>	93.80
$\{0.5, 0.7\}$	<b>100</b>	<b>100</b>	99.15	98.70	<b>98.86</b>	<b>100</b>	92.59	92.80	<b>96.08</b>
$\{0.7, 1.0\}$	<b>100</b>	<b>100</b>	98.50	97.39	98.29	<b>100</b>	<b>93.21</b>	93.56	94.45
$\{0.5, 1.0\}$	<b>100</b>	<b>100</b>	97.03	<b>99.23</b>	98.29	<b>100</b>	91.98	93.18	95.83

Note: 50-LOO and 50-4fold denote results on 50-class using leave-one-out and four cross-fold validation.

tion. For the filtering, we have introduced a novel DoDG kernel based on the difference of high-order Gaussian-gradients (see Section 6.3.1). This allows to point out DoDG-filtered outcomes that effectively deal with the well-known above issues thanks to robustness of invariant Gaussian-gradient-filtered features compared to the non-Gaussian-gradient-filtered ones of the conventional DoG, which is exploited in FoSIG [C2] and V-BIG [C5] but its ability has been just at a moderate level due to a lack of complementary filtered components involved in the DT encoding, i.e., only one DoG-filtered outcome (see Figure 6.4 line (a)) obtained by a DoG filtering operation with each pre-defined pair of standard deviations. Section 6.10.1 gives more thorough discussion of this significant point. For the local DT encoding, we investigate the effectiveness of the DoDG kernel in 2D and 3D dimensions for the pre-processing step. CLBP [3], a simple operator, can be then addressed for capturing local DoDG features of the obtained 2D/3D DoDG-filtered outcomes. As a result, DoDGF<sup>2D/3D</sup> descriptors with very good performances on DT recognition compared to recent methods are constructed.

In order to verify the ability of DoDG in dealing with the negative influences on DT representation, we take its 2D and 3D variations into account the pre-processing step of encoding a given video  $\mathcal{V}$  for noise-resistance. The DoDG-filtered outputs are then encoded by CLBP [3], a simple operator, in order

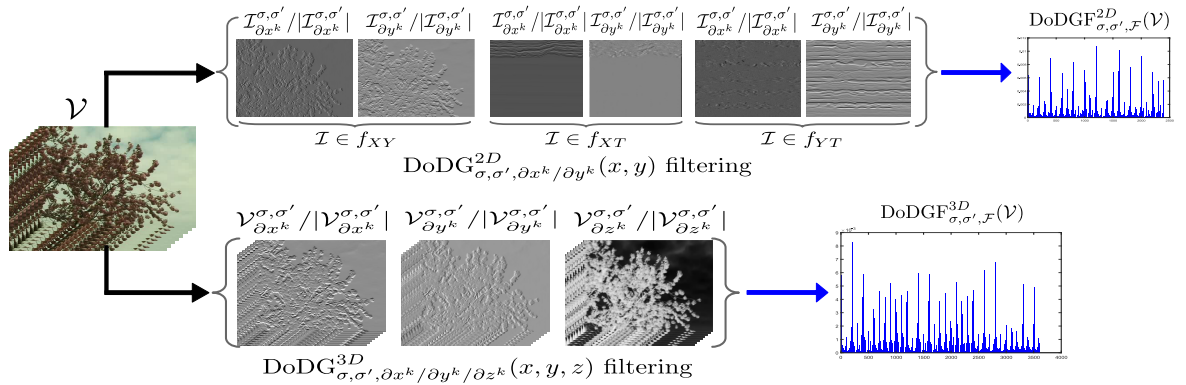


Figure 6.28: Our proposed framework for encoding a video  $\mathcal{V}$  based on its DoDG-filtered outcomes computed by the novel DoDG filtering kernels. Therein, the black arrows denote pre-processing steps, while the blue ones are for processes of DT encoding.

to correspondingly form robust local DT descriptors. Hereafter, we express these processes in detail.

**Proposed DoDGF $_{\sigma,\sigma',\mathcal{F}}^{2D}$  descriptor:** To be compliant with the DoDG $^{2D}$  filtering, the video  $\mathcal{V}$  is decomposed subject to its orthogonal planes to obtain separate collections of plane-images  $f_{XY}$ ,  $f_{XT}$ , and  $f_{YT}$ . With respect to each image  $\mathcal{I} \in f_{XY}$ , a  $k$ -order DoDG $^{2D}$  kernel is convolved on it to point out DoDG-filtered images as

$$\begin{cases} \mathcal{I}_{\partial x^k}^{\sigma,\sigma'} = \text{DoDG}_{\sigma,\sigma',\partial x^k}^{2D}(x, y) * \mathcal{I} \\ \mathcal{I}_{\partial y^k}^{\sigma,\sigma'} = \text{DoDG}_{\sigma,\sigma',\partial y^k}^{2D}(x, y) * \mathcal{I} \end{cases} \quad (6.57)$$

in which “\*” stands for a convolving operator;  $x, y$  are spatial coordinates. Samples of this filtering can be seen in Figure 6.4: line (b) for the odd gradients and line (c) for the even ones. Since  $\mathcal{I}_{\partial x^k}^{\sigma,\sigma'}$  and  $\mathcal{I}_{\partial y^k}^{\sigma,\sigma'}$  are bipolar-filtered images, it could be possible to consider their absolute outcomes (i.e.,  $|\mathcal{I}_{\partial x^k}^{\sigma,\sigma'}|$  and  $|\mathcal{I}_{\partial y^k}^{\sigma,\sigma'}|$ ) to explore more textural appearances for further improving discrimination (see Table 6.30 for their contributions). As a result, all plane-images  $\mathcal{I} \in f_{XY}$  are encoded as

$$\Gamma_{\sigma,\sigma',k}^{XY} = \frac{1}{\mathcal{N}} \sum_{\mathcal{I} \in f_{XY}} \left[ \Psi(\mathcal{I}_{\partial x^k}^{\sigma,\sigma'}), \Psi(|\mathcal{I}_{\partial x^k}^{\sigma,\sigma'}|), \Psi(\mathcal{I}_{\partial y^k}^{\sigma,\sigma'}), \Psi(|\mathcal{I}_{\partial y^k}^{\sigma,\sigma'}|) \right] \quad (6.58)$$

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

$$\text{DoDGF}_{\sigma,\sigma',\mathcal{F}}^{2D}(\mathcal{V}) = \biguplus_{k \in \mathcal{F}} \left[ \Gamma_{\sigma,\sigma',k}^{XY}, \Gamma_{\sigma,\sigma',k}^{XT}, \Gamma_{\sigma,\sigma',k}^{YT} \right] \quad (6.59)$$

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

**Proposed DoDGF $_{\sigma,\sigma',\mathcal{F}}^{3D}$  descriptor:** The DoDG $^{3D}$  filtering is used for pre-processing video  $\mathcal{V}$  as

$$\begin{cases} \mathcal{V}_{\partial x^k}^{\sigma,\sigma'} = \text{DoDG}_{\sigma,\sigma',\partial x^k}^{3D}(x, y, z) * \mathcal{V} \\ \mathcal{V}_{\partial y^k}^{\sigma,\sigma'} = \text{DoDG}_{\sigma,\sigma',\partial y^k}^{3D}(x, y, z) * \mathcal{V} \\ \mathcal{V}_{\partial z^k}^{\sigma,\sigma'} = \text{DoDG}_{\sigma,\sigma',\partial z^k}^{3D}(x, y, z) * \mathcal{V} \end{cases} \quad (6.60)$$

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

$$\Upsilon(\mathcal{V}_{\partial x^k}^{\sigma,\sigma'}) = \left[ \Psi(\mathcal{I} \in f'_{XY}), \Psi(\mathcal{I} \in f'_{XT}), \Psi(\mathcal{I} \in f'_{YT}) \right] \quad (6.61)$$

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

$$\text{DoDGF}_{\sigma,\sigma',\mathcal{F}}^{3D}(\mathcal{V}) = \biguplus_{k \in \mathcal{F}} \left[ \Upsilon(\mathcal{V}_{\partial x^k}^{\sigma,\sigma'}), \Upsilon(\mathcal{V}_{\partial y^k}^{\sigma,\sigma'}), \Upsilon(\mathcal{V}_{\partial z^k}^{\sigma,\sigma'}), \Upsilon(|\mathcal{V}_{\partial x^k}^{\sigma,\sigma'}|), \Upsilon(|\mathcal{V}_{\partial y^k}^{\sigma,\sigma'}|), \Upsilon(|\mathcal{V}_{\partial z^k}^{\sigma,\sigma'}|) \right] \quad (6.62)$$

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

**DoG-based descriptors for assessment:** In order to verify the interest of our novel DoDG kernels in local DT understanding compared to the well-known DoG kernel, we also implement local DoG-based descriptors based on the corresponding DoG filterings for comprehensive evaluations in Sections 6.9.2.3 and 6.10.1. Accordingly, the 2D and 3D DoG kernels are addressed for the filtering of video  $\mathcal{V}$  as follows.

$$\mathcal{I}_{DoG}^{\sigma, \sigma'} = \text{DoG}_{\sigma, \sigma'}^{2D}(x, y) * \mathcal{I} \quad , \quad \mathcal{V}_{DoG}^{\sigma, \sigma'} = \text{DoG}_{\sigma, \sigma'}^{3D}(x, y, z) * \mathcal{V} \quad (6.63)$$

Following the construction of the DoDGF<sup>2D</sup> descriptor, the 2D DoG-filtered features (DoGF<sup>2D</sup> <sub>$\sigma, \sigma'$</sub> ) are structured as

$$\text{DoGF}_{\sigma, \sigma'}^{2D}(\mathcal{V}) = \left[ \Lambda_{\sigma, \sigma'}^{XY}, \Lambda_{\sigma, \sigma'}^{XT}, \Lambda_{\sigma, \sigma'}^{YT} \right] \quad (6.64)$$

in which  $\Lambda_{\sigma, \sigma'}^{XY}$ ,  $\Lambda_{\sigma, \sigma'}^{XT}$ ,  $\Lambda_{\sigma, \sigma'}^{YT}$  are similarly defined as Equation (6.58), but for structuring DoG-filtered plane-images instead of addressing the DoDG-filtered ones. For instance of encoding the collection  $f_{XY}$  of raw plane-images,  $\Lambda_{\sigma, \sigma'}^{XY}$  is formed as

$$\Lambda_{\sigma, \sigma'}^{XY} = \frac{1}{N} \sum_{\mathcal{I} \in f_{XY}} \left[ \Psi(\mathcal{I}_{DoG}^{\sigma, \sigma'}), \Psi(|\mathcal{I}_{DoG}^{\sigma, \sigma'}|) \right] \quad (6.65)$$

Also based on the construction of DoDGF<sup>3D</sup>, the 3D DoG-filtered features (DoGF<sup>3D</sup> <sub>$\sigma, \sigma'$</sub> ) are formed as

$$\text{DoGF}_{\sigma, \sigma'}^{3D}(\mathcal{V}) = \left[ \Upsilon(\mathcal{V}_{DoG}^{\sigma, \sigma'}), \Upsilon(|\mathcal{V}_{DoG}^{\sigma, \sigma'}|) \right] \quad (6.66)$$

It should be noted that the 2D/3D DoG filterings were exploited in the prior works (i.e., FoSIG [C2], V-BIG [C5], RUBIG [J4]), but for capturing the absolute-filtered features. In the meanwhile, the DoGF<sup>2D/3D</sup> <sub>$\sigma, \sigma'$</sub>  descriptors are here proposed to capture more the bipolar-filtered ones of those filterings due to an objective comparison to DoDGF<sup>2D/3D</sup> <sub>$\sigma, \sigma', \mathcal{F}$</sub>  in abilities of DT classification.

Consequently, according to all of above those together with a comprehensive evaluation presented at Section 6.9.2.3, it can be stated that our DoDG-based descriptors have several following advantages to enhance the performance in comparison with other local Gaussian-based ones:

- Our DoDGF<sup>2D/3D</sup> <sub>$\sigma, \sigma', \mathcal{F}$</sub>  descriptors are enriched more spatio-temporal features extracted from both bipolar and absolute DoDG-filtered outcomes instead of only from the absolute DoG-filtered ones in FoSIG [C2], V-BIG [C5], and RUBIG [J4] (see Table 6.30 for evaluations of their contributions).
- It could take advantage of more complementary features by addressing DoDG in high-order gradients. This allows DoDGF<sup>2D/3D</sup> <sub>$\sigma, \sigma', \mathcal{F}$</sub>  to capture more scale-filtered information to enhance the performance (see Tables 6.28 and 6.29).
- Addressing the DoDG<sup>2D/3D</sup> kernels could produce more DoDG-filtered outcomes which are complementary for the local DT encoding due to Equations (6.57) and (6.60). In the meanwhile, only one done by the DoG<sup>2D/3D</sup> filterings is exploited in FoSIG [C2], V-BIG [C5], RUBIG [J4], and DoGF<sup>2D/3D</sup> <sub>$\sigma, \sigma'$</sub>  due to Equation (6.63).

## 6.9.2 Experiments and evaluations

### 6.9.2.1 Parameters for experimental implementation

**For DoDG filtering processes:** In experiments of this work, we conduct the proposed DoDG<sup>2D/3D</sup> <sub>$\sigma, \sigma', \partial x_i^k$</sub>  kernels in their four orders (i.e.,  $\{1^{st}, 2^{nd}, 3^{rd}, 4^{th}\}$ ) of partial derivatives with direction axes for the

Table 6.28: Rates (%) of DoDGF $^{2D}_{\sigma,\sigma',\mathcal{F}}$  and DoGF $^{2D}_{\sigma,\sigma'}$  descriptors on benchmark datasets.

