PhD-FSTM-2020-11
The Faculty of Sciences, Technology and Medicine

DISSERTATION

Defence held on 20/04/2020 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

by

Konstantinos PAPADOPOULOS
Born on 22 March 1990 in Patras (Greece)

FROM DENSE 2D TO SPARSE 3D TRAJECTORIES FOR HUMAN ACTION DETECTION AND RECOGNITION

Dissertation defence committee

Dr. Djamila Aouada, dissertation supervisor
Research scientist, Université du Luxembourg

Dr. François Bremond
Research Director, INRIA, Sophia Antipolis, Nice, France

Assist. Prof. Radu State, Chairman
Professor, Université du Luxembourg

Prof. Stefano Berretti
Professor, University of Florence, Italy

Prof. Dr. Björn Ottersten, Vice Chairman
Professor, Université du Luxembourg

To

My family

Acknowledgements

During my journey in the Interdisciplinary Centre for Security, Reliability and Trust (SnT) at the University of Luxembourg, I have met a lot of wonderful people. To these people, I owe my personal development and achievements. First and foremost, I would like to extend my wholehearted gratitude to my advisors Dr. Djamila Aouada and Prof. Björn Ottersten for their trust and guidance throughout my studies. Thanks to them, I improved as a researcher and learned to trust my strength. Special thanks to my friend and collaborator Dr. Enjie Ghorbel for her guidance and the motivation she gave me to move further. Her help has been valuable and greatly appreciated. I would like to express my gratitude to Prof. Stefano Berretti for his helpful remarks and discussions.

I would like to thank my collaborators Dr. Michel Antunes, Renato Baptista, Dr. Girum Demisse, Oyebade Oyedotun, Himadri Pathak and Abdelrahman Shabayek for their helpful insights and their fruitful discussions on the respective research articles. I also feel lucky that I met a lot of good friends in my working environment as well as outside of it. I have been sharing my office with my friends and colleagues Renato Baptista, Jevgenij Krivochiza and Sumit Gautam whom I thank for their support and the amazing experience. Many thanks, also, to my Computer Vision colleagues and friends Alexandre, Anis, Kassem, Mohamed, Rig, Kseniya, and Joe. I would finally like to give my special thanks to Prof. Kyriakos Vlachos, Sasan, Ashik, Matthias, Kerren, Hassan, Iraklis, Giorgos, Danilo, Anestis, Liz, Gabriel, Nouli, Georgia and all of my former colleagues in SIGCOM group.

The generous financial help from the Fonds National de la Recherche (FNR - Luxembourg National Research Fund) via the University of Luxembourg is gratefully acknowledged. The work

presented in this thesis was funded by the National Research Fund (FNR), Luxembourg, under the project C15/IS/10415355/3DACT/Björn Ottersten, and by the European Union's Horizon 2020 research and innovation project STARR under grant agreement No.689947. Moreover, most of the conducted experiments were carried out using the HPC facilities of the University of Luxembourg [1]—see `https://hpc.uni.lu`.

Index

List of Abbreviations

AD ..... Action Detection

AS-GCN ..... Actional Structural Graph Convolutional Networks

BN ..... Batch Normalization

BoW ..... Bag of Words

CNN ..... Convolutional Neural Networks

CRF ..... Conditional Random Field

DNN ..... Deep Neural Networks

DH-TCN ..... Dilated Hierarchical Temporal Convolutional Networks

DT ..... Dense Trajectories

FV ..... Fisher Vectors

GMM ..... Gaussian Mixture Model

GVFE ..... Graph Vertex Feature Encoding

HOD ..... Histogram of Oriented Displacements

HOF ..... Histogram of Optical Flow

HOG ..... Histogram of Optical Gradients

HSF ..... Histogram of Scene Flow

iDT+FV ..... Improved Dense Trajectories + Fisher Vectors

iDT+SW ..... Improved Dense Trajectories + Sliding Window

LSTM ..... Long Short-Term Memory

kNN ..... k-Nearest Neighbors

KSC ..... Kinematic Spline Curves

LARP ..... Lie Algebra Representation of body-Parts

MBH ..... Motion Boundary Histograms

MoCap ..... Motion Capture

RANSAC ..... Random Sample Consensus

ReLU ..... Rectified Linear Unit

RGB ..... Red Green Blue

RGB-D ..... Red Green Blue - Depth

RNN ..... Recurrent Neural Network

SE ..... Special Euclidean

ST-GCN ..... Spatial Temporal Graph Convolutional Networks

SURF ..... Speeded Up Robust Features

SVM ..... Support-Vector Machine

TCN ..... Temporal Convolutional Networks

TSD ..... Trajectory Shape Descriptor

List of Notations

In this thesis, matrices are denoted by boldface, uppercase letters, $\mathbf{M}$, and vectors are denoted by boldface, lowercase letters, $\mathbf{v}$. Scalars are denoted by italic letters, x and features are denoted by calligraphy letters $\mathcal{F}$. The following mathematical notations will be used:

$||\mathbf{v}||_2$ ..... L_2 norm of vector $\mathbf{v}$

$\frac{\partial f}{\partial x}$ ..... partial derivative of f with respect to x

$\hat{v}$ ..... estimate of v

$\mathbf{M}^T$ ..... transpose of matrix $\mathbf{M}$

$\mathbf{M}^{-1}$ ..... inverse of matrix $\mathbf{M}$

Q_t ..... skeleton sequence at frame t

q^j ..... skeleton joint j of skeleton Q

V^t ..... RGB sequence at frame t

Δx ..... displacement of x

P^m ..... trajectory P with id m

p^m ..... point of trajectory P^m

$log(x)$ ..... logarithm of x

$median()$ ..... median value of a vector

$argmax()$ ..... the maximizing argument

$Pool()$ ..... the pooling method

$f \circ g$ ..... composition of functions $f()$ and $g()$

List of Figures

List of Tables

Abstract

Human action recognition has been an active research topic in the field of computer vision and has attracted interest in multiple applications such as video surveillance, home-based rehabilitation, and human-computer interaction. In the literature, to model motion, trajectories have been widely employed given their effectiveness. There are different variants of trajectory-based representations. Among the most successful ones, one can refer to the dense trajectories, commonly extracted from an RGB stream using optical flow, and the sparse trajectories from either 2D or 3D skeleton joints, usually provided by 3D sensors. Although dense and sparse trajectory-based approaches have shown great performance, each of them presents different shortcomings. Despite their ability to track subtle motion with precision, dense trajectories are sensitive to noise and irrelevant background motion and lack locality awareness. Furthermore, due to their 2D nature, dense trajectories show limited performance in the presence of radial motion. Sparse trajectories, on the other hand, form a high-level and compact representation of human motion which is widely adopted in action recognition. However, they are barely applicable in real-life scenarios due to limitations coming from 3D sensors, such as close range requirements and sensitivity to outdoor illumination.

In this thesis, we propose to overcome the aforementioned issues by exploring and extending both representations; thus, going from 2D dense to 3D sparse trajectories. In the first part of this thesis, we combine both dense and sparse representations. First, we introduce *Localized Trajectories* which endow dense trajectories a local description power by clustering motion trajectories around human body joints and then encoding them using local Bag-of-Words. We also revisit action detection by exploiting dense trajectories and skeleton features in an alternative

way. Moreover, for a better description of radial motion, we extend Localized Trajectories to 3D by computing the scene flow from the depth modality.

In the second part of this thesis, we focus on representations purely based on 3D sparse trajectories. To overcome the limitations presented by 3D sensors, we exploit the advances in 3D pose estimation from a single RGB camera to generate synthetic sparse trajectories. Instead of relying on a traditional skeleton alignment, virtual viewpoints are used to augment the viewpoint variability in the training data. Nevertheless, the estimated 3D skeletons present naturally a higher amount of noise than the ones acquired using 3D sensors. For that reason, we introduce a network that implicitly smooths skeleton joint trajectories in an end-to-end manner. The successful Spatial Temporal Graph Convolutional Network (ST-GCN) which exploits effectively the graph structure of skeleton sequences is jointly used for recognizing the actions. However, raw skeleton features are not informative enough for such networks and important temporal dependencies are ignored. Therefore, we extend the ST-GCN by introducing two novel modules. The first module learns appropriate vertex features by encoding raw skeleton data into a new feature space. The second module uses a hierarchical dilated convolutional network for capturing both short-term and long-term temporal dependencies. Extensive experiments and analyses are conducted for validating all of our contributions showing their effectiveness with respect to the state-of-the-art.

Chapter 1

Introduction

Humans have the capacity to understand and interpret human motion intuitively. In reality, this results from a complex biological procedure. In order to identify actions, humans usually follow a specific process which consists of three stages: *observation*, *processing* and *recognition*. First, a sequence of observations is acquired using our sensory devices (eyes), which is then processed by specific neurons in our brains. These neurons are trained to focus on particular motion cues and, using the previously collected knowledge, the final decision of the action type is made. Thus, imitating the behavior of such a system is not straightforward.

Over the last decades, designing a system able to recognize actions automatically using only cameras and computers has gained a huge interest in computer vision. This is due to the increasing demand for automation and the several applications that are related to this field such as surveillance and security [2], healthcare, and assisted living [3]–[7], and human-computer interaction [8]. An illustration of an action recognition system is presented in Figure 1.1. Before presenting our work, we first summarize the taxonomy in the field of action recognition.

The objective of action recognition is to reproduce the different steps described above that are naturally performed by humans. Similar to the biological human recognition system, there are three defined stages in action recognition: sequence acquisition, feature extraction, and classification. The first stage concerns the acquisition of visual information. During the second stage, a processing step is applied to the obtained sequences to compute *features*. By definition, features are distinctive attributes and patterns computed from visual sequences that are relevant

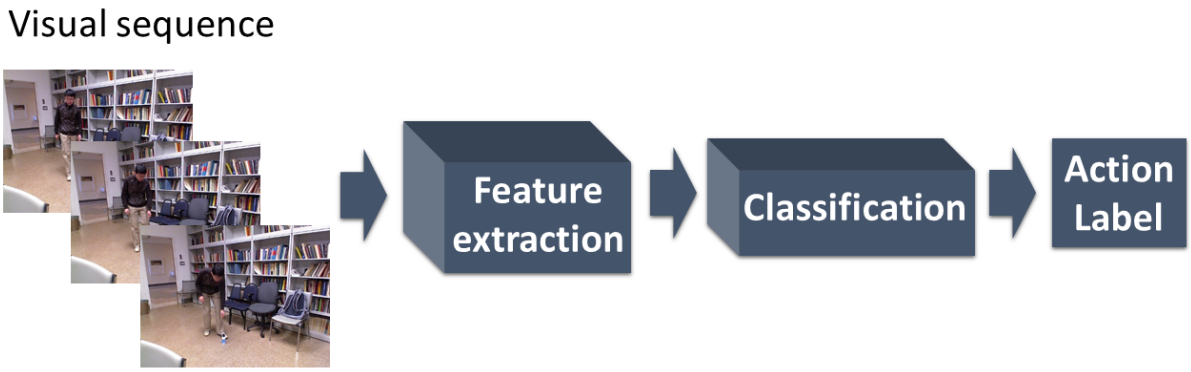


Figure 1.1: Example of an expert system for action recognition. Video sequences are received as input and based on the prior knowledge it has obtained through training with similar data, it generates an action label.

to the observed human motion. The classification step involves learning the correspondence between features and action labels allowing the distinction between different action classes. Recently, deep learning approaches have been widely popular for the feature extraction and classification steps.

Human motion is usually categorized into three groups, depending on the complexity of motion: gestures, actions, and activities. Gestures are the simplest form of actions since they involve only hand and finger movements. Actions are generally more complex, in the sense that they include the motion of one or more body parts concurrently. Finally, activities are semantically the most representative of a human's daily life and, consequently, the most complex. Moreover, they usually involve interaction with objects. In this thesis, we focus mainly on the recognition of actions and activities.

1.1 Motivation and Scope

Human action recognition is an active research topic in computer vision for several decades. Despite the wide interest of the scientific community, this domain remains particularly challenging due to several open issues, such as inter and intra-action class variations, changes in camera viewpoints, occlusions, and lighting conditions.

The way of describing motion plays an important role in identifying informative action patterns.

There are several ways of describing human motion. One of the most successful concepts, among others, is *dense trajectories*. Dense trajectories are usually extracted from RGB videos and involve the tracking of spatio-temporal points, projected on an image grid. To provide good coverage of the video with features, points are sampled uniformly and densely. This ensures the detailed tracking of subtle motion. To track points, the displacement vectors between consecutive frames should be given. For this reason, optical flow algorithms are used which estimate the spatial displacement of points given a pair of frames. Optical flow offers low-level motion representation, thus, improved quality of trajectories.

With the recent advances in Motion Capture systems and depth sensors, the human body can be represented by a set of 3D points, forming a skeleton structure. 3D skeletons are characterized by (a) a high-level representation, as a result of processing raw depth data or sequences captured from multi-camera setups and (b) compactness, requiring only a small set of points and links to portray a human body. Skeleton sequences include the tracking of joints, resulting in a sparse motion description. Compared to dense trajectories, skeleton trajectories or *sparse trajectories* are a more condense concept that summarizes the total amount of motion coming from one body part in a single trajectory. Therefore, various important details are still preserved and irrelevant information, such as the background motion, is eliminated. An illustration of a skeleton structure, along with the corresponding RGB and depth modalities is given in Figure 1.2.

In this thesis, we propose to explore and investigate both the aforementioned representations, going from 2D dense to 3D sparse trajectories. We also introduce a hybrid representation that combines both dense and sparse trajectories, similar to existing multimodal approaches in computer vision [9], [10].

1.2 Challenges in Dense Trajectory-based Action Recognition

Tracking specific points in an RGB or depth video sequence is important for action recognition. Discriminative motion patterns lie in the motion trajectories and the recognition of them is crucial for the description of actions. However, tracking points is not a straightforward task.

Figure 1.2: RGB (left), Depth (middle) and 3D Skeleton (right) modalities. [11]

The position changes of points have to be estimated assuming that the point remains within a certain spatial neighborhood. This challenge is mostly addressed when using both RGB and depth modalities by computing *optical flow* [12], [13] and *scene flow* [14], [15], respectively. Optical flow estimates the displacement of points between consecutive frames by estimating their new position in terms of intensity values in a local area. Scene flow extends the idea of optical flow to 3D. The estimation of 3D displacement is achieved by jointly estimating the new pixel positions in terms of intensity and depth. Thus, in this scope, we use the notion of *dense trajectories* corresponding to the spatio-temporal tracks of points using optical flow. The high density of such trajectories derives from the dense image point selection for tracking. Dense Trajectories proposed by [16] was among the earliest approaches in this domain and introduced the computation of trajectory-aligned spatio-temporal features.

In classical Dense Trajectories, there is no information on which body part each trajectory is related to. As a result, similar motion patterns which belong to different body parts may be confusing for the classifier. Moreover, Dense Trajectories include motion which is irrelevant to the main activity due to background motion and noise. Furthermore, 2D Dense Trajectories are generated using optical flow which fails to describe motion with radial orientation with respect to the image plane.

Approaches based on Dense Trajectories are effective in describing particular local motion which cannot be described by skeleton data, such as finger movements. In addition, information is extracted from the local spatial region of each trajectory. Therefore, interactions with objects

can be described more effectively than skeleton representations since there are no limitations coming from the missing background and texture. However, 2D Dense Trajectories reduce the dimensionality of the natural motion, since the depth factor is not considered. For a more effective motion representation, 3D Dense Trajectories are essential.

1.3 Challenges in Sparse Trajectory-based Action Recognition

With the recent advancements in depth sensors, the representation of the human body structure as a small set of 3D points became possible [17]. These points are connected using pre-defined skeleton links that correspond to body parts. Thus, human activities can be identified by tracking the 3D joint trajectories over time, to which we refer from now on as *sparse trajectories*. The obvious advantage of this skeletons is the reduction of data complexity, originating from the compact representation of the human body. As a result, sparse trajectories act as a summary of dense trajectories related to human body parts. This representation gained significant popularity and has been used in a wide range of applications [18]–[23]. 3D skeletons are compact and allow the efficient computation of view-invariant features for cross-view action recognition.

Early works in sparse 3D trajectories involved the modeling of the spatial-temporal evolution of actions using hand-crafted features [18], [19], [24]. Recently, deep learning architectures have shown great potential. Through intensive training, deep networks learn discriminative features from data modalities without the need for hand-crafted features. In skeleton-based action recognition, such approaches have achieved state-of-the-art results [20]–[22]. However, they do not exploit effectively the non-Euclidean structure of the skeletons. To overcome this, Spatial Temporal Graph Convolutional Networks (ST-GCN) [25] have been recently introduced. They represent skeleton sequences as graphs and apply graph convolutions on them, which are generalized convolutions from images to graphs.

Viewpoint variation is one of the most significant challenges in human action recognition. Typically, 3D representation is a straightforward way to achieve view invariance, since it requires basic geometric operations to align 3D skeletons to a canonical form. However, in the presence of noisy skeletons, the alignment step can be erroneous, affecting the view-invariance of the

representation. While RGB-D cameras offer a richer representation of the scene, compared to RGB sensors, by incorporating depth information, they present two main drawbacks. First, an acceptable estimation of depth maps is only possible within a specific range. As a result, the estimation of 3D skeletons can only be reliable within the same range. Second, RGB-D cameras are extremely sensitive to external lighting conditions, making them inadequate for most outdoor-related applications. Both cases result in noisy depth maps and several approaches have been proposed in an attempt to address this challenge [26]–[29].

Another unexplored region of skeleton-based action recognition is the suitability of raw skeleton features such as joint positions and bone lengths when using ST-GCN. Such features offer a human-interpretable representation of the body structure, but their discriminative power for action recognition has been limited as shown in [24], [30].

1.4 Objectives and Contributions

The objective of this thesis is to address the challenges presented in Sections 1.2 and 1.3 from RGB and RGB-D data. The main contributions are listed and presented below.

1.4.1 Localized Trajectories for 2D and 3D Action Recognition

Our first contribution, namely Localized Trajectories, utilizes human pose to enhance Dense Trajectories [16] for action recognition. By taking advantage of the availability of RGB-D cameras, we propose to use 2D human pose information in order to cluster Dense Trajectories around human body joints. This offers a local discriminative power compared to the original approach and increases the robustness to noise and background motion. Consequently, actions which have similar motion patterns, but are performed by different body parts, are more easily distinguished. Besides, our approach utilizes the concept of Local Bag-of-Words [31], which allows a more relevant feature encoding.

We further extend the Localized Trajectories concept to 3D by utilizing the depth modality provided by the RGB-D cameras. This extension offers a more effective description of the perpendicular to the camera plane motion, termed radial motion. The computation of *3D Local-*

ized Trajectories is based on scene flow estimation which captures point displacements in all three dimensions. This approach is further strengthened by a new feature sampling method for codebook generation based on confidence and ambiguity metrics.

This work has been published in [32] and [33].

1.4.2 Dense Trajectory-based Action Detection using Skeleton Data

In the context of action detection, we show how the utilization of human pose can also be beneficial in dense trajectory-based action detection, which is the detection and recognition of actions in untrimmed videos. In this scope, we developed a two-stage action detection concept. This concept segments temporal regions-of-interest using pose information in the first stage and performs dense trajectory-based action classification in the second stage. The main advantage of this approach is the temporal segmentation in the first stage, which addresses the performance issues of classical trajectory-based action detection resulting from background, low-motion action videos.

This work has been published in [34].

1.4.3 Cross-view Action Recognition using Sparse Trajectories from RGB Data

Knowledge transfer from 3D data has been widely used for addressing the problem of cross-view RGB-based action recognition. In this thesis, we approach this challenge from a novel perspective. We enforce view-invariance in RGB videos by taking advantage of recently developed 3D skeleton sequence estimation [35]. The proposed pipeline consists of two steps. The first step is the dimensionality augmentation using 3D skeleton sequence estimation from RGB sequences. The second step is the computation of view-invariant features from the estimated 3D sequences. For this purpose, two view-invariant skeleton-based descriptors are utilized and analyzed.

This work has been published in [36].

1.4.4 Learning Deep View-Invariant Human Action Representations using a Single RGB Camera

We propose a new RGB-based framework for cross-view action recognition from estimated 3D poses without the need for pose alignment. Since these poses can be noisy, we introduce two modules as parts of the overall framework. The first module is an end-to-end trainable network which applies temporal smoothing to the estimated sequences. The second module replaces the pose alignment step with viewpoint augmentation, forcing the network to learn view-invariant patterns directly from data. Although we rely on ST-GCN for action classification, these modules can be used with any skeleton-based action recognition network.

This work has been published in [37] and [38] is under submission.

1.4.5 Improving ST-GCNs for Skeleton-based Action Recognition

Representing skeleton sequences as graphs has already shown huge potential in action recognition. However, the construction of graphs relies on raw skeleton features which may be insufficient for this task and it also requires a significant number of ST-GCN blocks, increasing the network complexity. In addition, the temporal dependencies of the graph are modeled by a single temporal convolutional layer. As a result, critical long-term dependencies might be not consistently described. Thus, we propose a novel module to encode vertex features to a new feature space which is more appropriate for action recognition. Furthermore, we employ hierarchical dilated convolutional layers to model both short-term and long-term temporal dependencies. Both modules are trained in an end-to-end manner with the main network. The efficient encoding of both graph features and temporal dependencies allows the compression of the network [39] and the reduction of the trainable parameters.

This work is currently under review [40].

1.5 Publications

JOURNALS

1. **Papadopoulos K.**, Demisse G., Ghorbel E., Antunes M., Aouada D., Ottersten B. ”Localized Trajectories for 2D and 3D Action Recognition”. Sensors. 2019; 19(16):3503.

2. **Papadopoulos K.**, Ghorbel E., Oyedotun O., Aouada D., Ottersten B., ”Learning Deep View-Invariant Representations from Synthetic Viewpoints using a Monocular RGB Camera”, IEEE Transactions on Image Processing, 2020. **(To be submitted)**

CONFERENCES

1. **Papadopoulos, K.**, Antunes, M., Aouada, D., Ottersten, B. (2017, September). ”Enhanced trajectory-based action recognition using human pose”. In 2017 IEEE International Conference on Image Processing, Beijing, 17-20 September 2017.

2. **Papadopoulos, K.**, Antunes, M., Aouada, D., Ottersten, B. (2018, April). ”A revisit of action detection using improved trajectories”. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing, Calgary, 15-20 April 2018.

3. Ghorbel, E., **Papadopoulos, K.**, Baptista, R., Pathak, H., Demisse, G., Aouada, D., Ottersten, B. (2019). ”A view-invariant framework for fast skeleton-based action recognition using a single rgb camera”. In 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, Prague, 25-27 February 2018.

4. **Papadopoulos K.**, Ghorbel E., Oyedotun O., Aouada D., Ottersten B., ”DeepVI: A Novel Framework for Learning Deep View-Invariant Human Action Representations using a Single RGB Camera”, In 15th IEEE International Conference on Automatic Face and Gesture Recognition, Buenos Aires, 18-22 May 2020.

5. **Papadopoulos K.**, Ghorbel E., Aouada D., Ottersten B., ”Vertex Feature Encoding and Hierarchical Temporal Modeling in a Spatio-Temporal Graph Convolutional Network for

Action Recognition", In 25th IEEE International Conference on Pattern Recognition, Milan, 13-18 September 2020. **(Under Review)**

PUBLICATIONS NOT INCLUDED IN THE THESIS

1. Demisse, G. G., **Papadopoulos, K.**, Aouada, D., Ottersten, B. (2018). "Pose encoding for robust skeleton-based action recognition". In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 188-194).

2. Baptista, R., Ghorbel, E., **Papadopoulos, K.**, Demisse, G. G., Aouada, D., Ottersten, B. (2019). "View-invariant Action Recognition from Rgb Data via 3D Pose Estimation". In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 2542-2546).

3. **Papadopoulos, K.**, Ghorbel, E., Baptista, R., Aouada, D., Ottersten, B. (2019). "Two-stage RGB-based action detection using augmented 3d poses". In International Conference on Computer Analysis of Images and Patterns (pp. 26-35). Springer, Cham.

1.6 Thesis Outline

This dissertation is organized as follows:

- **Chapter 2:** In this chapter, the background of our proposed approaches is presented. The computation of both dense and sparse trajectories is described along with the generation of synthetic sparse trajectories.

- **Chapter 3:** In chapter 3, our proposed framework, named Localized Trajectories, is introduced. This framework extends the concept of dense trajectories by incorporating pose information to it. Pose information offers locality awareness and consequently enhances the discriminative power of dense trajectories. This concept is also extended to 3D in order to address the poor radial motion description.

- **Chapter 4:** In this chapter, our contribution to action detection is introduced. Dense trajectories are combined with human pose information in a novel way, resulting in an effective action detection concept.

- **Chapter 5:** A move from dense trajectories to the more concise sparse trajectories is made. In this chapter, synthetic 3D sparse trajectories are utilized for the task of cross-view action recognition. Our framework addresses the problem of cross-view action recognition from RGB data by generating view-invariant representations from synthetic skeleton sequences.

- **Chapter 6:** In this chapter, the DeepVI framework is introduced. This framework addresses the problem of cross-view action recognition from RGB data using synthetic sparse trajectories. Viewpoint augmentation and a novel trainable smoothing module are introduced to achieve view-invariant representations, in combination with ST-GCN.

- **Chapter 7:** In this chapter, a module is introduced for ST-GCNs. This module transfers input features to a new feature space that is more suitable for the task of action recognition. Our contribution, named Dilated Hierarchical Temporal Convolutional Network is also proposed, aiming to encode effectively temporal dependencies.

- **Chapter 8:** Concluding remarks and perspectives on future work building on the contributions of this thesis are discussed.

Chapter 2

Background

2.1 Introduction

Among the most popular data sequences used in the literature, one can cite RGB and RGB-D data sequences that introduce different challenges. In this thesis, as mentioned in Chapter 1, we focus on representations that are based on motion trajectories. Depending on the acquisition system, the way of extracting motion trajectories from visual sequences differs. In this chapter, we describe the different approaches commonly used in the literature for extracting dense as well as sparse motion trajectories.

2.2 Dense Trajectories for Action Recognition

For designing an effective human action recognition system, the use of a relevant description of motion is crucial. Numerically, motion is defined as the displacement of image points through time. However, raw video data do not explicitly offer such information. For this purpose, the concept of *optical flow* has been introduced [12], [13]. Optical flow, also known as motion estimation, is a group of methods for approximating true physical motion projected in the two-dimensional image plane. Then, the calculation of trajectories using optical flow is presented in the following section.

2.2.1 Optical Flow

Optical flow describes both the orientation and the velocity of motion. Using color intensity values, it approximates the displacement of each image pixel over time in a video volume.

Several approaches calculate the optical flow field $(\mathbf{u}, \mathbf{v})$ based on optimization techniques. They can be grouped in differential-based, region-based, energy-based, and phase-based techniques [41]. Differential methods [42], [43] are the most popular and widespread ones in the literature.

Given a video sequence V, the goal of optical flow is to calculate the pixel displacement $(\Delta x, \Delta y)$ of the spatial coordinates (x, y) in the time interval $[t, t + \Delta t]$, so that the following brightness constancy constraint is satisfied,

$$\mathsf{V}^{t}_{x,y} = \mathsf{V}^{\Delta t+t}_{\Delta x+x,\Delta y+y}. \tag{2.1}$$

With the assumption of the presence of subtle motion, Equation (2.1) can be reformulated as follows,

$$\mathsf{V}^{\Delta t+t}_{\Delta x+x,\Delta y+y} \approx \mathsf{V}^{t}_{x,y} + \frac{\partial \mathsf{V}}{\partial x}\Delta x + \frac{\partial \mathsf{V}}{\partial y}\Delta y + \frac{\partial \mathsf{V}}{\partial t}\Delta t. \tag{2.2}$$

Using Equation (2.1) and Equation (2.2), we obtain,

$$\frac{\partial \mathsf{V}}{\partial x}\Delta x + \frac{\partial \mathsf{V}}{\partial y}\Delta y + \frac{\partial \mathsf{V}}{\partial t}\Delta t = 0. \tag{2.3}$$

By dividing with Δt, we obtain,

$$\frac{\partial \mathsf{V}}{\partial x}\frac{\Delta x}{\Delta t} + \frac{\partial \mathsf{V}}{\partial y}\frac{\Delta y}{\Delta t} + \frac{\partial \mathsf{V}}{\partial t}\frac{\Delta t}{\Delta t} = 0, \tag{2.4}$$

which is finally expressed as,

$$\frac{\partial \mathsf{V}}{\partial x}\mathbf{u} + \frac{\partial \mathsf{V}}{\partial y}\mathbf{v} + \frac{\partial \mathsf{V}}{\partial t} = 0, \tag{2.5}$$

where $(\mathbf{u}, \mathbf{v})$ are the components of the optical flow in $\mathsf{V}^{t}_{x,y}$, and $\left(\frac{\partial \mathsf{V}}{\partial x}, \frac{\partial \mathsf{V}}{\partial y}, \frac{\partial \mathsf{V}}{\partial t}\right)$ are the partial derivatives of the image at the position (x, y, t). Since only one equation with two unknown variables, $\mathbf{u}$ and $\mathbf{v}$, is given, additional constraints are required, and they are usually provided by

neighboring pixels. In the next session, we show how optical flow is used for tracking points in a spatio-temporal volume.

2.2.2 Tracking 2D Dense Trajectories from RGB Data

Dense Trajectories were initially introduced by Wang, Klaser, Schmid, and Liu [16]. They are constructed by densely tracking sampled points over an RGB video stream and constructing representative features around the detected trajectories. They mainly owe their success to the fact that they incorporate low-level motion information. Below, we overview the Dense Trajectories approach.

Let V be a sequence of N images. Subsequently, representative points are sampled from each image grid with a constant stepping size—we denote each sampling grid position at frame t as $\mathsf{p}_t = \mathsf{V}^t_{x,y}$. The point p_t is then estimated in the next frame using a motion field $(\mathbf{u}_t, \mathbf{v}_t)$, derived by the optical flow estimation [13] such that,

$$\mathsf{p}_{t+1} = \mathsf{p}_t + \kappa \cdot \ (\mathbf{u}_t, \mathbf{v}_t), \tag{2.6}$$

where κ is a median filter kernel at the position p_{t+1}. As a result, large motion changes between subsequent frames are smoothed. Furthermore, to avoid drifting, trajectories longer than the assigned fixed length are rejected. Applying Equation (2.6) on a temporal window L results in a smoothed trajectory estimation of the point $\mathsf{p}_t = \mathsf{V}^t_{x,y}$. A trajectory P^m is defined as,

$$\mathsf{P}^m = \{\mathsf{p}^m_{t_0}, ..., \mathsf{p}^m_{t_0+L}\}, \tag{2.7}$$

with $m \in \{1, ..., M\}$, t_0 the first frame of the sequence V and M the total number of generated trajectories. The set of M trajectories generated in Equation (2.7) is used to construct descriptors aligned along a spatiotemporal volume.

One of the main drawbacks of Dense Trajectories is that points on the image grid are sampled uniformly, which potentially leads to the inclusion of a significant amount of noise deriving from background motion. Furthermore, the generated Dense Trajectories do no take into account

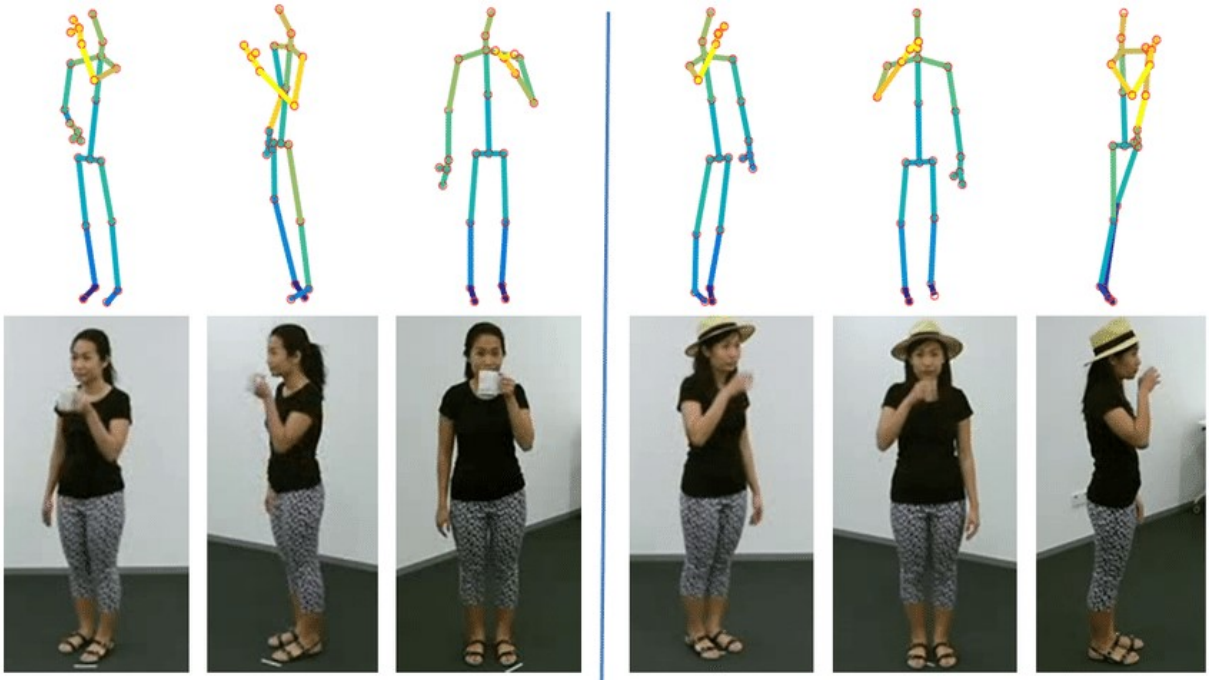


Figure 2.1: Example of skeleton structure extracted from the corresponding RGB frames.

the spatial human body structure. Thus, actions with similar motion patterns can potentially be confused during classification.

