2017

Action recognition from RGB-D data

Pichao Wang
University of Wollongong

UNIVERSITY
OF WOLLONGONG
AUSTRALIA

Action Recognition from RGB-D Data

Pichao Wang

Supervisor:
A/Prof. Wanqing Li
Co-supervisor:
Prof. Philip O. Ogunbona

This thesis is presented as required for the conferral of the degree:

Doctor of Philosophy

The University of Wollongong
School of Computing and Information Technology

October 25, 2017

Declaration

I, Pichao Wang, declare that this thesis submitted in fulfilment of the requirements for the conferral of the degree Doctor of Philosophy, from the University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. This document has not been submitted for qualifications at any other academic institution.

Pichao Wang
October 25, 2017

Abstract

In recent years, action recognition based on RGB-D data has attracted increasing attention. Different from traditional 2D action recognition, RGB-D data contains extra depth and skeleton modalities. Different modalities have their own characteristics. This thesis presents seven novel methods to take advantages of the three modalities for action recognition.

First, effective handcrafted features are designed and frequent pattern mining method is employed to mine the most discriminative, representative and non-redundant features for skeleton-based action recognition. *Second*, to take advantages of powerful Convolutional Neural Networks (ConvNets), it is proposed to represent spatio-temporal information carried in 3D skeleton sequences in three 2D images by encoding the joint trajectories and their dynamics into color distribution in the images, and ConvNets are adopted to learn the discriminative features for human action recognition. *Third*, for depth-based action recognition, three strategies of data augmentation are proposed to apply ConvNets to small training datasets. *Forth*, to take full advantage of the 3D structural information offered in the depth modality and its being insensitive to illumination variations, three simple, compact yet effective images-based representations are proposed and ConvNets are adopted for feature extraction and classification. However, both of previous two methods are sensitive to noise and could not differentiate well fine-grained actions. *Fifth*, it is proposed to represent a depth map sequence into three pairs of structured dynamic images at body, part and joint levels respectively through bidirectional rank pooling to deal with the issue. The structured dynamic image preserves the spatial-temporal information, enhances the structure information across both body parts/joints and different temporal scales, and takes advantages of ConvNets for action recognition. *Sixth*, it is proposed to extract and use scene flow for action recognition from RGB and depth data. *Last*, to exploit the joint information in multi-modal features arising from heterogeneous sources (RGB, depth), it is proposed to cooperatively train a single ConvNet (referred to as c-ConvNet) on both RGB features and depth features, and deeply aggregate the two modalities to achieve robust action recognition.

Acknowledgments

Many people have contributed in various ways to make my PhD study an exciting and memorable journey. Just to name a few:

It is a privilege to be a research student in the Advanced Multimedia Research Lab, School of Computing Information Technology, University of Wollongong. I also take this opportunity to thank the China Scholarship Council and University of Wollongong for their financial support to my study.

I would like to express my utmost gratitude to my supervisors: A/Prof. Wanqing Li and Prof. Philip Ogunbona, for all their invaluable help and advice during my studies. My principle supervisor, A/Prof. Wanqing Li, gave me freedom to explore my research interest and provide guidance to my research and life. I would never forget my co-supervisor, Prof. Philip Ogunbona, who taught me to write papers words by words, and explained every details in writing problems. I recognize my supervisors not only as my teachers but also my good friends. It would be impossible to be a qualified PhD without their enormous help.

I would like to thank Dr. Chang Tang, Dr. Jun Wan, Dr. Sergio Escalera, Jing Zhang, Song Liu, Yuyao Zhang, Zewei Ding, Chuankun Li, Zhaoyang Li, Shuang Wang, Liwei Wang and Huogen Wang, for all the cooperations with me. To be honest, I benefit a lot from their feedback when we were discussing either on my research works or on existing research papers. They also expanded my knowledge and broaden my views.

I would like to thank all my colleagues/friends in Australia, a non-exhaustive list of whom includes: Dr. Xingwang Liu, Dr. Hongda Tian, Dr. Jianjia Zhang, Dr. Rongmao Chen, Xiaoyu Yu, Lijuan Zhou, Shuai Li, Yanbo Gao, Wenfang Chen, Yan Zhao, Biting Yu, Zhongyan Zhang, Yangguang Tian and Jianchang Lai, for their company during my PhD study and living in Wollongong. Research and life in Australia could not have been as colorful as it has been without their company.

I would also thank Mr. Jun Hu and Mrs. Yuan Tian from IT support. They really helped me a lot on computer problems and saved me lots of valuable time. They are kind and patient, and every time when I have problem with my computer, it only needs to email them.

A huge thank to my Master degree supervisor A/Prof. Yonghong Hou and his wife Dr. Xinying Wang. Without their encouragement and financial support I do not think it is possible for me to pursue my PhD degree in Australia. They are my best friends and beloved family, and their support has been tremendously valuable

and appreciated indeed.

Many thanks go to my parents. Even though poor in money, I heritage the sprite of hard working and honest from them, and enjoy the love from them. Specific thanks to my mother, who has been looking after my father after his hemiplegia more than five years ago. It is my mother who also encouraged me to continue my PhD study without too much worries. Furthermore, I thank my brother, my sister and my lovely nephews and nieces, for their love and support. I am so lucky to have all of you in my family.

The list would be incomplete without expressing the most wholehearted gratitude to my life angel and love, Zhimin Gao. As a PhD candidate herself, she has been taking care of me for more than four years, especially when I was badly sick for one year and a half during the study. It is she who looked after me, encouraged me and helped me both in living and research. She has given up many things for me to finish my study; she has cherished with me every great moment and supported me whenever I needed it. Her unwavering love, patient endurance and tolerance of my occasional vulgar mood throughout the last four years deserve so much than just a "thank you". Not only now, but also in the years to come.

Publications

Published/Accepted Papers

1. Pichao Wang, Wanqing Li, Philip Ogunbona, Zhimin Gao and Hanling Zhang. Mining Mid-level Features for Action Recognition Based on Effective Skeleton Representation. Published in International Conference on Digital Image Computing: Techniques and Applications (DICTA), 2014, 1-8.

2. Pichao Wang, Wanqing Li, Zhimin Gao, Chang Tang, Jing Zhang and Philip Ogunbona. ConvNets-Based Action Recognition from Depth Maps Through Virtual Cameras and Pseudocoloring. Published in ACM Multimedia (ACM MM), 2015, 1119-1122.

3. Pichao Wang, Wanqing Li, Zhimin Gao, Jing Zhang, Chang Tang and Philip Ogunbona. Action Recognition from Depth Maps Using Deep Convolutional Neural Networks, Published in IEEE Transactions on Human Machine Systems, Vol. 46, No. 4, pp. 498-509, August, 2016

4. Pichao Wang, Zhaoyang Li, Yonghong Hou and Wanqing Li. Action Recognition Based on Joint Trajectory Maps Using Convolutional Neural Networks. Published in ACM Multimedia (ACM MM), 2016, 102-106.

5. Pichao Wang, Wanqing Li, Song Liu, Zhimin Gao, Chang Tang and Philip Ogunbona. Large-scale Isolated Gesture Recognition Using Convolutional Neural Networks. Published in International Conference on Pattern Recognition (ICPR), 2016, 7-12.

6. Pichao Wang, Wanqing Li, Song Liu, Yuyao Zhang, Zhimin Gao and Philip Ogunbona. Large-scale Continuous Gesture Recognition Using Convolutional Neural Networks. Published in International Conference on Pattern Recognition (ICPR), 2016, 13-18.

7. Pichao Wang, Wanqing Li, Zhimin Gao, Yuyao Zhang, Chang Tang and Philip Ogunbona. Scene Flow to Action Map: A New Representation for RGB-D Based Action Recognition with Convolutional Neural Networks. Accepted in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

8. Pichao Wang, Shuang Wang, Zhimin Gao, Yonghong Hou, and Wanqing Li. Structured Images for RGB-D Action Recognition. Accepted in International Conference on Computer Vision Workshops (ICCVW), 2017.

9. Chuankun Li, Yonghong Hou, Pichao Wang* and Wanqing Li. Joint Distance Maps for Human Action Recognition with Convolutional Neural Networks. Published in IEEE Signal Processing Letters, Vol. 24, No. 5, pp. 624-628, May, 2017. (*Corresponding author)

10. Yonghong Hou, Zhaoyang Li, Pichao Wang* and Wanqing Li. Skeleton Optical Spectra Based Action Recognition Using Convolutional Neural Networks. Accepted in IEEE Transactions on Circuits and Systems for Video Technology, in press. (*Corresponding author)

11. Chuankun Li*, Pichao Wang*, Shuang Wang, Yonghong Hou and Wanqing Li. Skeleton-based Action Recognition Using LSTM and CNN. Accepted in Large Scale 3D Human Activity Analysis Challenge in Depth Videos, ICMEW2017. (rank the first place) (* equal contribution)

12. Huogen Wang*, Pichao Wang*, Zhanjie Song, and Wanqing Li. Large-scale Multimodal Gesture Recognition Using Heterogeneous Networks. Accepted in ICCVW, 2017. (* equal contribution)

13. Huogen Wang*, Pichao Wang*, Zhanjie Song, and Wanqing Li. Large-scale Multimodal Gesture Segmentation and Recognition based on Convolutional Neural Network. Accepted in ICCVW, 2017. (* equal contribution)

14. Zewei Ding, Pichao Wang*, Philip Ogunbona and Wanqing Li. Investigation of Different Skeleton Features for CNN-based 3D Action Recognition. Accepted in Large Scale 3D Human Activity Analysis Challenge in Depth Videos, ICMEW2017. (*Corresponding author)

15. Jing Zhang, Wanqing Li, Philip Ogunbona, Pichao Wang and Chang Tang. RGB-D based Action Recognition Datasets: A Survey. Published in Pattern Recognition, Vol. 60, pp.86-105, Dec. 2016.

16. Jing Zhang, Wanqing Li, Pichao Wang, Philip Ogunbona, Song Liu and Chang Tang. A Large Scale RGB-D Dataset for Action Recognition. Accepted in ICPR workshop on UHA3DS, 2016.

17. Zewei Ding, Wanqing Li, Pichao Wang, Philip Ogunbona and Ling Qin. Weakly Structured Information Aggregation for Upper-body Posture Assessment Using ConvNets. Accepted in IEEE International Conference on Multimedia & Expo (ICME), 2017.

Submitted Papers

1. Pichao Wang, Wanqing Li, Chuankun Li and Yonghong Hou. Action Recognition Based on Joint Trajectory Maps with Convolutional Neural Networks. Submitted to Knowledge-Based Systems, under view.

2. Pichao Wang, Wanqing Li, Zhimin Gao, Chang Tang, and Philip Ogunbona. Depth Pooling Based Large-scale 3D Action Recognition with Convolutional Neural Networks. Submitted to IEEE Transactions on Multimedia, under review.

3. Pichao Wang, Wanqing Li, Philip Ogunbona, Jun Wan and Sergio Escalera. RGB-D-based Motion Recognition with Deep Learning: A Survey. Submitted to Computer Vision and Image Understanding, under review.

4. Pichao Wang, Wanqing Li, Jun Wan, Philp Ogunbona and Xinwang Liu. Cooperative Training of Deep Aggregation Networks for RGB-D Action Recognition. Submitted to AAAI Conference on Artificial Intelligence (AAAI), 2018, under review.

5. Yonghong Hou, Shuang Wang, Pichao Wang*, Zhimin Gao, and Wanqing Li. Spatially and Temporally Structured Global to Fine-grained Aggregation of Dynamic Depth Information for Action Recognition. Submitted to IEEE Access, under review. (*Corresponding author)

6. Chang Tang, Wanqing Li and Pichao Wang*. Online Action Recognition based on Incremental Learning of Weighted Covariance Descriptors. Submitted to IEEE Transactions on industrial informatics, under review. (*Corresponding author)

7. Chuankun Li, Yonghong Hou, Pichao Wang* and Wanqing Li. Decision Fusion Based Heterogeneous Networks for 3D Action Recognition. Submitted to Information Fusion, under review. (*Corresponding author)

[This page is intentionally left blank]

Contents

List of Figures

Chapter 1

Introduction

Among the several human-centered research domains (human detection, tracking, pose estimation and motion recognition) in computer vision, human motion recognition is particularly important due to its wide applicability in video surveillance, human computer interfaces, ambient assisted living, human-robot interaction, intelligent driving, etc. The task of motion recognition entails the automatic identification of human behaviors from images or video sequences. Depending on the complexity and duration of the motion, it can be broadly categorized into four kinds: gesture, action, interaction and group activity. Specifically, *gesture* is defined as the basic movement or position of the hand, arm, body, or head that is expressive of an idea, opinion, emotion, etc. "Hand waving" and "nodding" are some typical examples of gestures. Usually, a gesture has relatively short duration and the complexity is low. *Action* is considered as a type of activity that is performed by a single person and involves multiple body parts. Generally it is a combination of multiple gestures, such as "walking" and "punching". *Interaction* is a type of activity performed by two actors; one actor is a human while the other is a human or an object. This implies that the interaction entails human-human or human-object interaction. "Hugging each other" and "playing guitar" are examples of these two kinds of interaction, respectively. *Group activity* is the most complex type of activity, and it may combine gestures, actions and interactions. It involves more than two humans and a single or multiple objects. "Two teams playing basketball" and "group meeting" are examples of group activities.

Since the 1980s, researchers have been working on human motion recognition from 2D images or videos [AC99, WHT03, TCSU08, Pop10, GL14, ZSXF16]. Most of the early research efforts used color and texture cues in 2D images for recognition. However, due to problems such as background clutter, partial occlusion, view-point, lighting changes, execution rate and biometric variation, motion analysis from 2D images and videos is still a challenging task even for current deep learning approaches [HHP17, HAS+17]. With the development of cost-effective RGB-D sensors, such as Microsoft Kinect ™and Asus Xtion, RGB-D-based motion recognition has attracted much attention in recent years. Depth is insensitive to illumination changes and includes rich 3D structural information of the scene; 3D positions of body joints can be estimated from depth maps [SFC+11]. As a consequence, RGB-D-based human motion recognition has attracted more and more attention and shown a promising direction for human motion analysis.

RGB-D data for human motion analysis comprises three modalities: RGB, depth and skeleton. The main characteristics of RGB data is its shape, color and texture which brings the benefits of detecting interesting points and extracting optical flows. Compared to RGB videos, the ideal depth modality is insensitive to illumination variations, invariant to color and texture changes, reliable for estimating body silhouette and skeleton, and provides rich 3D structural information of the scene. Different from RGB and depth, skeleton data which consists of the positions of body joints, is a relatively high-level feature for motion recognition. It is robust to scale and illumination changes, and can be invariant to camera view as well as body rotation. In many state-of-the-art datasets, skeleton is computed from depth map. Many different methods have been proposed in the past decade to exploit the properties of the three modalities. These methods can be broadly classified into handcrafted and deeply learned representations. This thesis presents an extensive study on analyzing human motion from RGB-D modalities. The study ranges from hand-crafted features to deep-learning methods for segmented action recognition by addressing a number of challenging questions.

1.1 Research Questions

The main research questions addressed in this thesis are:

1. How to effectively mine the most frequent and relevant (discriminative, representative and non-redundant) features from skeleton data for action recognition?
2. How to effectively represent skeleton sequences for ConvNets-based recognition?
3. How to adopt ConvNets for depth-based recognition on small training data?
4. How to take full advantages of depth modality for large-scale action recognition based on ConvNets?
5. How to apply ConvNets to fine-grained action recognition using noisy depth modality?
6. How to fuse RGB and depth modalities at data-level for action recognition using ConvNets?
7. How to cooperatively train a single network using two heterogeneous input modalities (e.g. RGB and depth)?

1.2 Thesis Organization

The thesis is organized as follows:

Chapter 2 provides a review of the relevant literature. In this chapter, both hand-crafted features and deep learning methods for RGB, depth, skeleton and multi-modal-based action recognition are reviewed. Performance evaluation including benchmark datasets and evaluation metrics are also reviewed.

Chapter 3 addresses the first two research questions by presenting two methods for skeleton-based action recognition using hand-crafted features and a deep learning method, respectively.

Chapter 4 presents three methods to address the research questions 3-5. These methods adopt depth modality as input and take advantages of pre-trained deep learning models over ImageNet for action recognition.

Chapter 5 studies the research questions 6-7 and introduces two methods for RGB and depth based action recognition. The first method is based on extraction and use of scene flow for action recognition from RGB-D data. The second method introduces a concept of cooperatively training that takes two heterogeneous inputs and trains a single network for both homogeneous and heterogeneous action recognition.

The thesis is concluded with future research directions in *Chapter 6*.

Chapter 2

Literature Review

This chapter reviews key literature related to hand-crafted feature based and deeply learned feature based action recognition. Performance indicators and commonly used benchmark datasets are also reviewed.

2.1 Hand-crafted Features for Action Recognition

The process of action recognition based on hand-crafted features can be generally divided into two main steps, action representation and action classification. Action representation consists of feature extraction and feature selection. Features can be extracted from input sources such as depth maps, skeleton and/or RGB images. Regardless of the input source, there are two main approaches, space-time approach and sequential approach, to the representation of actions. The space-time approach usually extracts local or holistic features from space-time volume, without explicit modeling of temporal dynamics. By contrast, the sequential approach normally extracts local features from each frame of the input source and models the dynamics explicitly. Action classification is the step of learning a classifier based on action representation and classifying any new observations using the classifier. For space-time approaches, discriminative classifier, such as Support Vector Machine (SVM), is often used for classification. For the sequential approach, generative statistical models, such as Hidden Markov Model (HMM), are commonly used. In this literature review, we first give a brief review on RGB-based approach and then introduce the skeleton-based approach, depth-based approach, and the approach that fuses them together.

2.1.1 RGB-based Approach

The RGB-based methods rely on sequential RGB images, whether based on local representations or global representations. This approach enjoys a rich history but it is still very challenging for action recognition in the wild due to the difficulties such as great intra-class variance, scaling, occlusion and clutter. In this section, we only give a brief review, listing the typical works in corresponding modules. For more comprehensive review, survey papers [ZSXF16, HHP17] are recommended to read. In the following two sub-sections, we will first review space-time based approach and then sequential approach.

2.1.1.1 RGB-based Space-Time Methods

The space-time approach represents actions in volume, trajectories, and set of features and trains models for each kind of representation. For space time volume, Bobick and Davis [BD01a] proposed to recognize actions by using two components of vector images, namely, Motion Energy Images (MEI) and Motion History Images (MHI). It worked well in static background where the motion of object movement can be separated easily. Blank et al. [BGS+05] represented actions as space-time shapes, which contained both spatial and dynamic information. This method worked fast and did not need prior video alignment.

For trajectories, Campbell and Bobick [CB95] proposed to recognise nine atomic movements of a ballet dancer by tracking trajectories of joint positions in a 3-D XYT plane. Recently, Wang et al. [WKSL13] proposed a video representation based on dense trajectories and motion boundary descriptors, in which optical flow algorithm was used to extract trajectories. Their approach achieved promising results in several benchmark datasets but it was very time-consuming to calculate the dense optical flow and corresponding features along the flows.

For space-time features, extensive works have been done. To extract local spatio-temporal features, two main steps are: feature detection and feature description. The feature detector aims to detect locations of representative interest points with various scales. The shape and motion characteristics of the detected 3D patches (or interest regions surrounding the detected interest points) can be further described by feature descriptors. Many feature detectors have been proposed in the past years. For example, Laptev and Lindeberg [Lap05] proposed a generalization of Harris and Forstner interest point detector to localize the compact representation of the event; Gilber et al. [GIB09] used a 2D Harris corner detection and data mining approach to localize multiple actions in real-time. Dollar et al. [DRCB05] proposed a spatio-temporal feature as the cuboids prototype for the recognition of human actions. Other feature detectors, such as 3D-Hessian by Willems et al. [WTVG08], Dense Sampling by Fei-Fei and Perona [FFP05], Spatio-Temporal Regularity Based Feature (STRF) by Goodhart et al. [GYS08] also been proposed. And many STIP feature descriptors are also proposed, for example, HOG/HOF [Lap05], HOG3D [KMS08], Extended SURF [WTVG08] and MoSIFT [CH09]. The combinations of these feature detectors and feature descriptors have been used in many papers.

2.1.1.2 RGB-based Sequential Methods

For RGB-based sequential approach, Darrell and Pentland [DP93] proposed a Dynamic Time Warping (DTW) algorithm and used a view model to recognize gesture actions and effectively handle a variation in the execution of actions. Yamato et al.

[YOI92] adopted a HMM to represent and recognize the actions. Park and Aggarwal [PA04] proposed to estimate human body gestures using Bayesian networks and modelled the evolution of two persons interactions by Dynamic Bayesian Networks (DBN). Gupta and Davis [GD07] proposed a probabilistic model that exploit contextual information for visual action analysis to improve object recognition as well as action recognition. Ivanov and Bobick [IB00] suggested using stochastic context-free grammars (SCFGs) to model visual activities and used it on an upper layer to compute the probability of temporally consistent sequences of primitive actions. Many improved works have been done following above papers, which can be seen in the survey [VA13].

2.1.2 Skeleton-based Approach

The study of skeleton-based action recognition can date back to the pioneering work by Johansson [Joh75], which demonstrated that a large set of actions can be recognized solely from the joint positions. This idea has been followed and extensively explored ever since. However, the 3D joint positions extracted by the skeleton tracker [SFC$^+$11] are much noisy due to the possible failure caused by noisy depth maps or occlusions, which makes the design of an effective and efficient system not accurate enough. In the following two subsections, we also first review space-time approach, followed by sequential approach. Works [PLC16, HRHZ17] are referred to read for more comprehensive reviews.

2.1.2.1 Skeleton-based Space-Time Methods

For the skeleton-based space-time volume approach, Yang et al. [YT12] proposed a new feature descriptor called EigenJoints features which contained posture features, motion features and offset features. The pair-wise joint differences in current frames and their consecutive frames were used to encode the spatial and temporal information, which were called posture features and motion features, respectively. The difference of a pose with respect to the initial pose was called offset features. The initial pose was generally assumed as a neutral pose. The three channels were normalized and PCA was applied to reduce redundancy and noise to obtain the EigenJoints descriptor. A Naive-Bayes-Nearest-Neighbor (NBNN) classifier was adopted to recognize actions. Gowayyed et al. [GTHES13] proposed a new descriptor called Histograms of Oriented Displacements (HOD) to recognize actions. The displacement of each joint voted with its length in a histogram of oriented angles. Each 3D trajectory was represented by the HOD of its three 2D projection. In order to preserve temporal information, a temporal pyramid was proposed, where trajectories were considered as a whole, halves and quarters and then all the descriptors in

these three levels were concatenated to form the final descriptor. A linear SVM was used to classify actions based on the histograms. Similar to this work, Hussein et al. [HTGES13] proposed a descriptor called Covariance of 3D Joints (Cov3DJ) for human action recognition. This descriptor used covariance matrix to capture the dependence of locations of different joints on one another during an action. In order to capture the order of motion in time, a hierarchy of Cov3DJs was used, similarly to the work in [GTHES13].

Zanfir et al. [ZLS13] proposed a descriptor called moving pose which was formed by the position, velocity and acceleration of skeleton joints within a short time window around the current fame. To learn discriminative pose, a modified k-Nearest Neighbours (kNN) classifier was used that considered both the temporal location of a particular frame within the action sequence as well as the discrimination power of its moving pose descriptor compared to other frames in the training set. Wang et al. [WWY13] first estimated human joints positions from videos and then grouped the estimated joints into five parts. Each action was represented by computing sets of co-occurring spatial and temporal configurations of body parts. They used a bag of words method with the extracted features for classification. Ohn-Bar and Trivedi [OBT13a] tracked the joint angles and built a descriptor based on similarities between angle trajectories. This feature was further combined with a double-HOG descriptor that accounted for the spatio-temporal distribution of depth values around the joints. Theodorakopoulos et al. [TKEF14] initially processed the skeleton data from sensor coordinate to torso PCA frame in order to gain robust and invariant pose representation. Sparse coding in dissimilarity space was utilized to sparsely represent the actions. Chaaraoui et al. [CPLCPFR14] proposed to use an evolutionary algorithm to determine the optimal subset of skeleton joints, taking into account the topological structure of the skeleton. Vemulapalli et al. [VAC14] explicitly modelled the 3D geometric relationships between various body parts using rotations and translations in 3D space. Human actions were modelled as curves in Lie group and then they mapped the action curves from the Lie group to its Lie algebra. Following, they used DTW to handle rate variations and Fourier Temporal Pyramid (FTP) [WLWY14] representation to handle the temporal misalignment and noise issues.

2.1.2.2 Skeleton-based Sequential Methods

For the skeleton-based sequential approach, Xia et al. [XCA12] proposed a feature called Histograms of 3D Joint Locations (HOJ3D) as a representation of postures. The HOJ3D essentially encoded spatial occupancy information relative to the root joint, e.g. hip center. A modified spherical coordinate system was defined on the root joint and the 3D space was divided into N bins. The HOJ3D was reprojected

using LDA to reduce dimensionality and then clustered into K posture visual words which represented the prototypical poses of actions. HMMs were adopted to model the visual words and recognize actions. Radial distance was adopted in this spherical coordinate system which made the method to some extend view-invariant.

Koppula et al. [KGS13] explicitly modelled the motion hierarchy to enable their method to handle simple human-object interactions. The human activities and object affordances were jointly modelled as a Markov Random Field (MRF) where the nodes represented objects and sub-activities, and the edges represented the relationships between object affordances, their relations with sub-activities, and their evolution over time. Feature vectors that represented the object's location and the changing information in the scene were defined by training a Structural Support Vector Machine (SSVM). Similar to this approach, Sung et al. [SPSS12] proposed a hierarchical two-layer Maximum Entropy Markov Model (MEMM) to represent an activity. The lower layer nodes represented sub-activities while higher level nodes described more complex activities, for example, "lifting left hand" and "pouring water" could be described as a sub-activity and a complex activity, respectively. With the development of deep learning, Wu and Shao [WS14b] proposed a hierarchical dynamic framework that first extracted high level skeletal joints features and then used the learned representation for estimating emission probability to infer action sequences. They replaced Gaussian mixture models with deep neural networks that contained many layers of features to predict probability distribution over states of HMM, which achieved better results.

2.1.3 Depth-based Approach

The depth-based methods rely mainly on features, either local or global, extracted from the space time volume. Compared to visual data, depth maps provide geometric measurements that are invariant to lighting. However, it is still a challenging task using depth maps to design a system for action recognition which are both effective and efficient, even though depth can make segmentation of foreground and background easier. The reasons are three folds. First of all, depth sequence may contain serious occlusions, which makes the global features unstable. Secondly, the depth maps may have many "holes" due to no estimation of depth obtained in case of specific material, reflection, interference or fast motion. In addition, the depth maps do not have as much texture as color images do, and they are usually too noisy to apply local differential operators such as gradients. These challenges motivate researchers to develop features that are semi-local, highly discriminative and robust against occlusion. The majority of depth-based methods rely on space-time volume features, and we will review the literature space-time approach first, followed by

sequential approach.

2.1.3.1 Depth-based Space-Time Methods

For depth-based space-time approaches, Li et al. [LZL10] proposed a bag-of-points feature representation for activity recognition from depth map sequences, where the 3D points were sampled from the silhouettes of the depth maps. They used an action graph as their classification framework, where each action was encoded in one or multiple paths in the action graph. Each node of the action graph denoted a salient postures. One limitation of this approach was the loss of spatial context information between interest points. In addition, this approach was view-dependent, and this made it very difficult robustly sample the interest points in different views. To address these issues, Vieira et al. [VNO+12] proposed a feature descriptor called Space-Time Occupancy Patterns (STOP), in which the depth sequence was represented in a 4D space-time grid by dividing space and time axes into multiple segments. In this way, the descriptor could preserve spatial and temporal contextual information between space-time cells and be flexible to accommodate intra-action variations.

Yang et al. [YZT12] projected depth maps onto three orthogonal planes and accumulate global activities through entire video sequences to generate the Depth Motion Maps (DMM). Histograms of Oriented Gradients (HOG) were then computed from DMM as the representation of an action video. Oreifej and Liu [OL13] presented a new descriptor called histogram of oriented 4d surface normals (HON4D) to capture the complex joint shape-motion cues at pixel-level. The histogram could capture the distribution of the surface normal orientation in the 4d volume of time, depth and spatial coordinates. Wang et al. [WLC+12] treated a three-dimensional action sequence as a 4d shape and propose a semi-local features called random occupancy pattern (ROP) features. The ROP features were extracted from randomly sampled 4d sub-volumes with different sizes and at different locations.

Xia and Aggarwal [XA13] proposed a filtering scheme to find local spatio-temporal interest points (STIPs) from depth videos with noise suppression functions to deal with the noisy data and missing values in depth maps. They also proposed a self-similarity depth cuboid feature (DCSF) as the descriptor for a spatio-temporal depth cuboid which further handled the noisy measurements and missing values. Liu and Shao [LS13a] presented a Genetic Programming (GP) learning method to select discriminative spatio-temporal features from RGB-D sensors for action recognition. They proposed a restricted graph-based genetic programming (RGGP) method which assembled 3D operators as graph-based combinations, and then evolved generation by generation by evaluating the average error rate of the classification accuracy, finally obtained the discriminative representation of RGB

and depth information. Luo et al. [LWQ14] proposed a framework using both RGB videos and depth maps to action recognition. They proposed a sparse coding-based temporal pyramid structure matching approach (ScTPM) for feature representation, keeping the temporal information and reducing the approximation error compared to bag-of-words model and k-means, respectively. They proposed a center-symmetric motion local ternary pattern (CS-Mltp) descriptor to capture the spatial-temporal features from RGB videos. Then they fused the features captured from both depth maps and RGB videos to recognize actions.

Song et al. [STLY14] proposed a new depth descriptor called Body Surface Context (BSC) by utilizing 3D point cloud which contained points in the 3D real-world coordinate system to represent the external surface of human body. This descriptor described the distribution of relative locations of the neighbors for a reference point in the point cloud by encoding the cylindrical angular of the difference vector between the target point and the reference point. This descriptor was some kind of object-centered feature and robust to translations and rotations. Lu et al. [LJT14] proposed a binary range-sample feature in depth through τ tests. They developed six pixel pairs, Back-Both, Act-Both, Occ-Both, Back-Act, Back-Occ and Occ-Act. Their τ test only sampled pixel pairs Act-Both and Back-Act where they recognised Back-Act mostly describing the human body outline which was useful in recognition tasks. This descriptor worked in a high speed due to its binary property. Yang and Tian [YT14] clustered hypersurface normals in a depth sequence to form the polynormal which was used to jointly characterize the local motion and shape information. An adaptive spatio-temporal pyramid was introduced which subdivided a depth video into a set of space-time grids to globally capture the spatial and temporal orders. They then aggregated the low-level polynormals into the super normal vector (SNV) which was a simplified version of the Fisher kernel representation. Rahmani et al. [RMHM14] proposed a new descriptor and keypoint detection algorithm by directly processing process the pointclouds. The proposed descriptor was extracted at a point by encoding the Histogram of Oriented Principal Components (HOPC) within an adaptive spatio-temporal support volume around the point. By directly processing the pointclouds, their algorithm could handle view-point variations to some extent.

2.1.3.2 Depth-based Sequential Methods

For pure depth-based sequential approaches, there are few approaches to explore the possibility of explicitly modeling temporal dynamics from depth maps due to the difficulties in extracting reliable temporal correspondences, because local differential operators are not suitable for extracting features from depth maps. However, researchers try to design temporal motion features that are between pure depth-based

methods and skeleton-based methods, for skeletons are one of the most natural features that embed such motion information.

Inspired by the success of silhouette based methods developed for visual data, Jalal et al. [JUKK11] extracted depth silhouettes to construct feature vectors. They applied R transform on the depth silhouette to obtain compact shape representation reflecting time-sequential profiles of the activities. Principal Component Analysis (PCA) was then used to reduce feature dimension. Linear Discriminant Analysis (LDA) was adopted to extract most discriminant vectors and HMM was utilized for recognition.

2.1.4 Multi-modal-based Approach

Although skeleton extraction has become much easier thanks to the skeleton tracker [SFC+11], only the estimated 3D joint positions are still not sufficient to design a system for effective action recognition. One reason is that the estimated joint positions are very noisy and often incorrect when there are occlusions among human limbs such as two limbs crossing each other. Furthermore, the motion of 3D joint positions is insufficient to distinguish similar activities that involve interactions between objects and subjects. Consequently, some researchers start to explore fusion techniques that can be used to enhance the classification performance of human action recognition.

To fuse depth-based features with skeleton-based features, Althloothi et al. [AMZV14] presented two sets of features, features for shape representation extracted from depth data by using a spherical harmonics representation and features for kinematic structure extracted from skeleton data by estimating 3D joint positions. The shape features were used to describe the 3D silhouette structure while the kinematic features were used to describe the movement of the human body. Both sets of features were fused at the kernel level for action recognition by using Multiple Kernel Learning (MKL) technique. Similar to this direction, Chaaraoui et al. [CPLFR13] proposed a fusion method to combine skeleton and silhouette-based features. The skeletal features were obtained by normalizing the 3D position of original skeleton data while the silhouette-based features were generated by extracting contour points of the silhouette. After feature fusion, a model called bag of key poses was employed for action recognition. The key poses were obtained by K-means clustering algorithm and the words were made up of key poses. In recognition stage, unknown video sequences were classified based on sequence matching.

Rahmani et al. [RMMH14] proposed an algorithm combining the discriminative information from depth maps as well as from 3D joints positions for action recognition. To avoid the suppression of subtle discriminative information, local information

integration and normalization were performed. Joint importance was encoded by using joint motion volume. Random Decision Forest (RDF) was trained to select the discriminant features. Because of the low dimensionality of their features, their method turned to be efficient. Wang et al. [WLWY14] proposed a Local Occupancy Patterns (LOP) feature calculated from the 3D point cloud around a particular joint to discriminate different types of interactions and Fourier Temporal Pyramid (FTP) to represent the temporal structure. Based on above two types of features, a model called Actionlet Ensemble Model (AEM) was proposed which was a combination of the features for a subset of the joints. Due to the numerous actionlets, data mining technique was used to discover discriminative actionlets. Both skeleton and point cloud information were utilized to recognize human-objects interactions. To represent dynamics and appearance of parts, Shahroudy et al. [SNYW16] employed a heterogeneous set of depth and skeleton based features, and proposed a joint structured sparsity regression based learning method which integrated part selection into the learning process considering the heterogeneity of features for each joint.

2.2 Deep Learning for Action Recognitoin

In this section, we first review the basic concepts of deep learning and then present RGB-based, skeleton-based, depth-based and multi-modal-based methods with deep learning for action recognition.

2.2.1 Commonly-used Networks

In this section, we introduce the deep learning concepts and architectures that are relevant or have been applied to RGB-D-based motion recognition. Readers who are interested in more background and techniques are referred to [GBC16].

2.2.1.1 Neural Networks

Neural networks are the basis of most deep architectures, and it is a generation of linear or logistic regression. The activation a of each neuron denotes a linear combination of several input $\boldsymbol{x}$ and a set of learned parameters, $\boldsymbol{w}$ and b, followed by an element-wise non-linear activation function $\sigma(\cdot)$:

$$a = \sigma(\boldsymbol{w}^T \boldsymbol{x} + b) \tag{2.1}$$

A neural network consists of L layers of stacked neurons through which a signal is propagated as $\sigma(\boldsymbol{w}_L^T(\sigma(\boldsymbol{w}_{L-1}^T ...))$. When multiple, feed-forward layers are stacked in such a way, multi-layered perceptron (MLP) is constructed, where the intermedi-

ate layers are typically known as *hidden layers*. Deep neural network (DNN) is one kind of neural networks that contains many layers. Presently, there are various deep learning architectures and this research topic is fast-growing. In the following subsections, four main basic deep architectures including Stacked Auto-encoders (SAE), Restricted Boltzmann Machines (RBM), Convolutional Neural Networks (ConvNet) and Recurrent Neural Networks (RNN) and their corresponding variants are reviewed.

2.2.1.2 Auto-encoders (AE) and Its Variants

As a feed-forward neural network, auto-encoder (AE) consists of two phases including encoder and decoder. Encoder takes an input $\boldsymbol{x}$ and transforms it to a hidden representation $\boldsymbol{h}$ via a non-linear mapping as follows:

$$\boldsymbol{h} = \sigma(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}) \tag{2.2}$$

The decoder maps the hidden representation back to the original representation in a similar way:

$$\boldsymbol{z} = \sigma(\boldsymbol{W}^{'}\boldsymbol{h} + \boldsymbol{b}^{'}) \tag{2.3}$$

Model parameters $(\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{W}^{'}, \boldsymbol{b}^{'})$ are learned by minimizing the reconstruction error between $\boldsymbol{z}$ and $\boldsymbol{x}$. It is clearly shown that AE can be trained in an unsupervised way. And the hidden representation $\boldsymbol{h}$ can be regarded as a more abstract and meaningful representation for data sample $\boldsymbol{x}$. SAE is formed by placing AE on top of each other, and it can be used to learn high-level representations. Since SAE can be trained in an unsupervised way, it provides an effective pre-training solution via initialization of the weights of deep neural network (DNN) to train the model. Once initialized, supervised fine-tuning is performed to minimize prediction error on a labeled training data. Usually, a softmax/regression layer is added on top of the network to map the output of the last layer in AE to targets. The pre-training protocol based on SAE can make DNN models have better convergence property compared to arbitrary random initialization. Denoising AE (DAE) [VLL^{+}10] is one commonly used and improved AE which takes a corrupted version of data as input and is trained to reconstruct/denoise the clean input $\boldsymbol{x}$ from its corrupted sample. DAE can learn more robust representation and prevent the learning of the identity transformation.

2.2.1.3 Restricted Boltzmann Machines (RBM) and Its Variants

As a special type of Markov Random Field (MRF), RBM [Hin10] consists of an input layer (visible layer) $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$ and a hidden layer $\boldsymbol{h} = (h_1, h_2, \ldots, h_M)$.

The bidirectional connections between the two layers reveal it is a generative model; the latent feature representation $\boldsymbol{h}$ can be obtained by giving an input vector $\boldsymbol{x}$, and vice versa. Given the model parameters $(\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{a})$, the energy function is formulated as:

$$E(\boldsymbol{x}, \boldsymbol{h}) = \boldsymbol{h}^T \boldsymbol{W} \boldsymbol{x} - \boldsymbol{b}^T \boldsymbol{x} - \boldsymbol{a}^T \boldsymbol{h} \tag{2.4}$$

The joint distribution over all the neurons is calculated based on the energy function as:

$$p(\boldsymbol{x}, \boldsymbol{h}) = \frac{1}{Z} exp\{-E(\boldsymbol{x}, \boldsymbol{h})\} \tag{2.5}$$

where $Z = \Sigma_{\boldsymbol{x},\boldsymbol{h}} exp(-E(\boldsymbol{x}, \boldsymbol{h}))$ is the partition function and computing it is generally intractable. However, computing $\boldsymbol{h}$ conditioned on $\boldsymbol{x}$ or vice versa is tractable by conditional inference and it can be derived into a simple formula as:

$$P(h_j|\boldsymbol{x}) = \frac{1}{1 + exp\{-a_j - \boldsymbol{W}_j \boldsymbol{x}\}} \tag{2.6}$$

Since this network is bidirectional and symmetric, a similar expression holds for $P(x_i|\boldsymbol{h})$. RBM is trained to maximize the joint probability and the learning of $\boldsymbol{W}$ is conducted through the contrastive divergence method [Hin02].

Deep belief networks (DBN) [Hin09] are essentially SAEs where the AE layers are replaced by RBMs. Hence, it can be constructed by stacking multiple RBMs. Similarly to SAE, DBN can be trained in a greedy layer-wise unsupervised manner. Final fine-tuning is performed by adding a linear classifier to the top layer of the DBN and performing a supervised optimization.

2.2.1.4 Convolutional Neural Networks (ConvNet) and Its Variants

ConvNet was proposed by [LBD+90] and is renowned for image-based recognition. Convolution is the basic operation used to model a neuron to both learn and detect features, using a *kernel* convolved against an input window of pixels. Convolutions are used in a fashion akin to correlation template or feature detector. The output of each convolutional filter is assembled into an output image called *feature map*, which is sent along as input to the next layer. One output image is created for each filter, and there are usually tens of filters per layer. Each convolutional filter acts as both a feature detector and a filter. The process can be formulated as follows. Suppose at each layer the input image is convolved with a set of K kernels $(\boldsymbol{W}_1, \boldsymbol{W}_2, ..., \boldsymbol{W}_K)$ and subsequently bias $(b_1, b_2, ..., b_K)$ are added, each generating a new feature map $\boldsymbol{X}_k$. These features are subjected to an element-wise non-linear transform $\sigma(\cdot)$ and the same process is repeated for every convolutional layer l:

$$\boldsymbol{X}_k^l = \sigma(\boldsymbol{W}_k^{l-1} \otimes \boldsymbol{X}^{l-1} + b_k^{l-1}) \tag{2.7}$$

where $\otimes$ denotes the convolutional operation. Convolutional layers are typically alternated with pooling layers where pixel values of neighborhoods are aggregated using some permutation invariant function, typically the max or mean operations, which induce a certain amount of translation invariance and further minimize the number of model parameters. The strength of ConvNet lies in its weight sharing of kernels, exploiting the intuition that similar structures occur in different locations in an image. This drastically reduces the amount of parameters that need to be learned, and renders the network equivalent with respect to translations of the input. Analyzing filters learned by ConvNet suggests that the very first layers learn low level features (e.g., Gabor-like filters) while top layers learn high level semantics [ZF14]. At the end of the convolutional stream of the network, fully-connected layers are usually added to act as classification or regression layers, where weights are no longer shared. Differently to SAE and DBN, ConvNet is typically trained end-to-end (as opposed to layer-by-layer) in a supervised manner.

Till now, there are several classical ConvNet architectures: LeNet [LBBH98], AlexNet [KSH12], VGG [SZ14b], GoogLeNet [SLJ+15] and ResNet [HZRS16]. Based on these classical architectures, several commonly used variants are proposed for video-based recognition, such as siamese networks [CHL05] for feature learning from egomotion [ACM15], Generative Adversarial Net (GAN) [GPAM+14] to model scene dynamic for video segmentation and generation tasks [VPT16b], and attention model for video face recognition [YRZ+17].

2.2.1.5 Recurrent Neural Networks (RNN) and Its Variants

RNN [WZ89] is a class of dynamic, nonlinear systems for mapping sequences to sequences using the concept of virtual time. It uses an internal state space composed from a trace of the inputs seen so far. RNN also implements a form of *memory* via the recurrent inputs, which is useful for modeling sequences composed of current and past states. Compared to other finite state models such as Hidden Markov Model (HMM) [RJ86], RNN is trainable, and much more efficient and compact for sequence representation and prediction, distributing the memory states across the network in uniform memory cells, rather than forcing each state of the model to store all possible state transitions. The plain RNN maintains a latent or hidden state $\boldsymbol{h}$ at time t that is some non-linear mapping from its input $\boldsymbol{x}_t$ and previous state $\boldsymbol{h}_{t-1}$:

$$\boldsymbol{h}_t = \sigma(\boldsymbol{W}\boldsymbol{x}_t + \boldsymbol{V}\boldsymbol{h}_{t-1} + \boldsymbol{b}) \tag{2.8}$$

where weights matrices $\boldsymbol{W}$ and $\boldsymbol{V}$ are shared over time. For classification, some fully connected layers are typically added followed by a softmax to map the sequence to a posterior over the classes.

An RNN can be remapped as a flow graph over a sequence of inputs, and then the flow graph can be unfolded into a feed-forward network (FNN). Unfolding allows the forward pass and backward pass through the RNN to be visualized, and also enables back-propagation through time (BPTT [Wer88]). Weight sharing is an artifact of RNN unfolding into an FNN where the weights $\boldsymbol{W}$ and $\boldsymbol{V}$ are implicitly shared at each time step. The idea of sharing weights allows for generalization to new sequences similar to the learned sequences. But in this respect, weight sharing provides for a statistical modeling capability that allows generalization and approximation. Contrast this with an exhaustive, logical exact-match modeling capability that requires a larger memory system containing all known sequences to match against. Generalization and weight sharing for sequences and subsequences implies variable precision. The final effect of weight sharing is that the sequence matching is not precise, but rather approximated, so an appropriate distance function must be used to predict the match probability. Therefore the final classification and matching is similar to pooling each RNN cell in the sequence, and then combining the strength of the activations of each cell into the final match probability for a given sequence.

During the training process, gradient needs to be back-propagated from the output through time. RNN is inherently deep in time and consequently suffer from the problems of gradient fading or exploding [BSF94]. To this end, several variants have been developed. Long Short Term Memory (LSTM) [HS97] is a commonly used variant. For all the LSTM neurons in some layer, at time t, the recursive computation of activations of the units is:

$$\begin{pmatrix} \boldsymbol{i}_t \\ \boldsymbol{f}_t \\ \boldsymbol{o}_t \\ \boldsymbol{u}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} \left(\boldsymbol{W} \begin{pmatrix} \boldsymbol{x}_t \\ \boldsymbol{h}_{t-1} \end{pmatrix} \right) \tag{2.9}$$

$$\boldsymbol{c}_t = \boldsymbol{i}_t \circ \boldsymbol{u}_t + \boldsymbol{f}_t \circ \boldsymbol{c}_{t-1} \tag{2.10}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \circ \tanh(\boldsymbol{c}_t) \tag{2.11}$$

where $\boldsymbol{i}_t$, $\boldsymbol{f}_t$, $\boldsymbol{o}_t$, $\boldsymbol{u}_t$, $\boldsymbol{h}_t$, $\boldsymbol{c}_t$ are input gate, forget gate, output gate, input modulation gate, output state and internal memory cell state respectively. Operator $\circ$ indicates element-wise product. The input gate and forget gate govern the information flow into and out of the cell. The output gate controls how much information from the cell is passed to the output $\boldsymbol{h}_t$. The memory cell has a self-connected recurrent edge of weight 1, ensuring that the gradient can pass across many time steps without

vanishing or exploding. Therefore, it overcomes the difficulties in training the RNN model caused by the vanishing gradient effect.

A recent variation of the LSTM is the Gated Recurrent Unit (GRU) [CVMG+14], which used only two control gates: the input gate and the dynamic gate, providing a simpler model for back-propagation tuning. If considering the current cell value and the new input value as two inputs to the GRU neural function, then the GRU dynamic gate allows for weight combinations of current value and input value.