DoGF/DoDGF		UCLA				DynTex				DynTex++
Order(s)	$(\sigma, \sigma')$	50-LOO	50-4fold	9-class	8-class	DynTex35	Alpha	Beta	Gamma	
$0^{th}$	(0.5, 0.7)	<b>100</b>	<b>100</b>	98.10	96.41	97.17	95.00	93.83	93.18	94.94
	(0.5, 1)	<b>100</b>	<b>100</b>	98.25	96.74	97.71	98.33	92.59	91.29	94.62
	(0.7, 1)	<b>100</b>	<b>100</b>	97.70	96.20	98.00	<b>100</b>	91.98	92.42	94.86
	(1, 1.3)	<b>100</b>	<b>100</b>	97.15	97.83	97.71	98.33	91.98	93.18	94.08
	(1, 1.5)	<b>100</b>	<b>100</b>	97.50	97.83	98.29	98.33	92.59	90.53	94.18
$1^{st}$	(0.5, 0.7)	99.50	99.50	98.05	98.70	97.71	98.33	95.06	94.32	97.03
	(0.5, 1)	<b>100</b>	<b>100</b>	98.90	96.52	99.14	98.33	95.68	94.70	97.08
	(0.7, 1)	<b>100</b>	<b>100</b>	99.05	98.04	99.43	<b>100</b>	95.68	95.08	96.40
	(1, 1.3)	<b>100</b>	<b>100</b>	98.50	96.09	99.43	<b>100</b>	95.68	94.32	96.51
	(1, 1.5)	<b>100</b>	<b>100</b>	98.70	96.96	99.43	98.33	95.68	94.32	96.21
$2^{nd}$	(0.5, 0.7)	99.00	99.00	98.90	98.15	98.00	<b>100</b>	95.68	93.56	95.86
	(0.5, 1)	99.50	99.50	99.15	96.96	97.71	<b>100</b>	95.06	93.56	96.11
	(0.7, 1)	<b>100</b>	<b>100</b>	99.00	97.61	98.57	<b>100</b>	95.06	93.94	95.54
	(1, 1.3)	<b>100</b>	<b>100</b>	99.30	98.70	98.00	<b>100</b>	94.44	92.80	96.09
	(1, 1.5)	<b>100</b>	<b>100</b>	98.75	98.37	98.86	<b>100</b>	93.83	93.18	95.72
$3^{rd}$	(0.5, 0.7)	<b>100</b>	<b>100</b>	98.70	98.37	99.14	98.33	94.44	94.70	96.91
	(0.5, 1)	<b>100</b>	<b>100</b>	99.05	98.70	99.43	98.33	96.30	94.70	96.82
	(0.7, 1)	<b>100</b>	<b>100</b>	99.10	96.63	99.43	98.33	95.68	92.80	95.95
	(1, 1.3)	<b>100</b>	<b>100</b>	98.60	95.22	99.43	98.33	94.44	92.80	96.15
	(1, 1.5)	<b>100</b>	<b>100</b>	98.45	98.04	99.14	98.33	95.06	94.32	96.06
$4^{th}$	(0.5, 0.7)	99.00	98.50	98.10	96.30	98.29	98.33	96.30	92.80	95.56
	(0.5, 1)	<b>100</b>	<b>100</b>	99.35	97.39	98.29	98.33	93.21	91.67	95.69
	(0.7, 1)	<b>100</b>	<b>100</b>	98.80	97.93	99.14	98.33	96.91	92.42	96.30
	(1, 1.3)	99.00	99.00	98.35	97.61	96.86	98.33	94.44	92.80	95.27
	(1, 1.5)	<b>100</b>	<b>100</b>	99.30	97.93	97.71	<b>100</b>	93.21	93.18	95.49
$\{1^{st}, 2^{nd}\}$	(0.5, 0.7)	99.50	99.00	98.50	99.02	96.57	<b>100</b>	95.68	94.70	96.93
	(0.5, 1)	<b>100</b>	<b>100</b>	99.10	97.39	99.43	<b>100</b>	96.30	94.70	97.20
	(0.7, 1)	<b>100</b>	<b>100</b>	<b>99.55</b>	99.13	<b>99.71</b>	<b>100</b>	<b>97.53</b>	<b>96.21</b>	97.14
	(1, 1.3)	<b>100</b>	<b>100</b>	98.20	98.80	<b>99.71</b>	<b>100</b>	95.06	94.70	97.02
	(1, 1.5)	<b>100</b>	<b>100</b>	99.05	98.26	<b>99.71</b>	<b>100</b>	95.68	94.70	96.96
$\{1^{st}, 3^{rd}\}$	(0.5, 0.7)	<b>100</b>	<b>100</b>	99.40	97.39	98.86	98.33	95.68	94.32	<b>97.54</b>
	(0.5, 1)	<b>100</b>	<b>100</b>	99.40	98.91	99.43	98.33	96.30	94.70	97.23
	(0.7, 1)	<b>100</b>	<b>100</b>	98.70	98.70	99.43	98.33	95.68	93.56	96.73
	(1, 1.3)	<b>100</b>	<b>100</b>	98.50	95.76	99.43	98.33	95.68	93.18	96.82
	(1, 1.5)	<b>100</b>	<b>100</b>	99.15	95.00	99.14	98.33	96.91	95.45	96.86
$\{1^{st}, 4^{th}\}$	(0.5, 0.7)	99.00	99.00	98.65	96.85	97.71	98.33	96.91	<b>96.21</b>	96.91
	(0.5, 1)	<b>100</b>	<b>100</b>	99.25	99.13	99.14	<b>100</b>	<b>97.53</b>	93.18	97.47
	(0.7, 1)	<b>100</b>	<b>100</b>	99.35	98.70	<b>99.71</b>	<b>100</b>	96.91	94.32	97.21
	(1, 1.3)	99.00	99.00	98.55	96.20	<b>99.71</b>	<b>100</b>	96.91	94.32	97.07
	(1, 1.5)	<b>100</b>	<b>100</b>	99.10	97.83	99.14	<b>100</b>	95.68	94.32	96.79
$\{2^{nd}, 3^{rd}\}$	(0.5, 0.7)	<b>100</b>	99.50	98.75	<b>99.24</b>	99.43	<b>100</b>	96.30	94.70	97.09
	(0.5, 1)	<b>100</b>	<b>100</b>	98.95	97.72	99.43	<b>100</b>	<b>97.53</b>	94.70	96.95
	(0.7, 1)	<b>100</b>	<b>100</b>	98.40	98.70	99.43	<b>100</b>	96.91	<b>96.21</b>	96.93
	(1, 1.3)	<b>100</b>	<b>100</b>	97.80	96.52	99.43	<b>100</b>	96.30	95.45	97.04
	(1, 1.5)	<b>100</b>	<b>100</b>	98.95	97.83	99.14	<b>100</b>	96.30	94.70	97.03
$\{2^{nd}, 4^{th}\}$	(0.5, 0.7)	98.00	98.00	98.00	95.76	98.86	<b>100</b>	96.91	94.32	96.13
	(0.5, 1)	<b>100</b>	<b>100</b>	99.40	98.70	98.57	<b>100</b>	95.06	92.80	96.49
	(0.7, 1)	<b>100</b>	<b>100</b>	99.10	98.70	99.43	<b>100</b>	95.68	93.56	96.19
	(1, 1.3)	99.00	99.00	98.20	96.63	97.71	<b>100</b>	95.68	92.80	96.53
	(1, 1.5)	<b>100</b>	<b>100</b>	99.15	98.04	98.86	<b>100</b>	93.83	92.80	95.92
$\{3^{rd}, 4^{th}\}$	(0.5, 0.7)	99.00	99.00	99.00	95.76	97.71	<b>100</b>	96.91	95.45	97.31
	(0.5, 1)	<b>100</b>	<b>100</b>	99.30	97.83	99.43	<b>100</b>	<b>97.53</b>	94.32	97.04
	(0.7, 1)	<b>100</b>	<b>100</b>	99.45	99.13	99.43	<b>100</b>	<b>97.53</b>	<b>96.21</b>	97.09
	(1, 1.3)	99.00	99.00	97.95	95.65	99.43	<b>100</b>	<b>97.53</b>	95.08	96.91
	(1, 1.5)	<b>100</b>	<b>100</b>	98.25	96.96	98.86	<b>100</b>	96.91	<b>96.21</b>	96.88

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

Table 6.29: Rates (%) of DoDGF $^{3D}_{\sigma,\sigma',\mathcal{F}}$  and DoGF $^{3D}_{\sigma,\sigma'}$  descriptors on benchmark datasets.

DoGF/DoDGF		UCLA				DynTex				DynTex++
Order(s)	$(\sigma, \sigma')$	50-LOO	50-4fold	9-class	8-class	DynTex35	Alpha	Beta	Gamma	
$0^{th}$	(0.5, 0.7)	99.50	99.50	98.85	97.83	97.14	96.67	93.83	92.42	95.04
	(0.5, 1)	<b>100</b>	<b>100</b>	98.45	97.83	97.43	98.33	92.59	92.42	95.09
	(0.7, 1)	<b>100</b>	<b>100</b>	98.75	97.93	96.57	<b>100</b>	91.98	94.70	94.98
	(1, 1.3)	<b>100</b>	<b>100</b>	98.70	97.93	97.14	96.67	90.74	93.56	93.63
	(1, 1.5)	<b>100</b>	<b>100</b>	98.90	97.39	98.86	98.33	92.59	93.56	93.36
$1^{st}$	(0.5, 0.7)	99.50	99.50	98.40	98.59	98.29	98.33	96.91	95.45	97.19
	(0.5, 1)	<b>100</b>	<b>100</b>	98.20	98.15	99.43	98.33	96.30	95.45	96.88
	(0.7, 1)	<b>100</b>	<b>100</b>	99.10	99.24	<b>100</b>	98.33	97.53	96.21	97.15
	(1, 1.3)	<b>100</b>	<b>100</b>	98.90	98.04	<b>100</b>	98.33	96.91	96.59	96.99
	(1, 1.5)	<b>100</b>	<b>100</b>	99.40	98.26	98.86	98.33	97.53	95.83	96.52
$2^{nd}$	(0.5, 0.7)	99.00	99.00	98.65	97.39	98.29	<b>100</b>	96.30	96.21	97.09
	(0.5, 1)	99.50	99.50	98.90	98.48	98.00	<b>100</b>	94.44	96.21	96.84
	(0.7, 1)	<b>100</b>	<b>100</b>	98.60	<b>99.57</b>	99.14	<b>100</b>	96.30	95.08	96.47
	(1, 1.3)	<b>100</b>	<b>100</b>	99.10	98.26	98.29	<b>100</b>	94.44	96.21	96.89
	(1, 1.5)	<b>100</b>	<b>100</b>	99.25	99.13	98.29	98.33	94.44	95.45	96.27
$3^{rd}$	(0.5, 0.7)	99.50	99.50	98.80	98.70	99.14	98.33	96.91	95.45	97.15
	(0.5, 1)	99.50	99.50	98.75	98.15	99.14	98.33	96.30	95.08	96.51
	(0.7, 1)	<b>100</b>	<b>100</b>	99.35	99.46	98.57	98.33	96.30	95.83	96.39
	(1, 1.3)	<b>100</b>	<b>100</b>	99.10	98.70	98.57	98.33	96.30	95.83	96.24
	(1, 1.5)	<b>100</b>	<b>100</b>	<b>99.75</b>	97.83	98.86	96.67	96.30	96.59	96.72
$4^{th}$	(0.5, 0.7)	99.00	99.00	98.00	95.76	97.71	96.67	95.06	93.18	96.72
	(0.5, 1)	<b>100</b>	<b>100</b>	98.70	98.48	96.86	<b>100</b>	93.21	95.45	96.57
	(0.7, 1)	<b>100</b>	<b>100</b>	98.15	98.70	98.00	96.67	95.06	93.56	95.69
	(1, 1.3)	98.00	98.00	98.50	95.33	98.00	<b>100</b>	94.44	95.45	96.11
	(1, 1.5)	<b>100</b>	99.50	99.10	98.59	98.57	98.33	92.59	95.45	95.62
$\{1^{st}, 2^{nd}\}$	(0.5, 0.7)	99.50	99.50	98.55	98.70	96.29	<b>100</b>	96.92	96.21	97.62
	(0.5, 1)	99.50	99.50	98.50	97.61	99.43	<b>100</b>	95.68	96.97	97.55
	(0.7, 1)	<b>100</b>	<b>100</b>	99.25	<b>99.57</b>	99.71	<b>100</b>	<b>98.15</b>	96.97	97.52
	(1, 1.3)	<b>100</b>	<b>100</b>	99.40	96.96	99.71	<b>100</b>	96.30	96.97	97.40
	(1, 1.5)	<b>100</b>	<b>100</b>	98.75	97.72	99.71	98.33	95.68	96.59	96.97
$\{1^{st}, 3^{rd}\}$	(0.5, 0.7)	<b>100</b>	<b>100</b>	98.65	97.93	98.86	98.33	96.91	95.83	97.46
	(0.5, 1)	<b>100</b>	<b>100</b>	98.60	99.13	99.71	98.33	96.91	96.59	97.48
	(0.7, 1)	<b>100</b>	<b>100</b>	98.75	99.13	99.71	98.33	96.30	96.97	97.18
	(1, 1.3)	<b>100</b>	<b>100</b>	99.60	97.83	99.71	98.33	96.30	<b>97.35</b>	96.83
	(1, 1.5)	<b>100</b>	<b>100</b>	99.05	98.26	98.57	96.67	96.30	96.59	97.07
$\{1^{st}, 4^{th}\}$	(0.5, 0.7)	99.50	99.50	98.45	96.52	96.00	98.33	97.53	96.21	97.46
	(0.5, 1)	<b>100</b>	<b>100</b>	98.35	96.96	99.71	<b>100</b>	95.68	96.21	97.49
	(0.7, 1)	<b>100</b>	<b>100</b>	99.35	96.63	<b>100</b>	<b>100</b>	97.53	96.97	<b>97.95</b>
	(1, 1.3)	98.00	98.00	98.70	95.11	99.43	<b>100</b>	95.68	95.83	97.47
	(1, 1.5)	<b>100</b>	<b>100</b>	98.70	97.50	99.14	<b>100</b>	95.06	96.21	97.34
$\{2^{nd}, 3^{rd}\}$	(0.5, 0.7)	99.50	99.50	98.35	98.37	99.43	<b>100</b>	96.91	96.59	97.37
	(0.5, 1)	99.50	99.50	99.15	98.26	99.14	<b>100</b>	97.53	96.21	97.08
	(0.7, 1)	<b>100</b>	<b>100</b>	99.30	98.70	99.43	<b>100</b>	<b>98.15</b>	96.59	97.03
	(1, 1.3)	<b>100</b>	<b>100</b>	98.90	98.04	99.43	<b>100</b>	96.91	96.97	96.62
	(1, 1.5)	<b>100</b>	<b>100</b>	99.35	97.50	98.86	96.67	95.70	96.97	97.01
$\{2^{nd}, 4^{th}\}$	(0.5, 0.7)	99.50	99.50	98.15	97.93	97.14	<b>100</b>	95.06	95.83	97.01
	(0.5, 1)	99.50	99.50	99.30	97.39	98.86	<b>100</b>	94.44	<b>97.35</b>	97.11
	(0.7, 1)	<b>100</b>	<b>100</b>	99.20	97.39	98.00	<b>100</b>	96.91	94.70	96.88
	(1, 1.3)	98.00	98.00	98.80	97.83	97.71	<b>100</b>	95.06	95.83	96.87
	(1, 1.5)	<b>100</b>	<b>100</b>	99.25	99.13	98.86	<b>100</b>	94.44	96.21	96.62
$\{3^{rd}, 4^{th}\}$	(0.5, 0.7)	99.50	99.50	98.80	97.28	98.86	98.33	<b>98.15</b>	96.21	97.16
	(0.5, 1)	<b>100</b>	<b>100</b>	99.50	98.59	99.43	<b>100</b>	97.53	95.83	96.64
	(0.7, 1)	<b>100</b>	<b>100</b>	99.55	97.39	99.43	98.33	97.53	96.59	97.55
	(1, 1.3)	98.00	98.00	98.85	97.18	99.43	<b>100</b>	96.91	96.59	96.53
	(1, 1.5)	<b>100</b>	<b>100</b>	99.35	97.39	99.14	98.33	95.06	96.59	96.45

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



convolving operation  $x, y, z \in [-3\sigma, 3\sigma]$ . Pairs of standard deviations are empirically investigated as  $\{(\sigma, \sigma')\} = \{(0.5, 0.7), (0.5, 1), (0.7, 1), (1, 1.3), (1, 1.5)\}$ .