2.3 Tracking 3D Sparse Skeleton Trajectories from Depth Data

With the introduction of RGB-D sensors, such as Microsoft Kinect [44], the depth modality attracted scientific interest in action recognition. There are two types of depth sensors currently available: Structured-light and Time-of-Flight. Structured-Light sensors analyze the distortion of a known pattern which is projected in the scene to estimate the distance of each point of the pattern. Time-of-Flight sensors, on the other hand, estimate depth by measuring the round trip time of an artificial light signal emitted by the source.

Depth sequences can be used to extract the 3D skeletal representation of the human body. A desired property of the skeletal structure is the pre-defined joint connectivity that portrays each rigid body part as an edge and preserves the body part dependencies. Towards this

direction, Shotton, Sharp, Kipman, Fitzgibbon, Finocchio, Blake, Cook, and Moore [17] proposed a real-time algorithm for the extraction of human skeletons from depth sequences. This algorithm segments the human body into parts and marks the borders between two neighboring parts as joints. Thus, at a timestamp t, a skeleton representation Q_t is defined as,

$$\mathrm{Q}_t = \{\mathrm{q}_t^1, ..., \mathrm{q}_t^J\}, \tag{2.8}$$

where $\mathrm{q}_t \in \mathbb{R}^3$ is a skeleton joint and J is the number of joints. This representation allows the tracking of 3D trajectories over time and has several advantages since it is (a) compact, as each body part consists of two 3D joints and (b) human-focused since the background is removed. An illustration of a skeleton structure can be found in Figure 2.1.

Compared to classical dense trajectories motion tracking in skeleton sequences is straightforward. 3D motion from skeleton joint sequences Q^j is the tracking of each body joint $\mathrm{q}^j \in \mathbb{R}^3$ such that,

$$\mathrm{Q}^j = \{\mathrm{q}_1^j, ..., \mathrm{q}_N^j\}, \tag{2.9}$$

where N is the total action duration and j the joint id. Given their compactness, we refer to this representation as 3D sparse trajectories. However, this representation cannot model properly human-object interaction since it tracks only human body joints.

The compact and simple nature of skeleton sequences has turned them into one of the most common data representations in action recognition. Skeleton sequences include explicit temporal dynamics of actions in the form of spatio-temporal keypoints. Several approaches that encode skeletal action dynamics have been already proposed in the literature [19]–[22], [30], [45]–[47]. One way to categorize such approaches is based on feature extraction. Thus, there are approaches that perform hand-crafted feature extraction [19], [30], [45], [46] and deep-learning approaches [20]–[22], [47].

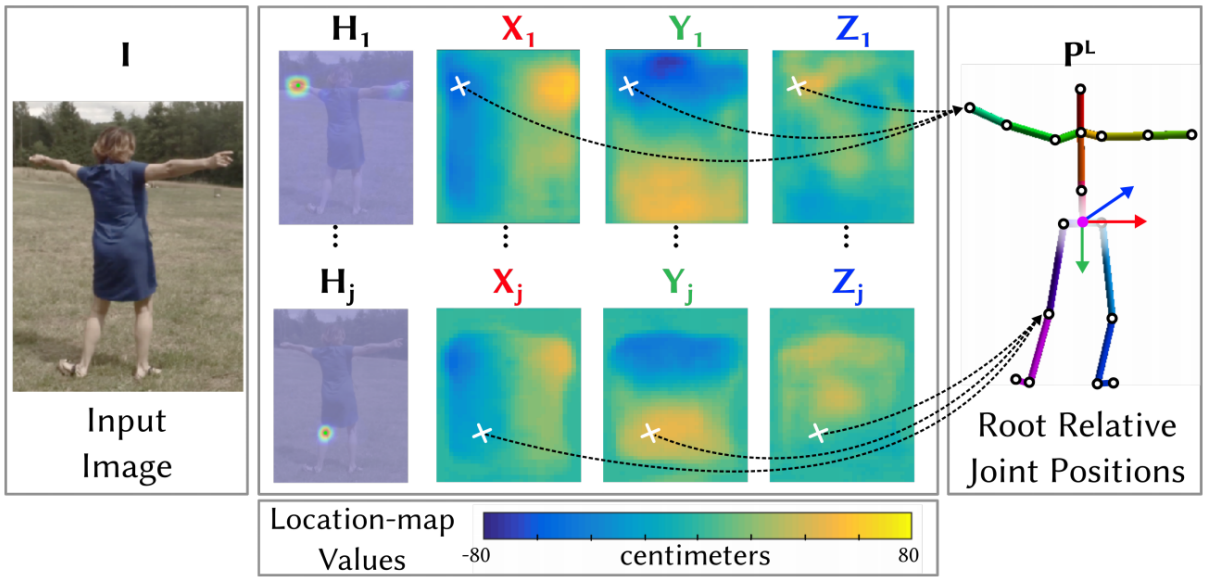


Figure 2.2: Estimation of 3D pose from RGB image. For each joint four types of feature maps are estimated. Feature map H is a likelihood map of the joint location on the image grid. Location heatmaps $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ provide the estimated three-dimensional location of joints.

2.4 Tracking 3D Sparse Trajectories from RGB sequences

Despite their popularity, skeleton representations may suffer from noise due to the limitations coming from depth sensors (e.g. Time-of-Flight cameras). In addition, these sensors show high sensitivity to external lighting conditions, and their range of capture is limited to $\sim$5 meters. Thus, they are hardly applicable to real-life scenarios. Estimating a 3D skeleton from a single RGB image has been, for a long time, considered as an ill-posed problem and an almost impossible task. Some first attempts have been made in the state-of-the-art [48]–[50] based on hand-made features and regression models. Nevertheless, the efficiency of these approaches remains very limited. Recently, thanks to the advances in deep learning, a wide range of more reliable approaches have started to emerge [51]–[56]. Thus, it becomes possible to generate 3D poses from an RGB stream.

Given a sequence of RGB images $\mathsf{V} = \{\mathsf{V}^1, \cdots, \mathsf{V}^t, \cdots, \mathsf{V}^N\}$, where N is the total number of frames, the goal of a pose estimation algorithm $f(\cdot)$ is the approximation of a 3D skeleton $\hat{\mathsf{Q}}$ composed of J joints. We denote the sequence of extracted skeletons by $\hat{\mathsf{Q}}_t = \{\hat{\mathsf{q}}_t^1, \hat{\mathsf{q}}_t^2, ..., \hat{\mathsf{q}}_t^J\}$

such that

$$\hat{\mathsf{Q}}_t = f(\mathsf{V}^t) \approx \mathsf{Q}_t, \tag{2.10}$$

where $f(\cdot)$ is a function that maps a single RGB image to an estimated representation of the human pose in three dimensions and where $t \in \{1, \cdots, N\}$ denotes the frame index. Most of the proposed DNN-based approaches estimate 3D skeletons from each RGB frame independently [57], [58]. However, this can result on temporally unstable 3D poses. Thus, some attempts have been made to consider the temporal consistency, as in [59].

A typical example of a state-of-the-art 3D pose estimator is VNect [35], known for its temporal consistency, its fast performance, and its high estimation accuracy. As in [60], [61], VNect makes use of Convolutional Neural Networks (CNN) models. However, authors select a smaller architecture based on Residual Networks (ResNet) to achieve real-time performance. This CNN pose regression estimates 2D and 3D skeletons using a monocular RGB camera. To that aim, for each joint j, the network is trained to estimate a 2D heatmap $\mathbf{H}_j$ of likelihood joint location scores along with joint location maps in each of the three dimensions, which we denote as $\mathbf{X}_j, \mathbf{Y}_j, \mathbf{Z}_j$. All four heatmaps are shown in Figure 2.2. The position of each joint j is therefore estimated by extracting the maximum values from the location maps of the associated heatmap $\mathbf{H}_j$.

The network is trained by considering the weighted L_2 norm difference between estimated joint location and the ground truth– the cost is summed over each dimension. For instance, the loss of predicting location $\hat{\mathsf{q}}^j$, is given as

$$\text{Loss} = \|\mathbf{H}_j^{GT} \odot (\mathbf{X}_j - \mathbf{X}_j^{GT})\|_2, \tag{2.11}$$

where GT refers to the Ground Truth and $\odot$ indicates the Hadamard product.
The network is pre-trained using the annotated 3D and 2D human datasets [60], [62], [63]. In order to ensure temporal coherence, the estimated joint positions are later smoothed.

Chapter 3

Localized Trajectories for 2D and 3D Action Recognition

The Dense Trajectories concept is one of the most successful approaches in action recognition, suitable for scenarios involving a significant amount of motion. However, due to noise and background motion, many generated trajectories are irrelevant to the actual human activity and can potentially lead to performance degradation. In this chapter, we introduce a novel 2D Localized Trajectories concept, which utilizes the body pose information in order to cluster trajectories that are semantically similar. Moreover, we extend Localized Trajectories from 2D to 3D thanks to the availability of depth data, which are directly used for 3D motion estimation. Finally, a novel feature selection concept for a robust codebook construction is presented, along with extensive experimental evaluation on several RGB-D datasets is presented to validate the discriminative power of the proposed approach.

3.1 Introduction

The Dense Trajectories approach [16] belongs to *local* approaches, where specific regions of interest are selected to generate features. Approaches based on Dense Trajectories are particularly effective when the amount of motion is high [64]. This is mainly because images

in a video are densely sampled and tracked for generating the trajectories. However, Dense Trajectories, by definition, include trajectories of points that are irrelevant for action recognition due to background motion and noise, thus resulting in the inclusion of irrelevant information. Furthermore, Dense Trajectories are typically generated using optical flow which fails to describe motion with radial orientation with respect to the image plane. Therefore, taking advantage of the availability of RGB-D cameras, we propose to redefine Dense Trajectories by giving them a local description power. This is achieved by clustering Dense Trajectories around human body joints provided by RGB-D sensors, which we refer to as *Localized Trajectories* henceforth.

The proposed approach offers two main advantages. First, since we only consider trajectories that are localized around human body joints, our approach is more robust to large irrelevant motion estimates. As a consequence, actions which have similar motion patterns, but involving different body parts, are more easily distinguished. Second, our approach allows the description of the relationship of "*action–motion–joint*", i.e., an action is associated with both; a type of motion and joint location, in contrast to classical Dense Trajectories described by the relationship "*action–motion*" where an action is associated with a type of motion only. This is done by generating features around the Localized Trajectories based on the concept of local BoWs [31]. One codebook is therefore constructed per group of Localized Trajectories. Each codebook corresponds to a specific body joint.

For a better description of radial motion, we further propose to explore Localized Trajectories using the three modalities provided by RGB-D cameras. Specifically, we introduce the *3D Localized Trajectories* concept, which requires the estimation of scene flow, the displacement vector field in 3D, instead of optical flow. Coupling 3D Trajectories and the corresponding motion descriptors with Localized Trajectories offers richer localized motion information, in both lateral and radial directions, allowing better discrimination of actions. However, scene flow estimation is generally noisier resulting in a less accurate temporal tracking of points. Thus, we propose to construct local codebooks by sampling trajectory-aligned features based on confidence and ambiguity metrics [65].

3.2 Related Work

In this section, we present some of the state-of-the-art action recognition approaches. First, we start by giving a general overview of RGB-D based action recognition approaches. Then, we focus on representations inspired by Dense Trajectories that are directly related to our work.

3.2.1 Dense Trajectories Related Approaches

Initially introduced by Wang, Klaser, Schmid, and Liu [16], Dense Trajectories are classically generated by computing motion and texture features around motion trajectories. Due to their popularity, many researchers have extended this original formulation in order to enhance their performance [64], [66]–[69].

As a first attempt, Wang and Schmid [66] proposed to reinforce Dense Trajectories by using the Random Sampling Consensus (RANSAC) algorithm to reduce the noise caused by motion. In addition to that, they replaced the Bag-of-Visual-Words representation with Fisher Vectors.

Then, Koperski, Bilinski, and Bremond [64] suggested enriching motion trajectories using depth information. They proposed a model grouping the videos in two types: videos with a high level of motion and others with a low amount of motion. For the first group, an extension of Trajectory Shape Descriptor [16], which includes depth information has been used, while for the second group a novel descriptor called Speeded Up Robust Features (SURF) has been introduced in order to generate local depth patterns.

To further improve the accuracy of recognition, Wang, Qiao and Tang [67] proposed to use deep learned features instead of heuristic spatiotemporal local ones such as Trajectory-Shape Descriptor (TSD) [16], Histogram of Oriented Gradients (HOG) [70], Histogram of Optical Flow (HOF) [71], and Motion Boundary Histogram (MBH) [16].

On the other hand, in [68], a novel approach to encode relations between motion trajectories is presented. Global and local reference points are used to compute Dense Trajectories, offering robustness to camera motion.

Finally, Ni, Moulin, Yang, and Yan [69] had the idea of focusing on trajectory groups that contribute more importantly to a specific action by defining an optimization problem. Towards the

same direction, Jhuang, Gall, Zuffi, Schmid, and Black [72] proposed the extraction of features around joint trajectories, increasing the discriminative power of the original Dense Trajectories approach [16].

Although all the aforementioned methods have shown their effectiveness, they, unfortunately, lack locality information related to the human body. This piece of information is crucial when actions include similar motion patterns performed by different body parts. For this reason, we propose a novel dense trajectory-based approach by taking into consideration the local spatial repartition of motion with respect to the human body.

3.2.2 Action Recognition from RGB-D Data

With the recent availability of affordable RGB-D cameras, a great effort in action recognition using both RGB and depth modalities has been made. For a more comprehensive state-of-the-art, we refer the reader to a recent survey [73], where RGB-D based action recognition methods have been grouped into two distinct categories (according to the nature of the descriptor), namely, *learned representations* [74]–[76] and *hand-crafted representations* [65], [77], [78]. Since this work deals with the description of actions using Dense Trajectories, we mainly focus on hand-crafted based approaches. In turn, they can be classified as follows: depth-based approaches, skeleton-based approaches, and hybrid approaches.

The first class of methods extracts directly human motion information from depth maps [77], [79]–[86]. The second group gathers approaches that make use of the 3D skeletons extracted from depth maps. During the past few years, a wide range of methods has been designed using this high-level modality [24], [30], [45], [46], [87]–[89].

Compared to depth-based descriptors, skeleton-based descriptors require low computational time, are easier to manipulate, and can better discriminate local motions. However, they are more sensitive to noise since they widely depend on the quality of the skeleton. Thus, to reinforce action recognition, a third class of methods called *hybrid* makes use of more than two modalities. These approaches usually exploit the skeleton information to compute local features using RGB and/or depth images. These local RGB-D based features have shown noteworthy potential [65], [78], [90]. Inspired by this relevant concept which aims at computing local depth-based and RGB-

based features around specific joints, we propose to adapt the same idea to Dense Trajectories which have been proven to be one of the most powerful action representations.

3.3 Localized Trajectories for Action Recognition

To enhance their robustness to irrelevant information, a reformulation of Dense Trajectories is proposed, called Localized Trajectories. The general overview of our approach is illustrated in Figure 3.1. The main idea of this new approach consists in attributing Dense Trajectories a local description: (1) to track the motion in specific and relevant spatial regions of the human body, more specifically around the joints; and (2) to remove redundant and irrelevant motion information, which can negatively affect the classifier performance.

To that end, the pose information through estimated 3D skeletons is used as prior information to estimate an optimal clustering configuration, as depicted in Figure 3.2. Let us consider the human skeleton extracted from RGB-D cameras composed of J joints and let us denote the trajectory of each skeleton joint j as $\mathsf{Q}^j = \{\mathsf{q}_1^j, ..., \mathsf{q}_N^j\}$. Note that we assume that the joints are always well detected. We use the distance proposed by Raptis, Kokkinos and Soatto [91] to group Dense Trajectories of an action around joints. Given a pair of dense and joint trajectories, respectively, P^m and Q^j, which co-exist in the temporal range τ, the spatiotemporal distance between two given trajectories is expressed using:

$$d(\mathsf{P}^m, \mathsf{Q}^j) = \max_{t \in \tau} s_t \cdot \frac{1}{L} \sum_{t \in \tau} r_t, \tag{3.1}$$

such that $s_t = ||\mathsf{p}_t^m - \mathsf{q}_t^j||_2$ is the spatial distance and $r_t = ||(\mathsf{p}_t^m - \mathsf{p}_{t-1}^m) - (\mathsf{q}_t^j - \mathsf{q}_{t-1}^j)||_2$ is the velocity difference between trajectories P^m and Q^j. Then, an affinity matrix is computed between every pair of trajectories $(\mathsf{P}^m, \mathsf{Q}^j)$ using Equation (3.1) as:

$$b(\mathsf{P}^m, \mathsf{Q}^j) = \exp(-d(\mathsf{P}^m, \mathsf{Q}^j)), \tag{3.2}$$

where the measure $d(\mathsf{P}^m, \mathsf{Q}^j)$ penalizes trajectories with significant variation in spatial location and velocity. After a hierarchical clustering procedure which is based on the affinity score [91],

a membership indicator function specifies the cluster G^{j^*} of joint j^* each trajectory belongs to.

$$\mathsf{G}^{j^*} = \{\mathsf{P}^m, \forall m \in \{1, ..., M\} \text{ and } \underset{j \in J}{\arg\min}\, b(\mathsf{P}^m, \mathsf{Q}^j) = j^*\}. \tag{3.3}$$

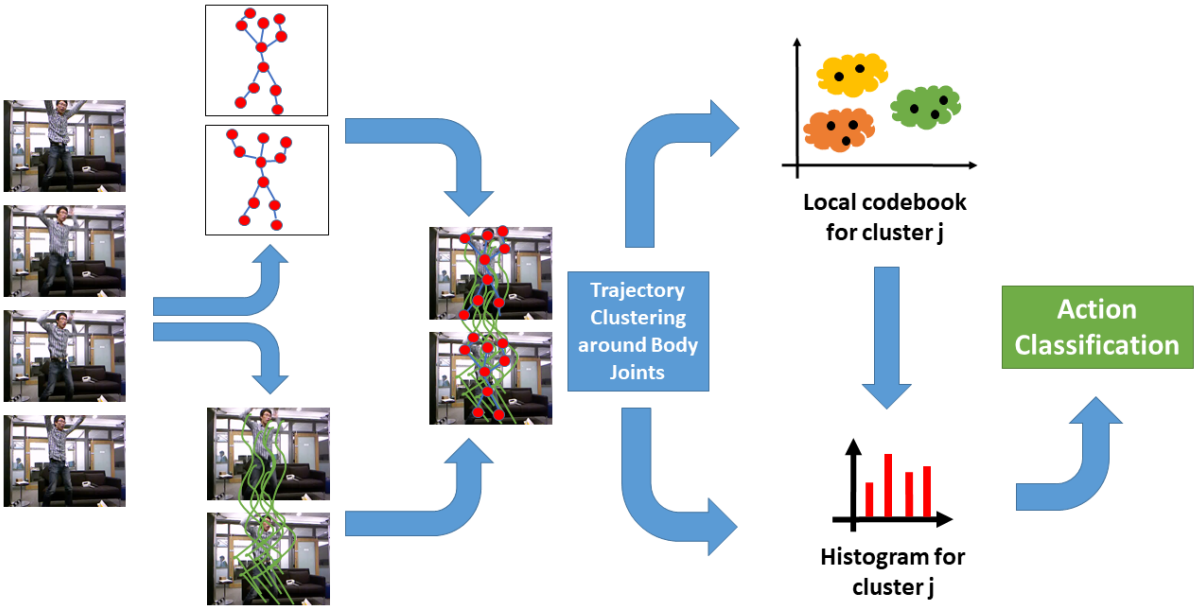


Figure 3.1: Proposed 2D Localized Trajectories approach. From an RGB sequence, Dense Trajectories are generated and, then, clustered around body joints using RGB-D pose information (only 2D information is used). Finally, local codebooks, for every cluster G^j, are constructed for the histogram representation of features. This feature representation is used in both training and testing phases of the classification.

Furthermore, trajectories that are above a certain threshold of distance are rejected. This condition ensures that irrelevant and noise-resulting trajectories will not be considered, e.g., background motion.

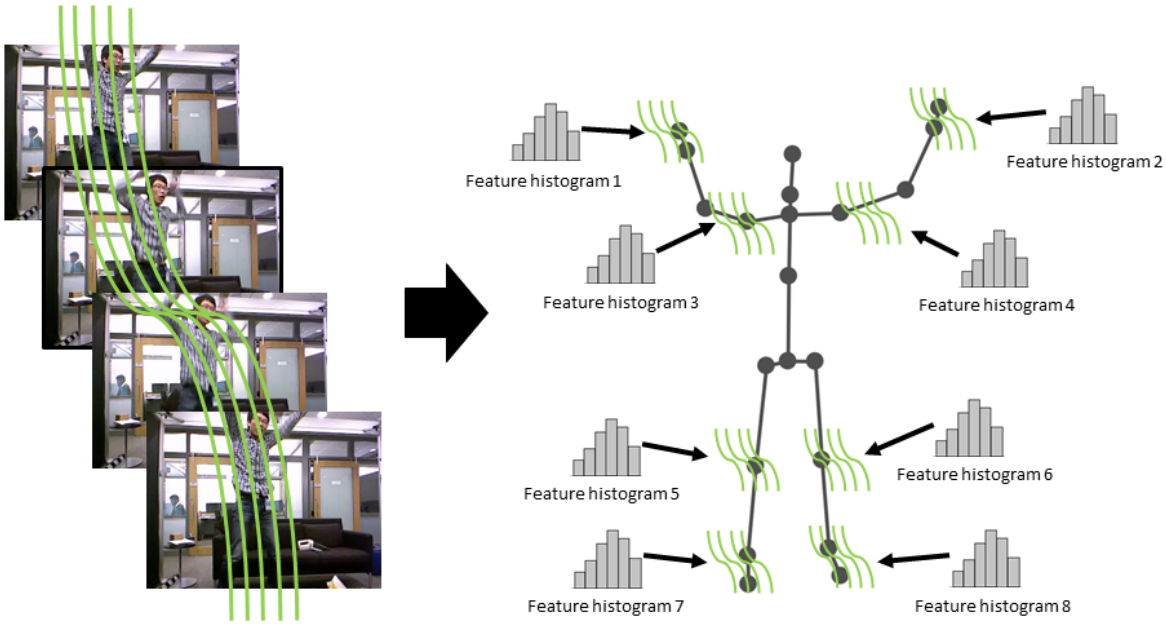


Figure 3.2: The two stages of Localized Trajectories: Left: clustering motion trajectories around body joints; and Right: local features computation which boosts the discriminative power of the original Dense Trajectories concept.

Feature Representation

As discussed in [16], features can be computed along each trajectory, and BoWs can be used to aggregate and encode the information. In such a case, however, a descriptor associated with each trajectory carries no locality information. On the contrary, we propose to exclusively assign trajectories and their corresponding descriptors to trajectory clusters. The main advantage of such a construction is that every trajectory-aligned descriptor does not only capture the spatiotemporal characteristics of the trajectory but it carries its location as well. Thus, we construct a local codebook for each trajectory group G^j. During feature encoding, one histogram is constructed per joint cluster and per descriptor denoted by $\mathcal{H}^j$:

$$\mathcal{H}^j = \left[\mathcal{H}^j_{TSD} | \mathcal{H}^j_{HOG} | \mathcal{H}^j_{HOF} | \mathcal{H}^j_{MBH}\right]. \tag{3.4}$$

The subscripts of the individual histograms identify the type of descriptors. In our case, the Trajectory-Shape Descriptor (TSD) [16], the Histogram of Oriented Gradients (HOG) [70], the Histogram of Optical Flow (HOF) [71], and the Motion Boundary Histogram (MBH) [16] are used. Finally, an action video is represented by the concatenation of the individual joint histograms in a final histogram $\mathcal{H}$, as follows:

$$\mathcal{H} = \bigcup_{j=1}^{J} \mathcal{H}^j. \tag{3.5}$$

3.4 3D Trajectories and Aligned Descriptors

Dense Trajectories, generated via optical flow, offer adequate performance when used for tracking movements that are lateral to the image plane. However, they struggle to track motion that happens radially, due to the fact that the occurring motion is subtle with respect to the 2D image plane. Consequently, in this section, we propose to extend localized Dense Trajectories to RGB-D input video stream by replacing optical flow with scene flow. The generated 3D trajectories are suitable for tracking motion in both lateral and radial directions, as illustrated in Figure 3.3.

3.4.1 Scene Flow Estimation Using RGB-D Data

To generalize the concept of Dense Trajectories from 2D to 3D, we propose to make use of the 3D extension of optical flow, called scene flow. Thanks to the emergence of RGB-D cameras, numerous approaches have been proposed to estimate scene flow from depth maps, e.g., the Primal-Dual Framework for Real-Time Dense RGB-D Scene Flow (PD-Flow) algorithm [14], the Dense semi-rigid scene flow estimation [92] and the Layered RGBD scene flow estimation [93].

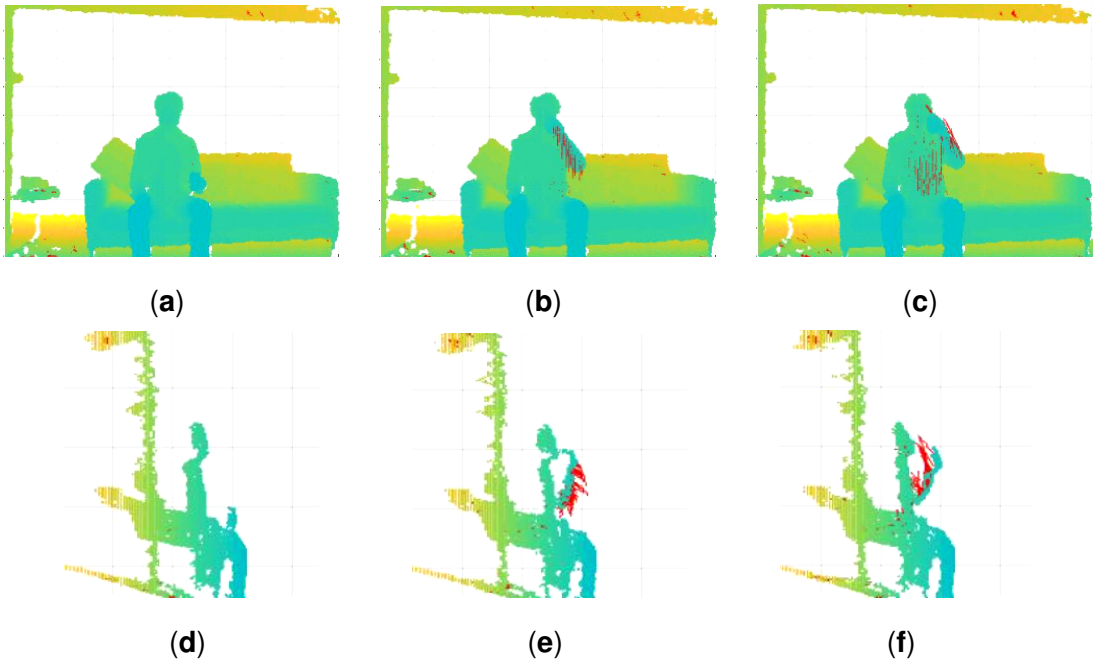


Figure 3.3: Scene flow-generated motion trajectories. Three phases of the same action are illustrated: (**a**–**c**) the frontal view of a subject drinking water is displayed as a point cloud, along with the corresponding motion trajectories in red; and (**d**–**f**) the same sequence is illustrated from the side. The capture of both lateral and radial motion shape is clearly depicted.

The scene flow $\mathbf{\Omega}$ is linearly dependent on the depth motion field $\mathbf{S} = (\mathbf{u}, \mathbf{v}, \mathbf{w})$, where $\mathbf{w}$ is the range flow. It is computed by mapping $\mathbf{S}$ to the 3D world coordinate system as below:

$$\mathbf{\Omega} = \begin{pmatrix} \frac{Z}{f_x} & 0 & \frac{X}{Z} \\ 0 & \frac{Z}{f_y} & \frac{Y}{Z} \\ 0 & 0 & 1 \end{pmatrix} \mathbf{S}^T, \tag{3.6}$$

where f_x and f_y are the camera focal lengths, and X, Y, Z are the 3D world coordinates of a specific point. On the other hand, the depth motion fields are estimated as a solution of a global variational problem, defined as:

$$\min_{\mathbf{S}} \{E_D(\mathbf{S}) + E_R(\mathbf{S})\}, \tag{3.7}$$

where $E_D(\mathbf{S})$ is a data term defined as the combined measure of the photometric and geometric

inconsistency of successive depth and intensity images and $E_R(\mathbf{S})$ is defined as a regularizer term. Multiple approximations of $\mathbf{S}$ exist based, for example, on decoupling the radial motion $\mathbf{w}$ from the lateral motion $(\mathbf{u}, \mathbf{v})$ [28], [29].

We choose PD-Flow [14] to estimate a dense scene flow field from an RGB-D video stream, since it has been shown to be one of the fastest and most accurate algorithms. In PD-Flow, the data term $E_D(\mathbf{S})$ is defined as:

$$E_D(\mathbf{S}) = \int_{\mathbf{\Omega}} \left| \left(\mathsf{V}^t_{x,y} - \mathsf{V}^{t+1}_{x+\mathbf{u},y+\mathbf{v}}\right) + \beta(x,y)\left(\mathbf{w} - \mathsf{Z}^{t+1}_{x+\mathbf{u},y+\mathbf{v}} + \mathsf{Z}^t_{x,y}\right) \right| dxdy \tag{3.8}$$

where Z is the depth sequence, and $\beta(x, y)$ is a a positive function that weights geometric consistency $\left(\mathbf{w} - \mathsf{Z}^{t+1}_{x+\mathbf{u},y+\mathbf{v}} + \mathsf{Z}^t_{x,y}\right)$ against brightness constancy $\left(\mathsf{V}^t_{x,y} - \mathsf{V}^{t+1}_{x+\mathbf{u},y+\mathbf{v}}\right)$. The regularization term, on the other hand, is defined as:

$$E_R(\mathbf{S}) = \lambda_{\mathsf{V}} \int_{\mathbf{\Omega}} \left| \left(r_x \frac{\partial \mathbf{u}}{\partial x}, r_y \frac{\partial \mathbf{u}}{\partial y}\right) \right| + \left| \left(r_x \frac{\partial \mathbf{v}}{\partial x}, r_y \frac{\partial \mathbf{v}}{\partial y}\right) \right| dxdy + \lambda_{\mathsf{Z}} \int_{\mathbf{\Omega}} \left| \left(r_x \frac{\partial \mathbf{w}}{\partial x}, r_y \frac{\partial \mathbf{w}}{\partial y}\right) \right| dxdy \tag{3.9}$$

where

$$r_x = \frac{1}{\sqrt{\frac{\partial X^2}{\partial x} + \frac{\partial Z^2}{\partial x}}}, \; r_y = \frac{1}{\sqrt{\frac{\partial Y^2}{\partial y} + \frac{\partial Z^2}{\partial y}}} \tag{3.10}$$

and $\lambda_{\mathsf{V}}, \lambda_{\mathsf{Z}}$ are constant weights.

3.4.2 3D Localized Trajectories

To estimate the 3D trajectories using scene flow, we start by uniformly sampling points from the 2D image grid. In this context, we define pixel coordinates as (x, y). Similar to Wang, Klaser, Schmid and Liu [16], we reject points belonging to homogeneous image areas without any structure. Next, each of the sampled points are mapped to a standard 3D world coordinate

system using the inverse of the intrinsic camera parameter matrix as described below:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} \frac{(x - c_x)D}{f_x} & \frac{(y - c_y)D}{f_y} & D \end{pmatrix}^T, \tag{3.11}$$

where c_x and c_y are the image plane central point coordinates, f_x and f_y are the respective x and y components of the focal length and $D = \mathsf{Z}^t_{x,y}$ is the depth value. Subsequently, trajectories of the mapped 3D points are estimated using Equation (2.6), except that the motion field is now based on an estimated scene flow. The estimated *3D Dense Trajectories* are denoted as:

$$(X_{t+1}, Y_{t+1}, Z_{t+1}) = (X_t, Y_t, Z_t) + \mathbf{\Omega_t}, \tag{3.12}$$

where $\mathbf{\Omega_t}$ is the scene flow field. Correspondence between estimated 3D points, with scene flow, and image pixels is derived by solving Equation (3.11) in terms of $(x, y, D)^T$.

The above procedure is repeated recurrently until each of the 3D trajectories reach the fixed temporal length we have set. Similar to Wang, Klaser, Schmid, and Liu [16], trajectories with a high total variation that corresponds to sudden displacements or small overall spatial length are considered irrelevant and are removed.