2.2.2 RGB-based Approach

RGB is one important channel of RGB-D data. Compared with depth and skeleton modalities, the main characteristics of RGB data are its shape, color and texture which bring the benefits of extracting interesting points and optical flows [WKSL13]. These properties also make it effective to directly use texture-driven feature extraction networks, such as 2D CNN [KSH12, SZ14b, HZRS16] to extract frame-level spatial information. Generally speaking, we define three categories, CNN-based, RNN-based and other-architecture-based approaches.

2.2.2.1 CNN-based Approach

For CNN-based approach, currently there are mainly four approaches to encode spatial-temporal-structural information.

The first approach applies CNN to extract features from individual frames and fuse the temporal information later. For example [KTS+14] investigated four temporal fusion methods, and proposed the concept of slow fusion where higher layers get access to progressively more global information in both spatial and temporal dimensions. The implementation extends the connectivity of all convolutional layers in time and carry out temporal convolutions. [NHV+15] explored several temporal pooling methods and concluded that max pooling in the temporal domain is preferable.

The second approach is to extend convolutional operation into temporal domain. In one such implementation, [JXYY13] proposed 3D-convolutional networks using 3D kernels (filters extended along the time axis) to extract features from both spatial and temporal dimensions. This work empirically showed that the 3D-convolutional networks outperform the 2D frame-based counterparts. With modern deep architectures, such as VGG [SZ14b], and large-scale supervised training datasets, such as Sports-1M [KTS+14], [TBF+15] extended the work [JXYY13] by inclusion of 3D pooling layers, and proposed a generic descriptor called *C3D* by averaging the outputs of the first fully connected layer of the networks. However, both of these works

break the video sequence into short clips and aggregate video-level information by late score fusion. This is likely to be suboptimal when considering some long action sequence, such as walking or swimming that lasts several seconds and spans tens or hundreds of video frames. To handle this problem, [VLS16] investigated the learning of long-term video representations and proposed Long-term Temporal Convolutions (LTC) at the expense of decreasing spatial resolution to keep the complexity of networks tractable. Even straightforward and mainstreamed, extending spatial kernels to 3D spatial-temporal ones inevitably increases the number of parameters of the network. To relieve the drawbacks of 3D kernels, [SJYS15] factorized a 3D filter into a combination of 2D and 1D filters.

The third approach is to encode the video into dynamic images that contain the spatial-temporal information and then apply CNN for image-based recognition. [BFG+16] proposed to adopt rank pooling [FGM+16] to encode the video into one dynamic set of images and used pre-trained models over ImageNet [KSH12] for fine-tuning. The end-to-end learning methods with rank pooling is also proposed in [BFG+16, FG16]. Hierarchical rank pooling [FAHG16] was proposed to learn higher order and non-linear representations compared to the original work. Generalized rank pooling [CFHG17] was introduced to improve the original method via a quadratic ranking function which jointly provided a low-rank approximation to the input data and preserved their temporal order in a subspace.

Besides the above works that aim to adopt one network to exploit both spatial-temporal information contained in the video, the fourth approach is to separate the two factors and adopt multiple stream networks. [SZ14a] proposed one spatial stream network fed with raw video frames, and one temporal stream network accepting optical flow fields as input, and the two streams were fused together using the softmax scores. [WQT15] extended the two-stream networks by integrating improved trajectories [WKSL13], where trajectory-constrained sampling and pooling were used to encode deep features learned from deep CNN architecture, into effective descriptors. To incorporate long-range temporal structure using the two-stream networks, [WXW+16] devised a temporal segment network (TSN) that used a sparse sampling scheme to extract short snippets over a long video sequence. With the removal of redundancy from consecutive frames and a segmental structure, aggregated information was obtained from the sampled snippets. To reduce the expensive calculation of optical flow, [ZWW+16] accelerated this two stream structure by replacing optical flow with motion vector which could be obtained directly from compressed videos without extra calculation. [WSW+16] leveraged semantic cues in video by using a two-stream semantic region-based CNNs (SR-CNNs) to incorporate human/object detection results into the framework. In their work, [CLS15] exploited spatial structure of the human pose and extracted a pose-based convolutional neural

network (P-CNN) feature from both RGB frames and optical flow for fine-grained action recognition. [WFG16] formulated the problem of action recognition from a new perspective and modelled an action as a transformation which changed the state of the environment before the action to the state after the action. They designed a Siamese network which modelled the action as a transformation on a high-level feature space based on the two-stream model. Based on the two-stream framework, [ZHS+16] proposed a key volume mining deep framework for action recognition, where they identified key volumes and conducted classification simultaneously. Inspired by the success of Residual Networks (ResNets) [HZRS16], [FPW16] injected residual connections between the two streams to allow spatial-temporal interaction between them. Instead of using optical flow for temporal stream, [LRVH16] adopted Motion History Image (MHI) [BD01b] as the motion clue. The MHI was combined with RGB frames in a spatio-temporal CNN for fined grained action recognition. However, all the methods reviewed above incorporated the two streams from separate training regimes; any registration of the two streams was neglected. In order to address this gap and propose a new architecture for spatial-temporal fusion of the two streams [FPZ16] investigated three aspects of fusion for the two streams: (i) how to fuse the two networks with consideration for spatial registration, (ii) where to fuse the two networks and, (iii) how to fuse the networks temporally.

2.2.2.2 RNN-based Approach

For RNN-based approach, [BMW+11] tackled the problem of action recognition through a cascade of 3D CNN and LSTM, where the two networks were trained separately. Differently from the separate training, [DAHG+15] proposed one Long-term Recurrent Convolutional Network (LRCN) to exploit end-to-end training of the two networks. To take full advantages of both CNN and RNN, [NHV+15] aggregated CNN features with both temporal pooling and LSTM for temporal exploitation, and fused the output scores from the feature pooling and LSTM network to conduct final action recognition. [PVDOD+16] proposed an end-to-end trainable neural network architecture incorporating temporal convolutions and bidirectional LSTM for gesture recognition. This provided opportunity to mine temporal information that was much discriminative for gesture recognition. [SKS16] proposed a soft attention model for action recognition based on LSTM. The attention model learns which parted in the frames were relevant for the task at hand and attached higher importance to them. To take advantages of both Fisher Vector [SPMV13] and RNN, [LSKW16] introduced a Recurrent Neural Network Fisher Vector (RNN-FV) where the GMM probabilistic model in the fisher vector was replaced by a RNN and thus avoided the need for the assumptions of data distribution in the GMM. Even though RNN was remarkably capable of modeling temporal dependences, it lacked an intu-

itive high-level spatial-temporal structure. To mine the spatio-temporal-structural information, [JZSS16] combined the power of spatio-temporal graphs and RNN for action recognition.

2.2.2.3 Other-architecture-based Approach

Besides the commonly used CNN- and RNN-based methods for motion recognition from RGB modality, there are several other architectures that have been adopted for this task. [JSWP07] used a feedforward hierarchical template matching architecture for action recognition with pre-defined spatio-temporal filters in the first layer. [Che10] adopted the convolutional RBM (CRBM) as the basic processing unit and proposed the so-called space-time Deep Belief Network (ST-DBN) that alternated the aggregation of spatial and temporal information so that higher layers captured longer range statistical dependencies in both space and time. [TFLB10] extended the Gated RBM (GRBM) [MH07] to convolutional GRBM (convGRBM) that shared weights at all locations in an image and inference was performed through convolution. [LZYN11] presented an extension of the independent subspace analysis algorithm [The07] to learn invariant spatio-temporal features from unlabeled video data. They scaled up the original ISA to larger input data by employing two important ideas from convolutional neural networks: convolution and stacking. This convolutional stacking idea enabled the algorithm to learn a hierarchical representation of the data suitable for recognition. [YCSC14] proposed *Dynencoder*, a three layer auto-encoder, to capture video dynamics. Dynencoder was shown to be successful in synthesizing dynamic textures, and one can think of a Dynencoder as a compact way of representing the spatio-temporal information of a video. Similarly, [SMS15] introduced a LSTM autoencoder model. The LSTM autoencoder model consisted of two RNNs, namely, the encoder LSTM and the decoder LSTM. The encoder LSTM accepted a sequence as input and learned the corresponding compact representation. The states of the encoder LSTM contained the appearance and dynamics of the sequence. The decoder LSTM received the learned representation to reconstruct the input sequence. Inspired by the Generative Adversarial Networks (GAN) [GPAM$^+$14], [MCL16] adopted the adversarial mechanism to train a multi-scale convolutional network to generate future frames given an input sequence. To deal with the inherently blurry predictions obtained from the standard Mean Squared Error (MSE) loss function, they proposed three different and complementary feature learning strategies: a multi-scale architecture, an adversarial training method, and an image gradient difference loss function.

2.2.3 Skeleton-based Approach

Differently from RGB and depth, skeleton data contains the positions of human joints, which can be considered relatively high-level features for motion recognition. Skeleton data is robust to scale and illumination changes, and can be invariant to camera view as well as human body rotation and motion speed. Currently, there are mainly three approaches to skeleton-based motion recognition using deep learning: (i) RNN-based, (ii) CNN-based, (iii) other-architecture-based approaches.

2.2.3.1 CNN-based Approach

The main step in this approach is to convert the skeleton sequences into images where the spatio-temporal information is reflected in the image properties including color and texture. [DFW15] represented a skeleton sequence as a matrix by concatenating the joint coordinates at each instant and arranging the vector representations in a chronological order. The matrix was then quantified into an image and normalized to handle the variable-length problem. The final image was fed into a CNN model for feature extraction and recognition. [WLHL16] proposed to encode spatio-temporal information contained in the skeleton sequence into multiple texture images, namely, Joint Trajectory Maps (JTM), by mapping the trajectories into HSV (hue, saturation, value) space. Pre-trained models over Imagenet was adopted for fine-tuning over the JTMs to extract features and recognize actions. Similarly, [HLWL16] drew the skeleton joints with a specific pen to three orthogonal canvases, and encoded the dynamic information in the skeleton sequences with color encoding. [LHWL17] proposed to encode the pair-wise distances of skeleton joints of single or multiple subjects into texture images, namely, Joint Distance Maps (JDM), as the input of CNN for action recognition. Compared with the works reported by [WLHL16] and [HLWL16], JDM was less sensitive to view variations. [LLC17] introduced an enhanced skeleton visualization method to represent a skeleton sequence as a series of visual and motion enhanced color images. They proposed a sequence-based view invariant transform to deal with the view variation problem, and multi-stream CNN fusion method was adopted to conduct recognition. [KAB+17] designed vector-based features for each body part of human skeleton sequences, which were translation, scale and rotation invariant, and transformed the features into images to feed into CNN for learning high level and discriminative representation. [KBA+17] represented the sequence as a clip with several gray images for each channel of the 3D coordinates, which reflected multiple spatial structural information of the joints. The images were fed to a deep CNN to learn high-level features, and the CNN features of all the three clips at the same time-step were concatenated in a feature vector. Each feature vector represented the temporal in-

formation of the entire skeleton sequence and one particular spatial relationship of the joints. A Multi-Task Learning Network (MTLN) was adopted to jointly process the feature vectors of all time-steps in parallel for action recognition. [KR17] approached the problem differently and proposed to use the Temporal Convolutional Neural Networks (TCN) [LFV+17] for skeleton based action recognition. They redesigned the original TCN into Res-TCN by factoring out the deeper layers into additive residual terms that yielded both interpretable hidden representations and model parameters.

2.2.3.2 RNN-based Approach

In this class of approaches, skeleton features are input to a RNN in order to exploit the temporal evolution. For instance, [DWW15, DFW16] divided the whole skeleton sequence into five parts according to the human physical structure, and separately fed them into five bidirectional RNNs/LSTMs. As the number of layers increased, the representations extracted by the subnets were hierarchically fused to build a higher-level representation. This method explicitly encoded the spatio-temporal-structural information into high level representation. [VZQ15] proposed a differential gating scheme for the LSTM neural network, which emphasized the change in information gain caused by the salient motions between the successive frames. This work was one of the first aimed at demonstrating the potential of learning complex time-series representations via high-order derivatives of states. [ZLX+16] designed two types of regularizations to learn effective features and motion dynamics. In the fully connected layers, they introduced regularization to drive the model to learn co-occurrence features of the joints at different layers. Furthermore, they derived a new dropout and applied it to the LSTM neurons in the last LSTM layer, which helped the network to learn complex motion dynamics. Instead of keeping a long-term memory of the entire body's motion in the cell, [SLNW16] proposed a part-aware LSTM human action learning model (P-LSTM) wherein memory was split across part-based cells. It was argued that keeping the context of each body part independent and representing the output of the P-LSTM unit as a combination of independent body part context information was more efficient. Previous RNN-based 3D-action recognition methods have adopted RNN to model the long-term contextual information in the temporal domain for motion-based dynamics representation. However, there was also strong dependency between joints in the spatial domain. In addition the spatial configuration of joints in video frames can be highly discriminative for 3D-action recognition task. To exploit this dependency, [LSXW16] proposed a spatio-temporal LSTM (ST-LSTM) network which extended the traditional LSTM-based learning to both temporal and spatial domains. Rather than concatenate the joint-based input features, ST-LSTM explicitly modelled the

dependencies between the joints and applied recurrent analysis over spatial and temporal domains concurrently. Besides, they introduced a trust gate mechanism to make LSTM robust to noisy input data. [SLX+17] proposed a spatio-temporal attention model with LSTM to automatically mine the discriminative joints and learn the respective and different attentions of each frame along the temporal axis. Similarly, [LWH+17] proposed a Global Context-Aware Attention LSTM (GCA-LSTM) to selectively focus on the informative joints in the action sequence with the assistance of global context information. Differently from previous works that adopted the coordinates of joints as input, [ZLX17] investigated a set of simple geometric features of skeleton using 3-layer LSTM framework, and showed that using joint-line distances as input requires less data for training.

2.2.3.3 Other-architecture-based Approach

Besides the RNN- and CNN-based approaches, there are several other deep learning-based methods. [STT13] proposed a new compositional learning architecture that integrated deep learning models with structured hierarchical Bayesian models. Specifically, this method learned a hierarchical Dirichlet process (HDP) [TJBB04] prior over the activities of the top-level features in a deep Boltzmann machine (DBM). This compound HDP-DBM model learned novel concepts from very few training examples by learning: (i) low-level generic features, (ii) high-level features that captured correlations among low-level features and, (iii) a category hierarchy for sharing priors over the high-level features that were typical of different kinds of concepts. [WS14a] adopted deep belief networks (DBN) to model the distribution of skeleton joint locations and extract high-level features to represent humans at each frame in 3D space. [I+16] adopted stacked auto encoder to learn the underlying features of input skeleton data. [HWPVG17] incorporated the Lie group structure into a deep learning architecture to learn more appropriate Lie group features for skeleton based action recognition.

2.2.4 Depth-based Approach

Compared with RGB videos, the depth modality is insensitive to illumination variations, invariant to color and texture changes, reliable for estimating body silhouette and skeleton, and provides rich 3D structural information of the scene. However, there are only few published results on depth based action recognition using deep learning methods. Two reasons can be adduced for this situation. First, the absence of color and texture in depth maps weakens the discriminative representation power of CNN models which are texture-driven feature extractor and classifier [LZT16]. Second, existing depth data is relatively small-scale. The conventional pipelines are

purely data-driven and learn representation directly from the pixels. Such model is likely to be at risk of overfitting when the network is optimized on limited training data. Currently, there are only CNN-based methods for depth-based motion recognition.

2.2.4.1 CNN-based Approach

[WLG+15, WLG+16] took advantage of the representation power of CNN on texture images and at the same time enlarge available training data by encoding depth map sequences into texture color images using the concepts of Depth Motion Maps (DMM) [YZT12] and pseudo-coloring; training data was enlarged by scene rotation on the 3D point cloud. Inspired by the promising results achieved by rank pooling method [BFG+16] on RGB data, [WLL+16b] encoded the depth map sequences into three kinds of dynamic images with rank pooling: Dynamic Depth Images (DDI), Dynamic Depth Normal Images (DDNI) and Dynamic Depth Motion Normal Images (DDMNI). These three representations captured the posture and motion information from three different levels for gesture recognition. Differently from the above texture image encoding method, [RM16] proposed a cross-view action recognition based on depth sequence. Their method comprised two steps: (i) learning a general view-invariant human pose model from synthetic depth images and, (ii) modelling the temporal action variations. To enlarge the training data for CNN, they generated the training data synthetically by fitting realistic synthetic 3D human models to real mocap data and then rendering each pose from a large number of viewpoints. For spatio-temporal representation, they used group sparse Fourier Temporal Pyramid which encodes the action-specific discriminative output features of the proposed human pose model.

2.2.5 Multi-modal-based Approach

As discussed in previous sections, RGB, depth and skeleton modalities have their own specific properties, and how to combine the strengths of these modalities with deep learning approach is a vital issue. To address this problem, several methods have been proposed. In general, these methods can be categorized as (i) CNN-based, (ii) RNN-based and (iii) other-architecture-based approaches.

2.2.5.1 CNN-based Approach

[ZZM+16a] fused RGB and depth in a pyramidal 3D convolutional network based on C3D [TBF+15] for gesture recognition. They designed pyramid input and pyramid fusion for each modality and late score fusion was adopted for final recognition. [DZW+16] proposed a convolutional two-stream consensus voting network (2SCVN)

which explicitly modelled both the short-term and long-term structure of the RGB sequences. To alleviate distractions from background, a 3D depth-saliency ConvNet stream (3DDSN) was aggregated in parallel to identify subtle motion characteristics. Later score fusion was adopted for final recognition. The methods described so far considered RGB and depth as separate channels and fused them later. [WLG+17] took a different approach and adopted scene flow to extract features that fused the RGB and depth from the onset. The new representation based on CNN and named Scene Flow to Action Map (SFAM) was used for motion recognition.

2.2.5.2 RNN-based Approach

For RGB and depth fusion, [PVDOD+16] directly considered the depth as the fourth channel and CNN was adopted to extract frame-based appearance features. Temporal convolutions and RNN were combined to capture the temporal information. [LMT+16a] adopted C3D [TBF+15] to extract features separately from RGB and depth modalities, and used the concatenated for SVM classifier. [ZZSS17] presented a gesture recognition method using C3D [TBF+15] and convolutional LSTM (convLSTM) [XCW+15] based on depth and RGB modalities. The major drawback of traditional LSTM in handling spatio-temporal data was its usage of full connections in input-to-state and state-to-state transitions in which no spatial information was encoded. The ConvLSTM determined the future state of a certain cell in the grid by the inputs and past states of its local neighbors. Average score fusion was adopted to fuse the two separate channel networks for the two modalities. [LPH+17] proposed to use a RNN-based encoder-decoder framework to learn a video representation by predicting a sequence of basic motions described as atomic 3D flows. The learned representation was then extracted from the generated model to recognize activities.

[SK17] fused depth and skeleton in a so-called privileged information (PI)-based RNN (PRNN) that exploited additional knowledge of skeleton sequences to obtain a better estimate of network parameters from depth map sequences. A bridging matrix was defined to connect softmax classification loss and regression loss by discovering latent PI in the refinement step.

For RGB and skeleton fusion, [MT16] presented a regularization of LSTM learning where the output of another encoder LSTM (eLSTM) grounded on 3D human-skeleton training data was used as the regularization. This regularization rested on the hypothesis that since videos and skeleton sequences were about human motions their respective feature representations should be similar. The skeleton sequences, being view-independent and devoid of background clutter, were expected to facilitate capturing important motion patterns of human-body joints in 3D space.

2.2.5.3 Other-architecture-based Approach

[SNGW17] extracted hand-crafted features which were neither independent nor fully correlated from RGB and depth, and embedded the input feature into a space of factorized common and modality-specific components. The combination of shared and specific components in input features could be very complex and highly non-linear. In order to disentangle them, they stacked layers of non-linear auto encoder-based component factorization to form a deep shared-specific analysis network.

In a RGB, depth and skeleton fusion method, [WPK+16b] adopted Gaussian-Bernouilli Deep Belief Network(DBN) to extract high-level skeletal joint features and the learned representation is used to estimate the emission probability needed to infer gesture sequences. A 3D Convolutional Neural Network (3DCNN) was used to extract features from 2D multiple channel inputs such as depth and RGB images stacked along the 1D temporal domain. In addition, intermediate and late fusion strategies were investigated in combination with the temporal modeling. The result of both mechanisms indicates that multiple-channel fusion can outperform individual modules.

2.3 Performance Evaluation

2.3.1 Benchmark Datasets

In the past a few years, a large number of RGB-D-based benchmark datasets were collected and made public, containing either RGB, depth, skeleton or their combinations [CYL16, ZLO+16a]. Generally speaking, these datasets were collected mainly from three types of devices [CWF13, CYL16, HRHZ17]: Motion capture (Mocap) system, structured-light cameras (e.g. Kinect v1) and time-of-flight (ToF) cameras (e.g. Kinect v2). This thesis summaries the commonly adopted datasets for evaluation.

2.3.1.1 CMU Mocap

CMU Graphics Lab Motion Capture Database (CMU Mocap) [CMU01](`http://mocap.cs.cmu.edu/`) is one of the earliest resources that consists of wide variety of human actions, including interaction between two subjects, human locomotion, interaction with uneven terrain, sports, and other human actions. It is capable of recording 120 Hz with images of 4 megapixel resolution. This dataset provides RGB and skeleton data.

2.3.1.2 HDM05

Motion Capture Database HDM05 [MRC+07] (http://resources.mpi-inf.mpg.de/HDM05/) was captured by an optical marker-based technology with the frequency of 120 Hz, which contains 2337 sequences for 130 actions performed by 5 non-professional actors, and 31 joints in each frame. Besides skeleton data, this dataset also provides RGB data.

2.3.1.3 MSR-Action3D

MSR-Action3D [LZL10] (http://www.uow.edu.au/~wanqing/#MSRAction3DDatasets) is the first public benchmark RGB-D action dataset collected using Kinect ™sensor by Microsoft Research, Redmond and University of Wollongong in 2010. The dataset contains 20 actions: *high arm wave*, *horizontal arm wave*, *hammer*, *hand catch*, *forward punch*, *high throw*, *draw x*, *draw tick*, *draw circle*, *hand clap*, *two hand wave*, *side-boxing*, *bend*, *forward kick*, *side kick*, *jogging*, *tennis serve*, *golf swing*, *pickup and throw*.

Ten subjects performed these actions three times. All the videos were recorded from a fixed point of view and the subjects were facing the camera while performing the actions. The background of the dataset was removed by some post-processing. Specifically, if an action needs to be performed with one arm or one leg, the actors were required to perform it using right arm or leg.

2.3.1.4 MSRC-12

MSRC-12 dataset [FMNK12](http://research.microsoft.com/en-us/um/cambridge/projects/msrc12/) was collected by Microsoft Research Cambridge and University of Cambridge in 2012. So, there are two types of gestures: Iconic gestures (*Crouch* or *hide*, *Shoot a pistol*, *Throw an object*, *Change weapon*, *Kick*, and *Put on night vision goggles*) and Metaphoric gestures (*Start Music/Raise Volume (of music)*, *Navigate to next menu*, *Wind up the music*, *Take a bow to end music session*, *Protest the music*, and *Move up the tempo of the song*). The authors provided three familiar and easy to prepare instruction modalities and their combinations to the participants. The modalities are (1) descriptive text breaking down the performance kinematics, (2) an ordered series of static images of a person performing the gesture with arrows annotating as appropriate, and (3) video (dynamic images) of a person performing the gesture.

There are 30 participants in total and for each gesture, the data were collected as: Text (10 people), Images (10 people), Video (10 people), Video with text (10 people), Images with text (10 people). The dataset was captured using one Kinect ™sensor and only the skeleton data are made available.

2.3.1.5 MSRDailyActivity3D

MSRDailyActivity3D Dataset [WLWY12](`http://www.uow.edu.au/~wanqing/#MSRAction3DDatasets`) was collected by Microsoft and the Northwestern University in 2012 and focused on daily activities. The motivation was to cover human daily activities in the living room. There are 16 activity types: *drink*, *eat*, *read book*, *call cellphone*, *write on a paper*, *use laptop*, *use vacuum cleaner*, *cheer up*, *sit still*, *toss paper*, *play game*, *lay down on sofa*, *walk, play guitar*, *stand up*, *sit down*.

The actions were performed by 10 actors while sitting on the sofa or standing close to the sofa. The camera was fixed in front of the sofa. In addition to depth data, skeleton data are also recorded, but the joint positions extracted by the tracker are very noisy due to the actors being either sitting on or standing close to the sofa.

2.3.1.6 MSR ActionPairs3D

The MSR ActionPairs3D dataset [OL13] (`http://www.cs.ucf.edu/~oreifej/HON4D.html`) is a paired-activity dataset captured by a Kinect camera. This dataset contains 12 activities (i.e. six pairs) of 10 subjects with each subject performing each activity 3 times. The pair actions are: Pick up a box/Put down a box, Lift a box/Place a box, Push a chair/Pull a chair, Wear a hat/Take off hat, Put on a backpack/Take off a backpack, Stick a poster/Remove a poster.

2.3.1.7 UTKinect

UTKinect dataset [XCA12](`http://cvrc.ece.utexas.edu/KinectDatasets/HOJ3D.html`) was collected by the University of Texas at Austin in 2012. Ten types of human actions were performed twice by 10 subjects. The actions include *walk*, *sit down*, *stand up*, *pick up*, *carry*, *throw*, *push*, *pull*, *wave*, *clap hands*.

The subjects performed the actions from a variety of views. One challenge of the dataset is due to the actions being performed with high actor-dependent variability. Furthermore, human-object occlusions and body parts being out of the field of view have further increased the difficulty of the dataset. Ground truth in terms of action labels and segmentation of sequences are provided.

2.3.1.8 G3D

Gaming 3D dataset (G3D) [BMA12](`http://dipersec.king.ac.uk/G3D/`) captured by Kingston University in 2012 focuses on real-time action recognition in gaming scenario. It contains 10 subjects performing 20 gaming actions: *punch right*, *punch left*, *kick right*, *kick left*, *defend*, *golf swing*, *tennis serve*, *throw bowling ball*, *aim and fire gun*, *walk*, *run*, *jump*, *climb*, *crouch*, *steer a car*, *wave*, *flap*, and *clap*.

Each subject performed these actions thrice. Two kinds of labels were provided as ground truth: the onset and offset of each action and the peak frame of each action.

2.3.1.9 SBU Kinect Interaction Dataset

SBU Kinect Interaction Dataset [YHC+12](`http://www3.cs.stonybrook.edu/~kyun/research/kinect_interaction/index.html`) was collected by Stony Brook University in 2012. It contains eight types of interactions, including: *approaching*, *departing*, *pushing*, *kicking*, *punching*, *exchanging objects*, *hugging*, and *shaking hands*.

All videos were recorded with the same indoor background. Seven participants were involved in performing the activities which have interactions between two actors. The dataset is segmented into 21 sets and each set contains one or two sequences of each action category. Two kinds of ground truth information are provided: action labels of each segmented video and identification of "active" actor and "inactive" actor.

2.3.1.10 RGBD-HuDaAct

RGBD-HuDaAct [NWM13] (`http://adsc.illinois.edu/sites/default/files/files/ADSC-RGBD-dataset-download-instructions.pdf`) was collected by Advanced Digital Sciences Center Singapore in 2011. Compared to MSR-Action3D dataset, this dataset consists of fewer actions (12 actions) and performed by more subjects (30 subjects). The action types are also different from MSR-Action3D dataset. This dataset focuses on human daily activities, such as *make a phone call*, *mop the floor*, *enter the room*, *exit the room*, *go to bed*, *get up, eat meal*, *drink water*, *sit down*, *stand up*, t*ake off the jacket*, and *put on the jacket*. Each actor performed 2-4 repetitions of each action. The background is also fixed as the camera was fixed when recording. However, there was no restriction on which leg or hand was used in the actions and the dataset contains human-object interaction.

2.3.1.11 Berkeley MHAD

Berkeley Multimodal Human Action Database (Berkeley MHAD) [OCK+13](`http://tele-immersion.citris-uc.org/berkeley_mhad#dl`), collected by University of California at Berkeley and Johns Hopkins University in 2013, was captured in five different modalities to expand the fields of application. The modalities are derived from: optical mocap system, four multi-view stereo vision cameras, two Microsoft Kinect v1 cameras, six wireless accelerometers and four microphones. Twelve subjects performed 11 actions, five times each. Three categories of actions are included: (1) actions with movement in full body parts, e.g., *jumping in place*, *jumping jacks*,

throwing, etc., (2) actions with high dynamics in upper extremities, e.g., *waving hands*, *clapping hands*, etc. and (3) actions with high dynamics in lower extremities, e.g., *sit down*, *stand up*. The actions were executed with style and speed variations. This dataset can be used in the development and evaluation of multimodal algorithms.

2.3.1.12 Northwestern-UCLA Multiview Action 3D

Northwestern-UCLA Multiview Action 3D [WNX+14](`http://users.eecs.northwestern.edu/~jwa368/my_data.html`) was collected by Northwestern University and University of California at Los Angles in 2014. This dataset contains data taken from a variety of viewpoints. The actions were performed by 10 actors and captured by three simultaneous Kinect™v1 cameras. There are 10 action categories: *pick up with one hand*, *pick up with two hands*, *drop trash*, *walk around*, *sit down*, *stand up*, *donning*, *doffing*, *throw*, *carry*.

2.3.1.13 UTD-MHAD

UTD Multimodal Human Action Dataset (UTD-MHAD) [CJK15](`http://www.utdallas.edu/~cxc123730/UTD-MHAD.html`) was collected by University of Texas at Dallas in 2015. Eight subjects performed 27 actions four times. The 27 actions are: *right arm swipe to the left, right arm swipe to the right, right hand wave, two hand front clap, right arm throw, cross arms in the chest, basketball shoot, right hand draw x, right hand draw circle (clockwise), right hand draw circle (counter clockwise), draw triangle, bowling (right hand), front boxing, baseball swing from right, tennis right hand forehand swing, arm curl (two arms), tennis serve, two hand push, right hand knock on door, right hand catch an object, right hand pick up and throw, jogging in place, walking in place, sit to stand, stand to sit, forward lunge (left foot forward)*, and *squat (two arms stretch out)*. All the actions were performed in a fixed background. An inertial sensor was worn on the subject's right wrist for action 1 to 21, and on the right thigh for action 22 to 27. Hence, four types of data modalities were captured, namely RGB videos, depth videos, skeleton joint positions, and the inertial sensor signals.

2.3.1.14 M²I Dataset

Multi-modal & Multi-view & Interactive (M²I) Dataset [LXN+16] (`http://media.tju.edu.cn/m2i.html`) provides person-person interaction actions and person-object interaction actions. It contains both the front and side views; denoted as Front View (FV) and Side View (SV). It consists of 22 action categories and a total of 22 unique individuals. Each action was performed twice by 20 groups (two

persons in a group). In total, M^2I dataset contains 1760 samples (22 actions $\times$ 20 groups $\times$ 2 views $\times$ 2 run).

2.3.1.15 SYSU 3D HOI Dataset

The SYSU 3D Human-Object Interaction Dataset (SYSU 3D HOI Dataset) [HZLZ15] (`https://sites.google.com/site/jianfanghusysuntu/`) was collected to focus on human-object interactions. There are 40 subjects performing 12 different activities. For each activity, each participants manipulate one of the six different objects: phone, chair, bag, wallet, mop and besom.

2.3.1.16 ChaLearn LAP IsoGD

ChaLearn LAP IsoGD Dataset [WLZ+16] (`http://www.cbsr.ia.ac.cn/users/jwan/database/isogd.html`) is a large RGB-D dataset for segmented gesture recognition, and it was collected by Kinect v1 camera. It includes 47933 RGB-D depth sequences, each RGB-D video representing one gesture instance. There are 249 gestures performed by 21 different individuals. The dataset is divided into training, validation and test sets. All three sets consist of samples of different subjects to ensure that the gestures of one subject in the validation and test sets will not appear in the training set.

2.3.1.17 NTU RGB+D

NTU RGB+D Dataset [SLNW16](`https://github.com/shahroudy/NTURGB-D`) is currently the largest action recognition dataset in terms of the number of samples per action. The RGB-D data is captured by Kinect v2 cameras. The dataset has more than 56 thousand sequences and 4 million frames, containing 60 actions performed by 40 subjects aging between 10 and 35. It consists of front view, two side views and left, right 45 degree views.

2.3.1.18 ChaLearn LAP ConGD

The ChaLearn LAP ConGD Dataset [WLZ+16] (`http://www.cbsr.ia.ac.cn/users/jwan/database/congd.html`) is a large RGB-D dataset for continuous gesture recognition. It was collected by Kinect v1 sensor and includes 47933 RGB-D gesture instances in 22535 RGB-D gesture videos. Each RGB-D video may contain one or more gestures. There are 249 gestures performed by 21 different individuals. The dataset is divided into training, validation and test sets. All three sets consist of samples of different subjects to ensure that the gestures of one subject in the validation and test sets will not appear in the training set.

2.3.1.19 PKU-MMD

PKU-MMD [CYY+17] (`http://www.icst.pku.edu.cn/struct/Projects/PKUMMD.html`) is a large scale dataset for continuous multi-modality 3D human action understanding and covers a wide range of complex human activities with well annotated information. It was captured via the Kinect v2 sensor. PKU-MMD contains 1076 long video sequences in 51 action categories, performed by 66 subjects in three camera views. It contains almost 20,000 action instances and 5.4 million frames in total. It provides multi-modality data sources, including RGB, depth, Infrared Radiation and Skeleton.

2.3.2 Evaluation Metric

The performance is evaluated using *accuracy* for segmented motion recognition, and *Jaccard Index* is added as another criteria for continuous motion recognition. The accuracy is calculated as:

$$r = \frac{1}{n}\delta(p_l(i), t_l(i)) \tag{2.12}$$

where n is the number of samples; p_l is the predicted label; t_l is the ground truth; $\delta(j_1, j_2) = 1$, if $j_1 = j_2$, otherwise $\delta(j_1, j_2) = 0$. The Jaccard index measures the average relative overlap between true and predicted sequences of frames for a given gesture/action. For a sequence s, let $G_{s,i}$ and $P_{s,i}$ be binary indicator vectors for which 1-values correspond to frames in which the i^{th} gesture/action label is being performed. The Jaccard Index for the i^{th} class is defined for the sequence s as:

$$J_{s,i} = \frac{G_{s,i} \bigcap P_{s,i}}{G_{s,i} \bigcup P_{s,i}}, \tag{2.13}$$

where $G_{s,i}$ is the ground truth of the i^{th} gesture/action label in sequence s, and $P_{s,i}$ is the prediction for the i^{th} label in sequence s. When $G_{s,i}$ and $P_{s,i}$ are empty, $J_{(s,i)}$ is defined to be 0. Then for the sequence s with l_s true labels, the Jaccard Index J_s is calculated as:

$$J_s = \frac{1}{l_s}\sum_{i=1}^{L} J_{s,i}. \tag{2.14}$$

For all test sequences $S = s_1, ..., s_n$ with n gestures/actions, the mean Jaccard Index $\overline{J_S}$ is used as the evaluation criteria and calculated as:

$$\overline{J_S} = \frac{1}{n}\sum_{j=1}^{n} J_{s_j}. \tag{2.15}$$

Chapter 3

Skeleton-based Action Recognition

A common and intuitive method to represent human motion is to use a sequence of skeletons. With the development of the cost-effective depth cameras and algorithms for real-time pose estimation [SFC+11], skeleton extraction has become more and more robust and skeleton-based action representation is becoming one of the most practical and promising approaches. In this chapter, we mainly study the research questions 1 and 2 (Section 1.2) and present two methods for action recognition from skeleton data.

3.1 Mining Frequent and Relevant Features

3.1.1 Prior Works and Our Contributions

For hand-crafted skeleton feature based action recognition, the process can be generally divided into two main steps, action representation and action classification, as described in section 2.1. Several works [YT12, XCA12, GTHES13, HTGES13, ZLS13, WWY13] have been proposed to address skeleton-based action recognition. However, in previous methods, most of them are based low-level features and need the whole skeletal description which leads to their weak adaptation to noise. In addition, most of them need to explore the spatial and temporal information, separately, and then combine them together. Besides, most of the methods used to explore temporal information are subject to the neural poses, which are shared by all actions. Inspired by the mid-level features mining techniques [FFT14] for image classification, we propose a new scheme applying pattern mining to obtain the most relevant combinations of parts in several continuous frames for action recognition rather than to utilize all the joints as most previous works did. In particular, a new descriptor called *bag-of-FLPs* is proposed to describe an action as illustrated in Fig. 3.1. The overall process of our method can be divided into four steps: feature extraction, building transactions, mining & selecting relevant patterns and building *Bag-of-FLPs* & classification. We first compute the orientations of limbs, i.e. connected joints, and then encode each orientation into one of the 27 states indicating the spatial relationship of the joints. Limbs are combined into parts and limb's states are mapped to part states. Local temporal information is included by combining part states of several, say, 5, continuous frames into one transaction for mining, with each state as one item. In order to keep motion information after

frequent pattern mining, the unique states of parts of the continuous frames are reserved, removing the repeated ones, ensuring the pose information and motion information be included in each transaction. The most relevant patterns, which we referred to *FLPs*, are mined and selected o represent frames and build *bag-of-FLPs* as new representation for a whole action. The new representation is much robust to the errors in the features, because the errors are usually not frequent patterns.

Our main contributions include the following four aspects. First, an effective and efficient method is proposed to extract skeleton features. Second, a novel method is developed to explore spatial and temporal information in skeleton data, simultaneously. Third, an effective scheme is proposed for applying pattern mining to action recognition by adapting the generic pattern mining tools to the features of skeleton. Our scheme is much robust to noise as most noisy data does not form frequent patterns. In addition, our scheme has achieved the state-of-the-art results on several benchmark datasets.

3.1.2 The Proposed Methods

The overall process of the proposed method is illustrated in Fig. 3.1. It can be divided into four steps: feature extraction, building transactions, mining & selecting relevant patterns and building *Bag-of-FLPs* & classification.

3.1.2.1 Feature Extraction

In our method, the orientations of human body limbs are considered as low-level features and they can be calculated from the two joints of the limbs. For Kinect skeleton data, 20 joint positions, as shown in Fig. 3.2, are tracked [SFC+11]. The skeleton data is first normalized using Algorithm 1 in [ZLS13] to suppress noise in the original skeleton data and to compensate for length variations across different subjects and different body parts. Each joint i has 3 coordinates, denoted as (x_i, y_i, z_i) after normalization.

For Kinect skeleton, it is found that the correct positions of *Hand Left*, *Hand Right*, *Foot Left*, *Foot Right*, and *Spine* joints are highly correlated with their neighbouring joints and dropping them will not lead to any loss of information, hence they are not used in our method. Thus, there are 15 joints 14 limbs (the connection between two adjacent joints). The joint Head is considered as the origin of the 15 points. For each limb, we compute a unit difference vector between its two joints:

$$(\Delta x_{ij}, \Delta y_{ij}, \Delta z_{ij}) = \frac{(x_i, y_i, z_i) - (x_j, y_j, z_j)}{d_{ij}} \tag{3.1}$$

where i and j represent the current joint and reference joint, respectively; d_{ij} is

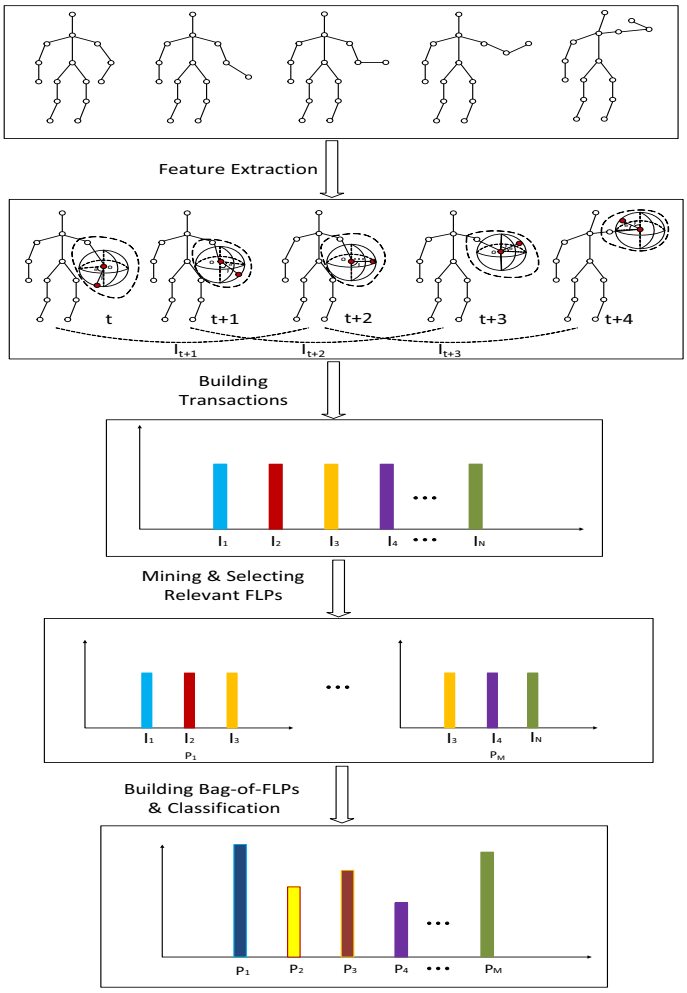

Figure 3.1: The general framework of the proposed method to mine frequent and relevant features for action recognition.

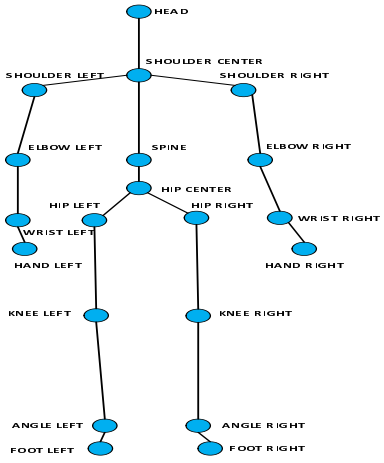

Figure 3.2: The human joints tracked with the skeleton tracker [SFC+11].

the Euclidean distance between the two joints. This representation is much robust against body/camera-view rotation as the difference between joints is not so sensitive

to rotation compared with the joint positions themselves. For example, as illustrated in Fig. 3.1, to compute the orientation of the limb between joint Hand Right and Wrist Right (highlighted in red), the Wrist Right joint is regarded as the sphere center and Eq. 3.1 is used to compute the unit difference vector.

Each element of the unit difference vector is quantized into three states: -1, 0 and 1. If $|\Delta x_{ij}| \leq threshold$ then $q(\Delta x_{ij}) = 0$; if $\Delta x_{ij} > threshold$ then $q(\Delta x_{ij}) = 1$; else $q(\Delta x_{ij}) = -1$. Thus, there are 27 possible states for each unit difference vector, and each state is encoded as one element of a feature vector, so the dimension of the feature vector for each pose is $14 \times 27 = 378$ after concatenating all feature vectors of the 14 limbs. For each element of the feature vector, if the corresponding orientation between two joints is bid to one state, then the relative position is labelled to 1, otherwise, it is 0. Therefore, the feature vectors are very sparse, only 14 positions in each feature vector are 1 (not zeros). The threshold is an empirical value which is dependent on the noise characteristics of the skeleton data.

For each frame of skeleton, a quantized 378 dimensional feature vector is calculated as described above. This feature vector is reduced to a 14 dimensional feature vector with each element being the index to a non-zero element of the 378-dimensional feature vector.

To extract mid-level features for action representation, the 14 limbs are combined into 7 body parts. As illustrated in Fig. 3.1, the dotted line contains joints Hand Right, Wrist Right and Elbow Right, and these three limbs form one part. In this way, seven body parts are formed, namely, Head-Shoulder Center, Should Center-Shoulder Left-Elbow Left-Wrist Left, Shoulder Center-Shoulder Right-Elbow Right-Wrist Right, Shoulder Center-Hip Center-Hip Left, Hip Left-Knee Left-Angle Left,Shoulder Center-Hip Center-Hip Right and Hip-Right-Knee Right-Angle Right. According to the Degree of Freedom (DoF) of joints [Zat98], each body part is encoded with different number of states and the total number of states is denoted as NDF, which is currently an empirical parameter. It should be adjusted according to the complexity of the actions to be recognized and noise level of the dataset.

To explore temporal information and keep motion information at the same time after frequent data mining (generally, frequent data mining can only mine the most frequent pattens which can not be guaranteed as discriminative patterns), a novel way is proposed. Seven states for each frame will be obtained after combination, and the unique states of continuous C frames, as illustrated in Fig. 3.1, where $C = 3$, are counted and form a new mid-level feature vector, denoted as $\{f_i | i = 1, ..., n_A\}$. This new feature vector contains both pose information of the current frame and the motion information in the continuous C frames, because the repeated states in the continuous frames can be regarded as static pose information and the different ones with other frames can capture the motion information. This feature vector is

used to build transactions described in the next section. The pattens after mining can be the combinations of several body parts in different frames, thus the temporal order information can be easily maintained.

3.1.2.2 Building Transactions

Each instance of action A is represented by a set of above mid-level features $\{f_i|i = 1, ..., n_A\}$ and a class label c, $c \in \{1...C\}$. The set of features for all the action samples is denoted by Ω. The dimensionality of the feature vector is denoted as W and in our case $|\mathrm{W}| \geq 7$.

Items, Transactions and Frequencies

Each element in a feature vector for continuous C poses is defined as an item, and an item is denoted as ω, where $\omega \in (0, NDF]$ and $\omega \in \mathbb{N}$.

The set of *transactions* X from the set Ω is created next. For each $\mathbf{x} \in \Omega$ there is one transaction x (i.e. a set of items). This transaction x contains all the items ω_j. A *local pattern* is an itemset $t \subseteq \Gamma$, where Γ represents the set of all possible items. For a local pattern t, the set of transactions that include the pattern t is defined as: $X(t) = \{x \in X|t \subseteq x\}$. The *frequency* of t is $|X(t)|$, also known as the *support* of the pattern t or $supp(t)$.

Frequent Local Part

For a given constant T, also known as the minimum support threshold, a local pattern t is *frequent* if $supp(t) \geq T$. A pattern t is said to be *closed* if there exists no pattern $t^{'}$ that $t \subset t^{'}$ and $supp(t) = supp(t^{'})$. The set of frequent closed patterns is a compact representation of the frequent patterns, and such a frequent and closed local part pattern is referred to as *Frequent Local Part* of *FLP*.

