Human Gesture Recognition with Depth Camera

Bin Liang

Charles Sturt University

This dissertation is submitted for the degree of

Doctor of Philosophy

September 2016
This thesis is dedicated to my parents and wife.
Acknowledgements

I am grateful to many people who made this work possible. First, I would like to thank my supervisor, Dr. Lihong Zheng. Her patient guidance through my studies, strong passion for scientific research, broad knowledge in different areas, and rigorous attitudes to research, always enlightened my path and supported me to solve bottlenecks. I am also indebted that she gave me all the freedom and resources to develop my skills as a researcher in a supportive team environment. My grateful thanks are also extended to my research co-supervisors, Manoranjan Paul and Xiaodi Huang. I have enjoyed our many conversations and discussions, and these have had a tremendous impact on my research. In addition, I would like to thank my colleagues for their support and help during my studies. They offered me many helping hands and shared valuable experience with me. I acknowledge and appreciate the help of Michelle Hey for proofreading my thesis. Her dedication far surpasses my expectation and makes the revision process fun and enjoyable. Last but not least, I would like to express my greatest gratitude to my parents and my wife for their continuous support for my academic pursuit. I feel very lucky and privileged to have them.
Abstract

Human gesture recognition, as one of the most active and promising research areas in computer vision, has attracted a significant amount of attention over the last few years. It has a wide range of applications including human-computer interaction, motion-sensing gaming, video surveillance, rehabilitation, and smart homes. Recently, the introduction of cost-effective depth cameras provides a new way to research motion analysis and gesture recognition. However, it also brings new challenges: how to generate effective gesture representations to characterize the spatio-temporal structures properly using depth cameras, and how to extract and represent meaningful features from the discriminative gesture representations. This thesis aims to solve these challenges with effective and efficient solutions, and evaluate the proposed approaches on four challenging gesture recognition benchmarks. Extensive experiments demonstrate the superior performances of the proposed approaches.

The thesis consists of three parts, in which the proposed approaches are self-contained but highly correlative. In the first part of the thesis, three novel frameworks are introduced to aid the recognition of human gestures using depth maps. Firstly, an effective approach based on DMHT-PHOG is presented to recognize human gestures in depth videos. For gesture representation, a depth motion history template (DMHT) is proposed to encode the temporal motion along with structural information in a compact and discriminative way. Pyramid histogram of oriented gradients (PHOG) is calculated with different levels of details according to the selected pyramid levels. Secondly, a framework based on spatio-temporal pyramid matching (STPCM) is put forward for gesture recognition using the discriminative motion information from both spatial and temporal aspects. In order to retain the inherent 3D spatial information, a novel cuboid fusion scheme is developed by grouping spatially dependent grids...
from projected planes of pyramid DMHT to construct spatio-temporal pyramid cuboids. Thirdly, to overcome the difficulties in correlation discovery between multiple views from depth maps, a novel method, specificity and latent correlation learning (SLCL), is proposed to learn the view-specific dictionaries (specificity) and the latent information between multiple views (latent correlation) for multi-view gesture recognition. The combination of the specificity and the latent correlation can consistently represent the gesture from multiple views for classification.

In the second part, in addition to depth maps, skeletal joints are adopted to learn the part-based skeleton representation for gesture recognition. A human body is represented as a set of body parts, each of which consists of multiple skeletal joints. Part-based skeleton features of each body part are proposed to deal with four types of variations, i.e., viewpoint, anthropometry, execution rate, and personal style. Given the part-based features, a dictionary learning approach is proposed to learn sub-dictionaries for each body part and correlation between them.

In the last part of this thesis, multi-modal fusion schemes are illustrated for gesture recognition. Given that different modalities have their own relative strength, their fusion yields a multi-modal semantic representation and improves the performance of gesture recognition. Various multi-modal fusion schemes are investigated at representation level and classifier level. In order to reach a stable and robust performance, a weight-learning classifier-level fusion method is proposed.
Publications


# Table of contents

List of figures ................................................. xvii

List of tables .................................................. xxi

Nomenclature ................................................... xxiii

1 Introduction .................................................. 1
   1.1 Background .............................................. 1
   1.2 Motivations .............................................. 8
       1.2.1 Gesture Representation ................................. 8
       1.2.2 Feature Extraction and Representation ..................... 9
       1.2.3 Multi-View Gesture Recognition ......................... 10
       1.2.4 Multi-Modal Fusion .................................. 11
   1.3 Thesis Contributions .................................... 12
   1.4 Thesis Structure ....................................... 15
   1.5 Notation ............................................... 15

2 Literature Review ........................................... 17
   2.1 Data Modalities ....................................... 17
       2.1.1 Depth Maps ......................................... 18
       2.1.2 Skeletal Joints ..................................... 19
   2.2 Gesture Representation and Feature Extraction ................. 21
       2.2.1 Depth Map-Based Features ............................ 21
       2.2.2 Skeleton-Based Features ............................. 24
   2.3 Feature Representation .................................. 26
## Table of contents

2.3.1 Bag of Visual Words ............................................. 26  
2.3.2 Sparse Coding .................................................... 28  
2.4 Classification ....................................................... 29  
2.5 Multi-Modal Fusion .................................................. 30  
2.6 Datasets ............................................................... 32  
2.6.1 MSR Action3D Dataset ......................................... 33  
2.6.2 MSR Gesture3D Dataset ........................................ 34  
2.6.3 MSR Action Pairs Dataset ...................................... 35  
2.6.4 ChaLearn Multi-Modal Dataset ................................. 36  

### 3 Depth Motion History Template Based Pyramid Histograms of Oriented Gradient for Gesture Recognition  

3.1 Introduction ......................................................... 39  
3.2 Depth Motion History Template .................................. 41  
3.3 The DMHT-PHOG Descriptor ....................................... 44  
3.4 Classification ....................................................... 47  
3.5 Experiments .......................................................... 50  
3.5.1 Dependence on $\xi$ and $\sigma$ ................................. 50  
3.5.2 Experimental Setup .............................................. 53  
3.5.3 Evaluation of DMHT-PHOG ...................................... 53  
3.6 Summary ............................................................ 56  

### 4 Spatio-Temporal Pyramid Cuboid Matching for Gesture Recognition Using Depth Maps  

4.1 Introduction .......................................................... 59  
4.2 Hierarchical DMHT .................................................. 60  
4.3 Depth Feature Extraction and Sparse Coding .................... 63  
4.4 Spatio-Temporal Pyramid Cuboid Matching ....................... 65  
4.4.1 Pyramid DMHT .................................................... 66
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4.2</td>
<td>Cuboid Fusion</td>
<td>68</td>
</tr>
<tr>
<td>4.5</td>
<td>Experiments</td>
<td>71</td>
</tr>
<tr>
<td>4.5.1</td>
<td>MSR Action3D Dataset</td>
<td>72</td>
</tr>
<tr>
<td>4.5.2</td>
<td>MSR Gesture3D Dataset</td>
<td>73</td>
</tr>
<tr>
<td>4.5.3</td>
<td>MSR Action Pairs Dataset</td>
<td>75</td>
</tr>
<tr>
<td>4.5.4</td>
<td>ChaLearn Multi-Modal Dataset</td>
<td>76</td>
</tr>
<tr>
<td>4.5.5</td>
<td>Evaluation of Parameters</td>
<td>80</td>
</tr>
<tr>
<td>4.6</td>
<td>Summary</td>
<td>81</td>
</tr>
<tr>
<td>5</td>
<td>Specificity and Latent Correlation Learning for Multi-View Gesture</td>
<td>83</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>83</td>
</tr>
<tr>
<td>5.2</td>
<td>Multiple Views from Depth Maps</td>
<td>86</td>
</tr>
<tr>
<td>5.3</td>
<td>Specificity and Latent Correlation Learning</td>
<td>89</td>
</tr>
<tr>
<td>5.4</td>
<td>Optimization of the Model</td>
<td>92</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Optimization Step — Coefficient Matrix $A_v$</td>
<td>92</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Optimization Step — Overall Dictionary $D$</td>
<td>94</td>
</tr>
<tr>
<td>5.4.3</td>
<td>Initialization of the Sub-Dictionaries</td>
<td>98</td>
</tr>
<tr>
<td>5.5</td>
<td>Spatial-Temporal Pyramid Matching</td>
<td>98</td>
</tr>
<tr>
<td>5.6</td>
<td>Experimental Validation</td>
<td>100</td>
</tr>
<tr>
<td>5.6.1</td>
<td>Parameter Setting</td>
<td>100</td>
</tr>
<tr>
<td>5.6.2</td>
<td>MSR Action3D Dataset</td>
<td>101</td>
</tr>
<tr>
<td>5.6.3</td>
<td>MSR Gesture3D Dataset</td>
<td>102</td>
</tr>
<tr>
<td>5.6.4</td>
<td>MSR Action Pairs Dataset</td>
<td>103</td>
</tr>
<tr>
<td>5.6.5</td>
<td>ChaLearn Multi-Modal Dataset</td>
<td>105</td>
</tr>
<tr>
<td>5.7</td>
<td>Summary</td>
<td>106</td>
</tr>
<tr>
<td>6</td>
<td>Part-based Skeleton Representation Learning for Gesture Recognition</td>
<td>109</td>
</tr>
<tr>
<td>6.1</td>
<td>Introduction</td>
<td>109</td>
</tr>
</tbody>
</table>
# Table of contents

## 6.2 Part-based Representation .......................... 112
## 6.3 Part-based Skeleton Features ...................... 113
## 6.4 Part-based Representation Learning ............... 116
## 6.5 Temporal Pyramid Matching ........................ 119
## 6.6 Experiments ........................................ 120
  - 6.6.1 Parameter Setting ................................ 120
  - 6.6.2 MSR Action3D Dataset .......................... 120
  - 6.6.3 MSR Action Pairs Dataset ....................... 122
  - 6.6.4 ChaLearn Multi-Modal Dataset .................. 125
## 6.7 Summary ............................................ 125

## 7 Multi-Modal Gesture Recognition Using Depth Maps and Skeletal Joints 127
## 7.1 Introduction ....................................... 127
## 7.2 Multi-Modal Fusion ................................ 128
  - 7.2.1 Representation-Level Fusion ..................... 128
  - 7.2.2 Classifier-Level Fusion ......................... 130
## 7.3 Experiments ....................................... 133
  - 7.3.1 MSR Action3D Dataset ......................... 134
  - 7.3.2 MSR Action Pairs Dataset .................... 135
  - 7.3.3 ChaLearn Multi-Modal Dataset ................ 137
  - 7.3.4 Evaluation of Multi-Modal Fusion Schemes ...... 138
## 7.4 Summary ............................................ 139

## 8 Conclusions and Future Work .......................... 141
## 8.1 Conclusions ....................................... 141
## 8.2 Future Work ....................................... 144

## References ............................................ 147
Table of contents

Appendix A  Derivations 163
   A.1  Proof of Proposition 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 163
   A.2  Deviation of $g(d'_o)$ in Eq. (5.18) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 163

Appendix B  Review of F-measure 167
Certificate of Authorship

I hereby declare that this submission is my own work and to the best of my knowledge and belief, understand that it contains no material previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any other degree or diploma at Charles Sturt University or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by colleagues with whom I have worked at Charles Sturt University or elsewhere during my candidature is fully acknowledged.