In depth maps, texture information is not present. Thus, in our case, only motion descriptors are considered. Three types of descriptors are used: *3D Trajectory Shape Descriptor (3DTSD), Histogram of Scene Flow [94] (HSF)*, and *3D Motion Boundary Histogram (3DMBH)*. 3DTSD is based on the original idea of the TSD for Dense Trajectories [16]. For each trajectory, the normalized displacement vector is computed as:

$$\mathcal{S}^m_{3DTSD} = \frac{(\Delta \mathsf{p}^m_t ... \Delta \mathsf{p}^m_{t+L-1})}{\sum_{i=t}^{t+L-1} ||\Delta \mathsf{p}^m_i||}, \tag{3.13}$$

where $\Delta \mathsf{p}^m_t = \mathsf{p}^m_{t+1} - \mathsf{p}^m_t$. The HSF descriptor captures the orientation and the magnitude of the local scene flow field. For a spatiotemporal volume aligned around a 3D trajectory, the orientation of the 3D displacement is calculated using the azimuth θ_{xy} and elevation θ_{yz} angles formed by

consecutive points as:

$$\theta_{xy} = \frac{\Delta Y_t}{\Delta X_t} \quad \text{and} \quad \theta_{yz} = \frac{\Delta Z_t}{\Delta Y_t}. \tag{3.14}$$

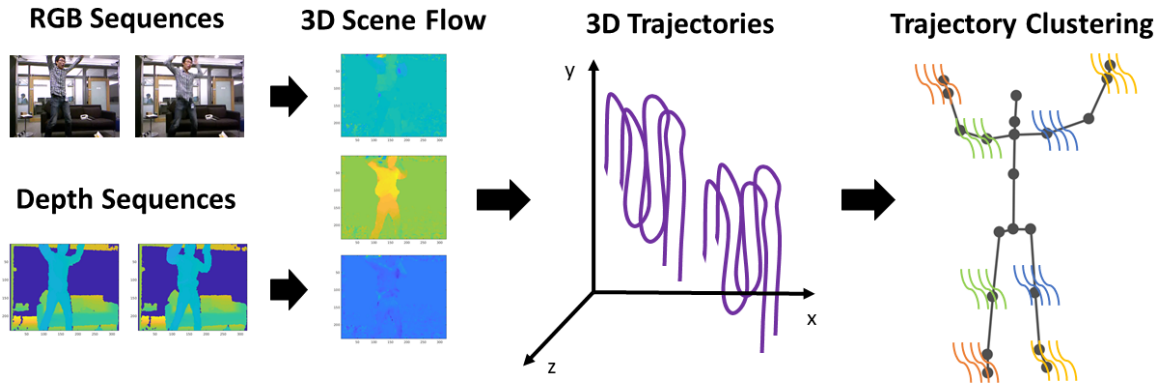


Figure 3.4: Computation steps of 3D Localized Trajectories. RGB and depth modalities are used for the estimation of the scene flow constituted of three components. Then, using the estimated scene flow, 3D Trajectories are generated. Finally, the latter are clustered around 3D body joints. Different color has been used for each cluster.

For the histogram construction, the 4D space is quantized into a fixed number of bins. Similarly, the 3DMBH is based on the same idea as HSF. First, the derivative of the scene flow field is computed and, then, for every pair of coordinates, the orientation angle is estimated using the azimuth θ_{uv} and elevation θ_{vw} angles as:

$$\theta_{uv} = \frac{\Delta \mathbf{v}_t}{\Delta \mathbf{u}_t} \quad \text{and} \quad \theta_{vw} = \frac{\Delta \mathbf{w}_t}{\Delta \mathbf{v}_t}. \tag{3.15}$$

3D Trajectories are adapted to *3D Localized Trajectories* by following the procedure described in Section 3.3, as depicted in Figure 3.4. Similarly as before, we propose to enhance the discriminative power of 3D Trajectories by grouping them around 3D body joints. Hence, Equations (3.1)–(3.3) are adapted accordingly to incorporate all three dimensions of 3D trajectories P^m_{3D} and 3D joint trajectories Q^j_{3D}. Then, during feature encoding, every histogram of joint clusters G^j

defined in Equation (3.4) is modified to include the descriptors used in this context, becoming:

$$\mathcal{H}^j = \left[\mathcal{H}^j_{3DTSD} | \mathcal{H}^j_{HSF} | \mathcal{H}^j_{3DMBH}\right]. \tag{3.16}$$

3.4.3 Feature Selection for Codebook Construction

While 3D Trajectories are advantageous in capturing radial motion, they are notably noisier compared to Dense Trajectories, due to the scene flow estimation. As a result, the quality of the codebooks is degraded, unfavorably affecting the general performance of the proposed approach. This is mainly caused by the random selection of features from the training set [16] which are used to compute the final codebook. To reduce the impact of noise, we propose to select features according to the classifier *confidence* and *ambiguity* probabilistic metrics. Confidence is the classifier's ability to quantify the reliability of its predictions, while ambiguity indicates the number of classes the classifier outputs for every prediction. The confidence ζ and ambiguity ψ metrics are defined as:

$$\zeta = \underset{m \in M_r}{\text{median}}(log(Pr(l_m = a|\mathcal{F}^m))), \quad \text{and} \quad \psi = \sum_{m \notin M_r} (log(Pr(l_m = a|\mathcal{F}^m))), \tag{3.17}$$

where $Pr(l_m = a|\mathcal{F}^m)$ is the posterior probability of label a given feature $\mathcal{F}^m$.

Hence, the classifier is trained several times with diverse sets of random training features which are, also, used to generate different codebooks. In our experiments, we chose 100 sets of training features. Then, based on the computed metrics, we select the codebook which provides the highest confidence score and lowest ambiguity. If the codebook with the highest confidence is different from the one with the lowest ambiguity, we randomly select one of them. Our concept is inspired by the joint selection proposed in [65].

3.5 Experimental Evaluation

We evaluated the proposed approaches on five challenging datasets: MSR DailyActivity3D [65], Online RGB-D (ORGBD) [95], G3D Gaming [96], Watch-n-Patch [97] and KARD datasets [98].

First, a brief description of each dataset is given followed by the presentation of the experimental setups. Then, the obtained results are reported and extensively analyzed.

3.5.1 Datasets and Experimental Settings

The first dataset used for the experimental evaluation is the MSR DailyActivity 3D dataset [65]. In this dataset, 10 actors perform 16 daily activities, which in some cases involve human-object interaction. The dataset was captured by the Kinect v1 device, providing therefore RGB, depth, and skeleton modalities. A distinctive characteristic of this dataset is that every actor repeats each action twice in both sitting and standing positions. For the experiments, we followed a cross-splitting protocol as in [65], where half of the subjects were used for training and the rest for testing.

The second dataset is called Online RGB-D Action (ORGBD) [95]. It can be used for both action recognition and action detection and includes seven common types of human-object interaction related to the living room environment. Three sets of video sequences were collected using a Kinect sensor. Thus, RGB, depth and skeleton modalities are available. The first set was captured in the context of action recognition in the same environment, whereas the second set was acquired for cross-environment action recognition and the third for on-line action detection. The splitting protocol requires two-fold cross-validation for the same-environment scenario, whereas, for cross-environment action recognition, training and testing sets should include different environments [95].

One challenging dataset used for the evaluation is the G3D Gaming Action Dataset [96]. This Kinect-acquired dataset can be used for both action recognition and temporal action detection. It consists of 10 subjects performing 20 gaming actions which are grouped into seven gaming scenarios: Fighting, playing golf, playing tennis, bowling, first-person shooter, driving a car, and miscellaneous. The first five actors were used for training and the rest were used for testing [96].

Watch-n-Patch [97] dataset, which was introduced by Cornell University, was also utilized. This dataset includes 21 types of actions (10 in an office and 11 in a kitchen) which involve interactions with 23 types of objects. Seven subjects perform 2–7 actions in each of the 458

videos. The dataset was recorded using a Kinect v2 camera. This dataset distinguishes itself by a high intra-class variability since the subjects perform different combinations of actions by ordering them differently each time. For the experiments, we used the provided splitting protocol proposed in [97], where, for every environment, almost half of the videos were used for training and the rest for testing.

The last dataset used for evaluation is called Kinect Activity Recognition Dataset (KARD) [98]. It contains 18 action classes which are performed by 10 subjects (nine males and one female). Half of the subjects were used for training and half for testing, as proposed in [98]. The dataset was captured by a Kinect device and consequently contains the three RGB-D modalities: RGB images, depth maps, and 3D skeletons.

3.5.2 Implementation Details

For extracting Dense Trajectories and features from videos, we used the implementation provided by the authors in [16] (`https://lear.inrialpes.fr/people/wang/dense_trajectories`). The trajectory temporal length was fixed to 15 frames. The features were computed on a spatiotemporal volume of $32 \times 32 \times 15$ aligned on the trajectory, as suggested in [16]. This volume was further divided into $2 \times 2 \times 3$ cells, where the histograms of the descriptors were computed. In the case of 3D trajectories, we used the same parameters for the spatiotemporal volume. The number of histogram bins for the 2D trajectories was set to eight for HOG and MBH descriptors and nine for HOF descriptor, whereas for 3D trajectories case we used nine-bin histograms for every descriptor. The distance threshold for each trajectory was set to 0.02. Moreover, a linear SVM was employed for classification.

For each one of the aforementioned datasets, we report the obtained recognition accuracy using the proposed Localized Trajectories and compare it to the classical Dense Trajectories and recent state-of-the-art approaches. In the following, we denote the original dense trajectory approach [16] by Dense Trajectories. We refer to the 2D proposed approach as 2D Localized Trajectories. Similarly, the proposed 3D extension of the classical and the local Dense Trajectories are, respectively, called 3D Dense Trajectories and 3D Localized Trajectories.

The number of skeleton joints defines the number of clusters. Subsequently, in the MSR

DailyActivity3D, ORGBD, and G3D datasets, the skeletons are composed of 20 joints, while, in Watch-n-Patch and KARD datasets, they are, respectively, formed by 25 and 15 joints. We also empirically chose to use 2000 random trajectories per video to construct the codebooks and 128 words per cluster and per descriptor for every dataset.

3.5.3 Performance of 2D Localized Dense Trajectories

In this subsection, an analysis of the obtained results is provided. First, we compare the performance of our approach against Dense Trajectories and other state-of-the-art methods. Later, we discuss some of the limitations of 2D Localized Trajectories.

Table 3.1: Mean accuracy of recognition (%) on MSR DailyActivity 3D dataset for Dense Trajectories and 2D Localized Trajectories approaches against literature.

Method	**Mean Accuracy**
Dynamic Temporal Warping [99]	54.0%
Local HON4D [77]	80.0%
Moving Pose [30]	73.8%
3D Trajectories [64]	72.0%
Skeleton only [65]	68.0%
Skeleton and LoP [65]	85.8%
Naive-Bayes-NN [87]	73.8%
TriViews [100]	83.8%
Skeletal Shape Trajectories [88]	70.0%
Long-Term Motion Dynamics [101]	86.9%
Spatiotemporal Multi-fusion [102]	94.1%
Dense Trajectories [16]	64.4%
3D Dense Trajectories (ours)	48.8%
2D Localized Trajectories (ours)	74.4%
3D Localized Trajectories (ours)	76.3%

2D Localized Dense Trajectories vs. Dense Trajectories

Since the aim of this work is to improve the discriminative power of classical Dense trajectories, we start by comparing our proposed 2D Localized Dense Trajectories with them. The results obtained on the five benchmarks prove the superiority of the proposed 2D Localized Trajectories. As reported in Tables 3.1–3.5, 2D Localized Dense Trajectories improve the accuracy by 10%, 7.7%, 3.1%, 16%, 13.8% and 0.4% on MSR DailyAvtivity3D, G3D, ORGB (same-environment settings), ORGB (cross-environment settings), Watch-n-Patch and KARD, respectively, compared to the classical Dense Trajectories [16].

Table 3.2: Mean accuracy of recognition (%) on G3D dataset for Dense Trajectories and 2D Localized Trajectories approaches against literature.

Method	**Mean Accuracy**
Dynamic Time Wrapping [103]	86.3%
Weighted Graph Matching [104]	89.2%
Adaptive Graph Kernels [105]	84.8%
Histogram [106]	79.5%
LPP and BoW [107]	87.5%
Spatial Graph Kernels [108]	95.7%
DL on Lie Group [21]	89.1%
Rolling Rotations [109]	88.0%
Dense Trajectories [16]	80.1%
Skeleton and LoP [65]	87.3%
2D Localized Trajectories (ours)	87.8%

Table 3.3: Mean accuracy of recognition (%) on ORGBD dataset for Dense Trajectories and 2D Localized Trajectories approaches against literature in both Same and Cross Environment Settings.

Method	**Mean Accuracy**	
	Same Env.	**Cross Env.**
Moving Pose [30]	38.4%	28.5%
Eigenjoints [87]	49.1%	35.7%
DSTIP and DCSF [79]	61.7%	21.5%
Skeleton and LoP [65]	66.0%	59.8%
Pairwise joint distance [95]	63.3%	–
Orderlet [95]	71.4%	–
Motion decomposition [110]	80.9%	–
Dense Trajectories [16]	64.3%	43.8%
2D Localized Trajectories (ours)	67.4%	59.8%
3D Localized Trajectories (ours)	64.5%	38.4%

Table 3.4: Mean accuracy of recognition (%) on Watch-n-Patch in both kitchen and office settings for Dense Trajectories and 2D Localized Trajectories approaches.

Method	**Mean Accuracy**
Dense Trajectories—office [16]	68.8%
Dense Trajectories—kitchen [16]	56.2%
2D Localized Trajectories—office (ours)	71.1%
2D Localized Trajectories—kitchen (ours)	81.5%

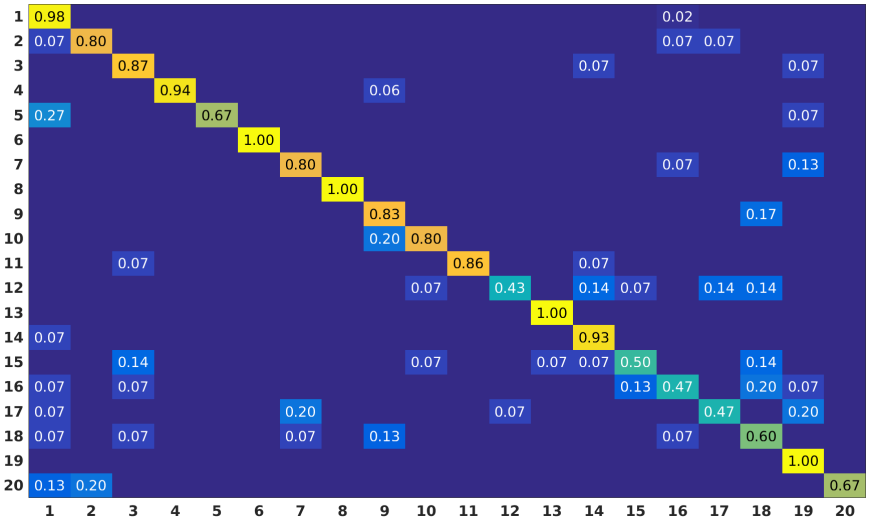
Table 3.5: Mean accuracy of recognition (%) of Dense Trajectories and 2D Localized Trajectories approaches on KARD dataset.

Method	**Mean Accuracy**
JTMI, LBP and FLD [111]	98.5%
JTMI and Gabor features [112]	96.0%
HOJ3D [19]	95.3%
EigenJoints [87]	96.2%
Dense Trajectories [16]	97.8%
2D Localized Trajectories (ours)	98.2%

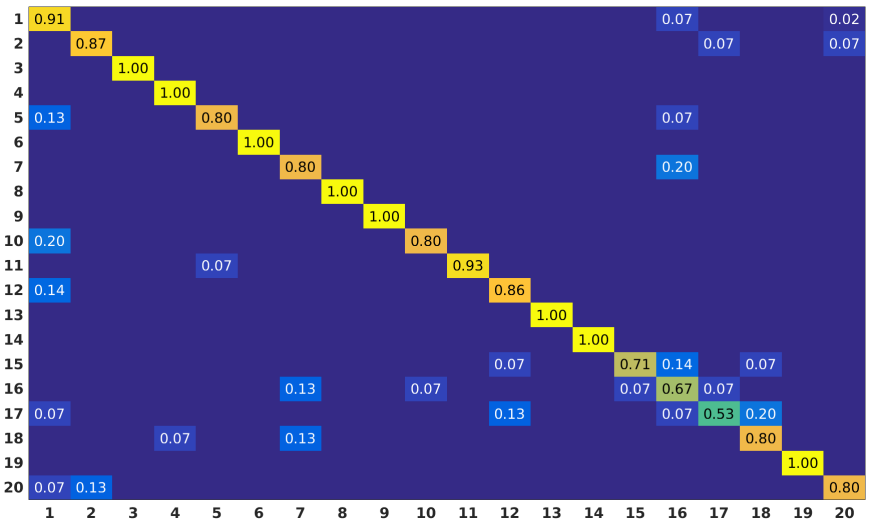
The reported results reflect the ability of 2D Localized Trajectories to distinguish actions with similar motion patterns that are performed by different body parts. This is shown in various cases when comparing confusion matrices obtained for 2D Localized Trajectories and Dense Trajectories. For instance, in the confusion matrices of the G3D dataset in Figure 3.5, 2D Localized Trajectories boost the performance of the following action pairs: Punch Right–Punch Left and Kick Right–Kick Left. In addition, in the same dataset, the recognition accuracy of both Tennis Swing Backhand and Throwing Bowling Ball activities which include similar motion shapes is improved by 20% and 6%, respectively. Furthermore, the accuracy of Drinking and Reading Book classes in the ORGBD dataset is increased by 33% and 31%, respectively (see Figure 3.6).

Another example of this enhancement can be the pair of actions Defend and Aim and Fire Gun in the G3D dataset. The motion shapes of both action classes are similar since both of them include arm raising. Nevertheless, the first is performed using both arms and the second by using only one arm. As we can see in Figure 3.5, the performance obtained for the action Defend is improved by 13%, and the confusion with the action Aim and Fire Gun is reduced by 14%. In addition, in the same dataset, actions Wave and Clap have similar lateral motion, and using the classical Dense Trajectories made their distinction challenging. However, with the use of 2D Localized Trajectories, motion trajectories were assigned to only one hand cluster in Wave action and to both hands in Clap action, reducing the confusion between these classes.

This results in an accuracy boost of 13% in Wave class, as shown in Figure 3.5.

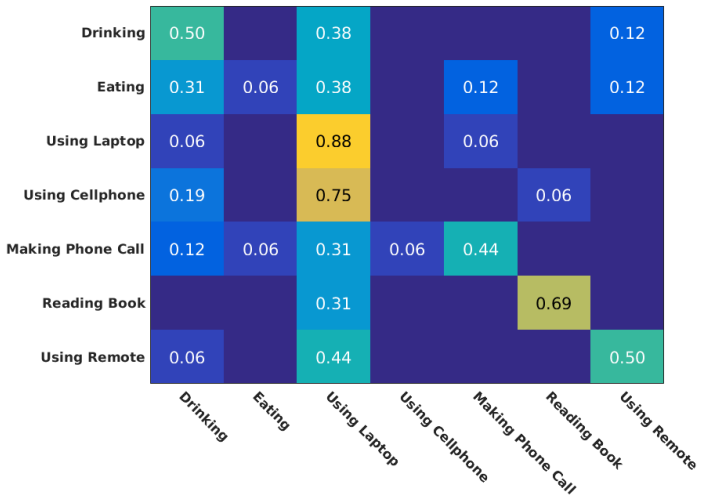


(**a**)

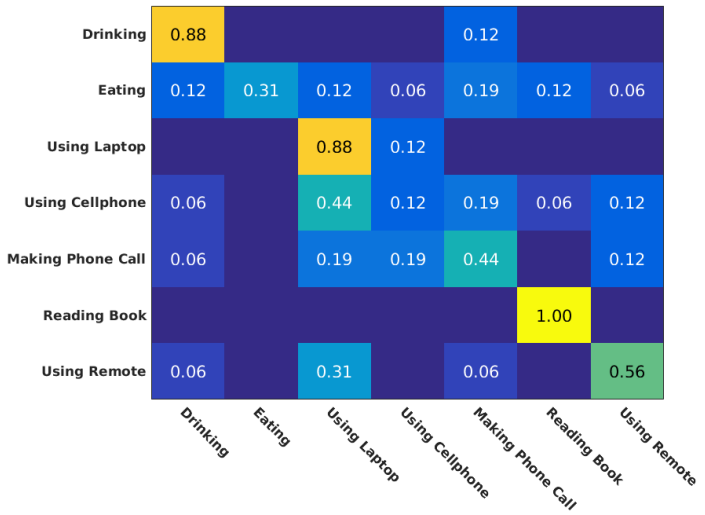


(**b**)

Figure 3.5: Confusion matrices obtained for Dense Trajectories (**a**) and 2D Localized Trajectories (**b**) approaches on G3D dataset. Actions list: (1) Aim and Fire Gun; (2) Clap; (3) Climb; (4) Crouch; (5) Defend; (6) Flap; (7) Golf Swing; (8) Jump; (9) Kick Left; (10) Kick Right; (11) Punch Left; (12) Punch Right; (13) Run; (14) Steer; (15) Tennis Serve; (16) Tennis Swing Backhand; (17) Tennis Swing Forehand; (18) Throw Bowling Ball; (19) Walk; and (20) Wave.



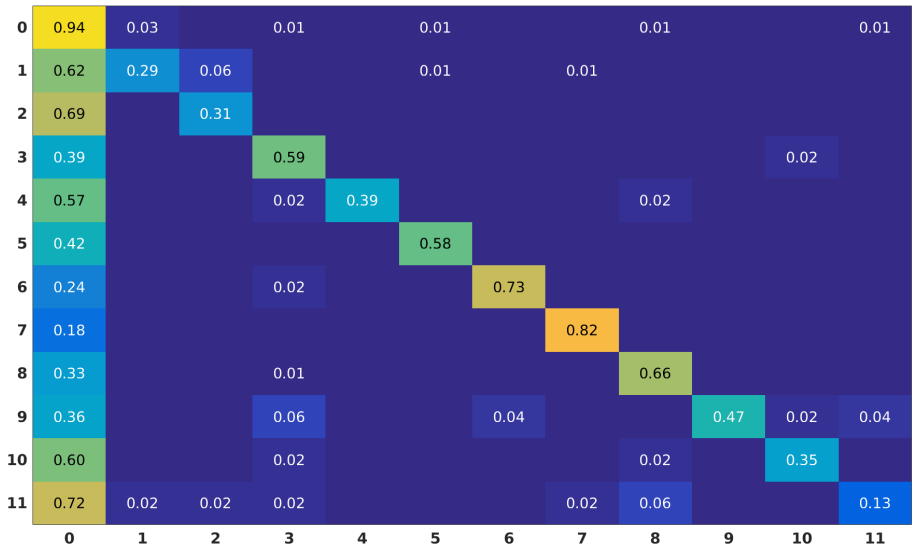
(**a**)



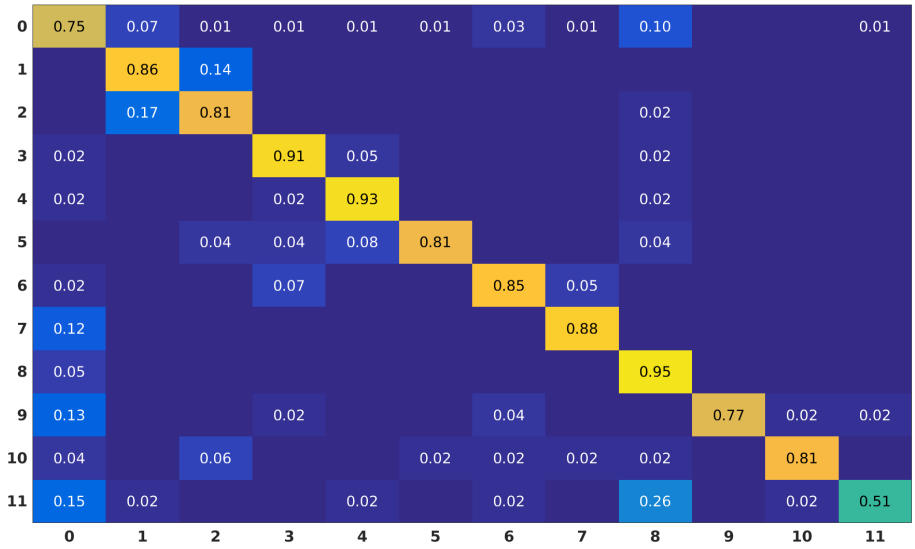
(**b**)

Figure 3.6: Confusion matrices obtained for Dense Trajectories (**a**) and 2D Localized Trajectories (**b**) approaches (ORGBD).

Moreover, in scenarios with full-body motion, such as the kitchen environment in the Watch-n-Patch dataset, 2D Localized Trajectories outperform the Dense Trajectories approach, as shown in Figure 3.7. Clusters isolate specific motion of body parts, therefore motion patterns related to the action can be identified more effectively.



(**a**)



(**b**)

Figure 3.7: Confusion matrices obtained for Dense Trajectories (**a**) and 2D Localized Trajectories (**b**) approaches (Watch-n-Patch) in the kitchen environment. The action labels are: (0) no-action; (1) fetch-from-fridge; (2) put-back-to-fridge; (3) prepare-food; (4) microwaving; (5) fetch-from-oven; (6) pouring; (7) drinking; (8) leave-kitchen; (9) fill-kettle; (10) plug-in-kettle; and (11) move-kettle.

Comparison with 3D-Based State-of-the-Art Approaches

Our 2D Localized Trajectories approach has shown competitive performance compared to 3D-based state-of-the-art approaches. In the ORGBD dataset, we achieve the third best performance in the same-environment setting (Table 3.3). We manage to match the state-of-the-art results of [65] in the cross-environment settings and, at the same time, increase the mean accuracy by 16% over the Dense Trajectories.

In Watch-n-Patch dataset, the 2D Localized Trajectories improved the performance of the Dense Trajectories by 2.3% in the office environment and by 25.3% in the kitchen environment, as illustrated in Table 3.4. The discriminative power of our approach boosts the performance of every action class, especially in the kitchen environment, as can be observed in Figure 3.7. On this dataset, we only compared our work with Dense Trajectories. To the best of our knowledge, there is no work in the literature reporting offline action recognition accuracy on it, since this dataset was initially acquired for action detection.

In the KARD dataset, our approach based on the 2D Localized Trajectories outperforms almost all state-of-the-art approaches, with a score of 98.2%, except JTMI, LBP, and FLD [111], which reaches a slightly superior score with only 0.3% difference.

The 2D Localized Trajectories approach offers the second largest improvement on the MSR DailyActivity3D dataset, by 10% compared to Dense Trajectories, as depicted in Table 3.1.

Finally, as reported in Table 3.2, our method achieves a competitive performance on the G3D dataset without the need for 3D information.

Despite the performance of 2D Localized Trajectories, it can be noted that some state-of-the-art approaches achieve better performance (e.g., [21], [65], [77], [100]–[102], [104], [108], [109], [111]), as reported in Tables 3.1–3.3 and 3.5. We remark that most of these state-of-the-art approaches rely on 3D features [21], [65], [77], [100], [102], [104], [108], [109], [111]. Indeed, 3D descriptors are directly extracted from depth maps and/or 3D skeleton sequences. In contrast, our method computes only RGB features around the extracted 2D trajectories. The 2D information of 3D skeletons is only used to cluster the trajectories. Moreover, some of these 3D approaches (e.g., [100], [102]) are even more reinforced with the use of fusion strategies. For instance, while we use only four 2D descriptors around 2D Localized Trajectories,

the two aforementioned approaches [100], [102] use five descriptors each. Finally, methods employing deep learning models (e.g., [21], [101]) can reach higher performance, since they learn appropriate features, instead of hand-crafting them. As a further investigation, it would be interesting to use a more important number of 3D features and define new strategies to fuse deeply learned and/or hand-crafted features computed around trajectories.

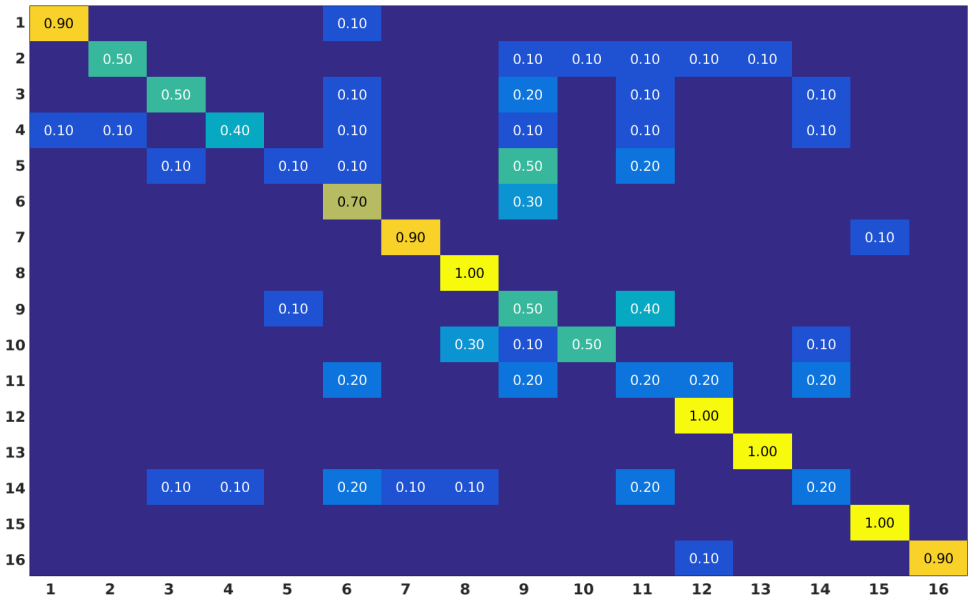
Limitations of 2D Localized Dense Trajectories

Despite its strong performances, 2D Localized trajectories action representation suffers from two limitations. First, the 2D Localized Trajectories approach presents low performance when the motion amount is small. This attribute is inherited from the Dense Trajectories approach and is clearly depicted in action classes such as Call Cellphone in both MSR DailyActivity 3D and ORGBD, as shown in Figures 3.8 and 3.6, respectively, and Write on a Paper in MSR DailyActivity 3D. Nonetheless, Sit Still class achieves adequate performance with the use of 2D Localized Trajectories, since it is an action class with almost no motion.

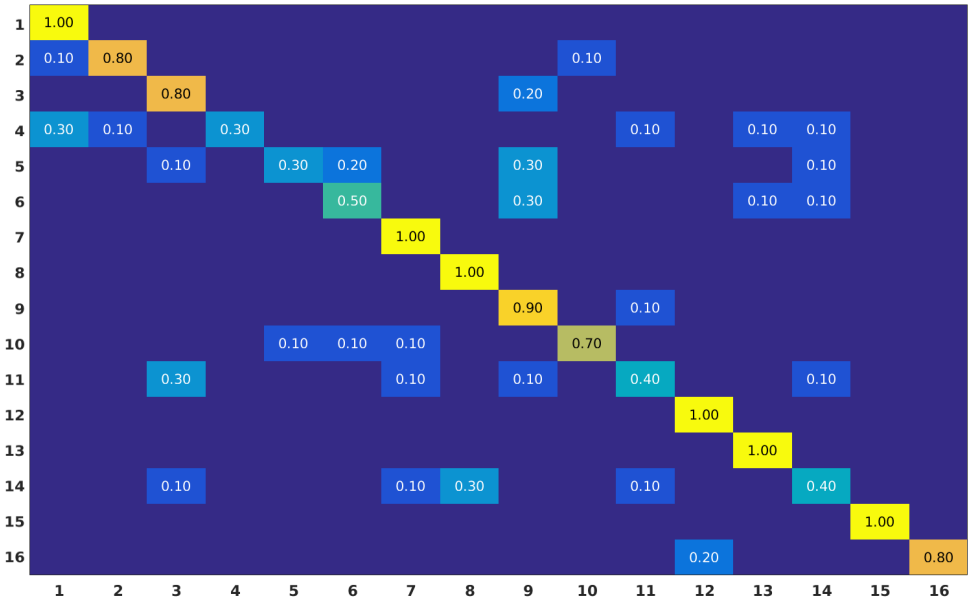
Second, the 2D Localized Trajectories approach does not capture radial motion sufficiently. Action classes such as Playing the guitar in the MSR DailyActivity3D dataset include a notable amount of radial motion and the accuracy results are consequently low, as demonstrated in Figure 3.8a,b. For that reason, as mentioned above, the proposed 3D Localized Trajectories presents as a good alternative to solve these two issues. The performance of the 3D Localized Trajectories is reported in the next section.

3.5.4 Performance of 3D Localized Trajectories

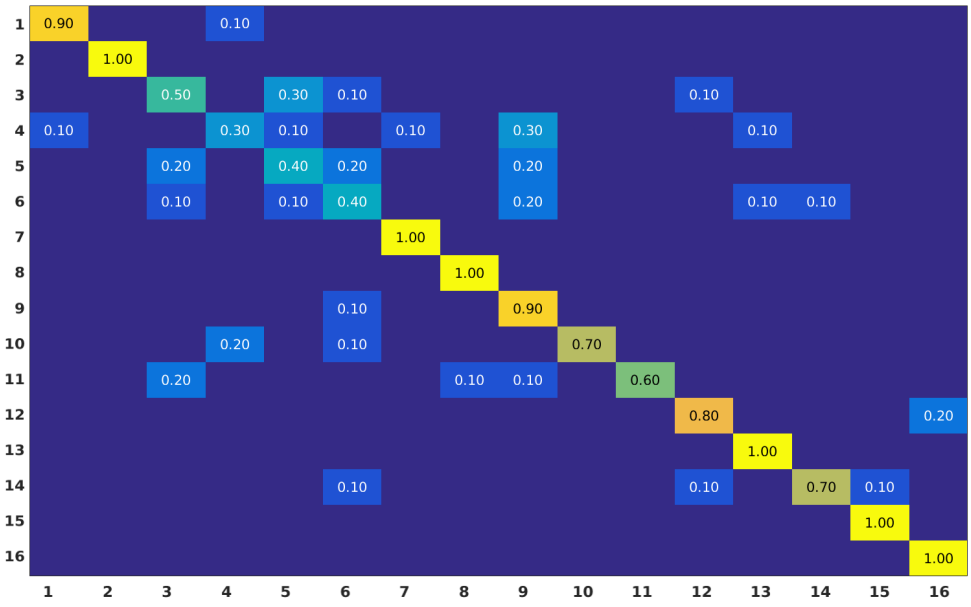
The proposed 3D Localized trajectories approach was evaluated on MSR DailyActivity3D and ORGBD datasets. The results reported in Figure 3.1 show its superiority against Dense Trajectories and 2D Localized Trajectories. In fact, the accuracy of Dense Trajectories and 2D Localized Trajectories are improved by 1.9% and 11.9%, respectively. However, the reported results in Table 3.3 are lower than the 2D Localized Trajectories in both settings, by 2.9% and 21.4%.