**For structuring DoDGF $^{2D/3D}_{\sigma, \sigma', \mathcal{F}}$  descriptors:** To construct the proposed DoDG-based descriptors, we simply utilize CLBP<sup>3</sup>, one of the most popular local operators, with 3D-joint setting of *riu2* mapping and a supporting region  $(P, R) = (8, 1)$ . It means  $\Psi = \text{CLBP}_{8,1}^{\text{riu2}}$  corresponding to  $\mathcal{H}_\Psi = 2(P+2)^2$  bins for a pattern description, where  $P$  denotes a number of neighbors involved in the DT encoding. Consequently, it takes a small dimension for single-scale analysis of high-order DoDG filterings (i.e.,  $|\mathcal{F}| = 1$ ) to describe a given video, just  $4 \times 3 \times |\mathcal{F}| \times \mathcal{H}_\Psi = 2400$  bins for DoDGF $^{2D}_{\sigma, \sigma', \mathcal{F}}$  and  $6 \times 3 \times |\mathcal{F}| \times \mathcal{H}_\Psi = 3600$  bins for DoDGF $^{3D}_{\sigma, \sigma', \mathcal{F}}$ , where  $|\mathcal{F}| = \text{card}(\mathcal{F})$  denotes the number of  $k$ -orders in  $\mathcal{F}$  taken into account a multi-order analysis. Table 6.2 shows a comprehensive comparison between dimension of DoDGF $^{2D/3D}$  descriptors and that of other LBP-based ones.

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

### 6.9.2.2 Assessments of DoDG-based descriptors

Based on the experimental results in Tables 6.28 and 6.29, it can be stated that our novel DoDG filtering kernel is the major factor in order to boost the discrimination of the DoDGF $^{2D/3D}$  descriptors. Hereafter, we discuss their performance thoroughly.

- The DoDG-based descriptors' performance is diminished subject to the increasing high-orders of DoDG involved in the filterings. It is due to the weakness of appearances in the larger-orders. Therein, the odd DoDG kernels often handle denoising in more effect (see Tables 6.28 and 6.29).
- Multi-order analysis of DoDG in both even and odd gradients points out better power, while just consisting of either entire even or odd ones is not (see Tables 6.28 and 6.29) due to a better complementarity between the former ones.
- Local patterns extracted from each of the DoDG-filtered outcomes are complementary to enhance the robustness. Indeed, Table 6.30 shows that DoDGF $^{2D}$  has higher rates when integrating all those, as mentioned in Section 6.9.1.
- It can be seen from Tables 6.28 and 6.29 that the DoDGF $^{2D/3D}$  descriptors have the nearly same rates on simple datasets (e.g., UCLA). However, for the challenging schemes (i.e., *Beta* and *Gamma*), the DoDGF $^{3D}$  one has much better results. This has proved that exploiting the 3D DoDG kernel could enrich more robust spacial-filtered information for DT representation compared to using the 2D one. Figure 6.29 intuitively shows this prominent point addressed in different orders of those.
- Taking a coherence of both odd and even DoDG filterings into account multi-order analysis gives better rates compared to doing that with the whole either odd or even ones (see Tables 6.28 and 6.29 for results in 2-scale of orders). It is due to the fact that the kernels of odd and even orders are complementary since the first ones are semi-symmetric shapes while the second ones are symmetric shapes (see Section 6.3.2 for these properties and Figure 6.3 for illustration with 1D DoDG kernels).

In general, the single-order DoDGF $^{2D/3D}_{\sigma, \sigma', \mathcal{F}}$  descriptors with the setting of  $(\sigma, \sigma') = (0.7, 1)$  often points out the best results on UCLA and Alpha datasets (see Tables 6.28 and 6.29). Moreover, the odd-even DoDGF $^{2D/3D}_{(0.7, 1), \mathcal{F}}$  descriptors in multi-order analysis (i.e.,  $\{1^{st}, 2^{nd}\}$ ,  $\{1^{st}, 4^{th}\}$ ,  $\{2^{nd}, 3^{rd}\}$ ,

<sup>3</sup>CLBP [3] operator is addressed in this work for a purpose of simplicity in implementing and evaluating the effectiveness of our novel DoDG filtering for DT representation compared to the well-known DoG. It could be absolutely replaced by other robust ones for further improvement in practice, e.g., CLBC [82], LDP-based [30, J5], LVP-based [100, J2], LRP [J4], MRELBP [78], etc.

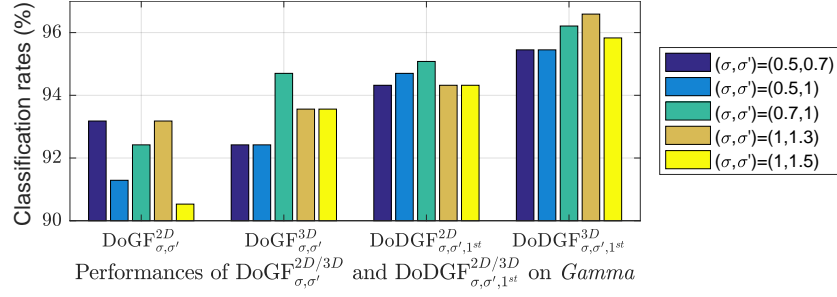


Figure 6.29: Comparing performances of DoGF $_{\sigma, \sigma'}^{2D/3D}$  with those of the 1<sup>st</sup>-order DoGF $_{\sigma, \sigma', 1st}^{2D/3D}$ .

Table 6.30: Comparing contributions of DoG and the 1<sup>st</sup>-order of DoDG.

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

Note: Dyn35 and Dyn++ are shortened for DynTex35 and DynTex++ respectively.

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

### 6.9.2.3 Comprehensive comparison to DoG-based descriptors

It can be verified from Tables 6.28 and 6.29 that our proposed DoDG-based descriptors using the novel DoDG filterings are much powerful execution compared to those using the well-known DoG kernel, i.e., which of the 0<sup>th</sup>-order in Tables 6.28 and 6.29. In consideration of the contributions of complementary filtered outcomes as shown in Table 6.30, it could assert the prominent performance of Gaussian-gradient parts in DoDG compared to that of non-Gaussian-gradient ones in DoG. This has proved that our novel filtering kernel is more influential for local DT understanding.

Furthermore, the DoDG-filtered features are also more discriminative than those of DoGs in FoSIG [C2] and V-BIG [C5], where both blurred and invariant Gaussian-based characteristics are taken into account the DT encoding. It can be seen from Figure 6.30 for a comprehensive comparison of their performances, where all the descriptors are constructed by the same CLBP $_{8,1}^{riu2}$  for capturing spatio-temporal features in DoG/DoDG-based outcomes.

## 6.10 Comprehensive evaluations in comparison with existing methods

As presented above, firstly, we have proposed the novel DoDG filtering kernel (see Section 6.3). After that, we have constructed several discriminative descriptors for DT representation using our en-

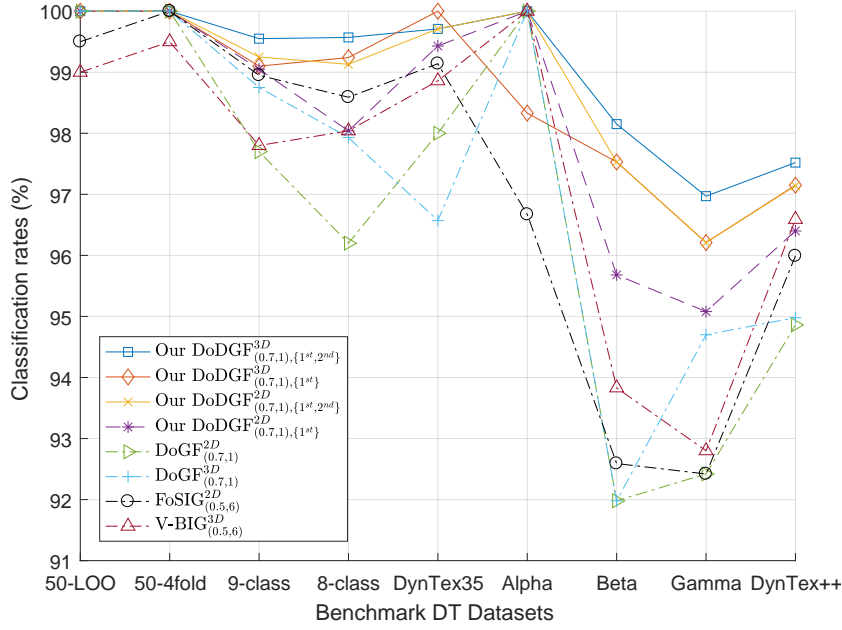


Figure 6.30: Comparison of performances of several local-feature-based descriptors using the same  $\text{CLBP}_{8,1}^{riu2}$  for encoding DoG/DoDG-based outcomes. Therein, it should be noted that rates of V-BIG [C5] and FoSIG [C2] are referred to their original works, where Gaussian-blurred outcomes are also addressed for the DT encoding as complemented features in addition to DoGs.

hanced operators in Chapter 3 in order to efficiently capture spatio-temporal features from robust filtered outcomes which are extracted by different variants of Gaussian-based filterings. Those descriptors are LOGIC [S4] in Section 6.6, CHILOP [S2] in 6.4, RUBIG [J4] in 6.5, SIOMF/SVOMF [S3] in 6.7, HoGF [J3] in 6.8, and DoDGF [S1] in 6.9. The experimental results have also presented for each of them, which have verified that our proposals have very good performances, especially, those of HoGF [J3] and DoDGF [S1]. Among of them, DoDGF [S1] has the best rates in small dimension, expected as one of appreciated solutions for slight applications in mobile devices and embedded sensor systems, which are required to execute their functions in restricted resources. Therefore, as a representative, we mainly address the performances of our DoDGF [S1] descriptor in below evaluations. Accordingly, in this section, we thoroughly discuss advantages of our proposals allowing to boost the discrimination power in DT recognition, compared to state of the art. Hereafter, we express those influential evaluations in detail.

### 6.10.1 Benefits of Gaussian-based filterings

#### 6.10.1.1 Robustness to the well-known issues of DT description

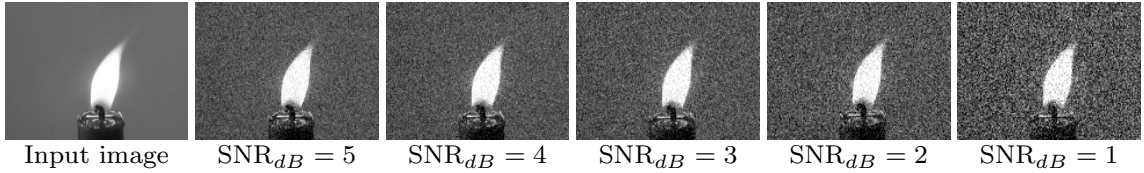
In general, it could be verified that addressing the Gaussian-based filterings for DT representation has pointed out the filtered responses in more robustness. This allows that local spatio-temporal features extracted from these outcomes are more insensitive for DT encoding compared to those extracted from a raw video. Indeed, in order to evaluate this advantageous property, we will investigate our proposed descriptors, based on different variants of the Gaussian-based filterings, on noisy datasets to evaluate their ability of noise-resistance.

Accordingly, we address the Gaussian zero-mean noise model with different signal-to-noise ratio (SNR) levels, i.e.,  $\text{SNR}_{dB} \in \{1, 2, 3, 4, 5\}$ , in order to add noise into UCLA [5] - the simple dataset, and DynTex [54] - the more challenging one (see Table 2.1 for their attributes in detail). For each of them, we have achieved 5 noise-datasets corresponding to 5  $\text{SNR}_{dB}$  levels used for the noise-adding process. Figure 6.31 shows noise-instances obtained by using different levels of  $\text{SNR}_{dB}$  on a plane-image in a video of UCLA dataset. We comprehensively evaluate our proposed descriptors in the ability

Table 6.31: Performances on different Gaussian noise subsets: 50-4fold and Gamma.

Descriptor	Filter	Order	$\{(\sigma, \sigma')\}$	$\{(P, \{R\})\}$	SNR <sub>dB</sub> for 50-4fold						SNR <sub>dB</sub> for Gamma					
					No-dB	dB=1	dB=2	dB=3	dB=4	dB=5	No-dB	dB=1	dB=2	dB=3	dB=4	dB=5
VLBP [14]	None	-	-	$\{(4, 1)\}$	96.00	91.00	93.00	92.00	94.00	94.00	92.80	87.12	88.64	89.02	90.91	90.53
LBP-TOP [14]	None	-	-	$\{(8, 1)\}$	97.50	97.50	99.00	99.50	99.00	98.50	93.56	77.65	81.82	84.47	86.36	87.12
CLSP-TOP [C1]	None	-	-	$\{(8, 1)\}$	99.00	98.00	<b>100</b>	99.50	99.50	99.00	93.18	82.95	84.85	84.47	86.36	87.50
HILOP [C3]	None	-	-	$\{(8, \{1, 2\})\}$	99.50	99.50	99.50	99.50	99.50	99.50	92.42	88.64	89.77	90.91	90.91	91.29
CLBP <sub>S/M/C</sub> [3]	None	-	-	$\{(8, 1)\}$	83.50	90.00	89.50	90.00	88.50	89.50	88.64	74.24	78.41	80.68	83.33	84.47
CLBP <sub>S/M/C</sub> [3]	None	-	-	$\{(8, 1)\}$	99.50	99.50	99.50	99.50	99.00	<b>99.50</b>	92.80	85.98	87.12	87.88	88.64	89.39
CLBP <sub>S/M/C</sub> [3]	Gaussian	0 <sup>th</sup>	$\{0.5, 1.0\}$	$\{(8, 1)\}$	97.50	98.00	99.00	99.00	99.00	99.00	89.77	87.50	87.88	88.64	88.64	88.64
CLBP <sub>S/M/C</sub> [3]	Gaussian	0 <sup>th</sup>	$\{0.5, 1.0\}$	$\{(8, 1)\}$	<b>100</b>	99.50	99.50	99.50	<b>99.50</b>	<b>99.50</b>	93.18	87.88	89.39	90.53	90.15	90.91
ZoGF <sup>2D</sup>	Gaussian	0 <sup>th</sup>	$\{1\}$	$\{(8, 1)\}$	<b>100</b>	<b>100</b>	<b>100</b>	99.50	99.00	99.00	92.42	88.64	90.15	89.39	89.39	88.64
ZoGF <sup>3D</sup>	Gaussian	0 <sup>th</sup>	$\{1\}$	$\{(8, 1)\}$	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	93.56	90.53	90.53	90.91	90.15	90.91
DoGF <sup>2D</sup>	DoG	0 <sup>th</sup>	$\{(0.7, 1)\}$	$\{(8, 1)\}$	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	92.42	81.06	86.74	88.64	89.02	88.26
DoGF <sup>3D</sup>	DoG	0 <sup>th</sup>	$\{(0.7, 1)\}$	$\{(8, 1)\}$	<b>100</b>	99.50	<b>100</b>	99.50	<b>100</b>	<b>100</b>	94.70	87.88	89.77	90.15	91.29	89.77
CHILOP <sub>H/M/C</sub> [S2]	None	-	-	$\{(8, \{1, 2\})\}$	<b>100</b>	99.50	<b>100</b>	99.50	<b>99.50</b>	<b>99.50</b>	91.29	85.98	88.29	87.88	87.88	87.88
CHILOP <sub>H/M/C</sub> [S2]	None	-	-	$\{(8, \{1, 2\})\}$	<b>100</b>	99.50	<b>100</b>	<b>100</b>	<b>99.50</b>	<b>99.50</b>	92.05	87.50	89.02	90.15	90.53	90.91
CHILOP <sub>H/M/C</sub> [S2]	Gaussian	0 <sup>th</sup>	$\{0.5, 1.0\}$	$\{(8, \{1, 2\})\}$	<b>100</b>	<b>100</b>	<b>100</b>	99.50	<b>99.50</b>	<b>99.50</b>	89.39	88.64	88.67	87.50	87.12	88.26
CHILOP <sub>H/M/C</sub> [S2]	Gaussian	0 <sup>th</sup>	$\{0.5, 1.0\}$	$\{(8, \{1, 2\})\}$	<b>100</b>	<b>100</b>	99.50	99.00	99.00	99.00	94.32	90.15	91.67	92.42	<b>92.05</b>	91.67
HoGF <sup>2D</sup> [J3]	Gaussian	1 <sup>st</sup>	$\{1\}$	$\{(8, 1)\}$	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	93.56	90.53	90.53	90.15	90.91	90.53
HoGF <sup>3D</sup> [J3]	Gaussian	1 <sup>st</sup>	$\{1\}$	$\{(8, 1)\}$	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>96.21</b>	<b>90.91</b>	<b>92.05</b>	<b>93.18</b>	<b>92.05</b>	92.05
DoDGF <sup>2D</sup> [S1]	DoDG	1 <sup>st</sup>	$\{(0.7, 1)\}$	$\{(8, 1)\}$	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	95.08	89.77	90.53	89.77	90.53	91.29
DoDGF <sup>3D</sup> [S1]	DoDG	1 <sup>st</sup>	$\{(0.7, 1)\}$	$\{(8, 1)\}$	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>96.21</b>	<b>90.91</b>	91.67	91.67	91.67	<b>92.80</b>

Note: “-” means “not available”. “<sub>S/M/C</sub>” and “<sub>S/M/C</sub>” respectively denote 2D and 3D jointing histograms of CLBP’s components, as mentioned in Section 2.7.2. No-dB denotes results without the Gaussian noise involved in. CLSP-TOP [C1] is structured using thresholding settings  $a = 0$  and  $b = 1$ . All above descriptors are reduced their dimension by using “riu2” mapping, excluding HILOP [C3] and LBP-TOP [14] using “u2” while no mapping is applied to VLBP [14].