I agree that this thesis be accessible for the purpose of study and research in accordance with normal conditions established by the Executive Director, Library Services, Charles Sturt University or nominee, for the care, loan and reproduction of thesis, subject to confidentiality provisions as approved by the University.

Name Bin Liang  
Signature  Bin Liang  
Date  02/05/2016
List of figures

1.1 Task of human gesture recognition. ......................... 2
1.2 Sample frames for gestures and activities .................. 3
1.3 Depth sensor examples ....................................... 4
1.4 Depth maps and skeletal joints of a gesture from Kinect. .... 6
1.5 Steps of gesture recognition. ............................... 7

2.1 Comparison between RGB image and depth image. ............ 19
2.2 Skeleton data depiction from the first-generation Kinect and second-generation Kinect. ......................... 20
2.3 Sample frames from the MSR Action3D dataset. .............. 34
2.4 Sample frames from the MSR Gesture3D dataset. ............ 35
2.5 Sample frames from the MSR Action Pairs dataset. ........... 36
2.6 Sample frames from ChaLearn Multi-Modal dataset. ........... 37

3.1 The framework based on DMHT-PHOG descriptor. .......... 40
3.2 Illustration of 3D point clouds calculation. .................. 42
3.3 3D point clouds projections. ............................... 43
3.4 An example of DMHT. ......................................... 45
3.5 An example of HOG feature extraction. ...................... 45
3.6 Examples of PHOG descriptor at different spatial levels. .... 48
3.7 Selection of threshold value ($\xi$) for DMHT. ................. 51
3.8 Selection of decay parameter ($\sigma$) for DMHT. .............. 52
3.9 Confusion matrices for CST of the MSR Action3D dataset using the DMHT-PHOG. ............................... 57
| 4.1 | Illustration of the spatio-temporal pyramid cuboid matching framework. | 61 |
| 4.2 | Illustration of hierarchical DMHT. | 62 |
| 4.3 | Front views of hierarchical DMHT. | 63 |
| 4.4 | Illustration of pyramid DMHT. | 67 |
| 4.5 | Illustration of spatial pyramid cuboids. | 69 |
| 4.6 | Comparison between normal fusion and cuboid fusion at the spatial level $l = 1$. | 69 |
| 4.7 | Confusion matrix for MSR Action3D dataset using STPCM. | 74 |
| 4.8 | Confusion matrix for MSR Gesture3D dataset using STPCM. | 75 |
| 4.9 | Confusion matrix for MSR Action Pairs dataset using STPCM. | 77 |
| 4.10 | An example from the ChaLearn multi-modal dataset. | 77 |
| 4.11 | Gesture regions segmentation. | 78 |
| 4.12 | Confusion matrix for ChaLearn multi-modal dataset using STPCM. | 79 |
| 4.13 | Evaluation of temporal stride parameter $w$ in hierarchical DMHT. | 80 |
| 4.14 | Evaluation of the temporal scale $S$ and spatial level $L$. | 81 |
| 5.1 | Illustration of 3D point clouds rotation. | 87 |
| 5.2 | Multiple views from a depth map. | 88 |
| 5.3 | Illustration of the spatio-temporal pyramid cuboid matching framework. | 99 |
| 5.4 | Recognition accuracies with different dictionary sizes. | 102 |
| 5.5 | Confusion matrix for MSR Action3D dataset using SLCL+STPM. | 103 |
| 5.6 | Confusion matrix for MSR Gesture3D dataset using SLCL+STPM. | 104 |
| 5.7 | Confusion matrix for MSR Action Pairs dataset using SLCL+STPM. | 106 |
| 5.8 | Confusion matrix for ChaLearn multi-modal dataset using SLCL+STPM. | 107 |
| 6.1 | Depth frames and skeletons from a sample gesture. | 110 |
| 6.2 | Illustration of human body parts | 113 |
| 6.3 | Example of pair-wise joint distance. | 115 |
| 6.4 | Illustration of skeleton-based temporal pyramid matching. | 121 |
6.5 Confusion matrix for MSR Action3D dataset using the proposed skeleton method. ........................................ 123
6.6 Confusion matrix for MSR Action Pairs dataset using the proposed skeleton method. ........................................ 124
6.7 Confusion matrix for ChaLearn multi-modal dataset using the proposed skeleton method. ................................. 126
7.1 General framework of the proposed approach. ....................... 129
7.2 Representation-level fusion. ........................................ 130
7.3 Classifier-level fusion. ............................................ 130
7.4 $F_1$ score for each class on MSR Action3D dataset. ............... 135
7.5 $F_1$ score for each class on MSR Action Pairs dataset. ............. 136
7.6 $F_1$ score for each class on ChaLearn multi-modal dataset. ........ 138
7.7 Evaluation of various multi-modal fusion schemes. ................. 139
List of tables

2.1 Human gesture datasets recorded by depth sensors. .................................. 33

3.1 Three action subsets of MSR Action3D dataset. ................................. 53

3.2 Recognition accuracy (%) comparison of the DMHT-PHOG with other methods on three subsets of MSR Action3D dataset. ............... 54

3.3 Recognition accuracy (%) comparison of the DMHT-PHOG with other methods on the cross subject test of MSR Action3D dataset. .... 55

4.1 Recognition accuracy (%) comparison of STPCM with other methods on the MSR Action3D dataset. .................................................. 72

4.2 Recognition accuracy (%) comparison of STPCM with other methods on the MSR Gesture3D dataset. ............................................. 74

4.3 Recognition accuracy (%) comparison of STPCM with other methods on the MSR Action Pairs dataset. ........................................... 76

4.4 Recognition accuracy (%) comparison of STPCM with other methods on the ChaLearn multi-modal dataset. ................................. 79

5.1 Recognition accuracy (%) comparison of SLCL+STPM with other methods on the MSR Action3D dataset. ...................................... 101

5.2 Recognition accuracy (%) comparison of SLCL+STPM with other methods on the MSR Gesture3D dataset. ................................. 104

5.3 Recognition accuracy (%) comparison of SLCL+STPM with other methods on the MSR Action Pairs dataset. ............................... 105

5.4 Recognition accuracy (%) comparison of SLCL+STPM with other methods on the ChaLearn multi-modal dataset. .......................... 106
6.1 Human body parts and their corresponding skeletal joints. . . . . . 112
6.2 Recognition accuracy(%) comparison of the proposed skeleton method with other methods on the MSR Action3D dataset. . . . . . 122
6.3 Recognition accuracy(%) comparison of the proposed skeleton method with other methods on the MSR Action Pairs dataset. . . . . 124
6.4 Recognition accuracy(%) comparison of the proposed skeleton method with other methods on the Chalearn Multi-Modal dataset. . 125
7.1 Recognition accuracy(%) comparison of the proposed multi-modal method with other methods on the MSR Action3D dataset. . . . . 134
7.2 Recognition accuracy(%) comparison of the proposed multi-modal method with other methods on the MSR Action Pairs Dataset. . . . 136
7.3 Recognition accuracy(%) comparison of the proposed multi-modal method with other methods on the Chalearn multi-modal dataset. . . 137
Nomenclature

**Acronyms / Abbreviations**

2DMTM 2D motion trail model
3DMTM 3D motion trail model
3D three-dimensional space
AI artificial intelligence
BoVW bag of visual words
DL-GSGC group sparsity and geometry constrained dictionary learning
dictionary learning
DMHT depth motion history template
DMM depth motion maps
HCI human-computer interaction
HOF histograms of optical flow
HOJ3D histogram of 3D joint locations
HOG histograms of oriented gradients
HON4D histogram of oriented 4D normals
LDA linear discriminant analysis
LOP local occupancy patterns
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MHI</td>
<td>motion history image</td>
</tr>
<tr>
<td>MKL</td>
<td>multiple kernel learning</td>
</tr>
<tr>
<td>MP</td>
<td>moving pose</td>
</tr>
<tr>
<td>PCA</td>
<td>principal component analysis</td>
</tr>
<tr>
<td>PHOG</td>
<td>pyramid histograms of oriented gradients</td>
</tr>
<tr>
<td>RGB</td>
<td>red-green-blue</td>
</tr>
<tr>
<td>ROP</td>
<td>random occupancy pattern</td>
</tr>
<tr>
<td>ScTPM</td>
<td>sparse coding-based temporal pyramid matching</td>
</tr>
<tr>
<td>SDK</td>
<td>software development kit</td>
</tr>
<tr>
<td>SIFT</td>
<td>scale-invariant feature transform</td>
</tr>
<tr>
<td>SLCL</td>
<td>specificity and latent correlation learning</td>
</tr>
<tr>
<td>SNV</td>
<td>super normal vector</td>
</tr>
<tr>
<td>SRC</td>
<td>sparse representation-based classification</td>
</tr>
<tr>
<td>SSS</td>
<td>structured streaming skeletons</td>
</tr>
<tr>
<td>STIP</td>
<td>space-time interest points</td>
</tr>
<tr>
<td>STOP</td>
<td>space-time occupancy patterns</td>
</tr>
<tr>
<td>STPCM</td>
<td>spatio-temporal pyramid cuboid matching</td>
</tr>
<tr>
<td>SVM</td>
<td>support vector machine</td>
</tr>
<tr>
<td>ToF</td>
<td>time-of-flight</td>
</tr>
<tr>
<td>TPM</td>
<td>temporal pyramid matching</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Background

Human gestures are expressive and meaningful body movements. People perform various gestures in daily life as a means of communication and interaction. Specifically, gestures involve physical movements of the fingers, hands, arms, head or body with the intention of conveying meaningful information or interacting with surroundings. Humans have the remarkable ability to perceive gestures from visual information, analyze human-object interactions, and even infer their intentions through gesture motions.

Gesture recognition is the process by which gestures performed by humans are made known to a computer device. In the past two decades, human gesture recognition has attracted considerable interest and gained increasing popularity in the area of computer vision and pattern recognition [97, 1, 2, 98, 116]. The task of gesture recognition is to analyze human motions, and to perceive various kinds of human gestures. As shown in Fig. 1.1, the input is a gesture sequence from a capture device, and the output is the predicted class label within a gesture dataset. This process is automatically handled by the designed algorithm. There has been a wide range of practical applications including human-computer interaction (HCI), motion-sensing gaming, video surveillance, rehabilitation and smart homes. For example, a prototype system called WiSee [117] can leverage wireless signals to enable users to control electronic devices at home with gestures. GestureCam [123] is designed to recognize simple predefined head and hand gestures so that an alarm
is raised when the given motion satisfies certain criteria. With the popularity of gesture-controlled devices like the Nintendo Wii [153], Microsoft Kinect [96], and Playstation Move [152], it is clear that individual players can interact and be more physically involved with games than ever before. Due to many possibilities in a large number of applications which benefit from human gesture recognition, this topic has gained more and more attention in the research community.