1 0.90 0.10
2 0.50 0.10 0.10 0.10 0.10 0.10
3 0.50 0.10 0.20 0.10 0.10
4 0.10 0.10 0.40 0.10 0.10 0.10 0.10
5 0.10 0.10 0.10 0.50 0.20
6 0.70 0.30
7 0.90 0.10
8 1.00
9 0.10 0.50 0.40
10 0.30 0.10 0.50 0.10
11 0.20 0.20 0.20 0.20 0.20
12 1.00
13 1.00
14 0.10 0.10 0.20 0.10 0.10 0.20 0.20
15 1.00
16 0.10 0.90
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

(**a**)


1 1.00
2 0.10 0.80 0.10
3 0.80 0.20
4 0.30 0.10 0.30 0.10 0.10 0.10
5 0.10 0.30 0.20 0.30 0.10
6 0.50 0.30 0.10 0.10
7 1.00
8 1.00
9 0.90 0.10
10 0.10 0.10 0.10 0.70
11 0.30 0.10 0.10 0.40 0.10
12 1.00
13 1.00
14 0.10 0.10 0.30 0.10 0.40
15 1.00
16 0.20 0.80
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

(**b**)



(**c**)

Figure 3.8: Confusion matrices obtained for (**a**) Dense Trajectories, (**b**) 2D Localized Trajectories and (**c**) 3D Localized Trajectories approaches on MSR DailyActivity 3D dataset. Actions list: (1) Drink; (2) Eat; (3) Read book; (4) Call cellphone; (5) Write on a paper; (6) Use laptop; (7) Use vacuum cleaner; (8) Cheer up; (9) Sit still; (10) Toss paper; (11) Play game; (12) Lie down on a sofa; (13) Walk; (14) Play guitar; (15) Stand up; and (16) Sit down.

The performance improvement happens mainly because of the inclusion of depth information in 3D trajectories. This helps in distinguishing actions that are performed radially with respect to the camera. The latter is particularly reflected in the confusion matrix of MSR DailyActivity 3D dataset in Figure 3.8, where actions such as play game and play guitar are more effectively discriminated using 3D information. The reported accuracies for the actions play game and play guitar are significantly improved. In particular, from 20% and 20% using Dense Trajectories and 40% and 40% using 2D Localized Trajectories, the accuracy climbed to 60% and 70% with the use of 3D Localized Trajectories, respectively.

Nevertheless, the results reported in Table 3.3 can be explained by two facts: (a) Current

scene flow estimation algorithms are still very sensitive to noise in comparison to optical flow. Thus, since this dataset is slightly noisier than MSR DailyActivity3D, it is predictable to have less impressive results. However, novel approaches for a more robust estimation of scene flow are being currently investigated with the expectation of improved performance in the future. (b) 3D Localized Trajectories are more efficient than 2D ones, especially in the presence of radial motion. However, the ORGBD dataset does not incorporate actions involving a significant amount of radial motion. On the other hand, we can notice that some state-of-the-art methods (e.g., [77], [95], [100]–[102], [110]) remain more accurate than the proposed 3D Localized Trajectories, as shown in Tables 3.1 and 3.3. As explained in Section 3.5.3, the methods mentioned above make use of multiple and sophisticated 3D features directly extracted from skeleton and depth map sequences. Unlike these 3D methods, the discrimination of the features computed around the 3D trajectories is not the focus of this chapter, but could be further investigated (only one 3D descriptor is used, namely HOF, while the 3D skeleton sequences are used only for the clustering of trajectories). Furthermore, our method that is based on scene flow estimation is effective especially in the presence of a high quantity of motion. On the contrary, the methods proposed in [95], [110] called Ordelet and LOP4D, respectively, are effective in the presence of both high or low amount of motion since they use local descriptors. This is confirmed by our experiments on the ORGBD dataset that incorporates actions with a low amount of motion.

These promising results highlight the potential of our first attempt to generalize Dense Trajectories to 3D and opens up new perspectives. Indeed, many components of this 3D concept can be reinforced to increase its effectiveness. For example, 3D trajectories are slightly more noisy than the Dense trajectories mainly because depth sensors introduce additional noise. This noise translated to a significant number of points belonging to the background which appeared to move radially, creating a lot of irrelevant 3D trajectories. Most importantly, the scene flow estimation is not optimal, since it relies on two different modalities which often appear to be misaligned. This fact is reflected in the performance of the 3D Trajectories (without locality), resulting in a notably lower accuracy than the Dense Trajectories, as demonstrated in Table 3.1. Nevertheless, the trajectory clustering around body joints is still able to remove a significant amount of noisy and irrelevant trajectories in 3D Localized Trajectories case.

3.5.5 Global BoW vs. Local BoW

To experimentally motivate the use of local BoWs, we compared the results obtained for 2D Localized trajectories using both a global BoW and a local BoWs. Hence, the experiments were conducted on the cross-environment scenario of the ORGBD dataset. The mean accuracy is notably lower compared to the 2D Localized Trajectories approach with Local BoW, reaching 53.6% vs. 59.8%. The results suggest that trajectories clustering combined with local BoWs contribute significantly to the enhancement of the local discriminative power of the overall approach. They also suggest that the local encoding is more effective since the codebooks are constructed using features that are specific to the motion of each body part.

3.5.6 Computational Complexity

Our approach considers only a local area around each body joint. Therefore, the complexity of the proposed approach is significantly lower than the complexity of the original Dense Trajectories [16] approach. Let us denote the complexity needed to extract features around one motion trajectory by $O(N)$, where N is the number of operations. While the original approach computes features around all the K_1 generated trajectories, our method conserves only K_2 trajectories within a small region around body joints (with $K_1 \gg K_2$). Thus, our approach presents a lower complexity with respect to the original approach ($O(K_2 N) \ll O(K_1 N)$).

3.6 Conclusions

In this chapter, we propose to solve two major shortcomings of the original Dense Trajectories approach using additional modalities provided by RGB-D cameras: the lack of locality information and the ineffectiveness in describing radial motion. Our contribution is two-fold. First, we enhance the discriminative power and locality-awareness of Dense Trajectories by clustering them around human body joints. This method is coupled with the local Bag-of-Words concept, strengthening further the framework. Second, we construct 3D Localized Trajectories for action recognition. For this purpose, we use (a) scene flow instead of optical flow for the generation of the 3D

Trajectories; and (b) 4D extension of the originally used spatiotemporal descriptors. The reported results show the robustness of the two proposed representations in various challenging datasets. As future work, we intend to develop an automatic way of choosing the optimal parameters. In addition, we intend to estimate more reliable and robust to noise 3D trajectories directly from point cloud data for the purposes of enhancing our current approach and extending it to view-invariant action recognition. In the next chapter, we present an alternative way of combining Dense Trajectories with pose information. This concept finds application exclusively in action detection.

Chapter 4

Dense Trajectory-based Action Detection using Human Pose

In the previous chapter, giving a local power to dense trajectories have been demonstrated to be an effective strategy for improving the accuracy of recognition. Similarly, we propose to combine dense trajectories with sparse skeleton trajectories to address effectively the problem of action detection. We introduce an efficient two-stage framework. This framework utilizes pose information for temporal localization and dense trajectories for action recognition. Finally, we conduct experimental validation on a challenging dataset of continuous activities.

4.1 Introduction

Human action detection has drawn significant attention over the past years. This active research topic of computer vision finds applications in various fields, such as video surveillance, healthcare, and human-computer interaction. Still, large background data variations, inaccurate detection of starting and ending points of action, and observation of partial actions [113] are challenges that need to be addressed.

There has been a substantial amount of work in the field of temporal action detection. Huang, Yao, Wang, and De La Torre [114] suggested a model that evaluates and discards action classes

by observing partial events. In addition, in [115], a similar early event detector of short video segments was developed, in which the labels of the expected actions are provided. In [116], authors proposed the segmentation of videos into a sequence of atomic action units. Moreover, a model of simultaneous action localization and detection was proposed in [117], where authors used 3D-HOG descriptors on a sliding window. Furthermore, Schiele [118] used both body pose and motion features for action detection, while web images were used for training a CNN-based activity detector through transfer learning in [119].

While Dense Trajectories (DT) have shown great potential in action recognition [16], [66], [67], their adoption in Action Detection (AD) remains a challenging task. To the best of our knowledge, De Geest, Gavves, Ghodrati, Li, Snoek, Tuytelaars [113], and Shu, Yun, Samaras [120] were the only ones using DT in a similar manner in this field. In particular, they extracted trajectory features from fixed-length video segments, facing two major issues: first, the splitting is performed uniformly and a significant amount of negative data can be mixed with positive data, and second, finding the optimal length of these splits remains an open challenge, which depends on many parameters, such as speed, action class, etc.

In this chapter, we propose an effective way to use dense motion trajectories in action detection. Instead of segmenting the video sequences using a sliding window and extracting trajectory features from them, we develop a two-step supervised algorithm for detection and classification. The first step includes the segmentation of the video sequences into temporal regions of interest. This is performed by classifying each frame as a positive or negative action. When the action proposals are generated, the second step, which is the classification using improved trajectories, is applied to each generated region. Our contribution is twofold. First, we propose an efficient way of detecting temporal regions of interest in videos which can be coupled with any descriptor for action recognition. Second, we avoid training the classifier with background trajectory data, which usually have a low amount of motion and can potentially lead to degradation of performance.

4.2 Background

In this section, we briefly review concepts that are used throughout the chapter and formulate the problem.

4.2.1 Improved Dense Trajectories

In order to represent actions in videos, Wang and Schmid [66] proposed to extract dense motion trajectories for aligning descriptors. This approach is similar to the original Dense Trajectories [16] described in Chapter 2.2.2. However, compared to the original dense trajectories approach [16], authors in [66] propose the removal of camera motion by estimating the homography between consecutive frames using the Random Sample Consensus (RANSAC) algorithm. In this case, Speeded Up Robust Features (SURF) [121] are computed and matched based on the nearest neighbor rule.

In [66], four different descriptors are used for representing videos: the Trajectory Shape Descriptor (TSD) [66], Histograms of Oriented Gradients (HOG), [122], Histogram of Optical Flow (HOF) [122], and Motion Boundary Histogram (MBH) [66]. In order to aggregate the information of the different descriptors and train a classifier for action recognition, Fisher Vectors [123] model is used. Fisher Vectors (FV) encode both first and second-order statistics between the video descriptors and a Gaussian Mixture Model (GMM). The individual FV are concatenated and used as input to a Support Vector Machine (SVM) classifier. Since there are multiple action classes to be recognized, a *one-vs.-all* approach is used [66].

4.2.2 Improved Dense Trajectories of Partial Actions

The improved dense trajectories with Fisher Vectors (iDT+FV) approach have shown great potential in action recognition thanks to two major advantages. Initially, the dense optical flow field offers a low-level motion analysis for videos without additional cost. Secondly, the tracking of fast and irregular motion patterns is robust since the optical field is being smoothed.

However, iDT+FV works inadequately when only a partial view of an action is available (refer to Table 4.1). In this case, the motion pattern is not descriptive and can lead to incorrect

classification. The previous observation makes the use of iDT+FV in action detection particularly challenging. According to a study we conducted on iDT+FV on partial actions, the utilization of a fixed window approach seems to be insufficient. In particular, we applied two different strategies for detecting and recognizing actions on the MSR DailyActivity 3D dataset [65]. In the Video Segmentation approach, videos are divided into segments of equal length and the iDT+FV features are extracted from each segment separately. During classification, we utilize one SVM per video segment. In the Features Grouping approach, we first extract the iDT+FV features from the video sequences and then group them together, following the above classification method. The mean accuracy results of both cases are shown in Table 4.1. As a reference point, we used the mean accuracy measure from the standard iDT+FV approach. The obtained results suggest that partial action recognition using the iDT+FV approach is a particularly challenging task. This leads us to the conclusion that this approach is not suitable for action detection.

Table 4.1: Mean accuracy results on action recognition using Video Segmentation and Features Grouping approaches.

Segments/video	**Video Segm.**	**Features Group.**
1 (baseline)	63.75%	
2	63.12%	60.62%
3	58.75%	65.00%
4	61.25%	60.62%

4.3 Proposed Model

Our goal is to create a trajectory-based action detection model that addresses the challenges discussed in Section 4.2.2. This is accomplished using a two-step supervised model: During the first step, a frame-based binary classifier extracts the action proposals from the video sequences, and, during the second step, these proposals are assigned an action label using a second classifier trained on trajectories features. The idea is to perform a non-uniform video segmentation that can detect full-length video proposals instead of partial action views, offering

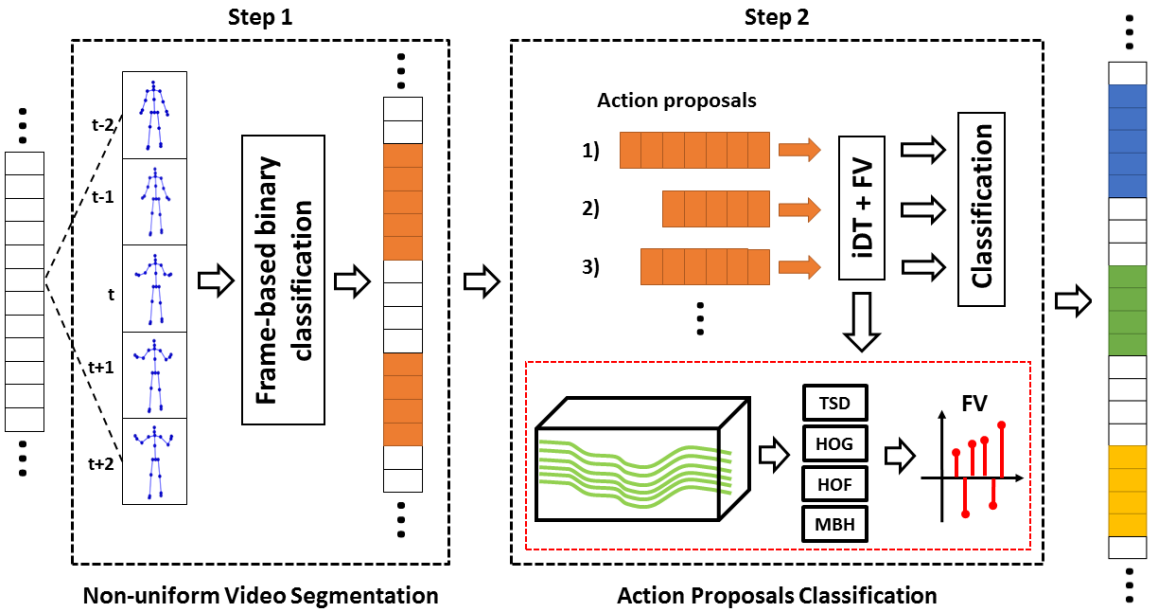


Figure 4.1: Our proposed model for action detection. During Step 1, we extract skeleton joint features (or likelihood areas of joints in 2D case) from a temporal window around the current frame and use them as input to a classifier in order to generate the action proposals from the input sequence. During Step 2, standard action recognition using improved trajectories is performed on the action proposals, resulting in the final labeled sequence.

a more appropriate solution for motion pattern descriptors. We propose two approaches in this regard: in the first one, 3D skeleton joints are available and skeleton-based features are extracted from them, while in the second method, we assume that only RGB video sequences are given. Therefore, a heatmap of likelihood scores of skeleton joint locations is used. For both cases, a second classifier is employed in order to recognize action in the generated proposals, trained on full-length action clips. The general pipeline of our approach is given in Fig. 4.1.

4.3.1 Video Segmentation using 3D skeleton-based features

The first method makes use of the explicit 3D skeleton joints q_t^j, where j is the index of a particular joint such that $j \in \{1, ..., J\}$ and t is the current video frame. In order to spatially describe 3D poses, we use relative joint position features, as proposed in [18]. Those features,

denoted as $\mathcal{D}_t^{ij}$ are generated by computing the distance between each pair of 3D joints:

$$\mathcal{D}_t^{ij} = \mathtt{q}_t^i - \mathtt{q}_t^j, \quad \forall\, i, j \in \{1, ..., J\}. \tag{4.1}$$

Then, a second descriptor, similar to the Histogram of Oriented Displacements (HOD) [124] is used in this context to describe the motion over time. In particular, HOD describes the orientation of each 3D skeleton joint as three 2D trajectories, one for each orthogonal Cartesian plane (xy, xz, yz). For each Cartesian plane, a direction angle θ_j is computed along a temporal window w, as shown below:

$$\theta_j = tan^{-1}\left(\frac{\mathtt{d}(j_{xz})}{\mathtt{d}(j_{xy})}\right), \tag{4.2}$$

where $\mathtt{d}(j_{xz})$ and $\mathtt{d}(j_{xy})$ are the spatial distances of joint j between consecutive frames in Cartesian planes xz and xy, respectively. The orientation features are computed between consecutive frames for a temporal window w around the current frame. The histogram representation is the accumulation of the motion orientation in the quantitized 2D space.

Finally, a binary k-Nearest Neighbor (kNN) classifier is employed for labeling each frame. This classifier seems to be the most suitable solution for our concept, because of its balance between simplicity and high accuracy. Indeed, action detection is very relevant for real-time applications making simplicity an important requirement.

4.3.2 Video segmentation using 2D features

In this approach, we assume that the 2D pose information is not provided and only RGB video sequences are available. Therefore, a state-of-the-art human pose detector [125] is used for the estimation of the likelihood areas of the 2D body joints. This Convolutional Neural Network (CNN)-based pose detector provides a likelihood heatmap $\mathbf{H}_j$ of the joint position j at frame t. These heatmaps are concatenated and used as pose features. They also seem to be more tolerant to erroneous estimation of body pose than raw 2D joints [32].

In addition, the computation of motion features in this context relies on the HOD descriptor, as shown in (4.2). In order to employ it, we need to estimate the exact position of body joints.

Therefore, for each joint and at each instant time, we extract the maximum likelihood denoted as $\hat{\mathrm{q}}^j$ from each corresponding heatmap, as shown below:

$$\hat{\mathrm{q}}^j = argmax(\mathbf{H}_j). \tag{4.3}$$

However, in this case, the 3D information is not provided and only the xy Cartesian plane is used for describing the motion evolution. Similarly to Section 4.3.2, a binary kNN classifier was utilized for frame-based labeling.

4.3.3 Action proposals classification

As a pre-classification step, we ensure the continuity of action proposals. Since some frames show a large variation of classifier scores, we apply a median filter on the classifier score and re-compute the label of each frame. In addition, window-based patching is applied for filling any temporal gaps within the detected action proposals.

The second step, as shown in Fig. 4.1, is common for both approaches. During this step, we use the iDT+FV approach on the generated action proposals. The trajectory-aligned descriptors used in this step are similar to [66], namely, TSD, HOG, HOF, and MBH. Finally, a *one-vs-all* linear SVM classifier is trained on the trimmed groundtruth action clips and tested on the generated action proposals.

4.4 Experiments

Improved trajectories and motion descriptors are computed using the implementation provided in [66][1]. In addition, in Section 4.3.2 the pre-trained CNN-based human pose detector [125][2] is used for obtaining the 2D body pose heatmap.

For generating action proposals, the two proposed approaches are evaluated. The first one, proposed in Section 4.3.1 and based on 3D skeleton joint-based descriptors is called **Skeleton-based Segmentation** and uses the 3D skeleton joint descriptors for action proposals

[1] `https://lear.inrialpes.fr/people/wang/improved_trajectories`
[2] `https://fling.seas.upenn.edu/~xiaowz/dynamic/wordpress/monocap/`

generation. On the other hand, the second approach utilizing the likelihood scores $\mathbf{H}_j$ provided by the output of the CNN-based pose estimator and introduced in Section 4.3.2 is referred to as **Heatmap-based Segmentation**.

Our approaches are tested on the Online Action Detection dataset [126]. It consists of 10 daily action classes *(drinking, eating, writing, opening cupboard, opening oven, washing hands, sweeping, gargling, throwing trash*, and *wiping*) captured continuously and mixed with a large amount of background motion. The sequences were captured using a Kinect v2 sensor, thus RGB, depth, and 3D skeleton joint data are available. We follow the dataset splitting protocol for training and testing our approach.

In both approaches, we used a window length of 11 frames for pose descriptors and a window length of 21 frames for motion descriptors. These parameters were chosen empirically. Moreover, we used 8 bins for computing the HOD features. The 3D body pose (used in the proposed **Skeleton-based Segmentation** approach) of the Online Action Detection dataset consists of 25 joints, whereas in the **Heatmap-based Segmentation** case, the 2D body pose is described by 16 likelihood areas of joints.

For measuring the performance of our approaches, we used the F1-score measure F_1, which is defined as:

$$F_1 = 2\frac{precision \cdot recall}{precision + recall} \tag{4.4}$$

The iDT+SW [120] is considered as a baseline. The obtained results for this model on the Online Action Detection dataset are detailed in Table 4.2. The average F1-score is 0.467 which is lower than the proposed **Heatmap-based Segmentation** approach by 0.076. The superior performance of our model is justified by the fact that it addresses the two major issues of iDT-based approaches: full action proposals are provided as input to the classifier that are more discriminative compared to partial action segments and most of the background (negative) frames are removed from the video segments. Using the informative 3D joint descriptors in the **Skeleton-based Segmentation** case, we achieve significant performance improvement. In particular, we reach an average F1-score of 0.671, which is higher than the JCR-RNN's reported

performance in [126] (0.653 - refer to Table 4.2). Despite the fact that the classification was performed on 2D data (RGB frames), the action proposals were significantly more accurate (88.28% frame-based detection accuracy) and compensated the absence of 3D data in Step 2.

Table 4.2: F1-score results for Heatmap-based and Skeleton-based approaches against JCR-RNN and iDT+SW on Online Action Detection Dataset.

	JCR-RNN [126]	**iDT+SW** [120]	**Heatmap-based**	**Skeleton-based**
Drinking	0.574	0.350	0.218	0.568
Eating	0.523	0.353	0.404	0.484
Writing	0.822	0.582	0.619	0.792
Opening cupboard	0.495	0.453	0.499	0.669
Opening oven	0.718	0.294	0.581	0.677
Washing hands	0.703	0.591	0.759	0.714
Sweeping	0.643	0.467	0.430	0.800
Gargling	0.623	0.505	0.550	0.619
Throwing trash	0.459	0.425	0.573	0.548
Wiping	0.780	0.647	0.802	0.842
Average	**0.653**	**0.467**	**0.543**	**0.671**

4.5 Conclusion

In this chapter, we proposed a novel procedure to use improved trajectories for action detection, by pre-defining the temporal regions of interest. The improved performance comes mainly from Step 1, where the generation of action proposals along with the removal of background frames take place. The positive impact of Step 1 is noticeable in step 2 when using iDT+FV features. The recognition of action proposals becomes more precise since negative data are widely removed and actions are fully visible. The obtained results (Table 4.2) show our model's superiority over some noteworthy approaches in action detection. As future work, we intend to extend the current approach to 3D motion trajectories and empower the second step of our

model by adding viewpoint invariance to it. In what follows, we show our contributions by utilizing sparse trajectories for action recognition.

Chapter 5

A View-invariant Framework for Fast Skeleton-based Action Recognition Using a Single RGB Camera

In this chapter, we propose to solve the issue of viewpoint variation in RGB-based action recognition. Indeed, using a monocular RGB camera, it is difficult to extract view-invariant features due to the 2D nature of these sensors. Instead of relying on complex knowledge transfer algorithms, we propose to take advantage of the recently introduced 3D pose estimation from a single RGB camera. The proposed pipeline can be seen as the association of two key steps. The first step is the estimation of a 3D skeleton from a single RGB image using a CNN-based pose estimator. The second one aims at computing view-invariant skeleton-based features based on the estimated 3D skeletons. A comparison of two different view-invariant skeleton-based descriptors integrated into the proposed framework is also conducted along with extensive experimental evaluation on two well-known datasets.

5.1 Introduction

A huge number of action recognition methods have been proposed and have proven their ability to efficiently recognize human actions as reflected in these two surveys [127], [128]. Usually, it is important to note that classical approaches assume ideal conditions. For example, in [16], [66], [129], the subject performing the action is considered to be facing the camera. However, in a real-world scenario, camera positioning, as well as human body orientation, can vary, and consequently affect the recognition task if the used method does not take into account the viewpoint variability. In fact, viewpoint invariance represents one of the most important challenges in human action recognition. Solving view-invariance requires relating a given acquisition of the subject to its 3D representation. While it is a simple task with RGB-D cameras, it is less obvious using RGB cameras, which only provide 2D information and no explicit 3D.

The development of low-cost RGB-D cameras has made possible the real-time extraction of 3D information via depth maps and skeletons. This has significantly boosted the research on viewpoint invariant action recognition [19], [130], [131]. However, the disadvantages of RGB-D based approaches are tied to RGB-D sensors. First, the estimation of an acceptable depth map and skeleton is limited within a specific range. Second, RGB-D cameras show a high sensitivity to external lighting conditions, making outdoor applications potentially challenging. Both of these reasons restrict their applicability in real-world scenarios such as in video surveillance.
There is, therefore, a need to solve the view-invariance problem using RGB cameras. Among the most successful state-of-the-art approaches are methods based on knowledge transfer [132], [133]. To ensure view-invariance, these methods find a view-independent latent space where the features are mapped and then compared. To achieve that, they use 3D synthetic data computed by fitting cylinders to real data captured with a Motion Capture (MoCap) system, and by projecting them to various viewpoints.

The aforementioned approaches make use of trajectory shape descriptors [16]. These descriptors are, by definition, not view-invariant. Indeed, motion shape in 2D can only be described as points on the image grid; therefore, any radial motion information is mostly lost. In addition, some actions include similar motion patterns from different body parts, which can

negatively impact the classification [32].

In this chapter, instead of relying on a set of 2D projections of synthetic data, we propose to augment 2D data by a third component. Motivated by the very recent encouraging progress on pose estimation from a single RGB image [35], [61], [134], we introduce a novel way of approaching the viewpoint invariant action recognition problem using a single 2D or RGB camera. Our approach consists in estimating human 3D poses from 2D sequences, then directly using this 3D information with a robust 3D skeleton descriptor. Using 3D skeleton-based descriptors makes the approach fully view-invariant since they involve 3D points for describing the body structure. Such descriptors have been proven robust in multiple scenarios [19], [87]. The main advantages of this framework are its simplicity and its low computation time thanks to the use of a high-level representation. In order to validate it, we propose to use *VNect*, presented in Chapter 2, for the estimation of 3D skeletons from 2D videos [35]. The VNect system was selected over related ones [35], [61], [134], because of its real-time performance and its ability to ensure temporal coherence. Two different view-invariant skeleton-based descriptors are used to test this framework, namely, *Kinematic Spline Curves* (KSC) [24], [135] and *Lie Algebra Representation of body-Parts* (LARP) [45]. Finally, the experiments are conducted on two different cross-view action recognition benchmarks: the Northwestern-UCLA [136] and the IXMAS [137] datasets.

5.2 Related Work

As mentioned in Section 5.1, invariance to viewpoint represents a major challenge in action recognition. Viewpoint invariant human action recognition can be categorized into two main classes: RGB-D and RGB based approaches as overviewed below. An extensive review may be found in the recent survey by Trong, Minh, Nguyen, Kazunori, and Hoai [138].

5.2.1 RGB-D based methods

The emergence of RGB-D cameras has importantly facilitated the task of viewpoint invariant action recognition thanks to the availability of 3D information [131], [139]. Indeed, RGB-D cameras provide depth images that may be directly used for defining view-invariant descriptors.

Depth images only provide partial 3D information. In the context of action recognition, human 3D skeletons estimated from depth images are considered to be a more complete high-level 3D representation, which is view-invariant by nature. In addition, with the rapid development of dedicated algorithms to estimate skeletons from depth maps such as [17], numerous view-invariant skeleton-based approaches have been proposed. One of the pioneering works has been introduced by Xia, Chen, and Aggarwal [19], where a descriptor encoding a histogram of 3D joints was proposed. Nevertheless, since the absolute position of joints is used, these features are sensitive to anthropometric variability. To resolve this issue and preserve view-invariance, some approaches proposed to describe actions using the distance between joints. For instance, in [87], actions are depicted using a novel descriptor called *eigenjoints*. The latter is computed by applying Principal Component Analysis (PCA) on the spatial and temporal Euclidean distances between joints.
To cope with viewpoint variability and increase accuracy, other approaches have modeled human actions using more sophisticated geometric tools. In [140], the authors proposed a novel view-invariant representation by introducing a descriptor based on the relative position of joint quadruples. Also, Vemulapalli, Arrate, and Chellappa [45] suggested a new representation called Lie Algebra Representation of body-Parts (LARP) by computing the geometric transformation between each pair of skeleton body-parts.

The presented descriptors are implicitly unaffected by the viewpoint variability as they are defined using invariant features such as the distance between joint, angles, transformation matrices, etc. Nevertheless, since the 3D skeleton contains the full 3D information, an alignment pre-processing can be simply applied before undertaking the descriptor computation. For example, we cite the work of Ghorbel, Boutteau, Boonaert, Savatier, and Lecoeuche [24], where the motion has been modeled by computing and interpolating kinematic features of joints. In this case, the *Kinematic Spline Curves* (KSC) descriptor is not view-invariant by nature; thus, the skeletons are initially transferred to a canonical pose.

Although these representations have shown their effectiveness in terms of computation time and accuracy, they are hardly applicable in various scenarios, since the skeletons are estimated using RGB-D cameras. Indeed, the skeleton estimation accuracy decays in the presence of a

non-frontal view [86] due to self-occlusions. Furthermore, as mentioned in Section 1, RGB-D cameras require specific conditions to optimally work such as outdoor environment, closeness to the camera, moderate illumination, etc. As a result, RGB-D based human action recognition has limited applications.

5.2.2 RGB-based methods

Very recent efforts have been made to propose view-invariant human action recognition methods using a monocular RGB camera. However, the challenge is that RGB images do not explicitly contain 3D information and consequently traditional descriptors, such as the Histogram of Oriented Gradients (HOG) [70] and Motion Boundary Histograms (MBH) [141], are highly affected by the introduction of additional views [142]. Thus, some RGB-based methods have been specifically designed to overcome viewpoint variation [132], [133], [136], [137], [143]–[146].

One way of approaching the problem is to match one viewpoint to another using geometric transformation as in [146], [147]. However, this category of methods which are usually based on 3D exemplars requires the use of labeled multi-view data. Another way consists in designing spatio-temporal features which are insensitive to viewpoint variation [143], [148], [149]. However, their discriminative power has been shown to be limited [133].

The most popular RGB-based approaches are *knowledge transfer*-based methods. The idea of knowledge transfer for view-invariant action recognition is to map features from any view to a canonical one by modeling the statistical properties between them. For instance, Gupta, Martinez, Little, and Woodham [132] introduced a novel knowledge transfer approach using a collection of data containing unlabeled MoCap sequences. Dense motion trajectories from RGB sequences are matched to projections of 3D trajectories generated from synthetic data (cylinders fitted to MoCap data). However, the number of these projections is finite, which means that not every viewing angle is represented. In addition, it is highly possible that different but similar-looking (from a specific angle) 2D motion patterns are incorrectly matched since the 2D descriptors used in this context are view-dependent.

In [133], dense motion trajectories [132] are computed using synthetic data similar to [132], and represented using a codebook. A histogram is then built in order to be used as a final descriptor.

This particular method is robust even when the testing view is completely different from the training views. This is due to the fact that the introduced Non-Linear Transfer Model (NKTM) allows the approximation of non-linear transformations. Despite their efficiency, the two methods proposed in [133] and in [132] rely on 2D-based descriptors that are not invariant to viewpoint changes.

5.3 Proposed Framework for RGB-based View-Invariant Action Recognition

In this section, we present the proposed framework to perform a fast view-invariant human recognition from a single RGB camera. Inspired by the advances in human pose estimation and the performance of skeleton-based approaches, we propose to first generate 3D human skeletons from a monocular RGB camera based on the recently introduced CNN-based approaches. Then, the extracted skeletons are used to compute skeleton-based features. Figure 5.1 illustrates the proposed pipeline. In what follows, we detail the different steps of this pipeline.

5.3.1 Feature extraction

The first stage of our approach is the 3D pose estimation from monocular RGB cameras. Towards this direction, we employ a state-of-the-art pose estimator such as [35]. Using the estimated skeletons, we propose to independently integrate two different view-invariant skeleton-based methods: LARP [45] and KSC [24]. In [45], the used features are view-invariant by nature, while in [24], a skeleton alignment pre-processing is realized. In the following two subsections, we describe both LARP and KSC.