 Figure 6.31: Noise-instances obtained by using different levels of SNR<sub>dB</sub> on a plane-image in a video of UCLA dataset.

of noise-resistance on these datasets, compared to other LBP-based ones:

- DoDG-based descriptors [S1] defined in Section 6.9.1.
- DoG-based descriptors, i.e., DoGF defined in Section 6.9.1.
- HoGF-based descriptors [J3] using high-order Gaussian gradients as defined in Section 6.8.
- ZoGF-based descriptors implemented in Section 6.8.2 using the original Gaussian filterings.
- CHILOP-based descriptors [S2] using the original 2D Gaussian filtering (see Section 6.4.1).
- Other LBP-based descriptors without taking any filters into account their DT encoding, e.g., VLBP [14], LBP-TOP [14], CLSP-TOP [C1], HILOP [C3].

The specific parameters for those along with the achieved results of DT recognition are presented in Table 6.31. It can be seen that taking the Gaussian-based kernels into account the DT encoding makes our proposed descriptors more robust against noise for local DT encodings compared to the ones with non-Gaussians applied to. Therein, those based on the Gaussian-gradient-based kernels, i.e., HoGF and DoDGF, obtain the best performance. Specifically, our DoDGF<sup>2D/3D</sup> and HoGF<sup>2D/3D</sup> descriptors almost absolutely resist to the Gaussian noise for the simple scheme, i.e., 50-4fold. In the meanwhile, except that the VLBP’s performance has decreased sharply by 3%, the noise-resistant ability of the rest is approximately the same execution in general (see Table 6.31). On the challenging scheme, i.e., Gamma, the performance of both DoDGF<sup>2D/3D</sup> has dropped by about 2%, by about 1% for HoGF<sup>3D</sup>, while that of HoGF<sup>2D</sup> is in more stability. In terms of the ability of other LBP-based variants without filters applied to, all of them have a sharp decrease compared to ours (see Figure 6.32 for a graphical view). Consequently, this has proved the impressive property of the proposed Gaussian-based kernels and their gradients making our descriptors more robust in noisy conditions.

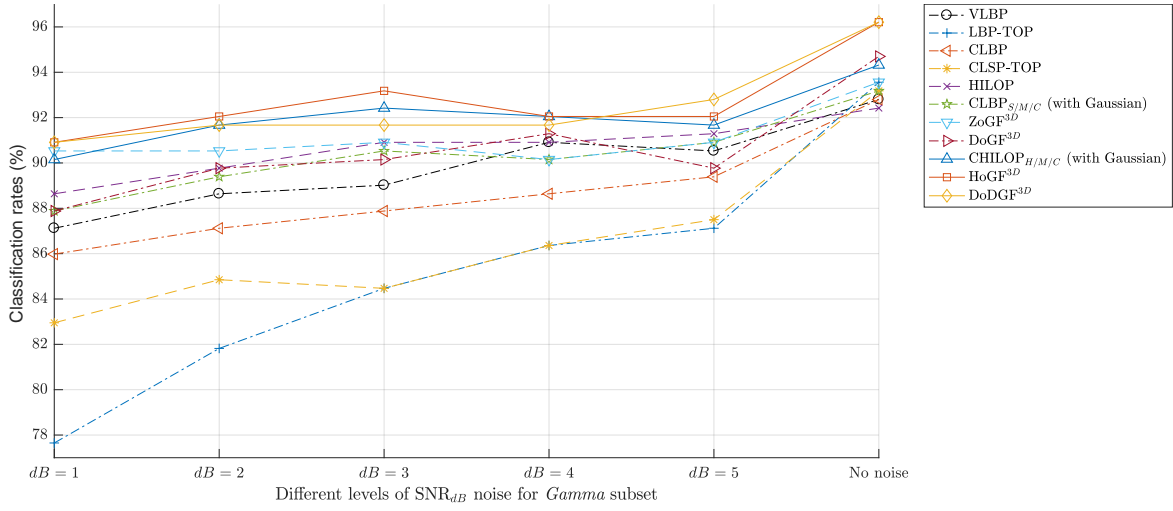


Figure 6.32: The impacts of Gaussian noise on performances of several proposed descriptors compared to others.

### 6.10.1.2 Rich and discriminative features of Gaussian-gradient-based filterings

The Gaussian-gradient-based filterings can point out more filtered outcomes than the well-known Gaussian kernel and its DoG (see Figure 6.24 for Gaussian gradients and 6.4 for DoDG filtering kernels). This allows to exploit spatio-temporal features in more forceful contexts to enhance the discrimination power. Table 6.30 shows contributions of DoDG-filtered parts, while Table 6.24 is contributions of Gaussian-gradient-filtered elements for structuring the HoGF-based features. Also, these filterings could be computed in higher orders of partial derivatives to conduct high-gradient features: the even orders for capturing the symmetric features, and the odd ones for the asymmetric characteristics and then combinations of them could boost improvement of the performance. It could be verified this beneficial points through their filtered samples in Figures 6.24 and 6.4, as well as their performances in Tables 6.22 and 6.23 for HoGF-based descriptors; Tables 6.28 and 6.29 for DoDG-based descriptors. In addition, the magnitude properties, elicited from different filtered components with the same level of derivatives, (i.e.,  $\nabla \mathcal{I}_{x^k, y^k}^\sigma$  and  $\nabla \mathcal{V}_{x^k, y^k, z^k}^\sigma$ , see Section 6.8.1) allow to address more amplitude characteristics to enrich informative discrimination of appearance and motion for DT understanding (see Table 6.24 for specific contributions of these components).

### 6.10.2 Complexity of our proposed descriptors

In general, it can be verified that the computational cost of our proposed descriptors is the same order as that of other LBP-based ones. For a representative, we specifically present the complexity of computing the  $\text{DoDGF}_{\sigma, \sigma', \mathcal{F}}^{2D/3D}$  descriptors. Thanks to the separable and linear properties of DoDG's convolving operation which is inherited from the well-known Gaussian filtering kernel, structuring of DoDGF is in simple computation. Indeed, for a video  $\mathcal{V}$  with  $\mathcal{H} \times \mathcal{W} \times \mathcal{T}$  dimension, let  $\mathcal{Q}_{\text{LBP-TOP}} = \mathcal{O}(P \times \mathcal{H} \times \mathcal{W} \times \mathcal{T})$  be the complexity of LBP-TOP [14] for encoding  $\mathcal{V}$ , where  $P \in \mathbb{Z}^+$  denotes a number of concerning neighbors. Since CLBP [3] with its three complementary components is taken into account encoding  $\mathcal{V}$  (see Section 6.9.2.1), it could be inferred that the computational cost of CLBP for encoding  $\mathcal{V}$  is  $\mathcal{Q}_{\text{CLBP}} \approx 3 \times \mathcal{Q}_{\text{LBP-TOP}}$ . Accordingly, the complexity of  $\text{DoDGF}_{\sigma, \sigma', \mathcal{F}}^{2D/3D}$  descriptors is estimated as  $\mathcal{Q}_{\text{DoDGF}^{2D}} = 4 \times |\mathcal{F}| \times \mathcal{Q}_{\text{CLBP}} + \mathcal{Q}_{\text{DoDG}^{2D}}$  for  $\text{DoDGF}_{\sigma, \sigma', \mathcal{F}}^{2D}$ , and  $\mathcal{Q}_{\text{DoDGF}^{3D}} = 6 \times |\mathcal{F}| \times \mathcal{Q}_{\text{CLBP}} + \mathcal{Q}_{\text{DoDG}^{3D}}$  for  $\text{DoDGF}_{\sigma, \sigma', \mathcal{F}}^{3D}$ , where  $\mathcal{Q}_{\text{DoDG}^{2D/3D}}$  is the cost of corresponding DoDG<sup>2D/3D</sup> filterings involved in the DT representation (refer to Section 6.9.1 for their constructions). Due to the separable and linear properties of the DoDG filterings as well as the much smallness of  $|\mathcal{F}|$  (e.g.,  $|\mathcal{F}| = 2$  for two orders in Tables 6.28 and 6.29),  $\mathcal{Q}_{\text{DoDG}^{2D/3D}}$  and  $|\mathcal{F}|$  could be

Table 6.32: Comparison of processing time of encoding a  $50 \times 50 \times 50$  video in DynTex++.

Descriptor	$\{(\sigma, \sigma')\}$	$\{(P, R)\}$	Mapping	Runtime (s)
VLBP [14]	-	$\{(4, 1)\}$	-	$\approx 0.22$
LBP-TOP [14]	-	$\{(8, 1)\}$	u2	$\approx 0.15$
CLSP-TOP [C1]	-	$\{(8, 1)\}$	riu2	$\approx 0.27$
CSAP-TOP [J1]	-	$\{(8, 1)\}$	riu2	$\approx 0.50$
HILOP [C3]	-	$\{(8, \{1, 2\})\}$	u2	$\approx 0.42$
FoSIG [C2]	$\{(0.5, 6)\}$	$\{(8, 1)\}$	riu2	$\approx 0.37$
V-BIG [C5]	$\{(0.5, 6)\}$	$\{(8, 1)\}$	riu2	$\approx 0.35$
RUBIG [J4]	$\{(0.5, 6)\}$	$\{(8, 1)\}$	riu2	$\approx 0.56$
HoGF <sup>2D</sup> [J3]	$\{\sigma = 1\}$	$\{(8, 1)\}$	riu2	$\approx 0.54$
HoGF <sup>3D</sup> [J3]	$\{\sigma = 1\}$	$\{(8, 1)\}$	riu2	$\approx 0.70$
LOGIC <sup>2D</sup> [S4]	$\{(0.5, 6)\}$	$\{(8, 1)\}$	riu2	$\approx 0.77$
LOGIC <sup>3D</sup> [S4]	$\{(0.5, 6)\}$	$\{(8, 1)\}$	riu2	$\approx 0.75$
DoDGF <sup>2D</sup> [S1]	$\{(0.7, 1)\}$	$\{(8, 1)\}$	riu2	$\approx 0.58$
DoDGF <sup>3D</sup> [S1]	$\{(0.7, 1)\}$	$\{(8, 1)\}$	riu2	$\approx 0.79$

Note: “-” means “not available”. The 1<sup>st</sup>-order Gaussian gradients are applied to HoGF and DoDGF in this case. Runtimes of all descriptors are estimated using their basic settings.

ignored. Consequently,  $\mathcal{Q}_{\text{DoDGF}^{2D/3D}} \approx \mathcal{O}(P \times \mathcal{H} \times \mathcal{W} \times \mathcal{T})$ . Also, addressing CLBP for encoding  $\mathcal{V}$  (see Sections 6.9.1 and 6.9.2.1), the computational cost of  $\text{DoGF}_{\sigma, \sigma'}^{2D/3D}$  could be conducted as  $\mathcal{Q}_{\text{DoGF}^{2D/3D}} \approx \mathcal{O}(P \times \mathcal{H} \times \mathcal{W} \times \mathcal{T})$ . Therefore, it could be authenticated that our  $\mathcal{Q}_{\text{DoDGF}^{2D/3D}}$  is also the same order as FoSIG [C2], V-BIG [C5], RUBIG [J4], CSAP-TOP [J1], CVLBP [91], CVLBC [90], VLBP [14], CHILOP [S2], HoGF [J3], etc. (refer to those works for more detail of computation). In regards to processing time, our proposed descriptors are implemented on the alike computing system: a 64-bit Linux desktop of single-thread CPU Core i7 3.4GHz 16G RAM. other LBP-based ones are also addressed this system for an impartial evaluation. Table 6.32 shows that runtime of encoding our descriptors of a  $50 \times 50 \times 50$  video is nearly the same as that of other LBP-based ones.

### 6.10.3 Comprehensive discussions of DT classification on different datasets

Mostly obtaining best performance in small dimension compared to our others, in this section, we mainly allocate the DoDGF’s ability of DT recognition for a comprehensive comparison with the state of the art. It should be noted that below discussions of our proposed descriptors are related to the best settings (correspondingly see the above sections for more detail) recommended for real applications as well as for thorough evaluations. Accordingly, it can be seen from Table 6.33 that our DoDG-based descriptors have obtained the best rates compared to all non-deep-learning methods. Their performances are also better than those of deep-learning-based approaches on UCLA as well as very closed to those on DynTex and DynTex++. This is certainly thanks to the leverage contribution of the novel DoDG kernels. Hereunder, we detail particular discussions of those on each benchmark dataset.