3.1.2.3 Mining & Selecting Relevant FLPs

FLPs Mining

Given the set of transaction X, any existing frequent mining algorithm can be used to find the set of *FLPs* Υ. In our work, the optimised *LCM* algorithm [UAUA03] is used as in [FFT14]. *LCM* uses a *prefix preserving closure extension* to completely enumerate closed itemsets.

Encoding a New Action with FLPs

Given a new action, the features can be extracted according to the section A and each feature vector can be converted into a transaction x and for each *FLP* pattern

$t \in \Upsilon$ it can be checked whether $t \subseteq x$. If $t \subseteq x$ is true, then x is an *instance* of the *FLP* pattern t. The frequency of a pattern t in a given action A_j (i.e. the number of instances of t in A_j) is denoted as $F(t|A_j)$.

Selecting the Best FLPs for Action Recognition

The *FLPs* set Υ is considered as a candidate set of mid-level features to represent an action. Therefore, the most useful *FLP* patterns from Υ is needed to be selected because *i*) the number of generated *FLP* patterns is huge and *ii*) not all discovered *FLP* patterns are equally important to the action recognition task. Usually, relevant patterns are those *discriminative* and *non-redundant*. On top of that, a new criterion, *representativity* is also used. As a result, some patterns may be frequent and appear to be discriminative but they may occur in very few actions (e.g. noise pose). Such features are not representative and therefore not the best choice for action recognition. A good *FLP* pattern should be at the same time discriminative, representative and non-redundant. In this section, how to select such patterns is discussed.

The methods used in [FFT14] are followed to find the most suitable pattern subset χ, where $\chi \subset \Upsilon$, for action recognition. To do this the *gain* of a pattern t is denoted by $G(t)$ (s.t. $t \notin \chi$ and $t \in \Upsilon$) and defined as follows:

$$G(t) = S(t) - \max_{s \in \chi} \{R(s,t) \cdot \min(S(t), S(s))\} \tag{3.2}$$

where $S(t)$ is the overall relevance of a pattern t and $R(s,t)$ is the redundancy between two patterns s, t. In Eq 3.2, a pattern t has a higher gain $G(t)$ if it has a higher relevance $S(t)$ (i.e. it is discriminative and representative) and if the pattern t is non redundant with any pattern s in set χ (i.e. $R(s,t)$ is small). $S(t)$ is defined as:

$$S(t) = D(t) \times O(t), \tag{3.3}$$

and $R(s,t)$ is defined as:

$$\begin{aligned} R(s,t) = \exp\{-[&p(t) \cdot D_{KL}(p(A|t)||p(A|\{t,s\})) \\ &+ p(s) \cdot D_{KL}(p(A|s)||p(A|\{t,s\}))]\}. \end{aligned} \tag{3.4}$$

Following a similar approach in [YCHX05] to find affinity between patterns, two patterns t and $s \in \Upsilon$ are redundant if they follow similar document distributions, i.e. if $p(A|t) \approx p(A|s) \approx p(A|\{t,s\})$ where $p(A|\{t,s\})$ is the document distribution given both patterns $\{t,s\}$.

In Eq. 3.3, $D(t)$ is the *discriminability score*. Following the entropy-based approach in [CYHH07], and a high value of $D(t)$ implies that the pattern t occurs

only in very few actions; $O(t)$ is the *representativity score* for a pattern t and it considers the divergence between the optimal distribution for class c $p(A|t_c^*)$ and the distribution for pattern t $p(A|t)$, and then takes the best match over all classes. The optimal distribution is such that i) the pattern occurs only in actions of class c, i.e. $p(c|t_c^*) = 1$ (giving also a discriminability score of 1), and ii) the pattern instances are equally distributed among all the actions of class c, i.e. $\forall A_j, A_k$ in class c, $p(A_j|t_c^*) = p(A_k|t_c^*) = (1/N_c)$ where N_c is the number of samples of class c. An optimal pattern, denoted by t_c^* for class c, is a pattern which has above two properties.

The *discriminability score* and *representativity score* are defined as:

$$D(t) = 1 + \frac{\sum_c p(c|t) \cdot \log p(c|t)}{\log C}, \tag{3.5}$$

$$O(t) = \max_c(\exp\{-[D_{KL}(p(A|t_c^*)||p(A|t))]\}) \tag{3.6}$$

where $p(c|t)$ is the probability of class c given the pattern t, computed as follows:

$$p(c|t) = \frac{\sum_{j=1}^{N} F(t|A_j) \cdot p(c|A_j)}{\sum_{j=1}^{N} F(t|A_j)}; \tag{3.7}$$

$D_{KL}(.||.)$ is the Kullback-Leibler divergence between two distributions; $p(A|t)$ is computed empirically from the frequencies $F(t|A_j)$ of the pattern t:

$$p(A|t) = \frac{F(t|A)}{\sum_j F(t|A_j)} \tag{3.8}$$

Here, A_j is the j^{th} action and N is the total number of actions in the dataset. $p(c|A) = 1$ if the class label of A_j is c and 0 otherwise; $p(c|t_c^*)$ is the optimal distribution with respect to a class c.
In Eq. 3.4, $p(t)$ is the probability of pattern t and it is defined as:

$$p(t) = \frac{\sum_{A_j} F(t|A_j)}{\sum_{t_j \in \Upsilon} \sum_{A_j} F(t_j|A_j)} \tag{3.9}$$

while $p(A|\{t,s\})$ is the document distribution given both patterns $\{t,s\}$ and it is defined as:

$$p(A|\{t,s\}) = \frac{F(t|A) + F(s|A)}{\sum_j F(t|A_j) + F(s|A_j)} \tag{3.10}$$

To find the best K patterns the following greedy process is used. First the most relevant pattern is added to the relevant pattern set χ. Then the pattern with the highest gain (non redundant but relevant) is searched out and this pattern is added

into the set χ until K patterns are added (or until no more relevant patterns can be found). For more detailed discussions, [FFT14] is recommended to refer to.

3.1.2.4 Building Bag-of-FLPs & Classification

After computing the K most relevant and non-redundant *FLPs*, each action can be represented by a new representation called *bag-of-FLPs* by counting the occurrences of such *FLPs* in the action. Let L be such a *bag-of-FLPs* for action A_L and M be the *bag-of-FLPs* for action A_M.

An SVM [CL11] is trained to classify the actions. The SVM uses the following kernel to calculate the similarities between the *bag-of-FLPs* of L and M.

$$K(L, M) = \sum_i \min(\sqrt{L(i)}, \sqrt{M(i)}) \tag{3.11}$$

Here $L(i)$ is the frequency of the i^{th} selected pattern in histogram L. It is a standard histogram intersection kernel with non-linear weighting. This reduces the importance of highly frequent patterns and is necessary since there is a large variability in pattern frequencies.

3.1.3 Experimental Results

Two benchmark datasets, MSR-DailyActivity3D [WLWY12] and MSR-ActionPairs3D [OL13], were used to evaluate the proposed method and the results are compared with those reported in other papers on the same datasets and under the same training and testing configuration.

3.1.3.1 Experimental Setup

In our method, there are several parameters that need to be tuned, the *threshold* T, the number of states NDF, the number of relevant patterns K, the continuous frames C, minimum support S and maximum support U. For different datasets, different sets of parameters were learned through cross-validation to optimize the performance. Specifically, two-third of the entire training dataset was used as training and the rest one-third was used for validation to tune the parameters. The ranges of the parameters are empirical. In general, the threshold T is dependent on the noise level of the dataset. The higher the noise the larger its value. This is an important parameter because it affects the states of limbs computed from the skeleton data. However, such sensitivity can be reduced by setting a large number, NDF (i.e. over 600) of states. The number of relevant patterns K is dependent on the complexity of the actions to be recognized, the more actions in the dataset, the larger number it should be. The number of continuous frames C is affected

by the complexity of required temporal information to encode the actions. If the dataset has pair actions, for example, two actions of each pair are similar in motion (have similar trajectories) and shape (have similar objects), the value of C should be large. However, a large C leads to high memory and post-processing requirement. The values of the minimum support S and maximum support U effect the number of generated patterns before pattern selection. We observed that if S is large, U should also be large; If S is small, U should also be small. Generally, S and U are set to reduce the computational time for post-processing. In fact, there are many combinations of these two parameters to get the best results. In the other words, the performance of the proposed method is not much sensitive to the choice of S and U.

3.1.3.2 MSR DailyActivity3D

The MSR DailyActivity3D dataset was adopted to evaluate the proposed method. This dataset has large intra-class variations and involves human-object interactions, which is challenging for recognition only by 3D joints. Experiments were performed based on cross-subject test setting described in [ZLS13], i.e. five subjects (1, 2, 3, 4, 5) were used for training and the rest 5 subjects were used for testing. Table 3.1 shows the results of our methods compared with other published results.

Table 3.1: Comparison on MSR-DailyActivity dataset.

Methods	Accuracy (%)
Dynamic Temporal Warping [MR06]	54.0
Moving Pose [ZLS13]	73.8
Actionlet Ensemble on Joint Features [WLWY14]	74.0
Proposed Method	78.8

For this dataset, $T = 0.15, NDF = 600, K = 30000, C = 3, S = 15, U = 180$. As seen, although this dataset is quite challenging, our method obtained promising results based only on skeleton data. The confusion matrix is illustrated in Fig. 3.3. From the confusion matrix, it can be seen that activities such as “Drink”, “Cheer Up”, “Sit Still”, “Toss Paper” are relatively easy to recognise, while “Eat” and “Use laptop” are relatively difficult to recognise. The reason for the difficulties is that for these human-object interactions, object information was not available from skeleton data which makes these interactions are almost the same in terms of motion reflected in the skeleton data.

3.1.3.3 MSR ActionPairs3D

The MSR ActionPairs3D dataset [OL13] was adopted to evaluate the proposed method. This dataset is collected to investigate how the temporal order affects

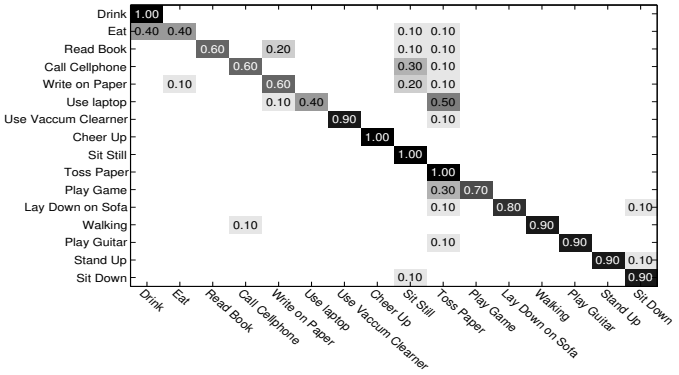

Figure 3.3: The confusion matrix of our proposed method for MSR-DailyActivity3D.

activity recognition. Experiments were set to the same configuration as [OL13],

Table 3.2: Comparison on MSR-ActionPairs dataset.

Methods	Accuracy (%)
Skeleton + LOP [WLWY12]	63.33
Depth Motion Maps [YZT12]	66.11
Proposed Method	75.56

namely, the first five actors are used for testing, and the rest for training. For this dataset, $T = 0.11, NDF = 1000, K = 10000, C = 4, S = 3, U = 100$. We compare our performance in this dataset with two methods whose results were reported in [OL13]. Table 3.2 shows the comparisons with other methods tested on this dataset.

The confusion matrix is shown in Fig. 3.4. From the confusion matrix, it can

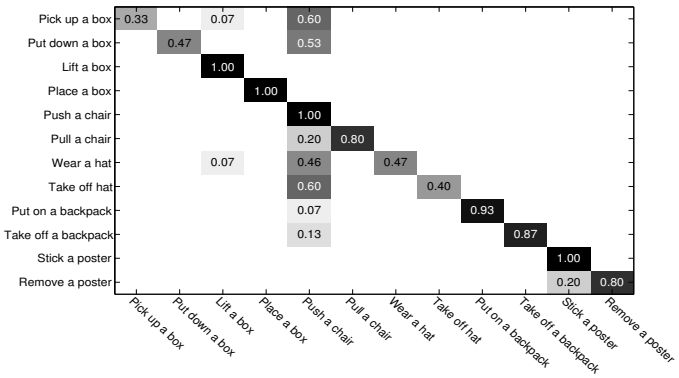

Figure 3.4: The confusion matrix of our proposed method for MSR-ActionPairs3D.

be seen that activities such as “Lift a box”, “Place a Box”, “Push a Chair”, “Stick a Poster” are easy for our method to recognise, while “Pich up a Box” and “Take off Hat” are relatively difficult to recognise. The results have verified that our method

can distinguish temporal orders in actions, however, it still can be confused with other actions which were not paired. One possible reason for causing the confusion between some actions, for instance, "Pick up a Box" and "Push a Chair", is the 3-state quantization of the unit different vectors. This issue can be addressed by quantizing the vector into more states.

3.2 Joint Trajectory Maps with ConvNets

3.2.1 Prior Works and Our Contributions

As the extraction of skeletons from depth maps [SFC+11] has become increasingly robust, more and more hand-designed skeleton features, such as skeleton joints based methods [YT12, XCA12, ZLS13, VAC16, VC16], group joints based methods [WLO+14, COK+13, YDT+16] and joint dynamics based methods [GTHES13, VAC14, SL13, DWB+15], have been devised to capture spatial information, and Dynamic Time Warpings (DTWs), Fourier Temporal Pyramid (FTP) or Hidden Markov Models (HMMs) are employed to model temporal information. However, these hand-crafted features are often either shallow, dataset-dependent, or not learned in an end-to-end fashion [KTF16]. Recently, Recurrent Neural Networks (RNNs) [DWW15, VZQ15, ZLX+16, SLNW16, LSXW16] have also been adopted for action recognition from skeleton data. RNNs tend to overemphasize the temporal information especially when the training data is not sufficient, leading to overfitting. Up to date, it remains unclear how skeleton sequences could be effectively represented and fed to deep neural networks for recognition. For example, one can conventionally consider a skeleton sequence as a set of individual frames with some form of temporal smoothness, or as a subspace of poses or pose features, or as the output of a neural network encoder. Which one among these and other possibilities would result in the best representation in the context of action recognition is not well understood.

In proposed method, we present an effective yet simple method that represent both spatial configuration and dynamics of joint trajectories into three texture images through color encoding, referred to as Joint Trajectory Maps (JTMs), as the input of ConvNets for action recognition. Such image-based representation enables us to fine-tune existing ConvNets models trained on ImageNet for classification of skeleton sequences without training the whole deep networks afresh. The three JTMs are complimentary to each other, and the final recognition accuracy is improved largely by a late score fusion method. One of the challenges in action recognition is how to properly model and use the spatio-temporal information. The commonly used bag-of-words model often ignores temporal information. On the other hand,

HMMs or RNNs based methods are likely to overstress the temporal information. The proposed method addresses this challenge in a novel way by encoding as much the spatio-temporal information as possible (without a need to decide which one is important and how important it is) into images, and employing ConvNets to learn the discriminative one. Consequently, the proposed method outperformed the start-of-the-art methods on popular benchmark datasets. The main contributions of proposed method include:

- A compact, effective yet simple image-based representation is proposed to represent the spatio-temporal information carried in the $3D$ skeleton sequences into three $2D$ images by encoding the dynamics of joint trajectories into three complementary Joint Trajectory Maps.
- To overcome the drawbacks of ConvNets not being rotation-invariant, and to make the proposed method suitable for cross-view action recognition, it is proposed to rotate the skeleton data to not only mimic the multiple views but also to augment data effectively for training.
- The proposed method was evaluated on four popular public benchmark datasets, namely, the large NTU RGB+D Dataset [SLNW16], MSRC-12 Kinect Gesture Dataset (MSRC-12) [FMNK12], G3D Dataset [BMA12] and UTD Multimodal Human Action Dataset (UTD-MHAD) [CJK15], and achieved the state-of-the-art recognition results.

3.2.2 The Proposed Methods

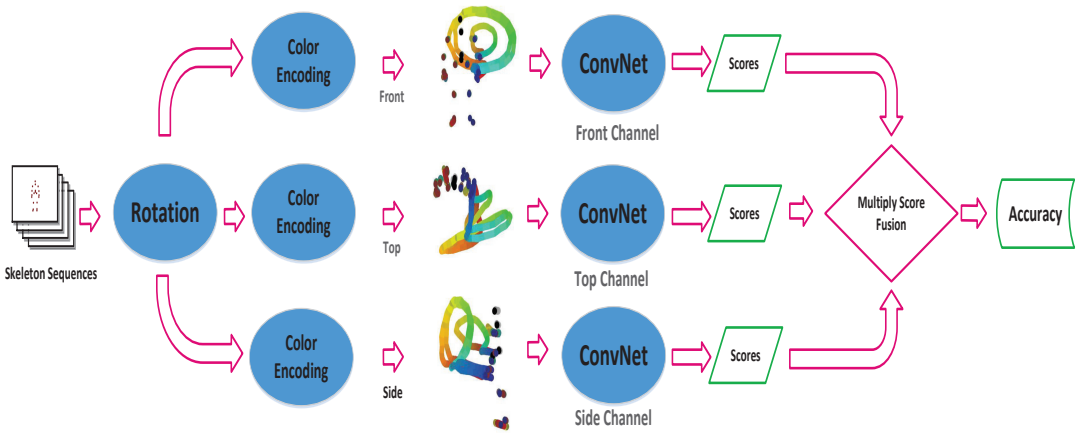

Figure 3.5: The framework of the proposed method.

The proposed method consists of four major components, as illustrated in Fig. 3.5, rotation to mimic the multiple views, construction of three JTMs as the

input of the ConvNets in three orthogonal planes from skeleton sequences, training the three ConvNets to learn discriminative features, and product score fusion for final classification. In the following sections, the four components are detailed.

3.2.2.1 Rotation

A skeleton is often represented by a set of joints in 3D space with respect to the real-world coordinate system centered at the optical central of the RGB-D camera. By rotating the skeleton data, it can 1) mimic multi-views for cross-view action recognition; 2) enlarge the data for training and overcome the drawback of ConvNets usually being not view-invariant.

The rotation was performed with a fixed step of 15° along the polar angle θ and azimuthal angle ψ, in the range of $[0°, 45°]$ for θ and $[-45°, 45°]$ for ψ. The ranges of θ and ψ would cover the possible views considering that the JTMs are generated by projecting the trajectories onto the three orthogonal planes as detailed below.

Let $\mathbf{T}r_y$ be the transform around y axis (right-handed coordinate system) and $\mathbf{T}r_x$ be the transform around x axis. The coordinates (x_r, y_r, z_r) of a joint at (x, y, z) after rotation can be expressed as

$$[x_r, y_r, z_r, 1]^{\mathrm{T}} = \mathbf{T}r_y \mathbf{T}r_x \left[x, y, z, 1\right]^{\mathrm{T}} \tag{3.12}$$

where

$$\mathbf{T}r_y = \begin{bmatrix} R_y(\psi) & T_y(\psi) \\ \mathbf{0} & 1 \end{bmatrix}; \mathbf{T}r_x = \begin{bmatrix} R_x(\theta) & T_x(\theta) \\ \mathbf{0} & 1 \end{bmatrix}, \tag{3.13}$$

and

$$\mathbf{R}_y(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\psi) & -\sin(\psi) \\ 0 & \sin(\psi) & \cos(\psi) \end{bmatrix} \mathbf{T}_y(\psi) = \begin{bmatrix} 0 \\ z \cdot \sin(\psi) \\ z \cdot (1 - \cos(\psi)) \end{bmatrix}; \mathbf{R}_x(\theta) = \begin{bmatrix} cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}$$

$$\mathbf{T}_x(\theta) = \begin{bmatrix} -z \cdot \sin(\theta) \\ 0 \\ z \cdot (1 - \cos(\theta)) \end{bmatrix}.$$

3.2.2.2 Construction of JTMs

We argue that an effective JTM should have the following properties to keep the spatial-temporal information of an action:

- The joints or group of joints should be distinct in the JTM such that the spatial information of the joints is well reserved.
- The JTM should encode effectively the temporal evolution, i.e. trajectories of the joints, including the direction and speed of joint motions.

- The JTM should be able to encode the difference in motion among the different joints or parts of the body to reflect how the joints are synchronized during the action.

Specifically, a JTM can be recursively defined as follows

$$JTM_i = JTM_{i-1} + f(i), \tag{3.14}$$

where $f(i)$ is a function encoding the spatial-temporal information at frame or time-stamp i. Since a JTM is accumulated over the period of an action, $f(i)$ has to be carefully defined such that the JTM for an action sample has the required properties discussed above and the accumulation over time has little adverse impact on the spatial-temporal information that has already been encoded in the JTM. This chapter proposes to use HSB(hue, saturation and brightness) color space to encode the spatial-temporal motion patterns. There are two main reasons to choose HSB color space in this chapter. First, unlike the widely used RGB color space, HSB color space separates *luminance* (image intensity) from *chrominance* (color information). This enables us to selectively encode the information into either luminance or chrominance channels. For instance, anything subtle can be encoded into the brightness component as texture. Second, HSB as a perceptual color space is more intuitive and perceptually relevant than RGB color space or even YUV or YCrCb color spaces though they also separate the luminance from the chominance.

(1) (2) (3)

Figure 3.6: The trajectories projected onto three Cartesian planes for action "right hand draw circle (clockwise)" in UTD-MHAD [CJK15]: (1) front plane; (2) top plane; (3) side plane.

Joint Trajectory Maps

Assume an action H has n frames of skeletons and each skeleton consists of m joints. The skeleton sequence is denoted as $H = \{F_1, F_2, ..., F_n\}$, where $F_i = \{P_1^i, P_2^i, ..., P_m^i\}$ is a vector of joint coordinates of frame i, and P_j^i is the $3D$ coordinates of the j^{th} joint in frame i. The skeleton trajectory T for an action of n frames consists of the

trajectories of all joints and is defined as:

$$T = \{T_1, T_2, \cdots, T_i, \cdots, T_{n-1}\}, \tag{3.15}$$

where $T_i = \{t_1^i, t_2^i, ..., t_m^i\} = F_{i+1} - F_i$, and the k^{th} joint trajectory is $t_k^i = P_k^{i+1} - P_k^i$. A simple form of function $f(i)$ would be T_i, that is,

$$f(i) = T_i = \{t_1^i, t_2^i, ..., t_m^i\}. \tag{3.16}$$

The skeleton trajectory is projected to three orthogonal planes, i.e. three Cartesian planes of the real world coordinates of the camera, to form three JTMs. Fig. 3.6 shows the three projected trajectories of the right hand joint for action "right hand draw circle (clockwise)" in the UTD-MHAD dataset. It can be seen that the spatial information of this joint over the period of the action is well represented in the JTMs but the direction of the motion is lost.

Encoding Joint Motion Direction

To capture the motion direction in the JTM, it is proposed to use hue to "color" the joint trajectories over the action period. Different colormaps may be chosen. In this chapter, the jet colormap, ranging from blue to red, and passing through the colors cyan, yellow, and orange, is adopted. Let the color of a joint trajectory be C, and the length of the trajectory be L, and $C_l, l \in (0, L)$ be the color at position l of a trajectory. For the q^{th} trajectory T_q from 1 to $n-1$, a color C_l, where $l = \frac{q}{n-1} \times L$ is assigned to location l of the joint trajectory, making the entire trajectory colored over the period of the sequence as illustrated in Fig. 3.7. Herein, a trajectory with color is denoted as $C\_t_k^i$ and the function $f(i)$ becomes:

$$f(i) = \{C\_t_1^i, C\_t_2^i, ..., C\_t_m^i\}. \tag{3.17}$$

Fig. 3.8 shows the front JTM of action "right hand draw circle (clockwise)" in the UTD-MHAD [CJK15] dataset. Sub-figure (1) is joint trajectories and sub-figure (2) is the trajectories with motion direction being encoded with hue. The color variations along the trajectories represent the motion direction.

Encoding Body Parts

Many actions, especially complex actions, often involve multiple body parts and these body parts move in a coordinating manner. It is important to capture such coordination in the JTMs. To distinguish different body parts, multiple colormaps are employed. Body parts can be defined at different levels of granularity. For

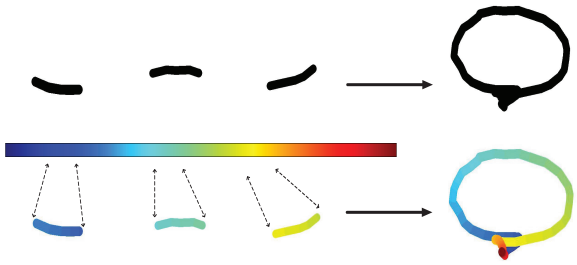

Figure 3.7: An example of colored coded joint trajectory with different colors reflecting the temporal order.

example, each joint can be considered independently as a "part" and is assigned to one colormap, or several groups of joints can be defined and all joints in each group are assigned to the same colormap and colormaps are chosen randomly to each group. Since arms and legs often move more than other body parts, a body is divided into three parts in this chapter. According to the joint configuration for Kinect V1 skeleton as shown in Fig. 3.2, the left body part consists of left shoulder, left elbow, left wrist, left hand, left hip, left knee, left ankle and left foot, the right body part consists of right shoulder, right elbow, right wrist, right hand, right hip, right knee, right ankle and right foot and the middle part consists of head, neck, torso and hip center. The three parts are assigned to three colormaps $(C1, C2, C3)$ respectively, where $C1$ is the same as C, i.e. the jet colormap, $C2$ is a colormap with reversely-ordered colors of $C1$, and $C3$ is a gray-scale map ranging from light gray to black. Let the trajectory encoded by multiple colormaps be $MC\_t_k^i$. Function $f(i)$ can be expressed as:

$$f(i) = \{MC\_t_1^i, MC\_t_2^i, ..., MC\_t_m^i\}. \tag{3.18}$$

The effect of encoding body parts with different colors for action "right hand draw circle (clockwise)" is illustrated in Fig. 3.8, sub-figure (3).

Encoding Motion Magnitude

Motion magnitude is one of the important factors in human motion. For an action, large magnitude of motion is likely to carry discriminative information. This chapter proposes to encode the motion magnitude of joints into saturation and brightness so that the changes in motion would result in texture in the JMTs. Such texture is expected to be beneficial for ConvNets to learn discriminative features. For joints with high motion magnitude or speed, high saturation will be assigned. Specifically,

the saturation is set to range from s_{min} to s_{max}. Given a trajectory, its saturation S_j^i along the path of the trajectory could be calculated as

$$S_j^i = \frac{v_j^i}{max\{v\}} \times (s_{max} - s_{min}) + s_{min} \tag{3.19}$$

where v_j^i is the speed of jth joint at the ith frame.

$$v_j^i = \|P_j^{i+1} - P_j^i\|_2 \tag{3.20}$$

Let a trajectory modulated by saturation be $MC_s\_t_k^i$, function $f(i)$ is refined as:

$$f(i) = \{MC_s\_t_1^i, MC_s\_t_2^i, ..., MC_s\_t_m^i\} \tag{3.21}$$

To further enhance the motion patterns in the JTMs, the brightness is modulated by the speed of joints. Given a trajectory t_j^i whose speed is v_j^i, its brightness B_j^i is computed as

$$B_j^i = \frac{v_j^i}{max\{v\}} \times (b_{max} - b_{min}) + b_{min} \tag{3.22}$$

where b_{min} and b_{max} represent the range of the brightness. Let $MC_b\_t_k^i$ be the trajectory with brightness and function $f(i)$ is then updated to:

$$f(i) = \{MC_b\_t_1^i, MC_b\_t_2^i, ..., MC_b\_t_m^i\}. \tag{3.23}$$

Finally, let $MC_{sb}\_t_k^i$ be the trajectory after encoding the motion magnitude into both saturation and brightness. Function $f(i)$ can be expressed as:

$$f(i) = \{MC_{sb}\_t_1^i, MC_{sb}\_t_2^i, ..., MC_{sb}\_t_m^i\}. \tag{3.24}$$

For the sample example in Fig. 3.8, the encoding effect can be seen in the subfigures (4), where the slow motion becomes diluted (e.g. trajectory of knees and ankles) while the fast motion becomes saturated (e.g. the green part of the circle), and the texture becomes apparent (e.g. the yellow parts of the circle).

3.2.2.3 ConvNets Training

After constructing the three JTMs on three orthogonal image planes, three ConvNets are fine-tuned individually, each ConvNet is initialized with the same AlexNet [KSH12] network parameters with only change made to the last fully connection layer by adjusting the the number of output to the number of classes of current tasks. In this chapter, the network is fine-tuned on all the evaluated datasets.

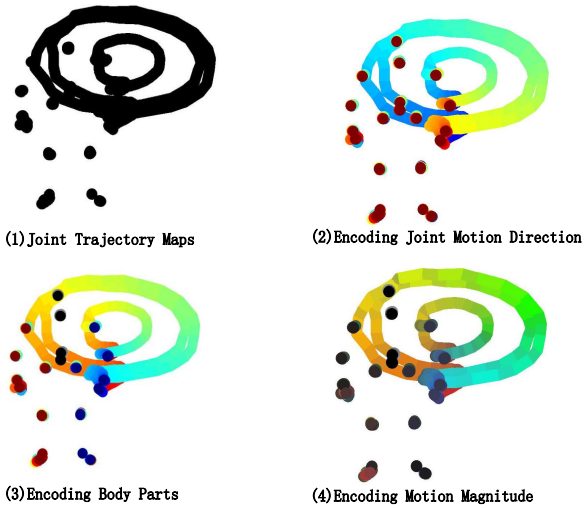

Figure 3.8: Step-by-step illustration of the front JTM for action "right hand draw circle (clockwise)" from the UTD-MHAD [CJK15] dataset. (1) Joint trajectory map without encoding any motion direction and magnitude; (2) encoding joint motion direction in hue, where color variations indicate motion direction; (3) encoding body parts with different colormaps; (4) encoding motion magnitude into saturation and brightness.

In fine-tuning, the pre-trained models on ILSVRC-2012 (Large Scale Visual Recognition Challenge 2012, a version of ImageNet) are used for initialization. The network weights are learned using the mini-batch stochastic gradient descent with the momentum being set to 0.9 and weight decay being set to 0.0005. All hidden weight layers use the rectification (RELU) activation function. At each iteration, a mini-batch of 256 samples is constructed by sampling 256 shuffled training samples. The images are resized to 256×256. The learning rate is set to 10^{-2} for training from scratch and set to 10^{-3} for fine-tuning with pre-trained models on ILSVRC-2012, and then it is decreased according to a fixed schedule. For each ConvNet the training undergoes 100 epochs and the learning rate decreases every 30 epochs. For all experiments, the dropout regularization ratio was set to 0.9 in order to reduce complex co-adaptations of neurons in the nets for both networks.

3.2.2.4 Product Score Fusion

Given a testing skeleton sequence (sample), three JTMs are generated and fed into the three ConvNets respectively. Product score fusion is used to combine the outputs from the individual ConvNets. Specifically, the score vectors outputted by the three ConvNets are multiplied in an element-wise way, and the max score in the resultant vector is assigned as the probability of the test sequence. The index of this max score corresponds to the recognized class label.

3.2.3 Experimental Results

The proposed method was evaluated on four public benchmark datasets: the large NTU RGB+D Dataset [SLNW16], MSRC-12 Kinect Gesture Dataset [FMNK12], G3D [BMA12] and UTD-MHAD [CJK15]. Experiments were conducted on the effectiveness of individual encoding scheme in the proposed method, the effectiveness of rotation, the role of fine-tuning, and the product score fusion compared with the max and average score fusion methods. The final recognition results were compared with the state-of-the-art reported on the same datasets. In all experiments, the saturation and brightness range from $0\% \sim 100\%$ (mapped to $0 \sim 255$ in the JTM images).

3.2.3.1 Evaluation of Key Design Factors

Different Encoding Schemes

The effectiveness of different encoding schemes was evaluated on the G3D dataset, and the recognition accuracies are listed in Table 3.3.

Table 3.3: Comparison of the different encoding schemes on the G3D dataset in terms of recognition accuracy.

Techniques	Front	Top	Side	Fusion
Trajectory: t_1^i	65.45%	72.18%	73.54%	80.58%
Trajectory: $C\_t_1^i$	76.12%	75.55%	76.56%	83.65%
Trajectory: $MC\_t_1^i$	79.98%	78.25%	79.40%	87.68%
Trajectory: $MC_s\_t_1^i$	83.52%	81.32%	82.08%	89.98%
Trajectory: $MC_b\_t_1^i$	84.46%	84.68%	85.60%	93.84%
Trajectory: $MC_{sb}\_t_1^i$	86.25%	87.56%	86.54%	96.02%

From Table 3.3 we can see that the proposed encoding methods effectively capture spatio-temporal information. Each encoding method gradually amends more information to the JTMs for the three ConvNets to learn the discriminative features and improves the recognition. The three JTMs are complimentary to each other to improve recognition significantly through fusion.

Rotation

Rotation is adopted to mimic multiple views, and this simple process makes the proposed method capable of cross-view action recognition. At the same time, the rotation enlarges the training data and enables the method to work on small datasets. Table 3.4 shows the comparison of the proposed method with and without rotation on the NTU RGB+D and G3D datasets. As expected, the rotation operation im-

proves the performance of cross-view recognition largely (by almost 3.5 percentage points).

Table 3.4: Comparison the proposed method with and without rotation on the NTU RGB+D and G3D datasets in terms of recognition accuracy.

Dataset	Without Rotation	With Rotation
NTU RGB+D (Cross Subject)	75.30%	76.32%
NTU RGB+D (Cross View)	77.67%	81.08%
G3D	95.12%	96.02%

Fine-tuning vs. Training from Scratch

Even though the number of training samples per class is over 600 for the NTU RGB+D Dataset, fine-tuning with available models from ImageNet is still preferred in terms of recognition accuracy. Table 3.5 shows the results of two settings, fine-tuning and training from scratch, on NTU RGB+D and G3D datasets. In both settings, no rotation was performed. Notice that fine-tuning improved the recognition by 5 percentage point on the NTU RGB+D Dataset and almost doubled the recognition accuracy on the small G3D Dataset compared to training from scratch.

Table 3.5: Comparisons of fine-tuning and tranining from scratch on the NTU RGB+D and G3D datasets in terms of recognition accuracy.

Dataset	Training from Scratch	Fine-tuning
NTU RGB+D (Cross Subject)	72.50%	75.30%
NTU RGB+D (Cross View)	73.77%	77.67%
G3D	46.64%	94.65%

Comparison of Three Score Fusion Methods

There are two common used late score fusion methods, namely, average score fusion method and max score fusion method. However, in this chapter, we propose to adopt product score fusion which turns out to be more effective on the evaluated datasets. The comparison of these three score fusion methods on the four datasets for final recognition are listed in Table 3.6. From the Table we can see that on the evaluated four datasets, the product score fusion consistently outperformed the average and max score fusion methods. This verifies that the three JTMs are likely to be statistically independent and provide complementary information.

Table 3.6: Comparison of three score fusion methods on the four datasets in terms of recognition accuracy.

Dataset	Max	Average	Product
NTU RGB+D (Cross Subject)	73.56%	75.05%	76.32%
NTU RGB+D (Cross View)	78.43%	79.88%	81.08%
MSRC-12	91.70%	93.42%	94.86%
G3D	93.78%	94.65%	96.02%
UTD-MHAD	85.81%	86.42%	87.90%

3.2.3.2 NTU RGB+D Dataset

NTU RGB+D Dataset, which is the largest dataset to our best knowledge, was adopted to evaluate the proposed method, and for fair comparison and evaluation, the same protocol as that in [SLNW16] was used. It has both cross-subject and cross-view evaluation. In the cross-subject evaluation, samples of subjects 1, 2, 4, 5, 8, 9, 13, 14, 15, 16, 17, 18, 19, 25, 27, 28, 31, 34, 35 and 38 were used as training and samples of the remaining subjects were reserved for testing. In the cross-view evaluation, samples taken by cameras 2 and 3 were used as training, testing set includes the samples of camera 1. Table 5.7 lists the performance of the proposed method and those reported before.

Table 3.7: Comparative accuracies of the proposed method and previous methods on NTU RGB+D dataset.

Method	Cross subject	Cross view
Lie Group [VAC14]	50.08%	52.76%
Dynamic Skeletons [OBT13b]	60.23%	65.22%
HBRNN [DWW15]	59.07%	63.97%
ELC-KSVD [ZLZ+14]	60.04%	57.78%
2 Layer RNN [SLNW16]	56.29%	64.09%
2 Layer LSTM [SLNW16]	60.69%	67.29%
Part-aware LSTM [SLNW16]	62.93%	70.27%
ST-LSTM [LSXW16]	65.20%	76.10%
ST-LSTM + Trust Gate [LSXW16]	69.20%	77.70%
SOS + CNN [HLWL16]	72.36%	75.47%
JTM + CNN [WLHL16]	73.38%	75.20%
Proposed Method	**76.32%**	**81.08%**

From this Table we can see that our proposed method achieved the state-of-the-art results compared with both hand-crafted features and deep learning methods. The work [VAC14] focused only on single person action and could not model multi-person interactions well. Dynamic Skeletons method [OBT13b] performed better than some RNN-based methods verifying the weakness of the RNNs [DWW15, SLNW16], which only mines the short-term dynamics and tends

to overemphasize the temporal information even on large training data. LSTM and its variants [SLNW16, LSXW16] performed better due to their ability to utilize long-term context compared to conventional RNNs, but it is still weak in exploiting spatial information. By conducting rotation and using proposed effective product score fusion method, the proposed method achieved much better results than our previous methods [WLHL16, HLWL16] in both cross-subject and cross-view evaluation.

3.2.3.3 MSRC-12 Kinect Gesture Dataset

MSRC-12 [FMNK12] is a relatively large dataset for gesture/action recognition from 3D skeleton data captured by a Kinect sensor. For this dataset, cross-subjects protocol was adopted, that is, odd subjects were used for training and even subjects were for testing. Table 5.5 lists the performance of the proposed method and the results reported before.

Table 3.8: Comparison of the proposed method with the existing methods on the MSRC-12 Kinect Gesture dataset.

Method	Accuracy (%)
HGM [YYHD14]	66.25%
Pose-Lexicon [ZLO16b]	85.86%
ELC-KSVD [ZLZ+14]	90.22%
Cov3DJ [HTGES13]	91.70%
JTM [WLHL16]	93.12%
SOS [HLWL16]	94.27%
Proposed Method	**94.86%**

The confusion matrix is shown in Fig. 3.9. From the confusion matrix we can see that the proposed method distinguishes most of actions very well, but it is not very effective to distinguish "goggles" and "had enough" which shares the similar appearance of JTMs probably caused by 3D to 2D projection.

3.2.3.4 G3D Dataset

Gaming 3D Dataset (G3D) [BMA12] which focuses on real-time action recognition in a gaming scenario, was adopted for evaluation. For this dataset, the first 4 subjects were used for training, the fifth for validation and the remaining 5 subjects were for testing as configured in [NWJ15]. Table 5.2 compared the performance of the proposed method and previous reported results.

The confusion matrix is shown in figure 3.10. From the confusion matrix we can see that the proposed method recognizes most of actions well. The proposed method outperformed LRBM. LRBM confused the actions among "tennis swing

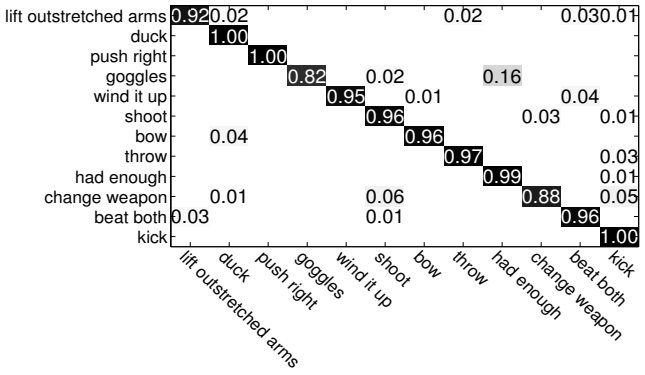

Figure 3.9: The confusion matrix of the proposed method on the MSRC-12 Kinect gesture dataset.

Table 3.9: Comparison of the proposed method with previous methods on the G3D dataset.

Method	Accuracy (%)
Cov3DJ [HTGES13]	71.95%
ELC-KSVD [ZLZ+14]	82.37%
LRBM [NWJ15]	90.50%
JTM [WLHL16]	94.24%
SOS [HLWL16]	95.45%
Proposed Method	**96.02%**

forehand" and "bowling", "golf" and "tennis swing backhand", "aim and fire gun" and "wave", "jump" and "walk", however, these actions are well distinguished by the proposed method likely because of the quality spatial information encoded in the JTMs. As for "aim and fire gun" and "wave", the proposed method could not distinguish them well without encoding the motion magnitude, but does well with the encoding of motion magnitude. However, the proposed method, confused "tennis swing forehand" and "tennis swing backhand". It's probably because the front and side projections of body shape of the two actions are too similar.

3.2.3.5 UTD-MHAD

UTD-MHAD [CJK15] is a multimodal action dataset, captured by one Microsoft Kinect camera and one wearable inertial sensor. For this dataset, cross-subjects protocol was adopted as in [CJK15], namely, the data from the subjects numbered 1, 3, 5, 7 were used for training while subjects 2, 4, 6, 8 were used for testing. Table 5.3 compares the performance of the proposed method and those reported before.

Please notice that the method used in [CJK15] is based on Depth and Inertial sensor data, not skeleton data alone.

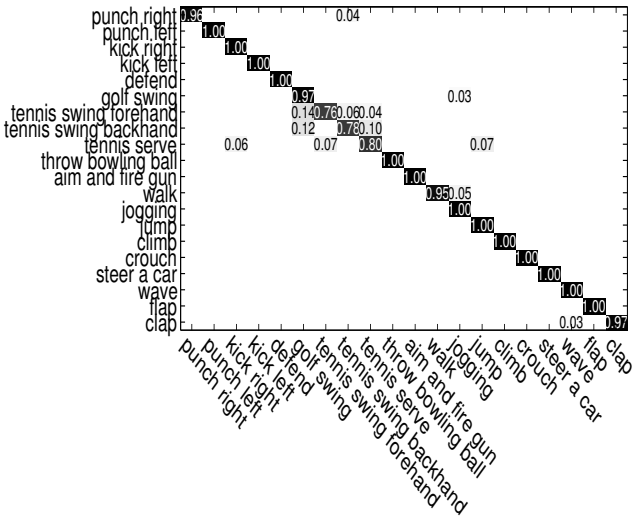

Figure 3.10: The confusion matrix of the proposed method on the G3D Dataset.

Table 3.10: Comparison of the proposed method with the previous methods on UTD-MHAD dataset.

Method	Accuracy (%)
ELC-KSVD [ZLZ+14]	76.19%
Kinect & Inertial [CJK15]	79.10%
Cov3DJ [HTGES13]	85.58%
JTM [WLHL16]	85.81%
SOS [HLWL16]	86.97%
Proposed Method	**87.90%**

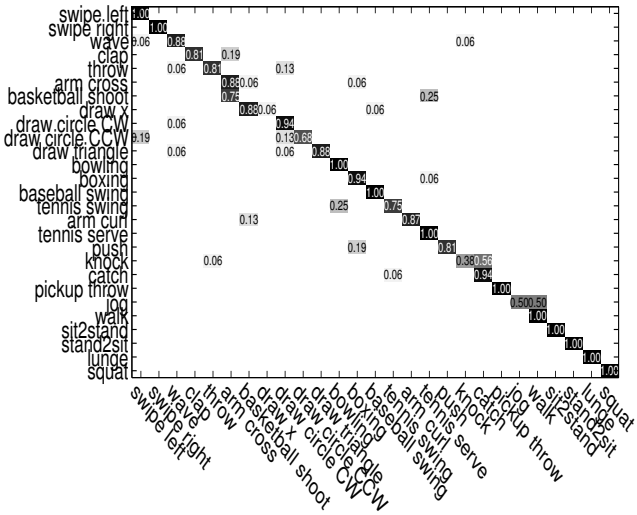

Figure 3.11: The confusion matrix of the proposed method on the UTD-MHAD dataset.

The confusion matrix is shown in Fig. 3.11. This dataset is much more challenging compared to the previous two datasets. From the confusion matrix we can

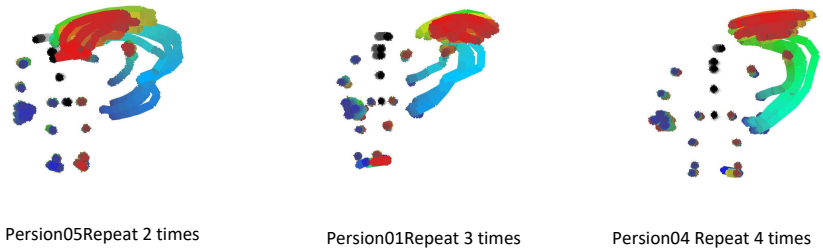

Figure 3.12: The generated JTMs of action "waving hand" performed by different persons and repeated different times from NTU RGB+D dataset [SLNW16].

see that the proposed method can not distinguish some actions well, for example, "jog" and "walk". A probable reason is that the proposed encoding process is also a temporal normalization process. The actions "jog" and "walk" would be normalized to have similar JTMs after the encoding.

3.2.3.6 Discussion

Optimal Orthogonal Planes

In this chapter, we adopt the three orthogonal planes of the natural real coordinates of the camera. One question is whether there are some orthogonal planes better than the natural ones. Generally speaking, there possibly exist three orthogonal views which are better than the natural coordinates if the three views result in less self-occlusion among the joints for all actions. Since only very sparse 20 joints are used to represent the skeleton, the likelihood of such self-occlusion of the joints would be very small. Consequently, no particular three orthogonal views would be obviously superior to others. However, the depth camera only captures $2\frac{1}{2}D$ in the natural coordinates and the skeleton is estimated from the $2\frac{1}{2}D$. It is likely that the natural coordinates could be slightly, but not significantly, better than other three orthogonal views.

To validate this, we conducted the following experiments on the G3D Dataset. Different three orthogonal views were generated by rotating the 3D points of joints and projecting them to the three orthogonal planes. The rotation was performed with a fixed step of 22.5° along the polar angle θ and azimuthal angle ψ, both in the range of $[-45°, 45°]$. Note that this range effectively covers all possible views since rotation beyond this range would result in swapping of views. Such swapping would not affect the recognition accuracy after fusion. Table 3.11 shows the recognition accuracies of different orthogonal views indicated by the values of θ and ψ.