Lie Algebra Representation of body-Parts (LARP)

In [45], an efficient skeleton-based action recognition approach is introduced. The approach is based on describing the geometric relationship between different coupled body segments. Let $\mathsf{S}(t) = (\hat{\mathsf{Q}}_t, \mathsf{E}(t))$ be a set of skeleton sequences $\hat{\mathsf{Q}}_t$ with J joints, and B rigid-oriented body parts

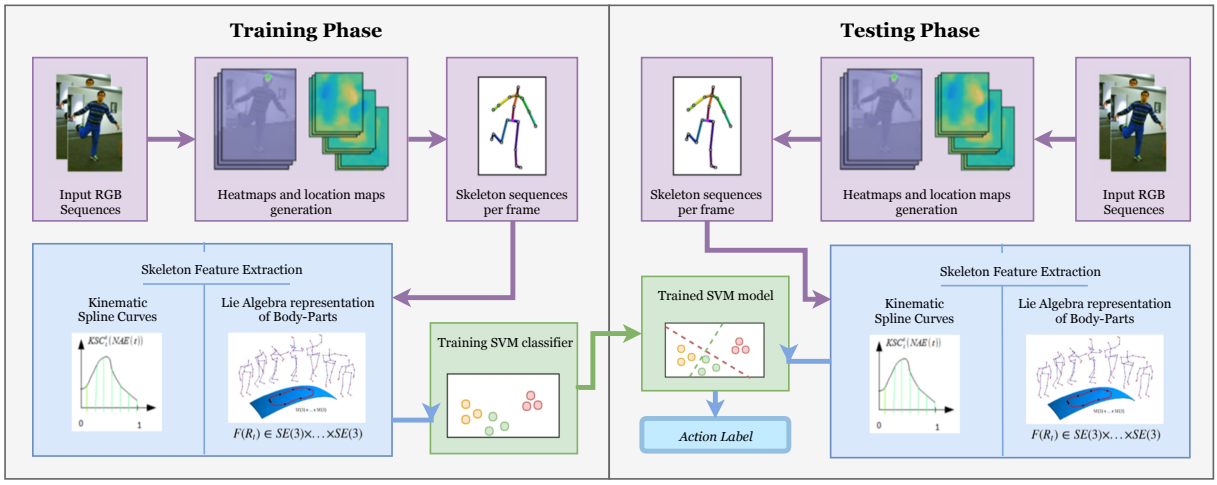


Figure 5.1: Overview of the proposed pipeline for fast and view-invariant human action recognition from a monocular RGB image: in both the training phase and the testing phase, skeletons are extracted from RGB images using the heatmaps and locations maps generated by the VNect algorithm [35]. Then, based on the estimated skeleton, skeleton features are computed e.g., LARP and KSC. Finally, in order to train a model of classification and use it to recognize actions, linear SVM is used.

$\mathsf{E}(t)$. The skeleton sequence are described in (2.10), while the rigid-body parts are defined as $\mathsf{E}(t) = \{\mathsf{e}_1(t), \mathsf{e}_2(t), \cdots, \mathsf{e}_B(t)\}$. Each body part $\mathsf{e}_1(t)$ is assigned a 3D local coordinate system. Then, between each couple of local coordinate systems attached to the body-parts $\mathsf{e}_i(t)$ and $\mathsf{e}_j(t)$, a 3D rigid transformation matrix $\mathbf{T}_{i,j}(t)$ is defined as:

$$\mathbf{T}_{i,j}(t) = \begin{bmatrix} \mathbf{R}_{i,j}(t) & \mathbf{t}_{i,j}(t) \\ 0 & 1 \end{bmatrix}, \tag{5.1}$$

where $\mathbf{R}_{B,J}$ is a 3×3 rotation matrix and $\mathbf{t}_{i,j}(t)$ a three-dimensional translation vector.

To completely encode the geometric relation between e_B and e_J, both $\mathbf{T}_{B,J}$ and $\mathbf{T}_{J,B}$ are estimated. Subsequently, a sequence of skeletons varying over time is represented as $\mathbf{\Theta}(t) = [\mathbf{T}_{1,2}(t), \mathbf{T}_{2,1}(t), ..., \mathbf{T}_{J,B}(t), \mathbf{T}_{B,J}(t)]$. The set of rigid transformation matrices defines a direct product of non-Euclidean observation space called the Special Euclidean group SE(3). As a result, each representation of a skeleton is a point and the skeleton sequence is a curve in $\mathrm{SE(3)}^{2C_B^2}$, with C_B^2 denoting the combination operation. Classification of the observed curves is done on the tangent space of the identity matrix, using the Support Vector Machine (SVM)

algorithm. Note that, a preliminary point matching is necessary to achieve temporal alignment which, in [45], is achieved via dynamic time warping and Fourier temporal pyramid representation. The use of 3D rigid transformation matrices between body-parts as features ensures the view-invariance since they are independent of the view of acquisition.

Kinematic Spline Curves (KSC)

This second skeleton-based representation has been introduced in [24] and is mainly characterized by its compromise between computational latency and accuracy. To do that, the chosen components are carefully selected to ensure accuracy and computational efficiency. The descriptor is based on the computation of kinematic values, more specifically joint position $\hat{\mathrm{Q}}_t$, joint velocity $\mathcal{V}(t)$ and joint acceleration $\mathcal{A}(t)$.

The key idea of this approach is to define a kinematic curve of a skeleton sequence as

$$\mathbf{KF}(t) = [\hat{\mathrm{Q}}_t, \mathcal{V}(t), \mathcal{A}(t)]. \tag{5.2}$$

Subsequently, a kinematic curve can be reparameterized such that it is invariant to execution rate using a novel method called Time Variable Replacement (TVR) [24]. As its name indicates, this method consists in changing the variable time by another variable that is less influenced by the variability in execution rate. It can be written as

$$\mathbf{KF}(\phi(t)) = [\hat{\mathrm{Q}}(\Phi(t)), \mathcal{V}(\Phi(t)), \mathcal{A}(\Phi(t))]. \tag{5.3}$$

The new parameter ϕ is constrained to be bijective, increasing with respect to t, and have a physical rate-invariant meaning. In our case, we use the Pose Motion Signal Energy function proposed in [24] to define ϕ. Subsequently, in order to obtain a meaningful descriptor, the discrete data point samples $\mathbf{KF}(\phi(t))$ are interpolated using a cubic spline interpolation, then, uniformly sampled. Finally, the classification is carried out using a linear SVM. It is important to note that the computation of this descriptor includes also skeleton normalization and skeleton alignment steps making it respectively invariant to anthropometric and viewpoint changes. The

Figure 5.2: Frame samples from the Northwestern-UCLA dataset: an example is given for each viewpoint V_1, V_2 and V_3

alignment is carried out by estimating a transformation matrix between each skeleton and a canonical pose.

5.4 Experiments

The proposed pipeline is tested on two different cross-view human action recognition benchmarks: the Northwestern-UCLA Multiview Action3D [136] denoted by N-UCLA and the INRIA Xmas Motion Acquisition Sequences dataset [133] denoted by IXMAS.

5.4.1 Datasets

Northwestern-UCLA dataset

The Northwestern-UCLA dataset consists of videos captured by using 3 different Kinect sensors from different viewpoints. Thus, this dataset contains in total 3 modalities: RGB images, depth maps, and skeleton sequences and includes 10 action classes: *pick with one hand, pick up with two hands, drop trash, walk around, sit down, stand up, donning, doffing, throw* and *carry*. Each action class is repeated by 10 subjects from 1 to 6 times. The main challenge of this dataset is that it contains very similar actions such as *pick with one hand* and *pick up with two hands*. Figure 5.2 illustrates examples from this benchmark.

IXMAS dataset

This dataset is captured using 5 synchronized RGB-cameras placed in 5 different viewpoints: four from the side and one from the top of the subject. IXMAS dataset is constituted from 11 different action categories: *check watch, cross arms, scratch head, sit down, get up, turn around, walk, wave, punch, kick* and *pick up*. This dataset is challenging since it contains complex viewpoints leading to self-occlusions. Such viewpoints are illustrated in Figure 5.4 (top row).

5.4.2 Experimental settings and implementation details

All the experiments were run on an i7 Dell Latitude laptop with 16GB RAM and implemented in *Matlab*. For both datasets, we follow the same experimental protocol used in [133]. For the case of the Northwestern dataset, two viewpoints are used for the training and the third for the testing. In total, 3 experiments are performed. Moreover, each test on the IXMAS dataset involves every combination of viewpoint pairs for training and testing, resulting in 20 experiments in total.

In this work, we consider two types of experiments: *VNect+KSC* and *VNect+LARP*. VNect+KSC refers to our framework combined with the KSC descriptor, while VNect+LARP denotes our framework merged with the LARP descriptor. We compare our framework with the recent RGB-based methods denoted in the rest of the chapter by Hanklets [143], Discriminative Virtual Views (DVV) [145], AND-OR Graph (AOG) [136], Continuous Virtual Pat (CVP) [144], Non-linear Circulant Temporal Encoding (nCTE) [132] and Non-linear Knowledge Transfer Model (NKTM) [133].

{Source} | {Target}	0| 1	0| 2	0| 3	0| 4	1| 0	1| 2	1| 3	1| 4	2| 0	2| 1	2| 3	2| 4	3| 0	3| 1	3| 2	3| 4	4| 0	4| 1	4| 2	4| 3
Hankelets [143]	83.7	59.2	57.4	33.6	84.3	61.6	62.8	26.9	62.5	65.2	72.0	60.1	57.1	61.5	71.0	31.2	39.6	32.8	68.1	37.4
DVV [145]	72.4	13.3	53.0	28.8	64.9	27.9	53.6	21.8	36.4	40.6	41.8	37.3	58.2	58.5	24.2	22.4	30.6	24.9	27.9	24.6
CVP [144]	78.5	19.5	60.4	33.4	67.9	29.8	55.5	27.0	41.0	44.9	47.0	41.0	64.3	62.2	24.3	26.1	34.9	28.2	29.8	27.6
nCTE [132]	**94.8**	69.1	**83.9**	39.1	90.6	79.7	79.1	30.6	72.1	**86.1**	77.3	62.7	82.4	**79.7**	70.9	37.9	48.8	40.9	70.3	49.4
NKTM [133]	92.7	**84.2**	**83.9**	**44.2**	**95.5**	77.6	**86.1**	**40.9**	82.4	79.4	85.8	71.5	82.4	**80.9**	**82.7**	**44.2**	**57.1**	**48.5**	**78.8**	**51.2**
VNect+LARP (ours)	46.6	42.1	53.9	9.7	50.6	37.5	47.3	10.0	43.4	33.0	53.6	11.8	51.2	37.8	53.6	9.1	10.9	8.7	10.9	7.9
VNect+KSC (ours)	86.7	80.6	82.4	15.5	91.5	79.4	81.8	15.8	**85.2**	77.0	**88.5**	16.4	**83.0**	77.9	82.4	12.1	28.1	24.8	29.1	24.2

Table 5.2: Accuracy of recognition (%) on the IXMAS dataset: the different tests are detailed. Each time, one viewpoint is used for training (Source) and another one for testing (Target).

{Source} | {Target}	{1,2}| 3	{1,3}| 2	{2,3}| 1	Mean
Hankelets [143]	45.2	-	-	-
dvv1 [145]	58.5	55.2	39.3	51.0
CVP [144]	60.6	55.8	39.5	52.0
AOG [136]	73.3	-	-	-
nCTE [132]	68.8	68.3	52.1	63.0
NKTM [133]	75.8	73.3	59.1	69.4
VNect+LARP (ours)	70.0	70.5	52.9	64.47
VNect+KSC (ours)	**86.29**	**79.72**	**66.53**	**77.51**

Table 5.1: Accuracy of recognition (%) on the Northwestern-UCLA dataset: We report the accuracy obtained for each test (when two viewpoints are used for training (Source) and one viewpoint for testing (Target)) and the average accuracy for the three tests (Mean).

5.4.3 Results and discussion

The results on the Northwestern-UCLA dataset are reported in Table 5.1 and prove that our method (VNect+KSC) outperforms state-of-the-art methods. Indeed, an increase of around 8% compared to the most competitive approach can be noted (NKTM [133]). Moreover, Figure 5.3 shows that for almost all action classes, VNect+KSC outperforms nCTE [132] and NKTM [133]. On the other hand, despite the fact that VNect+LARP shows a lower accuracy by 5% compared to NKTM, this approach stands among the best-performing ones, showing promising results.

The results for the IXMAS dataset are presented in Table 5.2 and Figure 5.3. Our proposed

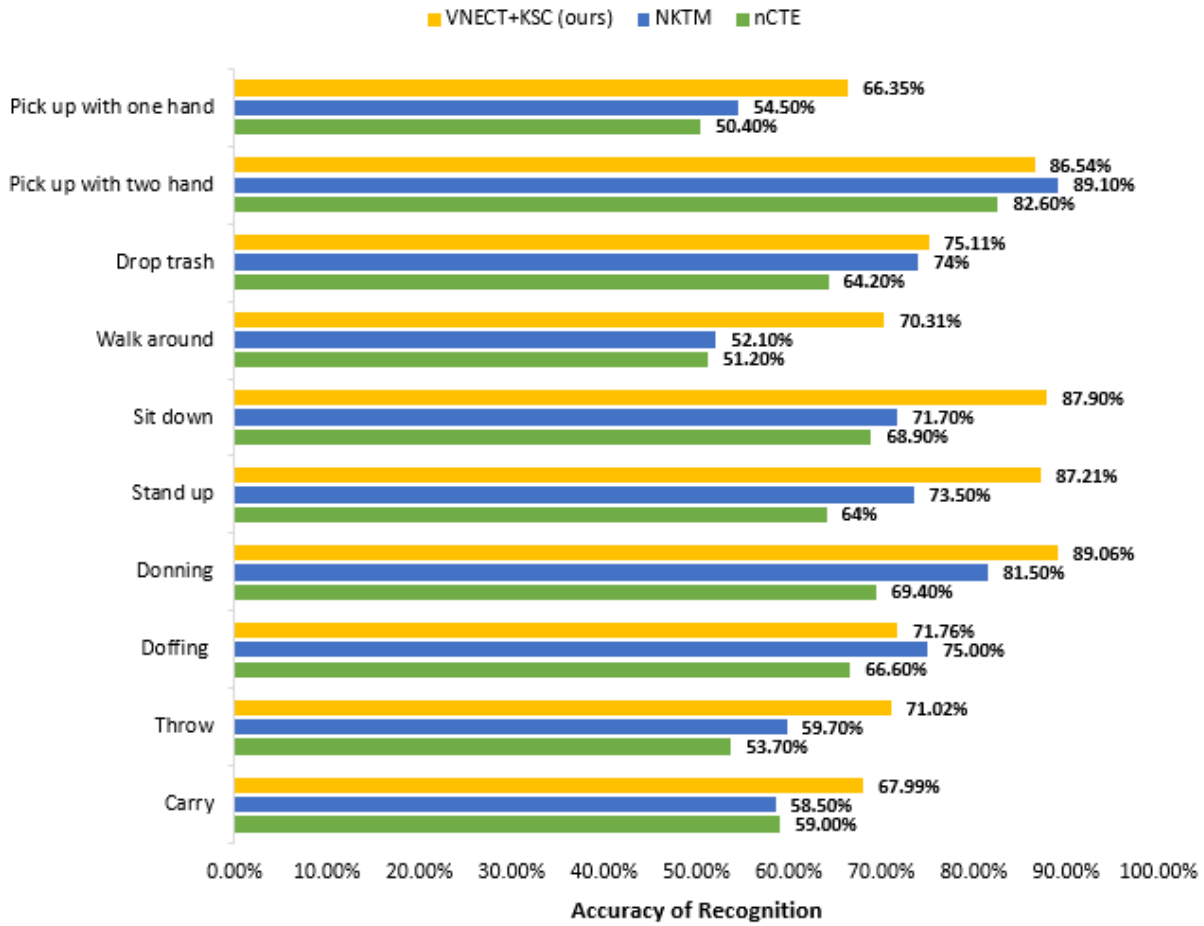


Figure 5.3: Action recognition accuracy for each action on the Northwestern-UCLA dataset: comparison of our method with NKTM[133] and nCTE[132]

approach (VNect+KSC) achieves the third-best mean recognition accuracy, achieving 58.12% (against 72.5% for NKTM [133] and 67.4% for NCTE [132]). However, as depicted in Table 5.2, for every viewpoint pair, our approach shows a competitive performance, except for the ones which include viewpoint V_4. For example, tests $2 \mid 0$ and $2 \mid 3$ outperform earlier works and respectively reach an accuracy of 85.2% and 88.5%, while tests $0 \mid 4$ and $2 \mid 4$ present very low results (respectively 15.5% and 16.4%). This poor performance is the result of erroneous and noisy skeleton estimation coming from the pose estimator. Figure 5.4 illustrates an example of the extraction of skeletons from different viewpoints using VNect. This figure highlights the fact that all skeletons are visually coherent except for the one extracted from V_4 which represents

the top viewpoint. The presence of self-occlusions in V_4 is crucial for the performance of VNect since it makes the skeleton estimation by nature more challenging. Nevertheless, this constraint can be generalized to other approaches, affecting their performance, as well. By investigating more on this question, we discovered that VNect is not trained on extreme viewpoints such as V_4. Thus, we underline a very interesting research issue to study in the future.

For this reason, we propose to evaluate the proposed concept by keeping in mind that the current version of VNect is not adapted yet to the estimation of skeletons from top views. Thus, we compute the average accuracy by ignoring the tests where V_4 has been considered. The results reported in Table 5.3 show that our approach competes with state-of-the-art by achieving 83.03% of recognition. It shows the second-highest accuracy after NKTM [133] approach (reaching 84.46%) with only 1% of difference.

$\{$Source$\}$ | $\{$Target$\}$	Mean with V_4	Mean without V_4
Hankelets [143]	56.4	61.41
DVV [145]	38.2	36.2
CVP [144]	42.2	49.60
NCTE [132]	67.4	80.45
NKTM [133]	**72.5**	**84.46**
LARP-VNect (ours)	31.50	45.91
KSC-VNect (ours)	58.12	83.03

Table 5.3: Average accuracy of recognition (%) on the IXMAS dataset: the first value (Mean with V_4) reports the average of all the tests done, while the second value (Mean without V_4) computes the average of all texts excepting the ones involving V_4.

RGB-based skeletons vs. RGB-D-based skeletons

In order to compare the quality of skeletons extracted from VNect compared to the ones provided by RGB-D cameras for the task of action recognition, we propose to compute the KSC descriptor using both the VNect-generated skeletons and the RGB-D skeletons.

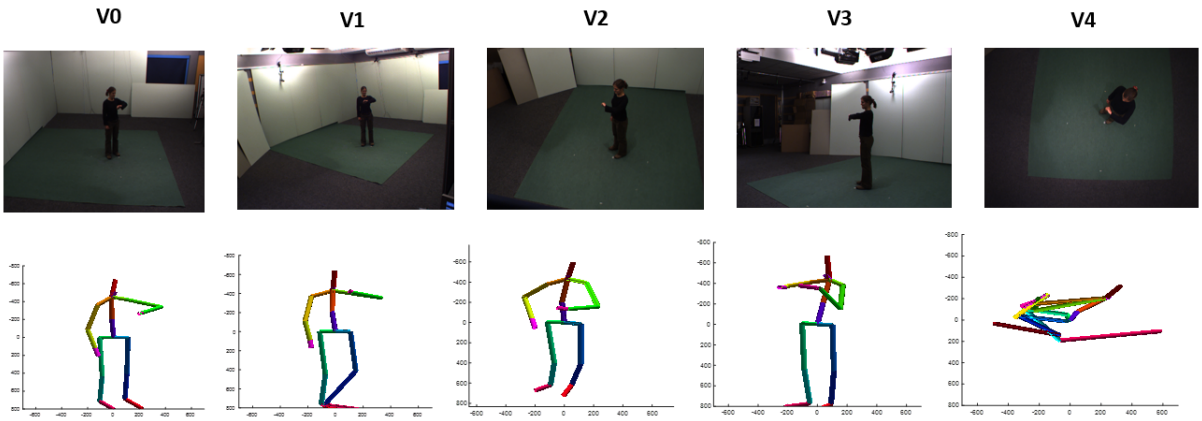


Figure 5.4: Illustration of skeleton extraction from the IXMAS dataset using VNect system: it can be noted that for the four first views (V_0, V_1, V_2, V_3), the quality of the estimated is visually acceptable. However, the quality of the last view V_4 is completely biased. This fact is confirmed by our experiments.

Results obtained on the Northwestern-UCLA dataset are reported in Table 5.4. Skeleton-RGB-D and skeleton-VNect refer to the results obtained by applying respectively the KSC descriptor to the skeletons provided by the Kinect and the skeletons provided by the VNect. The reported results show that action recognition can be more robust using VNect-generated skeleton sequences. In fact, using VNect skeletons, the mean accuracy increased by 7.4% compared to the utilization of the provided RGB-D skeleton sequences. The reason for that is the fact that the extraction of skeletons from RGB-D cameras is less accurate when the human body is not totally visible. With the variation of human body orientation with respect to the camera, self-occlusions occur, impacting negatively the skeleton estimation.

LARP vs. KSC

The results performed on the Northwestern dataset as well as on the IXMAS dataset show the superiority of KSC descriptor for viewpoint action recognition when combined with VNect skeletons. Indeed, KSC+VNect presents an average accuracy of 77.51% against 64.47% for VNect+LARP on the Northwestern UCLA dataset. On the IXMAS dataset, KSC outperforms LARP, as well, by achieving an average accuracy of 83.03% against 58.12% when ignoring V_4

and of 45.91% against 31.5% when considering it. The interpretation of this result lies on the fact that KSC+VNect is less sensitive to noise than LARP.

{Source} | {Target}	{1,2}| 3	{1,3}| 2	{2,3}| 1	Mean
skeleton-RGB-D	80.5	72.6	61.0	71.1
skeleton-VNect	**86.3**	**79.7**	**66.5**	**77.5**

Table 5.4: Accuracy of recognition (%) on the Northwestern dataset using the KSC descriptor: the performances obtained when using the skeletons provided by RGB-cameras and the ones extracted using VNect algorithm are compared. We report the accuracy obtained for each test (when two viewpoints are used for training and one viewpoint for testing) and the average accuracy (Mean).

Computation time and memory

The main advantage of our framework is its low computation time. The training plus testing process takes only 6 minutes, as presented in Table 5.5. This shows that our framework can be considered as a real-time system during testing.

On the other hand, the proposed framework, when using VNect for the skeleton estimation step, requires to consume only 58.5MB of further memory which is comparable to the memory needed to store the learned R-NKTM and the general codebook (57MB) in [133] and which is significantly lower than the memory needed to store the samples (30 GB) in [132].

Method	Training + Testing
AOG* [136]	1020
NCTE*[132]	612
NKTM*[133]	38
VNect+KSC	6

Table 5.5: Computation time in minutes on the Northwestren dataset by using V_1 and V_2 for training and V_3 for testing. All the reported computation time includes descriptor calculation. *We specify that the reported values for AOG [136], NCTE [132], NKTM [133] have been reported from the paper [133] and therefore the computation time has not been computed on the same computer.

5.5 Conclusion and Future Work

In this work, a simple but original framework has been proposed to resolve the issue of cross-view action recognition based on a single monocular RGB camera. For this purpose, a novel concept aiming at augmenting 2D images by a third dimension is proposed taking advantage of the recent advances in 3D pose estimation from a monocular RGB camera and the effectiveness of skeleton-based descriptors. A 3D skeleton is first estimated from a single 2D image using a CNN-based approach. Then, a view-invariant skeleton-based method is applied to the estimated skeletons. To prove the validity of our framework, the recently introduced VNect system has been chosen to extract 3D skeletons from RGB images. After that, two different view-invariant skeleton-based approaches have been tested: KSC [24] and LARP[45]. The experiments on two datasets have shown the superiority of KSC when integrated into that framework. The obtained results are competitive with respect to recent state-of-the-art approaches on both datasets, except for the cases where an extreme viewpoint (the top viewpoint) is considered. This suggests that it would be important to extend the 3D pose estimator to extreme viewpoints. This idea will be explored in future work. In the next chapter, we introduce our main contribution in cross-view action recognition from a monocular RGB camera.

Chapter 6

A Novel Framework for Learning Deep View-Invariant Human Action Representations using a Single RGB Camera

While in the previous chapter, the use of 3D estimated skeletons for the task of action recognition has been validated, in this chapter, we explore a more effective way to exploit this kind of data. In particular, we introduce a DNN-based framework for learning view-invariant features from a monocular camera called DeepVI. This framework incorporates a filtering module called SmoothNet based on a revisited version of Temporal Convolutional Networks (TCN) for implicitly smoothing skeleton sequences. Furthermore, we conduct an experimental validation and extensive analysis of our approach on two challenging datasets.

6.1 Introduction

Viewpoint variation results from the change of human body orientation with respect to the camera. Since RGB cameras project the 3D scene to a 2D plane, motion and appearance features vary

significantly from one viewpoint to another, as shown in Fig. 6.1. As a consequence, this impacts the recognition of actions negatively.

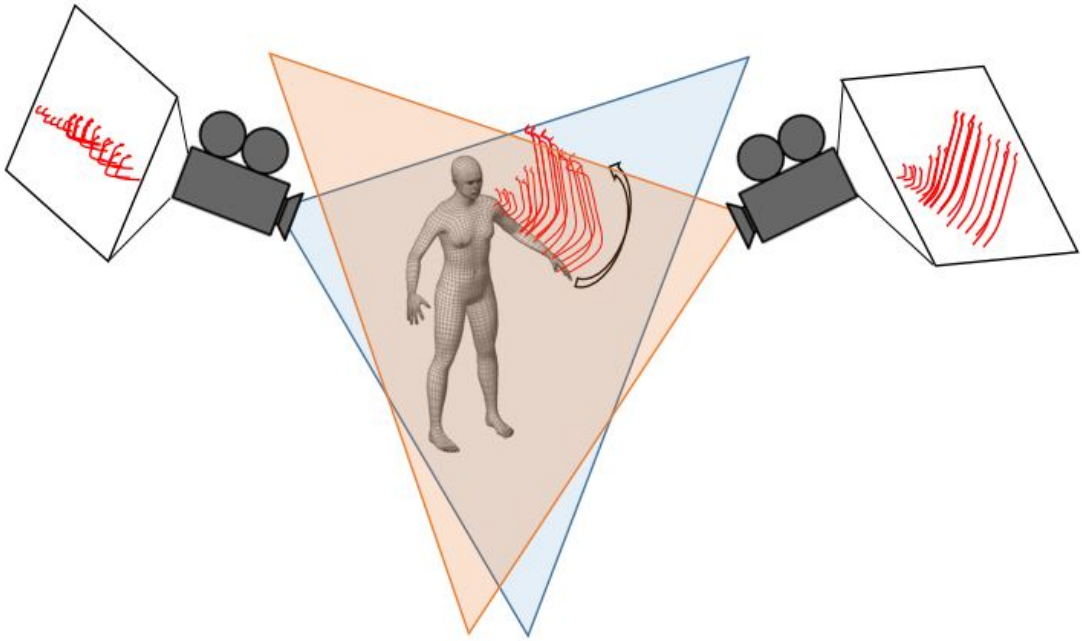


Figure 6.1: Illustration of the issue of viewpoint variation in the context of action recognition: the shape of classical 2D motion descriptors varies from one viewpoint to another.

The most intuitive way of addressing viewpoint variation referred to as *multi-view* action recognition is to train a classification model using data acquired from different viewpoints. In the literature, this simple idea has shown great potential [150], [151]. Nevertheless, in concrete scenarios, collecting and annotating data from different viewpoints can be constraining and costly. For this reason, *cross-view* approaches have been proposed in the state-of-the-art [132], [133]. In contrast to the previous class of approaches, cross-view methods aim at designing view-invariant representations without providing data acquired from different viewpoints during the training phase.

Knowledge transfer has been considered as one of the most efficient techniques used in the context of cross-view action recognition [132], [133], [152]. The most successful knowledge

transfer-based approaches generate 3D animated synthetic data, compute relevant features, and project them in various 2D planes. Then, using these 2D synthetic data, they train a network that maps features extracted from 2D videos to a view-independent latent space. However, as mentioned in [36], [153], these approaches usually rely on 2D representations such as dense trajectories that are by definition not view-invariant and do not incorporate radial motion [32], [33].

Thanks to the recent advances in 3D pose estimation from a single RGB image [35], [52], [55], [134], [154], 2D cross-view action recognition has been addressed from a novel perspective. In [36], [153], 3D pose (called also 3D skeleton) sequences are estimated and 3D skeleton-based action recognition methods are applied. View-invariance is achieved by carrying out a pre-processing of pose alignment. Nevertheless, this alignment step can generate errors in the presence of noisy skeletons, in addition to temporal noise due to the lack of temporal consistency. Indeed, the 3D pose is usually estimated per-frame and not by considering the full sequence [35], [52], [55].

Inspired by these previous methods, we propose to make use of 3D pose estimation in order to encode view-invariance. However, to overcome the issues mentioned above, instead of relying on pose alignment, we propose to learn view-invariant features using a Deep Neural Network (DNN). This is done by rotating the estimated 3D skeleton sequences in order to augment the amount of data directly. This framework termed *DeepVI*, allows us to ensure a compromise between viewpoint invariance and discrimination. Our DNN is composed of two main modules: (1) a smoothing network named *SmoothNet* that implicitly filters the skeleton joint trajectories temporally using a revisited version of *Temporal Convolutional Networks (TCN) [155]* and (2) the state-of-the-art *Spatial-Temporal Graph Convolutional Networks (ST-GCN)* [25] that takes into consideration the spatial and temporal skeleton sequence connectivity. Both modules are trained in an end-to-end manner. Experiments on two well-known datasets show the efficiency of the proposed framework.

6.2 Related Work

Over the last years, cross-view action recognition has attracted significant attention from the computer vision community due to its interest in several applications.

Earlier works attempted to build view-invariant features from 2D data [143], [156]. For instance, Junejo, Dexter, Laptev, and Pérez. [156] proposed a view-invariant feature measuring the temporal self-similarities of action sequences over time. Li, Camps, and Sznaier. [143] represented the motion trajectories using Hanklets encoding the dynamic properties of tracklets.

Instead of designing hand-crafted view-invariant features, other approaches proposed to learn features shared by different viewpoints [144], [145], [150], [151], [157]–[160]. Farhadi and Tabrizi [157] proposed to learn a knowledge transfer model mapping features extracted from a given source view to a target view. Similarly, Liu, Shah, Kuipers, and Savarese [160] adopted a view transfer knowledge model using a bipartite graph. To relax the constraints on the availability of training data, Li and Zickler [145] proposed to generate virtual views by assuming the continuity of the descriptors from one viewpoint to another. This allows the weakly supervised learning of the transfer model, in case of limited training data. Zhang, Wang, Xiao, Zhou, Liu, and Shi [144] adapted the previous work to unsupervised learning by imposing a logical constraint (each data belongs to only one class).

Hao, Wu, Wang, and Sun [150] employed sparse coding to transfer low-level features extracted from different views to a discriminative and high-level semantic space. Kong, Ding, Li, and Fu [151] trained a deep neural network in order to learn shared features. Moreover, Wang, Ouyang, Li, and Xu [161] introduced a two-level learning model by learning shared and specific features using Convolutional Neural Networks (CNNs) and Conditional Random Field (CRF). Ulhaq, Yin, He and Zhang [158], [159] also employed deep learning to learn a view-invariant latent space. These approaches have been proven to be accurate. However, as described in [133], it is essential to note that they learn a latent space based on the a priori knowledge of views included in a specific dataset. Thus, they are hardly applicable in real-world scenarios, mainly if the provided data are acquired from a single viewpoint.

For this reason, a second kind of knowledge-transfer approaches has been proposed that

ignores this a priori knowledge. These methods create synthetic samples from 3D available datasets to overcome the lack of dimensionality [132], [133], [152]. Gupta, Martinez, Little, and Woodham [132] proposed to train a classifier using 2D features generated from virtual viewpoints. To that aim, 3D synthetic data are computed by fitting cylinders to Motion Capture (MoCap) data. Rahmani, Mian, and Shah [133], [152] extended the previous work by incorporating a non-linear knowledge transfer model and by fitting a 3D human model to MoCap data instead of cylinders.

Despite their success, these approaches present some drawbacks: they usually describe human motion using 2D dense trajectories which are, by nature, not view-invariant and require a critical computational time [36]. Furthermore, as demonstrated in [32], the motion of different body parts can result in similar patterns and consequently may include bias in the recognition.

Recently, impressive progress has been made in 3D pose estimation from a single 2D image [35], [52], [55], [134], [154]. Taking advantage of this advance, 3D-skeleton based approaches have been applied to the estimated data [36], [153]. Since the estimated 3D skeletons are fully 3D, view-invariance can be satisfied by applying a simple alignment as a pre-processing step. The results have shown great potential. However, the estimated data remain relatively noisy and may affect the results, since they can impact the alignment and the feature extraction steps. More details about these issues are presented in Section 6.3. Thus, we propose a framework able to deeply learn view-invariant features without taking into consideration the viewpoint variation included in the dataset during the training phase. This framework also includes a module that is able to implicitly smooth the joint trajectories to reduce the impact of noise. This module is trained in an end-to-end manner.

6.3 Problem Formulation: Cross-view Action Recognition

Let us assume that a set of N cameras is installed around an observed scene S capturing the corresponding synchronized action videos $(\mathsf{V}^{(i)})_{1\leq i\leq N}$. Each video $\mathsf{V}^{(i)}$ is the projection of the scene S and is represented as a sequence of RGB frames ($\mathsf{V}^{(i)} \in \mathbb{R}^{2\times c\times T}$, with T their temporal length and $c = 3$ the number of channels). The goal of cross-view action recognition is to find a

map Φ that computes a view-invariant representation such that, given the same scene S;

$$\forall\, i, j \in \{1, 2, ..., N\}, \Phi(\mathsf{V}^{(i)}) = \Phi(\mathsf{V}^{(j)}) \tag{6.1}$$

This problem has been considered as very challenging due to the loss of dimensionality while using an RGB sensor. Indeed, as illustrated in Fig. 6.1, the viewpoint variation leads to the undesired variation of 2D human motion descriptors. Similar to [89], [153], we propose to exploit the recent progresses of 3D pose estimation from 2D images [35], [52], [55], [134], [154]. Thus, Φ can be seen as the composition of two functions Φ_1 and Φ_2, such that $\Phi = \Phi_2 \circ \Phi_1$ with $\Phi_1 : \mathbb{R}^{2\times c\times T} \rightarrow \mathbb{R}^{3\times J\times T}$ representing a mapping function able to estimate a 3D pose sequence from a 2D video (with J the number of joints) and $\Phi_2 : \mathbb{R}^{3\times J\times T} \rightarrow \mathbb{L}$ representing a function predicting the label of the action contained in the estimated 3D skeleton sequence (with $\mathbb{L}$ the label space). In other words, Φ_1 denotes the function making the transition from 2D to 3D data and Φ_2 is the recognition function, which encodes the feature extraction and the classification steps.