The recognition of human movements can be performed at various levels of abstraction [116]. Gesture, action and activity are three commonly used taxonomies. Similarities exist between them. According to the literature [136, 2], gestures are more related to elementary movements of human body parts, and are used to describe meaningful motions of human body as the atomic components. Atomic components correspond to instantaneous atomic entities upon which a gesture is formed [44]. For example, in a “leg kicking” gesture, the atomic component could be “one leg moving away from the other leg”. “Arm stretching” is another example of gesture. Actions emphasize the movement of the whole body that could be composed of multiple gestures with a temporal ordering, such as “waving”, “walking”, or “running”. Sometimes, actions could involve simple human-object interactions, such as “wearing a hat” or “pushing a chair”. Activities typically highlight more a series of actions which are usually complex, or refer to the interactions between two or more persons. Activities are usually captured in unconstrained scenes with complex backgrounds.
“Two persons hugging” and “having a meal” are examples of activities. Nevertheless, there are no clear boundaries between *gesture* and *action*. “Gesture recognition” and “action recognition” are always interchangeably used in computer vision and other fields of artificial intelligence (AI) [24, 156, 127]. In this thesis, the subtle differences between them are ignored, and “activity recognition” which could introduce contexts such as environment (e.g. [167]) and interactions between persons (e.g. [172]) is not considered. Fig. 1.2 shows the sample frames of gestures and activities.

![Sample frames for gestures and activities](image_url)

**Fig. 1.2** Sample frames for gestures and activities: (a) three gestures from Chalearn Multi-Modal Dataset [33], (b) three activities from Cornell Activity Dataset [65].

The initial step in recognizing human gestures is having sensors to capture body movements. Each sensing technology varies in several dimensions, including accuracy, resolution, latency, range of motion, user comfort, and cost [97]. Early attempts to tackle this issue resulted in contact devices which record spatial positions directly, such as Wii [153], CyberGlove [62] and Powergloves [59]. Contact devices or electromagnetic markers can provide accurate measurements of human gesture movement. However, they typically require the user to wear a cumbersome device and carry a load of cables connecting the device to a computer. This hinders the ease and naturalness of the user’s interaction with the computer. Furthermore, extensive
calibration is required when using these devices [83]. Many applications, especially in surveillance and HCI, would benefit from a solution that is markerless. A few years ago, research into human gesture recognition concentrated on learning and recognizing human gestures from videos captured by standard RGB cameras. To this end, vision-based methods are proposed to avoid these constraints by using cameras as sensors. In addition, vision-based methods have to overcome several challenges such as illumination changes, low contrast and noise problems. Gestures might not be captured precisely due to the aforementioned challenges. Furthermore, human gestures are performed in three-dimensional space (3D), and capturing 3D motion structure from RGB cameras is quite a challenging task. Such difficulty severely limits the performance of human gesture recognition from RGB cameras.

![Depth sensor examples](a) Stereo camera\(^1\) (b) ToF camera\(^2\) (c) Structured light camera\(^3\)

Fig. 1.3 Depth sensor examples: (a) stereo camera system comprised of two SLR cameras (b) SwissRanger SR4000 ToF camera from Mesa Imaging (c) Microsoft Kinect

With recent advances in depth sensing techniques, several types of depth sensors are currently available. These sensors utilize different methods to acquire a matrix of measurements from a scene, where each element denotes the distance of the corresponding point from the sensor. Available depth sensors can generally be classified into three main categories based on [22]: **stereo camera**, **time-of-flight (ToF) camera**, and **structured light camera**. The three types of depth sensors are shown in Fig. 1.3. A **stereo camera** system based on stereo triangulation infers the

\(^1\)Image from https://www.euro-fusion.org/2010/05/week-30-in-vessel-survey-in-3d/
\(^2\)Image from http://hptg.com/industrial/
\(^3\)Image from http://www.xbox.com/en-GB/xbox-one/accessories/kinect-for-xbox-one
3D structure of a scene from two or more images from various viewpoints [135]. It is inspired by human vision. Due to the complexity of geometry calculation and sensitivity to light changes, this kind of depth sensor is difficult to set up for real-time applications. Also, stereo intensity images are sensitive to illumination changes, which increases difficulties with correspondence matching for triangulation [51]. These issues also render the depth map reconstruction from a stereo camera impractical for real-time and real-world applications. A ToF camera estimates the distance to an object surface using active light pulses from a single camera. ToF cameras are able to achieve high frame rates, which makes them suitable for real-time applications. However, the relatively high price of ToF cameras limits practical usage. The major technical drawback is low resolution. By illuminating the scene with a specially designed light pattern, structured light, depth can be estimated using only a single image of the reflected light. Microsoft released the first-generation Kinect in 2010, and the second-generation Kinect in 2014. The first-generation Kinect relies on projecting IR patterns on the scene using the IR pattern projector, and estimates the depth map by measuring the distortion of the projected IR pattern using an IR camera and structured light algorithms [53]. On the other hand, the second-generation Kinect uses a ToF camera for its range imaging [174], enabling natural user interfaces and gaming applications using computer vision and gesture recognition techniques. The advantage of structured light cameras over ToF cameras is that they are much less expensive. A major issue with structured light cameras is that depth maps have noise, as some areas cannot be seen by both the projector and the camera. The introduction of the cost-effective depth sensor, Kinect, makes depth capture possible, which enables the estimation of 3D joint positions of the human skeleton in real time [124]. This has facilitated a number of visual recognition tasks including gesture recognition [145, 54]. In particular, a growing amount of research has focused on recognizing human gestures using Kinect — rather than other depth sensors — due to its immense popularity [22, 79, 169, 144].
As previously discussed, Kinect can provide both depth maps and skeletal joint positions in real time. A depth map is an image or image channel that contains distance information from a particular point of view. Compared with ordinary video sequences captured by RGB cameras, depth maps have several advantages in the context of gesture recognition. Depth maps can provide 2.5D structural information so that the motion information of actions can be more discriminative. Moreover, depth maps are insensitive to illumination changes. The huge color and texture variability induced by clothing, hair, skin and background could be reduced. In addition to depth maps, gestures can be also represented with a sequence of skeletal joints. Given a depth map, skeletal joints representing 3D positions of body joints can be tracked by the skeleton tracker [124]; it is intuitive to recognize human gestures using skeleton information. In this thesis, different characteristics of both depth maps and skeletal joints are explored for the task of human gesture recognition, and efficient gesture representation, feature extraction, and feature representation algorithms are presented. Fig. 1.4 shows the sample frames of a gesture captured by Kinect. Note that depth maps in this thesis are color-rendered for better visualization.

![Image of depth maps and skeletal joints](image)

(a) Depth maps  
(b) Skeletal joints

Fig. 1.4 Depth maps and skeletal joints of a gesture from Kinect.

This thesis focuses on recognizing gestures using depth maps and skeletal joints provided by the depth camera. In particular, the spatio-temporal information for
human gesture recognition is explored, and novel algorithms for different tasks are proposed. As shown in Fig. 1.5, this thesis follows four important and necessary steps to generate the recognition results:

- **Gesture representation**: gestures are collected as video sequences, each of which consists of a set of frames encoding depth or skeleton information. If all the motion information in the gesture sequence can be represented exactly, the features extracted from the representation in a given gesture would be discriminative enough for recognition. Gesture representation is the process that depicts how the subject moves over the course of the gesture.

- **Feature extraction**: gesture sequences encode massive spatio-temporal data that contribute nothing to the gesture itself, such as information related to the cluttered background. Feature extraction is the process that detects and extracts the most discriminative information from the gesture representation as features.

- **Feature representation**: any gesture sequence will generate a specific number of features, and different gesture sequences will have a different number of features. Feature representation is used to derive a unique representation for every video sequence based on the extracted features. The final feature representation should be of the same dimension within different gesture sequences.

- **Classification**: based on the generated feature representation, classification is the final step of assigning class labels to unknown gesture sequences according to the model, the parameters of which are learned from training samples.
1.2 Motivations

The solution to automatically recognizing gestures from depth sensors is more difficult than the question itself. Over the past few years, there have been many remarkable research achievements, as previously discussed. However, it is still a very challenging task to accurately recognize human gestures, due to the fact that gestures could be affected by visual appearances, styles, and performance speeds. This subsection introduces the motivations and challenges that are investigated in this thesis.

1.2.1 Gesture Representation

For human gesture recognition, the common approach is to extract motion features from the representation of gesture sequences and then to predict corresponding gesture class labels. However, different people perform the same gesture differently, and even the same person could perform a given gesture differently on different occasions. According to [140, 176], the performance of gesture recognition could be affected by four main sources of variations: viewpoint, anthropometry, execution rate, and personal style. Viewpoint illustrates the relationship between people and the camera. The same gesture, observed from various viewpoints, could lead to very different results. Anthropometry encodes the physical attributes of human body sizes, and it is movement invariant. Execution rate is related to the speed of gesture performance or the frame rate of the captured video sequence. The execution rate at which the gesture is performed has an important effect on the temporal information of a gesture, especially when temporal features are used. Personal style is dependent on each individual person, since different people could choose their own way to perform the same gesture. Intuitively, deriving an effective gesture representation from video sequences is a vital step in successful gesture recognition.

Many gesture representations and models have been proposed to address the questions of human gesture representation and motion analysis [97, 3, 116]. However,
it remains a challenge to generate a more complete representation of the human gesture in order to characterize the spatio-temporal structures properly. Gesture representation depends on the technology used in its capture. It may be a sequence of video frames from visual gesture acquisition, a stream of data points from a tracking device, or a set of measurements for a data glove, and many others [139]. Recent depth sensors are able to capture depth information and even provide skeleton information, which is different to early sensors. The question of how gestures should be represented using depth and skeleton information is still an ongoing problem. For example, the direct usage of depth maps and 3D joint positions extracted by the skeleton tracker [124] is not optimal due to possible failure caused by noise or occlusions in depth maps. An effective gesture representation should be insensitive to temporal sequence misalignment, robust to noise, and discriminative for gesture recognition.

1.2.2 Feature Extraction and Representation

Once the spatio-temporal structures of human gestures have effectively been represented, the next step is feature extraction and representation. The depth sensor provides depth maps and skeleton information. Conventional feature extraction approaches based on RGB video sequences do not perform well using depth maps, as depth maps and RGB video sequences have quite different properties [111]. Features detected from gesture representations should be distinctive among different gestures, and be similar to gestures from the same class. Depth and skeleton information provided by depth sensors is very useful for gesture recognition, but determining how to extract meaningful features from the discriminative gesture representation is not a trivial task.