#### 6.10.3.1 Classification on UCLA

It can be verified from Table 6.33 that thanks to the efficiently denoising processes of the novel DoDG filterings, our simple  $\text{DoDGF}^{2D/3D}$  descriptors perform very well compared to state of the art, including the deep-learning methods, i.e., DT-CNN [63]. More specifically, DoDGFs obtain the best rates of 100% on both schemes of *50-class* and *50-4fold*. In terms of classifying DTs on *9-class* and *8-class*, our proposal is just a little inferior to DNGP [38] (99.6%) on *9-class*, while achieving the highest rate of 99.57% on *8-class* by  $\text{DoDGF}_{(0.7, 1), \{1^{st}, 2^{nd}\}}^{3D}$ , the same as FD-MAP’s [C4]. It should be noted that

Table 6.33: Comparison of DT recognition rates (%) on benchmark DT datasets

	Dataset	UCLA				DynTex				Dyn++
	Encoding method	50-LOO	50-4fold	9-class	8-class	Dyn35	Alpha	Beta	Gamma	
A	FDT [C4]	98.50	99.00	97.70	99.35	98.86	98.33	93.21	91.67	95.31
	FD-MAP [C4]	99.50	99.00	99.35	<b>99.57</b>	98.86	98.33	92.59	91.67	95.69
	DDTP [J2]	99.00	99.50	98.75	98.04	99.71	96.67	93.83	91.29	95.09
B	AR-LDS [5]	89.90 <sup>N</sup>	-	-	-	-	-	-	-	-
	KDT-MD [40]	-	97.50	-	-	-	-	-	-	-
	NLDR [43]	-	-	-	80.00	-	-	-	-	-
	Chaotic vector [42]	-	-	85.10 <sup>N</sup>	85.00 <sup>N</sup>	-	-	-	-	-
C	3D-OTF [51]	-	87.10	97.23	99.50	96.70	83.61	73.22	72.53	89.17
	WMFS [52]	-	-	97.11	96.96	-	-	-	-	-
	NLSSA [114]	-	-	-	-	-	-	-	-	92.40
	DFS [50]	-	<b>100</b>	97.50	99.20	97.16	85.24	76.93	74.82	91.70
	2D+T [94]	-	-	-	-	-	85.00	67.00	63.00	-
	STLS [53]	-	99.50	97.40	99.50	98.20	89.40	80.80	79.80	94.50
D	MBSIF-TOP [72]	99.50 <sup>N</sup>	-	-	-	98.61 <sup>N</sup>	90.00 <sup>N</sup>	90.70 <sup>N</sup>	91.30 <sup>N</sup>	97.12 <sup>N</sup>
	B3DF.SMC [74]	99.50 <sup>N</sup>	99.50 <sup>N</sup>	98.85 <sup>N</sup>	98.15 <sup>N</sup>	99.71 <sup>N</sup>	95.00 <sup>N</sup>	90.12 <sup>N</sup>	90.91 <sup>N</sup>	95.58 <sup>N</sup>
	DNGP [38]	-	-	<b>99.60</b>	99.40	-	-	-	-	93.80
E	VLBP [14]	-	89.50 <sup>N</sup>	96.30 <sup>N</sup>	91.96 <sup>N</sup>	81.14 <sup>N</sup>	-	-	-	94.98 <sup>N</sup>
	LBP-TOP [14]	-	94.50 <sup>N</sup>	96.00 <sup>N</sup>	93.67 <sup>N</sup>	92.45 <sup>N</sup>	98.33	88.89	84.85 <sup>N</sup>	94.05 <sup>N</sup>
	DDLBP with MJMI [113]	-	-	-	-	-	-	-	-	95.80
	CVLBP [91]	-	93.00 <sup>N</sup>	96.90 <sup>N</sup>	95.65 <sup>N</sup>	85.14 <sup>N</sup>	-	-	-	-
	HLBP [92]	95.00 <sup>N</sup>	95.00 <sup>N</sup>	98.35 <sup>N</sup>	97.50 <sup>N</sup>	98.57 <sup>N</sup>	-	-	-	96.28 <sup>N</sup>
	MEWLSP [95]	96.50 <sup>N</sup>	96.50 <sup>N</sup>	98.55 <sup>N</sup>	98.04 <sup>N</sup>	<b>99.71<sup>N</sup></b>	-	-	-	98.48 <sup>N</sup>
	WLBPC [109]	-	96.50 <sup>N</sup>	97.17 <sup>N</sup>	97.61 <sup>N</sup>	-	-	-	-	95.01 <sup>N</sup>
	CVLBC [90]	98.50 <sup>N</sup>	99.00 <sup>N</sup>	99.20 <sup>N</sup>	99.02 <sup>N</sup>	98.86 <sup>N</sup>	-	-	-	91.31 <sup>N</sup>
	CLSP-TOP [C1]	99.00 <sup>N</sup>	99.00 <sup>N</sup>	98.60 <sup>N</sup>	97.72 <sup>N</sup>	98.29 <sup>N</sup>	95.00 <sup>N</sup>	91.98 <sup>N</sup>	91.29 <sup>N</sup>	95.50 <sup>N</sup>
	CSAP-TOP [J1]	99.50	99.50	96.80	95.98	<b>100</b>	96.67	92.59	90.53	-
	FoSIG [C2]	99.50	<b>100</b>	98.95	98.59	99.14	96.67	92.59	92.42	95.99
	V-BIG [C5]	99.50	99.50	97.95	97.50	99.43	<b>100</b>	95.06	94.32	96.65
	HILOP [C3]	99.50	99.50	97.80	96.30	99.71	96.67	91.36	92.05	96.21
	MMDP [J5]	<b>100</b>	<b>100</b>	98.70	98.70	99.43	98.33	96.91	92.05	95.86
	MEMDP [J5]	<b>100</b>	<b>100</b>	98.90	98.70	99.71	96.67	96.91	93.94	96.03
	RUBIG [J4]	<b>100</b>	<b>100</b>	99.20	99.13	98.86	<b>100</b>	95.68	93.56	97.08
F	CHILOP [S2]	<b>100</b>	<b>100</b>	99.45	99.02	99.71	96.67	95.68	94.70	98.06
	LOGIC <sup>2D</sup> [S4]	<b>100</b>	<b>100</b>	99.35	99.13	99.71	98.33	95.06	95.08	<b>99.14</b>
	MSVOMF [S3]	<b>100</b>	<b>100</b>	99.35	99.35	99.71	96.67	96.30	95.08	97.87
	HoGF <sup>2D</sup> [J3]	<b>100</b>	<b>100</b>	99.20	98.91	99.71	<b>100</b>	97.53	96.59	97.19
	HoGF <sup>3D</sup> [J3]	<b>100</b>	<b>100</b>	99.25	<b>99.57</b>	99.43	98.33	98.15	97.53	97.63
	DoDGF <sup>2D</sup> [S1]	<b>100</b>	<b>100</b>	99.25	99.13	99.71	<b>100</b>	97.53	96.21	97.14
	DoDGF <sup>3D</sup> [S1]	<b>100</b>	<b>100</b>	99.55	<b>99.57</b>	99.71	<b>100</b>	98.15	96.97	97.52
	DL-PEGASOS [55]	-	97.50	95.60	-	-	-	-	-	63.70
	PI-LBP+super hist [111]	-	<b>100<sup>N</sup></b>	98.20 <sup>N</sup>	-	-	-	-	-	-
	PD-LBP+super hist [111]	-	<b>100<sup>N</sup></b>	98.10 <sup>N</sup>	-	-	-	-	-	-
	Orthogonal Tensor DL [69]	-	99.80	98.20	99.50	-	87.80	76.70	74.80	94.70
	Equiangular Kernel DL [71]	-	-	-	-	-	88.80	77.40	75.60	93.40
	SOE-Net [120]	-	-	-	-	-	96.70	95.70	92.20	94.40
	st-TCof [62]	-	-	-	-	-	<b>100*</b>	<b>100*</b>	98.11*	-
	PCANet-TOP [64]	99.50*	-	-	-	-	96.67*	90.74*	89.39*	-
	D3 [66]	-	-	-	-	-	<b>100*</b>	<b>100*</b>	98.11*	-
	DT-CNN-AlexNet [63]	-	99.50*	98.05*	98.48*	-	<b>100*</b>	99.38*	<b>99.62*</b>	98.18*
	DT-CNN-GoogleNet [63]	-	99.50*	98.35*	99.02*	-	<b>100*</b>	<b>100*</b>	<b>99.62*</b>	98.58*

Note: “-” means “not available”. Superscript “\*” indicates results using deep learning algorithms. “N” indicates rates with 1-NN classifier. 50-LOO and 50-4fold denote results on 50-class breakdown using leave-one-out and four cross-fold validation respectively. Dyn35 and Dyn++ are abbreviated for DynTex35 and DynTex++ datasets respectively. Group A is *optical-flow-based methods*, B: *model-based*, C: *geometry-based*, D: *filter-based*, E: *local-feature-based*, F: *learning-based*.



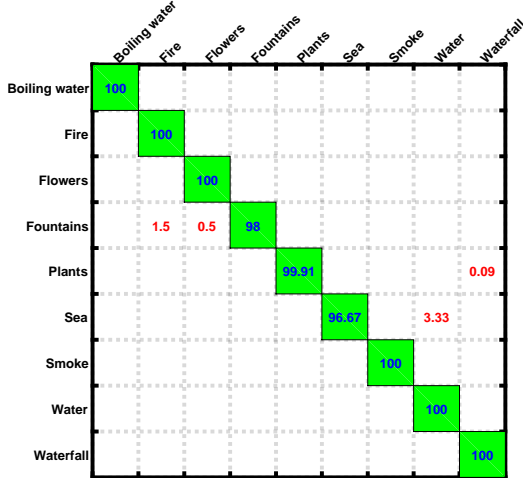


Figure 6.33: Confusion matrix (%) of  $\text{DoDGF}_{(0.7,1),\{1^{st},2^{nd}\}}^{3D}$  on 9-class.

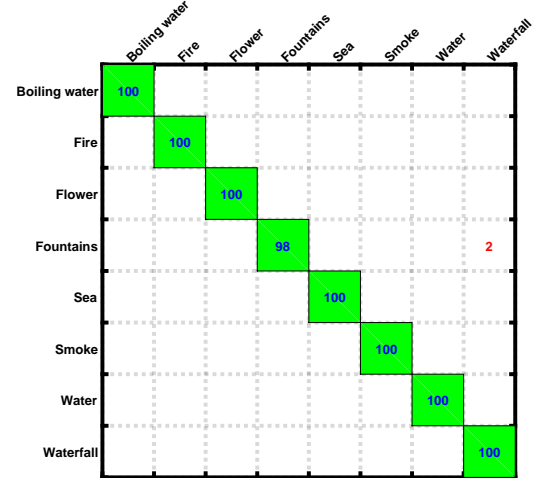


Figure 6.34: Confusion matrix (%) of  $\text{DoDGF}_{(0.7,1),\{1^{st},2^{nd}\}}^{3D}$  on 8-class.

DNGP's and FD-MAP's are not better than ours on other schemes (see Table 6.33). In the meanwhile, CVLBC [90] also obtains the nearly same performance as ours but it is not on DynTex35 and DynTex++. Also, it has not been verified on the challenging scenarios: *Alpha*, *Beta*, and *Gamma* (also see Table 6.33). In addition, it is noteworthy that our proposed others also have very good rates on UCLA such as MSVOMF [S3], CHILOP [S2], LOGIC [S4], HoGF [J3], and RUBIG [J4]. For further consideration of enhancement, we present specific confusions of  $\text{DoDGF}_{(0.7,1),\{1^{st},2^{nd}\}}^{3D}$  descriptor in DT recognition on these schemes. Accordingly, there are two categories mainly confused: “Sea” and “Water” on 9-class (see Figure 6.33), while “Fountains” and “Waterfall” on 8-class (see Figure 6.34), due to the very similar motions of DTs in those sequences.

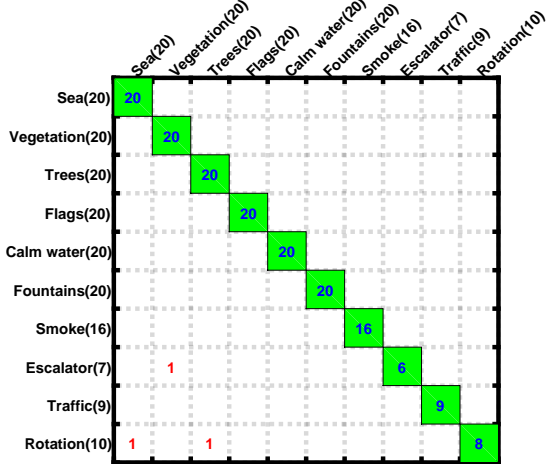
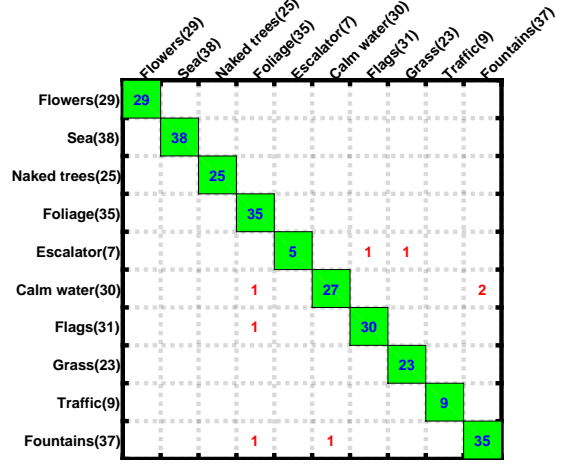
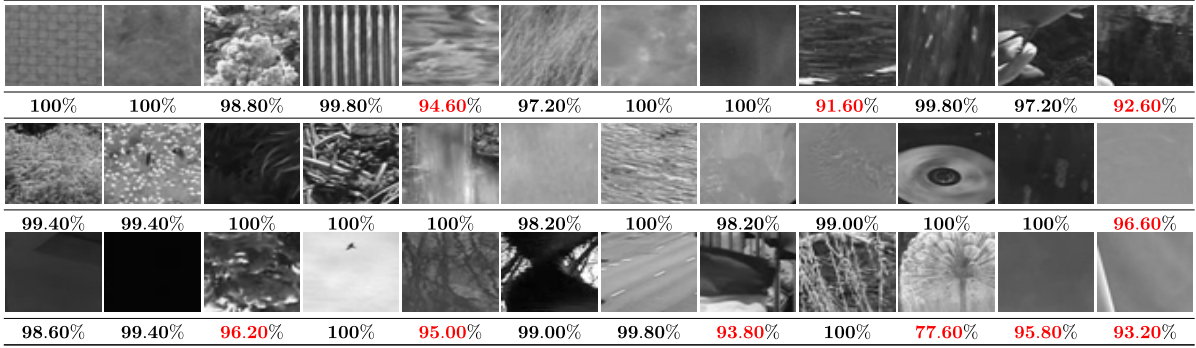
### 6.10.3.2 Classification on DynTex

It can be observed from Table 6.33 that our  $\text{DoDGF}^{2D/3D}$  descriptors mostly obtain the best rates compared to all non-deep-learning approaches, from over 1% to 3% higher improvement on the challenging schemes (i.e., *Beta* and *Gamma*) than those of MDP-based [J5], MSVOMF [S3], CHILOP [S2], LOGIC [S4], and RUBIG [J4] descriptors, very recent robust methods based on local features for DT representation. HoGF<sup>3D</sup> [J3] (9600 bins) has the same order as  $\text{DoDGF}^{3D}$  (7200 bins) on *Beta* (98.15%) and a little higher rate on *Gamma* (97.53%), but mostly not on the other schemes (see Table 6.33). In respect of comparing with deep-learning methods, with the highest rates of 100%, 100%, 98.15%, and 96.97% on *DynTex35*, *Alpha*, *Beta*, and *Gamma* respectively, these results are very closed to those of the deep-learning techniques, i.e., DT-CNN [63], st-TCof [62], and D3 [66]. It is worth noting that we just use the shallow framework for DT representation versus complicated algorithms addressed by those deep-learning models of which the deployment is restricted on mobile devices. For further consideration of improvement, we present specific confusions of  $\text{DoDGF}_{(0.7,1),\{1^{st},2^{nd}\}}^{3D}$  in DT recognition on challenging DynTex's schemes. Accordingly, DTs in “Rotation” have been mainly confused with those in “See” and “Trees” categories (see Figure 6.35), while those in “Clam water” have been mainly confused with those in “Foliage” and “Fountains” (see Figure 6.36).