As mentioned in Section 6.2, previous approaches [36], [153] have achieved view-invariance by aligning estimated 3D skeletons to a canonical form. In the state-of-the-art [19], [22], [24], [89], [162], [163], we distinguish mainly two variants of skeleton alignment. The first one [24], [162] starts by defining a reference skeleton at the rest state and assuming that the first skeleton (or first frame) of each sequence is at the rest state. A transformation matrix is therefore optimized between the first frame of each sequence and the reference skeleton. Then, the optimized matrix is applied to the full sequence. The major drawback of such an approach is the strong assumption stating that the first pose is at the rest state. Indeed, it is not necessarily the case in real-world applications. To avoid that, a second class of alignment techniques [22], [89], [163] has constructed a coordinate system attached to each skeleton of the sequence and has aligned each skeleton to the absolute coordinate system. Nevertheless, this kind of methods is very sensitive to skeleton noise since the local coordinate system is built using only three joints (that are potentially noisy). Furthermore, partial motion can be lost since every skeleton of the sequence is aligned to the same coordinate system.

Moreover, 3D pose estimation can introduce noise to the motion representation, since poses are independently estimated from each frame. To avoid relying on such a pre-processing, the intuitive solution would be to design 3D skeleton-based view-invariant features. Meanwhile, hand-crafting view-invariant features may not be discriminative enough for the task of action recognition, as discussed in [36]. Thus, the question is *how to ensure view-invariance and keep discriminative information at the same time while estimating* Φ_2*?*

6.4 DeepVI: A Novel Framework for View-Invariant Action Recognition

In this chapter, we propose a novel framework called DeepVI able to learn view-invariant features from a single RGB camera. To ensure their discriminative power, a DNN is used to learn the view-invariant features instead of heuristically designing them. Fig. 6.2 presents an overview of the full framework. First, a 3D pose estimation providing the 3D position of human skeleton joints is applied to 2D images. Then, since the estimated 3D skeletons are fully 3D, synthetic 3D skeleton sequences are generated from virtual viewpoints and used for training our network. More details about this step, that we call *data adaptation*, are presented in Section 6.4.1. Our network is composed of two modules trained in an end-to-end manner, namely SmoothNet and ST-GCN [25]. Since the estimated 3D skeletons incorporate noise, the first module allows the implicit smoothing of skeleton joint trajectories. This newly introduced module is depicted in Section 6.4.2. Afterward, the state-of-the-art ST-GCN network [25] is used for extracting the features and recognizing the actions while taking into account the structure of the skeleton sequence. The integration of the ST-GCN network in our framework is discussed in Section 6.4.3.

6.4.1 Data Adaptation

Let us denote an RGB sequence captured from the viewpoint i by $\mathrm{V}^{(i)}$ with T its temporal length. Thanks to the recent advances in deep learning, it became possible to estimate a relatively

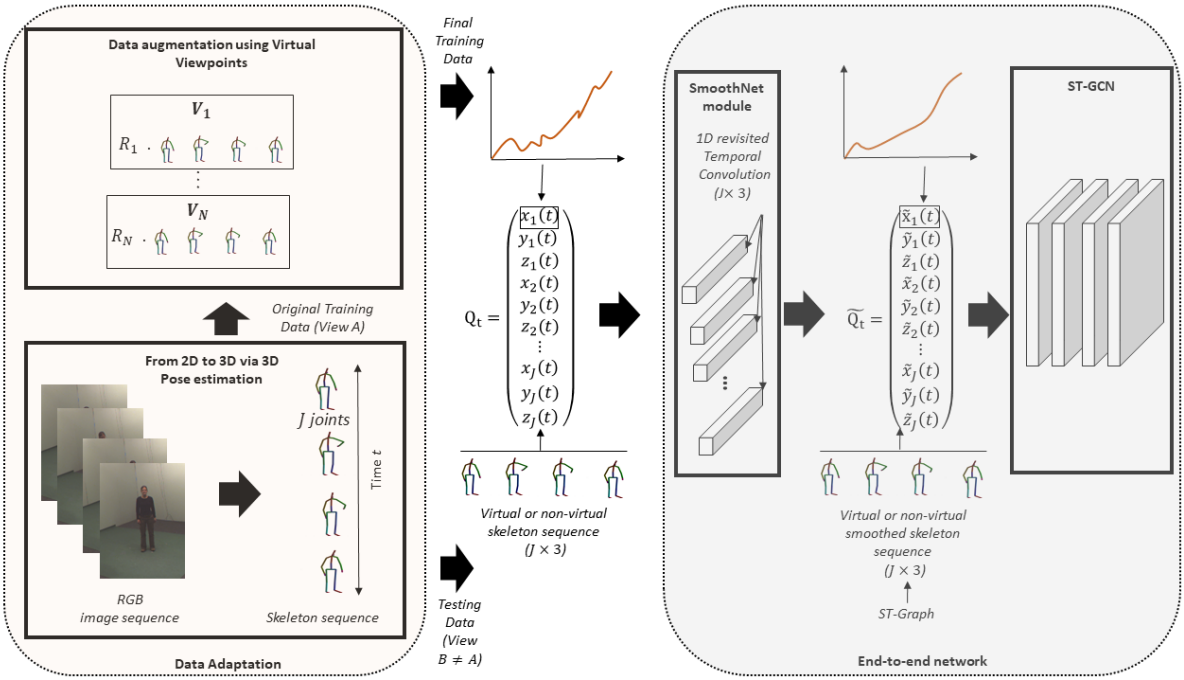


Figure 6.2: Overview of the full framework: our framework is composed of two main components. Thanks to the first component called data adaptation, 3D poses are estimated directly from RGB sequences, and each sequence is rotated according to the position of the virtual cameras $V_1, V_2, ..., V_N$. The augmented sequences are given as input to the end-to-end network representing the second component of our framework. The end-to-end network is composed of two modules. The first module, called SmoothNet implicitly smooths the joint component trajectories. On the other hand, the second module, named Spatial-Temporal Graph Convolutional Networks (ST-GCN), [25] learns the view-invariant features and recognizes the actions.

accurate 3D human pose from a single RGB image. Thus, a function Φ_1 given by a pose estimator such as [35], [52], [55], [134], [154] can be used to estimate a skeleton of J joints per frame, as follows,

$$\Phi_1(\mathrm{V}^{(i)}) = (\Phi_1(\mathrm{V}^{t,(i)}))_{1 \leq t \leq T} = \hat{\mathrm{Q}}, \tag{6.2}$$

such that $\forall\ t \in \{1, ..., T\}$. Note that $x_j(t), y_j(t)$ and $z_j(t)$ refer to the 3D coordinates of the joint j at instant t.

Hence, the availability of 3D information importantly simplifies the problem of view-invariance. As mentioned in Section 3, for better applicability in real-world scenarios, it is preferable to design view-invariant features rather than applying an alignment as a pre-processing step.

However, hand-crafted view-invariant features may not be informative enough, as shown in [36]. To guarantee the informativeness of view-invariant descriptors, we propose to use a deep neural network to learn them. Nevertheless, the paradox is that, commonly, deep neural networks require multi-view samples as training data in order to learn view-invariant features. This contradicts the main idea of cross-view action recognition. To bridge this gap, we propose to augment the 3D poses using virtual viewpoints.

To create realistic viewpoints, we assume that virtual cameras are uniformly placed around the skeleton. Then, the rotation matrices allowing the observation of the same skeleton sequence from these respective virtual cameras are estimated. Finally, we apply to each sequence these rotation matrices in order to augment the data. The augmented data are then given as input to the proposed DNN during the training phase. By doing that, we expect the DNN to be able to learn view-invariant features and consequently estimate a suitable function Φ_2.

6.4.2 SmoothNet: An Implicit Smoothing

The first module of our DNN termed SmoothNet and composed of revisited Temporal Convolutional Networks (TCN), is introduced in order to smooth the joint trajectory components in an end-to-end manner. 3D trajectory joints include noise that can be observed as small visible oscillations along time. This noise is mainly due to the quality of the used 3D pose sequences that are estimated per-frame. In this section, we start by recalling TCN. Then, we revisit them and present the proposed SmoothNet module.

Temporal Convolutional Networks TCN [155] are a class of Convolutional Neural Networks able to model the temporal information included in a sequence. As its name implies, convolutions are applied in the temporal domain instead of the spatial one. A layer of TCN consists of $J \times 3$ temporal filters $\mathbf{W} = \{\mathbf{W}_i\}_{1 \leq i \leq J \times 3}$ with $\mathbf{W}_i \in \mathbb{R}^{d^l \times T_w}$ and T_w the size of the temporal window. Given $\mathcal{S}^l$ the d^l-dimensional input signal of the layer l, the output signal of dimension d^{l+1} denoted by $\mathcal{S}^{l+1}$ is computed as follows,

$$\mathcal{S}^{l+1} = \text{Pool}(f(\mathbf{W} * \mathcal{S}^l + b)), \tag{6.3}$$

where Pool, $f(.)$, b and $*$ respectively refer to, the used pooling technique, the activation function, the bias and the convolution operator.

SmoothNet: Revisited TCN as Weighted Average Filters We propose to adapt TCN to act as a weighted average smoother on joint trajectory components. Weighted average filters represent low-pass filters; thus they are able to reduce high frequencies contained in a signal [164]. For that purpose, we revisit TCNs by defining the activation function as the identity function $f(x) = x$, assuming that the bias b is equal to 0 and neglecting the pooling operator such that,

$$\mathcal{S}^{l+1} = \mathbf{W} * \mathcal{S}^{l}. \tag{6.4}$$

Naturally, $\mathcal{S}^{l+1}$ and $\mathcal{S}^{l}$ are assumed to be single-dimensional signals such that $d^{l} = 1$ and $d^{l+1} = 1$. Furthermore, to impose a weighted average structure to the filters ($\forall k, \mathbf{W}(k) > 0$ and $\|\mathbf{W}\|_1 = 1$, with $\| \ \|_1$ the L_1 norm), we follow the same idea proposed in [165], which has been initially proposed for a faster convergence of DNNs. Optimization is done in the original $\mathbf{W}$-parameterization space. Then, the normalization is applied after each step of stochastic gradient descent such that,

$$\mathbf{W}(k) \longleftarrow \frac{\mid \mathbf{W}(k) \mid}{\|\mathbf{W}\|_1}, \forall k \in \{1, .., T_w\}. \tag{6.5}$$

To maintain the skeleton structure, we define $J \times 3$ TCN blocks that are applied to each 1D-component sequence $x_j(t), y_j(t)$ and $z_j(t), \forall j$ of $\hat{\mathrm{Q}}_t \in \mathbb{R}^{J \times 3}$ independently. The output which represents the smoothed skeleton is denoted by $\tilde{\mathrm{Q}}_t$. Fig. 6.3 depicts the structure of the proposed architecture called SmoothNet. The main advantage of such a module is that the weights of the filters are not empirically chosen, but are trained in an end-to-end manner; thus, particularly trained for the task of action recognition.

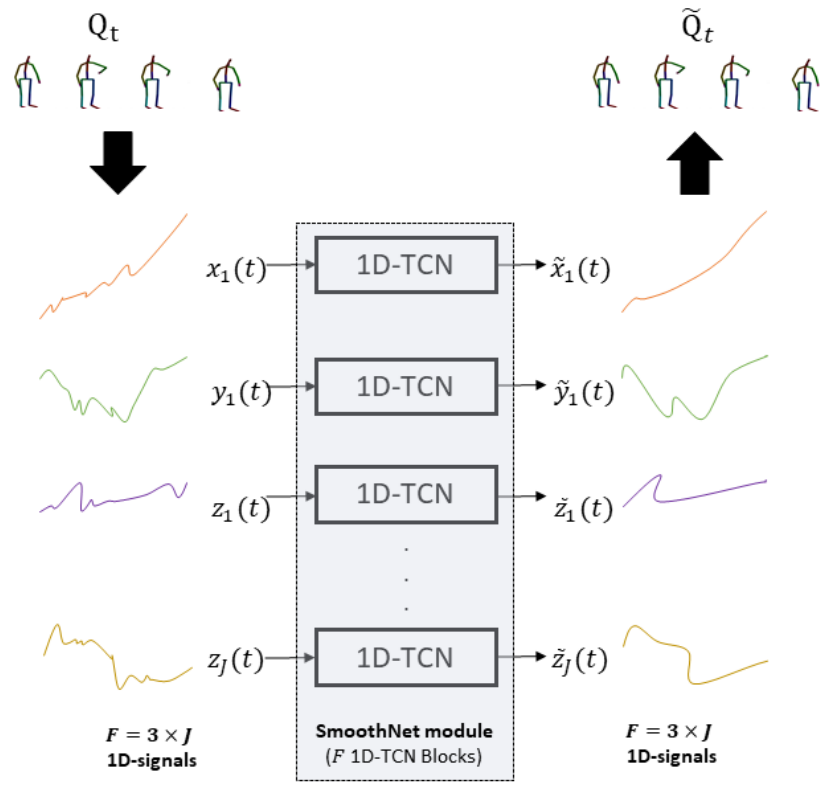


Figure 6.3: Structure of the SmoothNet module: it is composed of $J \times 3$ revisited 1D-TCN blocks. Each skeleton joint component trajectory is fed into one block. The outputs consists in smoothed skeleton joint component trajectories such as the skeleton sequence structure is conserved.

6.4.3 ST-GCN [25]

Finally, for extracting the view-invariant features and recognizing the actions, we use the state-of-the-art network called ST-GCN [25]. We choose this specific architecture since it has been proven to be one of the most successful approaches for 3D skeleton-based human action recognition. Nevertheless, it is important to note that this module can be replaced by any other 3D skeleton-based action recognition DNN. Since the graph structure of the skeleton is conserved after passing the data through the SmoothNet module, it is possible to directly provide the augmented and filtered skeleton sequences $\tilde{Q}$ as input to ST-GCN. The ST-GCN models skeleton sequences as a spatial temporal graph taking into account spatial and temporal connections. While spatial edges are defined based on the skeleton structure, temporal edges

connect the same joint across time. As in [25], we follow the same formula initially proposed in [166] to compute graph convolutions,

$$\mathcal{F}_{out} = \mathbf{\Lambda}^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I})\mathbf{\Lambda}^{-\frac{1}{2}}\mathcal{F}_{in}\mathbf{W}', \tag{6.6}$$

where $\mathbf{A}$ is the adjacent matrix which represents the intra-body connections of joints, $\mathbf{I}$ is the identity matrix, $\mathbf{W}'$ is the weight matrix and $\mathbf{\Lambda}^{ii} = \sum_j(\mathbf{A}^{ij} + \mathbf{I}^{ij})$. $\mathcal{F}_{out}$ and $\mathcal{F}_{in}$ refer respectively to the output and the input graphs.

The entire network, including the SmoothNet and the ST-GCN modules, is trained in an end-to-end manner. A cross-entropy loss function L is optimized during training,

$$L = -\sum_{i=1}^{M} \mathcal{Y}_i \log(\Pi_i), \tag{6.7}$$

where M is the number of classes, $\mathcal{Y}_i$ is the binary number indicating if the class label i is the correct classification and Π_i denotes the predicted probability of belonging to the class i.

6.5 Experiments

Our framework is tested on two cross-view action recognition benchmarks, namely NTU RGB+D (NTU) [167] and Northwestern-UCLA Multiview Action3D (NW-UCLA) [136] datasets.

6.5.1 Datasets and Experimental Settings

NTU RGB+D Dataset (NTU): This Kinect-based dataset is composed of 60 different activities performed by 40 subjects. Three Kinect cameras have been used and placed at $-45°, 0°$, and $45°$ with respect to the human body. To increase the viewpoint variability, on each setup, the height and distance of the camera were changed. In our experiments, we follow the same cross-view protocol proposed in [167]: data captured with cameras 2 and 3 are used for training, while data acquired with camera 1 are used for testing.

Northwestern-UCLA Multiview Action3D Dataset (NW-UCLA): This dataset is composed

of 1494 sequences including 10 different actions. Each action is performed by 10 subjects 1 to 6 times. Three Kinect sensors have been used and placed at three different viewpoints. We follow the same experimental protocol proposed in [136], where two cameras are used for training and one for testing.

6.5.2 Implementation Details

For extracting 3D skeletons from RGB videos, we use the VNect method introduced in [35]. The rest of the framework is implemented using PyTorch. We augment the data by 5 additional viewpoints on NTU and by 20 on NW-UCLA. The size of the filters of the SmoothNet module is set to 3. We use the same parameters suggested in [25] for the ST-GCN network when testing on NTU. However, on NW-UCLA, only 2 Spatial temporal Graph Convolutional layers are used, and dropout is set to 0 since this dataset contains a lower amount of data. Stochastic Gradient Descent optimizer is used with a decaying learning rate of 0.01.

Table 6.1: Comparison of our framework with state-of-the-art methods: Accuracy of recognition (%) on NTU dataset and NW datasets with cross-view settings is reported. *A fine-tuning of a trained model on NTU has been carried out to reach this performance. **These approaches are based on a pre-processing of alignment.

Method	**NTU**	**NW-UCLA**	**Modalities**
STA-Hands [168]	88.6%	-	RGB + Skeleton
STA-Pose [169]	94.2%	93.1%	RGB + Skeleton
VA-fusion [170]	95.0%	81.4%	RGB + Skeleton
3D-BCSM [171]	91.8%	94.0%	Skeleton
ST-GCN [25]	88.8%	-	Skeleton
DVV [144]	-	51.0%	RGB
nCTE [132]	-	63.0%	RGB
NKTM [133]	-	69.4%	RGB
DLVIF [151]	-	77.2%	RGB
VNect + KSC [36]	-	77.5%**	RGB
PEM [172]	84.2%	-	RGB
PDA [173]	80.5%	-	RGB
MD [174]	77.2%	86.7%*	RGB
VNect + LSTM [153]	-	79.9%**	RGB
NV [152]	-	78.1%	RGB
DeepVI (SmoothNet+DA+ST-GCN) (Ours)	$\mathbf{83.7}\%$	$\mathbf{78.3}\%$	RGB

6.5.3 Results

Comparison with the State-of-the-art As in [133], we report methods that only consider unseen views and for which no correspondence with the target view is available at the training time such as DVV [175], nCTE [132], NKTM [133], DVLIF [151], PEM [172], PDA [173], MD [174], VNect+LSTM [153], VNect+KSC [36] and NV [152]. The obtained results on NTU and NW-UCLA datasets are reported in Table 6.1.

On the NTU dataset, we achieve one of the best scores among state-of-the-art methods with an accuracy of 83.7%. Indeed, our approach outperforms the RGB-based methods proposed in [173], [174]. However, it can be noted that the approach proposed in [172] reaches a higher accuracy with 84.2%. Our slightly lower performance can be explained mainly by the limitations of the pose estimator (VNect [35]). Indeed, while NTU contains activities involving two persons, VNect is able to estimate the skeleton of only one person. Furthermore, as shown in [36], the performance of VNect is lower in the presence of extreme viewpoints. This is confirmed by the lower performance of ST-GCN when VNect-estimated poses are given as input instead of RGB-D skeletons. The accuracy drops from 88.6% to 79.2%. However, 3D pose estimation is actively being investigated by the computer vision community. Thus, in future works, the use of a more robust multi-person pose estimator such as [176] can importantly improve our performance.

On the NW-UCLA dataset, our framework outperforms most of state-of-the-art methods such as DVV [175], nCTE [132], NKTM [133], DLVIF [151], VNect+KSC [36] and NV [152] with an accuracy of 78.3%. The best performing RGB-based state-of-the-art approach is the method proposed in [174]. Notwithstanding, the learned model on NTU is fine-tuned on NW-UCLA since NW-UCLA contains a limited number of instances. In contrast to [174], we train the classification model from scratch, using only NW-UCLA data. Thus, comparing the performance of the two approaches on NW-UCLA is not entirely fair. Furthermore, it can be noted that on NTU, our framework exceeds this method in terms of accuracy by more than 5%. VNect+LSTM [153] also slightly outperforms our framework by only 1.6%. However, this approach is dependant on a pre-processing of alignment, which makes it unsuitable for real-world scenarios. This is mainly due to the strong assumption that sequences are segmented and that the first pose of each

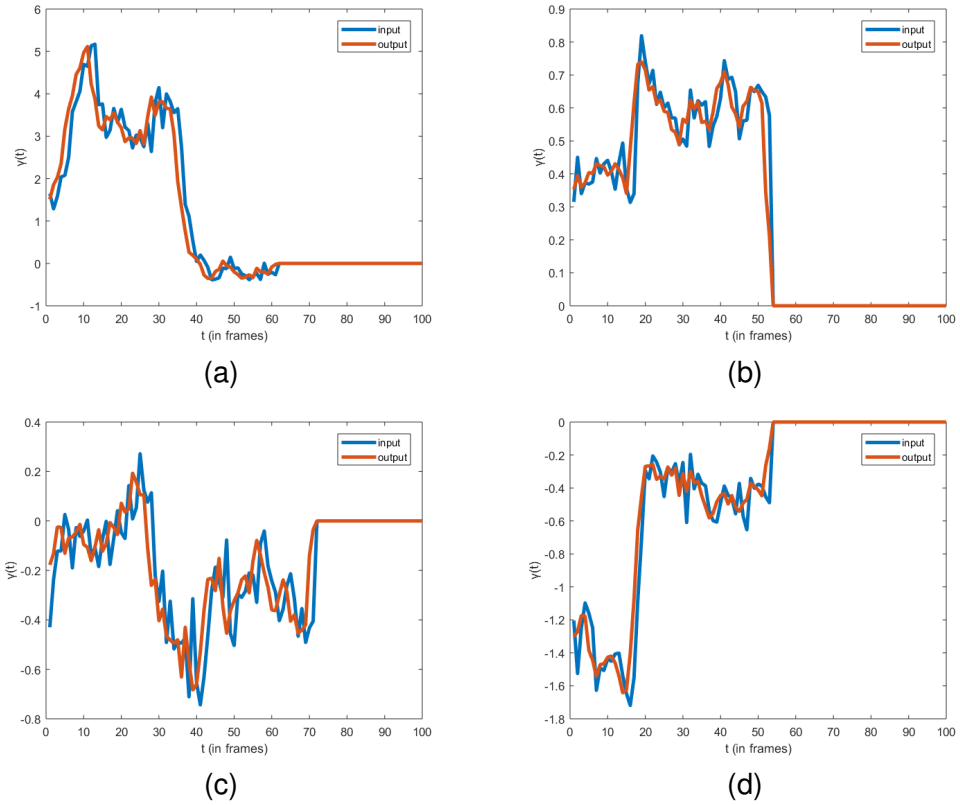


Figure 6.4: Some qualitative results of SmoothNet on 4 different input signals (a), (b), (c) and (d). The input signal is shown in blue, whereas the smoothed output signal is shown in orange. $\gamma(t)$ represents one of the joint trajectory components.

sequence is at the rest state.

Table 6.2: Ablation study: Accuracy of Recognition (%) on NTU and NW-UCLA datasets with Cross-View settings

Method	**NTU**	**NW-UCLA**
ST-GCN+VNect	79.2%	60.7%
SmoothNet+ST-GCN+VNect	82.9%	61.8%
DeepVI (Ours)	83.7%	78.3%

Ablation Study In order to analyze the contribution of each component of our framework, an ablation study is carried out. For this purpose, we remove each time a component and report the obtained performance on both NTU and NW-UCLA datasets. The results are reported in Table 6.2.

The baseline, representing the original approach ST-GCN when tested using VNect estimated skeletons (ST-GCN+VNect) reaches 79.2% on NTU and 60.7% on NW-UCLA. On the one hand, the use of the SmoothNet shows a slight improvement of the accuracy by almost 4% on NTU and around 1% on NW-UCLA.

On the other hand, the data augmentation step (denoted as DA in Table 6.2) significantly improves the results on NW-UCLA by reaching 78.3%, while enhancing them by less than 1% on NTU. This marginal improvement may be due to the high variation encoded in NTU. During the acquisition of the same viewpoint, the height and the distance of the camera were changed, as mentioned in Section 6.5.1. As a result, the dataset implicitly incorporates more than three viewpoints. Thus, introducing more viewpoint variation by augmenting the data does not necessarily contribute importantly to improve the results.

Moreover, we performed the aforementioned experiments using the provided RGB-D skeletons as input in order to prove the added value of our approach to the original ST-GCN. As shown in Table 6.3, we prove that our method applied to RGB-D skeletons improves the ST-GCN results by reaching 90.4%, instead of 88.8%.

Table 6.3: Accuracy of Recognition (%) using NTU VNect skeletons and NTU RGB-D skeletons.

Method	**NTU VNect**	**NTU RGB-D**
ST-GCN	79.2%	88.8%
SmoothNet+DA+ST-GCN (ours)	83.7%	90.4%

Qualitative Results Since the SmoothNet module is one of the central contributions of this chapter, we present some qualitative results. In Fig. 6.4, the input (blue) and the output (orange) signals of four random blocks contained in the SmoothNet module are shown. The plotted curves confirm our assumption stating that SmoothNet allows the smoothing of the 1D-signals inputs.

Indeed, in the fourth graphs, we can observe that the global shape of the curves is conserved, while small fluctuations are attenuated.

6.6 Conclusion

In this chapter, a simple yet effective solution is presented for the problem of cross-view action recognition from a single monocular RGB camera. In this context, we exploit the advances in 3D pose estimation from RGB sequences in an attempt to achieve view-invariance. Two emerging issues are also addressed. The first issue concerns the view-invariance from 3D data. Instead of relying on pose alignment, we propose to augment the training viewpoints and enforce the network to learn view-invariant features. The second issue is the noisy estimates of 3D skeleton sequences, which degrade the performance of action recognition. For this purpose, a filtering module called SmoothNet is introduced and trained in an end-to-end manner. In order to validate our approach, we rely on the Spatial-Temporal Graph Convolutional Network, which has shown significant effectiveness in skeleton-based action recognition. Nevertheless, our approach can be merged with any other DNN designed for skeleton-based action recognition. The obtained results on two datasets show the efficiency of our framework. Some improvements can still be made. For example, in future works, we will attempt to constrain the descent gradient to optimize the weights in the normalized parameter space. Furthermore, we aim at incorporating the data augmentation in the end-to-end network instead of performing it empirically. In the next chapter, we introduce two novel modules for ST-GCNs, which further improve their efficiency.

Chapter 7

Vertex Feature Encoding and Hierarchical Temporal Modeling in a Spatial-Temporal Graph Convolutional Network for Action Recognition

Spatio-temporal Graph Convolutional Networks (ST-GCNs) have shown great performance in the context of skeleton-based action recognition. Nevertheless, ST-GCNs use raw skeleton data as vertex features that have low dimensionality. Moreover, the temporal convolution in these networks can be insufficient for capturing both long-term and short-term dependencies. In this chapter, we introduce a Graph Vertex Feature Encoder (GVFE) module for encoding vertex features into a new feature space. We also propose a Dilated Hierarchical Temporal Graph Convolutional Network (DH-TCN) module for modeling both short-term and long-term temporal dependencies using a hierarchical dilated convolutional network. These two modules allow the design of a more compact and efficient graph-based framework for action recognition trained in an end-to-end manner. Finally, we conduct experimental validation and analysis of our approach on two challenging datasets.

7.1 Introduction

In skeleton-based action recognition [19], [36], [162], [177], [178], skeleton sequences are represented as vectors or 2D grids, ignoring inter-joint dependencies. To express joint correlations both spatially and temporally, Yan, Xiong, and Lin introduced the Spatial Temporal-Graph Convolutional Network (ST-GCN) [25]. Their work takes advantage of Graph Convolutional Networks (GCN) [179] extending the classical CNNs to graph convolutions. This architecture represents skeleton sequences as a graph composed of both temporal and spatial edges, by respectively considering the inter and intra-frame connections of joints. The effectiveness of this approach has motivated several extensions [180]–[182] which, consider the most informative connections between joints instead of the predefined natural skeleton structure or construct the spatio-temporal graphs using additional features such as bone lengths.

However, all these methods only use raw skeleton features (joint coordinates and/or bone lengths) for the construction of spatio-temporal graphs. While offering a high-level description of the human body structure, these features have low dimensionality and thus may be lacking discriminative power for action recognition. Indeed, hand-crafted approaches have shown the limitation of using only raw skeleton joints as features in action recognition [24], [30]. Furthermore, the temporal dependencies of the graph are modeled by a single temporal convolutional layer. As a result, critical long-term dependencies might be not consistently described. Moreover, these approaches make use of a considerable number of ST-GCN blocks (10, in most cases), which significantly increases the number of parameters and consequently the computational complexity and the required memory.

In this chapter, we assume that by encoding the vertex features in an end-to-end manner and modeling temporal long-term and short-term dependencies, less number of layers (and consequently parameters) will be needed. For that reason, two modules are introduced. The first module is referred to as Graph Vertex Feature Encoder (GVFE). GVFE is a trainable layer that transforms the feature space from the Euclidean coordinate system of joints to an end-to-end learned vertex feature space, optimized jointly with the ST-GCN. The new feature space offers more robust discriminative capabilities as a result of its higher dimensionality [183]. The second

module incorporates a hierarchical structure of dilated temporal convolutional layers for modeling short-term and long-term temporal dependencies by increasing the temporal receptive field in multiple levels. It is termed Dilated Hierarchical Temporal Graph Convolutional Network (DH-TCN) and replaces the standard temporal convolutional layers found in the ST-GCN block. With the use of these two modules, we show that fewer layers are needed to reach the same or even higher performance in action recognition while needing less memory and training time than previous ST-GCN based approaches such as [182].

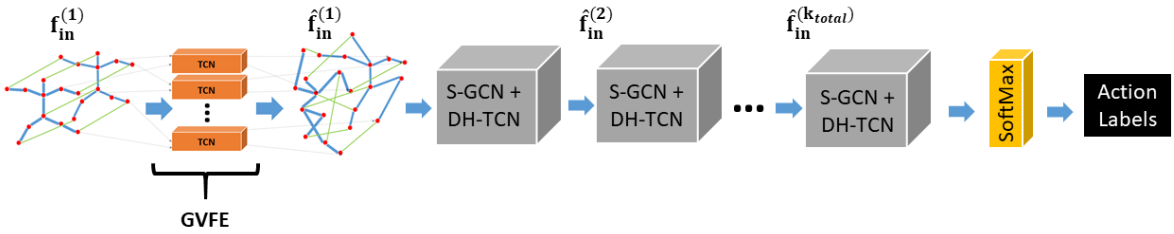


Figure 7.1: Illustration of the proposed approach. In the first step, the GVFE module generates graph features. The new graph is given as an input to the Modified ST-GCN blocks composed of a Spatial-Graph Convolutional Network (S-GCN) and a Dilated Hierarchical Temporal Convolutional Network (DH-TCN). Finally, a SoftMax layer classifies the spatial-temporal graph features resulting from the last Modified ST-GCN block.

7.2 Related Work

Over the last decade, the availability of 3D skeletons through RGB-D sensors has significantly boosted the development of numerous skeleton-based action recognition methods. Earlier methods have mainly introduced novel hand-crafted features aiming to describe the human motion. For example, human skeleton sequences can be modeled as trajectories lying in Euclidean or Riemannian spaces [24], [36], [46], [177], [184], as statistical-based representations [19], [162] or as pairwise relative positions of joints [65], [87], etc.

Recently, deep-learning-based approaches have gained popularity and have shown notable performance, especially on large-scale datasets [21], [22], [89], [185]–[191]. Instead of hand-crafting features, deep-learning-based approaches are able to learn them automatically. Long Short-Term Memory (LSTM) networks, initially designed for modeling sequential data, have

particularly shown great potential in action recognition. In fact, compared to conventional Recurrent Neural Networks (RNN), LSTM can handle long-term dependencies and, thus, mitigate the problem of vanishing gradients [22], [89], [186]. However, LSTM-based models cannot be parallelized and thus are generally hard to train. CNN have also shown their efficiency for the action recognition task [185], [187]–[189], [192]–[194].

Nevertheless, both CNN and LSTM fail to exploit the spatio-temporal structure of 3D skeletons that can naturally be seen as graphs rather than Euclidean data. Recently, Graph Convolution Networks (GCN) [179], [195]–[198] generalizing CNN from 2D grids to graphs have been introduced and adopted for skeleton-based action recognition [25], [180], [182], [199], [200]. Yan, Xiong, and Lin [25] were among the first to utilize GCN in skeleton-based action recognition. They represented skeleton sequences as spatio-temporal graphs by preserving the inter-joint connections and linking temporally the same joints from different time steps and consequently designed a suitable network called Spatio-Temporal Graph Convolution Network (ST-GCN). Considering the fact that the graph edges defined by the natural skeleton structure might be not optimal for the task of action recognition, some approaches extended ST-GCN in order to capture more relevant dependencies among joints [180], [182]. To respectively capture action-specific and higher-order dependencies, an encoder-decoder module called A-link inference has been designed and a higher polynomial within the Spatial Graph Convolution has been used [182]. Shi, Zhang, Cheng, and Lu [180] proposed an Adaptive ST-GCN to adaptively learn joint connections in an end-to-end manner. Moreover, they made use of a two-stream network that combines first-order and second-order joint information. On the other hand, Shi, Zhang, Cheng, and Lu [181] extended ST-GCN to Directed acyclic ST-GCN (D-ST-GCN) in order to capture the relationship between bones and joints. Si, Chen, Wang, Wang, and Tan [200] were the only ones attempting to extend the temporal modeling of ST-GCN that considers only short-term dependencies. To that aim, they introduced the Attention Enhanced Graph Convolutional Long Short-Term Memory network (AGC-LSTM). Despite the relevance of such an approach, LSTM remains difficult to parallelize, as mentioned earlier in this section. Furthermore, all the presented graph-based approaches rely solely on the joint and/or bone length features which might not be optimal for action recognition. In Section 7.3.1 and Section 7.3.2, two novel graph-based

modules aiming to overcome the two mentioned issues are presented.