The results in Table 3.11 have shown small and insignificant variation of the recognition accuracy among the views and the natural coordinates produced the

Table 3.11: The recognition accuracy (%) of different orthognal views.

ψ \ θ	$-45°$	$-22.5°$	$0°$	$22.5°$	$45°$
$-45°$	94.45	92.12	94.85	92.24	92.42
$-22.5°$	95.40	94.45	94.45	92.73	92.24
$0°$	94.45	95.05	95.12	94.85	94.15
$22.5°$	94.85	94.85	94.45	94.85	93.45
$45°$	92.24	95.00	94.85	94.15	94.15

best result.

In this chapter, we fuse three orthogonal image planes to improve the final accuracy. Another questions is whether adding more views will lead to better recognition. Some experiments were conducted on the G3D dataset to answer this question. Firstly, the views of the natural coordinates were fused with the views after rotating the points by the specified angles in θ and ψ. Table 3.12 shows the results by fusing two pairs of three orthogonal planes, one is the natural coordinates and the other is specified by the rotation angles θ and ψ. The accuracies of all cases are almost same.

Table 3.12: The results of fusing the original three orthogonal planes and rotated three planes.

ψ \ θ	$-45°$	$-22.5°$	$0°$	$22.5°$	$45°$
$-45°$	94.45	93.45	94.85	95.15	95.12
$-22.5°$	94.85	94.54	95.15	95.12	94.85
$0°$	95.12	95.15	95.12	94.54	94.54
$22.5°$	94.85	94.24	95.15	95.15	94.85
$45°$	94.85	94.85	95.15	95.12	95.15

We also evaluated the performance by fusing all views of the 9 coordinates including the natural ones, where $\theta \in \{-22.5°, 0°, 22.5°\}$ and $\psi \in \{-22.5°, 0°, 22.5°\}$, and all views of the 25 coordinates, where $\theta \in \{-45°, -22.5°, 0°, 22.5°, 45°\}$ and $\psi \in \{-45°, -22.5°, 0°, 22.5°, 45°\}$ respectively. The results are shown in Table 3.13. It can be seen that fusing views of multiple orthogonal coordinates did not improve the performance on this dataset. Similar results would be expected on other datasets for the reason explained above.

The above analysis and experiments have demonstrated that the three orthogonal views in the natural coordinates are likely to be sufficient.

Table 3.13: The results for fusing views of multiple coordinates.

Number of views	Accuracy (%)
9	95.15
25	94.85

Execution Rate Variation and Repetition

Theoretically, the proposed algorithm may suffer from the variations of the ratios of different actions, especially those actions that are repetitive. In fact, variation in executing rate and repetition is one of the key challenges in action recognition. However, the proposed method can deal with such situations to some extend. In the proposed method, the spatial-temporal information is encoded into texture images. For those actions that are repetitive or have large variations of executing rate, the shape formed by the joint trajectories help to distinguish them, even though the texture information would vary a lot. Fig 3.12 shows several examples of "waving hand" from NTU RGB+D dataset. We picked up three samples that were performed by different persons with different executing rates and repetitions. From the figure we can see that with variations of the ratios and repetitions, the textures of the JTMs are different, but the shape is still similar. To further justify this point, one experiment was conducted as follows. We divided the samples for action "waving hand" from NTU RGB+D dataset into two groups, one group with samples repeating less than 3 times, and one group of samples repeating with more than 3 times. We used the first group together with the "kicking" and "clapping" actions as training, and used the second group together with the "kicking" and "clapping" actions as testing and also swapped the training and testing samples. We found that in both settings, the proposed method sucessfully distinguished the three actions, and the accuracy is 100% for both settings.

3.3 Summary

This chapter presented two methods to address the research questions based on skeleton modality: (a) how to effectively mine the most frequent and relevant (discriminative, representative and non-redundant) features based on skeleton data for action recognition? (b) how to effectively represent skeleton sequence data for ConvNets-based recognition? A novel method to explore temporal information and mine the different combinations of different body parts in different frames is developed for research question (a). An effective method that projects the joint trajectories to three orthogonal JTMs to encode the spatial-temporal information into texture patterns is developed for research question (b). The strength of the proposed methods

has been demonstrated through the state-of-the-art results obtained on the recent and challenging benchmark datasets for activity and action recognition. With the increasing popularity of Kinect-based action recognition and advances in data mining and deep learning methods, the proposed methods are promising for practical applications.

Chapter 4

Depth-based Action Recognition

Research on action recognition has mainly focused on conventional RGB (red, green and blue) video and hand-crafted features. However, there is no universally best hand-engineered feature for all datasets [WUK+09]. Microsoft Kinect sensors provide an affordable technology to capture depth maps and RGB images in real-time. Compared to traditional images, depth maps offer better geometric cues and less sensitivity to illumination changes for action recognition [LZL10, WLWY12, OL13, YT14]. Current approaches to recognizing actions from RGB-D data are still based on hand-crafted features, which are often shallow. In addition, their high-dimensional description of local or global spatio-temporal information and performance vary across datasets. How to apply deep learning methods to depth-based action recognition is still an open problem. This chapter presents the studies that address the research questions 3, 4 and 5 listed in Section 1.2.

4.1 Weighted Hierarchical Depth Motion Maps with ConvNets

4.1.1 Prior Works and Our Contributions

With Microsoft Kinect Sensors researchers have developed methods for depth map-based action recognition. Li et al. [LZL10] sampled points from a depth map to obtain a bag of 3D points to encode spatial information and employ an expandable graphical model to encode temporal information [LZL08]. One limitation of this method is view-dependency. Yang et al. [YZT12] stacked differences between projected depth maps as a depth motion map (DMM) and then used HOG to extract relevant features from the DMM. This method transforms the problem of action recognition from spatio-temporal space to spatial space. However, this method is also view-dependent. In [OL13], a feature called Histogram of Oriented 4D Normals (HON4D) was proposed; surface normal is extended to 4D space and quantized by regular polychorons. Following this method, Yang and Tian [YT14] cluster hypersurface normals and form the polynormal which can be used to jointly capture the local motion and geometry information. Super Normal Vector (SNV) is generated by aggregating the low-level polynormals. In [LJT14], a fast binary range-sample feature was proposed based on a test statistic by carefully designing the sampling

scheme to exclude most pixels that fall into the background and to incorporate spatio-temporal cues.

Depth maps have been augmented with skeleton data in order to improve recognition. Wang et al. [WLWY12] designed a 3D Local Occupancy Patterns (LOP) feature to describe the local depth appearance at joint locations to capture the information related to subject-object interactions. The intuition is to count the number of points that fall into a spatio-temporal bin when the space around the joint is occupied by the object. Wang et al. [WLWY14] adopted LOP feature calculated from the 3D point cloud around a particular joint to discriminate different types of interactions and Fourier Temporal Pyramid (FTP) to represent the temporal structure. Based on these two types of features, the Actionlet Ensemble Model (AEM) was proposed which combines the features of a subset of the joints. To fuse depth-based features with skeleton-based features, Althloothi et al. [AMZV14] presented two sets of features; features for shape representation extracted from depth data by using a spherical harmonics representation and features for kinematic structure extracted from skeleton data. The shape features are used to describe the 3D silhouette structure while the kinematic features are used to describe the movement of the human body. Both sets of features are fused at the kernel level for action recognition by using Multiple Kernel Learning (MKL) technique.

However, all of previous methods are based on hand-crafted features, which are often shallow and dataset-dependent. We presents a novel method to apply ConvNets trained on ImageNet to depth map sequences for action recognition with a small number of training samples. Generally speaking, ConvNets require a sufficiently large number of training samples and how to apply the ConvNets to small datasets is still an unsolved problem. To address this issue, an architecture comprising Weighted Hierarchical Depth Motion Maps (WHDMM) and Three Channel Convolutional Neural Network (3ConvNets) is proposed. WHDMM is a strategy for transforming the problem of action recognition to image classification and making effective use of the rich information offered by the depth maps. Specifically, three-dimensional (3D) pointclouds constructed from the original depth data are rotated to mimic the different camera viewpoints, so that our algorithm becomes view-tolerant. Each rotated depth frame is first projected onto three orthogonal Cartesian planes, and then for each projected view, the absolute differences (motion energy) between consecutive frames or sub-sampled frames are accumulated through an entire depth sequence. To encode the temporal order of body poses, weights are introduced such that recent frames contribute more to WHDMMs so that pair-actions (e.g. “sit down” and “stand up”, having similar but reverse temporal patterns) can be distinguished. To leverage the ConvNets trained over ImageNets, the WHDMM are encoded into pseudo-color images. Such encoding converts the spatio-temporal mo-

tion patterns in videos into spatial structures (edges and textures) thus enabling the ConvNets to learn the filters [ZF14]. Three ConvNets are trained on the three WHDMMs constructed from the projected Cartesian planes independently and the results are fused to produce the final classification score.

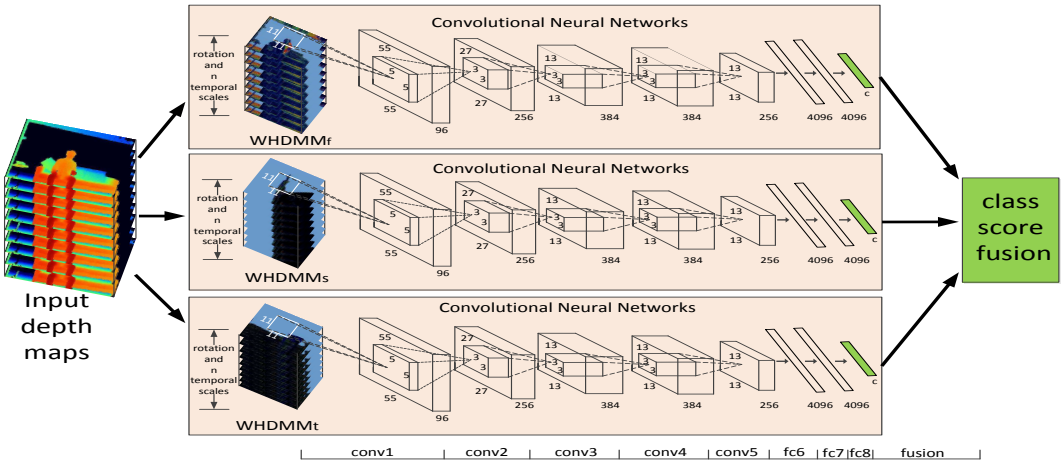

Figure 4.1: The proposed WHDMM + 3ConvNets architecture for depth-based action recognition.

4.1.2 The Proposed Methods

The proposed WHDMM + 3ConvNets method consists of two major components (Fig. 4.1): three ConvNets and the construction of WHDMMs from sequences of depth maps as the input to the ConvNets. Given a sequence of depth maps, 3D points are created and three WHDMMs are constructed by projecting the 3D points to the three orthogonal planes. Each WHDMM serves as an input to one ConvNet for classification. Final classification of the given depth sequence is obtained through a late fusion of the three ConvNets. A number of strategies have been developed to deal with the challenges posed by small datasets. Firstly, more training data are synthesized by (a) rotating the input 3D points to mimic different viewpoints and (b) constructing WHDMMs at different temporal scales. Secondly, the same ConvNet architecture as used for ImageNet is adopted so that the model trained over ImageNet [KSH12] can be adapted to our problem through transfer learning. Thirdly, each WHDMM goes through a pseudo-color coding process to encode, with enhancement, different motion patterns into the pseudo-RGB channels before being input to the ConvNets. In the rest of this section, rotation of the 3D points, construction and pseudo-color coding of WHDMMs and training of the ConvNets are described.

4.1.2.1 Rotation To Mimic Different Camera Viewpoints

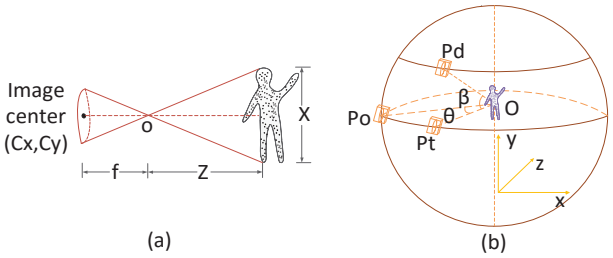

Figure 4.2: Process of rotating 3D points to mimic different camera viewpoints.

Fig 4.2 (a) illustrates how to covert a pixel in a depth map into a 3D point by calculating its location (X, Y, Z) in the real-world coordinate system centered on the camera by using the pair of equations,

$$X = \frac{Z \cdot (U - C_x)}{f_x}, Y = \frac{Z \cdot (V - C_y)}{f_y}. \tag{4.1}$$

In Equation 4.1, (U, V) and Z denote screen coordinates and depth value respectively; C_x, C_y denote the center of a depth map; f_x and f_y are the focal lengths of the camera. For Kinect-V1 cameras, $f_x = f_y = 580$ [SJP11].

The rotation of the 3D points can be performed equivalently by assuming that a virtual RGB-D camera moves around and points at the subject from different viewpoints (Fig. 4.2). Suppose the virtual camera moves from position P_o to P_d, its motion can be decomposed into two steps: first move from P_o to P_t, with rotation angle denoted by θ and then moves from P_t to P_d, with rotation angle denoted by β. The coordinates after rotation can be computed through multiplication by the transformation matrices $\mathbf{T}r_y$ and $\mathbf{T}r_x$, as

$$[X', Y', Z', 1]^{\mathrm{T}} = \mathbf{T}r_y \mathbf{T}r_x \left[X, Y, Z, 1\right]^{\mathrm{T}} \tag{4.2}$$

where X', Y', Z' represent the 3D coordinates after rotation, $\mathbf{T}r_y$ denotes the transform around Y axis (right-handed coordinate system) and $\mathbf{T}r_x$ denotes the transform around X axis. The transformation matrices can be expressed as

$$\mathbf{T}r_y = \begin{bmatrix} R_y(\theta) & T_y(\theta) \\ \mathbf{0} & 1 \end{bmatrix}; \mathbf{T}r_x = \begin{bmatrix} R_x(\beta) & T_x(\beta) \\ \mathbf{0} & 1 \end{bmatrix} \tag{4.3}$$

where

$$\mathbf{R}_y(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \mathbf{T}_y(\theta) = \begin{bmatrix} 0 \\ Z \cdot \sin(\theta) \\ Z \cdot (1 - \cos(\theta)) \end{bmatrix}; \mathbf{R}_x(\beta) = \begin{bmatrix} cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix}$$

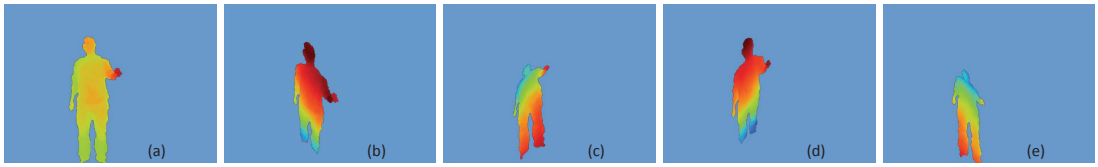

Figure 4.3: Example depth maps synthesized by the virtual RGB-D camera. a) original depth map, depth maps synthesized respectively with the parameters b) ($\theta = 45°$, $\beta = 45°$), c) ($\theta = 45°, \beta = -45°$), d) ($\theta = -45°, \beta = 45°$) and e) ($\theta = -45°, \beta = -45°$).

$$\mathbf{T}_x(\beta) = \begin{bmatrix} -Z \cdot \sin(\beta) \\ 0 \\ Z \cdot (1 - \cos(\beta)) \end{bmatrix}.$$

After rotation, a depth map from a different viewpoint can be obtained from

$$U' = \frac{X' \cdot f_x}{Z'} + C_x; V' = \frac{Y' \cdot f_y}{Z'} + C_y, \tag{4.4}$$

where U', V' and Z' respectively denote the new screen coordinates and their corresponding depth value.

Since RGB-D camera only captures $2\frac{1}{2}D$, and not the full $3D$ information, the rotation has to be within a range such that the synthesized depth maps still capture sufficient spatio-temporal information of the actions. In other words, both θ and β have to be limited to a certain range. Fig.4.3 shows some examples of the synthesized depth maps and the original one from which they were created. Even at relatively large angles ($|\theta| = 45°, |\beta| = 45°$), the synthesized depth maps still capture the shape of the body well. In some extreme cases where θ and β become very large (Fig. 4.4), the synthesized depth maps do not capture sufficient spatial information of the subject. Empirically, the useful range of the angles is between $(-60°, 60°)$ for both θ and β.

4.1.2.2 Construction of WHDMM

Each of the original and synthesized depth maps is projected to three orthogonal Cartesian planes, referred to as front, side and top views and denoted by map_p where $p \in \{f, s, t\}$. Unlike the Depth Motion Map (DMM) [YZT12] where it is calculated by accumulating the thresholded difference between consecutive frames, three extensions are proposed to construct a WHDMM. Firstly, in order to preserve subtle motion information, for example, turning a page when reading books, for each projected map, the motion energy is calculated as the absolute difference between consecutive or sub-sampled depth frames without thresholding. Secondly, to exploit speed invariance and suppress noise, several temporal scales, referred to as hierarchical temporal scales, are generated as illustrated in Fig. 4.5, where N is the

number of frames, and a WHDMM is constructed for each of the temporal scales. Through this process, the number of training samples are further increased and at the same time the issue of speed variation is addressed. Lastly, in order to differentiate motion direction, a temporal weight is introduced, giving a higher weight to recent depth frames than to past frames.

Let $WHDMM_{p_n}^t$ be the WHDMM being projected to view p and accumulated up to frame t at the n^{th} temporal scale. It can be expressed as:

$$\begin{aligned} WHDMM_{p_n}^t &= \gamma |map_p^{(t-1)n+1} - map_p^{(t-2)n+1}| \\ &+ (1-\gamma) WHDMM_{p_n}^{t-1}, \end{aligned} \tag{4.5}$$

where map_p^i denotes the i^{th} depth map in the original video sequence and being projected to view p; $\gamma \in (0, 1)$ stands for the temporal weight and $WHDMM_{p_n}^1 = |map_p^{n+1} - map_p^1|$.

Figure 4.4: Examples of synthesised depth maps for cases where θ and β are very large. a) $(\theta = -75°, \beta = -75°)$; b) $(\theta = -85°, \beta = -85°)$.

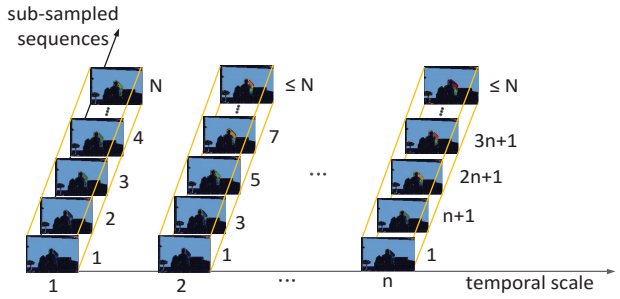

Figure 4.5: Hierarchical temporal scales: for the n^{th} temporal scale, the sub-sampled sequence is constructed by taking one frame, starting from the first frame, from every n frames.

Using this simple temporal weighting scheme along with the pseudo-color coding of the WHDMMs (to be described in the next section), pair actions, such as "sit down" and "stand up", can be distinguished.

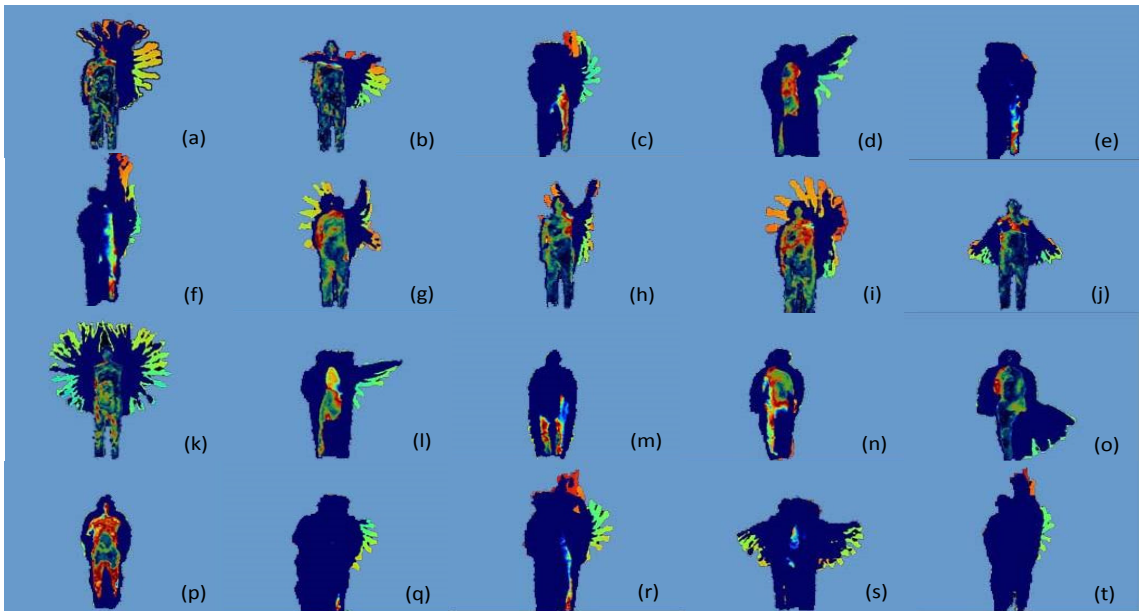

Figure 4.6: Examples of pseudo-color coded WHDMMs of actions in the MSRAction3D dataset performed by randomly selected subjects: a) high arm wave, b) horizontal arm wave, c) hammer, d) hand catch, e) forward punch, f) high throw, g) draw X, h) draw tick, i) draw circle, j) hand clap, k) two hand wave, l) side-boxing, m) bend, n) forward kick, o) side kick, p) jogging, q) tennis swing, r) tennis serve, s) golf swing, and t) pick up & throw.

4.1.2.3 Pseudo-Color Coding of WHDMMs

Abidi et al. [AZGA06] reported gaining an enhanced perceptual quality and more information from gray scale texture through a human perception-based color coding. Motivated by this result, we transform the WHDMM into a pseudo-color that enhances motion patterns of actions and improves signal-to-noise ratio. Methods of pseudo-color coding include spectrum-based maps, naturally ordered maps, uniformly varying maps, shape and value-based maps, and function-based maps [AZGA06]. Furthermore, nonlinear transformations can increase/decrease the contrast of certain gray levels without truncating low/high pixel intensities [Joh12]. In this chapter, an improved rainbow transform (a special case of the sine transform) which is a variant of the spectrum-based mapping method is developed. The improved rainbow transform is expressed as:

$$C_{i=1,2,3} = \{\sin[2\pi \cdot (-I + \varphi_i) \cdot \frac{1}{2} + \frac{1}{2}]\}^{\alpha} \cdot f(I) \tag{4.6}$$

where $C_{i=1,2,3}$ represent the BGR channels, respectively; I is the normalized gray value; φ_i denotes the phase of the three channels; α is the power; $f(I)$ is an amplitude modulation function; the added value, $\frac{1}{2}$, guarantees non-negativity. The value of parameter α can be chosen to vary noise suppression. In our work, $\varphi_{i=1,2,3}$ and $f(I)$

are set to $\frac{1}{5} - \frac{1}{2}\pi, \frac{1}{5} - \frac{1}{2}\pi - \frac{3}{14}, \frac{1}{5} - \frac{1}{2}\pi - \frac{6}{14}, \frac{1}{4} + \frac{3}{4}I$, respectively.

To encode a WHDMM, linear mapping is used to convert WHDMM values to $I \in [0, 1]$. Fig. 4.7 shows the transform with $\alpha = 1$ and $\alpha = 10$. As observed, higher level noise in WHDMM is found in the areas of background or at edges of the subjects where the WHDMM values are either very large or very small. Thus, in order to improve the signal-to-noise-ratio (SNR) of WHDMM, the parameters of the transform in Eq. (4.6) are chosen so as to suppress both the small and large values of WHDMM. In addition, Fig. 4.7 shows that the improved rainbow transform with a relatively large α encodes the gray intensities to RGB values in a drastic manner than small α (e.g. $\alpha = 1$), and suppresses the noise in the color-encoded WHDMMs more effectively.

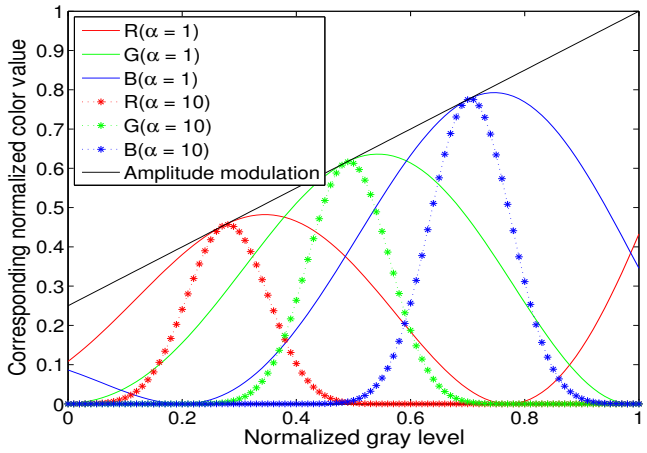

Figure 4.7: Visual comparison of improved rainbow transform with $\alpha = 1$ and $\alpha = 10$.

Fig. 4.6 shows sample pseudo-color coded WHDMMs of the actions from the MSRAction3D dataset. Although the WHDMMs for the actions "forward kick" (Fig. 4.6n)and "jogging" (Fig. 4.6p) appear similar, the pseudo-coloring has highlighted the differences. Since the texture and edges in a WHDMM are accumulation of spatial and temporal information, the pseudo-color coding remaps the spatio-temporal information of actions.

The value of α controls how well the pseudo-color coding improves the SNR of a WHDMM. Fig. 4.8 shows the pseudo-color coded WHDMM of action "eat" at the 5^{th} temporal scale. Notice that when $\alpha = 1$ both the background (i.e. sofa and person in the background) and the foreground subject are clearly noticeable. With increased α value, say $\alpha = 10$, the background is suppressed with little loss of the foreground information. However, if the value of α is very large (say $\alpha = 20$), both the foreground and background are suppressed.

Fig. 4.9 illustrates how the recognition accuracy on the MSRDailyActivity3D

dataset varies with the value of α when only the WHDMMs of frontal projection are used.

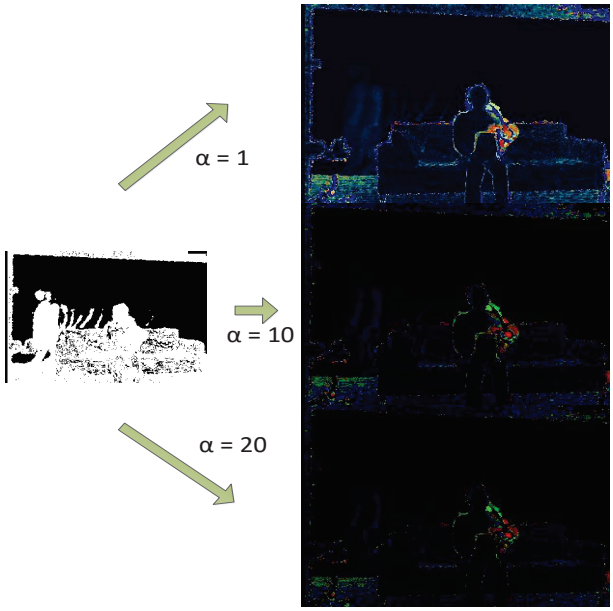

Figure 4.8: A sample color-coded WHDMM of action "eat" with different α values.

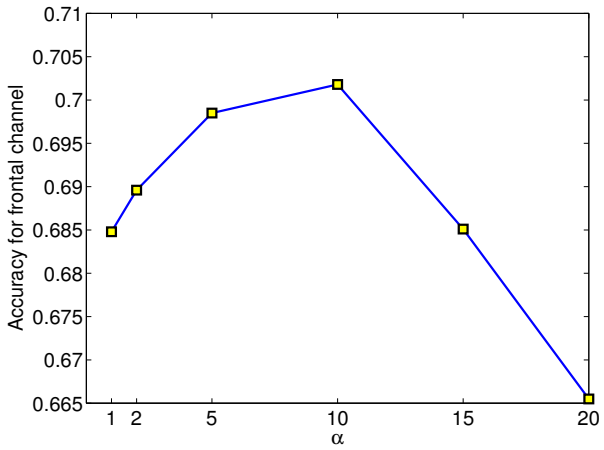

Figure 4.9: Variation of recognition accuracy with increasing value of α. MSR-DailyActivity3D dataset has been used with only the frontal channel.

4.1.2.4 Network Training & Class Score Fusion

Three ConvNets are trained on the pseudo-color coded WHDMMs in the three Cartesian planes. The layer configuration of the ConvNets follows those in [KSH12] and is schematically shown in Fig. 4.1. Each ConvNet contains eight layers with

weights; the first five are convolutional layers and the remaining three are fully-connected layers. The implementation is derived from the publicly available Caffe toolbox [JSD+14] based on one NVIDIA Tesla K40 card.

Training

The training procedure is similar to that in [KSH12], wherein the network weights are learned using the mini-batch stochastic gradient descent with the momentum value set to 0.9 and weight decay set to 0.0005. All hidden weight layers use the rectification (RELU) activation function. At each iteration, a mini-batch of 256 samples is constructed by sampling 256 shuffled training color-coded WHDMMs. All color-coded WHDMMs are resized to 256×256. The learning rate is initially set to 10^{-2} and used to directly train the networks from data without initializing the weights with pre-trained models on ILSVRC-2012 (Large Scale Visual Recognition Challenge 2012, a version of ImageNet). The rate is set to 10^{-3} for fine-tuning with pre-trained models on ILSVRC-2012, and then it is decreased according to a fixed schedule, which is kept the same for all training sets. For each ConvNet the training undergoes 100 cycles and the learning rate decreases every 20 cycles. For all experiments, the dropout regularization ratio was set to 0.5 in order to reduce complex co-adaptations of neurons in the nets.

Class Score Fusion

Given a test depth video sequence (sample), WHDMMs at different temporal scales are classified using the trained ConvNets. The average scores of n scales for each test sample are calculated for each of the three ConvNets. The final class score for a test sample is the average of the outputs from the three ConvNets. Thus

$$\text{score}_{\text{test}} = \frac{\sum_{c=1}^{3} \sum_{i=1}^{n} \text{score}_c^i}{3n}. \tag{4.7}$$

where $\text{score}_{\text{test}}$ represents the final class score for a test sample while score_c^i denotes the score of i-th temporal scale for c-th channel.

4.1.3 Experimental Results

The proposed method was evaluated on three public benchmark datasets: MSRAction3D [LZL10], UTKinect-Action [XCA12] and MSRDailyActivity3D [WLWY12]. An extension of MSRAction3D, called MSRAction3DExt dataset, was used. It contains more than twice (i.e. 23) as many subjects as in the previous dataset performing the same set of actions. In order to test the stability of the proposed method with respect to the number of actions, a new dataset was created by combining

MSRAction3DExt, UTKinect-Action and MSRDailiyActivity3D datasets; the new dataset is referred to as Combined dataset. In all experiments, θ varied over the range $(-30° : 15° : 30°)$ and β varied over the range $(-5° : 5° : 5°)$. For WHDMM, γ was set to 0.495. Different temporal scales, as detailed below, were set according to the noise level, complexity and average cycle in frames of actions performed in different datasets. In order to evaluate the proposed strategies, six scenarios were designed based on a) whether the training samples consists of: a subset T1 of the original samples, samples (T2) synthesized from T1 through rotation and samples T3 generated through temporal scaling of T1; b) how the ConvNets were initialized: random initialization or use of pre-trained model over ImageNet.

S1: Use of T1 and training the ConvNets with random initialization.

S2: Use of T1 and T2, training the ConvNets with random initialization.

S3: Use of T1 and training the ConvNets with pre-trained model.

S4: Use of T1 and T2, training the ConvNets with pre-trained model.

S5: Use of T1 and T3, training the ConvNets with pre-trained model.

S6: Use of T1, T2 and T3, training the ConvNets with pre-trained model.

The six scenarios evaluate the proposed method from different perspectives and effectiveness of the proposed strategies. Scenario S6 provides an evaluation of the overall performance of the proposed method.

4.1.3.1 MSRAction3D Dataset

The MSRAction3D dataset [LZL10] was adopted to evaluate the proposed method. In order to obtain a fair comparison, the same experimental setting as that in [WLWY12] is followed, namely, the cross-subjects settings: subjects 1, 3, 5, 7, 9 for training and subjects 2, 4, 6, 8, 10 for testing. For this dataset, temporal scale $n = 1$ and $\alpha = 2$, and the proposed method achieved 100% accuracy. Results of scenarios S1-S4 are shown in Table 4.1. As seen, without using temporal scaling, i.e. $n = 1$, the recognition can reach 100%.

Table 4.1: Recognition results achieved on the MSRAction3D dataset.

Training Setting	Accuracy
S1	7.12%
S2	34.23%
S3	100.00%
S4	100.00%

Pre-training on ILSVRC-2012 (i.e. S3 and S4) is very effective. Because the volume of training data is not enough to train millions of parameters of the deep networks, without good initialization, overfitting becomes inevitable. When the networks were directly trained from the original samples (i.e. S1), the performance is only slightly better than a random guess.

Table 4.2 compares the performance of the proposed WHDMM + 3ConvNets with recently reported depth-based results.

Table 4.2: Comparison of the proposed method with existing depth-based methods on the MSRAction3D dataset.

Method	Accuracy
Bag of 3D Points [LZL10]	74.70%
Actionlet Ensemble [WLWY12]	82.22%
Depth Motion Maps [YZT12]	88.73%
HON4D [OL13]	88.89%
SNV [YT14]	93.09%
Range Sample [LJT14]	95.62%
Proposed Method	**100.00%**

The proposed method outperforms all previous methods. This is probably because (1) the WHDMM can filter out the simple and static background in MSRAction3D; (2) the pre-trained model can initialize the three ConvNets well, so that they can learn the filters well even though action recognition and image classification belong to different domains; and (3) the WHDMM and pseudo-color coding can encode the spatio-temporal information into a single image.

4.1.3.2 MSRAction3DExt Dataset

The MSRAction3DExt dataset is an extension of MSRAction3D dataset with an additional 13 subjects performing the same 20 actions 2 to 4 times in a similar environment as that of MSRAction3D. Thus, there are 20 actions, 23 subjects and 1379 video clips. Similarly to MSRAction3D, the proposed method was evaluated under four settings and the results are listed in Table 4.3. For this dataset, samples of odd-numbered subjects were used for training and samples of the even-numbered subjects were used for testing.

Table 4.3: Recognition results achieved on the MSRAction3DExt dataset.

Training Setting	Accuracy
S1	10.00%
S2	53.05%
S3	100.00%
S4	100.00%

Table 4.1 and Table 4.3 show that as the volume of dataset increases with respect to the MSRAction3D dataset, the performance improved from 34.23% to 53.05% when the Nets are directly trained from the original and synthesized samples. However, the performance is not comparable to that obtained when the pre-trained model on ImageNet was used for initialization. The proposed method achieved again 100% using the pre-trained model followed by fine-tuning even though this dataset has more variations across subjects. Table 4.4 shows the results of SNV [YT14] on the MSRAction3DExt dataset. As seen, the proposed method outperforms SNV [YT14].

Table 4.4: Comparision of the proposed method with SNV on the MSRAction3DExt dataset.

Method	Accuracy
SNV [YT14]	90.54%
Proposed Method	**100.00%**

Using the pre-trained model followed by fine-tuning is effective on small datasets. In the following experiments, results obtained by training the networks using the pre-trained model and fine-tuning, i.e S3-6, are reported.

4.1.3.3 UTKinect-Action Dataset

The UTKinect-Action dataset [XCA12] was captured using a stationary Kinect-V1 sensor. Notice that one of the challenges of this dataset is viewpoint variation.

For this dataset, temporal scale $n = 5$ and $\alpha = 2$ were set to exploit the temporal information because, unlike, the MSRAction3D dataset, each samples in this dataset contains multiple cycles of the actions. The cross-subject evaluation scheme used in [XA13] was adopted. This is different from the scheme used in [XCA12] where more subjects were used for training in each round. Experiments on the four training scenarios S3, S4, S5 and S6 were conducted and the results are shown in Table 4.5.

Table 4.5: Recognition results achieved on the UTKinect-Action dataset using different training settings.

Training Setting	Accuracy
S3	82.83%
S4	88.89%
S5	86.87%
S6	90.91%

Table 4.5, shows that inclusion of synthesized samples improved tolerance to viewpoint variation and, thus the recognition accuracy was 6 percentage points higher. The confusion matrix for S6 is shown in Fig. 4.10. The most confused actions are *hand clap* and *wave* which share similar appearance of WHDMMs.

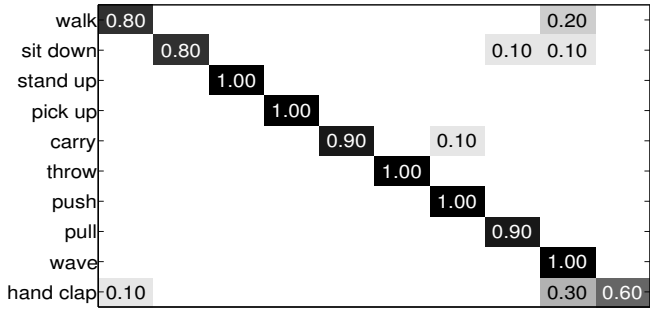

Figure 4.10: The confusion matrix of proposed method for UTKinect-Action dataset.

Table 4.6 shows the performance of the proposed method compared to the previous depth-based methods on the UTKinect-Action dataset. The improved performance will suggest that the proposed method has better viewpoint tolerance than other depth-based algorithms.

Table 4.6: Comparative accuracy of proposed method and previous depth-based methods using the UTKinect-Action dataset.

Method	Accuracy
DSTIP+DCSF [XA13]	78.78%
Random Forests [ZCG13]	87.90%
SNV [YT14]	88.89%
Proposed Method	**90.91%**

4.1.3.4 MSRDailyActivity3D Dataset

The MSRDailyActivity3D dataset [WLWY12] was used to evaluate the proposed method. Compared with MSRAction3D(Ext) and UTKinect-Action datasets, actors in this dataset present large spatial and temporal changes. Most activities in this dataset involve human-object interactions.

For this dataset, the temporal scale was set to $n = 21$ and $\alpha = 10$, a larger number of scales and power than those used for MSRAction3D and UTKinect-Action datasets. This choice of values was made to exploit temporal information and suppress the high level noise in this dataset. The same experimental setting as in [WLWY12] was adopted and the final recognition accuracy reached 85.00%. Results for the training settings, S3, S4, S5 and S6, are reported in Table 4.7. The samples (T2) synthesized through rotation improved the recognition accuracy by 15 percentage points and the samples (T3) synthesized through temporal scaling further improved the recognition accuracy by additional 15 percentage points

Table 4.8 compared the performance of the proposed method and that of existing depth-based methods and Fig. 4.11 depicts the confusion matrix of the proposed

Table 4.7: Recognition results achieved on the MSRDailyActivity3D dataset using different training settings.

Training Setting	Accuracy
S3	46.25%
S4	61.25%
S5	75.62%
S6	85.00%

method.

Table 4.8: Comparative accuracy of proposed method and previous depth-based methods using the MSRDailyActivity3D dataset.

Method	Accuracy
LOP [WLWY12]	42.50%
Depth Motion Maps [YZT12]	43.13%
Local HON4D [OL13]	80.00%
Actionlet Ensemble [WLWY12]	85.75%
SNV [YT14]	86.25%
Range Sample [LJT14]	**95.63%**
Proposed Method	85.00%

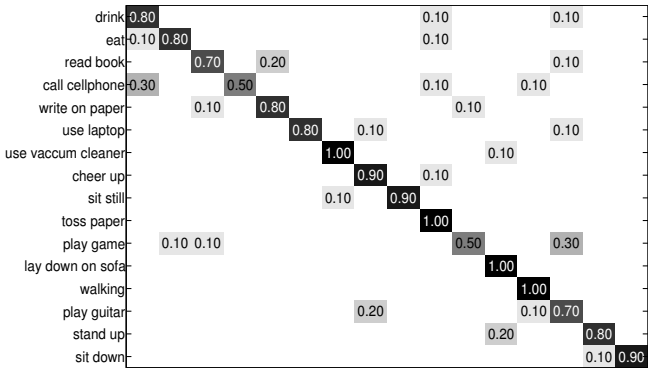

Figure 4.11: The confusion matrix of proposed method for MSRDailyActivity3D dataset.

Table 4.8 shows that the proposed method outperforms the Depth Motion Map method [YZT12] and, has comparable performance to SNV [YT14] and Actionlet Ensemble [WLWY12]. Notice that local HON4D [OL13] used skeleton data for localizing the subjects in depth maps while Actionlet Ensemble [WLWY12] and SNV [YT14] both used depth and skeleton data to extract features. However, the proposed method performed worse than the Range Sample [LJT14]. Two reasons are adduced for this observation. First, the background of this dataset is complex and much more temporally dynamic compared with MSRAction3D(Ext) and this introduced noise in the WHDMMs. However, the Range Sample [LJT14] method has

a mechanism to remove/reduce the interference from the background using skeleton data for preprocessing. Second, WHDMM is not sufficient to differentiate subtle differences in motion between some actions when interactions with objects become a key differentiating factor (e.g. *call cellphone*, *drink* and *eat*).

4.1.3.5 Combined Dataset

The Combined dataset is a combination of MSRAction3DExt, UTKinect-Action and MSRDailyActivity3D datasets and was created to test the performance of the proposed method when the number of actions increased. The Combined dataset is challenging due to its large variations in background, subjects, viewpoints and imbalanced number of samples for each actions. The same actions in different datasets are combined into one action and there are in total 40 distinct actions in the Combined dataset.

Table 4.9: Description of the Combined dataset: A denotes MSRAction3DExt dataset; U denotes UTKinect-Action dataset; D denotes MSRDailyActivity3D dataset.

Action Label & Name	From	Action Label & Name	From
1. high arm wave	A	21. walk	U&D
2. horizontal arm wave	A	22. sit down	U&D
3. hammer	A	23. stand up	U&D
4. hand catch	A	24. pick up	U
5. forward punch	A	25. carry	U
6. high throw	A&U	26. push	U
7. draw X	A	27. pull	U
8. draw tick	A	28. drink	D
9. draw circle	A	29. eat	D
10. hand clap	A&U	30. read book	D
11. two hand wave	A&U	31. call cellphone	D
12. side-boxing	A	32. write on a paper	D
13. bend	A	33. use laptop	D
14. forward kick	A	34. use vacuum cleaner	D
15. side kick	A	35. cheer up	D
16. jogging	A	36. sit still	D
17. tennis swing	A	37. toss paper	D
18. tennis serve	A	38. play game	D
19. golf swing	A	39. lay down on sofa	D
20. pickup & throw	A	40. play guitar	D

For this dataset, cross-subject scheme was adopted with half of the subjects used for training and the rest for testing, the choice was made such that the training and testing subjects were the same as those when each of the original individual datasets was used for evaluation. This provides a fair basis for the evaluation of the

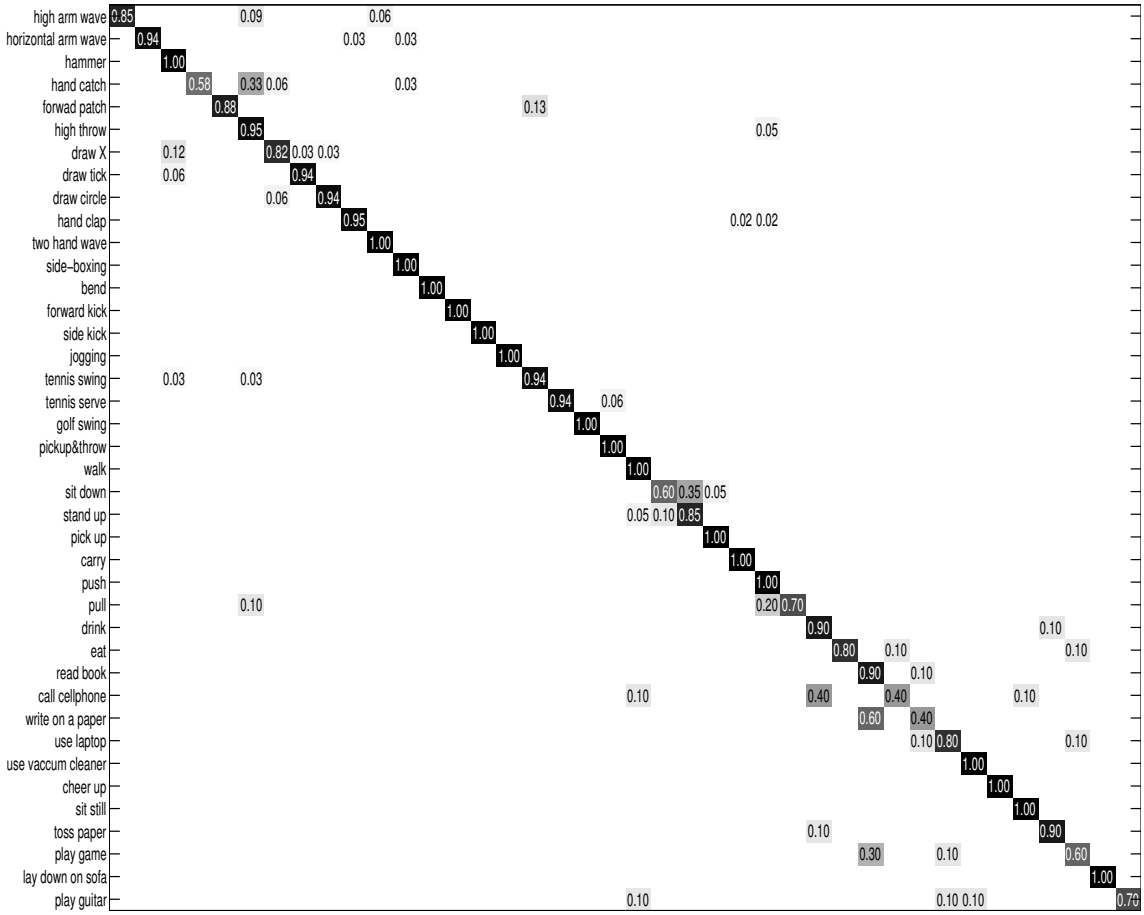

Figure 4.12: The confusion matrix of proposed method for Combined dataset.

performance of the proposed method on the individual datasets when trained using the Combined dataset.