Another major challenge is establishing how the extracted features for any gesture sequence should be represented. The number of extracted features can be large, and
different for given gesture sequences. Moreover, consideration should be made of how to incorporate spatial and temporal information during the process of feature representation for best results. The bag of visual words (BoVW) framework with local features and its variants [157, 61, 103, 113] have dominated the research work of gesture recognition at this step and showed their effectiveness. One of the notorious disadvantages of the BoVW framework is that it ignores the spatial and temporal relationships within the extracted features, which is a very important aspect of the feature representation of a gesture sequence.

1.2.3 Multi-View Gesture Recognition

Encouraged by the success of template matching methods on 2D gesture sequences, many approaches to gesture recognition using depth maps have been proposed to transform the problem from 3D to 2D [165, 20, 78, 56]. Typically, the depth map of each frame is projected onto predefined 2D planes, and then recognition is performed on the projected 2D planes. In this way, a multi-view gesture recognition task is generalized, where each view refers to the corresponding projected plane. Obviously, these methods are able to reduce the computational complexity, but the inherent 3D information could be lost when depth maps are projected onto multiple 2D planes, which limits the performance of gesture recognition.

Although many multi-view gesture recognition algorithms have been proposed, there are still some problems. The current methods focus on single-view learning, which lacks the ability to discover correlations between various views. Despite the discriminative feature representation owned by each specific view, the latent correlation (e.g., 3D structure) from multiple views of a gesture contributes to the representation of the gesture. If only the view-specific feature representation is used for classification, the common patterns shared by various views do not facilitate classification performance, and may even degrade the accuracy. To overcome the
difficulties, latent correlation between multiple views needs to be considered in addition to the view-specific information. Moreover, the latent correlation is always heuristically defined with prior knowledge, and seldom literatures work on adaptive latent correlation discovery by model learning.

1.2.4 Multi-Modal Fusion

Human gestures are highly related to different modalities, such as RGB data, depth maps, and skeletal joints. A single modality is usually insufficient to solve the problem of recognizing human gestures. For example, two different actions, “eating” and “drinking”, may exhibit very similar skeleton motions in the human-object interaction. Additional modalities, such as depth maps, are necessary to compensate the lost spatio-temporal context in this scenario. Multi-modal data-based methods utilize different properties of multiple modalities, thus usually improving the performance of recognition. A significant issue with these methods is how the results from multiple modalities should be combined to obtain a stable performance.

Many research works have been devoted to fusing multiple modalities for boosting performance [145, 101, 76, 158]. Typical fusion schemes include early fusion [150, 69] and late fusion [133, 104]. For early fusion, the fusion is conducted at feature level or representation level, where features or representations are firstly generated from the multiple modalities and then concatenated to train a final classifier. For late fusion, the scores from multiple classifiers related to different modalities are fused using the arithmetic mean or geometric mean. In general, these fusion schemes are developed in different scenarios and adopted for gesture recognition by different works. While in most practical applications, late fusion of the scores output by multiple classifiers offers a cheap and surprisingly effective solution [60], both early and late fusion of either intermediate or final data representations remain under
active investigation. How these fusion methods influence the final recognition is an interesting question and certainly worthy of detailed investigation.

1.3 Thesis Contributions

Based on the motivations introduced in the previous section, effective and efficient solutions are proposed and evaluated using several benchmarks. The goal of this thesis is the recognition of human gestures using depth camera. The first part of the work is based on representations and features from depth maps employed for gesture recognition. Three gesture recognition frameworks based on depth maps are proposed in this part. The second part of this work introduces a new method for gesture recognition using skeletal joints from depth camera, while in the third part, multi-modal fusion schemes are adopted to recognize human gestures using both depth maps and skeletal joints. The following briefly introduces the solutions and summarizes the main contributions of each individual work.

In the first part, an effective human gesture recognition method using DMHT-PHOG descriptor is firstly proposed. This work aims to extract pyramid histograms of oriented gradient (PHOG) descriptors from the depth motion history template (DMHT) of gestures. The proposed gesture representation, DMHT, is generalized to represent gesture motion information in 3D space by projecting depth maps onto three orthogonal planes. DMHT contains motion history information from the corresponding depth maps along the temporal dimension. Encouraged by the success of histograms of oriented gradient (HOG) in human detection [28], this work extends the HOG and proposes a spatial pyramid representation to encode the DMHT for gesture recognition. By adding pyramid representation, PHOG is able to characterize local shapes and appearances from DMHT at different spatial block sizes. Compared to the original depth data, the proposed DMHT-PHOG descriptor is
more compact and more discriminative in encoding human gestures. Experiments are then conducted to show the feasibility of the proposed approach.

Secondly, a robust framework using spatio-temporal pyramid cuboid matching (STPCM) is put forward, with the objective of recognizing human gestures based on DMHT and cuboid fusion. In this work, training depth gestures are represented using hierarchical DMHT with various temporal strides to exploit speed invariance. Three-channel codebooks are generated from all the projected planes of hierarchical DMHT. The gesture model is based on linear SVM supported by combined spatio-temporal features from the cuboid fusion scheme in order to capture the inherent spatially dependent information from the projected planes of pyramid DMHT. A novel cuboid fusion scheme is presented, which groups spatial-dependent grids from projected planes into pyramid cuboids to enhance the discriminant information. In the classification stage, the unknown gesture sequence is then partitioned into pyramidal sub-volumes and represented using pyramid DMHT which encodes the multi-scale temporal motion and shape information in 3D. Comprehensive experiments demonstrate the robustness and recognition accuracy of the proposed approach.

Thirdly, a novel method, specificity and latent correlation learning (SLCL), is introduced for multi-view gesture recognition. This work further extends the previous methods in more general cases to generate multiple views from a depth map instead of only three projected planes. In addition, this work tries to overcome the difficulties in correlation discovery between multiple views of gestures. Different to most of the current methods that explicitly utilize view-based knowledge for multi-view gesture recognition, this method focuses on the task based on a dictionary learning framework. SLCL is able to learn a view-specific sub-dictionary (specificity) for each view, and the latent dictionary (latent correlation) in multiple views. The specificity captures the most discriminative features of each view, while the latent correlation contributes the essential 3D information between all the views. With the help of
the specificity and latent correlation, the overall dictionary can be more compact and more discriminative for classification. Evaluation of the proposed method is undertaken by performing an extensive series of experiments on benchmark gesture datasets, with very promising results achieved.

In the second part, a method of part-based skeleton representation learning is introduced for gesture recognition using skeletal joints. This representation is able to represent human motion characteristics from not only the body parts but also the inherent relationships between them. This work proposes to explicitly learn the sub-dictionaries for each body part, as well as the dictionary representing the correlation between these body parts. To retain the temporal information during the feature representation, temporal pyramid matching (TPM) based on max pooling is used to yield a histogram representation for every gesture sequence. Experiments are also conducted using this method based on skeletal joints, and the results outperform the existing skeleton-based methods on public datasets.

In the third part of this thesis, a framework of multi-modal gesture recognition is proposed using depth maps and skeletal joints. A single modality is usually insufficient to solve the problem of recognizing human gestures. Rather than using only one modality, this work exploits multi-modal data, i.e., depth maps and skeletal joints, to represent and recognize human gestures as a spatio-temporal pyramid representation. Two kinds of multi-modal fusion schemes are investigated: representation-level and classifier-level fusion. In order to achieve stable performances when using multi-modal fusion schemes, a weight-learning fusion scheme at classifier level is presented, with the objective of learning weights for the fusion of different modalities. Comprehensive experiments demonstrate that this proposed method is able to achieve higher recognition accuracies than most state-of-the-art methods, while obtaining stable performances.
1.4 Thesis Structure

The remainder of this thesis is organized as follows:

Chapter 2 provides a brief overview of the topics related to this thesis. It reviews previous work and compares methods for addressing the challenges of gesture recognition using depth camera.

Chapter 3 presents an approach to gesture recognition using depth motion history template based pyramid histograms of oriented gradient (DMHT-PHOG).

Chapter 4 provides the details of a spatio-temporal pyramid cuboid matching (STPCM) framework for gesture recognition using depth maps, attempting to retain inherent spatial information when depth maps are projected onto 2D planes. This proposed framework is based on the DMHT and cuboid fusion scheme.

Chapter 5 introduces a novel method — specificity and latent correlation learning (SLCL) — to learn the specificity and latent correlation for multi-view gesture recognition.

Chapter 6 focuses on learning part-based skeleton representation for gesture recognition based on skeletal joints, following gesture recognition using depth maps.

Chapter 7 presents a multi-modal gesture-recognition framework using both depth maps and skeletal joints, which is illustrated for better performance of gesture recognition. Furthermore, several multi-modal fusion schemes are investigated in this chapter.

Chapter 8 finally summarizes conclusions and future work of the thesis.

1.5 Notation

In this thesis, vectors are denoted by bold lower-case letters and matrices by upper-case ones. For instance, for this paragraph consider a vector $\mathbf{x}$ in $\mathbb{R}^n$ and a matrix $\mathbf{X}$ in $\mathbb{R}^{m \times n}$. The columns of $\mathbf{X}$ are represented by indexed vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ such
that $X = [x_1, \ldots, x_n]$ can be written. The $i$-th entry of $x$ is denoted by $x[i]$, and the $i$-th entry of the $j$-th column of $X$ is represented by $X[i, j]$. For $q \geq 1$, the $\ell_q$-norm of $x$ is defined as $\|x\|_q = (\sum_{i=1}^{n} |x[i]|^q)^{1/q}$, and the $\ell_\infty$-norm is defined as $\|x\|_\infty = \lim_{q \to +\infty} \|x\|_q = \max_{i=1,\ldots,n} |x[i]|$. For $q < 1$, the $\ell_q$-penalty is defined as $\|x\|_q = \sum_{i=1}^{n} |x[i]|^q$, which, with an abuse of terminology, is often referred to as $\ell_q$-norm. The $\ell_0$-penalty simply counts the number of non-zero entries in a vector: $\|x\|_0 = \# \{ i \text{ s.t. } x[i] \neq 0 \}$. For a matrix $X$, the Frobenius norm is defined as $\|X\|_F = \left( \sum_{i=1}^{m} \sum_{j=1}^{n} X[i, j]^2 \right)^{1/2}$. 
Literature Review

Gesture recognition is an active and challenging research topic in the field of computer vision. In this chapter, the literature pertaining to human gesture recognition with depth camera is explored and discussed. Specifically, this chapter begins with an overview of the multiple data modalities used in the task of gesture recognition. The relevant research work on the techniques used in this thesis — including gesture representation, feature extraction, feature representation, classification, and multimodal fusion — are then discussed. And lastly, detailed descriptions of the datasets used in the experiments of Chapters 3–7 are described in detail.