7.3 Proposed Approach

In this section, the two novel modules, namely GVFE and DH-TCN, are presented. While GVFE aims at learning vertex features, DH-TCN temporally summarizes spatio-temporal graphs and consequently models long-term as well as short-term dependencies. These two modules are integrated with the original ST-GCN [25] framework. This full pipeline is depicted in Fig. 7.1 and is trained in an end-to-end manner. It is important to note that these modules are also complementary to other ST-GCN extensions such as AS-GCN [182].

7.3.1 Graph Vertex Feature Encoding (GVFE)

In ST-GCN [25], for an input feature map $\mathcal{F}_{in}$, a spatial graph convolution is applied, such that:

$$\mathcal{F}_{out} = \mathbf{\Lambda}^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I})\mathbf{\Lambda}^{-\frac{1}{2}}\mathcal{F}_{in}\mathbf{W}, \tag{7.1}$$

where $\mathcal{F}_{out}$ is the output feature map, $\mathbf{A}$ the adjacency matrix, $\mathbf{I}$ the identity matrix, $\mathbf{\Lambda} = [\mathbf{\Lambda}^{ii}]_{i\in\{1,...,J\}}$ such that $\mathbf{\Lambda}^{ii} = \sum_j(\mathbf{A}^{ij}+\mathbf{I}^{ij})$ and $\mathbf{W}$ is the weight matrix. For a graph of size (C_{in}, J, T), the dimension of the resulting tensor is (C_{out}, J, T), with C_{in} and C_{out} denoting respectively the number of input and output channels.

The input features $\mathcal{F}_{in}^{(1)}$ incorporated in the first ST-GCN layer correspond to the joint coordinates such that $\forall i$, $\mathcal{F}_{in}^{(1)}(v_i, t) = \mathsf{q}_t^i$ with v_i the vertex of the graph corresponding to joint i, q_t^i the 3D coordinate of the joint i at an instant t and consequently $C_{in} = 3$.

As mentioned in Section 7.1, considering raw skeleton joint data as vertex features might not be informative enough for action recognition. The dimensionality of the raw skeleton joints is low and consequently not sufficient enough for effective feature discrimination. To enhance the discriminative power of vertex features, we introduce the GVFE module that is directly placed before the first ST-GCN block. GVFE maps 3D skeleton coordinates, traditionally used as input features to the first ST-GCN block $\mathcal{F}_{in}^{(1)}(v_i) = \mathsf{Q}^i$ with $i \in \{1, ..., J\}$, from the Cartesian coordinate

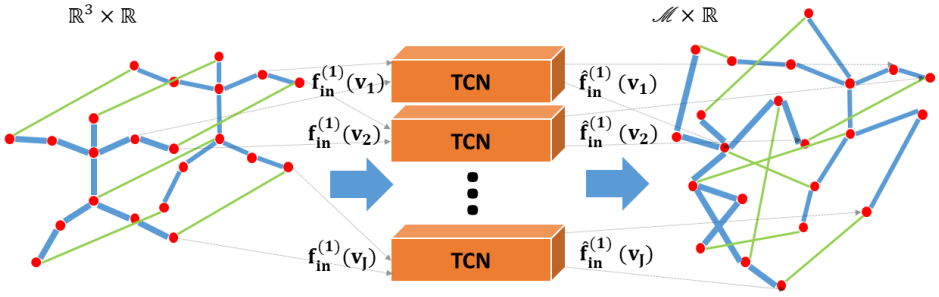


Figure 7.2: Illustration of the GVFE module structure: it is composed of J TCN blocks. For each joint, one TCN block is separately used in order to conserve the natural skeleton structure.

system $\mathbb{R}^3$ to a learned feature space $\mathcal{M} \subseteq \mathbb{R}^{C_{out}}$ of higher dimensionality $C_{out} > 3$. The higher dimensionality offers robust discriminative capabilities and better generalization, as discussed in [183]. GVFE module preserves the spatial structure of skeletons so that the joint dependencies are modeled. Since this module is trained in an end-to-end manner by optimizing the recognition error, we expect to obtain a more sufficient for action recognition feature space $\mathcal{M}$.

For each joint i, a separate Temporal Convolutional Network (TCN) is employed to encode raw data, as illustrated in Fig. 7.2. In this context, TCNs show strong potential since (a) they do not allow information to flow from the future states to the past states, (b) the input and output sequences have the same length and (c) they model temporal dependencies. For each joint i, the new graph vertex features $\hat{\mathcal{F}}_{in}^{(1)}(v_i)$ obtained after applying the TCN are computed as follows,

$$\hat{\mathcal{F}}_{in}^{(1)}(v_i) = \mathbf{W}_i^{TCN} * \mathcal{F}_{in}^{(1)}(v_i) = \mathbf{W}_i^{TCN} * \mathrm{Q}^i, \tag{7.2}$$

where $\{\mathbf{W}_i^{TCN}\}_{1 \leq i \leq J}$ is the collection of tensors containing the kernel filters $\{\mathbf{W}_{i,j}^{TCN}\}$ of dimension $\mathbb{R}^{C_{out} \times T_w \times C_{in}}$, with $j \in \{1, ..., C_{out}\}$ the index of the filter and T_w the temporal size of the filters. Note that we use the identity activation function. This module preserves the skeleton structure and has the advantage of being applicable to any graph-based network, regardless of the application.

7.3.2 Dilated Hierarchical Temporal Graph Convolutional Network

The modeling of temporal dependencies is crucial in action recognition. However, in several ST-GCN-based approaches [25], [180]–[182], temporal dependencies are modeled using only one convolutional layer. As a result, long-term dependencies that can be important for modeling actions are not well encoded.

To that end, we propose to replace the temporal convolutions of each ST-GCN block with a module that encodes both short-term and long-term dependencies. Given the output feature map $\mathcal{F}_{out}^{(k)}$ resulting from the k^{th} Spatial GCN (S-GCN) block (with $k \in [1, k_{total}]$ and k_{total} the total number of ST-GCN blocks), this module, termed Dilated Hierarchical Temporal Convolutional Network (DH-TCN), is composed of N successive dilated temporal convolutions. The association of these two blocks is illustrated in Fig. 7.4. Each layer output $\mathcal{F}_{temp}^{(k,n)}$ of order n of DH-TCN is obtained as follows,

$$\mathcal{F}_{temp}^{(k,n)} = F\left(\mathbf{W}_i^{DH} *_\lambda \mathcal{F}_{temp}^{(k,n-1)}\right), \text{ with } \mathcal{F}_{temp}^{(k,0)} = \mathcal{F}_{out}^{(k)}, \tag{7.3}$$

where $\{\mathbf{W}^{DH}\}_{1 \leq i \leq J}$ is the tensor containing the trainable temporal filters of dimension $\mathbb{R}^{C_{out} \times T_{w_1} \times C_{out}}$ with T_{w_1} their temporal dimension and $*_\lambda$ refers to the convolution operator with a dilation of $\lambda = 2^n, n \in [0, N-1]$.

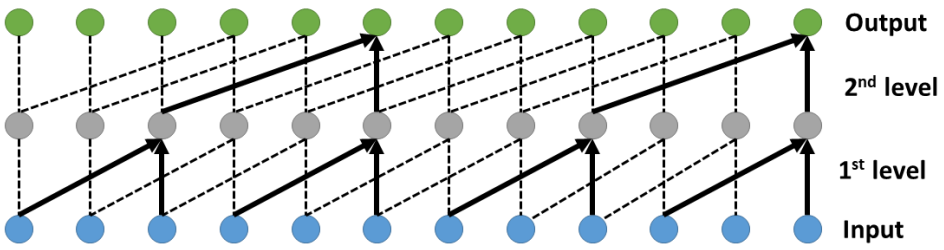


Figure 7.3: Example of a 2-level dilated convolution on an input sequence. The first level encodes short-term dependencies, while the second level increases the receptive field and encodes longer-term dependencies.

The hierarchical architecture with different dilation ensures the modeling of long-term dependencies. Dilated convolutions are proven to be efficient in modeling long-term dependencies

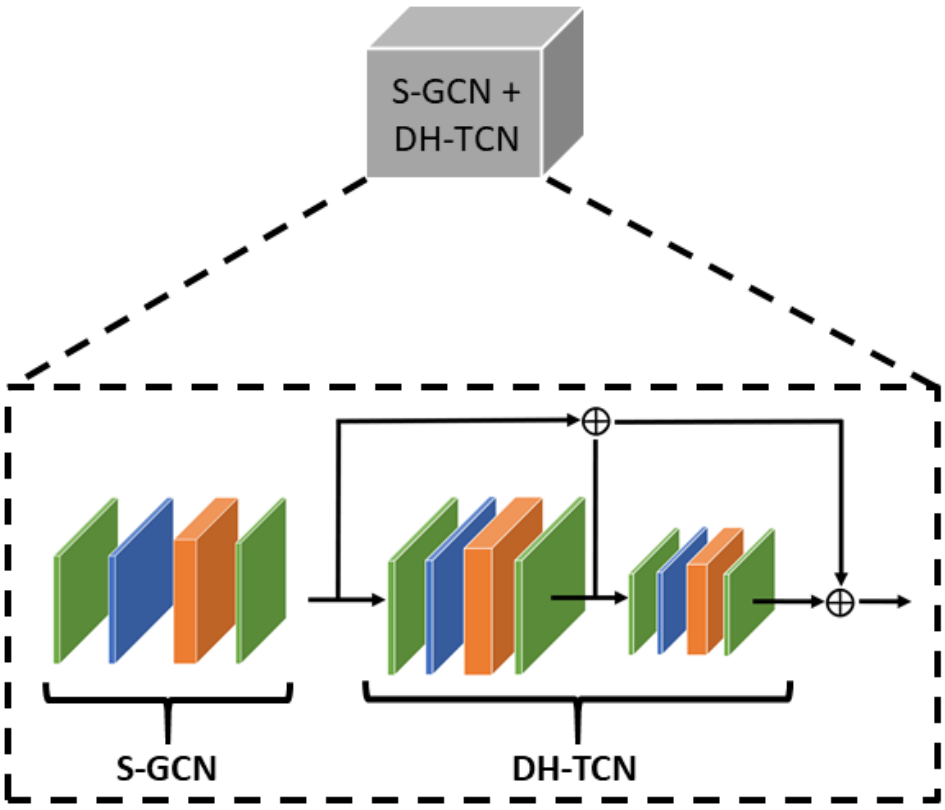


Figure 7.4: Illustration of S-GCN + DH-TCN block. Spatial features are extracted from the S-GCN module and are, then, fed into DH-TCN module. Green color is used for Batch Normalization units, blue for ReLU and orange for 2D Convolutional Layers.

[59] while at the same time they maintain efficiency. Architectures with dilated convolutions have been successful for audio generation [201], semantic segmentation [202] and machine translation [203]. An example of how dilated convolutions are applied in a two-level manner is illustrated in Fig. 7.3. At the same time, the residual connection depicted in Fig. 7.4 enables the preservation of the information of short-term dependencies.

The entire DH-TCN module is illustrated in Figure 7.4. Each hierarchical layer is composed of a dilated temporal convolution, a ReLU activation function, and a batch normalization.

7.4 Experiments

Our framework has been tested on two well-known benchmarks, namely NTU RGB+D 60 (NTU-60) [167] and NTU RGB+D 120 (NTU-120) [204] datasets.

7.4.1 Datasets and Experimental Settings

Table 7.1: Accuracy of recognition (%) on NTU-60 and NTU-120 datasets. The evaluation is performed using cross-view and cross-subject settings on NTU-60 and cross-subject and cross-setup settings on NTU-120. *These values have not been reported in the state-of-the-art and the available codes have been used to obtain the recognition accuracy of these algorithms on NTU-120.

Method	**NTU-60 (%)**		**NTU-120 (%)**	
	X-subject	X-view	X-subject	X-setup
SkeleMotion [192]	76.5	84.7	67.7	66.9
Body Pose Evolution Map [172]	91.7	95.3	64.6	66.9
Multi-Task CNN with RotClips [205]	81.1	87.4	62.2	61.8
Two-Stream Attention LSTM [206]	76.1	84.0	61.2	63.3
Skeleton Visualization (Single Stream) [163]	80.0	87.2	60.3	63.2
Multi-Task Learning Network [185]	79.6	84.8	58.4	57.9
ST-GCN (10 blocks) [25]	81.5	88.3	72.4*	71.3*
GVFE + ST-GCN w/ DH-TCN (4 blocks - ours)	79.1	88.2	73.0	74.2
AS-GCN (10 blocks) [182]	**86.8**	**94.2**	77.7*	78.9*
GVFE + AS-GCN w/ DH-TCN (4 blocks - ours)	85.3	92.8	**78.3**	**79.8**

NTU RGB+D 120 Dataset (NTU-120): NTU RGB+D 120 Dataset extends the original NTU dataset by adding 60 additional action classes to the existing ones and 66 more subjects. The recording angles remain the same at $-45°, 0°$ and $45°$ with respect to the human body, but more setups (height and distance) are considered (32 instead of 18). We consider the same evaluation protocol (cross-setup and cross-subject settings) suggested in [204].

7.4.2 Implementation Details

The implementation of our approach is based on the PyTorch ST-GCN [25] and AS-GCN [182] codes. In both approaches, we include the GVFE module before the first ST-GCN block and

we replace the temporal convolutions of each block with the DH-TCN module. For the spatial GCN, we use the same parameters suggested in [25]. The number of output channels in GVFE is set to $C_{out} = 8$ and we use $N = 2$ hierarchical modules in DH-TCN. The temporal window of the DH-TCN module is set to $T_w = 9$. The Stochastic Gradient Descent optimizer is used with a decaying learning rate of 0.01. In contrast to [25], [182] that makes use of 10 ST-GCN or AS-GCN blocks, we use only 4 blocks with $k \in \{1, ..., 4\}$.

7.4.3 Results

Comparison with state-of-the-art

In this section, we compare our approach with recent skeleton-based methods, such as Skele-Motion [192], Body Pose Evolution Map [172], Multi-Task CNN with RotClips [205], Two-Stream Attention LSTM [206], Skeleton Visualization (Single Stream) [163], Multi-Task Learning Network [185] and more particularly with two the graph-based baselines namely ST-GCN [25] and AS-GCN [182]. GVFE and DH-TCN modules are incorporated in both ST-GCN [25] and AS-GCN [182] methods. The obtained accuracy of recognition on NTU-60 and NTU-120 datasets are reported in Table 7.1.

On NTU-120, we obtain the best accuracy of recognition of the state-of-the-art for both settings. Indeed, our approach used with AS-GCN (GVFE+AS-GCN w/ DH-TCN) reaches 78.3% and 79.8% for cross-subject and cross-setup settings, respectively. These positive results are also confirmed when testing our approach with ST-GCN (GVFE + ST-GCN w/ DH-TCN). For instance, we improve the accuracy of the original ST-GCN by 0.6% up to 2.9%.

On NTU-60, the achieved scores are among the best of the state-of-the-art but remain slightly inferior to the original ST-GCN and AS-GCN (with respectively $79.1\% - 88.2\%$ against $81.5\% - 88.3\%$ and $85.3\% - 92.8\%$ against $86.8\% - 94.2\%$). Although being slightly inferior, it is important to highlight that only 4 blocks are used in our case (against 10 for ST-GCN and AS-GCN). The method based on Body Pose Evolution Map [172] remains the best performing approach on NTU-60. However, this method registers an accuracy inferior to our approach by $13.7\% - 12.9\%$, while the difference is less important on NTU-60 with only a gap of $6.4\% - 2.5\%$

making our method more stable.

The initial feature space of skeleton joints seems to be sufficient for the NTU-60 dataset. Nevertheless, the NTU-120 dataset contains a significantly larger amount of videos and action classes, making our approach a suitable solution. This is justified by the need of a more discriminative feature space for such a large dataset that is offered by the GVFE module. A more detailed analysis in this follows in the ablation study.

Impact of the number of blocks

As mentioned earlier, our approach utilizes only 4 ST-GCN or AS-GCN blocks instead of 10. For a fair comparison with the baselines, we also test ST-GCN [25] and AS-GCN [182] when using only 4 blocks. The recognition accuracy of these experiments is reported in Table 7.2. Our method (GVFE + ST-GCN w/ DH-TCN) shows a significant performance boost in both settings of over 22% compared to ST-GCN with 4 blocks. Similarly, the recognition accuracy remains higher than the original AS-GCN compared to our method (GVFE + AS-GCN w/ DH-TCN). However, in this case, the accuracy boost is less impressive with an increase of 1.4% for cross-subject settings and 0.4% for cross-setup settings. This could be explained by the 7 extra spatio-temporal convolutional blocks after the *maxPooling* layer in the AS-GCN network, which add more discriminative power to the full pipeline.

Table 7.2: Accuracy of recognition (%) using only 4 ST-GCN or AS-GCN blocks on NTU-120 dataset for cross-subject and cross-setup settings. *These values are not reported in the state-of-the-art. Thus, the available codes have been used to obtain these results.

Method	X-subject	X-setup
ST-GCN (4 blocks) [25]	45.3^*	51.8^*
GVFE + ST-GCN w/ DH-TCN (4 blocks - ours)	$\mathbf{73.0}$	$\mathbf{74.2}$
AS-GCN (4 blocks) [182]	76.9^*	79.4^*
GVFE + AS-GCN w/ DH-TCN (4 blocks - ours)	$\mathbf{78.3}$	$\mathbf{79.8}$

Ablation Study

To analyze the contribution of each component of our framework, an ablation study was conducted. For this purpose, we removed each time a component and report the obtained performance on both NTU-120 dataset for the cross-setup setting. The results are reported in Table 7.3.

Our approach, which combines both the GVFE and the DH-TCN modules, achieves 74.2% mean accuracy, which is higher by 22.4% than the original ST-GCN approach with 4 ST-GCN blocks. When using only the GVFE, the mean accuracy reaches 70.9%. We tested different configurations in this case, such as attaching a Rectified Linear Unit (ReLU) or a Batch Normalization Unit (BN). In both cases, the performance was degraded (68.9% and 66.7%, respectively), since these units distort the joint motion trajectories.

Moreover, we conducted experiments by incorporating only the DH-TCN module. The mean accuracy, in this case, reached 68.3%, showing that GVFE and DH-TCN modules trained in an end-to-end manner can offer a significant performance boost.

Table 7.3: Ablation study: accuracy of recognition (%) on NTU-120 dataset for cross-setup settings using ST-GCN as a baseline. *These values are not reported in the state-of-the-art. Thus, the available codes have been used to obtain these results

Method	**Accuracy (%)**
ST-GCN (4 blocks) [25]	51.8^*
GVFE + ST-GCN (4 blocks)	70.9
ST-GCN w/ DH-TCN (4 blocks)	68.3
GVFE + ST-GCN w/ DH-TCN (4 blocks - ours)	$\mathbf{74.2}$

Number of parameters and training time

Although our method makes use of two additional modules compared to the baselines, the use of only 4 blocks reduces the number of parameters. For instance, When using our method (GVFE + AS-GCN w/ DH-TCN) with 4 blocks, the number of parameters drops from 7420696 to 7370568 compared to the original AS-GCN with 10 blocks, while keeping almost the same accuracy on NTU-60 or even increasing it on NTU-120. Consequently, the training time is also

reduced. As an example, on NTU-120 for cross-setup settings, our approach requires 24029 seconds less than the original AS-GCN for training.

7.5 Conclusion

In this chapter, two novel modules for ST-GCN based methods have been proposed called GVFE and DH-TCN. These modules enable the reduction of the number of needed blocks and parameters while conserving almost the same or improving the recognition accuracy. Instead of relying on raw skeleton features such as skeleton joints, GVFE learns and generates graph vertex features in an end-to-end manner. To model simultaneously long-term and short-term dependencies, DH-TCN makes use of hierarchical dilated temporal convolutional layers. The relevance of these modules has been confirmed thanks to the performance achieved on two well-known datasets. Some future extensions are under consideration, such as applying a similar hierarchical model to replace the spatial graph convolutional layer.

Chapter 8

Conclusions

In this chapter, we summarize the findings and contributions of our research. In addition, we present interesting future directions and open discussions on our research orientation.

8.1 Summary

In this thesis, we addressed some major challenges linked to dense trajectories and sparse trajectories in action recognition and detection. Our first contribution is addressing two major challenges of dense trajectory-based approaches: (a) the lack of locality information relative to the human body and (b) the ineffectiveness in describing radial motion. Towards this direction, we proposed localized trajectories for action recognition. Localized trajectories incorporate locality-awareness derived from the clusters around human body joints. This method also capitalizes on the discriminative power of the local Bag-of-Words concept. Moreover, we extended localized trajectories to 3D to improve the description of radial motion.

In the context of action detection, we proposed a novel way of combining dense trajectories with pose information. A two-stage action detection framework was presented which uses skeleton descriptors to segment temporal regions of interest and then dense trajectories for recognizing the corresponding actions. The stage of temporal segmentation is beneficial for action detection since it removes uninformative background activities.

Sparse trajectories have shown great potential in action recognition. 3D skeleton sequences

have increased their popularity over the last years, especially in cross-view action recognition. However, under certain conditions, the extraction of 3D skeletons can be challenging. Thus, by taking advantage of the recent advances in 3D pose estimation from a monocular RGB camera we proposed a framework for view-invariant skeleton-based representations. In particular, we used a state-of-the-art 3D pose estimator to augment 2D images by a third dimension, and then we tested two different view-invariant skeleton-based approaches, namely, KSC [24] and LARP[45].

In the same direction, we proposed a simple and effective cross-view action recognition framework which uses as input sequences captured by a single monocular RGB camera. Similar to the previous approach, this framework relies on a 3D pose estimator from RGB frames and addresses two emerging issues. The first issue is the invariance to viewpoints from 3D data. For this purpose, we used viewpoint augmentation on training sequences to enforce the learning of view-invariant representations while training a deep neural network. The second challenge is the temporal inconsistency of the estimated 3D poses, which was addressed by a novel filtering module, named SmoothNet. This module is trained in an end-to-end manner with the action recognition module - the Spatial-Temporal Graph Convolutional Network (ST-GCN). Nevertheless, our contribution can be combined with any deep neural network designed for skeleton-based action recognition.

Finally, recognizing the strength of ST-GCN, we proposed two improvements for it. The first improvement concerns the limitations of the input skeleton graph representations. Thus, we developed a module named GVFE. GVFE learns and generates suitable graph vertex features for action recognition. Moreover, the modeling of temporal dependencies has been mitigated by the introduction of DH-TCN, which relies on hierarchical dilated temporal convolutional layers.

8.2 Future Directions

The work conducted throughout this thesis has triggered new research questions. In this section, we present two major future directions of our research which are (a) the generation of 3D dense trajectories from the 3D human body and the weighted viewpoint augmentation.

8.2.1 3D Dense Trajectories from 3D Human Body

As observed in Chapter 3, scene flow estimation can be noisy and impact the extraction of 3D dense trajectories. Moreover, most existing methods rely on a single RGB-D camera for estimating scene flow maps, resulting in self-occlusions and partial views of objects. Therefore, an interesting research direction is the estimation and fitting of 3D body models to 2D data [57], which would allow the calculation of scene flow directly from point cloud data. Consequently, 3D dense trajectories would be less noisy and describe motion from occluded body parts. Similarly, body shape can be more effective than 3D skeletons in scenarios involving subtle motion, such as finger movements.

8.2.2 Weighted Viewpoint Augmentation

In Chapter 6, we introduced a novel framework for cross-view action recognition from RGB sequences. In this framework, we used viewpoint augmentation for enforcing view-invariance in the ST-GCN network. However, synthetic cameras were uniformly placed around the human body and shared equal contributions to the computation of features. As a future direction, we propose soft-voting of the contribution of each synthetic viewpoint. In this concept, the weighting parameters of the viewpoints are trainable in an end-to-end manner with the classification network. Thus, the network will be able to make decisions on the most suitable viewpoints needed for the generation of view-invariant features during training time.

References

[1] S. Varrette, P. Bouvry, H. Cartiaux, and F. Georgatos, “Management of an academic hpc cluster: The ul experience,” in *2014 International Conference on High Performance Computing Simulation (HPCS)*, Jul. 2014, pp. 959–967. DOI: 10.1109/HPCSim.2014.6903792.

[2] R. Baptista., M. Antunes., D. Aouada., and B. Ottersten., “Anticipating suspicious actions using a small dataset of action templates,” in *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP,*, INSTICC, SciTePress, 2018, pp. 380–386, ISBN: 978-989-758-290-5. DOI: 10.5220/0006648703800386.

[3] R. Baptista, M. Antunes, A. E. R. Shabayek, D. Aouada, and B. Ottersten, “Flexible feedback system for posture monitoring and correction,” in *2017 Fourth International Conference on Image Information Processing (ICIIP)*, Dec. 2017, pp. 1–6. DOI: 10.1109/ICIIP.2017.8313687.

[4] R. Baptista., M. Antunes., D. Aouada., and B. Ottersten., “Video-based feedback for assisting physical activity,” in *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP, (VISIGRAPP 2017)*, INSTICC, SciTePress, 2017, pp. 274–280, ISBN: 978-989-758-226-4. DOI: 10.5220/0006132302740280.

[5] R. Baptista, E. Ghorbel, A. E. R. Shabayek, F. Moissenet, D. Aouada, A. Douchet, M. André, J. Pager, and S. Bouilland, “Home self-training: Visual feedback for assisting

physical activity for stroke survivors," *Computer methods and programs in biomedicine*, vol. 176, pp. 111–120, 2019.

[6] R. Baptista, E. Ghorbel, A. E. R. Shabayek, D. Aouada, and B. Ottersten, "Key-skeleton based feedback tool for assisting physical activity," in *2018 Zooming Innovation in Consumer Technologies Conference (ZINC)*, IEEE, 2018, pp. 175–176.

[7] R. Baptista, G. Demisse, D. Aouada, and B. Ottersten, "Deformation-based abnormal motion detection using 3d skeletons," in *2018 Eighth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, IEEE, 2018, pp. 1–6.

[8] Y. Song, D. Demirdjian, and R. Davis, "Continuous body and hand gesture recognition for natural human-computer interaction," *ACM Trans. Interact. Intell. Syst.*, vol. 2, no. 1, 5:1–5:28, Mar. 2012, ISSN: 2160-6455. DOI: 10.1145/2133366.2133371. [Online]. Available: http://doi.acm.org/10.1145/2133366.2133371.

[9] O. K. Oyedotun, D. Aouada, and B. Ottersten, "Learning to fuse latent representations for multimodal data," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 3122–3126.

[10] O. K. Oyedotun, G. Demisse, A. El Rahman Shabayek, D. Aouada, and B. Ottersten, "Facial expression recognition via joint deep learning of rgb-depth map latent representations," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 3161–3168.

[11] M. Devanne, "3d human behavior understanding by shape analysis of human motion and pose," PhD thesis, Dec. 2015.

[12] D. Fleet and Y. Weiss, "Optical flow estimation," in *Handbook of mathematical models in computer vision*, Springer, 2006, pp. 237–257.

[13] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Proceedings of the 13th Scandinavian Conference on Image Analysis*, ser. SCIA'03, Halmstad, Sweden: Springer-Verlag, 2003, pp. 363–370, ISBN: 3-540-40601-8. DOI: https://doi.org/10.1007/3-540-45103-X_50. [Online]. Available: http://dl.acm.org/citation.cfm?id=1763974.1764031.

[14] M. Jaimez, M. Souiai, J. Gonzalez-Jimenez, and D. Cremers, “A primal-dual framework for real-time dense rgb-d scene flow,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 98–104. DOI: 10.1109/ICRA.2015.7138986.

[15] Z. Lv, K. Kim, A. Troccoli, D. Sun, J. M. Rehg, and J. Kautz, “Learning rigidity in dynamic scenes with a moving camera for 3d motion field estimation,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 468–484.

[16] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu, “Action recognition by dense trajectories,” in *CVPR 2011*, IEEE, Jun. 2011. DOI: 10.1109/cvpr.2011.5995407. [Online]. Available: https://doi.org/10.1109/cvpr.2011.5995407.

[17] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, “Real-time human pose recognition in parts from single depth images,” *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[18] R. Vemulapalli, F. Arrate, and R. Chellappa, “Human Action Recognition by Representing 3D Human Skeletons as Points in a Lie Group,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[19] L. Xia, C. Chen, and J. K. Aggarwal, “View invariant human action recognition using histograms of 3d joints,” in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Jun. 2012, pp. 20–27. DOI: 10.1109/CVPRW.2012.6239233.

[20] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie, “Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks,” in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[21] Z. Huang, C. Wan, T. Probst, and L. Van Gool, “Deep learning on lie groups for skeleton-based action recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6099–6108.

[22] I. Lee, D. Kim, S. Kang, and S. Lee, “Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1012–1020.

[23] M. Antunes, D. Aouada, and B. Ottersten, “A revisit to human action recognition from depth sequences: Guided svm-sampling for joint selection,” in *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, IEEE, 2016, pp. 1–8.

[24] E. Ghorbel, R. Boutteau, J. Boonaert, X. Savatier, and S. Lecoeuche, “Kinematic spline curves: A temporal invariant descriptor for fast action recognition,” *Image and Vision Computing*, vol. 77, pp. 60–71, 2018.

[25] S. Yan, Y. Xiong, and D. Lin, “Spatial temporal graph convolutional networks for skeleton-based action recognition,” in *AAAI*, 2018.

[26] F. Garcia, D. Aouada, H. K. Abdella, T. Solignac, B. Mirbach, and B. Ottersten, “Depth enhancement by fusion for passive and active sensing,” in *European Conference on Computer Vision*, Springer, 2012, pp. 506–515.

[27] F. Garcia, D. Aouada, B. Mirbach, and B. Ottersten, “Spatio-temporal tof data enhancement by fusion,” in *2012 19th IEEE International Conference on Image Processing*, IEEE, 2012, pp. 981–984.

[28] K. Al Ismaeil, D. Aouada, T. Solignac, B. Mirbach, and B. Ottersten, “Real-time enhancement of dynamic depth videos with non-rigid deformations,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 10, pp. 2045–2059, Oct. 2017, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2016.2622698.

[29] ——, “Real-time non-rigid multi-frame depth video super-resolution,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2015, pp. 8–16. DOI: 10.1109/CVPRW.2015.7301389.

[30] M. Zanfir, M. Leordeanu, and C. Sminchisescu, “The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection,” in *2013 IEEE International Conference on Computer Vision*, Dec. 2013, pp. 2752–2759. DOI: 10.1109/ICCV.2013.342.

[31] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in *2006 IEEE Computer Society*

Conference on Computer Vision and Pattern Recognition (CVPR'06), vol. 2, Jun. 2006, pp. 2169–2178. DOI: 10.1109/CVPR.2006.68.

[32] K. Papadopoulos, M. Antunes, D. Aouada, and B. Ottersten, "Enhanced trajectory-based action recognition using human pose," in *2017 IEEE International Conference on Image Processing (ICIP)*, Sep. 2017, pp. 1807–1811. DOI: 10.1109/ICIP.2017.8296593.

[33] K. Papadopoulos, G. Demisse, E. Ghorbel, M. Antunes, D. Aouada, and B. Ottersten, "Localized trajectories for 2d and 3d action recognition," *Sensors*, vol. 19, no. 16, p. 3503, 2019.

[34] K. Papadopoulos, M. Antunes, D. Aouada, and B. Ottersten, "A revisit of action detection using improved trajectories," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*, IEEE, 2018.

[35] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt, "Vnect: Real-time 3d human pose estimation with a single rgb camera," 4, vol. 36, 2017. DOI: 10.1145/3072959.3073596. [Online]. Available: http://gvv.mpi-inf.mpg.de/projects/VNect/.

[36] E. Ghorbel, K. Papadopoulos, R. Baptista, H. Pathak, G. Demisse, D. Aouada, and B. Ottersten, "A view-invariant framework for fast skeleton-based action recognition using a single rgb camera," in *2019 International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, 2019.

[37] K. Papadopoulos, E. Ghorbel, O. Oyedotun, D. Aouada, and B. Ottersten, "Deepvi: A novel framework for learning deep view-invariant human action representations using a single rgb camera," in *IEEE International Conference on Automatic Face and Gesture Recognition, Buenos Aires 18-22 May 2020*, 2020.

[38] ——, "Learning deep view-invariant representations from synthetic viewpoints using a monocular rgb camera," in *IEEE Transactions on Image Processing (To be submitted)*, 2020.

[39] O. Oyedotun, D. Aouada, and B. Ottersten, “Structured compression of deep neural networks with debiased elastic group lasso,” in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 2277–2286.

[40] K. Papadopoulos, E. Ghorbel, D. Aouada, and B. Ottersten, “Vertex feature encoding and hierarchical temporal modeling in a spatial-temporal graph convolutional network for action recognition,” *IEEE International Conference on Pattern Recognition, Milan 13-18 September 2020 (Under review)*, 2020.

[41] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, “Performance of optical flow techniques,” *International journal of computer vision*, vol. 12, no. 1, pp. 43–77, 1994.

[42] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” 1981.

[43] B. K. Horn and B. G. Schunck, “Determining optical flow,” in *Techniques and Applications of Image Understanding*, International Society for Optics and Photonics, vol. 281, 1981, pp. 319–331.

[44] T. M. Kinect, `https://developer.microsoft.com/en-us/windows/kinect/`.

[45] R. Vemulapalli, F. Arrate, and R. Chellappa, “Human action recognition by representing 3d skeletons as points in a lie group,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2014. DOI: `10.1109/cvpr.2014.82`. [Online]. Available: `https://doi.org/10.1109/cvpr.2014.82`.

[46] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. Del Bimbo, “3-d human action recognition by shape analysis of motion trajectories on riemannian manifold,” *IEEE transactions on cybernetics*, vol. 45, no. 7, pp. 1340–1352, 2014.

[47] Y. Du, W. Wang, and L. Wang, “Hierarchical recurrent neural network for skeleton based action recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1110–1118.