### 6.10.3.3 Classification on DynTex++

Our DoDG-based descriptors have significant performance on this scheme with over 97% in 2-scale analyses of orders (see Table 6.33). For instance,  $\text{DoDGF}_{(0.7,1),\{1^{st},2^{nd}\}}^{3D}$  just obtains 97.52% due to the challenging categories highlighted in red rates in Figure 6.37. These results are the best compared to most of methods, excluding our others: LOGIC<sup>2D</sup> [S4] (99.14%), CHILOP [S2] (98.06%) as well as the



Figure 6.35: Confusion matrix (%) of  $\text{DoDGF}_{(0.7,1),\{1^{st},2^{nd}\}}^{3D}$  on *Beta*.Figure 6.36: Confusion matrix (%) of  $\text{DoDGF}_{(0.7,1),\{1^{st},2^{nd}\}}^{3D}$  on *Gamma*.Figure 6.37: The specific results of DT recognition of  $\text{DoDGF}_{(0.7,1),\{1^{st},2^{nd}\}}^{3D}$  on each category of DynTex++. The challenging categories are highlighted in red rates.

existing approaches: MEWLSP (98.48%) [95], and DT-CNN [63] (98.18% for AlexNet and 98.58% for GoogleNet frameworks). It is noteworthy that MEWLSP's performance is inferior to ours on UCLA (see Table 6.33), while not being verified on more challenging schemes, i.e., *Alpha*, *Beta*, *Gamma*.  $\text{LOGIC}^{2D}$  [S4] and CHILOP [S2] are not better than our DoDGFs on DynTex in general. In the meantime, DT-CNN [63] taking a large number of learned parameters for those frameworks just obtains about 0.5~1% higher than ours.

#### 6.10.3.4 Classification on DTDB dataset

Due to the large scale of DTDB, we utilize the best settings discussed in Section 6.9.2.2 as: high-orders of DoDG filterings  $\mathcal{F} = \{1^{st}, 2^{nd}\}$  and a pre-defined pair of standard deviations  $(\sigma, \sigma') = \{(0.7, 1)\}$  in order to structure  $\text{DoDGF}_{\sigma, \sigma', \mathcal{F}}^{2D/3D}$  descriptors. For thoroughly evaluating the effectiveness of the bipolar-filtered features compared to the Gaussian-gradient-filtered ones. The HoGF-based descriptors [J3] are also implemented using their best settings: 2-scale analysis of local neighborhoods  $\{(P, R)\} = \{(8, 1), (8, 2)\}$ , the standard deviation  $\sigma = 1$ , the 2-scale orders  $\{2^{nd}, 3^{rd}\}$  for  $\text{HoGF}^{2D}$  and  $\{3^{rd}, 4^{th}\}$  for  $\text{HoGF}^{3D}$  (refer to 6.8.3 for more detail). HILOP [C3] and its completed model, CHILOP [S2] which is based on the conventional Gaussian filtering, are also addressed using their best settings for this purpose (refer to 6.4.2 for more detail). In addition, two basic local operators, LBP-TOP [14] and CLBP [3] are also implemented in the corresponding set of multi-scale neighbors  $\{(P, R)\} = \{(8, 1), (8, 2), (8, 3)\}$  for arriving at objective evaluations in recognizing DTs on DTDB. Table 6.34 presents results of our proposed  $\text{DoDGF}_{\sigma, \sigma', \mathcal{F}}^{2D/3D}$  descriptors on two challenging subsets of

Table 6.34: Comparison of rates (%) on two challenging subsets of the large scale DTDB [4] dataset.

Group	Encoding method	$\{(P, R)\}$	Dynamics	Appearance
E	LBP-TOP <sup>u2</sup> [14]	$\{(8, 1)\}$	48.30	47.50
	LBP-TOP <sup>u2</sup> [14]	$\{(8, 1), (8, 2), (8, 3)\}$	58.05	59.07
	CLBP <sup>riu2</sup> <sub>S/M/C</sub> [3]	$\{(8, 1)\}$	60.35	60.72
	CLBP <sup>riu2</sup> <sub>S/M/C</sub> [3]	$\{(8, 1), (8, 2), (8, 3)\}$	66.56	67.06
	DoGF <sup>2D</sup> <sub>(0.7,1)</sub>	$\{(8, 1)\}$	63.27	64.14
	DoGF <sup>3D</sup> <sub>(0.7,1)</sub>	$\{(8, 1)\}$	65.07	65.11
	HILOP <sup>u2</sup> [C3]	$\{(8, 1), (8, 2), (8, 3)\}$	65.54	66.58
	CHILOP [S2]	$\{(8, 1), (8, 2), (8, 3)\}$	68.67	69.22
	HoGF <sup>3D</sup> [J3]	$\{(8, 1), (8, 2)\}$	71.08	71.03
	DoDGF <sup>3D</sup> [S1]	$\{(8, 1)\}$	72.06	72.10
F	MSOE Stream [93]	-	80.10	72.20
	SOE-Net [120]	-	<b>86.80</b>	79.00
	C3D [67]	-	74.90*	75.50*
	RGB Stream [68]	-	76.40*	76.10*
	Flow Stream [68]	-	72.60*	64.80*
	MSOE-two-Stream [4]	-	84.00*	<b>80.00*</b>

Note: “-” means “not available”. Superscript “\*” expresses results using deep learning algorithms. “<sub>S/M/C</sub>” denotes a 3D-jointed histogram of CLBP’s components. Group E denotes *local-feature-based* methods, while F: *learning-based*. Results of above learning-based methods are referred to [4].

DTDB, *Dynamics* and *Appearance*. Also, those of the other LBP-based ones and learning-based methods are expressed in this table for a purpose of comprehensive comparison. It should be noted that the performances of the learning-based methods are referred to implementations in [4].

It can be seen from Table 6.34 that our DoDG-based descriptors have performed very well in DT recognition on both *Dynamics* and *Appearance*. Those results are about 7~9% better than those of the DoG-based ones. For instance, on *Dynamics*, DoGF<sup>3D</sup><sub>(0.7,1)</sub> just obtains rate of 65.07%, inferior to ~7% compared to ours, i.e., DoDGF<sup>3D</sup> with rate of 72.06%. This has consolidated the prominent ability of DoDG filterings in noise reduction compared to the traditional DoGs. Furthermore, DoDGF (7200 bins) is also about 1% higher than HoGF (9600 bins). It means that the DoDG-based filtering is more robust in denosing than the Gaussian-gradient filtering. Exploiting the CHILOP [S2] operator to capture spatio-temporal features of the original Gaussian-filtered elements achieves the better rates compared to using the typical CLBP [3], but about 4% lower than DoDGF. It could be deduced that addressing CHILOP for the DoDG-filtered outcomes can improve the performance potentially. In terms of comparison to CLBP and LBP-TOP without addressing any filters in their encoding, our proposed descriptors based on the novel DoDGs obtain about ~12% and ~24% higher than CLBP’s and LBP-TOP’s respectively (see Table 6.34). In the meanwhile, the DoGF<sup>2D/3D</sup> descriptors based on the well-known DoGs are also ~5% and ~17% better than theirs respectively. This has proved the importance of filterings in noise reduction for DT representation, especially, the prominent contribution of our novel DoDGs.

Regarding comparison to the learning-based methods, in general, our DoDG-based descriptors have performance being very close to most of those methods, particularly, better than some of them. Indeed, with 72.10% on *Appearance*, our DoDGF<sup>3D</sup> is about 8% better than deep-learning-based Flow Stream (64.80%) [68] while being as good as learning-based MSOE Stream [93]. For DT recognition on *Dynamics*, ours (72.06%) is the same execution as that of Flow Stream [68] while being very close to that of C3D (74.90%) [67] and RGB Stream (76.40%) [68] (see Table 6.34). Furthermore, it should be pointed out that SOE-Net [120] obtains the nearly highest rates on both schemes of DTDB, but not mean that it also has the same performance on other datasets. Certainly, all SOE-Net’s performances on DynTex and DynTex++ are much lower than our DoDG-based descriptors. For instance, it could be seen from Table 6.33 that SOE-Net just obtains 96.70%, 95.70%, 92.20%, and 94.40% on *Alpha*, *Beta*, *Gamma*, and DynTex++ respectively. In the meanwhile, our DoDGF<sup>3D</sup> is 100%, 98.15%, 96.97%, and 97.52% respectively. This has restated the interest of our proposal.

Table 6.35: Performances (%) of DoDGF $^{2D/3D}_{\{(\sigma, \sigma')\}, \mathcal{F}}$  in further scale analysis.

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

## 6.11 Global discussions

### 6.11.1 Further evaluations for Gaussian-gradient-based descriptors

As mentioned above, our Gaussian-gradient-based descriptors, DoDGF [S1] and HoGF [J3], could be two best descriptors with high performance in DT recognition, expected as appreciated solutions for embedded applications which have been required to execute their functions in restricted resources. In addition to thorough evaluations discussed in Section 6.10, it can be asserted their properties in further contexts based on more experimental results as follows.

- The experimental results, presented in Sections 6.8.3 and 6.9.2, have verified that the 3D Gaussian-gradient filterings are better than the 2D ones in most cases. It may be deduced that addressing the higher directions of these kernels can improve the performance. On the other word, addressing jointly shape and motion cues based on the 3D filterings is more effective than a separate consideration in the 2D ones.
- Taking multi-scale of  $\{(\sigma, \sigma')\}$  into account the DT encoding does not make the DoDG-based descriptors more robust, except 97.35%, a little higher rate on *Gamma* of DoDGF $^{3D}_{\{(0.5,1),(0.7,1),(1,1.3)\}, \{1^{st}\}}$  (see Table 6.35(a)). This is agreed with the HoGF-based descriptors while their dimension increases up to 14000 bins (see Table 6.36(a)).
- Also, addressing multi-scale of both high-order kernels of DoDG and Gaussian-gradient filterings is not for further enhancement of the correspondingly obtained descriptors while its dimension grows up sharply (see Tables 6.35(b) for DoDGF and 6.36(b) for HoGF).
- Combining an odd order and an even one often obtains better performances than other configurations since this addresses two complementary kinds of Gaussian-gradients.
- In addition, combining two kinds of above multi-scale analyses obtains a better rate of 97.73% on *Gamma* for the DoDGF $^{3D}_{\{(\sigma, \sigma')\}, \mathcal{F}}$ , while facing with the curse of larger dimensions, up to 21600 bins, (see Table 6.35(c)). For the HoGF descriptors, taking the Gaussian-gradient filterings into account multi-analysis of different orders and deviations leads to larger dimensions, e.g., HoGF $^{2D/3D}_{\{2^{nd}, 3^{rd}\}, \{0.5, 1\}}$ , but not boost the performance in DT classification. In case of full-scale of all those, i.e., HoGF $^{2D/3D}_{\{1^{st}, 2^{nd}, 3^{rd}, 4^{th}\}, \{0.5, 0.7, 1\}}$ , the performance is mostly not improved, except a little higher rate ( $\approx 98\%$ ) on DynTex++, while facing with the curse of dimension: 43200 bins for HoGF $^{2D}$  and 57600 bins for HoGF $^{3D}$  (see Table 6.36(c)). Due to above problems, those combinations should not be recommended for real applications.

Table 6.36: Performances (%) of HoGF<sup>2D/3D</sup> in further scale analysis.

	Descriptor	#bins	Dyn35	Beta	Gamma	Dyn++
(a)	HoGF <sup>2D</sup> <sub>{1<sup>st</sup>}, {0.5, 0.7, 1}</sub>	10800	99.14	95.68	96.21	97.48
	HoGF <sup>2D</sup> <sub>{2<sup>nd</sup>}, {0.5, 0.7, 1}</sub>	10800	<b>100</b>	95.06	94.70	97.34
	HoGF <sup>2D</sup> <sub>{3<sup>rd</sup>}, {0.5, 0.7, 1}</sub>	10800	99.14	96.30	96.21	97.24
	HoGF <sup>2D</sup> <sub>{4<sup>th</sup>}, {0.5, 0.7, 1}</sub>	10800	99.43	95.06	93.18	97.14
	HoGF <sup>3D</sup> <sub>{1<sup>st</sup>}, {0.5, 0.7, 1}</sub>	14400	99.43	96.30	96.59	97.71
	HoGF <sup>3D</sup> <sub>{2<sup>nd</sup>}, {0.5, 0.7, 1}</sub>	14400	<b>100</b>	96.30	95.08	97.93
	HoGF <sup>3D</sup> <sub>{3<sup>rd</sup>}, {0.5, 0.7, 1}</sub>	14400	99.14	96.91	95.83	97.47
	HoGF <sup>3D</sup> <sub>{4<sup>th</sup>}, {0.5, 0.7, 1}</sub>	14400	99.43	95.69	93.94	97.14
(b)	HoGF <sup>2D</sup> <sub>{1<sup>st</sup>, 2<sup>nd</sup>, 3<sup>rd</sup>}, {1}</sub>	10800	99.14	96.91	96.21	97.24
	HoGF <sup>2D</sup> <sub>{1<sup>st</sup>, 2<sup>nd</sup>, 3<sup>rd</sup>, 4<sup>th</sup>}, {1}</sub>	14400	<b>100</b>	97.53	96.59	97.41
	HoGF <sup>3D</sup> <sub>{1<sup>st</sup>, 2<sup>nd</sup>, 3<sup>rd</sup>}, {1}</sub>	14400	99.43	97.53	<b>97.35</b>	97.82
	HoGF <sup>3D</sup> <sub>{1<sup>st</sup>, 2<sup>nd</sup>, 3<sup>rd</sup>, 4<sup>th</sup>}, {1}</sub>	19200	<b>100</b>	<b>98.15</b>	<b>97.35</b>	97.61
(c)	HoGF <sup>2D</sup> <sub>{2<sup>nd</sup>, 3<sup>rd</sup>}, {0.5, 1}</sub>	14400	99.71	96.91	95.45	97.67
	HoGF <sup>2D</sup> <sub>{3<sup>rd</sup>, 4<sup>th</sup>}, {0.5, 1}</sub>	14400	99.71	96.91	95.08	97.62
	HoGF <sup>2D</sup> <sub>{1<sup>st</sup>, 2<sup>nd</sup>, 3<sup>rd</sup>, 4<sup>th</sup>}, {0.5, 0.7, 1}</sub>	43200	99.71	96.30	95.45	98.03
	HoGF <sup>3D</sup> <sub>{2<sup>nd</sup>, 3<sup>rd</sup>}, {0.5, 1}</sub>	19200	99.71	<b>98.15</b>	95.45	97.87
	HoGF <sup>3D</sup> <sub>{3<sup>rd</sup>, 4<sup>th</sup>}, {0.5, 1}</sub>	19200	<b>100</b>	<b>98.15</b>	95.83	97.80
	HoGF <sup>3D</sup> <sub>{1<sup>st</sup>, 2<sup>nd</sup>, 3<sup>rd</sup>, 4<sup>th</sup>}, {0.5, 0.7, 1}</sub>	57600	<b>100</b>	<b>98.15</b>	95.45	<b>98.06</b>

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

### 6.11.2 Evaluating appropriation of our proposals for real applications

Currently, deep-learning-based methods are going on the major stream for computer vision community. They often obtain significant results in DT recognition (see Tables 6.33 and 6.34). However, it takes much time for them to learn millions of parameters using complex learning algorithms in multi-deep-layer networks. For instance, it takes  $\sim 80\text{M}$  for C3D [67],  $\sim 88\text{M}$  for MSOE-two-Stream [4], while  $\sim 61\text{M}$  for AlexNet and  $\sim 6.8\text{M}$  for GoogleNet for DT-CNN [63]. This is one of crucial barriers in order to bring those into real applications for mobile devices as well as embedded sensor systems, those which have strictly required tiny resources for their functions.