The temporal scale was set to $n = 5$ and $\alpha = (2, 5)$ in the experiments. The proposed method was tested with four settings and the results are shown in Table 4.10. As expected, the strategy of using synthesized samples and multiple temporal scaling become less apparent in improving the overall performance. One probable reason is that the training of ConvNets has benefited from the large number of samples in the Combined dataset.

Table 4.10: Comparative performance of the proposed method based on the Combined dataset and with respect to the four training settings.

Training Setting	Accuracy ($\alpha = 2$)	Accuracy ($\alpha = 5$)
S3	87.20%	87.59%
S4	-	90.63%
S5	-	89.94%
S6	90.92%	91.56%

The performance of proposed method is compared with SNV [YT14] on this

Table 4.11: Comparative recognition accuracies of the SNV and proposed methods using the Combined dataset and its original datasets.

Dataset	Method		
	SNV	Proposed ($\alpha = 2$)	Proposed ($\alpha = 5$)
MSRAction3D	89.83%	94.58%	**94.92%**
MSRAction3DExt	91.15%	94.05%	**94.35%**
UTKinect-Action	**93.94%**	91.92%	92.93%
MSRDailyActivity3D	60.63%	78.12%	**80.63%**
Combined	86.11%	90.92%	**91.56%**

Table 4.12: Comparative performance of the SNV and proposed methods using individual and Combined dataset. Recognition accuracy and change of accuracy are reported.

Dataset	Method					
	SNV	SNV_c	η	$Proposed$	$Proposed_c$	η
MSRAction3D	93.09%	89.58%	3.77%	100.00%	94.92%	5.08%
MSRAction3DExt	90.54%	91.15%	0.67%	100.00%	94.35%	5.65%
UTKinect-Action	88.89%	93.94%	5.68%	90.91%	92.93%	2.22%
MSRDailyActivity3D	86.25%	60.63%	28.70%	85.00%	80.83%	4.91%

dataset in the following manner. A model is first trained over the Combined dataset and then tested on the original individual datasets and Combined dataset. Note that this was done according to the cross-subject evaluation scheme as described and, the training and testing samples were kept the same as when the methods were applied to individual datasets separately. The results and corresponding confusion matrices are shown in Tables 4.11, 4.12 and Fig. 4.12 respectively. In order to compare the performance of the methods on individual datasets and the combined case, the rate change, $\eta = \frac{|X_c - X|}{X} \times 100\%$ was calculated, where X and X_c denote respectively the accuracies when performing the training and recognition on individual datasets separately and on the combined dataset.

Table 4.11 and Table 4.12 shows that the proposed method can maintain the accuracy without a large drop while the number of actions increased and the dataset becomes more complex. In addition, it outperformed the SNV method on the Combined dataset.

4.1.3.6 Analysis

In the proposed method, ConvNets serve the purpose of feature extraction and classification. Generally, ConvNets require a large amount of data to tune millions of parameters to avoid overfitting. Directly training the ConvNets with a small set of data would lead to poor performance due to overfitting, and this has been

demonstrated in Table 4.1 and Table 4.3. However, the small amount (even for the Combined dataset) of available data can be compensated with data augmentation. In our method, two strategies are used for this purpose: synthesized viewpoints and temporal scaling with additional benefits of making the method viewpoint and speed tolerant respectively. However, without initializing the ConvNets with a model pre-trained over ImageNet, the artificially augmented data seems insufficient to train the nets. This is probably because the data synthesized from the original data do not contain the same amount of independent information as would have been captured by real cameras. Nonetheless, their contribution to the training is apparent as demonstrated in the experiments. In addition, the scheme of pre-training followed by fine-tuning provides a promising remedy for small datasets.

For each dataset, a different temporal scale was set to obtain the best results and the reasons are as follows. For simple actions (or gestures), such as MSRAction3D(Ext), one scale is sufficient to distinguish the differences between actions (gestures), due to their low motion complexity and short duration of motion. For activities, such as those in the UTKinect-Action and MSRDailyActivity3D datasets, more scales (e.g. 5) are needed, because the duration of the actions are long and each action usually contains several simple actions (gestures). Use of a large number (e.g. over 5) of scales can capture the motion information in different temporal scales. When we consider noisy samples, such as the MSRDailyActivity3D dataset, larger temporal scales (e.g. 21) should be set in order to suppress the effects of complex background in these datasets. However, the performance is not sensitive to the number of temporal scales and gain in performance by tuning the scale is rather marginal (around 2 percentage points).

For pseudo-coloring, the power α is in general set between 2 and 10 according the characteristic of noise. A large value of α can suppress the noise in areas having small or large WHDMM values. However, the performance gain over different α values is around 3 percentage points for the datasets used in this chapter.

4.1.3.7 Computational Cost

Table 4.13 compares the computational cost of SNV and proposed method on the MSRAction3D dataset. The dataset has 567 samples; 292 samples were used for training and the rest for testing. The average number of frames per sample is 42. The CPU platform used in the testing is a small HPC running CentOS6.5 with 2x Intel(R) Xeon(R) Processor E5-4620 at 2.20GHz and the GPU test platform is equipped with an NVIDIA Tesla K40 card. The SNV method was implemented in Matlab and executed on the CPU platform. The proposed method was implemented in C/C++ and much of its computation is performed by the GPU. It should be pointed out that the computational cost of the SVN method increases exponentially

with the number of frames whereas the computation cost of the proposed method increases linearly.

Table 4.13: Comparative computational cost of SNV and the proposed method based on MSRAction3D dataset.

Cost	Method	
	SNV	Proposed
Training (seconds)	22913 (CPU time)	667 (CPU time) + 2246 (GPU time)
Testing (seconds per sample)	76 (CPU time)	0.80 (CPU time) + 0.24 (GPU time)
Memory usage	16G RAM	4G video RAM + 4G RAM

4.2 Dynamic Depth Maps with ConvNets

4.2.1 Prior Works and Our Contributions

In our previous work, we applied ConvNets to depth action recognition based on the variants of DMM [YZT12], which is sensitive to noise and cannot work well with clutter background. Wu. et al. [WPK+16a] adopted a 3D ConvNet to extract features from depth data, which requires a large amount of training data to achieve the best performance. Compared to traditional RGB images, depth maps offer better geometric cues and less sensitivity to illumination changes for action recognition. In order to make full use of these properties and take advantages of ConvNets, we propose three simple, compact yet effective representations of depth sequences, referred to respectively as Dynamic Depth Images (DDI), Dynamic Depth Normal Images (DDNI) and Dynamic Depth Motion Normal Images (DDMNI), for both isolated and continuous action recognition. These dynamic images are constructed from a segmented sequence of depth maps using hierarchical bidirectional rank pooling to effectively capture the spatial-temporal information. Specifically, DDI exploits the dynamics of postures over time and DDNI and DDMNI exploit the 3D structural information captured by depth maps. Upon the proposed representations, a ConvNet based method is developed for action recognition. The image-based representations enable us to fine-tune the existing Convolutional Neural Network (ConvNet) models trained on image data without training a large number of parameters from scratch. The proposed method was evaluated on three large datasets, namely, the Large-scale Continuous Gesture Recognition Dataset, the Large-scale Isolated Gesture Recognition Dataset, and the NTU RGB+D Dataset. State-of-the-arts results were achieved on all datasets even though only the depth data was used.

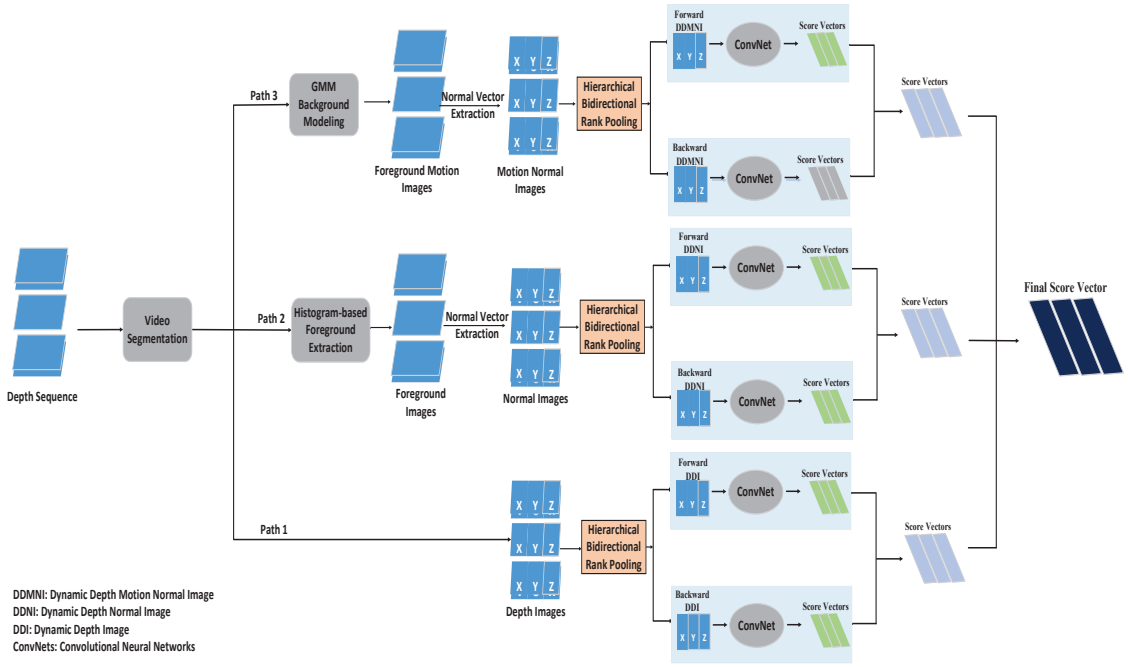

Figure 4.13: The framework of the proposed method.

4.2.2 The Proposed Methods

The proposed method consists of four stages: action segmentation, construction of the three sets of dynamic images, ConvNets training and score fusion for classification. The framework is illustrated in Fig. 4.13. Given a sequence of depth maps consisting of multiple actions, the *start* and *end* frames of each action are identified based on quantity of movement (QOM) [JZW+15]. Then, three sets of dynamic images are constructed for each action segment and used as the input to six ConvNets for product score fusion-based classification. Details are presented in the rest of this section.

4.2.2.1 Action Segmentation

Previous works on action recognition mainly focus on the classification of segmented actions. In the case of continuous recognition, both segmentation and recognition have to be solved. This chapter tackles the segmentation and classification of actions separately and sequentially.

Given a sequence of depth maps that contains multiple actions, each frame has the relevant movement with respect to its adjacent frame and the first frame. The *start* and *end* frames of each action is detected based on quantity of movement (QOM) [JZW+15] by assuming that all actions starts from a similar pose. For a multi-action depth sequence I, the QOM for frame t is defined as a two-dimensional

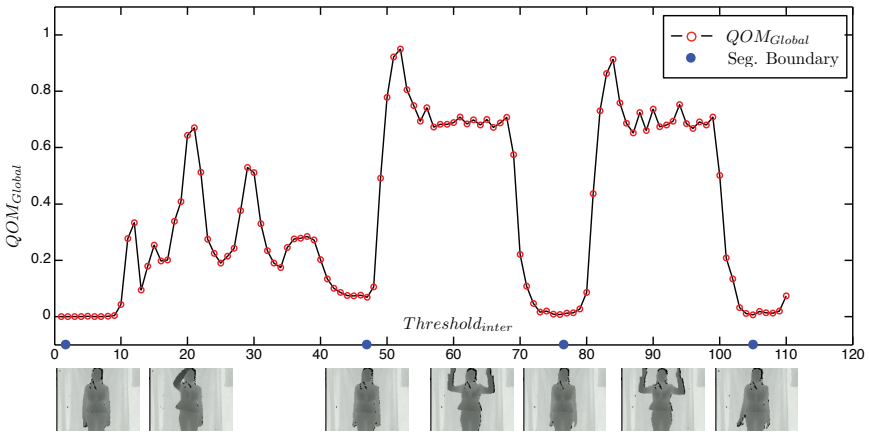

Figure 4.14: An example of illustrating the inter-action segmentation results. Figure from [JZW+15].

vector

$$QOM(I,t) = [QOM_{Local}(I,t), QOM_{Global}(I,t)], \tag{4.8}$$

where $QOM_{Local}(I,t)$ and $QOM_{Global}(I,t)$ measure the relative movement of frame t with respect to its adjacent frame and the first frame. They are defined as

$$\begin{aligned} QOM_{Local}(I,t) &= \sum_{m,n} \psi(I_t(m,n), I_{t-1}(m,n)) \\ QOM_{Global}(I,t) &= \sum_{m,n} \psi(I_t(m,n), I_1(m,n)) \end{aligned}, \tag{4.9}$$

where (m,n) is the pixel location and the indicator function $\psi(x,y)$ is defined as

$$\psi(x,y) = \begin{cases} 1 & if|x-y| \geqslant Threshold_{QOM}; \\ 0 & \text{otherwise} \end{cases}$$

$Threshold_{QOM}$ is a predefined threshold, which is set to 60 empirically in this chapter. A set of frame indices of candidate delimiting frames is initialized by choosing frames with lower global QOMs than a $threshold_{inter}$. The $threshold_{inter}$ is calculated by adding the mean to twice the standard deviation of global QOMs extracted from first and last 12.5% of the average action sequence length L which is calculated from the training actions. A sliding window with a size of $\frac{L}{2}$ is then used to refine the candidate set and in each windowing session only the index of frame with a minimum global QOM is retained. After the refinement, the remaining frames are expected to be the delimiting frames of actions, as shown in Fig. 4.14.

4.2.2.2 Construction of Dynamic Images

The three sets of dynamic images, Dynamic Depth Images (DDIs), Dynamic Depth Normal Images (DDNIs) and Dynamic Depth Motion Normal Images (DDMNIs) are constructed from a segmented sequence of depth maps through hierarchical bidirectional rank pooling. They aim to exploit shape, motion and structural information captured by a depth sequence at different spatial and temporal scales. To this end, the conventional ranking pooling [BFG+16] is extended to the hierarchical bidirectional rank pooling.

The conventional rank pooling [BFG+16] aggregates spatio-temporal information from one video sequence into one dynamic image. It defines a function that maps a video clip into one feature vector [BFG+16]. A *rank pooling function* is formally defined as follows.

Rank Pooling Let a depth map sequence with k frames be represented as $< d_1, d_2, ..., d_t, ..., d_k >$, where d_t is the average of depth features over the frames up to t-timestamp. At each time t, a score $r_t = \omega^T \cdot d_t$ is assigned. The score satisfies $r_i > r_j \Longleftrightarrow i > j$. In general, more recent frames are associated with larger scores. The process of rank pooling is to find ω^* that satisfies the following objective function:

$$\begin{aligned} \arg\min_{\omega} \frac{1}{2} \parallel \omega \parallel^2 + \lambda \sum_{i>j} \xi_{ij} \\ s.t. \;\; \omega^T \cdot (d_i - d_j) \geq 1 - \xi_{ij}, \xi_{ij} \geq 0 \end{aligned}, \tag{4.10}$$

where ξ_{ij} is a slack variable. Since the score r_i assigned to frame i is often defined as the order of the frame in the sequence, ω^* aggregates information from all of the frames in the sequence and can be used as a descriptor of the sequence. In this chapter, the rank pooling is directly applied on the pixels of depth maps and the ω^* is of the same size as depth maps and forms a dynamic depth image (DDI).

However, the conventional ranking pooling method has two drawbacks. Firstly, it treats a video sequence in a single temporal scale which is usually too shallow [FAHG16]. Secondly, since in rank pooling the averaged feature up to time t is used to classify frame t, the pooled feature is biased towards beginning frames of a depth sequence, hence, frames at the beginning has more influence to ω^*. This is not justifiable in action recognition as there is no prior knowledge on which frames are more important than other frames.

To overcome the first drawback, it is proposed that the ranking pooling is applied recursively to sliding windows over several *rank pooling layer*. This recursive process can effectively explore the high-order and non-linear dynamics of a depth sequence. The *rank pooling layer* is defined as follows:

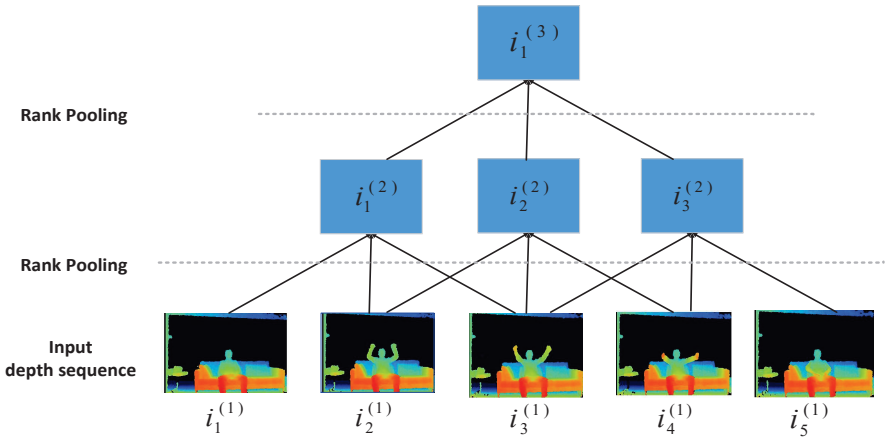

Figure 4.15: Illustration of a two layered rank pooling with window size three ($M_l = 3$) and stride one ($S_l = 1$).

Definition 2 (Rank Pooling Layer). Let $I^{(l)} = \left\langle i_1^{(l)}, ..., i_n^{(l)} \right\rangle$ denote the input sequence/subsequence that contains n frames; M_l is the window size; and S_l is a stride in the l_{th} layer. The subsequences of $I^{(l)}$ can be defined as $I_t^{(l)} = \left\langle i_t^{(l)}, ..., i_{t+M_l-1}^{(l)} \right\rangle$, where $t \in \{1,\ S_l + 1,\ 2S_l + 1, \ldots\}$. By applying the *rank pooling function* on the subsequences respectively, the outputs of l_{th} layer constitute the $(l+1)_{th}$ layer, which can be represented as $I^{(l+1)} = \left\langle \ldots, i_t^{(l+1)}, ... \right\rangle$.

$I^{(l)}$ to $I^{(l+1)}$ forms one layer of temporal hierarchy. Multiple *rank pooling layers* can be stacked together to make the pooling higher-order. In this case, each successive layer obtains the dynamics of the previous layer. Figure 4.15 shows a hierarchical rank pooling with two layers. For the first layer, the sequence is the input depth sequence, thus $l = 1$, $n = 5$; for the second layer, $l = 2$, $n = 3$. By adjusting the window size and stride of each layer, the hierarchical rank pooling can explore high-order and non-linear dynamics effectively.

To address the second drawback, it is proposed to to apply the rank pooling bidirectionally.

Bidirectional Rank Pooling is to apply the rank pooling forward and backward to a sequence of depth maps. In the forward rank pooling, the r_i is defined in the same order as the time-stamps of the frames. In the backward rank pooling, r_i is defined in the reverse order of the time-stamps of the frames. When bidirectional rank pooling is applied to a sequence of depth maps, two DDIs, forward DDI and backward DDI, are generated.

By employing the hierarchical and bidirectional pooling together, the hierarchical bidirectional rank pooling exploits the dynamics of a depth sequence at different temporal scales and bidirectionally at the same time. It has been empirically observed that, for most actions with relatively short durations, two layers of bidi-

rectional rank pooling is sufficient.

Construction of DDI

Given a segmented sequence of depth maps, the hierarchical bidirectional rank pooling method described above is employed directly on the depth pixels to generate two dynamic depth images (DDIs), forward DDI and backward DDI. Even though rank pooling method exploits the evolution of videos and aims to encode both the spatial and motion information into one image, it is likely to lose much motion information due to the insensitivity of depth pixels to motion. As shown in Fig. 4.16, DDIs effectively capture the posture information, similar to key poses. Moreover, compared with the dynamic images (DIs [BFG+16]), the DDIs are more effective, without having interfering texture on the body.

Construction of DDNI

Depth images well represent the geometry of surfaces in the scene, and norm vectors is sensitive to motion of depth pixels. In order to simultaneously exploit the spatial and motion information in depth sequences, it is proposed to extract normals from depth maps and construct the so-called DDNIs (dynamic depth normal images). For each depth map, a surface normal (n_x, n_y, n_z) is calculated at each pixel. Three channels (N_x, N_y, N_z), referred to as a Depth Normal Image, are generated from the normals, where (N_x, N_y, N_z) are respectively normal images of the three components (n_x, n_y, n_z). The sequence of each DNI goes through hierarchical bidirectional rank pooling to generate two DDNIs, one being the forward DDNI and the other is the backward DDNI.

To minimize the interference of the background, it is assumed that the background in the histogram of depth maps occupies the last peak representing far distances. Specifically, pixels whose depth values are greater than a threshold defined by the last peak of the depth histogram minus a fixed tolerance are considered as background and removed from the calculation of DDNIs by re-setting their depth values to zero. Through this simple process, most of the background can be removed and has much contribution to the DDNIs. Samples of DDNIs can be seen in Fig. 4.16.

Construction of DDMNI

The purpose of constructing a DDMNI is to further exploit the motion in depth maps. Gaussian mixture model (GMM) is applied to depth sequences in order to detect moving foreground. The norm vectors are extracted from the moving foreground and Depth Normal Image is constructed from the norm vectors for each

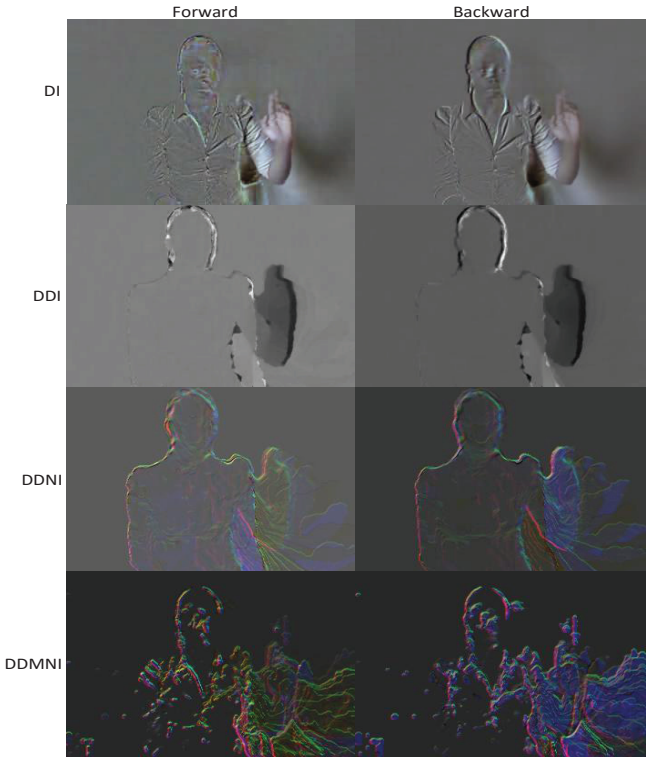

Figure 4.16: Samples of generated forward and backward DIs [BFG+16], DDIs, DDNIs and DDMNIs for gesture Mudra1/Ardhapataka.

depth map. Hierarchical bidirectional rank pooling is applied to the Depth Norm Image sequence, and two DDMNIs, forward DDMNI and backward DDMNI, are generated, which capture the motion information specifically well (see the illustration in Fig. 4.16).

4.2.2.3 Network Training

After the construction of DDIs, DDNIs and DDMNIs, there are six dynamic images, as illustrated in Fig. 4.16, for each depth map sequence. Six ConvNets were trained on the six channels individually. VGG-16 [SZ14b] is adopted in this chapter. The implementation is derived from the publicly available Caffe toolbox [JSD+14] based on three NVIDIA Tesla K40 GPU cards and one Pascal TITAN X.

The training procedure is similar to those in [SZ14b]. The network weights are learned using the mini-batch stochastic gradient descent with the momentum set to 0.9 and weight decay set to 0.0005. All hidden weight layers use the rectification (RELU) activation function. At each iteration, a mini-batch of 32 samples is constructed by sampling 256 shuffled training samples, and all the images are resized to 224×224. The learning rate is set to 10^{-3} for fine-tuning with pre-trained models on ILSVRC-2012, and then it is decreased according to a fixed schedule, which is kept the same for all training sets. The training undergoes 100 epochs and the learn-

ing rate decreases every 30 epochs for each ConvNet. The dropout regularization ratio is set to 0.9 to reduce complex co-adaptations of neurons in nets.

4.2.2.4 Score Fusion for Classification

Given a test depth video sequence (sample), three pairs of dynamic images (DDIs, DDNIs, DDMNIs) are generated and fed into six different trained ConvNets. For each image pair, product score fusion was used. The score vector output from the two pair of ConvNets are multiplied in an element-wise manner and the resultant score vectors are normalized using L_1 norm. The three normalized score vectors are then multiplied in an element-wise fashion and the max score in the resultant vector is assigned as the probability of the test sequence being the recognized class. The index of this max score corresponds to the recognized class label and expressed as follows:

$$label = Fin(max(v_1 \circ v_2 \circ v_3 \circ v_4 \circ v_5 \circ v_6)) \tag{4.11}$$

where v is a score vector, $\circ$ refers to element-wise multiplication and $Fin(\cdot)$ is a function to find the index of the element having the maximum score.

4.2.3 Experimental Results

In this section, the Large-scale Isolated and Continuous Gesture Recognition datasets at the ChaLearn LAP challenge 2016 (ChaLearn LAP IsoGD Dataset and ChaLearn LAP ConGD Dataset) [EPLW+16], the NTU RGB+D dataset [SLNW16], and the corresponding evaluation protocols and results & analysis are described. On ChaLearn LAP ConGD Dataset, action segmentation was first conducted to segment the continuous actions to isolated actions. For all the experiments, two layered hierarchical bidirectional rank pooling method is adopted, with window size $M_l = 3$ and stride step $S_l = 1$.

4.2.3.1 ChaLearn LAP IsoGD Dataset

The ChaLearn LAP IsoGD Dataset was adopted to evaluate the proposed method. In this chapter, only depth maps are used to evaluate the performance of the proposed method.

Table 4.14 shows the results of each channel. From the results we can see that DDIs achieved much better results than DDNIs and DDMNIs, and the reasons are as follows: first, the depth values are not the real depth, but they are normalized to [0,255], which distort the true 3D structure information and affects the norm vectors extraction; second, for storage benefit, the videos are compressed at a loss level, which leads to lots of compression blocking artifacts, which makes the extraction of

moving foreground and norm vectors very noisy. Even though, the three kinds of dynamic images still provide complimentary information to each other. In addition, it can be seen that the bidirectional rank pooling exploits more useful information compared to one-way rank pooling [BFG+16], and by adopting product score fusion method, the accuracy is largely improved. Moreover, hierarchical rank pooling encodes the dynamic of depth sequences better compared with the conventional rank pooling method.

Table 4.14: Comparative accuracy of the three set of dynamic images on the validation set of the ChaLearn LAP IsoGD dataset. RP denotes conventional rank pooling; HRP represents hierarchical rank pooling.

Method	Accuracy for RP	Accuracy for HRP
DDI (forward)	36.13%	36.92%
DDI (backward)	30.45%	31.24%
DDI (fusion)	37.52%	37.68%
DDNI (forward)	24.86%	25.02%
DDNI (backward)	24.58%	24.64%
DDNI (fusion)	29.26%	29.48%
DDMNI (forward)	24.81%	24.69%
DDMNI (backward)	23.14%	23.57%
DDMNI (fusion)	27.75%	27.89%
Fusion All	42.56%	43.72%

The results obtained by the proposed method on the validation and test sets are listed and compared with previous methods in Table 5.2. These methods include MFSK combined 3D SMoSIFT [WRL+14] with (HOG, HOF and MBH) [WS13] descriptors. MFSK+DeepID further included Deep hidden IDentity (Deep ID) feature [SWT14]. Thus, these two methods utilized not only handcrafted features but also deep learning features. Moreover, they extracted features from RGB and depth separately, concatenated them together, and adopted Bag-of-Words (BoW) model as the final video representation. The other methods, WHDMM+SDI [WLG+16, BFG+16], extracted features and conducted classification with ConvNets from depth and RGB individually and adopted product score fusion for final recognition. SFAM [WLG+17] adopted scene flow to extract features and encoded the flow vectors into action maps, which fused RGB and depth data from the onset of the process. C3D [LMT+16b] applied 3D convolutional networks to both depth and RGB channels and fused them in a late fusion method. Pyramidal 3D CNN [ZZM+16b] adopted 3D convolutional networks to pyramid input to recognize gesture from both clip videos and entire video. It is noteworthy that the results of the proposed method have been obtained using a single modality viz., depth data, while all compared methods are based on RGB and depth modalities. From this table, we can see that the proposed method outperformed all of these

recent works significantly, and illustrated its effectiveness.

Table 4.15: Comparative accuracy of proposed method and baseline methods on the ChaLearn LAP IsoGD dataset.

Method	Set	Recognition rate r
MFSK [WGL16]	Validation	18.65%
MFSK+DeepID [WGL16]	Validation	18.23%
SDI [BFG+16]	Validation	20.83%
WHDMM [WLG+16]	Validation	25.10%
Scene Flow [WLG+17]	Validation	36.27%
Proposed Method	Validation	**43.72%**
MFSK [WGL16]	Testing	24.19%
MFSK+DeepID [WGL16]	Testing	23.67%
Pyramidal 3D CNN [ZZM+16b]	Testing	50.93%
C3D [LMT+16b]	Testing	56.90%
Proposed Method	Testing	**59.21%**

Table 4.16: Accuracies of the proposed method and previous methods on the ChaLearn LAP ConGD dataset.

Method	Set	Mean Jaccard Index $\overline{J_S}$
MFSK [WGL16]	Validation	0.0918
MFSK+DeepID [WGL16]	Validation	0.0902
Proposed Method	Validation	**0.3905**
MFSK [WGL16]	Testing	0.1464
MFSK+DeepID [WGL16]	Testing	0.1435
IDMM + ConvNet [WLL+16a]	Testing	0.2655
C3D [CHKB16]	Testing	0.2692
Two-stream RNNs [CLY+16]	Testing	0.2869
Proposed Method	Testing	**0.4109**

4.2.3.2 ChaLearn LAP ConGD Dataset

The ChaLearn LAP ConGD Dataset was adopted to evaluate the proposed method. In this chapter, only depth data was used in the proposed method.

The results of the proposed method on the validation and test sets and their comparisons with the results of previous methods are shown in Table 5.3. MFSK and MFSK+DeepID [WGL16] methods first segmented the continuous videos to segments and then extracted the features over the segments over two modalities to train and classify the actions. IDMM + ConvNet [WLL+16a] also adopted the action segmentation method and then extracted one improved depth motion map using color coding method over the segments, and ConvNet was adopted to train and classify segmented actions. C3D [CHKB16] applied 3D convolutional networks to RGB

Table 4.17: Comparative accuracy of the three set of dynamic images on the NTU RGB+D Dataset. RP denotes conventional rank pooling; HRP represents hierarchical rank pooling.

Method	Cross subject Accuracy for RP	Cross subject Accuracy for HRP	Cross view Accuracy for RP	Cross view Accuracy for HRP
DDI (forward)	75.80%	76.10%	76.50%	76.75%
DDI (backward)	70.99%	75.45%	75.62%	75.48%
DDI (fusion)	81.66%	82.01%	81.53%	81.60%
DDNI (forward)	79.79%	79.98%	54.57%	55.01%
DDNI (backward)	81.46%	81.28%	56.61%	57.43%
DDNI (fusion)	84.18%	84.24%	61.07%	62.35%
DDMNI (forward)	68.89%	69.33%	50.01%	50.67%
DDMNI (backward)	70.04%	71.11%	49.53%	49.27%
DDMNI (fusion)	73.56%	74.27%	54.98%	55.09%
Fusion All	86.72%	87.08%	83.75%	84.22%

video and jointly learn the features and classifier. Two-stream RNNs [CLY$^+$16] first adopted R-CNN to extract the hand and then conducted temporal segmentation. Two-stream RNNs were adopted to fuse multi-modality features for final recognition based on segments. The results showed that the proposed method outperformed all previous methods largely, even though only single modality, i.e. depth data, was used.

4.2.3.3 NTU RGB+D Dataset

The large NTU RGB+D Dataset was used to evaluate the proposed method. It consists of front view, two side views and left, right 45 degree views. This dataset is challenging due to large intra-class and viewpoint variations. For fair comparison and evaluation, the same protocol as that in [SLNW16] was used. It has both cross-subject and cross-view evaluation. In the cross-subject evaluation, samples of subjects 1, 2, 4, 5, 8, 9, 13, 14, 15, 16, 17, 18, 19, 25, 27, 28, 31, 34, 35 and 38 were used as training and samples of the remaining subjects were reserved for testing. In the cross-view evaluation, samples taken by cameras 2 and 3 were used as training, while the testing set includes samples from camera 1.

Similarly to LAP IsoGD Dataset, we conducted several experiments to compare the three set of dynamic images using conventional rank pooling method and the proposed hierarchical bidirectional rank pooling method. The comparisons are shown in Table 5.6. From the Table it can be seen that compared with DDIs, DDNIs achieved much better results than DDI in cross-subject setting, due to the sensitivity of norm vectors to motion over real depth values. This justified the effectiveness of proposed depth norm images for rank pooling. However, due to the sensitivity of norm vectors to motion and view angles, in cross-view setting, much worse results were achieved for DDNIs and DDMNIs. From the final fusion results we can see that the three set of dynamic images exploit the shape and motion at different levels, and provide complimentary information to each other.

Table 5.7 lists the performance of the proposed method and those previous works. The proposed method was compared with some skeleton-based methods and depth-based methods previously reported on this dataset. We can see that the proposed method outperformed all the previous works significantly.

Table 4.18: Comparative accuracies of the proposed method and previous methods on NTU RGB+D dataset.

Method	Cross subject	Cross view
Lie Group [VAC14]	50.08%	52.76%
HBRNN [DWW15]	59.07%	63.97%
2 Layer RNN [SLNW16]	56.29%	64.09%
2 Layer LSTM [SLNW16]	60.69%	67.29%
Part-aware LSTM [SLNW16]	62.93%	70.27%
ST-LSTM [LSXW16]	65.20%	76.10%
ST-LSTM+ Trust Gate [LSXW16]	69.20%	77.70%
JTM [WLHL16]	73.40%	75.20%
HON4D [OL13]	30.56%	7.26%
SNV [YT14]	31.82%	13.61%
SLTEP [JCT$^+$17]	58 .22%	–
Proposed Method	**87.08%**	**84.22%**

4.3 Structured Images with ConvNets

4.3.1 Prior Works and Our Contributions

In our previous work, we proposed to adopt rank pooling method to encode depth map sequences into three kinds of dynamic images. However, our empirical study has demonstrated the rank pooling method is limited in the spatial domain. Due to the unsupervised learning process, the rank pooling method mainly encodes the salient global features in the temporal domain, without mining the discriminative motion patterns in both spatial and temporal domains simultaneously. It is also found that by applying the rank pooling method directly on the full body sequences, the small but discriminative motion information to recognize actions is usually suppressed by large motion, especially for these fine-grained actions where the local spatio-temporal sub-volume motion is more important compared with the global motion of the whole sequences. As shown in Figure 4.17, the action "play game" from the MSRDailyActivity3D dataset, the large interference of body swaying motion occupies the motion in structured body DDI, and hands motion which is essential for recognition is not well highlighted in the DDI.

To address this problem, this chapter proposes to apply rank pooling method on depth map sequences at three hierarchical spatial levels, namely, body level, part level and joint level based on our proposed non-scaling method. Different from

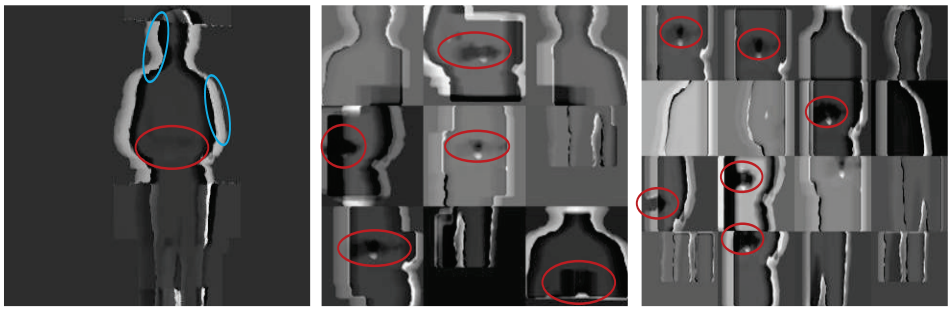

Figure 4.17: The three hierarchical structured DDIs for action "play game" from the MSRDailyActivity3D Dataset [WLWY12]. From left to right: structured body DDI, structured part DDI and structured joint DDI. The red circle denotes the hand motion need to be recognized while the blue one represents the large body swaying motion.

previous method [CLS15] that adopted one ConvNet for each human body part, it is proposed to construct one structured dynamic depth image as the input of a ConvNet for each level such that the structured dynamic images not only preserve the spatial-temporal information but also enhance the structure information, i.e. the coordination and synchronization of body parts over the period of the action. Such construction requires low computational cost and memory requirement. This representation, referred to as Spatially Structured Dynamic Depth Images (S^2DDI), aggregates motion and structure information from global to fine-grained levels for action recognition. In this way, the interference of large motion with small motion can be minimized. As shown in Figure 4.17, for action "play game", in the structured part DDI and structured joint DDI, the small hand motion is easy to recognize compared with that in structured body DDI. Moreover, the three structured dynamic images are complementary to each other, and an effective product score fusion method is adopted to improve the final recognition accuracy. The proposed image-based representation can take advantage of the available pre-trained models for standard ConvNet architectures without training millions of parameters afresh. It is evaluated on five benchmark datasets, namely, MSRAction3D [LZL10], G3D [BMA12], MSRDailyActivity3D [WLWY12], SYSU 3D HOI [HZLZ15] and UTD-MHAD [CJK15], and achieves the state-of-the-art results.

The key contributions of this method are four folds. (1) A simple yet effective video representation, S^2DDI, is proposed for RGB-D video based action recognition by constructing three level structured dynamic depth images through bidirectional rank pooling. (2) An efficient non-scaling method is proposed to construct the S^2DDI. (3) The three level structured dynamic images aggregate motion and structure information from global to fine-grained levels for action recognition. A product score fusion method is adopted to improve the final action recognition accuracy. (4)

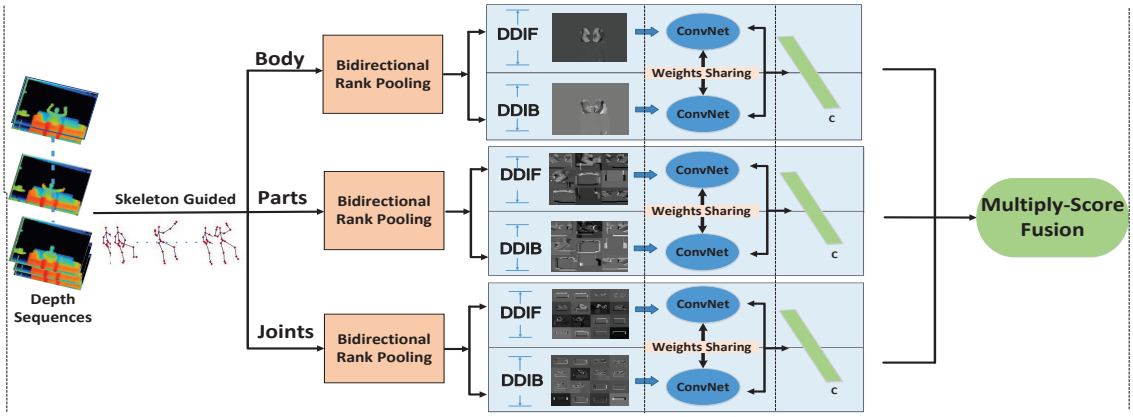

Figure 4.18: The framework of proposed method for action recognition using structured images.

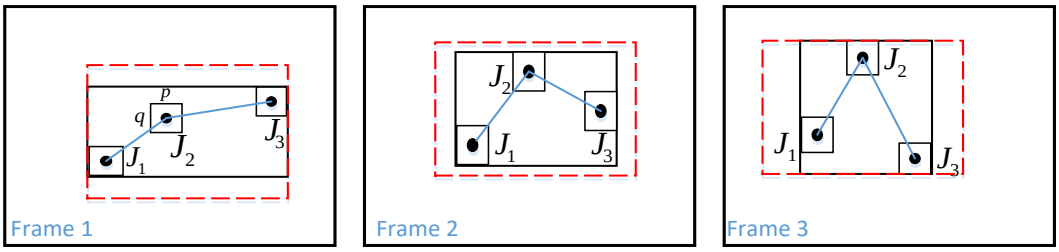

Figure 4.19: Illustration of non-scaled component patches of a component consisted of three joints $\{J_1, J_2, J_3\}$ from three frames. The solid black boxes are the bounding boxes of the component in each frame, while the dashed red box is the sequence-based bounding box of the component.

The proposed method achieves state-of-the-art results on five benchmark datasets.

4.3.2 The Proposed Methods

The proposed method mainly consists of three phases, as illustrated in Figure 4.18, the constructions of S^2DDI guided by skeletons, three weights-shared ConvNets training and product score fusion for final action recognition. The first phase is an unsupervised learning process. It applies bidirectional rank pooling method to three hierarchical levels of a depth sequence to generate the structured DDIs, with each level of DDIs being represented by two motion images, forward (DDIF) and backward (DDIB). In the following sections, the three phases will be described in detail. The rank pooling method [BFG+16], that aggregates spatio-temporal information from one video sequence into one dynamic image, is also briefly summarized.

4.3.2.1 Construction of S^{2}DDI

In the construction of S^{2}DDI, a human body is processed hierarchically at three spatial levels, namely, joint level, part level and body level. At each level, the body is divided into several components, and each component is composed of several joints. Specifically in this chapter, there are 16 components at the joint level, each component containing 1 joint; at body part level, there are 9 components, each component consisting of 3 joints as defined below; at body level, the entire body is treated as a single component consisting of 16 joints. For each component, a Dynamic Depth Images (DDI) is generated by applying the rank pooling forward or backward to a sequence of depth patches that encloses the component. Two DDIs, i.e. DDIF and DDIB, at each level are constructed by simply stitching their component DDIs in a predefined arrangement. The three DDIFs and three DDIBs at body, part and joint levels together are referred to as S^{2}DDI. Note the rank pooling requires that the frames in a depth patch sequences be of same size.

Let $C = \{j_1, j_2, \ldots, j_n\}$ be a component consisting of n joints. Centered at each joint in the image plane, a depth patch, referred to as a joint patch, of size $p \times q$ pixels is cropped. A patch for the component C at frame t is extracted from the depth map based on the bounding box of C by keeping the depth values inside the joint patches and setting depth values outside of the joint patches but within the bounding box to zero. Notice that size of the component bounding box varies from frame to frame due to movement of the joints on one hand and, on the other hand, rank pooling requires the same size of the component patches over a sequence. Conventionally, the component patches would be scaled to a same size, referring to as *scaled patches*. The obvious disadvantage of such scaling is the distortion of the spatial information within a frame and, hence, motion information over the sequence. It is proposed in this chapter to define a sequence-based component bounding box that is able to enclose the instances of the component over the sequence instead of using the bounding box at each frame. A component patch at each frame is then extracted by centering the sequence-based bounding box onto the component in the frame, referring to as *non-scaled patches*. In this way, the spatial and temporal distortion due to scaling can be eliminated. Figure 4.19 illustrates the extraction of non-scaling patches of a component consisting of three joints $\{J_1, J_2, J_3\}$ from three frames. In the figure, the solid black boxes are the bounding boxes of the component in each frame, while the dashed red box is the sequence-based bounding box of the component.

For the structured body DDIs, all the 20 joints are included in a single component. For the structured part DDIs, 9 components are defined according to the joint configuration in Figure 3.2 as follows.

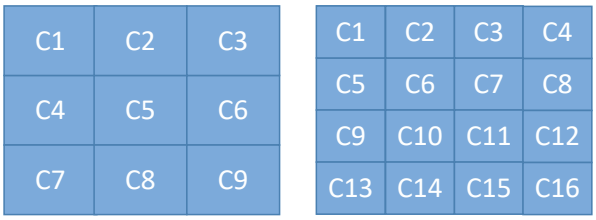

Figure 4.20: Stitching of component DDIs to a structured part DDI (left) and structured joint DDI (right).

C1	head,shoulder center,shoulder left
C2	head,shoulder center,shoulder right
C3	elbow left,wrist left,hand left
C4	elbow right,wrist right,hand right
C5	spine,hip center,hip right
C6	spine,hip center,hip left
C7	knee left,ankle left,foot left
C8	knee right,ankle right,foot right
C9	shoulder left,shoulder center,shoulder right

For the structured joint DDIs, the following 16 out of the 20 joints which usually bear relatively small noise are used and each joint forms a component.

hip center	spine	shoulder center	head
shoulder left	elbow left	hand left	shoulder right
elbow right	hand right	hand left	knee left
foot left	hip right	knee right	foot right

Different from the work in [CLS15] that adopted one ConvNet for each component, all component DDIs at the part level are stitched together to form a structured part DDI and the component DDIs at the joint level are stitched together to form a structured joint DDI as shown in Figure 4.20. Such arrangement of component DDIs into a single structured DDI at each spatial level enables ConvNets to explore more effectively the structured information of an action than any late fusion approach.

4.3.2.2 Network Training

After the construction of structured DDIs at three levels, there are six dynamic images for each depth map sequence, as illustrated in Figure 4.18. Three ConvNets are trained on the three kinds of DDIs individually. The AlexNet [KSH12] is adopted in this chapter. The network weights are learned using the mini-batch stochastic gradient descent with the momentum being set to 0.9 and weight decay being set to

0.0005. All hidden weight layers use the rectification (RELU) activation function. At each iteration, a mini-batch of 256 samples is constructed by sampling 256 shuffled training samples. All the images are resized to 256×256. The learning rate is set to 10^{-3} for fine-tuning the pre-trained models on ILSVRC-2012, and then it is decreased according to a fixed schedule, which is kept the same for all training sets. For each ConvNet, the training undergoes 3K iterations and the learning rate decreases every 1K iterations. For all experiments, the dropout regularization ratio is set to 0.5 in order to reduce complex co-adaptations of neurons in the nets.