[48] G. Othmezouri, I. Sakata, B. Schiele, M. Andriluka, and S. Roth, *Monocular 3d pose estimation and tracking by detection*, US Patent 8,958,600, Feb. 2015.

[49] X. K. Wei and J. Chai, “Modeling 3d human poses from uncalibrated monocular images,” in *2009 IEEE 12th International Conference on Computer Vision*, IEEE, 2009, pp. 1873–1880.

[50] A. Agarwal and B. Triggs, “Recovering 3d human pose from monocular images,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 1, pp. 44–58, 2005.

[51] X. Zhou, M. Zhu, S. Leonardos, K. G. Derpanis, and K. Daniilidis, “Sparseness meets deepness: 3d human pose estimation from monocular video,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4966–4975.

[52] H. Yasin, U. Iqbal, B. Kruger, A. Weber, and J. Gall, “A dual-source approach for 3d pose estimation from a single image,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4948–4956.

[53] A. Toshev and C. Szegedy, “Deeppose: Human pose estimation via deep neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1653–1660.

[54] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, “Joint training of a convolutional network and a graphical model for human pose estimation,” in *Advances in neural information processing systems*, 2014, pp. 1799–1807.

[55] D. Tome, C. Russell, and L. Agapito, “Lifting from the deep: Convolutional 3d pose estimation from a single image,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2500–2509.

[56] J. Martinez, R. Hossain, J. Romero, and J. J. Little, “A simple yet effective baseline for 3d human pose estimation,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2640–2649.

[57] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, “Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image,” in *Computer Vision – ECCV 2016*, ser. Lecture Notes in Computer Science, Springer International Publishing, Oct. 2016.

[58] B. Tekin, P. Márquez-Neila, M. Salzmann, and P. Fua, "Fusing 2d uncertainty and 3d cues for monocular body pose estimation," *CoRR*, vol. abs/1611.05708, 2016. arXiv: 1611.05708. [Online]. Available: http://arxiv.org/abs/1611.05708.

[59] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli, "3d human pose estimation in video with temporal convolutions and semi-supervised training," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[60] D. Mehta, H. Rhodin, D. Casas, O. Sotnychenko, W. Xu, and C. Theobalt, "Monocular 3d human pose estimation using transfer learning and improved CNN supervision," *CoRR*, vol. abs/1611.09813, 2016. arXiv: 1611.09813. [Online]. Available: http://arxiv.org/abs/1611.09813.

[61] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3d human pose," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, IEEE, 2017, pp. 1263–1272.

[62] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1325–1339, Jul. 2014. DOI: 10.1109/tpami.2013.248. [Online]. Available: https://doi.org/10.1109/tpami.2013.248.

[63] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2014. DOI: 10.1109/cvpr.2014.471. [Online]. Available: https://doi.org/10.1109/cvpr.2014.471.

[64] M. Koperski, P. Bilinski, and F. Bremond, "3d trajectories for action recognition," in *2014 IEEE International Conference on Image Processing (ICIP)*, Oct. 2014, pp. 4176–4180. DOI: 10.1109/ICIP.2014.7025848.

[65] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2012, pp. 1290–1297. DOI: 10.1109/CVPR.2012.6247813.

[66] H. Wang and C. Schmid, “Action recognition with improved trajectories,” in *2013 IEEE International Conference on Computer Vision*, Dec. 2013, pp. 3551–3558. DOI: 10.1109/ICCV.2013.441.

[67] L. Wang, Y. Qiao, and X. Tang, “Action recognition with trajectory-pooled deep-convolutional descriptors,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 4305–4314. DOI: 10.1109/CVPR.2015.7299059.

[68] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, and C.-W. Ngo, “Trajectory-based modeling of human actions with motion reference points,” in *Proceedings of the 12th European Conference on Computer Vision - Volume Part V*, ser. ECCV’12, Florence, Italy: Springer-Verlag, 2012, pp. 425–438, ISBN: 978-3-642-33714-7. DOI: 10.1007/978-3-642-33715-4_31. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-33715-4_31.

[69] B. Ni, P. Moulin, X. Yang, and S. Yan, “Motion part regularization: Improving action recognition via trajectory group selection,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 3698–3706. DOI: 10.1109/CVPR.2015.7298993.

[70] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, vol. 1, Jun. 2005, 886–893 vol. 1.

[71] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal, “Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 1932–1939. DOI: 10.1109/CVPR.2009.5206821.

[72] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black, “Towards understanding action recognition,” in *2013 IEEE International Conference on Computer Vision*, Dec. 2013, pp. 3192–3199. DOI: 10.1109/ICCV.2013.396.

[73] F. Zhu, L. Shao, J. Xie, and Y. Fang, “From handcrafted to learned representations for human action recognition,” *Image Vision Comput.*, vol. 55, no. P2, pp. 42–52, Nov.

2016, ISSN: 0262-8856. DOI: 10.1016/j.imavis.2016.06.007. [Online]. Available: https://doi.org/10.1016/j.imavis.2016.06.007.

[74] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A new representation of skeleton sequences for 3D action recognition," in *Conference on Computer Vision and Pattern Recognition, IEEE*, 2017, pp. 4570–4579.

[75] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, ser. AAAI'17, San Francisco, California, USA: AAAI Press, 2017, pp. 4263–4270. [Online]. Available: http://dl.acm.org/citation.cfm?id=3298023.3298186.

[76] Q. Ke, S. An, M. Bennamoun, F. Sohel, and F. Boussaid, "Skeletonnet: Mining deep part features for 3-d action recognition," *IEEE Signal Processing Letters*, vol. 24, no. 6, pp. 731–735, Jun. 2017, ISSN: 1070-9908. DOI: 10.1109/LSP.2017.2690339.

[77] O. Oreifej and Z. Liu, "Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2013, pp. 716–723. DOI: 10.1109/CVPR.2013.98.

[78] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, "Robust 3d action recognition with random occupancy patterns," in *Proceedings, Part II, of the 12th European Conference on Computer Vision — ECCV 2012 - Volume 7573*, Berlin, Heidelberg: Springer-Verlag, 2012, pp. 872–885, ISBN: 978-3-642-33708-6. DOI: 10.1007/978-3-642-33709-3_62. [Online]. Available: https://doi.org/10.1007/978-3-642-33709-3_62.

[79] L. Xia and J. K. Aggarwal, "Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2013, pp. 2834–2841. DOI: 10.1109/CVPR.2013.365.

[80] A. Klaeser, M. Marszalek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *Proceedings of the British Machine Vision Conference*, BMVA Press, 2008, pp. 99.1–99.10, ISBN: 1-901725-36-7. DOI: 10.5244/C.22.99.

[81] E. Ohn-Bar and M. M. Trivedi, “Joint angles similarities and hog2 for action recognition,” in *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Jun. 2013, pp. 465–470. DOI: 10.1109/CVPRW.2013.76.

[82] P. Foggia, G. Percannella, A. Saggese, and M. Vento, “Recognizing human actions by a bag of visual words,” in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, Oct. 2013, pp. 2910–2915. DOI: 10.1109/SMC.2013.496.

[83] P. Shukla, K. K. Biswas, and P. K. Kalra, “Action recognition using temporal bag-of-words from depth maps,” in *International Conference on Machine Vision Applications, IEEE*, 2013, pp. 41–44.

[84] X. Yang and Y. Tian, “Super normal vector for human activity recognition with depth cameras,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 5, pp. 1028–1039, May 2017, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2016.2565479.

[85] R. Slama, H. Wannous, and M. Daoudi, “Grassmannian representation of motion depth for 3d human gesture and action recognition,” in *2014 22nd International Conference on Pattern Recognition*, Aug. 2014, pp. 3499–3504. DOI: 10.1109/ICPR.2014.602.

[86] H. Rahmani, A. Mahmood, D. Huynh, and A. Mian, “Histogram of oriented principal components for cross-view action recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 12, pp. 2430–2443, 2016.

[87] X. Yang and Y. L. Tian, “Eigenjoints-based action recognition using naïve-bayes-nearest-neighbor,” in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Jun. 2012, pp. 14–19. DOI: 10.1109/CVPRW.2012.6239232.

[88] B. B. Amor, J. Su, and A. Srivastava, “Action recognition using rate-invariant analysis of skeletal shape trajectories,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 1–13, Jan. 2016, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2015.2439257.

[89] G. G. Demisse, K. Papadopoulos, D. Aouada, and B. Ottersten, “Pose encoding for robust skeleton-based action recognition,” in *2018 IEEE/CVF Conference on Computer*

Vision and Pattern Recognition Workshops (CVPRW), Jun. 2018, pp. 301–3016. DOI: 10.1109/CVPRW.2018.00056.

[90] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3d points," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, Jun. 2010, pp. 9–14. DOI: 10.1109/CVPRW.2010.5543273.

[91] M. Raptis, I. Kokkinos, and S. Soatto, "Discovering discriminative action parts from mid-level video representations," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2012, pp. 1242–1249. DOI: 10.1109/CVPR.2012.6247807.

[92] J. Quiroga, T. Brox, F. Devernay, and J. L. Crowley, "Dense Semi-Rigid Scene Flow Estimation from RGBD images," in *ECCV 2014 - European Conference on Computer Vision*, Zurich, Switzerland, Sep. 2014. DOI: 10.1007/978-3-319-10584-0_37. [Online]. Available: https://hal.inria.fr/hal-01021925.

[93] D. Sun, E. B. Sudderth, and H. Pfister, "Layered rgbd scene flow estimation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 548–556. DOI: 10.1109/CVPR.2015.7298653.

[94] M. B. Holte, B. Chakraborty, J. Gonzalez, and T. B. Moeslund, "A local 3-d motion descriptor for multi-view human action recognition from 4-d spatio-temporal interest points," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 5, pp. 553–565, Sep. 2012, ISSN: 1932-4553. DOI: 10.1109/JSTSP.2012.2193556.

[95] G. Yu, Z. Liu, and J. Yuan, "Discriminative orderlet mining for real-time recognition of human-object interaction," in *Computer Vision – ACCV 2014*, D. Cremers, I. Reid, H. Saito, and M.-H. Yang, Eds., Cham: Springer International Publishing, 2015, pp. 50–65, ISBN: 978-3-319-16814-2. DOI: 10.1007/978-3-319-16814-2_4.

[96] V. Bloom, V. Argyriou, and D. Makris, "Hierarchical transfer learning for online recognition of compound actions," *Comput. Vis. Image Underst.*, vol. 144, no. C, pp. 62–72, Mar. 2016, ISSN: 1077-3142. DOI: 10.1016/j.cviu.2015.12.001. [Online]. Available: https://doi.org/10.1016/j.cviu.2015.12.001.

[97] C. Wu, J. Zhang, O. Sener, B. Selman, S. Savarese, and A. Saxena, "Watch-n-patch: Unsupervised learning of actions and relations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 2, pp. 467–481, Feb. 2018, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2017.2679054.

[98] S. Gaglio, G. L. Re, and M. Morana, "Human activity recognition process using 3-d posture data," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 5, pp. 586–597, Oct. 2015, ISSN: 2168-2291. DOI: 10.1109/THMS.2014.2377111.

[99] M. Müller and T. Röder, "Motion templates for automatic classification and retrieval of motion capture data," in *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '06, Vienna, Austria: Eurographics Association, 2006, pp. 137–146, ISBN: 3-905673-34-7. [Online]. Available: http://dl.acm.org/citation.cfm?id=1218064.1218083.

[100] W. Chen and G. Guo, "Triviews: A general framework to use 3d depth data effectively for action recognition," *Journal of Visual Communication and Image Representation*, vol. 26, pp. 182–191, 2015, ISSN: 1047-3203. DOI: https://doi.org/10.1016/j.jvcir.2014.11.008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1047320314001898.

[101] Z. Luo, B. Peng, D. Huang, A. Alahi, and L. Fei-Fei, "Unsupervised learning of long-term motion dynamics for videos," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 7101–7110. DOI: 10.1109/CVPR.2017.751.

[102] A. Jalal, Y.-H. Kim, Y.-J. Kim, S. Kamal, and D. Kim, "Robust human activity recognition from depth video using spatiotemporal multi-fused features," *Pattern Recogn.*, vol. 61, no. C, pp. 295–308, Jan. 2017, ISSN: 0031-3203. DOI: 10.1016/j.patcog.2016.08.003. [Online]. Available: https://doi.org/10.1016/j.patcog.2016.08.003.

[103] D. Leightley, B. Li, J. S. McPhee, M. H. Yap, and J. Darby, "Exemplar-based human action recognition with template matching from a stream of motion capture," in *Image Analysis and Recognition*, A. Campilho and M. Kamel, Eds., Cham: Springer International

Publishing, 2014, pp. 12–20, ISBN: 978-3-319-11755-3. DOI: 10.1007/978-3-319-11755-3_2.

[104] Q. Xiao, Y. Wang, and H. Wang, "Motion retrieval using weighted graph matching," *Soft Comput.*, vol. 19, no. 1, pp. 133–144, Jan. 2015, ISSN: 1432-7643. DOI: 10.1007/s00500-014-1237-5. [Online]. Available: http://dx.doi.org/10.1007/s00500-014-1237-5.

[105] M. Li, H. Leung, Z. Liu, and L. Zhou, "3d human motion retrieval using graph kernels based on adaptive graph construction," *Computers & Graphics*, vol. 54, pp. 104–112, 2016, Special Issue on CAD/Graphics 2015, ISSN: 0097-8493. DOI: https://doi.org/10.1016/j.cag.2015.07.005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0097849315001089.

[106] M. Barnachon, S. Bouakaz, B. Boufama, and E. Guillou, "Ongoing human action recognition with motion capture," *Pattern Recognition*, vol. 47, no. 1, pp. 238–247, 2014, ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2013.06.020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320313002720.

[107] "Activity-based methods for person recognition in motion capture sequences," *Pattern Recognition Letters*, vol. 49, pp. 48–54, 2014, ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2014.06.005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865514001822.

[108] P. Kishore, P. S. Kameswari, K. Niharika, M. Tanuja, M. Bindu, D. A. Kumar, E. K. Kumar, and M. T. Kiran, "Spatial joint features for 3d human skeletal action recognition system using spatial graph kernels," *International Journal of Engineering & Technology*, vol. 7, no. 1.1, pp. 489–493, 2018. DOI: 10.14419/ijet.v7i1.1.10152.

[109] R. Vemulapalli and R. Chellappa, "Rolling rotations for recognizing human actions from 3d skeletal data," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 4471–4479. DOI: 10.1109/CVPR.2016.484.

[110] M. Devanne, S. Berretti, P. Pala, H. Wannous, M. Daoudi, and A. Del Bimbo, "Motion segment decomposition of RGB-D sequences for human behavior understanding," *Pattern Recognition*, vol. 61, pp. 222–233, 2017.

[111] F. Ahmed, P. P. Paul, and M. L. Gavrilova, "Joint-triplet motion image and local binary pattern for 3d action recognition using kinect," in *Proceedings of the 29th International Conference on Computer Animation and Social Agents*, ser. CASA '16, Geneva, Switzerland: ACM, 2016, pp. 111–119, ISBN: 978-1-4503-4745-7. DOI: 10.1145/2915926.2915937. [Online]. Available: http://doi.acm.org/10.1145/2915926.2915937.

[112] Ying-li Tian, T. Kanade, and J. F. Cohn, "Evaluation of gabor-wavelet-based facial action unit recognition in image sequences of increasing complexity," in *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*, May 2002, pp. 229–234. DOI: 10.1109/AFGR.2002.1004159.

[113] R. D. Geest, E. Gavves, A. Ghodrati, Z. Li, C. Snoek, and T. Tuytelaars, "Online action detection," *CoRR*, vol. abs/1604.06506, 2016. arXiv: 1604.06506. [Online]. Available: http://arxiv.org/abs/1604.06506.

[114] D. Huang, S. Yao, Y. Wang, and F. De La Torre, "Sequential max-margin event detectors," in *European Conference on Computer Vision (ECCV)*, 2014.

[115] M. Hoai and F. Torre, "Max-margin early event detectors," in *International Journal of Computer Vision (IJCV)*, 2014.

[116] A. Gaidon, Z. Harchaoui, and C. Schmid, "Actom Sequence Models for Efficient Action Detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2011.

[117] A. Kläser, M. Marszalek, C. Schmid, and A. Zisserman, "Human focused action localization in video," in *European Conference on Computer Vision (ECCV)*, 2012.

[118] B. Schiele, "A database for fine grained activity detection of cooking activities," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[119] C. Sun, S. Shetty, R. Sukthankar, and R. Nevatia, "Temporal localization of fine-grained actions in videos by domain transfer from web images," in *ACM Multimedia Conference (MM)*, 2015.

[120] Z. Shu, K. Yun, and D. Samaras, “Action detection with improved dense trajectories and sliding window,” in *European Conference on Computer Vision Workshop (ECCVW)*, 2015.

[121] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (surf),” *Comput. Vis. Image Underst.*, Jun. 2008.

[122] C. Schmid, B. Rozenfeld, M. Marszalek, and I. Laptev, “Learning realistic human actions from movies,” *IEEE Conference on Computer Vision & Pattern Recognition*, pp. 1–8, 2008.

[123] F. Perronnin, J. Sánchez, and T. Mensink, “Improving the fisher kernel for large-scale image classification,” in *European Conference on Computer Vision (ECCV)*, 2010.

[124] M. A. Gowayyed, M. Torki, M. E. Hussein, and M. El-Saban, “Histogram of oriented displacements (hod): Describing trajectories of human joints for action recognition,” in *International Joint Conferences on Artificial Intelligence*, 2013.

[125] A. Newell, K. Yang, and J. Deng, “Stacked hourglass networks for human pose estimation,” in *European Conference on Computer Vision*, Springer, 2016, pp. 483–499.

[126] Y. Li, C. Lan, J. Xing, W. Zeng, C. Yuan, and J. Liu, “Online human action detection using joint classification-regression recurrent neural networks,” in *European Conference on Computer Vision (ECCV)*, 2016.

[127] J. Aggarwal and L. Xia, “Human activity recognition from 3d data: A review,” *Pattern Recognition Letters*, vol. 48, pp. 70–80, Oct. 2014. DOI: 10.1016/j.patrec.2014.04.011. [Online]. Available: https://doi.org/10.1016/j.patrec.2014.04.011.

[128] R. Poppe, “A survey on vision-based human action recognition,” *Image and vision computing*, vol. 28, no. 6, pp. 976–990, 2010.

[129] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars, “Modeling video evolution for action recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5378–5387.

[130] A. Haque, B. Peng, Z. Luo, A. Alahi, S. Yeung, and L. Fei-Fei, “Towards viewpoint invariant 3d human pose estimation,” in *European Conference on Computer Vision*, Springer, 2016, pp. 160–177.

[131] Y.-P. Hsu, C. Liu, T.-Y. Chen, and L.-C. Fu, “Online view-invariant human action recognition using rgb-d spatio-temporal matrix,” *Pattern recognition*, vol. 60, pp. 215–226, 2016.

[132] A. Gupta, J. Martinez, J. J. Little, and R. J. Woodham, “3d pose from motion for cross-view action recognition via non-linear circulant temporal encoding,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2014. DOI: 10.1109/cvpr.2014.333. [Online]. Available: https://doi.org/10.1109/cvpr.2014.333.

[133] H. Rahmani and A. Mian, “Learning a non-linear knowledge transfer model for cross-view action recognition,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2015. DOI: 10.1109/cvpr.2015.7298860. [Online]. Available: https://doi.org/10.1109/cvpr.2015.7298860.

[134] W. Yang, W. Ouyang, X. Wang, J. Ren, H. Li, and X. Wang, “3d human pose estimation in the wild by adversarial learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5255–5264.

[135] E. Ghorbel, R. Boutteau, J. Bonnaert, X. Savatier, and S. Lecoeuche, “A fast and accurate motion descriptor for human action recognition applications,” in *International Conference on Pattern Recognition, IEEE*, 2016, pp. 919–924.

[136] J. Wang, X. Nie, Y. Xia, Y. Wu, and S.-C. Zhu, “Cross-view action modeling, learning and recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2649–2656.

[137] D. Weinland, R. Ronfard, and E. Boyer, “Free viewpoint action recognition using motion history volumes,” *Computer Vision and Image Understanding*, vol. 104, no. 2-3, pp. 249–257, Nov. 2006. DOI: 10.1016/j.cviu.2006.07.013. [Online]. Available: https://doi.org/10.1016/j.cviu.2006.07.013.

[138] N. P. Trong, A. T. Minh, H. Nguyen, K. Kazunori, and B. L. Hoai, "A survey about view-invariant human action recognition," in *2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, IEEE, Sep. 2017. DOI: 10.23919/sice.2017.8105762. [Online]. Available: https://doi.org/10.23919/sice.2017.8105762.

[139] H. Rahmani, A. Mahmood, D. Q. Huynh, and A. Mian, "Hopc: Histogram of oriented principal components of 3d pointclouds for action recognition," in *European conference on computer vision*, Springer, 2014, pp. 742–757.

[140] G. Evangelidis, G. Singh, and R. Horaud, "Skeletal quads: Human action recognition using joint quadruples," in *International Conference on Pattern Recognition, IEEE*, 2014, pp. 4513–4518.

[141] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *Proceedings of the 9th European Conference on Computer Vision - Volume Part II*, ser. ECCV'06, Graz, Austria: Springer-Verlag, 2006, pp. 428–441, ISBN: 3-540-33834-9, 978-3-540-33834-5. DOI: 10.1007/11744047_33. [Online]. Available: http://dx.doi.org/10.1007/11744047_33.

[142] L. L. Presti and M. L. Cascia, "3d skeleton-based human action classification: A survey," *Pattern Recognition*, vol. 53, pp. 130–147, May 2016. DOI: 10.1016/j.patcog.2015.11.019. [Online]. Available: https://doi.org/10.1016/j.patcog.2015.11.019.

[143] B. Li, O. I. Camps, and M. Sznaier, "Cross-view activity recognition using hankelets," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2012. DOI: 10.1109/cvpr.2012.6247822. [Online]. Available: https://doi.org/10.1109/cvpr.2012.6247822.

[144] Z. Zhang, C. Wang, B. Xiao, W. Zhou, S. Liu, and C. Shi, "Cross-view action recognition via a continuous virtual path," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2013. DOI: 10.1109/cvpr.2013.347. [Online]. Available: https://doi.org/10.1109/cvpr.2013.347.

[145] R. Li and T. Zickler, “Discriminative virtual views for cross-view action recognition,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2012. DOI: 10.1109/cvpr.2012.6248011. [Online]. Available: https://doi.org/10.1109/cvpr.2012.6248011.

[146] F. Lv and R. Nevatia, “Single view human action recognition using key pose matching and viterbi path searching,” in *Computer Vision and Pattern Recognition, 2007. CVPR’07. IEEE Conference on*, IEEE, 2007, pp. 1–8.

[147] D. Weinland, E. Boyer, and R. Ronfard, “Action recognition from arbitrary views using 3d exemplars,” in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, IEEE, 2007, pp. 1–7.

[148] V. Parameswaran and R. Chellappa, “View invariance for human action recognition,” *International Journal of Computer Vision*, vol. 66, no. 1, pp. 83–101, 2006.

[149] C. Rao, A. Yilmaz, and M. Shah, “View-invariant representation and recognition of actions,” *International Journal of Computer Vision*, vol. 50, no. 2, pp. 203–226, 2002.

[150] T. Hao, D. Wu, Q. Wang, and J.-S. Sun, “Multi-view representation learning for multi-view action recognition,” *Journal of Visual Communication and Image Representation*, vol. 48, pp. 453–460, 2017.

[151] Y. Kong, Z. Ding, J. Li, and Y. Fu, “Deeply learned view-invariant features for cross-view action recognition,” *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 3028–3037, 2017.

[152] H. Rahmani, A. Mian, and M. Shah, “Learning a deep model for human action recognition from novel viewpoints,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 667–681, 2018.

[153] R. Baptista, E. Ghorbel, K. Papadopoulos, G. Demisse, D. Aouada, and B. Ottersten, “View-invariant action recognition from rgb data via 3d pose estimation,” in *ICASSP*, IEEE, 2019.

[154] G. Rogez, P. Weinzaepfel, and C. Schmid, “Lcr-net++: Multi-person 2d and 3d pose detection in natural images,” *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[155] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, “Temporal convolutional networks for action segmentation and detection,” in *CVPR*, 2017, pp. 156–165.

[156] I. N. Junejo, E. Dexter, I. Laptev, and P. Pérez, “Cross-view action recognition from temporal self-similarities,” in *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2008, pp. 293–306. DOI: 10.1007/978-3-540-88688-4_22. [Online]. Available: https://doi.org/10.1007/978-3-540-88688-4_22.

[157] A. Farhadi and M. K. Tabrizi, “Learning to recognize activities from the wrong view point,” in *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2008, pp. 154–166. DOI: 10.1007/978-3-540-88682-2_13. [Online]. Available: https://doi.org/10.1007/978-3-540-88682-2_13.

[158] A. Ulhaq, X. Yin, J. He, and Y. Zhang, “On space-time filtering framework for matching human actions across different viewpoints,” *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1230–1242, Mar. 2018. DOI: 10.1109/TIP.2017.2765821.

[159] A. Ulhaq, “Deep cross-view convolutional features for view-invariant action recognition,” in *2018 IEEE International Conference on Image Processing, Applications and Systems (IPAS)*, Dec. 2018, pp. 137–142. DOI: 10.1109/IPAS.2018.8708853.

[160] J. Liu, M. Shah, B. Kuipers, and S. Savarese, “Cross-view action recognition via view knowledge transfer,” 2011.

[161] D. Wang, W. Ouyang, W. Li, and D. Xu, “Dividing and aggregating network for multi-view action recognition,” in *ECCV*, 2018, pp. 451–467.

[162] E. Ghorbel, J. Boonaert, R. Boutteau, S. Lecoeuche, and X. Savatier, “An extension of kernel learning methods using a modified log-euclidean distance for fast and accurate skeleton-based human action recognition,” *Computer Vision and Image Understanding*, vol. 175, pp. 32–43, 2018.

[163] M. Liu, H. Liu, and C. Chen, "Enhanced skeleton visualization for view invariant human action recognition," *Pattern Recognition*, vol. 68, pp. 346–362, Aug. 2017. DOI: 10.1016/j.patcog.2017.02.030. [Online]. Available: https://doi.org/10.1016/j.patcog.2017.02.030.

[164] D. R. K. Brownrigg, "The weighted median filter," *Commun. ACM*, vol. 27, no. 8, pp. 807–818, Aug. 1984, ISSN: 0001-0782. DOI: 10.1145/358198.358222. [Online]. Available: http://doi.acm.org/10.1145/358198.358222.

[165] S. Zhang, H. Jiang, S. Wei, and L.-R. Dai, "Rectified linear neural networks with tied-scalar regularization for lvcsr," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[166] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[167] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+D: A large scale dataset for 3D human activity analysis," in *Conference on Computer Vision and Pattern Recognition, IEEE*, 2016, pp. 1010–1019.

[168] F. Baradel, C. Wolf, and J. Mille, "Human action recognition: Pose-based attention draws focus to hands," in *ICCV*, 2017, pp. 604–613.

[169] S. Das, A. Chaudhary, F. Bremond, and M. Thonnat, "Where to focus on for human action recognition?" In *WACV*, Jan. 2019, pp. 71–80. DOI: 10.1109/WACV.2019.00015.

[170] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, "View adaptive neural networks for high performance skeleton-based human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1963–1978, 2019.

[171] Q. Nie, J. Wang, X. Wang, and Y. Liu, "View-invariant human action recognition based on a 3d bio-constrained skeleton model," *IEEE Transactions on Image Processing*, vol. 28, no. 8, pp. 3959–3972, Aug. 2019, ISSN: 1941-0042. DOI: 10.1109/TIP.2019.2907048.

[172] M. Liu and J. Yuan, "Recognizing human actions as the evolution of pose estimation maps," in *CVPR*, 2018, pp. 1159–1168.

[173] F. Baradel, C. Wolf, and J. Mille, “Human Activity Recognition with Pose-driven Attention to RGB,” in *BMVC*, Newcastle, United Kingdom, Sep. 2018, pp. 1–14. [Online]. Available: https://hal.inria.fr/hal-01828083.

[174] N. C. Garcia, P. Morerio, and V. Murino, “Modality distillation with multiple stream networks for action recognition,” in *ECCV*, 2018, pp. 103–118.

[175] J. Zheng and Z. Jiang, “Learning view-invariant sparse representations for cross-view action recognition,” in *ICCV*, 2013, pp. 3176–3183.

[176] A. Zanfir, E. Marinoiu, and C. Sminchisescu, “Monocular 3d pose and shape estimation of multiple people in natural scenes-the importance of multiple scene constraints,” in *CVPR*, 2018, pp. 2148–2157.

[177] D. Wu and L. Shao, “Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition,” in *2014 CVPR*, Jun. 2014, pp. 724–731. DOI: 10.1109/CVPR.2014.98.

[178] K. Papadopoulos, E. Ghorbel, R. Baptista, D. Aouada, and B. Ottersten, “Two-stage rgb-based action detection using augmented 3d poses,” in *International Conference on Computer Analysis of Images and Patterns*, Springer, 2019, pp. 26–35.

[179] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, “Spectral networks and locally connected networks on graphs,” *arXiv preprint arXiv:1312.6203*, 2013.

[180] L. Shi, Y. Zhang, J. Cheng, and H. Lu, “Two-stream adaptive graph convolutional networks for skeleton-based action recognition,” in *CVPR*, 2019, pp. 12 026–12 035.

[181] ——, “Skeleton-based action recognition with directed graph neural networks,” in *CVPR*, 2019, pp. 7912–7921.

[182] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, “Actional-structural graph convolutional networks for skeleton-based action recognition,” in *CVPR*, 2019, pp. 3595–3603.

[183] J. C. Meza and M. Woods, “A numerical comparison of rule ensemble methods and support vector machines,” Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States), Tech. Rep., 2009.

[184] A. B. Tanfous, H. Drira, and B. B. Amor, “Coding kendall’s shape trajectories for 3d action recognition,” in *IEEE Computer Vision and Pattern Recognition*, 2018.

[185] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, “A new representation of skeleton sequences for 3d action recognition,” in *CVPR*, 2017, pp. 3288–3297.

[186] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, “View adaptive recurrent neural networks for high performance human action recognition from skeleton data,” in *2017 ICCV*, Oct. 2017, pp. 2136–2145. DOI: 10.1109/ICCV.2017.233.

[187] C. Li, Q. Zhong, D. Xie, and S. Pu, “Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation,” in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, AAAI Press, 2018, pp. 786–792.

[188] ——, “Skeleton-based action recognition with convolutional neural networks,” in *2017 ICME Workshops (ICMEW)*, IEEE, 2017, pp. 597–600.

[189] Y. Du, Y. Fu, and L. Wang, “Skeleton based action recognition with convolutional neural network,” in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, IEEE, 2015, pp. 579–583.

[190] D. Xie, C. Deng, H. Wang, C. Li, and D. Tao, “Semantic adversarial network with multi-scale pyramid attention for video classification,” in *AAAI*, vol. 33, 2019, pp. 9030–9037.

[191] M. Adel Musallam, R. Baptista, K. Al Ismaeil, and D. Aouada, “Temporal 3d human pose estimation for action recognition from arbitrary viewpoints,” in *6th Annual Conf. on Computational Science & Computational Intelligence, Las Vegas 5-7 December 2019*, Conference Publishing Services, 2019.

[192] C. Caetano, J. Sena, F. Brémond, J. A. d. Santos, and W. R. Schwartz, “Skelemotion: A new representation of skeleton joint sequences based on motion information for 3d action recognition,” *arXiv preprint arXiv:1907.13025*, 2019.

[193] D. Liang, G. Fan, G. Lin, W. Chen, X. Pan, and H. Zhu, “Three-stream convolutional neural network with multi-task and ensemble learning for 3d action recognition,” in *CVPR Workshops*, 2019, pp. 0–0.

[194] G. Hu, B. Cui, and S. Yu, “Skeleton-based action recognition with synchronous local and non-local spatio-temporal learning and frequency attention,” in *2019 ICME*, IEEE, 2019, pp. 1216–1221.

[195] M. Henaff, J. Bruna, and Y. LeCun, “Deep convolutional networks on graph-structured data,” *arXiv preprint arXiv:1506.05163*, 2015.

[196] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, “Convolutional networks on graphs for learning molecular fingerprints,” in *Advances in neural information processing systems*, 2015, pp. 2224–2232.

[197] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, “Gated graph sequence neural networks,” *arXiv preprint arXiv:1511.05493*, 2015.

[198] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in *Advances in neural information processing systems*, 2016, pp. 3844–3852.

[199] Y. Tang, Y. Tian, J. Lu, P. Li, and J. Zhou, “Deep progressive reinforcement learning for skeleton-based action recognition,” in *2018 CVPR*, Jun. 2018, pp. 5323–5332. DOI: 10.1109/CVPR.2018.00558.

[200] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan, “An attention enhanced graph convolutional lstm network for skeleton-based action recognition,” in *CVPR*, 2019, pp. 1227–1236.

[201] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, “Wavenet: A generative model for raw audio,” *arXiv preprint arXiv:1609.03499*, 2016.

[202] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” *arXiv preprint arXiv:1511.07122*, 2015.

[203] N. Kalchbrenner, L. Espeholt, K. Simonyan, A. v. d. Oord, A. Graves, and K. Kavukcuoglu, "Neural machine translation in linear time," *arXiv preprint arXiv:1610.10099*, 2016.

[204] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot, "Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. DOI: 10.1109/TPAMI.2019.2916873.

[205] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "Learning clip representations for skeleton-based 3d action recognition," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 2842–2855, 2018.

[206] J. Liu, G. Wang, L.-Y. Duan, K. Abdiyeva, and A. C. Kot, "Skeleton-based human action recognition with global context-aware attention lstm networks," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 1586–1599, 2017.