2.1 Data Modalities

A data modality refers to a particular type of technique used to capture human gestures, for example, text, audio, or video. For the last two decades, considerable research has been conducted on the task of gesture recognition, although mainly dealing with RGB video data [97]. Many public datasets available online [7, 120, 121] provide a platform for researchers to evaluate their methods using common benchmarks. With the development of computing ability and the improvement of sensor techniques, a large number of methods have been proposed to address gesture recognition using various data modalities from depth sensors. A substantial body of research exists on the recognition of gestures from RGB video data [97, 116, 151, 2, 67]. In contrast, the research focus of this thesis is on the gesture recognition from the other two modalities: depth maps and skeletal joints. Depth maps and skeletal joints are two
of the most commonly used data modalities in this area. These modalities not only facilitate a rather powerful human motion-capturing technique, but also make it possible to efficiently model human movements.

2.1.1 Depth Maps

The introduction of depth sensors greatly extends the ability of computer systems to sense the 3D visual world and capture low-level visual information. Intrinsically different to RGB data, pixels in a depth map encode the distance information of a scene rather than a measure of the intensity of color. Fig. 2.1 shows the RGB image and corresponding depth image from one gesture sequence in ChaLearn gesture dataset 2011 [49]. As can be seen, the motion ambiguity of capture, such as the huge color and texture variability induced by clothing, hair, skin and background, could be bypassed in the depth image. Texture and color information from depth maps is much less than that in RGB data. Working in low light environment, and being color and texture invariant, depth maps offer several advantages over RGB data as mentioned in Section 1.1. Furthermore, in a depth map, the depth silhouettes of an object can usually be extracted more easily and accurately. In order to effectively represent human gestures, segmenting the gesture regions of each video frame is an essential step in gesture recognition. However it is a challenging task for traditional methods where there is a dynamic background, or where no prior background image exists [115, 14]. In depth images, the values of pixels belonging to the background are greatly different to those belonging to the object, as shown in the histogram of the depth image in Fig. 2.1. Note that the values of the depth map are normalized from 0–255 to plot the histogram. Utilizing the property, the gesture regions in a depth map can be easily segmented from the background using Otsu’s method [112] to classify the pixels [77]. However, depth maps have other drawbacks, one of which is the noise at the edge of objects. With bits missing and a reasonably significant flickering
issue, noise in depth maps resembles a type of salt and pepper noise. Flicker noise and occlusions in 3D silhouettes could negatively affect the performance of action recognition. Hence, a number of robust depth map-based features are proposed to address these issues, which will be reviewed in Section 2.2.1.

![Fig. 2.1 Comparison between RGB image and depth image.](image)

2.1.2 Skeletal Joints

In addition to depth maps, another data modality is accessible — the skeletal joints. Skeletal joints encode 3D human joint positions for each frame in real-time. Compared with modeling the skeleton structures from RGB data, the depth information makes the modeling more feasible and stable. Several algorithms have been proposed and applied to model the skeleton from depth data [43, 124, 130]. The basic premise underlying these methods is to segment the depth data of the human body into multiple parts with dense probabilistic labeling. Segmentation of body parts can be considered a classification task for each pixel in the depth data. The 3D joint positions are obtained based on the spatial modes of the inferred
per-pixel distribution. The first-generation Kinect provides 20 joints for each video frame, while the second-generation Kinect allows up to 25 joints. Fig. 2.2 depicts the positions in which these skeleton joints are located by Kinect.

![First-generation Kinect](https://msdn.microsoft.com/en-us/library/jj131025.aspx)

![Second-generation Kinect](https://msdn.microsoft.com/en-us/library/microsoft.kinect.jointtype.aspx)

Fig. 2.2 Skeleton data depiction from the first-generation Kinect and second-generation Kinect.

The 3D human joint positions output by Kinect are usually noisy when self-occlusions or object occlusions occur, or when the camera is facing the side of the human subject. For example, when one person is crossing arms, bending, or interacting with objects or other people, the skeleton tracker may produce very noisy skeleton information, or even fail. As a consequence, directly utilizing skeletal joints does not usually provide encouraging results. It is necessary to develop skeleton-based features that are robust to noise and occlusions. These approaches will be discussed in Section 2.2.2.

---

2.2 Gesture Representation and Feature Extraction

Instead of working on the raw data from the depth camera which contain many values (e.g., for a 100-frame gesture sequence, the number of values will be 7,680,000 with a frame size of $320 \times 240$ pixels) and redundant information (e.g., the temporal differences could be very small between two consecutive frames), it is necessary to extract a set of features which are considered to be a more compact representation of input data. This process is referred to as feature extraction. According to the modalities from depth sensors, various remarkable feature extraction approaches can be classified into two categories: depth map-based and skeleton-based approaches. Moreover, each category is further divided into two types depending on different gesture representations: space-time features and sequential features. Gesture representation refers to the process of how the gesture sequence is modeled. Space-time features are extracted by representing a gesture sequence as a space-time volume, while sequential features are based on frame-level or subsequence-level information by representing a gesture as a sequence of temporal observations.

2.2.1 Depth Map-Based Features

Depth maps provide complementary information about the 3D body shape and appearance to color data. Many algorithms have been proposed to recognize gestures using features from depth maps.

Depth map-based space-time features mainly utilize features, either local or global, from the space-time volume. It is intuitive to treat depth maps as gray images and to extract 2D video features. The widely used 2D video features include SIFT [84], HOG [28], HOF [69], STIP [69], and kernel descriptors [8]. These features have shown good performance for 3D object recognition using depth silhouettes in RGB-D object datasets [68]. However, depth silhouettes inherently encode 3D shapes and geometric relationships. Oreifej and Liu [111] described the depth sequence using
a histogram to capture the distribution of the surface normal orientation in the 4D space of time (HON4D), depth, and spatial coordinates. The HON4D features treat a depth map sequence as a 4D spatio-temporal shape, compute a 4D normal for each point on this shape, and construct a histogram of the 4D normal vectors. It is able to capture the observed changing structure. Yang and Tian [164] subdivided a depth video into a set of space-time grids, and then adopted a novel scheme of aggregating the low-level polynormals into the super normal vector (SNV). Additionally, an adaptive spatial-temporal pyramid was introduced to capture the spatial layout and temporal order in a global way. This was shown to be a better adaption in terms of retaining the spatial and temporal orders. Inspired by the way in which how 3D computer (or human) vision effectively performs geometric reasoning based on the spatial configuration, Lu [85] proposed binary range-sample features from depth maps. It is based on \( \tau \) tests [16] and achieves reasonable invariance with respect to possible changes in scale, viewpoint, and background. Due to the binary property of the range-sample features, the speed and storage efficiency is impressive. Wang et al. [143] presented semi-local features called random occupancy pattern (ROP) features for the purposes of dealing with occlusions. The ROP features are extracted from randomly sampled 4D subvolumes with different sizes and at different locations. As ROP features are extracted at a larger scale and they only encode information of the most discriminative regions, ROP features are robust to noise and occlusions.

Given the success of template matching methods (e.g., MHI [9]) on 2D video sequences, many approaches have been proposed to transform the problem from 3D to 2D, and to perform recognition on projected 2D planes. Yang et al. [165] developed depth motion maps (DMM) to capture the aggregated temporal motion energies. The 3D silhouettes are projected onto three predefined orthogonal Cartesian planes and then normalized. The DMM-HOG descriptors are constructed by concatenating HOG features from summed binary maps on each plane. Liang and Zheng [78] proposed a 3DMTM-PHOG descriptor to perform human action recognition on
depth sequences. The 3D motion trail model (3DMTM) is generated through the entire depth video sequence to encode additional motion information from three projected orthogonal planes. By adding pyramid representation, the 3DMTM-PHOG descriptor is able to represent the 3DMTM in different degrees of detail according to the selected levels.

To briefly summarize, depth map-based space-time features are extracted by considering each gesture sequence as a 4D volume in spatial \((x, y, z)\) and temporal \(t\) directions. The sequence can be processed either as a whole, or as a set of local feature points. The features based on 2D projected planes have shown promising results. Nevertheless, the major limitation of these features is that they are highly dependent on very large motions, which means that subtle motions will decrease the performance.

*Depth map-based sequential features* are extracted through explicitly representing temporal dynamics from depth sequences. Bag-of-words models have previously been used in text recognition and object recognition [36, 126]. For gesture recognition, local features are extracted from the spatial and temporal space. Li et al. [75] employed an action graph approach on a bag of 3D points to model actions. Each bag of points extracted from a 3D silhouette represents a salient posture which is one node in an expandable graphical model of actions. However, this method is view-dependent. Moreover, due to noise and occlusions in the depth maps, the projections of silhouettes may not be reliable. To address these issues, Vieira et al. [142] proposed a new representation for 3D action recognition, named space-time occupancy patterns (STOP). This method roughly characterizes the 4D space-time patterns of human actions by partitioning the 4D video volume into 4D space-time cells, and aggregating the occupancy information in each cell. STOP descriptors leverage the spatial and temporal contextual information while allowing for intra-class variations. Jalal et al. [55] utilized depth silhouettes and \(\mathbb{R}\) transformation [132] to continuously recognize human actions in an indoor environment. \(\mathbb{R}\) transformation
Literature Review

is applied on the depth silhouettes to compute a 2D angular projection map of an action silhouette. PCA and LDA are then applied to extract robust features from the \( \Re \) transformed profiles of depth silhouettes. Malgireddy et al. [94] implemented HOG and HOF features to obtain a dense population of descriptors. The visual words clustered from descriptors are then re-expressed in terms of topics so that each frame can be seen to be made up of visual words which belong to different topics through LDA process. LDA is used here to reduce the size and refine the meaning of the observable features within each frame.

In contrast to the amount of research on depth map-based space-time features, relatively less study has been made of gesture recognition using depth map-based sequential features, at the time of writing. Analyzing human motions is difficult because great variety exists. Directly applying the sequential features for 2D video may not be appropriate, so depth map-specific techniques should be considered.

2.2.2 Skeleton-Based Features

Skeleton-based features are normally extracted from skeletal joints of human gestures. They are typically easier to extract and require less computational cost and memory than depth map-based features.

*Skeleton-based space-time features* are usually extracted from the skeletal joints to encode temporal motion information. Yang et al. [163] developed the EigenJoints features from the skeleton data of action sequences. EigenJoints feature employs position differences between joints to represent human actions. PCA is applied to the concatenated feature vector within the current frame and consecutive frames to extract “EigenJoints”, which encode the most important human pose information for action recognition. The work by Wang et al. [145] utilized both skeleton and point cloud information. These authors extracted local occupancy patterns (LOP) features at each joint and learned an actionlet ensemble model to represent each
action, and to capture the intra-class invariance. The features are robust to noise, and invariant to translational and temporal misalignments. Luo et al. [86] proposed a new discriminative dictionary-learning algorithm (DL-GSGC) that incorporated both group sparsity and geometry constraints to better represent skeleton features. In addition, the temporal pyramid matching method was applied to each action sequence to keep the temporal information in the representation.

Most of the previous skeleton-based space-time features use either the joint locations or the joint angles to represent human actions. Recently, Vemulapalli et al. [141] proposed a new skeletal representation that explicitly models the 3D geometric relationships between various body parts using rotations and translations in 3D space. They represented a human skeleton as a point in the Lie group so that human actions are modeled as curves in the Lie group. The experimental results showed that the representation based on the Lie group performs better than many existing skeleton representations.