4.3.2.3 Product Score Fusion for Classification

Given a test depth video sequence (sample), three pairs of dynamic images (structured body DDIs, structured part DDIs and structured joint DDIs) are generated and fed into three different trained ConvNets. For each image pair, product score fusion is used. The score vectors outputted by the weight sharing ConvNets are multiplied in an element-wise way, and then the resultant score vectors are normalized using L_1 norm. The three normalized score vectors are then multiplied in an element-wise fashion and the max score in the resultant vector is assigned as the probability of the test sequence. The index of this max score corresponds to the recognized class label.

4.3.3 Experimental Results

The proposed method is evaluated on five widely used benchmark RGB-D datasets [ZLO+16a], namely, MSRAction3D [LZL10], G3D [BMA12], MSRDailyActivity3D [WLWY12], SYSU 3D HOI [HZLZ15] and UTD-MHAD [CJK15] datasets. These five datasets cover a wide range of different types of actions including simple actions, actions for gaming, daily activities, human-object interactions and fine-grained activities. For the experiments on all datasets, the offset parameters (p, q) are empirically set. Specifically, for the construction of structured body DDI, they are $(80, 30)$ for head, two feet and two hands, and $(80, 50)$ for other joints. For structured part DDI, they are fixed to $(30, 30)$ for all joints. For structured joint DDI, they are set to be $(20, 30)$. In the following, the merit of applying rank pooling method to depth is first compared with raking pooling on RGB, the effectiveness of using non-scaled patches in the construction of DDIs and the product score fusion method is then demonstrated. Finally, the results on the five datasets are presented and the detailed analysis on MSRDailyActivity3D Dataset are described. The detailed analysis based on the confusion matrices for the other four datasets are described in the supplementary material.

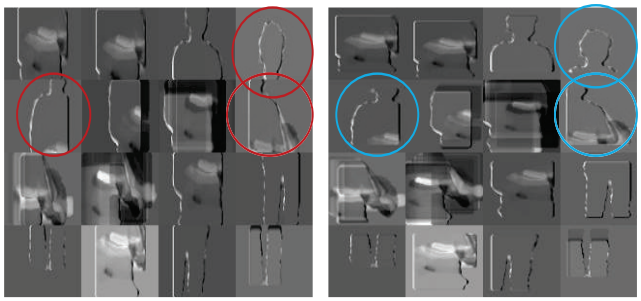

Figure 4.21: Illustration of using scaled component patches (left) and non-scaled component patches (right) for action "write on a paper" from MSRDailyActivity3D Dataset [WLWY12] for construction of structured joint DDI. The red circle denotes spatial distortion among human body while the blue one represents the preservation of aspect ratio among the parts and joints.

4.3.3.1 Effects of Design Choices

DDI vs. DI

Table 4.19 compares the performance of body DDI from depth and DI [BFG+16] from RGB for action recognition on the MSRDailyActivity3D dataset. Three DDIs are generated, one without foreground extraction, one using bounding box as foreground extraction, and the last one using the proposed method. From the results it can be seen that the DDI, especially the proposed structured body DDI, achieves much better results than DI. This verifies that the proposed method is robust to the noise in skeleton data.

Table 4.19: Comparison of DDI and DI on the MSRDailyActivity3D dataset.

Method	Accuracy
DI [BFG+16]	52.13%
DDI (without foreground extraction)	53.01%
DDI (with foreground bounding box)	58.75%
Structured body DDI (proposed)	61.00%

Scaled vs. Non-Scaled Component Patches in Construting DDI

Experiments are conducted to evaluate on the S^2DDI constructed using scaled and non-scaled component patches. Table 4.20 shows the comparisons of these two methods in terms of recognition accuracy. It can be seen that using non-scaled patches greatly outperforms using scaled-patches mainly due to the elimination of distortion induced by the scaling.

Table 4.20: Comparison of Construction of S^2DDI using scaled and non-scaled component patches on the MSRDailyActivity3D dataset.

Method	Accuracy
Structured part DDI (scaled)	67.88%
Structured joint DDI (scaled)	85.15%
S^2DDI (scaled)	87.04%
Structured part DDI (non-scaled)	81.88%
Structured joint DDI (non-scaled)	93.13%
S^2DDI (non-scaled)	97.50%

Structured Images vs. Channel Fusion

To verify the effectiveness of proposed structured images, taking part level from MSRDailyActivity3D dataset for example, we compared the structured images with channel fusion using ConvNets and SIFT+FV+SVM [GWZZ17], as in Table 4.21. It can be seen that the proposed structured part DDI not only outperforms the fusion of 9 separate DDIs, but also has computational advantage (1 channel vs. 9 channels). This is probably because the structural information is explored by the ConvNet from the structured part DDI. But such structural information can hardly be explored if each DDI is input to separate ConvNets and fused at the score level. From the comparisons we can also see that the proposed method can take advantages of the pre-trained models over ImagesNet for recognition compared with the traditional classifiers (e.g. SVM).

Table 4.21: Comparison of structured images and channel fusion on the MSR-DailyActivity3D dataset.

Method	Acc
Structured part DDI (ConvNet)	81.88%
Structured part DDI (SIFT+FV+SVM)	76.25%
9 channel part DDIs fusion (ConvNet)	72.81%
9 channel part DDIs fusion (SIFT+FV+SVM)	71.88%

Traditional Rank pooling vs. Bidirectional Rank Pooling

Traditional pooling emphasizes the earlier frames in the pooling segment more than later frames. One of the key motivations of bidirectional rank pooling is to overcome this so that reversing cyclic movement patterns can be well distinguished. In addition, it effectively arguments the training data. The effectiveness of bidirectional rank pooling is shown in Table 4.22, taking MSRDailyActivity3D dataset for example.

Table 4.22: Comparison of traditional rank pooling and bidirectional rank pooling on the MSRDailyActivity3D dataset.

Method	body DDI	part DDI	joint DDI	fusion
Traditional rank pooling(SIFT+FV+SVM)	42.50%	68.75%	80.00%	86.25%
Traditional rank pooling(ConvNets)	59.38%	80.00%	89.37%	95.63%
Bidirectional rank pooling(SIFT+FV+SVM)	49.69%	76.25%	81.25%	88.75%
Bidirectional rank pooling(ConvNets)	61.00%	81.88%	93.130%	97.50%

Product vs. Average vs. Max Score Fusion

This chapter adopts product score fusion method to improve the final accuracy on the three structured DDIs. The other two commonly used late score fusion methods are average and maximum score fusion. The comparisons among the three late score fusion methods are shown in Table 5.10. It can be seen that the product score fusion method achieves the best results on all the five datasets. This verifies that the three structured DDIs are likely to be statistically independent and carry complementary information.

Table 4.23: Comparison of three different late score fusion methods on the five datasets.

Dataset	Score Fusion Method		
	Max	Average	Product
MSRAction3D	93.67%	97.56%	**100%**
G3D	94.83%	94.83%	**96.05%**
MSRDailyActivity3D	93.75%	95.00%	**97.50%**
SYSU 3D HOI	91.25%	94.17%	**95.42%**
UTD-MHAD	87.44%	88.54%	**89.04%**

4.3.3.2 MSRAction3D Dataset

The MSRAction3D Dataset [LZL10] was adopted to evaluate the proposed method. The same experimental setting adopted in [WLWY12] is followed, namely, the cross-subjects settings: subjects 1, 3, 5, 7, 9 for training and subjects 2, 4, 6, 8, 10 for testing. Table 4.24 lists the performance of the proposed method, as well as the results of several methods reported in recent three years. From the results, we can see that the proposed method can well recognize the simple actions, because the three hierarchical spatial dynamic image patches generated via bidirectional rank pooling can aggregate rich spatio-temporal information in each level, and the structure information of human body is explicitly exploited by the proposed non-scaled component patches and structured motion images.

Table 4.24: Comparison of the proposed method with existing methods on the MSRAction3D dataset.

Method	Accuracy
Lie Group [VAC14]	89.48%
HCM [LCNS16]	93.00%
SNV [YT14]	93.09%
Range Sample [LJT14]	95.62%
MTDMM + FV [CLZ+16]	95.97%
Structured body DDI	79.18%
Structured part DDI	83.83%
Structured joint DDI	95.40%
S^2DDI	**100%**

4.3.3.3 G3D Dataset

Gaming 3D Dataset (G3D) [BMA12] was adopted to evaluate the proposed method. For this dataset, the first 4 subjects are used for training, the fifth for validation and the remaining 5 subjects for testing, following the configuration in [NWJ15]. Table 4.25 compares the performance of the proposed method with that reported in [NWJ15, WLHL16]. It can been seen that S^2DDI achieves better results.

Table 4.25: Comparison of the proposed method with previous methods on the G3D dataset.

Method	Accuracy
LRBM [NWJ15]	90.50%
JTM [WLHL16]	94.24%
Structured body DDI	74.81%
Structured part DDI	89.97%
Structured joint DDI	93.62%
S^2DDI	**96.05%**

4.3.3.4 MSRDailyActivity3D Dataset

The MSRDailyActivity3D Dataset [WLWY12] was used to evaluate the proposed method. Most activities in this dataset involve human-object interactions. The same cross-subject experimental setting as in [WLWY12] is adopted. Compared with existing methods on this dataset, the results in Table 4.26 show that the proposed method is superior for dataset having fine-grained human-object interaction actions.

The confusion matrices for structured body DDI, structured part DDI and structured joint DDI are shown in Figure 4.22 and S^2DDI in Figure 4.23. From the confusion matrix, we can see that the structured body DDI confuses most activities, especially “Eat”, “Read book”, “Write on a paper” and “play game”. This is because

Table 4.26: Comparison of the proposed method with previous methods on the MSRDailyActivity3D dataset.

Method	Accuracy
IPM [ZNH+15]	83.30%
WHDMMs+ConvNets [WLG+16]	85.00%
SNV [YT14]	86.25%
DS+DCP+DDP+JOULE-SVM [HZLZ15]	95.00%
Range Sample [LJT14]	95.63%
MFSK+BoVW [WGL16]	95.70%
Structured body DDI	61.00%
Structured part DDI	81.88%
Structured joint DDI	93.13%
S^2DDI	**97.50%**

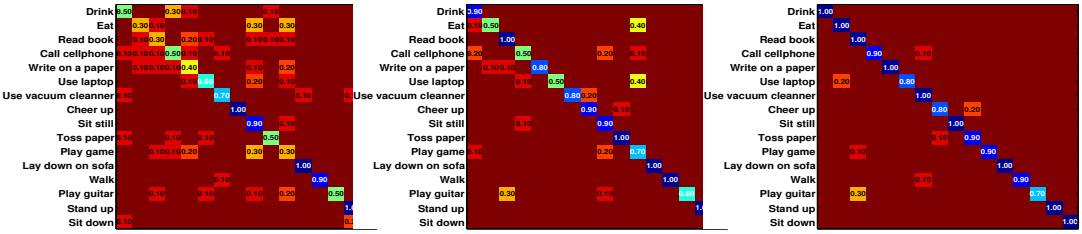

Figure 4.22: Confusion matrix for structured body DDI (left), structured part DDI (middle) and structured joint DDI (right) on MSRDailyActivity3D Dataset.

the structured body DDIs of these activities have similar shapes, and the motion to be recognized is very small compared with the interference of large body swaying motion, as illustrated in Figure 4.17. But as the granularity increases, most of the activities can be well recognized, because the fine-grained small motion is enhanced in the patches of parts and joints. By fusion of the three levels, most of the activities are better recognized, which reflects that the three structured motion images are complementary to each other. Compared with the method proposed in [HZLZ15], ours can better recognize "Drink", "Read book","Write on a paper" and "Play game" activities, due to the capability of both global to fine-grained motion and structure information aggregation of our method. These activities are very easily confused by global motion information aggregation method. However, the skeleton guided decomposition can not work well for human-large object interaction. For example, due to the large size of guitar, the proposed method loses much object information and confused "play guitar" with "Read book". This can be improved by setting larger extension around the joints.

4.3.3.5 SYSU 3D HOI Dataset

The SYSU 3D Human-Object Interaction Dataset (SYSU 3D HOI Dataset) [HZLZ15] was adopted to evaluate the proposed method. Table 5.9

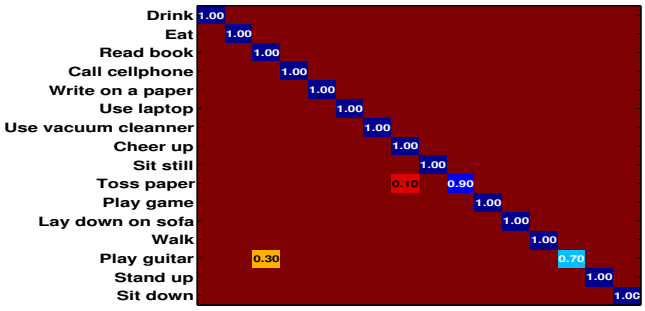

Figure 4.23: Confusion matrix for S^{2}DDI on the MSRDailyActivity3D dataset.

compares the performances of the proposed method and that of existing methods on this dataset using cross-subject settings as in [HZLZ15]. It can bee seen that, our proposed method outperforms previous methods largely. It should be noticed that on this dataset, the structured joint DDI achieves the best performance. From the confusion matrices in the supplementary material we can see that the "Taking from wallet" action is greatly confused in structured body and part DDIs, that affects the final performance of S^{2}DDI.

Table 4.27: Comparison of the proposed method with previous approaches on SYSU 3D HOI Dataset.

Method	Accuracy
HON4D [OL13]	79.22%
DS+DCP+DDP+MTDA [ZY11]	84.21%
DS+DCP+DDP+JOULE-SVM [HZLZ15]	84.89%
structured body DDI	65.00%
structured part DDI	85.83%
structured joint DDI	**95.83%**
S^{2}DDI	95.42%

4.3.3.6 UTD-MHAD Dataset

UTD-MHAD [CJK15] was adopted to evaluate the proposed method. For this dataset, cross-subjects protocol is adopted as in [CJK15], namely, the data from the subject numbers 1, 3, 5, 7 used for training while 2, 4, 6, 8 used for testing. The results are shown in Table 4.28. It can be seen that even the structured joint DDI itself can achieve better result than previous methods. From the performances on the five datasets, we can conclude that as the granularity increases, the proposed method achieves higher accuracy.

Table 4.28: Comparison of the proposed method with previous approaches on UTD-MHAD Dataset.

Method	Accuracy
WHDMMs+ConvNets [WLG+16]	73.95%
ELC-KSVD [ZLZ+14]	76.19%
Kinect & Inertial [CJK15]	79.10%
Cov3DJ [HTGES13]	85.58%
JTM [WLHL16]	85.81%
structured body DDI	66.05%
structured part DDI	78.70%
structured joint DDI	86.81%
S^2DDI	**89.04**%

4.4 Summary

In this section, we proposed three methods to address research questions 3, 4 and 5 listed in Section 1.2. Based on depth map sequences, we studied the strategies to apply ConvNets to small training data for action recognition. In order to make full use of the properties of depth and take advantages of ConvNets, three simple, compact yet effective image-based representations of depth sequences were proposed for large-scale action recognition. Structured images were further proposed to capture the spatial-temporal-structural information contained in the depth sequences, and aggregate motion and structure information from global to fine-grained levels for action recognition. State-of-the-art results were achieved on both small datasets and large datasets.

Chapter 5

RGB and Depth based Action Recognition

Recognition of human actions from RGB-D data has generated renewed interest in the computer vision community due to the recent availability of easy-to-use and low-cost depth sensors (e.g. Microsoft Kinect ™sensor). In addition to tristimulus visual data captured by conventional RGB cameras, depth data are provided in RGB-D cameras, thus encoding rich 3D structural information of the entire scene. How to adopt these two modalities together for action recognition is attracting more and more attention. This chapter studied this issue by addressing the research question 6 and 7 (Section 1.2): how to fuse the depth and RGB modalities at data-level and how to cooperatively train a single networks using these two heterogeneous inputs.

5.1 Scene Flow with ConvNets

5.1.1 Prior Works and Our Contributions

Previous works [NWJ15, KF15, HZLZ15, WZSS15, YLS16, JF16] showed the effectiveness of fusing the two modalities for 3D action recognition. However, all the previous methods consider the depth and RGB modalities as separate channels from which to extract features and fuse them at a later stage for action recognition. Since the depth and RGB data are captured simultaneously, it will be interesting to extract features considering them jointly as one entity. Optical flow-based methods for 2D action recognition [WS13, LLL$^+$15, PZQP14, PWWQ16, WQT15] have provided the state-of-the-art results for several years. In contrast to optical flow which provides the projection of the scene motion onto the image plane, scene flow [VRCK05, HB14, MG15, HFR14, JSGJC15, SSP15, QBDC14] estimates the actual 3D motion field. Thus, we propose the use of scene flow for 3D action recognition. Differently from the optical flow-based late fusion methods on RGB and depth data, scene flow extracts the real 3D motion and also explicitly preserves the spatial structural information contained in RGB and depth modalities.

There are two critical issues that need to be addressed when adopting scene flow for action recognition: how to organize the scene flow vectors and how to effectively exploit the spatio-temporal dynamics. Two kinds of motion representations can be identified: Lagrangian motion [WS13, LLL$^+$15, PZQP14, WQT15,

PWWQ16, WLHL16] and Eulerian motion [BD01b, MB06, YZT12, WLG+15, WLG+16, BFG+16]. Lagrangian motion focuses on individual points and analyses their change in location over time. Such trajectories requires reliable point tracking over long term and is prone to error. Eulerian motion considers a set of locations in the image and analyses the changes at these locations over time, thus avoiding the need for point tracking.

Since scene flow vectors could be noisy and to avoid the difficulty of long term point tracking of Lagrangian motion, we adopted the Eulerian approach in constructing the final representation for action recognition. Furthermore, the scene flow between two consecutive pair of RGB-D frames (two RGB images and two corresponding depth images) is one simple Lagrangian motion with only two frames matching/tracking. This property provides a better representation than is possible with Eulerian motion obtained from raw pixels.

However, it remains unclear as to how video could be effectively represented and fed to deep neural networks for classification. For example, one can conventionally consider a video as a sequence of still images with some form of temporal smoothness, or as a subspace of images or image features, or as the output of a neural network encoder. Which one among these and other possibilities would result in the best representation in the context of action recognition is not well understood. The promising performance of existing temporal encoding works [WLG+15, WLG+16, WLHL16, BFG+16] provides a source of motivation. These works encode the spatio-temporal information as dynamic images and enable the use of existing ConvNets models directly without training the whole networks afresh. Thus, we propose to encode the RGB-D video sequences based on scene flow into one motion map, called Scene Flow to Action Map (SFAM), for 3D action recognition. Intuitively and similarly to the three channels of color images, the three elements of a scene flow vector can be considered as three channels. Such consideration allows the scene flow between two consecutive pairs of RGB-D frames to be reorganized as one three-channel Scene Flow Map (SFM), and the RGB-D video sequence can be represented as SFM sequence. In the spirits of Eulerian motion and rank pooling methods [FGM+16, BFG+16], we propose to encode SFM sequence into SFAM. Several variants of SFAM are developed. They capture the spatio-temporal information from different perspectives and are complementary to each other for final recognition. However, two issues arise with these hand-crafted SFAMs: 1) direct organization of the scene flow vectors in SFM may sacrifice the relations among the three elements; 2) in order to take advantage of available model trained over ImageNet, the input needs to be analogous to RGB images; that is, the input for the ConvNets need to have similar properties to conventional RGB images as used in trained filters. Based on these two observations, we propose to

learn Channel Transform Kernels with rank pooling method and ConvNets, that convert the three channels into suitable three new channels capable of exploiting the relations among the three elements and have similar RGB image features. With this transformation, the dynamic SFAM can describe both the spatial and temporal information of a given video. It can be used as the input to available and already trained ConvNets along with fine-tuning.

The contributions of this chapter are summarized as follows:1) The proposed SFAM is the first attempt, to our best knowledge, to extract features from depth and RGB modalities as joint entity through scene flow, in the context of ConvNets; 2) we propose an effective self-calibration method that enables the estimation of scene flow from unregistered captured RGB-D data; 3) several variants of SFAM that encode the spatio-temporal information from different aspects and are complementary to each other for final 3D action recognition are proposed; 4) we introduce Channel Transform Kernels which learn the relations among the three channels of SFM and convert the scene flow vectors to RGB-like images to take advantages of trained ConvNets models and 5) the proposed method achieved state-of-the-art results on two relatively large datasets.

5.1.2 The Proposed Methods

SFAM encodes the dynamics of RGB-D sequences based on scene flow vectors. To make our description self-contained, in Section 5.1.2.1 we briefly present the primal-dual framework for real-time dense RGB-D scene flow computation (hereafter denoted by PD-flow [JSGJC15]). For scene flow computation, we assume that the depth and RGB data are prealigned. If this is not the case, the videos can be quickly realigned as described in Section 5.1.2.2. Then, in Section 5.1.2.3 we present several hand-crafted constructions of SFAM and we propose an end-to-end learning method for SFAM through Channel Transform Kernels in Section 5.1.2.4.

5.1.2.1 PD-flow

The PD-flow estimates the dense 3D motion field of a scene between two instants of time t and $t+1$ using RGB and depth images provided by an RGB-D camera. This motion field $\mathbf{M} : (\Omega \in \mathbb{R}^2) \rightarrow \mathbb{R}^3$ defined over the image domain Ω, is described with respect to the camera reference frame and expressed in meters per second. For simplicity, the bijective relationship $\Gamma : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ between $\mathbf{M}$ and $\mathbf{s} = (\mu, \upsilon, \omega)^T$ is given by:

$$\mathbf{M} = \Gamma(\mathbf{s}) = \begin{pmatrix} \frac{Z}{f_x} & 0 & \frac{X}{Z} \\ 0 & \frac{Z}{f_y} & \frac{Y}{Z} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mu \\ \upsilon \\ \omega \end{pmatrix}, \tag{5.1}$$

where μ, υ represent the optical flow and ω denotes the range flow; f_x, f_y are the camera focal length values, and X, Y, Z the spatial coordinates of the observed point. Thus, estimating the optical and range flows is equivalent to estimating the 3D motion field but leads to a simplified implementation. In order to compute the motion field a minimization problem over $\mathbf{s}$ is formulated where photometric and geometric consistency are imposed as well as a regularity of the solution:

$$\min_{\mathbf{s}}\{E_D(\mathbf{s}) + E_R(\mathbf{s})\}. \tag{5.2}$$

In Eq. (5.2), $E_D(\mathbf{s})$ is the data term, representing a two-fold restriction for both intensity and depth matching between pairs of frames; $E_R(\mathbf{s})$ is the regularization term which both smooths the flow field and constrains the solution space.

For data term $E_D(\mathbf{s})$, the L_1 norm of photometric consistency $\rho_I(\mathbf{s}, x, y)$ and geometric consistency $\rho_z(\mathbf{s}, x, y)$ is minimized as:

$$E_D(\mathbf{s}) = \int |\rho_I(\mathbf{s}, x, y)| + \varepsilon(x, y)|\rho_z(\mathbf{s}, x, y)|dxdy, \tag{5.3}$$

where $\varepsilon(x, y)$ is a positive function that weights geometric consistency against brightness constancy; $\rho_I(\mathbf{s}, x, y) = I_0(x, y) - I_1(x + \mu, y + \upsilon)$ and $\rho_z(\mathbf{s}, x, y) = \omega - Z_1(x + \mu, y + \upsilon) + Z_0(x, y)$ with I_0, I_1 being the intensity images while Z_0, Z_1 the depth images taken at instants t and $t + 1$.

The regularization term $E_R(\mathbf{s})$ is based on the total variation and takes into consideration the geometry of the scene which is formulated as:

$$\begin{aligned} E_R(\mathbf{s}) = \lambda_I \int_\Omega |(r_x\frac{\partial\mu}{\partial x}, r_y\frac{\partial\mu}{\partial y})| + |(r_x\frac{\partial\upsilon}{\partial x}, r_y\frac{\partial\upsilon}{\partial y})|dxdy \\ +\lambda_D \int_\Omega |(r_x\frac{\partial\omega}{\partial x}, r_y\frac{\partial\omega}{\partial y})|dxdy, \end{aligned} \tag{5.4}$$

where λ_I, λ_D are constant weights and $r_x = \frac{1}{\sqrt{\frac{\partial X^2}{\partial x} + \frac{\partial Z^2}{\partial x}}}$, $r_y = \frac{1}{\sqrt{\frac{\partial Y^2}{\partial y} + \frac{\partial Z^2}{\partial y}}}$.

As the energy function (Eq. (5.2)) is based on a linearisation of the data term (Eq. (5.3)) and convex TV regularizer (Eq. (5.4)), the energy function can be solved using convex solver. An iterative solver can be obtained by deriving the energy function (Eq. (5.2)) as its primal-dual formulation and implemented in parallel on GPUs. For more implementation details, the keen reader is recommended to read [JSGJC15].

5.1.2.2 Self-Calibration

Scene flow computation requires that the RGB and depth data be spatially aligned and temporally synchronized. The data considered in this chapter were captured by Kinect sensors and are temporally synchronized. However, the RGB and depth channels may not be spatially registered if calibration was not performed properly before recording the data. For the RGB-D datasets with spatial misalignment, we propose an effective self-calibration method to perform spatial alignment without knowledge of the cameras parameters. The alignment is based on a pinhole model through which depth maps are transformed into the same view of the RGB video. Let p_i be a point in an RGB frame and p'_i be the corresponding point in the depth map. The 2D homography mapping H satisfying $p_i = Hp'_i$ is a 3×3 projective transformation for the alignment. Following the method in [HZ03], we chose a set of matching points in an RGB frame and its corresponding depth map. Using four pairs of corresponding points, H is obtained through direct linear transformation. Let $p'_i = (x'_i, y'_i, 1)^T$, h_j^T be the j_{th} row of H and $\mathbf{0} = [0, 0, 0]^T$. The vector cross product equation $p_i \times Hp'_i = \mathbf{0}$ is written as [HZ03]:

$$\begin{bmatrix} \mathbf{0}^T & -p_i^T & y'_i p_i^T \\ p_i^T & \mathbf{0}^T & -x'_i p_i^T \end{bmatrix} \begin{pmatrix} h_1 \\ h_2 \\ h_3 \end{pmatrix} = \mathbf{0}, \tag{5.5}$$

where the up-to-scale equation is omitted. A better estimation of H is achieved by minimising (for example, using Levenberg-Marquardt algorithm [KYF05]) the following objective function with more matching points:

$$\begin{aligned} \arg\min_{\hat{H}, \hat{p}_i, \hat{p}_i{}'} & \sum_i [d(p_i, \hat{p}_i)^2 + d(p'_i, \hat{p}_i{}')] \\ \text{s.t. } & \hat{p}_i = \hat{H}\hat{p}_i{}' \text{ for } \forall i \end{aligned} \tag{5.6}$$

In Eq. (5.6), $d(\cdot)$ is the distance function and $\hat{H}$ is the optimal estimation of the homography mapping while $\hat{p}_i$ and $\hat{p}_i{}'$ are estimated matching points from $\{p_i, p'_i\}$. Because the process of selecting matching points may not be reliable, the random sample consensus (RANSAC) algorithm is applied to exclude outliers. By transforming the depth map using the 2D projective transformation H, the RGB video and its corresponding depth video are spatially aligned.

5.1.2.3 Construction of Hand-crafted SFAM

SFAM encodes a video sample into a single dynamic image to take advantage of the available pre-trained models for standard ConvNets architecture without training millions of parameters afresh. There are several ways to encode the video sequences

into dynamic images [BD01b, MB06, YZT12, WLG+15, WLG+16, BFG+16], but how to encode the scene flow vectors into one dynamic image still needs to be explored. As described in Section 5.1.2.1, one scene flow vector $\mathbf{s} = (\mu, \upsilon, \omega)^T$ is obtained by matching/tracking one point in the current frame to another in the reference frame; this is one simple Lagrangian motion. In order to avoid error in tracking Lagrangian motion over long term, we construct SFAM using the Eulerian motion approach and thus, the SFAM inherits the merits of both the Eulerian and Lagrangian motion. As we argued earlier, the three entries (μ, υ, ω) in the scene flow vector $\mathbf{s}$ for each point can be considered as three channels. Hence a scene flow between two pairs of RGB-D images (I_0, Z_0 and I_1, Z_1) can be reorganized as one three-channel SFM (X_μ, X_υ, X_ω), and the RGB-D video sequences can be represented as SFM sequences. Based on the SFM sequences, there are several ways to construct the SFAM.

SFAM-D

Inspired by the construction of Depth Motion Maps (DMM) [YZT12], we accumulate the absolute differences between consecutive SFMs and denote it as SFAM-D. It is written as:

$$\text{SFAM-D}_i = \sum_{t=1}^{T-1} |X_i^{t+1} - X_i^t| \quad i \in (\mu, \upsilon, \omega), \tag{5.7}$$

where t denotes the map number and T represents the total number of maps (the same for the following sections). This representation characterizes the distribution of the accumulated motion difference energy.

SFAM-S

Similarly to SFAM-D, we construct the SFAM-S (S here denotes the sum) by accumulating the sum between consecutive SFMs. This can be written as:

$$\text{SFAM-S}_i = \sum_{t=1}^{T-1} (X_i^{t+1} + X_i^t) \quad i \in (\mu, \upsilon, \omega). \tag{5.8}$$

This representation mainly captures the large motion of an action after normalization.

SFAM-RP

Inspired by the work reported in [BFG+16], we adopt the rank pooling method to encode SFM sequence into one action image. Let $X_1, ..., X_T$ denote the SFM sequence where each X_t contains three channels (X_μ, X_υ, X_ω), and $\varphi(X_t) \in \mathbb{R}^d$ be a representation or feature vector extracted from each individual map, X_t. Herein,

we directly apply rank pooling to the X, thus, $\varphi(\cdot)$ equals to identity matrix. Let $V_t = \frac{1}{t}\sum_{\tau=1}^{t}\varphi(X_\tau)$ be time average of these features up to time t. The ranking function associates with each time t a score $S(t|\mathbf{d}) =< \mathbf{d}, V_t >$, where $\mathbf{d} \in \mathbb{R}^d$ is a vector of parameters. The function parameters $\mathbf{d}$ are learned so that the scores reflect the order of the maps in the video. In general, more recent frames are associated with larger scores, i.e. $q > t \Rightarrow S(q|\mathbf{d}) > S(t|\mathbf{d})$. Learning $\mathbf{d}$ is formulated as a convex optimization problem using RankSVM [SS04]:

$$\begin{aligned} \mathbf{d}^* &= \rho(X_1, ..., X_T; \varphi) = \arg\min_{\mathbf{d}} E(\mathbf{d}), \\ E(\mathbf{d}) &= \frac{\lambda}{2} \parallel \mathbf{d} \parallel^2 + \\ &\quad \frac{2}{T(T-1)} \times \sum_{q>t} \max\{0, 1 - S(q|\mathbf{d}) + S(t|\mathbf{d})\}. \end{aligned} \tag{5.9}$$

The first term in this objective function is the usual quadratic regular term used in SVMs. The second term is a hinge-loss soft-counting how many pairs $q > t$ are incorrectly ranked by the scoring function. Note in particular that a pair is considered correctly ranked only if scores are separated by at least a unit margin, *i.e.* $S(q|\mathbf{d}) > S(t|\mathbf{d}) + 1$.

Optimizing the above equation defines a function $\rho(X_1, ..., X_T; \varphi)$ that maps a sequence of T SFMs to a single vector $\mathbf{d}^*$. Since this vector contains enough information to rank all the frames in the SFM sequence, it aggregates information from all of them and can be used as a sequence descriptor. In our work, the rank pooling is applied in a bidirectional manner to convert each SFM sequence into two action maps, SFAM-RPf (forward) and SFAM-RPb (backward). This representation captures the different types of importance associated with frames in one action and assigns more weight to recent frames.

SFAM-AMRP

In previous sections, all the three channels are considered as separate channels in constructing SFAM. However, the specific relationship (independent or otherwise) between them is yet to be ascertained. To study this relationship, we adopt a simple method *viz.*, using amplitude of the scene flow vector $\mathbf{s}$ to represent the relations between the three components. Thus, for each triple (X_μ, X_v, X_ω) we obtain a new amplitude map, X_{am}. Based on the $X_{am} = \sqrt{X_\mu^2 + X_v^2 + X_\omega^2}$, the rank pooling method is applied to encode the scene flow maps into two action maps, SFAM-AMRPf and SFAM-AMRPb. This representation exploits the weights of frames based on the motion magnitude.

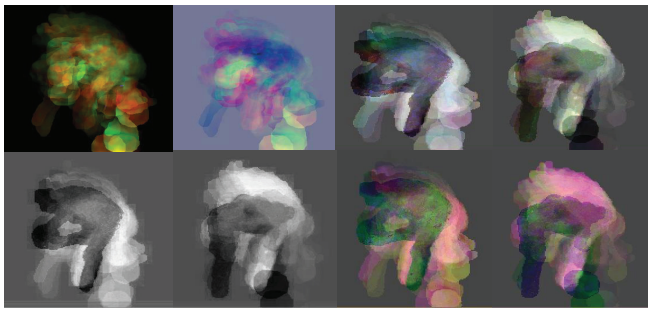

Figure 5.1: Samples of variants of SFAM for action "Bounce Basketball" from M^{2}I Dataset [LXN$^+$16]. For top-left to bottom-right, the images correspond to SFAM-D, SFAM-S, SFAM-RPf, SFAM-RPb, SFAM-AMRPf, SFAM-AMRPb, SFAM-LABRPf, SFAM-LABRPb.

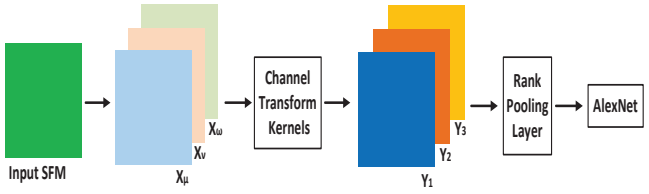

Figure 5.2: The framework for constructing SFAM with Channel Transform Kernels using ConvNets.

SFAM-LABRP

To further investigate the relationship amongst the triple (X_μ, X_v, X_ω), they are transformed nonlinearly into another space, similarly to the manner of transforming RGB color space to *Lab* space. The *Lab* space is designed to approximate the human visual system. Based on these transformed maps, the rank pooling method is applied to encode the sequence into two action maps, SFAM-LABRPf and SFAM-LABRPb.

A few examples of the SFAM variants are shown in Figure 5.1 for action "Bounce Basketball" from M^{2}I Dataset [LXN$^+$16]. It can be seen that different variants of SFAM capture and encode SFM sequence into action maps with large visual differences.

5.1.2.4 Constructing SFAM with Channel Transform Kernels (SFAM-CTKRP)

In previous sections, we have presented the concept of SFAM and its several variants. However, it has been empirically observed that none of them can achieve the best results for all the datasets or scenarios. One reason adduced for this is that during the construction of the SFAM, the relationship amongst the triple (X_μ, X_v, X_ω) are hand-crafted. To learn the relationship amongst the elements of the triple (X_μ, X_v, X_ω) from data with ConvNets, we propose a Channel Transform Kernels as follows.

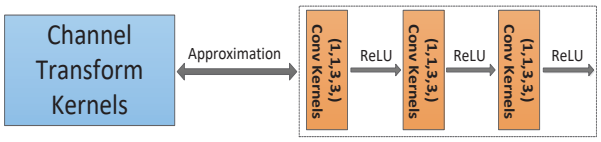

Figure 5.3: Illustration of approximate computation for Channel Transform Kernels using convolution kernels followed by nonlinear transforms.

Let Y_1, Y_2, Y_3 be the new learned maps from the original triple (X_μ, X_v, X_ω), the relationship between them can be formulated as:

$$\begin{aligned} Y_1 &= \varphi_1(\omega_1 X_\mu + \omega_2 X_v + \omega_3 X_\omega) \\ Y_2 &= \varphi_2(\omega_4 X_\mu + \omega_5 X_v + \omega_6 X_\omega) \\ Y_3 &= \varphi_3(\omega_7 X_\mu + \omega_8 X_v + \omega_9 X_\omega) \end{aligned} \tag{5.10}$$

where Y has the same size with X, ω are scalar values and φ denotes the transforms that need to be learned. The learning framework is illustrated in Figure 5.2. There are different ways to learn these Channel Transform Kernels. For sake of simplicity, in this work we achieved the non-linear channel transformations by three successive convolution layers, where each layer comprises nine convolutional kernels with size 1×1 and followed by ReLU nonlinear transform, as illustrated in Figure 5.3. Based on RankPool layer [BFG+16] for temporal encoding, we can construct the SFAM with the proposed Channel Transform Kernels using ConvNets.

5.1.2.5 Product Score Fusion for Classification

After construction of the several variants of SFAM, we propose to adopt one effective late score fusion method, namely, product score fusion method, to improve the final recognition accuracy. Take SFAM-RP for example, as illustrated in Figure 5.4, two SFAM-RP, one SFAM-RPf and one SFAM-RPb, are generated for one pair of RGB-D videos and they are fed into two different trained ConvNets channels. The score vectors output by the two ConvNets are multiplied element-wisely and the max score in the resultant vector is assigned as the probability of the test sequence. The index of this max score corresponds to the recognized class label. This process can be easily extended into multiple channels.

5.1.3 Experimental Results

According to the survey of RGB-D datasets [ZLO+16a], we chose two public benchmark datasets, which contain both RGB+depth modalities and have relatively large

training samples to evaluate the proposed method. Specifically we chose ChaLearn LAP IsoGD Dataset [WLZ+16] and M^{2}I Dataset [LXN+16]. In the following, we proceed by briefly describing the implementation details and then present the experiments and results.

5.1.3.1 Implementation Details

For scene flow computation, we adopted the public codes provided by [JSGJC15]. For rank pooling, we followed the work reported in [BFG+16] where each channel was generated into one channel dynamic map and then merged the three channels into one three-channel map. Differently from [BFG+16], we used bidirectional rank pooling. For ChaLearn LAP IsoGD Dataset, in order to minimize the interference of the background, it is assumed that the background in the histogram of depth maps occupies the last peak representing far distances. Specifically, pixels whose depth values are greater than a threshold defined by the last peak of the depth histogram minus a fixed tolerance (0.1 was set in our experiments) are considered as background and removed from the calculation of scene flow by setting their depth values to zero. Through this simple process, most of the background can be removed and has much contribution to the SFAM.

The AlexNet [KSH12] was adopted in this chapter. The training procedure of the hand-crafted SFAMs was similar to that described in [KSH12]. The network weights were learned using the mini-batch stochastic gradient descent with the momentum set to 0.9 and weight decay set to 0.0005. All hidden weight layers used the rectification (RELU) activation function. At each iteration, a mini-batch of 256 samples was constructed by sampling 256 shuffled training samples. All the images were resized to 256×256. The learning rate was set to 10^{-3} for fine-tuning with pre-trained models on ILSVRC-2012, and then it was decreased according to a fixed schedule, which was kept the same for all training sets. Different datasets underwent different iterations according to their number of training samples. For all experiments, the dropout regularization ratio was set to 0.5 in order to reduce complex co-adaptations of neurons in the nets. The implementation was derived from the publicly available Caffe toolbox [JSD+14] based on one NVIDIA Tesla K40 GPU card. Unless otherwise specified, all the networks were initialized with the models trained over ImageNet [KSH12]. For SFAM-CTKRP, we revised the codes of paper [BFG+16] based on MatConvNet [VL15]. The product score fusion method is compared with the other two commonly used late score fusion methods, average and maximum score fusion on both datasets. This verifies that the SFAMs are likely to be statistically independent and provide complementary information.

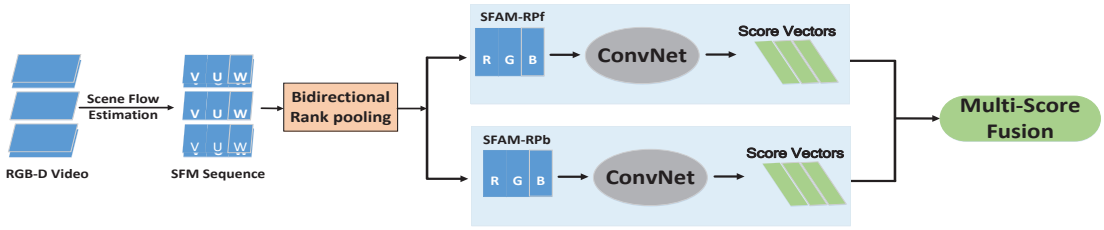

Figure 5.4: Illustration of Product Score Fusion for SFAM-RP.

5.1.3.2 ChaLearn LAP IsoGD Dataset

The ChaLearn LAP IsoGD Dataset [WLZ+16] was adopted to evaluate the proposed method. The dataset is divided into training, validation and test sets. As the test set is not available for public usage, we report the results on the validation set. For this dataset the training underwent 25K iterations and the learning rate decreased every 10K iterations.

Table 5.5 shows the results of six variants of SFAM, and compares them with methods in the literature [WGL16, WLZ+16, BFG+16, WLG+16]. Among these methods, MFSK combined 3D SMoSIFT [WRL+14] with (HOG, HOF and MBH) [WS13] descriptors. MFSK+DeepID further included Deep hidden IDentity (Deep ID) feature [SWT14]. Thus, these two methods utilized not only hand-crafted features but also deep learning features. Moreover, they extracted features from RGB and depth separately, concatenated them together, and adopted Bag-of-Words (BoW) model as the final video representation. The other methods, WHDMM+SDI [WLG+16, BFG+16], extracted features and conducted classification with ConvNets from depth and RGB individually and adopted product score fusion for final recognition.

Compared with these methods, the proposed SFAM outperformed all of them significantly. It is worth noting that all the depth values used in the proposed SFAM were estimated rather than the exact real depth values. Despite the possible estimation errors, our method still achieved promising results. Interestingly, the proposed variants of SFAM are complementary to each other and can improve each other largely by using product score fusion. Even though this dataset is large, on average 144 video clips per class, it is still much smaller compared with 1200 images per class in ImageNet. Thus, directly training from scratch cannot compete with fine-tuning the trained models over ImageNet and this is evident in the results reported in Table 5.5. By comparing different types of SFAM, we can see that the simple SFAM-S method achieved the best results among all types of hand-designed SFAMs. Due to the relatively large training data, SFAM-CTKRP achieved the best result among all the variants, even though the approximate rank pooling in the

work reported in [BFG+16] was shown to be worse than rank pooling solved by RankSVM [SS04]. The reasons for these two phenomenona probably are as follows: under the inaccurate estimation of the depth values, the scene flow computation will be affected and based on this inaccurate scene flow vectors, rank pooling can not achieve its full efficacy. In other words, the rank pooling method is sensitive to noise. Instead, the proposed Channel Transform Kernels cannot only exploit the relations amongst the channels but also decrease the effects of noise after channel transforms.

Table 5.1: Results and comparison on the ChaLearn LAP IsoGD dataset.

Method	Accuracy
MFSK [WGL16, WLZ+16]	18.65%
MFSK+DeepID [WGL16, WLZ+16]	18.23%
SDI [BFG+16]	20.83%
WHDMM [WLG+16]	25.10%
WHDMM+SDI [WLG+16, BFG+16]	25.52%
SFAM-D (training from scratch)	9.23%
SFAM-D	18.86%
SFAM-S (training from scratch)	18.10%
SFAM-S	25.83%
SFAM-RP	23.62%
SFAM-AMRP	18.21%
SFAM-LABRP	23.35%
SFAM-CTKRP	27.48%
Max Score Fusion All	33.24%
Average Score Fusion All	34.86%
Product Score Fusion All	**36.27%**

5.1.3.3 M²I Dataset

Multi-modal & Multi-view & Interactive (M**2**I) Dataset [LXN+16] was adopted to evaluate the proposed method. For evaluation, all samples were divided with respect to the groups into a training set (8 groups), a validation set (6 groups) and a test set (6 groups). The final action recognition results are obtained with the test set. For this dataset the training underwent 6K iterations and the learning rate decreased every 3K iterations.

We followed the experimental settings as in [LXN+16] and compared the results on two scenarios: single task scenario and cross-view scenario. The baseline methods were based on iDT features [WS13] generated from optical flow and has been shown to be very effective in 2D action recognition. Specifically, for the BoW framework, a set of local spatio-temporal features were extracted, including iDT-Tra, iDT-HOG, iDT-HOF, iDT-MBH, iDT-HOG+HOF, iDT-HOF+MBH and iDT-COM (concate-

Table 5.2: Comparison on the M**2**I Dataset for single task scenario (learning and testing in the same view).

Method	Accuracy	
	SV	FV
iDT-Tra (BoW) [LXN+16]	69.8%	65.8%
iDT-COM (BoW) [LXN+16]	76.9%	75.3%
iDT-COM (FV) [LXN+16]	80.7%	79.5%
iDT-MBH (BoW) [LXN+16]	77.2%	79.6%
SFAM-D	71.2%	83.0%
SFAM-S	70.1%	75.0%
SFAM-RP	79.9%	81.8%
SFAM-AMRP	82.2%	78.0%
SFAM-LABRP	72.0%	83.7%
Max Score Fusion All	87.6%	88.8%
Average Score Fusion All	88.2%	89.1%
Product Score Fusion All	**89.4%**	**91.2%**

Table 5.3: Comparison on the M**2**I Dataset for cross-view scenario.(SV → FV: learning in the side view and test in the front view; FV → SV: learning in the front view and testing in the side view.)

Method	Accuracy	
	SV → FV	FV → SV
iDT-Tra [LXN+16]	43.3%	39.2%
iDT-COM [LXN+16]	70.2%	67.7%
iDT-HOG+MBH [LXN+16]	75.8%	71.8%
iDT-HOG+HOF [LXN+16]	78.2%	72.1%
SFAM-D	66.7%	65.2%
SFAM-S	68.2%	60.2%
SFAM-RP	71.6%	65.2%
SFAM-AMRP	77.7%	66.7%
SFAM-LABRP	76.9%	65.9%
Max Score Fusion All	84.7%	73.8%
Average Score Fusion All	85.3%	75.3%
Product Score Fusion All	**87.6%**	**76.5%**

nation of all descriptors); for fisher vector framework, they only used the iDT-COM feature for evaluation. For comparisons, we only show several best results achieved by baseline methods for each scenario. Table 5.2 shows the comparisons on the M**2**I Dataset for single task scenario, that is, learning and testing in the same view while Table 5.3 presents the comparisons for cross-view scenario. Due to the lack of training data, SFAM-CTKRP could not converge steadily and the results varied largely, thus, we did not show its results. For this dataset, SFAM-AMRP achieved the best result for side view while SFAM-LABRP achieved the best result for front view. From Table 5.2 we can see that for scene flow estimation based on real true

depth values, the rank pooling-based method achieved better results than SFAM-D and SFAM-S, which are consistent with the conclusion in [LXN+16]. SFAM-AMRP achieved the best results for two cross-view scenarios which can be seen from Table 5.3. Interestingly, even though our proposed SFAM did not solve any transfer learning problem as in [LXN+16] but directly training with the side/front view and testing in the front/side view, it still outperformed the best baseline method significantly, especially in the SV → FV setting. This bonus advantage reflects the effectiveness of proposed method.