In this chapter, our proposed framework can mitigate those shortcomings in low computational complexity, expected to be potential for mobile implementations. It just utilizes a simple operator to capture spatio-temporal features from the DoDG-filtered outcomes that are pointed out by the novel DoDG filtering, proved to be much better than the well-known DoG one and others in denoising. In slight dimension, the obtained DoDG-based descriptors DoDGF<sup>2D/3D</sup> <sub>$\sigma, \sigma', \mathcal{F}$</sub>  have the highest performances compared to all non-deep-learning methods, while being close to those of deep-learning ones. Indeed, Tables 6.33 and 6.34 show the very good performances of our 2-order descriptor DoDGF<sup>3D</sup><sub>(0.7, 1), {1<sup>st</sup>, 2<sup>nd</sup>}</sub> with 7200 bins as well as those of the single order DoDGF<sup>2D</sup><sub>(0.7, 1), {1<sup>st</sup>}</sub> with only 2400 bins. Those can be easily applied on edge devices while maintaining a comparable performance related to deep learning models. In addition, instead of using CLBP [3], it is able to take other LBP-based operators into account our proposed framework for a purpose of further enhancement, e.g., CLBC [82], LDP-based [30, J5], LVP-based [100, J2], LRP [J4], MRELBP [78], etc.

## 6.12 Summary

In this chapter, taking advantage of our local robust operators proposed in Chapter 3, we have proposed several efficient frameworks for DT representation, which are based on local feature extraction from filtered outcomes computed by the conventional Gaussian filtering and its variants: DoG, gradients,

the novel DoDG kernel. Just using a shallow analysis to represent DTs, we have effectively constructed discriminative descriptors with very good performance in comparison with state of the art. Among of our proposed descriptors, the experiments have indicated that those based on the Gaussian-gradient and DoDG filterings, i.e., HoGF [J3] and DoDGF [S1], mostly obtain the best performance compared to the others. Those are also recommended for mobile systems due to their simple computations and small dimension. In case of dealing with the curse of large dimension, DoDGF's analysis in multi-scale solutions of supporting regions (e.g.,  $\{(P, R)\} = \{(8, 1), (8, 2), (8, 3)\}$ ) can be considered in future works to capture more extensively local relationships for further improvement. In addition, motivated by the approach of LOGIC [S4], the bipolar properties of Gaussian-gradient filtered outcomes have been exploited in the our latest work [S5] to address a potential alternative solution for HoGF [J3].

---

---

# CHAPTER 7

---

## CONCLUSIONS AND PERSPECTIVES

### Contents

---

7.1 Conclusions . . . . .	141
7.2 Perspectives . . . . .	142

---

### 7.1 Conclusions

The thesis has concentrated on how to efficiently represent DTs for recognition issue. To this end, we have proposed two main streams of techniques to deal with the well-known problems (e.g., noise, changes of environment, illumination, scales, etc.) which negatively impact on capturing turbulent characteristics for DT representation. According to the proposals and their experiments for DT recognition task, which have been done in the above chapters, it could be stated several conclusions as follows.

- In the thesis, we have introduced several discriminative operators for the local encoding which are adopted to different contexts of DT representation. More concretely, the directional-based local patterns, xLVP and xLDP, are more discrimination power compared to the originals, LVP [100] and LDP [30] respectively. CAIP could be a suitable alternative for the popular local operator, CLBP [3], in order to fix the close-to-zero problem caused by the separately bipolar features. In the meanwhile, the experiments have also proved the important contributions of the LRP and CHILOP operators in extracting spatio-temporal patterns from the Gaussian-filtered outcomes, compared to those which have been done by CLBP [3].
- We have proposed an efficient framework to exploit local features for DT description based on dense trajectories extracted from a given video, instead of taking into account the entire video. The descriptor construction is an analysis of directional features of beam trajectories which are combined with those of motion points captured along the path of the corresponding trajectories. The experiments have validated the good performance in recognizing DTs, compared to state of the art. Subject to types of vision implementations in practice, addressing the length of dense trajectory  $L$  can momentarily affects the execution of structuring the corresponding descriptor as well as its performance. For instance, due to the short “living” of DTs in videos, the higher value of  $L$  is addressed, the smaller number of dense trajectories is extracted. This leads to less spatio-temporal features taken into account the encoding. In addition, the proposed framework also depends on which turbulent levels of DTs are recorded in videos, as an example shown in Table 6.32.
- We have introduced a novel filtering model of moment volumes, which is motivated by the moment images [2]. After that, we have proposed two corresponding frameworks in order to take both of

them into account video analysis to obtain robust filtered outcomes for DT representation. The experiments have proved the eminent performance of our proposals in comparison with the existing methods. Therein, the moment-volume model is more robustness than the moment-image one and should be applied to further implementations in practice.

- Also, we have proposed various flowcharts to take the Gaussian-based kernel and variants of high-order Gaussian gradients into account the video filtering. Many robust descriptors have constructed by using local operators to encode the obtained filtering responses. Specifically, in regard to the Gaussian-based kernel, we have LOGIC descriptor using CAIP operator, RUBIG using LRP, CHILOP<sup>G<sub>F</sub></sup> descriptor using CHILOP. Meantime, based on the variants of high-order Gaussian gradients, we have HoGF and IOM/VOM-based using CLBP [3] to encode the obtained Gaussian-gradient-based outcomes. Prominently, we have introduced a novel DoDG filtering kernel in consideration of the difference of Gaussian gradients, which allows to point out the outstanding DoDG-filtered outcomes. The eminent DoDG-based descriptors are structured by using a simple local encoding of CLBP [3]. The experiments in DT recognition have shown the significant performance of these proposed descriptors in comparison with the current approaches. Therein, the DoDGF and HoGF descriptors have the best performance compared to all non-deep-learning models, while being close to that of the deep-learning approaches. Particularly, in small dimension, the DoDG-based descriptors could be expected as appreciate solutions for mobile applications as well as embedded sensor systems, those which require restricted resources for their functions.

## 7.2 Perspectives

In the thesis, various techniques have been introduced for efficiently describing DTs. In future works, it can be in consideration of the following improvements:

- It can be seen that our proposed operators, xLVP, xLDP, CAIP, and CHILOP, can be applied to other local encodings in applications of computer vision which are related to both video representation and still image description. In the meantime, LRP prefers to video analysis because its principle relies on local neighbors interpolated by a cube shape centering at a voxel. Also, it should be noted that problems of the curse dimension can be seriously raised in real implementations if LRP and CHILOP are located in higher multi-scale analysis of supporting regions. In addition, it can address xLVP and xLDP in full directions, higher orders, or both of them to investigate more directional relationships for enhancing the discrimination of the obtained patterns. Like CAIP (an adaptation of CLBP [3]), the operators, xLVP, xLDP, CHILOP, and LRP, can be extended to deal with the close-to-zero problem and applied to encoding the bipolar filtered features. They have been considered as very potential solutions since the experiments have validated their better performance in comparison with that of the original CLBP [3].
- Exploiting the short length of dense trajectories for DT representation can enrich more informative patterns. However, due to the curse of a grand number of extracted trajectories, the speed of the computation should be considered in real applications. Instead of using xLVP, addressing others (e.g., xLDP, LRP, etc.) to encode the features of dense trajectories may enhance the performance.
- In consideration of treating the large dimension problem, addressing the moment volumes in higher orders may obtain more robust filtered outcomes for capturing spatio-temporal relationships to boost the discrimination power. Furthermore, xLVP, CHILOP, and LRP can be potential alternatives to encode the moment-filtered volumes for further enhancement.
- As mentioned in Chapter 6, most of the proposed descriptors have been constructed by using CLBP [3] to capture spatio-temporal features from filtered outcomes which are extracted by the Gaussian kernel and variants of Gaussian gradients. Our proposed descriptors (e.g., xLVP, xLDP, CHILOP, etc.) could be applied to those local encodings for further improvement.

---

# BIBLIOGRAPHY

- [1] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Conference on Neural Information Processing Systems (NIPS), 2012, pp. 1106–1114.
- [2] T. P. Nguyen, N. Vu, A. Manzanera, Statistical binary patterns for rotational invariant texture classification, *Neurocomputing* 173 (2016) 1565–1577.
- [3] Z. Guo, L. Zhang, D. Zhang, A completed modeling of local binary pattern operator for texture classification, *IEEE Transactions on Image Processing* 19 (6) (2010) 1657–1663.
- [4] I. Hadji, R. P. Wildes, A new large scale dynamic texture dataset with application to convnet understanding, in: European Conference on Computer Vision (ECCV), 2018, pp. 334–351.
- [5] P. Saisan, G. Doretto, Y. N. Wu, S. Soatto, Dynamic texture recognition, in: Conference on Computer Vision and Pattern Recognition (CVPR), 2001, pp. 58–63.
- [6] Y. Li, W. Liu, Q. Huang, Traffic anomaly detection based on image descriptor in videos, *Multimedia Tools and Applications* 75 (5) (2016) 2487–2505.
- [7] Y. Tang, C. Zhang, R. Gu, P. Li, B. Yang, Vehicle detection and recognition for intelligent traffic surveillance system, *Multimedia Tools and Applications* 76 (4) (2017) 5817–5832.
- [8] W. Zeng, C. Xie, Z. Yang, X. Lu, A universal sample-based background subtraction method for traffic surveillance videos, *Multimedia Tools and Applications* 79 (31-32) (2020) 22211–22234.
- [9] A. B. Chan, N. Vasconcelos, Modeling, clustering, and segmenting video with mixtures of dynamic textures, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30 (5) (2008) 909–926.
- [10] J. Cui, H. Zha, H. Zhao, R. Shibasaki, Laser-based detection and tracking of multiple people in crowds, *Computer Vision and Image Understanding* 106 (2-3) (2007) 300–312.
- [11] T. Germa, F. Lerasle, N. Ouadah, V. Cadenat, Vision and RFID data fusion for tracking people in crowds by a mobile robot, *Computer Vision and Image Understanding* 114 (6) (2010) 641–651.
- [12] A. Pennisi, D. D. Bloisi, L. Iocchi, Online real-time crowd behavior detection in video sequences, *Computer Vision and Image Understanding* 144 (2016) 166–176.
- [13] X. S. Nguyen, T. P. Nguyen, F. Charpillet, N.-S. Vu, Local derivative pattern for action recognition in depth images, *Multimedia Tools and Applications* 77 (7) (2018) 8531–8549.
- [14] G. Zhao, M. Pietikäinen, Dynamic texture recognition using local binary patterns with an application to facial expressions, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (6) (2007) 915–928.
- [15] W. Zhang, M. L. Smith, L. N. Smith, A. R. Farooq, Gender and gaze gesture recognition for human-computer interaction, *Computer Vision and Image Understanding* 149 (2016) 32–50.
- [16] A. I. Maqueda, C. R. del-Blanco, F. Jaureguizar, N. N. García, Human-computer interaction based on visual hand-gesture recognition using volumetric spatiograms of local binary patterns, *Computer Vision and Image Understanding* 141 (2015) 126–137.



- [17] P. Barmpoutis, K. Dimitropoulos, N. Grammalidis, Smoke detection using spatio-temporal analysis, motion modeling and dynamic texture recognition, in: *European Signal Processing Conference (EUSIPCO)*, 2014, pp. 1078–1082.
- [18] P. Mettes, R. T. Tan, R. C. Veltkamp, Water detection through spatio-temporal invariant descriptors, *Computer Vision and Image Understanding* 154 (2017) 182–191.
- [19] X. Wu, X. Lu, H. Leung, Video smoke separation and detection via sparse representation, *Neurocomputing* 360 (2019) 61–74.
- [20] C. Zhang, F. Zhou, B. Xue, W. Xue, Stabilization of atmospheric turbulence-distorted video containing moving objects using the monogenic signal, *Signal Processing: Image Communication* 63 (2018) 19–29.
- [21] Dynamic texture based smoke detection using surfacelet transform and {HMT} model, *Fire Safety Journal* 73 (2015) 91–101.
- [22] T. P. Nguyen, A. Manzanera, M. Garrigues, N. Vu, Spatial motion patterns: Action models from semi-dense trajectories, *International Journal of Pattern Recognition and Artificial Intelligence* 28 (7) (2014).
- [23] O. J. Makhura, J. C. Woods, Learn-select-track: An approach to multi-object tracking, *Signal Processing: Image Communication* 74 (2019) 153–161.
- [24] D. Ortego, J. C. SanMiguel, J. M. Martínez, Stand-alone quality estimation of background subtraction algorithms, *Computer Vision and Image Understanding* 162 (2017) 87–102.
- [25] Z. Zeng, J. Jia, Z. Zhu, D. Yu, Adaptive maintenance scheme for codebook-based dynamic background subtraction, *Computer Vision and Image Understanding* 152 (2016) 58–66.
- [26] D. Zamaliev, A. Yilmaz, Background subtraction for the moving camera: A geometric approach, *Computer Vision and Image Understanding* 127 (2014) 73–85.
- [27] H. Sajid, S. S. Cheung, N. Jacobs, Motion and appearance based background subtraction for freely moving cameras, *Signal Processing: Image Communication* 75 (2019) 11–21.
- [28] Z. Xu, B. Min, R. C. C. Cheung, A robust background initialization algorithm with superpixel motion detection, *Signal Processing: Image Communication* 71 (2019) 1–12.
- [29] N. Shrivastava, V. Tyagi, An effective scheme for image texture classification based on binary local structure pattern, *The Visual Computer* 30 (11) (2014) 1223–1232.
- [30] B. Zhang, Y. Gao, S. Zhao, J. Liu, Local derivative pattern versus local binary pattern: Face recognition with high-order local pattern descriptor, *IEEE Transactions on Image Processing* 19 (2) (2010) 533–544.
- [31] A. K. Jain, F. Farrokhnia, Unsupervised texture segmentation using gabor filters, *Pattern Recognition* 24 (12) (1991) 1167–1186.
- [32] B. K. P. Horn, B. G. Schunck, Determining optical flow, *Artificial Intelligence* 17 (1-3) (1981) 185–203.
- [33] J. L. Barron, D. J. Fleet, S. S. Beauchemin, Performance of optical flow techniques, *International Journal of Computer Vision* 12 (1) (1994) 43–77.
- [34] C. Peh, L. F. Cheong, Synergizing spatial and temporal texture, *IEEE Transactions on Image Processing* 11 (10) (2002) 1179–1191.
- [35] R. Péteri, D. Chetverikov, Qualitative characterization of dynamic textures for video retrieval, in: *International Conference on Computer Vision and Graphics (ICCVG)*, 2004, pp. 33–38.
- [36] R. Péteri, D. Chetverikov, Dynamic texture recognition using normal flow and texture regularity, in: *Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*, 2005, pp. 223–230.
- [37] Z. Lu, W. Xie, J. Pei, J. Huang, Dynamic texture recognition by spatio-temporal multiresolution histograms, in: *Workshop on Applications of Computer Vision/Workshop on Motion and Video Computing (WACV/MOTION)*, 2005, pp. 241–246.