*Skeleton-based sequential features* are typically extracted from skeletal joints within frames or subsequences to model temporal dynamics of actions. Skeleton-based sequential features are able to offer low-latency responses that allow the rapid identification of an action long before it ends. Xia et al. [160] proposed a feature called histogram of 3D joint locations (HOJ3D) using modified spherical coordinates. HOJ3D is essentially able to encode spatial occupancy information relative to the skeleton root, i.e., hip center. The main advantage is the real-time performance. In order to continuously recognize human actions from unsegmented sequences, Zhao et al. [176] proposed an effective and efficient framework for online human action recognition. In addition, their proposed structured streaming skeletons (SSS) feature is able to handle intra-class variations including viewpoint, anthropometry, execution rate, and personal style. To cope with unsegmented action sequences, Zanfir et al. [173] proposed a fast, simple, yet powerful non-parametric moving pose (MP) descriptor for low-latency human action recognition. The proposed MP descriptor is
a novel frame-based dynamic representation. It captures not only 3D body poses but also differential properties like the speed and acceleration of the human body joints within a short time window around the current frame. More recently, Wu and Shao [156] proposed a hierarchical dynamic framework based on high-level skeletal joints features to segment and recognize actions simultaneously. They make feature extraction from skeleton data an implicit approach by deep belief networks. The features themselves are discovered by building a multi-layer generative model of much richer information in the configurations of skeletal joints.

2.3 Feature Representation

Given that different gesture sequences will generate different numbers of features, it is necessary to generate a vector representation for the gesture sequence, with the dimension of such vector being the same for all the sequences to be classified. The process of generating a unique-length vector representation based on extracted features is known as feature representation. Feature representation treats each gesture sequence as a collection of extracted features from the corresponding gesture representation, and generates a vector representation for each sequence to encode the human gesture features.

2.3.1 Bag of Visual Words

To obtain the final representation of a gesture, the bag of visual words (BoVW) model [36] has been widely used, achieving good results in human gesture recognition tasks [30, 52, 108]. The BoVW model is actually based on mapping local features of each video sequence onto a pre-learned codebook, with every cluster center corresponding to a “codeword”. Specifically, this method treats each video sequence
as a collection of features and quantizes them into the nearest “codewords” in terms of the Euclidean distance.

In order to keep the temporal order of extracted features, existing methods based on the BoVW framework usually use a pyramid matching scheme by dividing the features into different segments according to spatial and temporal location of interest points. However, the BoVW model will generate large quantization error into the final representation, as each feature is assigned to the nearest codeword only, impacting negatively on recognition performance. In addition, computational cost is expensive as the BoVW model performs well when combined with a particular type of nonlinear Mercer kernels [11], for example, the intersection kernel [93] or the chi-square kernel [99]. Furthermore, the size of the codebook must be empirically determined, and codewords obtained by $k$-means are less effective for action primitives. Another disadvantage of the BoVW model is that it ignores the spatial and temporal relationships among the extracted features, which is very important in spatio-temporal representation for gesture sequences. Therefore, robust features are required to achieve robustness against the noise and occlusion. Moreover, extracted features should be properly represented to solve the problem in the BoVW model.

Recently, a number of encoding methods have been developed to improve the performance of BoVW, such as vector quantization [27], soft assignment encoding [82], sparse coding [162], locality-constrained linear (LLC) encoding [147] and Fisher kernel encoding [114]. Vector quantization, known as hard assignment, represents each local feature descriptor by its nearest visual word in the dictionary. In soft-assignment, for each local feature, the $k$-th coefficient represents the degree of membership of the local feature being to the $k$-th visual word. Sparse coding represents a local feature by a sparse linear combination of basis vectors. Unlike the sparse coding, LLC enforces locality instead of sparsity and this leads to smaller coefficient for the basis vectors far away from the local feature. Different to previous encoding
methods based on a codebook, the fisher kernel characterizes input feature descriptor with a gradient vector derived from a generative probability model. Based on the comprehensive experiments in [150], among the encoding methods, sparse coding and fisher kernel methods seem to receive better performance. The computational cost of training classifier for fisher kernel is much larger than the other encoding methods. Therefore, this work adopts sparse coding in the framework of BoVW.

2.3.2 Sparse Coding

Sparse coding is a method for discovering good basis vectors automatically using only unlabeled data [71]. Sparse representation has recently been introduced for gesture representation based on local features [47]. Compared with the codebook used in the BoVW framework, the learned overcomplete basis set in sparse coding is referred to as the “dictionary”, consisting of a set of representative vectors learned from a large number of features. These representative vectors are referred to as the “codewords” in BoVW and “atoms” otherwise. A considerable amount of quantization error is incurred by the approximation process whereby each feature vector is assigned to the nearest codeword in BoVW. This can largely be improved by sparse coding that allows a linear and sparse combination of atoms to be used in the approximation process. The calculated sparse codes for one feature vector correspond to the response of that feature to all the atoms in the dictionary.

In the computer vision community, dictionary learning (DL), as a particular sparse coding model, aims to learn a set of atoms that can be linearly combined to well approximate a given feature vector. From the viewpoint of compressive sensing, it was originally designed to learn an adaptive dictionary to faithfully represent the signals with sparsity constraint [91]. In recent years, researchers have applied DL frameworks to many applications, such as image denoising [31], inpainting [32], clustering [25, 155, 64] and classification [125, 13, 92].
Concretely, given a dataset of \( n \) training feature vectors \( \mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{m \times n} \), dictionary learning can be formulated as the following minimization problem:

\[
\min_{\mathbf{D} \in \mathcal{D}, \mathbf{A} \in \mathbb{R}^{p \times n}} \sum_{i=1}^{n} \| \mathbf{x}_i - \mathbf{D} \alpha_i \|_2^2 + \lambda \psi(\alpha_i)
\]  

(2.1)

where \( \mathbf{A} = [\alpha_1, \ldots, \alpha_n] \) encodes the decomposition coefficients of the feature vector \( \mathbf{x}_1, \ldots, \mathbf{x}_n \), \( \psi \) is the sparsity-inducing regularization function, and \( \mathcal{D} \) is typically chosen as the following set:

\[
\mathcal{D} \triangleq \{ \mathbf{D} \in \mathbb{R}^{m \times p} | \forall j, \| \mathbf{d}_j \|_2 \leq 1 \}.
\]  

(2.2)

### 2.4 Classification

Constructing a good feature representation for depth sequences is a vital step in the task of human gesture recognition. The task also requires effective machine learning algorithms to understand and interpret the semantic meanings of human gestures. This section will give a brief overview of the classification of human gesture recognition using depth sensors after feature extraction and representation.

Space-time features represent a gesture sequence as a space-time volume, in which local or global features are extracted for discriminative models (e.g., SVM [26]). Direct classification is the simplest classification. If robust space-time features are extracted from gesture sequences, a classification model can be learned by directly applying discriminative machine learning algorithms. SVM is widely applied in the reviewed space-time features, such as, ROP [143], HON4D [111], DMM-HOG [165], 3DMTM-PHOG [78], SNV [164], range-sample feature [85], actinolet ensemble [145] and Lie group-based features [141]. The main advantages of direct classification are easy implementation and discriminative power. However, this is only feasible when there is little noise in the depth and skeleton data, and
the gestures have relatively simple temporal structures. To address these issues, some approaches perform classification using the BoVW framework, such as DL-GSGC [86]. The BoVW framework can build gesture classifiers that are robust to variation in subject location, background, and movement speed. Its performance, nevertheless, is largely limited by the discriminative power of local descriptors, and it fails to capture the temporal structure of gestures. Nearest neighbor-based methods such as EigenJoints [163] can also be used for action recognition.

On the other hand, modeling the temporal structure is an essential element in successfully representing complex human gestures. Sequential features view a gesture as a sequence of temporal observations. Local features extracted from the data of each temporal observation can be used for a generative model. The action graph approach used for bag-of-3D-points [75] and STOP features [142] models an action as an acyclic directed graph, where every node represents a hidden posture state and every edge represents a transition between the states. Another generative model, the HMM-based approach [118], is also utilized to model the temporal dynamics of sequential features, such as the ℜ transformation-based feature [55], HOJ3D [160] and the high-level skeleton feature [156]. Other approaches have also been proposed for sequential features. Classic least square regression was used in [176] to transfer SSS features to gesture labels, while in [173] the $k$-NN classifier was modified by considering both the temporal location as well as the discriminative power of the MP descriptor.

2.5 Multi-Modal Fusion

Multi-modal fusion is the process of combining data from multiple modalities such that the resulting performance is in many regards better than that provided by any of the individual modalities. Human gestures are highly dependent on various modalities, such as depth maps and skeletal joints. Using the multiple modalities
provided by the Kinect sensor, depth features can be extracted from depth maps to encode shape information, and skeleton features can be extracted from skeletal joints to encode motion information. Based on the complementary properties of these particular features, it is beneficial to utilize multiple modalities for gesture recognition. Combining heterogeneous features leads to a multi-modal combination, which demands sophisticated fusion algorithms.

Early fusion and late fusion are two main strategies. Early fusion is usually conducted at feature level or representation level. The features or representations are firstly generated from the multiple modalities and then concatenated to train a final classifier. In [131], skeletal information was integrated in two ways to extract HOG features from RGB and depth images: either from global bounding boxes containing a whole body or from regions containing an arm, a torso and a head. Similarly, skeletal information was fused with HOG features extracted from either RGB or depth [23, 101, 18, 76]. Monnier et al. [100] combined a sliding-window gesture detector with features extracted at multiple temporal scales from skeleton data, color imagery, and depth data. In the work of [18], a general probabilistic model was formalized and constructed by nonparametrically estimating multi-modal densities from training samples. The skeletal joint data and RGB images were utilized as the multi-modal input to their proposed method.

For late fusion, the scores from multiple classifiers related to different modalities or multiple classifiers are fused or combined. Wu et al. [158], the winning team in the 2013 Multi-Modal Gesture Recognition Challenge [34], proposed a novel multi-modal continuous gesture recognition framework, which makes a full exploration of both audio and skeleton data. The final recognition score was a linear combination of two gesture classifiers, based on audio and skeleton features. A multi-modal gesture recognition system has also been developed in [105] to detect as well as recognize the gestures. This system adopted audio, RGB video, and skeleton joint models for weighted likelihood fusion from multiple classifiers.
Multi-modal data based methods utilize different properties of multiple modalities, thus usually improving recognition performance. A major issue with these methods, however, is how to fuse the results from multiple modalities. While in most practical applications, late fusion of scores output by several models offers a cheap and surprisingly effective solution [60]; both late and early fusion of either final or intermediate data representations remain under active investigation. A significant amount of work on early combining of diverse feature types has been applied to object and action recognition. Multiple kernel learning (MKL) [5] has been actively discussed in this context. Ye et al. [168] proposed a late fusion strategy compensating for errors of individual classifiers by minimizing the rank of a score matrix. Nataranjan et al. [106] employed multiple strategies, including MKL-based combinations of features, Bayesian model combination, and weighted average fusion of scores from multiple systems. The recent work of [107] proposed a simple voting strategy for fusion with a single weight per modality to obtain the final predication. Bayer and Silbermann [6] presented an algorithm for recognizing gestures by combining two data sources (audio and video) through weighted averaging.