5.2 Cooperative Training of ConvNets for RGB and Depth Modalities

5.2.1 Prior Works and Our Contributions

RGB-D based action recognition has attracted much attention in recent years due to the advantages that depth information brings to the combined data modality. For example, depth is insensitive to illumination changes and includes rich 3D structural information of the scene. However, depth alone is insufficient for recognizing some actions. In the task of recognizing human-object interactions where texture is vital for successful recognition, depth does not capture the necessary texture context. To exploit the complementary nature of the two modalities, several works [JKDF14, NWJ15, KF15, HZLZ15, WZSS15, KF17] have combined the two modalities for RGB-D action recognition and demonstrated the effectiveness of modality fusion. Ni et al. [NWM11] constructed one color-depth video dataset and developed two color-depth fusion techniques based on hand-designed features for human action recognition. Liu and Shao [LS13b] proposed to adopt genetic programming method to simultaneously extract and fuse the color and depth information into one feature representation. Jia et al. [JKDF14] proposed one transfer learning method that transfered the knowledge from depth information to the RGB dataset for effective RGB-based action recognition. Hu et al. [HZLZ15] proposed a multi-task learning method to simultaneously explore the shared and feature-specific components for heterogeneous features fusion. Sharing similar ideas, Kong and Fu [KF15] compressed and projected the heterogeneous features to a shared space while Kong and Fu [KF17] learned both the shared space and independent private spaces to capture the useful information for action recognition. However, all these efforts are based on hand-crafted features and tend to be dataset-dependent.

The advent of deep learning has brought about the development of methods [JXYY13, TBF+15, SZ14a, WLG+15, WLG+16, JG16, DAHG+15] based on

ConvNet or RNN. These methods take as input either RGB or depth or both of them as independent channels with late fusion. It is noteworthy that none of these methods address the problem of using heterogeneous inputs (such as RGB and depth) in a cooperative manner to train a single network for action recognition. This cooperative training paradigm allows the powerful representation capability of deep neural network to be fully leveraged to explore the complementary information contained in the two modalities in one single network architecture. The need for independent processing channels is thus obviated. Motivated by this observation, we propose to adopt deep cooperative neural networks for RGB-D action recognition based on these two modalities.

However, it remains unclear as to how a RGB-D sequence could be effectively represented and fed to deep neural networks for recognition. For example, one can conventionally consider it as a sequence of still images (RGB and depth) with some form of temporal smoothness, or as a subspace of images or image features, or as the output of a neural network encoder. Which one among these and other possibilities would result in the best representation in the context of action recognition is not well understood. In addition, it is not clear either how the two heterogeneous RGB and depth channels can be represented and fed into a single deep neural network for the cooperative training. Inspired by the promising performance of the recently introduced rank pooling machine [FGO$^+$15, BFG$^+$16] on RGB videos, the rank pooling method is adopted to encode both RGB and depth sequences into compatible dynamic images. A dynamic image contains the temporal evolution information of a video sequence and keeps the spatio-temporal structured relationships of the video; this has been demonstrated to be an effective video descriptor [BFG$^+$16]. Based on this pair of dynamic images, namely, RGB visual dynamic images (VDIs) and depth dynamic images (DDIs), a cooperatively trained convolutional neural networks (c-ConvNet) is proposed to exploit the two modality features and enhance the capability of ConvNets for cases in which the features arise either from heterogeneous or homogeneous sources.

There are two issues in using a single c-ConvNet for either homogeneous or heterogeneous modality action recognition. First, how to enhance the discriminative power of ConvNets and second, how to reduce the modality discrepancy. Specifically, in most classification cases, the conventional ConvNets can learn separable features but they are often not compact enough to be discriminative [WZLQ16]. Modality discrepancy arises because the feature variations in different modalities pose a challenge for a single network to learn modality-independent features for classification. To handle these two issues, we propose to jointly train a ranking loss and a softmax loss for action recognition. The ranking loss consists of two intra-modality and cross-modality triplet losses, which reduces variations in both

intra-modality and cross-modality. Together with the softmax loss, the supervision signal intra-modality triplet loss enables the c-ConvNet to learn more discriminative features, while the inter-modality triplet loss weakens or eliminates the modalities distribution variations and only focuses on action distinctions. Moreover, in this way, knowledge about the correlations between RGB and depth data are incorporated in the c-ConvNet, and enables the use of additional depth information for the case where only RGB information is available. Furthermore, due to the image structure of dynamic images, the proposed c-ConvNet can be fine-tuned on the pre-trained networks on ImageNet, thus making it possible to work on small datasets. The c-ConvNet was evaluated extensively on three datasets: two large datasets, ChaLearn LAP IsoGD [WLZ+16] and NTU RGB+D [SLNW16] datasets, and one small dataset, SYSU 3D HOI [HZLZ15] dataset. Experimental results achieved are state-of-the-art. The c-ConvNet showed promising results compared with conventional ConvNet, and it is suitable for use with single or both modalities for action recognition.

The contributions of this chapter are summarized as follows: 1) to our best knowledge, this is the first attempt to adopt ConvNet for a cooperatively trained network taking heterogeneous input (RGB-D) for action recognition. Thus the correlation between RGB and depth modalities are efficiently exploited; 2) a c-ConvNet is proposed by jointly training both ranking and classification loss functions, and the extra ranking loss function makes the ConvNets more discriminative and modality independent; 3) State-of-the-art results are achieved on three datasets.

5.2.2 The Proposed Methods

The proposed method consists of three phases, as illustrated in Figure 5.5, viz., the constructions of RGB visual dynamic images (VDIs) and depth dynamic images (DDIs), c-ConvNets and product-score fusion for final heterogeneous-feature-based action recognition. The first phase is an unsupervised learning process. It applies bidirectional rank pooling method to generate the VDIs and DDIs and represented by two dynamic images (forward (DDIf) and backward (DDIb)). In the following sections, we describe the three phases in detail. The rank pooling method [BFG+16], that aggregates spatio-temporal-structural information from one video sequence into one dynamic image, is also briefly summarized.

5.2.2.1 Construction of VDIs & DDIs

Rank pooling defines a rank function that encodes the video into one feature vector. Let the RGB/depth video sequence with k frames be represented as $< d_1, d_2, ..., d_t, ..., d_k >$, where d_t is the average of RGB/depth features over time

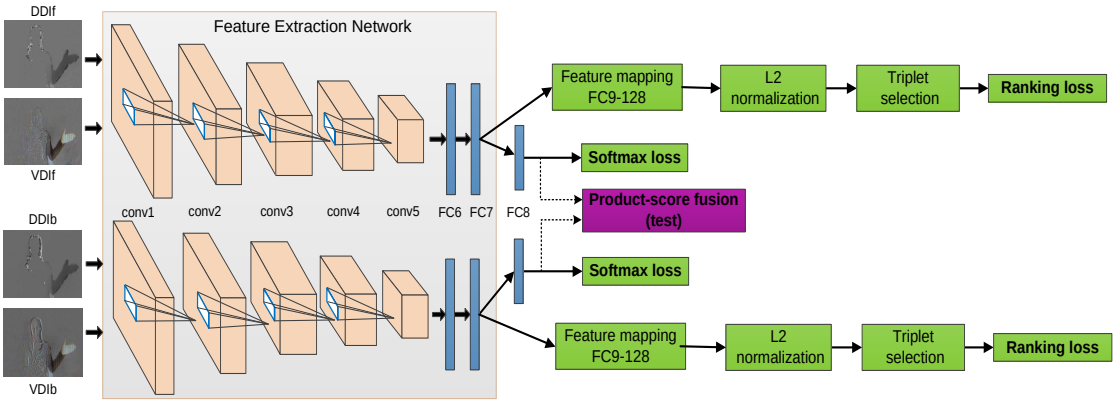

Figure 5.5: The framework of proposed method. A c-ConvNet consists of one feature extraction network shared by the ranking loss and softmax loss, and two separate branches for the two losses. Two distinct c-ConvNets are adopted to exploit bidirectional information in videos. The inputs of the two c-ConvNets are two paired DDIs and VDIs, namely, DDIf & VDIf, and DDIb & VDIb. During training process, the ranking loss and softmax loss are jointly optimized; during testing process, an effective product-score fusion method is adopted for action recognition. The softmax loss serves to learn separable features for action recognition while the ranking loss encourages the c-ConvNet to learn discriminative and modality-independent features.

up to t-frame or t-timestamp. At each time t, a score $r_t = \omega^T \cdot d_t$ is assigned. The score satisfies $r_i > r_j \Longleftrightarrow i > j$. In general, more recent frames are associated with larger scores. This process can be formulated as:

$$\begin{aligned} &\arg\min_{\omega} \frac{1}{2} \parallel \omega \parallel^2 + \delta \sum_{i>j} \xi_{ij} \\ s.t. \;\; &\omega^T \cdot (d_i - d_j) \geq 1 - \xi_{ij}, \xi_{ij} \geq 0 \end{aligned}, \tag{5.11}$$

where ξ_{ij} is the slack variable. Optimizing the above equation defines the rank function that maps a sequence of k RGB/depth video frames to a single vector ω^*. Since this vector aggregates information from all the frames in the sequence, it can be used as a video descriptor. The process of obtaining ω^* is called rank pooling. In this chapter, rank pooling is directly applied on the pixels of RGB/depth frames and the ω^* is of the same size as RGB/depth frames and forms a dynamic image. Since in rank pooling the averaged feature up to time t is used to classify frame t, the pooled feature is biased towards beginning frames of the depth sequence, hence, frames at the beginning has more influence to ω^*. This is not justifiable in action recognition as there is no prior knowledge on which frames are more important than other frames. Therefore, unlike the work of Bilen et al. [BFG+16], the rank pooling is applied bidirectionally RGB/Depth sequences to reduce such bias.

Visual comparisons of DDIf (forward), DDIb (backward), VDIf (forward) and

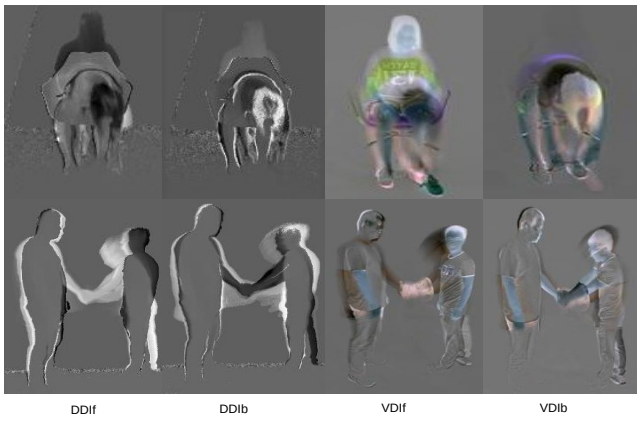

Figure 5.6: Visual comparisons of DDIf, DDIb, VDIf and VDIb. The left two columns are the "wear a shoe" action and the right two columns are the action "handshaking" from NTU RGB+D Dataset [SLNW16].

VDIb (backward) are illustrated in Figure 5.6. From this figure, it can be seen that compared with VDIs, DDIs lose the texture information of the object (shoes) and human, which is beneficial for simple action recognition without human-object interactions but bad for interactions. The two directional DDIs and VDIs also capture different order of information for actions which are complementary to each other. Besides, the dynamic images also capture the structured information of an action, that illustrates the coordination and synchronization of body parts over the period of the action, and describe the relations of spatial configurations of human body across different time slots.

5.2.2.2 c-ConvNet

Joint Ranking and Classification The softmax loss adopted in the ConvNet can only learn separable features for homogeneous modalities, and is not guaranteed to be discriminative [WZLQ16]. In order to make the ConvNet more discriminative for both RGB and depth modalities, the softmax and ranking losses are proposed to be jointly optimized as in Figure 5.5. Triplet loss is a type of ranking loss, and has proven effective in several applications, such as face recognition [SKP15, LSWT16], pose estimation [KCL16] and image retrieval [JWF16]. In this chapter, the triplet loss is adopted as the ranking loss. In common usage, the triplet loss works on the homogeneous triplet data, namely, anchor, positive and negative samples, (x_a^i, x_p^i, x_n^i), where (x_a^i, x_p^i) have the same class label and (x_a^i, x_n^i) have different class labels. The training encourages the network to find an embedding $f(x)$ such that the distance between the positive sample and the anchor sample $d^i_{<a,p>} = ||f(x_a^i) - f(x_p^i)||_2^2$ is smaller than the distance $d^i_{<a,n>} = ||f(x_a^i) - f(x_n^i)||_2^2$ between the negative sample and the anchor sample by a margin, α. Thus the triplet loss l

can be formulated as:

$$l = \sum_{i}^{N} [||f(x_a^i) - f(x_p^i)||_2^2 - ||f(x_a^i) - f(x_n^i)||_2^2 + \alpha]_+, \tag{5.12}$$

where N is the number of possible triplets.

In order to make the triplet loss suitable for both homogeneous and heterogeneous modality-based recognition, a new triplet loss made up of both intra-modality and inter-modality triplet losses is designed (see Figure 5.7). For the sake of computational efficiency and consideration of both intra and inter modalities variations, four types of triplets are defined in this chapter. If the anchor is one depth sample, then two positive and negative depth samples are assigned to intra-modality triplet while two RGB samples are assigned to cross-modality triplet; if the anchor is one RGB sample, then two positive and negative RGB samples are assigned to intra-modality triplet while two depth samples are assigned to cross-modality triplet. Thus, the new ranking loss can be defined as:

$$L_r = (l^{Dep,Dep} + l^{RGB,RGB}) + \lambda(l^{Dep,RGB} + l^{RGB,Dep}), \tag{5.13}$$

where $l^{Dep,Dep}$ denotes the intra-modality loss function of triplet $(x^i_{a_{depth}}, x^i_{p_{depth}}, x^i_{n_{depth}})$; $l^{Dep,RGB}$ represents inter-modality loss function of triplet $(x^i_{a_{depth}}, x^i_{p_{RGB}}, x^i_{n_{RGB}})$; and it is analogous to $l^{RGB,RGB}$ and $l^{RGB,Dep}$; λ trades off between the two kinds of losses. With the constraint of these four triplet losses, the network is forced more towards action distinction so that the cross-modality variance is weakened or even eliminated. In this way, the knowledge about the correlations between RGB and depth data are also incorporated in the c-ConvNet, and enables the use of additional depth information for the case where only RGB information is available.

Together with the softmax loss, the final loss function to be optimized in this chapter is formulated as:

$$L = L_s + \gamma L_r, \tag{5.14}$$

where L_s denotes the softmax loss and γ is a weight to balance the different loss functions.

Network Structure The c-ConvNet consists of one feature extraction network, a branch each for ranking loss and softmax loss, as illustrated in Figure 5.5. The feature extraction network is shared by the two losses and it can be any available pre-trained network over ImageNet. In this chapter, VGG-16 [SZ14b] network is adopted due to its promising results in various vision tasks. The softmax loss branch is built on the FC8 layer which is same as VGG-16. The ranking loss branch consists of one feature mapping layer (FC9-128), one L2 normalization layer, one triplet selection

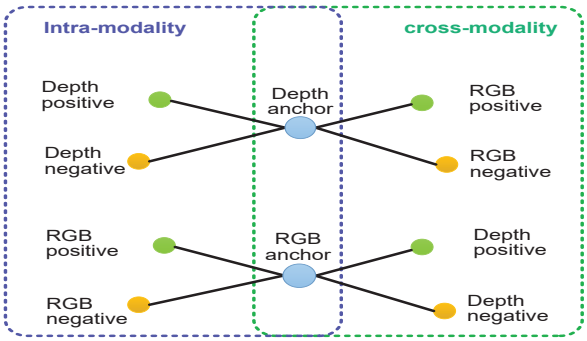

Figure 5.7: Illustration of the intra-modality and inter-modality triplets.

layer and one ranking loss layer. The feature mapping layer built on the FC7 layer of VGG-16, aims to learn a compact representation for the triplet embedding. Inspired by [SKP15], L2 normalization layer is followed to constrain the embedding to live on the hypersphere space. Triplet is selected online using one triplet selection layer to generate the four kinds of triplets. In this layer, every training sample will be selected as the anchor sample, and its corresponding positive and negative samples randomly selected according to Figure 5.7. The ranking loss is built on the triplet selection layer to minimize the loss according to Equation 5.13. In order to leverage the bidirectional information of videos, two c-ConvNets are trained separately based on forward and backward dynamic images. An effective product-score fusion method is adopted for final action recognition based on FC8 layer.

5.2.2.3 Product Score Fusion

Given a test RGB and depth video sequences, two pairs of dynamic images, VDIf & DDIf, and VDIb & VDIb are constructed and fed into two different trained c-ConvNets. For each image pair, product score fusion is used. The score vectors output of the weight sharing c-ConvNets are multiplied in an element-wise manner, and then the resultant score vectors (product-score) are normalized using L_1 norm. The two normalized score vectors are multiplied, element-wise, and the max score in the resultant vector is assigned as the probability of the test sequences. The index of this max score corresponds to the recognized class label.

5.2.3 Experimental Results

The proposed method was evaluated on three benchmark RGB-D datasets, namely, two large ones, ChaLearn LAP IsoGD [WLZ+16] and NTU RGB+D [SLNW16] datasets, and a small one, SYSU 3D HOI [HZLZ15] dataset. These three datasets cover a wide range of different types of actions including gestures, simple actions,

daily activities, human-object interactions and human-human interactions. In the following, we proceed by briefly describing the implementation details and then present the experiments and results.

5.2.3.1 Implementation Details

The proposed method was implemented using the Caffe framework [JSD+14] based on one NVIDIA Tesla K40, one TITAN X and two TITAN X Pascal GPU cards. First, the feature extraction network was fine-tuned on both depth and RGB modalities. Then, the c-ConvNet was trained 30 epochs. The initial learning rate was set to 0.001 and decreased by a factor of 10 every 12 epochs. The batch size was set as 50 images, with 5 actions in each batch. The network weights are learned using the mini-batch stochastic gradient descent with the momentum set to the value 0.9 and weight decay set to the value 0.0005. The parameter γ was assigned the value 10 in order to ensure that the two losses are of comparable magnitude. Parameters α and λ were assigned values that depend on the level of difficulty of the datasets.

Table 5.4: Results and comparison on the ChaLearn LAP IsoGD Dataset using ConvNet and c-ConvNet.

Method	Accuracy
DDIf (ConvNet)	36.13%
VDIf (ConvNet)	16.20%
DDIb (ConvNet)	30.45%
VDIb (ConvNet)	14.99%
DDIf + VDIf (ConvNet)	33.64%
DDIb + VDIb (ConvNet)	30.48%
DDIf + DDIb (ConvNet)	**37.52%**
VDIf + VDIb (ConvNet)	17.60%
DDIf + VDIf + DDIb + VDIb (ConvNet)	35.65%
DDIf (c-ConvNet)	36.36%
VDIf (c-ConvNet)	28.44%
DDIb (c-ConvNet)	36.55%
VDIb (c-ConvNet)	31.95%
DDIf + VDIf (c-ConvNet)	41.01%
DDIb + VDIb (c-ConvNet)	40.78%
DDIf + DDIb (c-ConvNet)	40.08%
VDIf + VDIb (c-ConvNet)	36.60%
DDIf + VDIf + DDIb + VDIb (c-ConvNet)	**44.80%**

5.2.3.2 ChaLearn LAP IsoGD Dataset

The ChaLearn LAP IsoGD Dataset [WLZ+16] was adopted to evaluate the proposed method. The dataset is divided into training, validation and test sets. All three sets

consist of samples of different subjects to ensure that the gestures of one subject in the validation and test sets will not appear in the training set. As the test set is not available for public usage, we report the results on the validation set. For this dataset, the margin α was set to 0.2. The parameter, λ, was set to a value of 5 to solve the more difficult task of learning large cross-modality discrepancy.

Table 5.5: Results and comparison on the ChaLearn LAP IsoGD Dataset with previous papers.

Method	Modality	Accuracy
MFSK [WGL16, WLZ+16]	RGB+depth	18.65%
MFSK+DeepID [WGL16, WLZ+16]	RGB+depth	18.23%
SDI [BFG+16]	RGB	20.83%
WHDMM [WLG+16]	Depth	25.10%
WHDMM+SDI [WLG+16, BFG+16]	RGB+depth	25.52%
SFAM [WLG+17]	RGB+Depth	36.27%
Proposed Method	RGB+Depth	**44.80%**

To compare the ConvNet with the c-ConvNet, four ConvNets (VGG-16) on DDIf, VDIf, DDIb and VDIb were trained separately for 40 epochs, initialized with the pre-trained models over ImageNet. The initial learning rate was set to 0.001 and decreased by a factor of 10 every 16 epochs. The momentum and weight decay parameters were set similarly as c-ConvNet. It is found that 40 epochs were enough to achieve good results; increasing the training epochs would not increase but even decreased the results. For c-ConvNet, two c-ConvNets are trained separately based on DDIf&VDIf, and DDIb&VDIb, as illustrated in Figure 5.5. The trained c-ConvNet can be used for single or both modalities testing. For both cases, the product-score fusion method was adopted to aggregate different channels. The comparisons of ConvNet and c-ConvNet are shown in Table 5.4. From this Table it can be seen that for depth channels, DDIf and DDIb, the c-ConvNet only increases the accuracy slightly, but for RGB channels, VDIf and VDIb, the improvements are over 10 percentage points. Interestingly, for ConvNet, due to the poor results of RGB features, the fusion of additional RGB channels decreased the final accuracy compared with those in which only depth was adopted. Meanwhile, the proposed c-ConvNet significantly improved the RGB channel, and the fusion of two modalities improved the final results. These results demonstrate that knowledge about the correlations between RGB and depth data are incorporated in the c-ConvNet, and enables the use of additional depth information for the case where only RGB information is available for testing. The fusion of both forward and backward dynamic images improved the final accuracy by around 5 percentage points. Thus justifying that bidirectional motion information are mutually beneficial and can improve action recognition. The results of c-ConvNet in the final fusion over the four channels

improved by nearly 10 percentage points; a strong demonstration of the effectiveness of the proposed method.

Table 5.5 shows the comparisons of proposed method with previous works. Previous methods include MFSK combined 3D SMoSIFT [WRL+14] with (HOG, HOF and MBH) [WS13] descriptors. MFSK+DeepID further included Deep hidden IDentity (Deep ID) feature [SWT14]. Thus, these two methods utilized not only hand-crafted features but also deep learning features. Moreover, they extracted features from RGB and depth separately, concatenated them together, and adopted Bag-of-Words (BoW) model as the final video representation. The other methods, WHDMM+SDI [WLG+16, BFG+16], extracted features and conducted classification with ConvNets from depth and RGB individually and adopted product-score fusion for final recognition. SFAM [WLG+17] adopted scene flow to extract features and encoded the flow vectors into action maps, which fused RGB and depth data from the onset of the process. From this table, we can see that the proposed method outperformed all of these recent works significantly, and illustrated its effectiveness.

5.2.3.3 NTU RGB+D Dataset

The largest NTU RGB+D Dataset was adopted to evaluate the proposed method. It consists of front view, two side views and left, right 45 degree views. This dataset is challenging due to large intra-class and viewpoint variations. For fair comparison and evaluation, the same protocol as that in [SLNW16] was used. It has both cross-subject and cross-view evaluation. In the cross-subject evaluation, samples of subjects 1, 2, 4, 5, 8, 9, 13, 14, 15, 16, 17, 18, 19, 25, 27, 28, 31, 34, 35 and 38 were used as training and samples of the remaining subjects were reserved for testing. In the cross-view evaluation, samples taken by cameras 2 and 3 were used as training, while the testing set includes samples from camera 1. For this dataset, the margin α was set to 0.1 while λ was set to 2.

Similarly to LAP IsoGD Dataset, we conducted several experiments to compare the conventional ConvNet and c-ConvNet, and the comparisons are shown in Table 5.6. From this table, we can see that the c-ConNet learned more discriminative features compared to conventional ConvNet. Analysis of this results and the comparative results on LAP IsoGD Dataset indicates that the improvements gained on NTU RGB+D Dataset are less than those of LAP IsoGD Dataset. This is probably due to the high accuracy already achieved on this dataset by ConvNet. From these two comparisons it may be conclude that c-ConvNet works better on the difficult datasets for recognition.

Table 5.7 lists the performance of the proposed method and those previous works. The proposed method was compared with some skeleton-based methods, depth-based methods and RGB+Depth based methods that are previously reported

Table 5.6: Results and comparison on the NTU RGB+D Dataset using ConvNet and c-ConvNet.

Method	Cross subject	Cross view
DDIf (ConvNet)	75.80%	76.50%
VDIf (ConvNet)	70.99%	75.45%
DDIb (ConvNet)	76.44%	75.62%
VDIb (ConvNet)	71.37%	76.57%
DDIf + VDIf (ConvNet)	80.77%	83.19%
DDIb + VDIb (ConvNet)	80.74%	83.04%
DDIf + DDIb (ConvNet)	81.66%	81.53%
VDIf + VDIb (ConvNet)	78.31%	83.58%
DDIf + VDIf + DDIb + VDIb (ConvNet)	**84.99%**	**87.51%**
DDIf (c-ConvNet)	76.58%	78.22%
VDIf (c-ConvNet)	71.35%	77.41%
DDIb (c-ConvNet)	77.69%	76.55%
VDIb (c-ConvNet)	73.24%	78.02%
DDIf + VDIf (c-ConvNet)	82.64%	85.21%
DDIb + VDIb (c-ConvNet)	82.81%	85.62%
DDIf + DDIb (c-ConvNet)	82.51%	83.26%
VDIf + VDIb (c-ConvNet)	78.59%	84.68%
DDIf + VDIf + DDIb + VDIb (c-ConvNet)	**86.42%**	**89.08%**

on this dataset. We can see that the proposed method outperformed all the previous works significantly. Curious observation of the results shown in Table 5.6 and Table 5.7 indicates that when only one channel of the dynamic images (e.g. DDIf or VDIf) is adopted, the proposed method still achieved the best results. This is a strong demonstration of the effectiveness of dynamic images using ConvNets.

5.2.3.4 SYSU 3D HOI Dataset

The SYSU 3D Human-Object Interaction Dataset (SYSU 3D HOI Dataset) [HZLZ15] was adopted to evaluate the proposed method. As this dataset is quite noisy, especially the depth data, and the subjects are relatively small in the scene, the ranking pooling has been affected and the constructed DDIs and VDIs become noisy as well. Only 69% recognition accuracy was achieved by using the noisy dynamic images. In order to reduce the noise impact, skeleton data were used to locate the joints of subjects, and around each joint (16 joints in total were selected for the body) one VDI or DDI was generated and the VDIs or DDIs of all 16 joints are stitched together into one VDI or DDI as input to the c-ConvNets. For this dataset, the margin α was set to 0 while λ was set to 1.

Similarly to the above two large datasets, we conducted the following experiments to compare the ConvNet and c-ConvNet as in Table 5.8. From this table, it

Table 5.7: Comparative accuracies of the proposed method and previous methods on NTU RGB+D Dataset.

Method	Modality	CS	CV
Lie Group [VAC14]	Skeleton	50.08%	52.76%
HBRNN [DWW15]	Skeleton	59.07%	63.97%
2 Layer RNN [SLNW16]	Skeleton	56.29%	64.09%
2 Layer LSTM [SLNW16]	Skeleton	60.69%	67.29%
Part-aware LSTM [SLNW16]	Skeleton	62.93%	70.27%
ST-LSTM [LSXW16]	Skeleton	65.20%	76.10%
Trust Gate [LSXW16]	Skeleton	69.20%	77.70%
HON4D [OL13]	Depth	30.56%	7.26%
SNV [YT14]	Depth	31.82%	13.61%
SLTEP [JCT$^+$17]	Depth	58 .22%	–
SSSCA-SSLM [SNGW17]	RGB+Depth	74.86%	–
Proposed Method	RGB+Depth	**86.42%**	**89.08%**

can be inferred that the proposed method would still work on these small simple datasets, albeit with a slight increase the final accuracy.

Table 5.9 compares the performances of the proposed method and those of existing methods on this dataset using cross-subject settings as in [HZLZ15]. It can bee seen that, the proposed method outperformed previous methods significantly.

Table 5.8: Results and comparison on the SYSU 3D HOI Dataset using ConvNet and c-ConvNet.

Method	Accuracy
DDIf (ConvNet)	97.92%
VDIf (ConvNet)	91.25%
DDIb (ConvNet)	92.50%
VDIb (ConvNet)	92.92%
DDIf + VDIf (ConvNet)	97.08%
DDIb + VDIb (ConvNet)	94.58%
DDIf + DDIb (ConvNet)	97.92%
VDIf + VDIb (ConvNet)	93.33%
DDIf + VDIf + DDIb + VDIb (ConvNet)	**97.92%**
DDIf (c-ConvNet)	97.92%
VDIf (c-ConvNet)	92.50%
DDIb (c-ConvNet)	92.50%
VDIb (c-ConvNet)	92.50%
DDIf + VDIf (c-ConvNet)	97.08%
DDIb + VDIb (c-ConvNet)	95.00%
DDIf + DDIb (c-ConvNet)	97.92%
VDIf + VDIb (c-ConvNet)	95.00%
DDIf + VDIf + DDIb + VDIb (c-ConvNet)	**98.33%**

Table 5.9: Comparison of the proposed method with previous approaches on SYSU 3D HOI Dataset.

Method	Modality	Accuracy
HON4D [OL13]	Depth	79.22%
MTDA [ZY11]	RGB+Depth	84.21%
JOULE-SVM [HZLZ15]	RGB+Depth	84.89%
Proposed Method	RGB+Depth	**98.33%**

5.2.3.5 Further Analysis

Score-fusion

In this chapter, an effective product-score fusion method was adopted to improve the final accuracy on the four-channel dynamic images. The other two commonly used late score fusion methods are average and maximum score fusion. The comparisons among the three late score fusion methods are shown in Table 5.10. We can see that the product-score fusion method achieved the best results on all the three datasets. This verifies that the four-channel dynamic images, namely, DDIf, VDIf, DDIb and VDIb, provide mutually complementary information.

Table 5.10: Comparison of three different late score fusion methods on the three datasets.

Dataset	Score Fusion Method		
	Max	Average	Product
LAP IsoGD	42.01%	43.48%	**44.80%**
NTU RGB+D (Cross subject)	84.69%	85.86%	**86.42%**
NTU RGB+D (Cross view)	87.01%	87.98%	**89.08%**
SYSU 3D HOI	97.08%	97.92%	**98.33%**

Table 5.11: Comparison of margin α on LAP IsoGD and NTU RGB+D (Cross subject setting) datasets in terms of accuracy(%).

Dataset	α								
	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
LAP IsoGD	40.39	40.46	41.01	40.46	39.68	40.54	39.11	36.57	29.10
NTU RGB+D	82.12	82.64	80.51	80.30	78.56	77.60	-	-	-

Margin parameter, α

In the triplet loss, the parameter α refers to the margin between the anchor/positive and negative. A small alpha value enforces less on the similarities between the anchor/positive and negative, but results in faster convergence for the loss. On the

other hand, a large alpha value may lead to a network with good performance, but slow convergence during training. The channel DDIf&VDIf was taken for example on both LAP IsoGD and NTU RGB+D datasets (cross subject setting) to illustrate the effects of this parameter, and the comparisons are listed in Table 5.11. From the table it can be seen that on LAP IsoGD Dataset, it achieved best accuracy when α was set to 0.2, and with the with the increase of the α, the accuracy decreased significantly. On NTU RGB+D Dataset, best accuracy was obtained when α was set to 0.1, and decreased dramatically when α increased. This evidence suggests that the accuracy is sensitive to this parameter, and it is advisable to set relatively small α values for reasonable results.

Table 5.12: Comparison of weight λ on LAP IsoGD and NTU RGB+D (Cross subject setting) datasets in terms of accuracy(%).

Dataset	λ					
	0	1	2	3	5	7
LAP IsoGD	39.68	39.51	39.61	39.71	41.01	40.13
NTU RGB+D	80.36	81.15	82.64	80.18	80.06	80.11

Weight parameter, λ

In this section, the impact of the weight parameter, λ, as it balances the intra-modality and inter-modality triplet losses is discussed. The channel DDIf&VDIf were taken for example, and the comparisons are listed in Table 5.12. From this Table, it can be seen that assigning a relatively large weight λ (i.e. putting more weight on cross-modality triplet loss), will improve the final accuracy for the difficult datasets (e.g. LAP IsoGD Dataset). However, the accuracy is comparatively less sensitive to this parameter than α.

5.3 Summary

In this section, we proposed two methods to address the research questions 6 and 7 (Section 1.2). Based on the RGB and depth modalities, we first proposed to adopt scene flow for action recognition. Differently from previous late fusion based methods on RGB and depth data, scene flow extracts the real 3D motion and also explicitly preserves the spatial structural information contained in RGB and depth modalities. ConvNets are adopted to transform the scene flow vectors to analogous RGB color space to take advantage of the pre-trained models over ImageNet for action recognition. In the second piece of work, we addressed the problem of using heterogeneous inputs (RGB and depth) in a cooperative manner to train a single network for both homogeneous and heterogeneous action recognition. It was implemented by jointly

training both ranking and classification loss functions. State-of-the-art results were achieved on both large-scale datasets and small datasets for the two proposed methods. However, the key advantage of pooling feature features over temporal axis is to turn a spatial-temporal problem into a spatial problem and to enable us to leverage CNNs. But pooling features over temporal axis may inevitably result in some loss of spatial and/or temporal information for complex actions. For those actions, RNN or spatial-temporal tree based method may have advantage if there are sufficient training data.

Chapter 6

Conclusions and Future Work

This chapter summarizes the contributions of this thesis and discusses the potential directions of future work.

6.1 Conclusions

The central task of this thesis is human action recognition from RGB-D data. We have studied this problem from three perspectives: skeleton-based, depth-based and RGB and Depth based action recognition. Some conclusions are mainly drawn from the proposed works as follows.

In chapter 3, we mainly studied the problem of skeleton-based action recognition using hand-crafted features and ConvNets. For hand-crafted features, we proposed to apply pattern mining method to obtain the most relevant (discriminative, representative and non-redundant) combinations of parts in several continuous frames for action recognition rather than to utilize all the joints as most previous works did. The new representation is much robust to the errors in the features, because the errors are usually not frequent patterns. For ConvNets-based action recognition on skeleton data, we proposed to encode the both spatial configuration and dynamics of joint trajectories into three texture images through color encoding, referred to as Joint Trajectory Maps (JTMs), as the input of ConvNets for action recognition. Such image-based representation enables us to fine-tune existing ConvNets models trained on ImageNet for classification of skeleton sequences without training the whole deep networks afresh.

In chapter 4, we mainly studied the problem of depth-based action recognition using ConvNets. However, there are mainly two reasons that make this task difficult. First, the preclusion of color and texture in depth maps weakens the discriminative representation power of ConvNet models which are texture-driven feature extractor and classifier. Second, existing depth data is relative small-scale. The conventional pipelines are purely data-driven and learn representation directly from the pixels. Such model is likely to be at risk of overfitting when the network is optimized on limited training data. To handle these two restrictions, we took advantage of the representation power of CNN on texture images and at the same time enlarge available training data by encoding depth map sequences into texture color images using the concepts of Depth Motion Maps (DMM) and pseudo-coloring; training data was enlarged by scene rotation on the 3D point cloud. Inspired by the promising

results achieved by rank pooling method on RGB data, we also encoded the depth map sequences into three kinds of dynamic images with rank pooling: Dynamic Depth Images (DDI), Dynamic Depth Normal Images (DDNI) and Dynamic Depth Motion Normal Images (DDMNI). These three representations takes advantages of depth modality that is insensitive to illumination changes and provides better geometric clues, and capture the posture and motion information from three different levels for action recognition. However, due to the unsupervised learning process, the rank pooling method mainly encodes the salient global features in the temporal domain, without mining the discriminative motion patterns in both spatial and temporal domains simultaneously, the conventional rank pooling method is weak in fine-grained action recognition. To deal with this problem, we then proposed to apply rank pooling method on depth map sequences at three hierarchical spatial levels, namely, body level, part level and joint level based on our proposed non-scaling method. Different from previous method that adopted one ConvNet for each human body part, it is proposed to construct one structured dynamic depth image as the input of a ConvNet for each level such that the structured dynamic images not only preserve the spatial-temporal information but also enhance the structure information. Such structured-image-based representation can also take advantages of pre-trained models over ImageNet using ConvNets.

In chapter 5, two methods that adopted both RGB and depth modalities were proposed. In the first method, we proposed to use scene flow to extract the real 3D motion for action recognition. Based on the scene flow vectors, a new representation, namely, Scene Flow to Action Map (SFAM) is proposed for RGB-D action recognition. We adopt a channel transform kernel to transform the scene flow vectors to an optimal color space analogous to RGB. This transformation takes better advantage of the trained ConvNets models over ImageNet for final classification. To exploit the conjoint information in multi-modal features arising from heterogeneous sources (RGB, depth), we then proposed to cooperatively train a single convolutional neural network (named c-ConvNet) on both RGB visual features and depth features, and deeply aggregates the two kinds of features for action recognition. The c-ConvNet enhances the discriminative power of the deeply learned features and weakens the modality discrepancy by jointly optimizing a ranking loss and a softmax loss for both homogeneous and heterogeneous modality-based action recognition. Furthermore, knowledge about the correlations between RGB and depth data are incorporated in the c-ConvNet, and enables the use as additional depth information for the case where only RGB information is available.

6.2 Future Work

In this section, we first highlight some challenges for action recognition. The discussion on challenges then provides a basis to outline potential future research directions.

6.2.1 Challenges

The advent of low-cost RGB-D sensors that have access to extra depth and skeleton data, has motivated the significant development of human motion recognition. Promising have been achieved with deep learning approaches [WLG+16, ZLX17, LSXW16], on several constrained simple datasets, such as MSR-Action3D, Berkeley MHAD and SBU Kinect Interaction. Despite this success, results are far from satisfactory on some large complex datasets, such as ChaLearn LAP IsoGD and NTU RGB+D datasets. In fact, it is still very difficult to build a practical intelligent recognition system. Such goal poses several challenges:

Encoding temporal information. There are several methods to encode temporal information. We can use CNN to extract frame-based features and then conduct temporal fusion [KTS+14], or adopt 3D filter and 3D pooling layers to learn motion features [TBF+15], or use optical/scene flow to extract motion information [SZ14a, WLG+17], or encode the video into images [BFG+16, WLG+16, WLHL16], or use RNN/LSTM to model the temporal dependences [DAHG+15, DWW15, LWH+17]. However, all these approaches have their drawbacks. Temporal fusion method tends to neglect the temporal order; 3D filters and 3D pooling filters have a very rigid temporal structure and they only accept a predefined number of frames as input which is always short; optical/scene flow methods are computationally expensive; sequence to images methods inevitably loses temporal information during encoding; the weight sharing mechanism of RNN/LSTM methods make the sequence matching imprecise, but rather approximated, so an appropriate distance function must be used to predict the match probability. In fact, there is still no perfect method for temporal encoding, and how to model temporal information is a big challenge.

Small training data. Most of available deep learning methods rely on large labeled training data [KTS+14, TBF+15]. However, in practical scenarios, obtaining large labeled training data is costly and laborious, even impossible, especially in medical-related applications. It has been shown that fine-tuning motion-based networks with spatial data (ImageNet) is more effective than training from scratch [SZ14a, WLHL16, BFG+16, WLG+17]. Strategies for data augmentation are also commonly used [WLG+16]. Likewise, training mechanisms to avoid overfit-

ting and control learning rate have also been studied [SHK+14]. However, it is still a challenge to effectively train deep networks from small training data.

Viewpoint variation and occlusion. Viewpoint variation might cause significantly different appearance of the same action, and occlusion would crash the skeleton data. Occlusion includes inter-occlusion caused by other subjects or objects, and self-occlusion created by the object/subject itself. Most of available datasets require subjects to perform actions in a visible and restricted view to avoid occlusion, and this results in limited view data collection and less occlusion. However, occlusion is inevitable in practical scenarios, especially for interactions. This makes it challenging to isolate individuals in overlapping area and extract features of a unique person; leading to the ineffectiveness of many of available approaches [DWW15, SLNW16, LHWL17]. Possible solutions to handle viewpoint variation and occlusion include the use of multi-sensor systems [OCK+13, WNX+14, SLNW16, CYY+17]. The multi-camera systems is able to generate multi-view data, but the drawback is the requirement of synchronization and feature/recognition fusion among different views. This usually increases processing complexity and computation cost. Several methods have been proposed to handle the viewpoint variation and occlusion. [WLG+15] proposed to rotate the depth data in 3D point clouds through different angles to deal with viewpoint invariance; spherical coordinates system corresponding to body center was developed to achieve view-independent motion recognition [HWPVG17]. However, these methods become less effective when occlusion occurs. How to effectively handle occlusion using deep learning methods is a new challenge.

Execution rate variation and repetition. The execution rate may vary due to the different performing styles and states of individuals. The varying rate results in different frames for the same motion. Repetition also bring about this issue. The global encoding methods [HLWL16, KAB+17, LLC17] would become less effective due to the repetition. The commonly used methods to handle this problem is up/down sampling [ZLX+16, ZLX17, LHWL17]. However, sampling methods would inevitable bring redundant or loss of useful information. Effective handling of this problem remains a challenge.

Cross-datasets. Many research works have been carried out to recognize human actions from RGB-D video clips. To learn an effective action classifier, most of the previous approaches rely on enough training labels. When being required to recognize the action in a different dataset, these approaches have to re-train the model using new labels. However, labeling video sequences is a very tedious and time-consuming task, especially when detailed spatial locations and time durations are required. Even though some works have studied this topic [CLH10, SS14, ZLO17], they are all based on hand-crafted features, and the results are far from satisfac-

tory due to the large distribution variances between different datasets, including different scenarios, different modalities, different views, different persons, and even different actions. How to deal with cross-datasets RGB-D motion recognition is a big challenge.

Online motion recognition. Most of available methods rely on segmented data, and their capability for online recognition is quite limited. Even though continuous motion recognition is one improved version where the videos are untrimmed, it still assumes that all the videos are available before processing. Thus, proposal-based methods [SWC16, WXLVG17] can be adopted for offline processing. Differently from continuous motion recognition, online motion recognition aims to receive continuous streams of unprocessed visual data and recognize actions from an unsegmented stream of data in a continuous manner. Generally speaking, there are two main approaches for online recognition, sliding window-based and RNN-based. Sliding window-based methods [CYY+17] are simple extension of segmented-based action recognition methods. They often consider the temporal coherence within the window for prediction and the window-based predictions are further fused to achieve online recognition. However, the performance of these methods are sensitive to the window size which depends on actions and is hard to set. Either too large or too small a window size could lead to significant drop in recognition. For RNN-based methods [MYG+16, LLX+16], even though promising results have been achieved, it is still far from satisfactory in terms of performance. How to design effective practical online recognition system is a big challenge.

Action prediction. We are faced with numerous situations in which we must predict what actions other people are about to do in the near future. Predicting future actions before they are actually executed is a critical ingredient for enabling us to effectively interact with other humans on a daily basis [Ryo11, HDlT14, LCS14, VOL+14]. There are mainly two challenges for this task: first, we need to capture the subtle details inherent in human movements that may imply a future action; second, predictions usually should be carried out as quickly as possible in the social world, when limited prior observations are available. Predicting the action of a person before it is actually executed has a wide range of applications in autonomous robots, surveillance and health care. How to develop effective algorithms for action prediction is really challenging.

6.2.2 Future Research Directions

The discussion on the challenges faced by available methods allows us to outline several future research directions for the development of deep learning methods for motion recognition. While the list is not exhaustive, they point at research activities

that may advance the field.

Hybrid networks. Most of previous methods adopted one type of neural networks for motion recognition. As discussed, there is no perfect solution for temporal encoding using single networks. Even though available works such as C3D+ConvLSTM [ZZSS17] used two types of networks, the cascaded connection makes them dependent on each other during training. How to cooperatively train different kinds of networks would be a good research direction; for example, using the output of CNN to regularize RNN training in parallel.

Simultaneous exploitation of spatial-temporal-structural information. A video sequence has three important inherent properties that should be considered for motion analysis: spatial information, temporal information and structural information. Spatial information refers to the spatial configuration of human body at an instant of time (e.g. relative positions of the human body parts); temporal information characterizes the spatial configuration of the body over time or the dynamics of the body; structural information refers to the coordination and synchronization of body parts over the period of actions, and it describes the relations of spatial configurations of human body across different time slots. Several previous methods tend to exploit the spatio-temporal information for motion recognition, however, structural information contained in the video is rarely explicitly mined. Concurrent mining of these three kinds of information with deep learning would be an interesting topic in the future [JZSS16].

Fusion of multiple modalities. While significant progress has been achieved by singly using RGB, skeleton or depth modality, effective deep networks for fusion of multi-modal data would be a promising direction. For example, methods such as SFAM [WLG+17] and PRNN [SK17] have pioneered the research in this direction. The work SFAM [WLG+17] proposed to extract scene flow for motion analysis. The strategy of fusing the RGB and depth modalities at the outset allowed the capture of rich 3D motion information. In PRNN [SK17] the concept of privileged information (side information) was introduced for deep networks training and showed some promise. So far, most methods considered the three modalities as separate channels and fused them at later or score stage using different fusion methods without cooperatively exploiting their complementary properties. Cooperative training using different modalities would be a promising research area.

Large-scale datasets. With the development of data-hungry deep learning approach, there is demand for large scale RGB-D datasets. Even though there are several large datasets, such as NTU RGB+D Dataset [SLNW16] and ChaLearn LAP IsoGD Dataset [WLZ+16], they are focused on specific tasks. Various large-scale RGB-D datasets are needed to facilitate research in this field. For instance, large-scale fine-grained RGB-D motion recognition datasets and large-scale occlusion-

based RGB-D motion recognition datasets are urgently needed.