- 
- [38] A. R. Rivera, O. Chae, Spatiotemporal directional number transitional graph for dynamic texture recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37 (10) (2015) 2146–2152.
  - [39] G. Doretto, A. Chiuso, Y. N. Wu, S. Soatto, Dynamic textures, *International Journal of Computer Vision* 51 (2) (2003) 91–109.
  - [40] A. B. Chan, N. Vasconcelos, Classifying video with kernel dynamic textures, in: *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007, pp. 1–6.
  - [41] A. Mumtaz, E. Coviello, G. R. G. Lanckriet, A. B. Chan, Clustering dynamic textures with the hierarchical EM algorithm for modeling video, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (7) (2013) 1606–1621.
  - [42] Y. Wang, S. Hu, Chaotic features for dynamic textures recognition, *Soft Computing* 20 (5) (2016) 1977–1989.
  - [43] A. Ravichandran, R. Chaudhry, R. Vidal, View-invariant dynamic texture recognition using a bag of dynamical systems, in: *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 1651–1657.
  - [44] A. Mumtaz, E. Coviello, G. R. G. Lanckriet, A. B. Chan, A scalable and accurate descriptor for dynamic textures using bag of system trees, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37 (4) (2015) 697–712.
  - [45] X. Wei, Y. Li, H. Shen, F. Chen, M. Kleinsteuber, Z. Wang, Dynamical textures modeling via joint video dictionary learning, *IEEE Transactions on Image Processing* 26 (6) (2017) 2929–2943.
  - [46] B. Mandelbrot, *Fractal Geometry of Nature*, W. H. Freeman, 1977.
  - [47] K. Falconer, *Techniques in fractal geometry*, John Wiley, 1997.
  - [48] Y. Xu, H. Ji, C. Fermüller, Viewpoint invariant texture description using fractal analysis, *International Journal of Computer Vision* 83 (1) (2009) 85–100.
  - [49] Y. Xu, Y. Quan, H. Ling, H. Ji, Dynamic texture classification using dynamic fractal analysis, in: *International Conference on Computational Vision (ICCV)*, 2011, pp. 1219–1226.
  - [50] Y. Xu, Y. Quan, Z. Zhang, H. Ling, H. Ji, Classifying dynamic textures via spatiotemporal fractal analysis, *Pattern Recognition* 48 (10) (2015) 3239–3248.
  - [51] Y. Xu, S. B. Huang, H. Ji, C. Fermüller, Scale-space texture description on sift-like textons, *Computer Vision and Image Understanding* 116 (9) (2012) 999–1013.
  - [52] H. Ji, X. Yang, H. Ling, Y. Xu, Wavelet domain multifractal analysis for static and dynamic texture classification, *IEEE Transactions on Image Processing* 22 (1) (2013) 286–299.
  - [53] Y. Quan, Y. Sun, Y. Xu, Spatiotemporal lacunarity spectrum for dynamic texture classification, *Computer Vision and Image Understanding* 165 (2017) 85–96.
  - [54] R. Péteri, S. Fazekas, M. J. Huiskes, Dyntex: A comprehensive database of dynamic textures, *Pattern Recognition Letters* 31 (12) (2010) 1627–1632.
  - [55] B. Ghanem, N. Ahuja, Maximum margin distance learning for dynamic texture recognition, in: *European Conference on Computer Vision (ECCV)*, 2010, pp. 223–236.
  - [56] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner., Gradient-based learning applied to document recognition, *Proceedings of the IEEE* (1998) 2278–2324.
  - [57] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, L. D. Jackel, Backpropagation applied to handwritten zip code recognition, *Neural Computation* 1 (4) (1989) 541–551.
  - [58] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: *European Conference on Computer Vision (ECCV)*, 2014, pp. 818–833.

- [59] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9.
- [60] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations (ICLR), 2015.
- [61] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
- [62] X. Qi, C. Li, G. Zhao, X. Hong, M. Pietikäinen, Dynamic texture and scene classification by transferring deep image features, *Neurocomputing* 171 (2016) 1230–1241.
- [63] V. Andrearczyk, P. F. Whelan, Convolutional neural network on three orthogonal planes for dynamic texture classification, *Pattern Recognition* 76 (2018) 36–49.
- [64] S. R. Arashloo, M. C. Amirani, A. Noroozi, Dynamic texture representation using a deep multi-scale convolutional network, *Journal of Visual Communication and Image Representation* 43 (2017) 89–97.
- [65] T. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, Y. Ma, Pcanet: A simple deep learning baseline for image classification?, *IEEE Transactions on Image Processing* 24 (12) (2015) 5017–5032.
- [66] S. Hong, J. Ryu, W. Im, H. S. Yang, D3: recognizing dynamic scenes with deep dual descriptor based on key frames and key segments, *Neurocomputing* 273 (2018) 611–621.
- [67] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3d convolutional networks, in: International Conference on Computational Vision (ICCV), 2015, pp. 4489–4497.
- [68] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Conference on Neural Information Processing Systems (NIPS), 2014, pp. 568–576.
- [69] Y. Quan, Y. Huang, H. Ji, Dynamic texture recognition via orthogonal tensor dictionary learning, in: International Conference on Computational Vision (ICCV), 2015, pp. 73–81.
- [70] M. Aharon, M. Elad, A. M. Bruckstein, K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation, *IEEE Transactions on Signal Processing* 54 (11) (2006) 4311–4322.
- [71] Y. Quan, C. Bao, H. Ji, Equiangular kernel dictionary learning with applications to dynamic texture analysis, in: Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 308–316.
- [72] S. R. Arashloo, J. Kittler, Dynamic texture recognition using multiscale binarized statistical image features, *IEEE Transactions on Multimedia* 16 (8) (2014) 2099–2109.
- [73] J. Kannala, E. Rahtu, BSIF: binarized statistical image features, in: International Conference on Pattern Recognition (ICPR), 2012, pp. 1363–1366.
- [74] X. Zhao, Y. Lin, L. Liu, J. Heikkilä, W. Zheng, Dynamic texture classification using unsupervised 3d filter learning and local binary encoding, *IEEE Transactions on Multimedia* 21 (7) (2019) 1694–1708.
- [75] X. Zhao, Y. Lin, J. Heikkilä, Dynamic texture recognition using multiscale pca-learned filters, in: International Conference on Image Processing (ICIP), 2017, pp. 4152–4156.
- [76] J. Ngiam, P. W. Koh, Z. Chen, S. A. Bhaskar, A. Y. Ng, Sparse filtering, in: Conference on Neural Information Processing Systems (NIPS), 2011, pp. 1125–1133.
- [77] A. Coates, A. Y. Ng, Learning feature representations with k-means, in: G. Montavon, G. B. Orr, K. Müller (Eds.), *Neural Networks: Tricks of the Trade - Second Edition*, 2012, pp. 561–580.
- [78] L. Liu, S. Lao, P. W. Fieguth, Y. Guo, X. Wang, M. Pietikäinen, Median robust extended local binary pattern for texture classification, *IEEE Transactions on Image Processing* 25 (3) (2016) 1368–1381.
- [79] M. Alkhatib, A. Hafiane, Robust adaptive median binary pattern for noisy texture classification and retrieval, *IEEE Transactions on Image Processing* 28 (11) (2019) 5407–5418.

- 
- [80] R. A. Kirsch, Computer determination of the constituent structure of biological images, *Computers and Biomedical Research* 4 (3) (1971) 315–328.
  - [81] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (7) (2002) 971–987.
  - [82] Y. Zhao, D.-S. Huang, W. Jia, Completed Local Binary Count for Rotation Invariant Texture Classification, *IEEE Transactions on Image Processing* 21 (10) (2012) 4492–4497.
  - [83] A. Fathi, A. R. Naghsh-Nilchi, Noise Tolerant Local Binary Pattern Operator for Efficient Texture Analysis, *Pattern Recognition Letters* 33 (9) (2012) 1093–1100.
  - [84] T. P. Nguyen, A. Manzanera, W. G. Kropatsch, X. S. N’Guyen, Topological attribute patterns for texture recognition, *Pattern Recognition Letters* 80 (2016) 91–97.
  - [85] L. Liu, J. Chen, P. W. Fieguth, G. Zhao, R. Chellappa, M. Pietikäinen, From bow to CNN: two decades of texture representation for texture classification, *International Journal of Computer Vision* 127 (1) (2019) 74–109.
  - [86] L. Liu, P. W. Fieguth, Y. Guo, X. Wang, M. Pietikäinen, Local binary features for texture classification: Taxonomy and experimental study, *Pattern Recognition* 62 (2017) 135–160.
  - [87] L. Liu, Y. Long, P. W. Fieguth, S. Lao, G. Zhao, BRINT: binary rotation invariant and noise tolerant texture classification, *IEEE Transactions on Image Processing* 23 (7) (2014) 3071–3084.
  - [88] L. Liu, L. Zhao, Y. Long, G. Kuang, P. W. Fieguth, Extended local binary patterns for texture classification, *Image and Vision Computing* 30 (2) (2012) 86–99.
  - [89] Z. Guo, X. Wang, J. Zhou, J. You, Robust texture image representation by scale selective local binary patterns, *IEEE Transactions on Image Processing* 25 (2) (2016) 687–699.
  - [90] X. Zhao, Y. Lin, J. Heikkilä, Dynamic texture recognition using volume local binary count patterns with an application to 2d face spoofing detection, *IEEE Transactions on Multimedia* 20 (3) (2018) 552–566.
  - [91] D. Tiwari, V. Tyagi, Dynamic texture recognition based on completed volume local binary pattern, *Multidimensional Systems and Signal Processing* 27 (2) (2016) 563–575.
  - [92] D. Tiwari, V. Tyagi, A novel scheme based on local binary pattern for dynamic texture recognition, *Computer Vision and Image Understanding* 150 (2016) 58–65.
  - [93] K. G. Derpanis, R. P. Wildes, Spacetime texture representation and recognition based on a spatiotemporal orientation analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34 (6) (2012) 1193–1205.
  - [94] S. Dubois, R. Péteri, M. Ménard, Characterization and recognition of dynamic textures based on the 2d+t curvelet transform, *Signal, Image and Video Processing* 9 (4) (2015) 819–830.
  - [95] D. Tiwari, V. Tyagi, Dynamic texture recognition using multiresolution edge-weighted local structure pattern, *Computers & Electrical Engineering* 62 (2017) 485–498.
  - [96] E. Fix, J. Hodges, Discriminatory analysis, nonparametric discrimination: Consistency properties, Technical report 4, USAF School of Aviation Medicine, Randolph Field (1951).
  - [97] B. E. Boser, I. Guyon, V. Vapnik, A training algorithm for optimal margin classifiers, in: *ACM Conference on Computational Learning Theory, (COLT)*, 1992, pp. 144–152.
  - [98] C. Cortes, V. Vapnik, Support-vector networks, *Machine Learning* 20 (3) (1995) 273–297.
  - [99] R. Fan, K. Chang, C. Hsieh, X. Wang, C. Lin, LIBLINEAR: A library for large linear classification, *Journal of Machine Learning Research* 9 (2008) 1871–1874.
  - [100] K. Fan, T. Hung, A novel local pattern descriptor - local vector pattern in high-order derivative space for face recognition, *IEEE Transactions on Image Processing* 23 (7) (2014) 2877–2891.
  - [101] H. Jin, Q. Liu, H. Lu, X. Tong, Face detection using improved LBP under bayesian framework, in: *International Conference on Image and Graphics (ICIG)*, 2004, pp. 306–309.

- [102] X. S. Nguyen, A. Mouaddib, T. P. Nguyen, L. Jeanpierre, Action recognition in depth videos using hierarchical gaussian descriptor, *Multimedia Tools and Applications* 77 (16) (2018) 21617–21652.
- [103] J. B. Naik, C. Srinivasarao, G. B. Kande, Local vector pattern with global index angles for a content-based image retrieval system, *Journal of the Association for Information Science and Technology* 68 (12) (2017) 2755–2770.
- [104] H. Wang, A. Kläser, C. Schmid, C. Liu, Dense trajectories and motion boundary descriptors for action recognition, *International Journal of Computer Vision* 103 (1) (2013) 60–79.
- [105] G. Farnebäck, Two-frame motion estimation based on polynomial expansion, in: *Scandinavian Conference on Image Analysis (SCIA)*, 2003, pp. 363–370.
- [106] S. Mukherjee, K. K. Singh, Human action and event recognition using a novel descriptor based on improved dense trajectories, *Multimedia Tools and Applications* 77 (11) (2018) 13661–13678.
- [107] L. Chen, J. Shen, W. Wang, B. Ni, Video object segmentation via dense trajectories, *IEEE Transactions on Multimedia* 17 (12) (2015) 2225–2234.
- [108] T. Mäenpää, M. Pietikäinen, Multi-scale binary patterns for texture analysis, in: *Scandinavian Conference on Image Analysis (SCIA)*, 2003, pp. 885–892.
- [109] D. Tiwari, V. Tyagi, Improved weber’s law based local binary pattern for dynamic texture recognition, *Multimedia Tools and Applications* 76 (5) (2017) 6623–6640.
- [110] L. C. Ribas, W. N. Gonçalves, O. M. Bruno, Dynamic texture analysis with diffusion in networks, *Digital Signal Processing* 92 (2019) 109–126.
- [111] J. Ren, X. Jiang, J. Yuan, Dynamic texture recognition using enhanced LBP features, in: *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 2400–2404.
- [112] J. J. de Mesquita Sá Junior, L. C. Ribas, O. M. Bruno, Randomized neural network based signature for dynamic texture classification, *Expert Systems with Applications* 135 (2019) 194–200.
- [113] J. Ren, X. Jiang, J. Yuan, G. Wang, Optimizing LBP structure for visual recognition using binary quadratic programming, *IEEE Signal Processing Letters* 21 (11) (2014) 1346–1350.
- [114] M. Baktashmotlagh, M. T. Harandi, A. , B. C. C. Lovell, M. Salzmann, Discriminative non-linear stationary subspace analysis for video classification, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36 (12) (2014) 2353–2366.
- [115] Y. Jansson, T. Lindeberg, Dynamic texture recognition using time-causal and time-recursive spatio-temporal receptive fields, *Journal of Mathematical Imaging and Vision* 60 (9) (2018) 1369–1398.
- [116] N. Vu, T. P. Nguyen, C. Garcia, Improving texture categorization with biologically-inspired filtering, *Image and Vision Computing* 32 (6-7) (2014) 424–436.
- [117] T. C. M. Lee, M. Berman, Nonparametric estimation and simulation of two-dimensional gaussian image textures, *CVGIP: Graphical Model and Image Processing* 59 (6) (1997) 434–445.
- [118] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 886–893.
- [119] T. Song, L. Luo, C. Gao, G. Zhang, Texture representation using local binary encoding across scales, frequency bands and image domains, in: *International Conference on Image Processing (ICIP)*, 2019, pp. 4405–4409.
- [120] I. Hadji, R. P. Wildes, A spatiotemporal oriented energy network for dynamic texture recognition, in: *International Conference on Computational Vision (ICCV)*, 2017, pp. 3085–3093.