2.6 Datasets

As depth sensors such as Kinect have attracted attention from researchers in many areas, several research groups have built human gesture datasets by depth sensors using various settings for different purposes. These datasets have been made publicly available. Table 2.1 shows a summary of human gesture datasets recorded by depth sensors, that will be used in Chapters 3–7. These datasets emphasize different application purposes, and therefore the gesture categories vary between each. The details of these datasets are presented in this section.
### 2.6 Datasets

**Table 2.1 Human gesture datasets recorded by depth sensors.**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#cls.</th>
<th>#seq.</th>
<th>Modalities</th>
<th>Settings</th>
<th>Technical details</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSR Action3D dataset [75]</td>
<td>20</td>
<td>567</td>
<td>depth maps, skeletal joints</td>
<td>10 subjects, 2 or 3 repetitions</td>
<td>Resolution: 320×240; Without background; Segmented sequences</td>
</tr>
<tr>
<td>MSR Gesture3D dataset [143]</td>
<td>12</td>
<td>336</td>
<td>depth maps</td>
<td>10 subjects, 2 or 3 repetitions</td>
<td>Various resolutions; Without background; Segmented sequences</td>
</tr>
<tr>
<td>MSR Action Pairs dataset [111]</td>
<td>12</td>
<td>360</td>
<td>RGB data, depth maps, skeletal joints</td>
<td>10 subjects, 2 or 3 repetitions</td>
<td>Resolution: 320×240; With background; Segmented sequences</td>
</tr>
<tr>
<td>ChaLearn Multi-Modal dataset [33]</td>
<td>20</td>
<td>13858</td>
<td>RGB data, depth maps, user mask, skeletal joints</td>
<td>20 subjects; Each sequence contains 8~20 gesture samples</td>
<td>Resolution: 640×480; With background; Unsegmented gesture sequences</td>
</tr>
</tbody>
</table>

*a* the number of gesture classes  
*b* the number of video sequences

### 2.6.1 MSR Action3D Dataset

The MSR Action3D dataset [75] is an action dataset of depth sequences captured by a depth sensor similar to the Kinect device. It contains 567 depth map sequences. There are 20 action types: “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 swing”, “tennis serve”, “golf swing”, and “pick up & throw”. These actions were chosen in the context of gaming and cover a variety of movements related to arms, legs, torso, and their combinations. Each action is performed two or three times by ten subjects. The resolution of the depth map is $320 \times 240$. Sample frames of the action sequences are shown in Fig. 2.3. The background in this dataset is preprocessed to clear the discontinuities induced by undefined depth regions.

---

Nevertheless, this dataset remains challenging because many of the actions are highly similar to each other.

Fig. 2.3 Sample frames from the MSR Action3D dataset.

### 2.6.2 MSR Gesture3D Dataset

The MSR Gesture3D dataset [143]\(^1\) is a dynamic hand gesture dataset of depth sequences captured by a depth camera. It contains 12 dynamic hand gestures defined by the American Sign Language (ASL): “z”, “j”, “where”, “store”, “pig”, “past”, “hungry”, “green”, “finish”, “blue”, “bathroom”, and “milk”. There are ten subjects, each one performing each gesture two or three times. The resolutions of the depth maps are varied. Some example frames of the gestures are shown in Fig. 2.4. This dataset presents more self-occlusions than other datasets. Notice that all of the

---

\(^1\)Available at http://research.microsoft.com/en-us/um/people/zliu/actionrecorsrc/
2.6 Datasets

Fig. 2.4 Sample frames from the MSR Gesture3D dataset.

gestures in this dataset are dynamic gestures, which means both the shape and movement of the hands are important for the semantics of the gesture.

2.6.3 MSR Action Pairs Dataset

The MSR Action Pairs dataset [111] selects pairs of actions, so that within each pair the motion and the shape cues are similar, although their correlations are different. For example, “push” and “pull” actions have a similar motion and shape, but the co-occurrence of the object shape and hand motion is in different spatio-temporal order. Therefore, this dataset can be used to evaluate the performance of action representation in the context of capturing the prominent cues in the sequence. Six pairs of actions are collected: “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 a hat”, “put on a backpack / take off a backpack”, and “stick a poster / remove a poster”. The sample frames are shown in Fig. 2.5. Each action is performed three times by ten different subjects.

1Available at http://www.cs.ucf.edu/oreifej/HON4D.html
2.6.4 ChaLearn Multi-Modal Dataset

ChaLearn Multi-Modal dataset [33] is based on Italian gestures. It is recorded with Kinect. There are twenty Italian sign gestures: “vattene”, “vieniqui”, “perfetto”, “furbo”, “cheduepalle”, “chevuoi”, “daccordo”, “seipazzo”, “combinato”, “freganiente”, “ok”, “cosatifarei”, “basta”, “prendere”, “noncenepiu”, “fame”, “tantotempo”, “buonissimo”, “messidaccordo”, and “sonostufo”. The sample frames are shown in Fig. 2.6. The dataset is focused on “multiple instances, user independent learning” of gestures. ChaLearn Multi-Modal dataset contains RGB videos, depth videos, user mask videos, and skeleton data. This dataset is challenging due to the high intra-class variability of gesture samples and low inter-class variability for some gesture categories. Gestures in this dataset are continuous and some of them are distractor gestures.

Fig. 2.5 Sample frames from the MSR Action Pairs dataset.

---

Fig. 2.6 Sample frames from ChaLearn Multi-Modal dataset.
3.1 Introduction

Gestures in video sequences primarily contain temporal motion and 3D structure information. Temporal motion refers to the movement of gestures within the temporal domain, while 3D structure refers to the poses of a human body and the relative positions between them in 3D space. This chapter presents an effective approach for recognizing human gestures in depth videos, where depth maps provide additional body shape and motion cues. In the proposed approach, depth maps are projected onto three orthogonal planes, and a depth motion history template (DMHT) is generated through the entire video sequence to encode temporal motion along with structure information. Pyramid histogram of oriented gradients (PHOG) is then calculated from DMHT as the features of a gesture sequence. Experimental results from the MSR Action3D dataset [75] demonstrate that the approach outperforms the previous methods.

The depth maps are able to provide additional body shape and motion information to distinguish gestures with similar projections from a single view. This has spurred recent research which explores gesture recognition based on depth maps, as discussed in Section 2.2.1. This chapter focuses on recognizing human gestures using sequences of depth maps. An effective human gesture recognition framework based on the DMHT-PHOG descriptor is proposed, as illustrated in Fig. 3.1.
DMHT-PHOG for Gesture Recognition

Fig. 3.1 The framework based on DMHT-PHOG descriptor.
The framework consists of three components, gesture representation using DMHT, feature extraction from DMHT using PHOG descriptor, and classification. More specifically, DMHT is an extension of motion history image (MHI) [9]. It is generated by capturing the temporal motion changes of depth maps projected onto three orthogonal planes. The captured temporal motion changes of each gesture category produces a specific appearance and shape cues on DMHT. It has the ability to capture gesture motion information within temporal ordering in 3D space. This can be used to characterize the corresponding gesture categories. Motivated by the success of the histogram of oriented gradients (HOG) [28], this work suggests that pyramid histograms of oriented gradients (PHOG) can explicitly extract features from DMHT by using spatial pyramid representation. In contrast to PHOG descriptors proposed in [10, 146], the PHOG proposed in this work is directly extracted from all the DMHT templates, without extracting the edges of objects or capturing the interesting regions, which was necessary in [10, 146]. Compared with the original depth data, the proposed DMHT-PHOG descriptor is more compact and more discriminative in encoding human gestures. Finally, a support vector machine (SVM) [15] is employed to recognize multiple gesture categories in the final stage.

### 3.2 Depth Motion History Template

In this work, DMHT is positioned to represent human gestures from depth sequences. It is based on the motion history image (MHI) method because of its effectiveness in capturing gesture changes and computational efficiency [9]. MHI has been widely used for gesture representation, presenting the temporal motion changes by condensing the gesture sequence into a single grayscale image. Intensity of each pixel in the MHI is a function of motion density at that location. Dominant motion history information is preserved by computing the differences between consecutive frames. MHI is capable of encoding the dynamics for a sequence of moving silhouettes in
the front view. Although silhouette-based images are able to represent a wide variety of body configurations, they could produce ambiguities in the presence of occlusions of the body. In addition, MHI can only capture the motion history induced by the lateral movement of the scene parallel to the image plane. Thus, information lost in the depth channel could cause significant degradation of the representation.

In order to make the best use of the additional motion information from depth maps in this work, each depth frame will firstly be converted to point clouds in 3D space. Fig. 3.2 illustrates how to convert a depth value in a depth map to the corresponding 3D point. The coordinate \((x_w, y_w, z_w)\) of the 3D point can be calculated according to the following equations:

\[
x_w = \frac{(x_p - c_x) \cdot z_w}{f_x}; \quad y_w = \frac{(y_p - c_y) \cdot z_w}{f_y}
\]

where \((x_p, y_p)\) and \(z_w\) denote screen coordinates and the depth value respectively, \((c_x, c_y)\) denotes the center of the depth map, and \(f_x\) and \(f_y\) are the focal lengths of the Kinect camera. Here for Kinect, \(f_x = f_y = 580\) [128].

![Fig. 3.2 Illustration of 3D point clouds calculation.](image)

The point clouds in each frame are then projected onto three orthogonal planes as shown in Fig 3.3, including front, top and side views, denoted by \(I_v\), where \(v \in \{\hat{f}, \hat{t}, \hat{s}\}\). When \(P_v\) denotes the projected coordinates on the plane \(v\) with respect to 3D points of the depth data, the corresponding projected coordinates of a 3D point \((x_w, y_w, z_w)\) from a depth frame are \(P_{\hat{f}} = (x_w, y_w)\), \(P_{\hat{t}} = (x_w, z_w)\), and \(P_{\hat{s}} = (y_w, z_w)\),
respectively. The $t$-th depth frame $I(x_w, y_w, z_w, t)$ can be denoted as a set of three projected gray images: $\{I_f(P_f, t), I_t(P_t, t), I_s(P_s, t)\}$. Each value in $I_f$ is the distance from the depth camera, and the values of $I_t$ and $I_s$ are the numbers of 3D points that have the same projected coordinates. These projected images are then converted to grayscale by normalization. In this way, three 2D planes are generated from each depth frame according to front, top and side views.

![3D point clouds projections](image)

Fig. 3.3 3D point clouds projections.