Zero/One-shot learning. As discussed, it is not always easy to collect large scale labeled data. Learning from a few examples remains a key challenge in machine learning. Despite recent advances in important domains such as vision and language, the standard supervised deep learning paradigm does not offer a satisfactory solution for learning new concepts rapidly from little data. How to adopt deep learning methods for zero/one shot RGB-D-based motion recognition would be an interesting research direction. Zero/one-shot learning is about being able to recognize gesture/action classes that are never seen or only one training sample per class before. This type of recognition should carry embedded information universal to all other gestures/actions. In the past few year, there are some works on zero/one-shot learning. For example, Wan et al. [WGL16] proposed the novel spatial-temporal features for one-shot learning gesture recognition and have got promising performances on Chalearn Gesture Dataset CGD) [GAJE14]. For zero-shot learning, Madapana and Wachs [MW17] proposed a new paradigm based on adaptive learning which it is possible to determine the amount of transfer learning carried out by the algorithm and how much knowledge is acquired for a new gesture observation. However, the mentioned works are used traditional methods (such as bag of visual words model [WRLD13]). How to adopt deep learning methods for zero/one shot RGB-D based motion recognition would be an interesting research direction when it used only very few training samples.

Outdoor practical scenarios. Although lots of RGB-D datasets have been collected during the last few years, there is a big gap between the collected datasets and wild environment due to constrained environment setting and insufficient categories and samples. For example, most available datasets do not involve much occlusion cases probably due to the collapse of skeleton dataset in case of occlusion. However, in practical scenarios, occlusion is inevitable. How to recover or find cues from multi-modal data for such recognition tasks would be an interesting research direction. Besides, with the development of depth sensors, further distances could be captured, and recognition in outdoor practical scenarios will gain the attention of researchers.

Unsupervised learning/Self-learning. Collecting labeled datasets are time-consuming and costly, hence learning from unsupervised video data is required. Mobile robots mounted with RGB-D cameras need to continuously learn from the environment and without human intervention. How to automatically learn from the unlabeled data stream to improve the learning capability of deep networks would be a fruitful and useful research direction. Generative Adversarial Net (GAN) [HE16] has got much processes recently in image generation task, such as face generation, text-to-image task. Besides, it also can be used for recognition task. For example,

Luan et al. [TYL17] proposed a Disentangled Representation learning Generative Adversarial Networks (DR-GAN) for pose-invariant face recognition. Therefore, we believe the GAN-based techniques also can be used for action/gesture recognition, which is a great excited direction for researches. Carl et al. [VPT16b] proposed a generative adversarial network for video with spatial-temporal convolutional architecture that untangles the scene's foreground from backgrounds. This is an initial works to capitalize on large amounts of unlabeled video in order to learn a model of scene dynamic for both video recognition tasks (e.g. action classification) and video generation tasks (e.g. future prediction). Increasing research will be reported in the coming years on GAN-based methods for video-based recognition.

Online motion recognition and prediction. Online motion recognition and prediction is required in practical applications, and arguably this is the final goal of motion recognition systems. Differently from segmented recognition, online motion recognition requires the analysis of human behavior in a continuous manner, and prediction aims to recognize or anticipate actions that would happen. How to design effective online recognition and prediction systems with deep learning methods has attracted researchers' eyes, for example, Vondrick et al. [VPT16a] introduced a framework that capitalizes on temporal structure in unlabeled video to learn to anticipate human actions and objects based on CNN, and it is likely to emerge as an active research area.

Bibliography

[AC99] JK Aggarwal and Q Cai. Human motion analysis: A review. In *Computer Vision and Image Understanding*, 1999.

[ACM15] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 37–45, 2015.

[AMZV14] Salah Althloothi, Mohammad H. Mahoor, Xiao Zhang, and Richard M. Voyles. Human activity recognition using multi-features and multiple kernel learning. *Pattern Recognition*, 47(5):1800–1812, 2014.

[AZGA06] Besma R Abidi, Yue Zheng, Andrei V Gribok, and Mongi A Abidi. Improving weapon detection in single energy X-ray images through pseudocoloring. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 36(6):784–796, 2006.

[BD01a] Aaron F. Bobick and James W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:257–267, 2001.

[BD01b] Aaron F. Bobick and James W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on pattern analysis and machine intelligence*, 23(3):257–267, 2001.

[BFG+16] Hakan Bilen, Basura Fernando, Efstratios Gavves, Andrea Vedaldi, and Stephen Gould. Dynamic image networks for action recognition. In *CVPR*, 2016.

[BGS+05] Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1395–1402, 2005.

[BMA12] Victoria Bloom, Dimitrios Makris, and Vasileios Argyriou. G3D: A gaming action dataset and real time action recognition evaluation framework. In *CVPRW*, 2012.

[BMW+11] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt. Sequential deep learning for human action recognition. In *International Workshop on Human Behavior Understanding*, pages 29–39. Springer, 2011.

[BSF94] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[CB95] Lee W Campbell and Aaron F Bobick. Recognition of human body motion using phase space constraints. In *Proc. International Conference on Computer Vision (ICCV)*, pages 624–630, 1995.

[CFHG17] Anoop Cherian, Basura Fernando, Mehrtash Harandi, and Stephen Gould. Generalized rank pooling for activity recognition. In *CVPR*, 2017.

[CH09] Ming yu Chen and Alexander Hauptmann. MoSOFT: Recognizing human actions in surveillance videos. *Transform*, pages 1–16, 2009.

[Che10] Bo Chen. *Deep learning of invariant spatio-temporal features from video*. PhD thesis, University of British Columbia, 2010.

[CHKB16] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, and Richard Bowden. Using convolutional 3d neural networks for user-independent continuous gesture recognition. In *Proceedings of ICPRW*, 2016.

[CHL05] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 539–546. IEEE, 2005.

[CJK15] Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 168–172, 2015.

[CL11] Chih Chung Chang and Chih Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[CLH10] Liangliang Cao, Zicheng Liu, and Thomas S Huang. Cross-dataset action detection. In *Computer vision and pattern recognition (CVPR)*, pages 1998–2005. IEEE, 2010.

[CLS15] Guilhem Chéron, Ivan Laptev, and Cordelia Schmid. P-cnn: Pose-based cnn features for action recognition. In *ICCV*, pages 3218–3226, 2015.

[CLY+16] Xiujuan Chai, Zhipeng Liu, Fang Yin, Zhuang Liu, and Xilin Chen. Two streams recurrent neural networks for large-scale continuous gesture recognition. In *Proceedings of ICPRW*, 2016.

[CLZ+16] Chen Chen, Mengyuan Liu, Baochang Zhang, Jungong Han, Junjun Jiang, and Hong Liu. 3D action recognition using multi-temporal depth motion maps and fisher vector. In *IJCAI*, pages 3331–3337, 2016.

[CMU01] *CMU Graphics Lab Motion Capture Database, http://mocap.cs.cmu.edu/*, 2001.

[COK+13] Rizwan Chaudhry, Ferda Ofli, Gregorij Kurillo, Ruzena Bajcsy, and Rene Vidal. Bio-inspired dynamic 3D discriminative skeletal features for human action recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 471–478, 2013.

[CPLCPFR14] Alexandros Andre Chaaraoui, JosÃĺ RamÃşn Padilla-LÃşpez, Pau Climent-PÃĺrez, and Francisco FlÃşrez-Revuelta. Evolutionary joint selection to improve human action recognition with RGB-D devices. *Expert Systems with Applications*, 41(3):786 – 794, 2014.

[CPLFR13] A.A. Chaaraoui, J.R. Padilla-Lopez, and F. Florez Revuelta. Fusion of skeletal and silhouette-based features for human action recognition with RGB-D devices. In *Proc. IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 91–97, Dec 2013.

[CVMG+14] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[CWF13] Lulu Chen, Hong Wei, and James Ferryman. A survey of human motion analysis using depth imagery. *Pattern Recognition Letters*, 34(15):1995–2006, 2013.

[CYHH07] Hong Cheng, Xifeng Yan, Jiawei Han, and Chih Wei Hsu. Discriminative frequent pattern analysis for effective classification. In *Proc. .IEEE International Conference on Data Engineering (ICDE)*, pages 716–725, 2007.

[CYL16] Hong Cheng, Lu Yang, and Zicheng Liu. Survey on 3d hand gesture recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(9):1659–1673, 2016.

[CYY+17] Liu Chunhui, Hu Yueyu, Li Yanghao, Song Sijie, and Liu Jiaying. Pku-mmd: A large scale benchmark for continuous multi-modal human action understanding. *arXiv preprint arXiv:1703.07475*, 2017.

[DAHG+15] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[DFW15] Yong Du, Yun Fu, and Liang Wang. Skeleton based action recognition with convolutional neural network. In *Pattern Recognition (ACPR), 2015 3rd IAPR Asian Conference on*, pages 579–583. IEEE, 2015.

[DFW16] Yong Du, Yun Fu, and Liang Wang. Representation learning of temporal dynamics for skeleton-based action recognition. *IEEE Transactions on Image Processing*, 25(7):3010–3022, 2016.

[DP93] Trevor Darrell and Alex Pentland. Space-time gestures. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 335–340, 1993.

[DRCB05] Piotr Dollár, Vincent Rabaud, Garrison Cottrell, and Serge Belongie. Behavior recognition via sparse spatio-temporal features. In *Proc. Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, pages 65–72, 2005.

[DWB+15] Maxime Devanne, Hazem Wannous, Stefano Berretti, Pietro Pala, Mohamed Daoudi, and Alberto Del Bimbo. 3-D human action recognition by shape analysis of motion trajectories on riemannian manifold. *IEEE transactions on cybernetics*, 45(7):1340–1352, 2015.

[DWW15] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*, 2015.

[DZW+16] Jiali Duan, Shuai Zhou, Jun Wan, Xiaoyuan Guo, and Stan Z Li. Multi-modality fusion based on consensus-voting and 3d convolution for isolated gesture recognition. *arXiv preprint arXiv:1611.06689*, 2016.

[EPLW+16] H. J. Escalante, V. Ponce-LÃşpez, J. Wan, M. A. Riegler, B. Chen, A. ClapÃĺs, S. Escalera, I. Guyon, X. BarÃş, P. Halvorsen, H. MÃijller, and M. Larson. Chalearn joint contest on multimedia challenges beyond visual analysis: An overview. In *Proceedings of ICPRW*, 2016.

[FAHG16] Basura Fernando, Peter Anderson, Marcus Hutter, and Stephen Gould. Discriminative hierarchical rank pooling for activity recognition. In *CVPR*, 2016.

[FFP05] Li Fei Fei and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 524–531, 2005.

[FFT14] Basura Fernando, Elisa Fromont, and Tinne Tuytelaars. Mining mid-level features for image classification. *International Journal of Computer Vision*, 108(3):186–203, 2014.

[FG16] Basura Fernando and Stephen Gould. Learning end-to-end video classification with rank-pooling. In *ICML*, 2016.

[FGM+16] Basura Fernando, Stratis Gavves, Oramas Mogrovejo, José Antonio, Amir Ghodrati, and Tinne Tuytelaars. Rank pooling for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[FGO+15] Basura Fernando, Efstratios Gavves, Jose Oramas, Amir Ghodrati, and Tinne Tuytelaars. Modeling video evolution for action recognition. In *CVPR*, 2015.

[FMNK12] Simon Fothergill, Helena M. Mentis, Sebastian Nowozin, and Pushmeet Kohli. Instructing people for training gestural interactive systems. In *ACM Conference on Computer-Human Interaction (ACM HCI)*, 2012.

[FPW16] Christoph Feichtenhofer, Axel Pinz, and Richard Wildes. Spatiotemporal residual networks for video action recognition. In *Advances in Neural Information Processing Systems*, pages 3468–3476, 2016.

[FPZ16] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1933–1941, 2016.

[GAJE14] Isabelle Guyon, Vassilis Athitsos, Pat Jangyodsuk, and Hugo Jair Escalante. The chalearn gesture dataset (CGD 2011). *Machine Vision and Applications*, 25(8):1929–1951, 2014.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[GD07] Abhinav Gupta and Larry S Davis. Objects in action: An approach for combining action understanding and object perception. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[GIB09] Andrew Gilbert, John Illingworth, and Richard Bowden. Fast realistic multi-action recognition using mined dense spatio-temporal features. In *Proc. International Conference on Computer Vision (ICCV)*, pages 925–931, 2009.

[GL14] Guodong Guo and Alice Lai. A survey on still image based human action recognition. *Pattern Recognition*, 47(10):3343–3361, 2014.

[GPAM+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[GTHES13] Mohammad A. Gowayyed, Marwan Torki, Mohamed E. Hussein, and Motaz El-Saban. Histogram of oriented displacements (HOD): Describing trajectories of human joints for action recognition. In *IJCAI*, pages 1351–1357, 2013.

[GWZZ17] Zhimin Gao, Lei Wang, Luping Zhou, and Jianjia Zhang. Hep-2 cell image classification with deep convolutional neural networks. *IEEE journal of biomedical and health informatics*, 21(2):416–428, 2017.

[GYS08] Taylor Goodhart, Pingkun Yan, and Mubarak Shah. Action recognition using spatio-temporal regularity based features. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 745–748, 2008.

[HAS+17] Escalante Hugo, ClapÃls Albert, Escalera Sergio, Bella Marco, Asadi Maryam, Baro Xavier, Ponce-LÃşpez Victor, e Guyon Isabell, and Kasaei Shohreh. A survey on deep learning based approaches for action and gesture recognition in image sequences. In *FG*, 2017.

[HB14] Simon Hadfield and Richard Bowden. Scene particles: Unregularized particle-based scene flow estimation. *IEEE transactions on pattern analysis and machine intelligence*, 36(3):564–576, 2014.

[HDlT14] Minh Hoai and Fernando De la Torre. Max-margin early event detectors. *International Journal of Computer Vision*, 107(2):191–202, 2014.

[HE16] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, pages 4565–4573, 2016.

[HFR14] Michael Hornacek, Andrew Fitzgibbon, and Carsten Rother. Sphereflow: 6 DoF scene flow from RGB-D pairs. In *CVPR*, pages 3526–3533, 2014.

[HHP17] Samitha Herath, Mehrtash Harandi, and Fatih Porikli. Going deeper into action recognition: A survey. *Image and Vision Computing*, 60:4–21, 2017.

[Hin02] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[Hin09] Geoffrey E Hinton. Deep belief networks. *Scholarpedia*, 4(5):5947, 2009.

[Hin10] Geoffrey Hinton. A practical guide to training restricted boltzmann machines. *Momentum*, 9(1):926, 2010.

[HLWL16] Yonghong Hou, Zhaoyang Li, Pichao Wang, and Wanqing Li. Skeleton optical spectra based action recognition using convolutional neural networks. In *Circuits and Systems for Video Technology, IEEE Transactions on*, pages 1–5, 2016.

[HRHZ17] Fei Han, Brian Reily, William Hoff, and Hao Zhang. Space-time representation of people based on 3d skeletal data: A review. *Computer Vision and Image Understanding*, 2017.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[HTGES13] Mohamed E. Hussein, Marwan Torki, Mohammad A. Gowayyed, and Motaz El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3D joint locations. In *IJCAI*, pages 2466–2472, 2013.

[HWPVG17] Zhiwu Huang, Chengde Wan, Thomas Probst, and Luc Van Gool. Deep learning on lie groups for skeleton-based action recognition. In *CVPR*, 2017.

[HZ03] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[HZLZ15] Jian-Fang Hu, Wei-Shi Zheng, Jianhuang Lai, and Jianguo Zhang. Jointly learning heterogeneous features for RGB-D activity recognition. In *CVPR*, 2015.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[I+16] Earnest Paul Ijjina et al. Classification of human actions using pose-based features and stacked auto encoder. *Pattern Recognition Letters*, 83:268–277, 2016.

[IB00] Yuri A. Ivanov and Aaron F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):852–872, 2000.

[JCT+17] Xiaopeng Ji, Jun Cheng, Dapeng Tao, Xinyu Wu, and Wei Feng. The spatial laplacian and temporal energy pyramid representation for human action recognition using depth sequences. *Knowledge-Based Systems*, 2017.

[JF16] Chengcheng Jia and Yun Fu. Low-rank tensor subspace learning for rgb-d action recognition. *IEEE Transactions on Image Processing*, 25(10):4641–4652, 2016.

[JG16] Dinesh Jayaraman and Kristen Grauman. Slow and steady feature analysis: higher order temporal coherence in video. In *CVPR*, 2016.

[JKDF14] Chengcheng Jia, Yu Kong, Zhengming Ding, and Yun Raymond Fu. Latent tensor transfer learning for rgb-d action recognition. In *ACM MM*, 2014.

[Joh75] Gunnar Johansson. Visual motion perception. *Scientific American*, 1975.

[Joh12] Jayme Johnson. Not seeing is not believing: improving the visibility of your fluorescence images. *Molecular Biology of the Cell*, 23(5):754–757, 2012.

[JSD+14] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross B Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM MM*, 2014.

[JSGJC15] Mariano Jaimez, Mohamed Souiai, Javier Gonzalez-Jimenez, and Daniel Cremers. A primal-dual framework for real-time dense RGB-D scene flow. In *ICRA*, pages 98–104, 2015.

[JSWP07] Hueihan Jhuang, Thomas Serre, Lior Wolf, and Tomaso Poggio. A biologically inspired system for action recognition. In *Proc. IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.

[JUKK11] Ahmad Jalal, Md Zia Uddin, Jeong Tai Kim, and Tae-Seong Kim. Recognition of human home activities via depth silhouettes and $\mathbb{R}$ transformation for smart homes. *Indoor and Built Environment*, pages 467–475, 2011.

[JWF16] Shuhui Jiang, Yue Wu, and Yun Fu. Deep bi-directional cross-triplet embedding for cross-domain clothing retrieval. In *ACM MM*, 2016.

[JXYY13] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3D convolutional neural networks for human action recognition. *TPAMI*, 35(1):221–231, 2013.

[JZSS16] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5308–5317, 2016.

[JZW+15] Feng Jiang, Shengping Zhang, Shen Wu, Yang Gao, and Debin Zhao. Multi-layered gesture recognition with kinect. *Journal of Machine Learning Research*, 16(2):227–254, 2015.

[KAB+17] Qiuhong Ke, Senjian An, Mohammed Bennamoun, Ferdous Sohel, and Farid Boussaid. Skeletonnet: Mining deep part features for 3d action recognition. *IEEE Signal Processing Letters*, 2017.

[KBA+17] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. A new representation of skeleton sequences for 3d action recognition. In *CVPR*, 2017.

[KCL16] Suha Kwak, Minsu Cho, and Ivan Laptev. Thin-slicing for pose: Learning to understand pose without explicit pose estimation. In *CVPR*, 2016.

[KF15] Yu Kong and Yun Fu. Bilinear heterogeneous information machine for RGB-D action recognition. In *CVPR*, 2015.

[KF17] Yu Kong and Yun Fu. Max-margin heterogeneous information machine for rgb-d action recognition. *IJCV*, pages 1–22, 2017.

[KGS13] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from RGB-D videos. In *International Journal of Robotics Research (IJRR)*, volume 32, pages 951–970, July 2013.

[KMS08] Alexander Kläser, Marcin Marszałek, and Cordelia Schmid. A spatio-temporal descriptor based on 3D-gradients. In *Proc. British Machine Vision Conference*, pages 995–1004, 2008.

[KR17] Tae Soo Kim and Austin Reiter. Interpretable 3d human action analysis with temporal convolutional networks. In *CVPR*, 2017.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1106–1114, 2012.

[KTF16] Dimitris Kastaniotis, Ilias Theodorakopoulos, and Spiros Fotopoulos. Pose-based gait recognition with local gradient descriptors and hierarchically aggregated residuals. *Journal of Electronic Imaging*, 25(6):063019–063019, 2016.

[KTS+14] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1725–1732, 2014.

[KYF05] Christian Kanzow, Nobuo Yamashita, and Masao Fukushima. Withdrawn: Levenberg–marquardt methods with strong local convergence properties for solving nonlinear equations with convex con-

straints. *Journal of Computational and Applied Mathematics*, 173(2):321–343, 2005.

[Lap05] Ivan Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005.

[LBBH98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[LBD+90] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.

[LCNS16] Ivan Lillo, Juan Carlos Niebles, and Alvaro Soto. A hierarchical pose-based approach to complex action understanding using dictionaries of actionlets and motion poselets. In *CVPR*, 2016.

[LCS14] Tian Lan, Tsung-Chuan Chen, and Silvio Savarese. A hierarchical representation for future action prediction. In *European Conference on Computer Vision*, pages 689–704. Springer, 2014.

[LFV+17] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *CVPR*, 2017.

[LHWL17] Chuankun Li, Yonghong Hou, Pichao Wang, and Wanqing Li. Joint distance maps based action recognition with convolutional neural networks. *IEEE Signal Processing Letters*, 24(5):624–628, 2017.

[LJT14] Cewu Lu, Jiaya Jia, and Chi-Keung Tang. Range-sample depth feature for action recognition. In *CVPR*, 2014.

[LLC17] Mengyuan Liu, Hong Liu, and Chen Chen. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68:346–362, 2017.

[LLL+15] Zhengzhong Lan, Ming Lin, Xuanchong Li, Alex G Hauptmann, and Bhiksha Raj. Beyond gaussian pyramid: Multi-skip feature stacking for action recognition. In *CVPR*, pages 204–212, 2015.

[LLX+16] Yanghao Li, Cuiling Lan, Junliang Xing, Wenjun Zeng, Chunfeng Yuan, and Jiaying Liu. Online human action detection using joint classification-regression recurrent neural networks. In *European Conference on Computer Vision*, pages 203–220. Springer, 2016.

[LMT+16a] Yunan Li, Qiguang Miao, Kuan Tian, Yingying Fan, Xin Xu, Rui Li, and Jianfeng Song. Large-scale gesture recognition with a fusion of rgb-d data based on the c3d model. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 25–30. IEEE, 2016.

[LMT+16b] Yunan Li, Qiguang Miao, Kuan Tian, Yingying Fan, Xin Xu, Rui Li, and Jianfeng Song. Large-scale gesture recognition with a fusion of RGB-D data based on the C3D model. In *Proceedings of ICPRW*, 2016.

[LPH+17] Zelun Luo, Boya Peng, De-An Huang, Alexandre Alahi, and Li Fei-Fei. Unsupervised learning of long-term motion dynamics for videos. In *CVPR*, 2017.

[LRVH16] Colin Lea, Austin Reiter, René Vidal, and Gregory D Hager. Segmental spatiotemporal cnns for fine-grained action segmentation. In *European Conference on Computer Vision*, pages 36–52. Springer, 2016.

[LS13a] Li Liu and Ling Shao. Learning discriminative representations from RGB-D video data. In *Proc. International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1493–1500, 2013.

[LS13b] Li Liu and Ling Shao. Learning discriminative representations from rgb-d video data. In *IJCAI*, 2013.

[LSKW16] Guy Lev, Gil Sadeh, Benjamin Klein, and Lior Wolf. Rnn fisher vectors for action recognition and image annotation. In *European Conference on Computer Vision*, pages 833–850. Springer, 2016.

[LSWT16] Xiaoxiang Liu, Lingxiao Song, Xiang Wu, and Tieniu Tan. Transferring deep representation for nir-vis heterogeneous face recognition. In *ICB*, 2016.

[LSXW16] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. Spatio-temporal LSTM with trust gates for 3D human action recognition. In *ECCV*, 2016.

[LWH+17] Jun Liu, Gang Wang, Ping Hu, Ling-Yu Duan, and Alex C.Kot. Global context-aware attention lstm networks for 3d action recognition. In *CVPR*, 2017.

[LWQ14] Jiajia Luo, Wei Wang, and Hairong Qi. Spatio-temporal feature extraction and representation for RGB-D human action recognition. *Pattern Recognition Letters*, 50:139 – 148, 2014.

[LXN+16] An-An Liu, Ning Xu, Wei-Zhi Nie, Yu-Ting Su, Yongkang Wong, and Mohan Kankanhalli. Benchmarking a multimodal and multiview and interactive dataset for human action recognition. *IEEE Transactions on cybernetics*, 2016.

[LZL08] W. Li, Z. Zhang, and Z. Liu. Expandable data-driven graphical modeling of human actions based on salient postures. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(11):1499–1510, 2008.

[LZL10] Wanqing Li, Zhengyou Zhang, and Zicheng Liu. Action recognition based on a bag of 3D points. In *CVPRW*, 2010.

[LZT16] Zhi Liu, Chenyang Zhang, and Yingli Tian. 3d-based deep convolutional neural network for action recognition with depth sequences. *Image and Vision Computing*, 55:93–100, 2016.

[LZYN11] Quoc V Le, Will Y Zou, Serena Y Yeung, and Andrew Y Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3361–3368. IEEE, 2011.

[MB06] Ju Man and Bir Bhanu. Individual recognition using gait energy image. *IEEE transactions on pattern analysis and machine intelligence*, 28(2):316–322, 2006.

[MCL16] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multiscale video prediction beyond mean square error. In *Proc. International Conference on Learning Representations (ICLR)*, 2016.

[MG15] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *CVPR*, pages 3061–3070, 2015.

[MH07] Roland Memisevic and Geoffrey Hinton. Unsupervised learning of image transformations. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 1–8. IEEE, 2007.

[MR06] Meinard Müller and Tido Röder. Motion templates for automatic classification and retrieval of motion capture data. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation (ACM SIGGRAPH/ESCA)*, pages 137–146, 2006.

[MRC+07] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber. Documentation mocap database hdm05. Technical Report CG-2007-2, Universität Bonn, June 2007.

[MT16] Behrooz Mahasseni and Sinisa Todorovic. Regularizing long short term memory with 3d human-skeleton sequences for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3054–3062, 2016.

[MW17] Naveen Madapana and Juan P Wachs. A semantical & analytical approach for zero shot gesture learning. In *FG workshop*, 2017.

[MYG+16] Pavlo Molchanov, Xiaodong Yang, Shalini Gupta, Kihwan Kim, Stephen Tyree, and Jan Kautz. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4207–4215, 2016.

[NHV+15] Joe Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, 2015.

[NWJ15] Siqi Nie, Ziheng Wang, and Qiang Ji. A generative restricted boltzmann machine based method for high-dimensional motion data modeling. *CVIU*, 136:14–22, 2015.

[NWM11] B. Ni, Gang Wang, and P. Moulin. A colour-depth video database for human daily activity recognition. In *ICCVW*, 2011.

[NWM13] Bingbing Ni, Gang Wang, and Pierre Moulin. RGBD-HuDaAct: A color-depth video database for human daily activity recognition. In *Consumer Depth Cameras for Computer Vision*, pages 193–208. 2013.

[OBT13a] E. Ohn-Bar and M.M. Trivedi. Joint angles similarities and HOG2 for action recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 465–470, 2013.

[OBT13b] Eshed Ohn-Bar and Mohan Trivedi. Joint angles similarities and HOG2 for action recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 465–470, 2013.

[OCK+13] Ferda Ofli, Rizwan Chaudhry, Gregorij Kurillo, René Vidal, and Ruzena Bajcsy. Berkeley MHAD: A comprehensive multimodal human action database. In *Proc. IEEE Workshop on Applications of Computer Vision (WACV)*, pages 53–60, 2013.

[OL13] O. Oreifej and Zicheng Liu. HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences. In *CVPR*, 2013.

[PA04] Sangho Park and Jake K Aggarwal. A hierarchical bayesian network for event recognition of human actions and interactions. *Multimedia systems*, 10(2):164–179, 2004.

[PLC16] Liliana Lo Presti and Marco La Cascia. 3d skeleton-based human action classification: A survey. *Pattern Recognition*, 53:130–147, 2016.

[Pop10] Ronald Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010.

[PVDOD+16] Lionel Pigou, Aäron Van Den Oord, Sander Dieleman, Mieke Van Herreweghe, and Joni Dambre. Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video. *International Journal of Computer Vision*, pages 1–10, 2016.

[PWWQ16] Xiaojiang Peng, Limin Wang, Xingxing Wang, and Yu Qiao. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *Computer Vision and Image Understanding*, 2016.

[PZQP14] Xiaojiang Peng, Changqing Zou, Yu Qiao, and Qiang Peng. Action recognition with stacked fisher vectors. In *ECCV*, pages 581–595, 2014.

[QBDC14] Julian Quiroga, Thomas Brox, Frédéric Devernay, and James Crowley. Dense semi-rigid scene flow estimation from RGBD images. In *ECCV*, pages 567–582. 2014.

[RJ86] Lawrence Rabiner and B Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.

[RM16] Hossein Rahmani and Ajmal Mian. 3d action recognition from novel viewpoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1506–1515, 2016.

[RMHM14] Hossein Rahmani, Arif Mahmood, Du Q Huynh, and Ajmal Mian. HOPC: Histogram of oriented principal components of 3D pointclouds for action recognition. In *Proc. European Conference on Computer Vision (ECCV)*, pages 742–757. 2014.

[RMMH14] Hossein Rahmani, Arif Mahmood, Ajmal Mian, and Du Huynh. Real time action recognition using histograms of depth gradients and random decision forests. In *Proc. IEEE Winter Applications of Computer Vision Conference (WACV)*, pages 14–19, 2014.

[Ryo11] Michael S Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1036–1043. IEEE, 2011.

[SFC+11] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1297–1304, 2011.

[SHK+14] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[SJP11] J. Smisek, M. Jancosek, and T. Pajdla. 3D with kinect. In *CVPRW*, 2011.

[SJYS15] Lin Sun, Kui Jia, Dit-Yan Yeung, and Bertram E Shi. Human action recognition using factorized spatio-temporal convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4597–4605, 2015.

[SK17] Zhiyuan Shi and Tae-Kyun Kim. Learning and refining of privileged information-based rnns for action recognition from depth sequences. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[SKP15] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[SKS16] Shikhar Sharma, Ryan Kiros, and Ruslan Salakhutdinov. Action recognition using visual attention. *ICLRW*, 2016.

[SL13] Zhanpeng Shao and YF Li. A new descriptor for multiple 3D motion trajectories recognition. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 4749–4754, 2013.

[SLJ+15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[SLNW16] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. NTU RGB+ D: A large scale dataset for 3D human activity analysis. In *CVPR*, 2016.

[SLX+17] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[SMS15] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *ICML*, pages 843–852, 2015.

[SNGW17] Amir Shahroudy, Tian-Tsong Ng, Yihong Gong, and Gang Wang. Deep multimodal feature analysis for action recognition in rgb+ d videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[SNYW16] Amir Shahroudy, Tian-Tsong Ng, Qingxiong Yang, and Gang Wang. Multimodal multipart learning for action recognition in depth videos. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2123–2129, 2016.

[SPMV13] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image classification with the fisher vector: Theory and practice. *International journal of computer vision*, 105(3):222–245, 2013.

[SPSS12] Jaeyong Sung, Colin Ponce, Bart Selman, and Ashutosh Saxena. Unstructured human activity detection from RGBD images. In *Proc. IEEE Computer Society Conference on Robotics and Automation (ICRA)*, pages 842–849, 2012.

[SS04] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[SS14] W. Sultani and I. Saleemi. Human action recognition across datasets by foreground-weighted histogram decomposition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 764–771, 2014.

[SSP15] Deqing Sun, Erik B Sudderth, and Hanspeter Pfister. Layered RGBD scene flow estimation. In *CVPR*, pages 548–556, 2015.

[STLY14] Y. Song, J. Tang, F. Liu, and S. Yan. Body surface context: A new robust feature for action recognition from depth videos. *Circuits and Systems for Video Technology, IEEE Transactions on*, (99):952–964, 2014.

[STT13] R. Salakhutdinov, J. B. Tenenbaum, and A. Torralba. Learning with hierarchical-deep models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1958–1971, Aug 2013.

[SWC16] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1049–1058, 2016.

[SWT14] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. In *CVPR*, 2014.

[SZ14a] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

[SZ14b] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[TBF+15] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *ICCV*, 2015.

[TCSU08] Pavan Turaga, Rama Chellappa, Venkatramana S Subrahmanian, and Octavian Udrea. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488, 2008.

[TFLB10] Graham W Taylor, Rob Fergus, Yann LeCun, and Christoph Bregler. Convolutional learning of spatio-temporal features. In *Proc. European Conference on Computer Vision (ECCV)*, pages 140–153. 2010.

[The07] Fabian J Theis. Towards a general independent subspace analysis. In *Advances in Neural Information Processing Systems*, pages 1361–1368, 2007.

[TJBB04] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Sharing clusters among related groups: Hierarchical dirichlet processes. In *NIPS*, pages 1385–1392, 2004.

[TKEF14] Ilias Theodorakopoulos, Dimitris Kastaniotis, George Economou, and Spiros Fotopoulos. Pose-based human action recognition via sparse representation in dissimilarity space. *Journal of Visual Communication and Image Representation*, 25(1):12 – 23, 2014.

[TYL17] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In *CVPR*, volume 4, page 7, 2017.

[UAUA03] Takeaki Uno, Tatsuya Asai, Yuzo Uchida, and Hiroki Arimura. LCM: An efficient algorithm for enumerating frequent closed item sets. In *Proc. Workshop on Frequent Itemset Mining Implementations (FIMI)*, 2003.

[VA13] Sarvesh Vishwakarma and Anupam Agrawal. A survey on activity recognition and behavior understanding in video surveillance. *The Visual Computer*, 29(10):983–1009, 2013.

[VAC14] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3D skeletons as points in a lie group. In *CVPR*, 2014.

[VAC16] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. R3DG features: Relative 3d geometry-based skeletal representations for human action recognition. *CVIU*, 152:155 – 166, 2016.

[VC16] Raviteja Vemulapalli and Rama Chellappa. Rolling rotations for recognizing human actions from 3D skeletal data. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[VL15] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. In *Proceeding of the ACM Int. Conf. on Multimedia*, 2015.

[VLL+10] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

[VLS16] Gül Varol, Ivan Laptev, and Cordelia Schmid. Long-term temporal convolutions for action recognition. *arXiv preprint arXiv:1604.04494*, 2016.

[VNO+12] AntÃťnio Wilson Vieira, Erickson R. Nascimento, Gabriel L. Oliveira, Zicheng Liu, and Mario Fernando Montenegro Campos. STOP: Space-time occupancy patterns for 3D action recognition from depth map sequences. In *Proc. Countdown for Iberoamerican Congress on Pattern Recognition (CIARP)*, pages 252–259, 2012.

[VOL+14] Tuan-Hung Vu, Catherine Olsson, Ivan Laptev, Aude Oliva, and Josef Sivic. Predicting actions from static scenes. In *European Conference on Computer Vision*, pages 421–436. Springer, 2014.

[VPT16a] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabeled video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 98–106, 2016.

[VPT16b] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, pages 613–621, 2016.

[VRCK05] Srinivas Vedula, Peter Rander, Robert Collins, and Takeo Kanade. Three-dimensional scene flow. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(3):475–480, 2005.

[VZQ15] Vivek Veeriah, Naifan Zhuang, and Guo-Jun Qi. Differential recurrent neural networks for action recognition. In *ICCV*, 2015.

[Wer88] Paul J Werbos. Generalization of backpropagation with application to a recurrent gas market model. *Neural networks*, 1(4):339–356, 1988.

[WFG16] Xiaolong Wang, Ali Farhadi, and Abhinav Gupta. Actions˜ transformations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2658–2667, 2016.

[WGL16] J. Wan, G. Guo, and S. Z. Li. Explore efficient local features from RGB-D data for one-shot learning gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(8):1626–1639, Aug 2016.

[WHT03] Liang Wang, Weiming Hu, and Tieniu Tan. Recent developments in human motion analysis. *Pattern recognition*, 36(3):585–601, 2003.

[WKSL13] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 103(1):60–79, 2013.

[WLC+12] Jiang Wang, Zicheng Liu, Jan Chorowski, Zhuoyuan Chen, and Ying Wu. Robust 3D action recognition with random occupancy patterns. In *Proc. European Conference on Computer Vision (ECCV)*, pages 872–885. 2012.

[WLG+15] Pichao Wang, Wanqing Li, Zhimin Gao, Chang Tang, Jing Zhang, and Philip O. Ogunbona. Convnets-based action recognition from depth maps through virtual cameras and pseudocoloring. In *ACM MM*, 2015.

[WLG+16] Pichao Wang, Wanqing Li, Zhimin Gao, Jing Zhang, Chang Tang, and Phlip Ogunbona. Action recognition from depth maps using deep convolutional neural networks. *THMS*, 46(4):498–509, 2016.

[WLG+17] Pichao Wang, Wanqing Li, Zhimin Gao, Yuyao Zhang, Chang Tang, and Philip Ogunbona. Scene flow to action map: A new representation for rgb-d based action recognition with convolutional neural networks. In *CVPR*, 2017.

[WLHL16] Pichao Wang, Zhaoyang Li, Yonghong Hou, and Wanqing Li. Action recognition based on joint trajectory maps using convolutional neural networks. In *ACM MM*, 2016.

[WLL+16a] P. Wang, W. Li, S. Liu, Y. Zhang, Z. Gao, and P. Ogunbona. Large-scale continuous gesture recognition using convolutional neural networks. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 13–18, 2016.

[WLL+16b] Pichao Wang, Wanqing Li, Song Liu, Zhimin Gao, Chang Tang, and Philip Ogunbona. Large-scale isolated gesture recognition using convolutional neural networks. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 7–12. IEEE, 2016.

[WLO+14] Pichao Wang, Wanqing Li, P. Ogunbona, Zhimin Gao, and Hanling Zhang. Mining mid-level features for action recognition based on effective skeleton representation. In *DICTA*, 2014.

[WLWY12] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *CVPR*, 2012.

[WLWY14] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Learning actionlet ensemble for 3D human action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(5):914–927, 2014.

[WLZ+16] Jun Wan, Stan Z Li, Yibing Zhao, Shuai Zhou, Isabelle Guyon, and Sergio Escalera. Chalearn looking at people RGB-D isolated and continuous datasets for gesture recognition. In *CVPRW*, 2016.

[WNX+14] Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu, and Song-Chun Zhu. Cross-view action modeling, learning and recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2649–2656, 2014.

[WPK+16a] D. Wu, L. Pigou, P. J. Kindermans, N. D. H. Le, L. Shao, J. Dambre, and J. M. Odobez. Deep dynamic neural networks for multimodal gesture segmentation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(8):1583–1597, Aug 2016.

[WPK+16b] Di Wu, Lionel Pigou, Pieter-Jan Kindermans, Nam Do-Hoang Le, Ling Shao, Joni Dambre, and Jean-Marc Odobez. Deep dynamic

neural networks for multimodal gesture segmentation and recognition. *IEEE transactions on pattern analysis and machine intelligence*, 38(8):1583–1597, 2016.

[WQT15] Limin Wang, Yu Qiao, and Xiaoou Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *CVPR*, pages 4305–4314, 2015.

[WRL+14] Jun Wan, Qiuqi Ruan, Wei Li, Gaoyun An, and Ruizhen Zhao. 3d smosift: three-dimensional sparse motion scale invariant feature transform for activity recognition from rgb-d videos. *Journal of Electronic Imaging*, 23(2), 2014.

[WRLD13] Jun Wan, Qiuqi Ruan, Wei Li, and Shuang Deng. One-shot learning gesture recognition from RGB-D data using bag of features. *Journal of Machine Learning Research*, 14:2549–2582, 2013.

[WS13] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *ICCV*, pages 3551–3558, 2013.

[WS14a] D. Wu and L. Shao. Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–731, 2014.

[WS14b] Di Wu and Ling Shao. Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition. In *Proc. IEEE Conference on Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[WSW+16] Yifan Wang, Jie Song, Limin Wang, Luc Van Gool, and Otmar Hilliges. Two-stream sr-cnns for action recognition in videos. BMVC, 2016.

[WTVG08] Geert Willems, Tinne Tuytelaars, and Luc Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *Proc. European Conference on Computer Vision (ECCV)*, pages 650–663. 2008.

[WUK+09] Heng Wang, Muhammad Muneeb Ullah, Alexander Klaser, Ivan Laptev, Cordelia Schmid, et al. Evaluation of local spatio-temporal features for action recognition. In *Proc. British Machine Vision Conference (BMVC)*, pages 124.1–124.11, 2009.

[WWY13] Chunyu Wang, Yizhou Wang, and Alan L Yuille. An approach to pose-based action recognition. In *CVPR*, 2013.

[WXLVG17] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *CVPR*, 2017.

[WXW+16] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: towards good practices for deep action recognition. In *European Conference on Computer Vision*, pages 20–36, 2016.

[WZ89] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

[WZLQ16] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, 2016.

[WZSS15] Chenxia Wu, Jiemi Zhang, Silvio Savarese, and Ashutosh Saxena. Watch-n-patch: Unsupervised understanding of actions and relations. In *CVPR*, 2015.

[XA13] Lu Xia and JK Aggarwal. Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2834–2841, 2013.

[XCA12] Lu Xia, Chia-Chih Chen, and JK Aggarwal. View invariant human action recognition using histograms of 3D joints. In *CVPRW*, 2012.

[XCW+15] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, pages 802–810, 2015.

[YCHX05] Xifeng Yan, Hong Cheng, Jiawei Han, and Dong Xin. Summarizing itemset patterns: a profile-based approach. In *Proc. ACM SIGKDD international conference on Knowledge discovery in data mining (KDD)*, pages 314–323, 2005.

[YCSC14] Xing Yan, Hong Chang, Shiguang Shan, and Xilin Chen. Modeling video dynamics with deep dynencoder. In *European Conference on Computer Vision*, pages 215–230. Springer, 2014.

[YDT+16] Yanhua Yang, Cheng Deng, Dapeng Tao, Shaoting Zhang, Wei Liu, and Xinbo Gao. Latent max-margin multitask learning with skelets for 3-D action recognition. *IEEE transactions on cybernetics*, 2016.

[YHC+12] Kiwon Yun, Jean Honorio, Debaleena Chattopadhyay, Tamara L Berg, and Dimitris Samaras. Two-person interaction detection using body-pose features and multiple instance learning. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 28–35. IEEE, 2012.

[YLS16] Mengyang Yu, Li Liu, and Ling Shao. Structure-preserving binary representations for rgb-d action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 38(8):1651–1664, 2016.

[YOI92] Junji Yamato, Jun Ohya, and Kenichiro Ishii. Recognizing human action in time-sequential images using hidden markov model. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 379–385, 1992.

[YRZ+17] Jiaolong Yang, Peiran Ren, Dongqing Zhang, Dong Chen, Fang Wen, Hongdong Li, and Gang Hua. Neural aggregation network for video face recognition. In *Proceedings of the 32th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[YT12] Xiaodong Yang and YingLi Tian. Eigenjoints-based action recognition using Naive-Bayes-Nearest-Neighbor. In *CVPRW*, 2012.

[YT14] Xiaodong Yang and YingLi Tian. Super normal vector for activity recognition using depth sequences. In *CVPR*, 2014.

[YYHD14] Shuang Yang, Chunfeng Yuan, Weiming Hu, and Xinmiao Ding. A hierarchical model based on latent dirichlet allocation for action recognition. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 2613–2618. IEEE, 2014.

[YZT12] Xiaodong Yang, Chenyang Zhang, and YingLi Tian. Recognizing actions using depth motion maps-based histograms of oriented gradients. In *ACM MM*, 2012.

[Zat98] Vladimir M Zatsiorsky. *Kinematics of human motion*. Human Kinetics, 1998.

[ZCG13] Yu Zhu, Wenbin Chen, and Guodong Guo. Fusing spatiotemporal features and joints for 3d action recognition. In *CVPRW*, 2013.

[ZF14] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Proc. European Conference on Computer Vision (ECCV)*, pages 818–833. 2014.

[ZHS+16] Wangjiang Zhu, Jie Hu, Gang Sun, Xudong Cao, and Yu Qiao. A key volume mining deep framework for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1991–1999, 2016.

[ZLO+16a] Jing Zhang, Wanqing Li, Philip O Ogunbona, Pichao Wang, and Chang Tang. RGB-D-based action recognition datasets: A survey. *Pattern Recognition*, 60:86–105, 2016.

[ZLO16b] L. Zhou, W. Li, and P. Ogunbona. Learning a pose lexicon for semantic action recognition. In *2016 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2016.

[ZLO17] Jing Zhang, Wanqing Li, and Philip Ogunbona. Joint geometrical and statistical alignment for visual domain adaptation. In *CVPR*, 2017.

[ZLS13] Mihai Zanfir, Marius Leordeanu, and Cristian Sminchisescu. The moving pose: An efficient 3D kinematics descriptor for low-latency action recognition and detection. In *ICCV*, 2013.

[ZLX+16] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie. Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks. In *AAAI*, 2016.

[ZLX17] Songyang Zhang, Xiaoming Liu, and Jun Xiao. On geometric features for skeleton-based action recognition using multilayer lstm networks. In *WACV*, 2017.

[ZLZ+14] Lijuan Zhou, Wanqing Li, Yuyao Zhang, Philip Ogunbona, Duc Thanh Nguyen, and Hanling Zhang. Discriminative key pose extraction using extended lc-ksvd for action recognition. In *DICTA*. IEEE, 2014.

[ZNH+15] Yang Zhou, Bingbing Ni, Richang Hong, Meng Wang, and Qi Tian. Interaction part mining: A mid-level approach for fine-grained action recognition. In *CVPR*, pages 3323–3331, 2015.

[ZSXF16] Fan Zhu, Ling Shao, Jin Xie, and Yi Fang. From handcrafted to learned representations for human action recognition: a survey. *Image and Vision Computing*, 55:42–52, 2016.

[ZWW+16] Bowen Zhang, Limin Wang, Zhe Wang, Yu Qiao, and Hanli Wang. Real-time action recognition with enhanced motion vector cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2718–2726, 2016.

[ZY11] Yu Zhang and Dit Yan Yeung. Multi-task learning in heterogeneous feature spaces. In *AAAI*, 2011.

[ZZM+16a] Guangming Zhu, Liang Zhang, Lin Mei, Jie Shao, Juan Song, and Peiyi Shen. Large-scale isolated gesture recognition using pyramidal 3d convolutional networks. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 19–24. IEEE, 2016.

[ZZM+16b] Guangming Zhu, Liang Zhang, Lin Mei, Jie Shao, Juan Song, and Peiyi Shen. Large-scale isolated gesture recognition using pyramidal 3d convolutional networks. In *Proceedings of ICPRW*, 2016.

[ZZSS17] Guangming Zhu, Liang Zhang, Peiyi Shen, and Juan Song. Multimodal gesture recognition using 3d convolution and convolutional lstm. *IEEE Access*, 2017.