Different to [165], where depth motion maps are described by accumulating motion distribution of three planes, this work uses the proposed DMHT to represent depth sequences. The DMHT not only captures the 3D motion distribution and intensity, but also encodes the motion changes in the temporal direction; crucial in capturing the spatio-temporal orders to distinguish gestures from those with similar motion. With the purpose of finding motion regions, $D_v(P_v, t) = |I_v(P_v, t) - I_v(P_v, t - 1)|$ is defined to calculate the differences of the $t$-th frame and the previous frame. Then, by arranging the successive $D_v(P_v, t)$, the DMHT of a depth sequence can be obtained in a recursive manner:

$$M_v(P_v, t) = \begin{cases} T & \text{if } D_v(P_v, t) > \xi \\ \max(0, M_v(P_v, t - 1) - \sigma) & \text{otherwise} \end{cases}$$

(3.2)
where $\xi$ is the minimal intensity difference between two frames for motion detection, and $\sigma$ is the decay parameter. Here, $T$ is chosen as the whole length of the sequence, and the value of $\sigma$ can be 1 or more. Note that the operation of motion detection is recursively processed for every frame or set of consecutive frames that are analyzed in that depth map sequence. It could be necessary to perform median filtering to smooth the DMHT and remove the salt-pepper noise, or to implement a Gaussian filter for the same purpose. An example of the DMHT of one gesture from the MSR Action3D dataset is shown in Fig. 3.4. There are 34 frames in the depth sequence. The DMHT of the gesture can be obtained through Eq. (3.2). For this gesture, the DMHT of the duration of the gesture ($t = 34$) is used as the final gesture representation, including three gray images $M_\hat{f}(34)$, $M_\hat{t}(34)$ and $M_\hat{s}(34)$. It can be seen that the proposed DMHT is able to capture the temporal changes of the gesture in a compact way.

### 3.3 The DMHT-PHOG Descriptor

The histograms of oriented gradients (HOG) has been experimentally proved to outperform other features that encode human figures in human detection [28]. HOG, as a dense descriptor is computed across the whole image. Fig 3.5 shows an example of the computation of the HOG features. In this example, the image is resized to $128 \times 128$ pixels. It is divided into $16 \times 16$ grids with the size of each cell being $8 \times 8$ pixels. The gradient vector for each pixel in the cell is calculated, and the orientations of the gradient vectors are stored into a 9-bin histogram ranging from 0 to 180 degrees. According to [28], the magnitude of the gradient vector is added to the histogram bins. The magnitude $m(x, y)$ and orientation $\theta(x, y)$ of the gradient on a pixel $(x, y)$ are calculated as:

$$m(x, y) = \sqrt{g_x(x, y)^2 + g_y(x, y)^2}$$ (3.3)
3.3 The DMHT-PHOG Descriptor

Fig. 3.4 An example of the DMHT. (a) $I_j(t)$ is the front projection of the $t$-th depth frame (b) $M_j(t)$ is the front view of DMHT of the temporal duration between 1 and $t$ (c) $M_j(t)$ is the top view of DMHT of the temporal duration between 1 and $t$ (d) $M_j(t)$ is the side view of DMHT of the temporal duration between 1 and $t$.

Fig. 3.5 An example of HOG feature extraction. (a) front view of DMHT (b) cells and blocks assigned to the image (c) visualization of the HOG features.
\[ \theta(x, y) = \arctan \frac{g_x(x, y)}{g_y(x, y)} \]  

(3.4)

where \( g_x(x, y) \) and \( g_y(x, y) \) are image gradients along the \( x \) and \( y \) directions, respectively. After calculating the histograms for all cells, the cells are grouped into blocks of size 2 cells, and the histograms are concatenated into one feature vector. The final HOG descriptor is then obtained as the concatenation of all block histograms.

Edges and regions of interest can effectively encode object shapes and areas, and they have been widely used for action representation. Bosch et al. [10] extended HOG and proposed a spatial pyramid representation of object edges based on HOG to encode object shapes. In contrast to [10], the edges of human subjects are not extracted in the PHOG as proposed in [146]. Instead, the gradients of the regions of interest are used to accumulate a histogram. However, edges or regions of interest are usually difficult to segment or extract in practice. To this end, this work proposes to extract PHOG from DMHT in order to characterize local shapes at different spatial scales for gesture recognition.

In order to introduce spatial information, a method based on spatial pyramid matching [45] is employed. Each image is divided into a sequence of increasingly finer spatial blocks by repeatedly doubling the number of divisions in each axis direction. The number of pixels in each block is then recorded. This is a pyramid representation of HOG. The proposed DMHT-PHOG descriptor is extracted in a pyramid way according to different pyramid levels. It is directly performed on the DMHT, which requires no edge or regions of interest extraction. The pyramid at level \( l \) has \( 2^l \times 2^l \) blocks (\( l = 0, 1, \ldots \)), each of which has \( 2 \times 2 \) cells. For example, there are \( 4 \times 4 \) blocks when pyramid level is 2 (\( l = 2 \)), and each block has \( 2 \times 2 \) cells. Within each cell, each gradient orientation is quantized into \( K \) bins. Gradients from all the pixels within a cell are accumulated to form a local \( K \) bins 1-D histogram, so each block has a \( 4 \times K \) bins histogram. Therefore, each view from DMHT at level \( l \) is represented by a \( 4 \times K \times 2^l \times 2^l \) dimension vector. Since there are three views in DMHT, the
three PHOG vectors from the views are concatenated as the final DMHT-PHOG descriptor. The obtained feature vector, $f \in \mathbb{R}^d$ ($d = 3 \times 4 \times K \times \sum_{l=0}^{L} (2^l \times 2^l)$), is the DMHT-PHOG descriptor of the DMHT. It is straightforward that more important view is supposed to have larger weight, but the important view varies from different gestures. It is a challenging task to use different weights with respect to different views. Thus, the proposed DMHT-PHOG descriptor assigns an equal weight to each view. However, it is worthwhile to investigate methods to learn distinctive weights for different views.

Fig. 3.6 shows examples of PHOG extracted from the DMHT of two different gestures. The PHOG descriptor in the example is at up to two pyramid levels for different gestures performed by different subjects. The Fig. 3.6(a) and Fig. 3.6(b) are front views of the DMHT of “horizontal arm wave” performed by two subjects. The corresponding HOG descriptors at different pyramid levels are shown in the same rows. It is evident that the similar human gestures have similar HOG descriptors at each pyramid level (i.e., $l = 0, 1, 2$). Fig. 3.6(c) and Fig. 3.6(d) are front views of DMHT of “two hand wave” performed by two different subjects. Obviously, these two gestures have similar HOG descriptors at different levels; however, HOG descriptors differ significantly between the “horizontal arm wave” and “two hand wave” gestures. Therefore, the proposed DMHT-PHOG descriptor has the ability to discriminate between human gestures.

### 3.4 Classification

For classification, the support vector machine (SVM) [17] is the most popular classifier with strong discriminative power. An SVM is a binary classifier that aims to distinguish between two classes of instances by finding the maximum separating
Fig. 3.6 Examples of PHOG descriptor at different spatial levels.
hyperplane between the two [138]. The decision function is defined as

\[
    f(x) = \sum_{i=1}^{n} w_i \mathcal{K}(x, x_i) + b
\]  

(3.5)

where \(\{(x_i, y_i)\}_{i=1}^{n}\) is the training set and \(y_i \in \{-1, 1\}\) is the label.

To speed up the process of training and testing, a linear kernel \(\mathcal{K}(\cdot, \cdot)\) is used on the extracted DMHT-PHOG descriptors. Given the training data \(x_i\) and the test sample \(x\), the kernel can be calculated as

\[
    \mathcal{K}(x, x_i) = x_i^T x
\]  

(3.6)

Combining Eqs.(3.5) and (3.6), the binary SVM decision function can further be represented as

\[
    f(x) = \left( \sum_{i=1}^{n} w_i x_i \right)^T x + b = w^T x + b
\]  

(3.7)

For multi-class, the linear SVM is equivalent to learning \(Q\) linear functions \(\{w_q^T x \mid q \in \mathcal{Y}\}\), given the training data \(\{(x_i, y_i)\}_{i=1}^{n}\), \(y_i \in \mathcal{Y} = \{1, \ldots, Q\}\). For a test sample \(x\), its class label is predicated by

\[
    y = \arg \max_{q \in \mathcal{Y}} w_q^T x
\]  

(3.8)

The one-against-all strategy [162] is adopted to train the \(Q\) linear SVMs by solving the optimization problem

\[
    \min_{w_q} \|w_q\|_2^2 + C \sum_{i=1}^{n} \ell(w_q; y_i^q, x_i)
\]  

(3.9)

where \(\ell(w_q; y_i^q, x_i) = \left[ \max(0, w_q^T x_i \cdot y_i^q - 1) \right]^2\) is the differentiable quadratic hinge loss and can be solved by gradient-based optimization methods. \(y_i^q = 1\) if \(y_i = q\), otherwise \(y_i^q = -1\).
3.5 Experiments

The proposed method has been evaluated using the MSR Action3D dataset [75]. To comprehensively evaluate the proposed DMHT, various values selection for motion threshold $\xi$ and decay parameter $\sigma$ is illustrated in subsection 3.5.1. The experimental setup is illustrated in subsection 3.5.2. Subsection 3.5.3 extensively compares the proposed method with the previous methods in a variety of experimental settings.

3.5.1 Dependence on $\xi$ and $\sigma$

Fig. 3.7 demonstrates typical examples of the selection of threshold values ($\xi$) for the frame difference in Eq. 3.2. It presents DMHTs for a gesture with different threshold values (i.e., 25, 50, 75 and 100 from left to right). Noise is present when the threshold is set to 25, especially in the top views and side views of the DMHT. It can also be seen that with the increase of $\xi$, some parts of the motion information are missing. Therefore, the selection of the threshold value ($\xi$) is crucial for generating DMHT for a gesture. In the experiments, $\xi$ is chosen empirically as 50.

Fig. 3.8 shows the dependence on the decay parameter ($\sigma$) for generating DMHT. When loading the depth frames, if there is no change (or no presence) of motion in a specific pixel where earlier there was motion, the pixel value is reduced by $\sigma$. However, having different $\sigma$ values may provide slightly different information. Hence, the value can be chosen empirically. Fig. 3.8 shows the final DMHT for the same gesture with different $\sigma$ values (i.e., 1, 3, 5 and 10). It can be seen that the higher values for $\sigma$ remove the earlier trail of motion region. For example, when $\sigma = 10$, part of the earlier motion information is missing, which could negatively impact the performance of gesture recognition. In the following experiments, the value of $\sigma$ is chosen as 1 empirically.
3.5 Experiments

(a) Front views of DMHT

(b) Top views of DMHT

(c) Side views of DMHT

Fig. 3.7 Selection of threshold value ($\xi$) for DMHT. Note the presence of noise or holes in various images. $\xi$ is set to 25, 50, 75 and 100 from left to right.
Fig. 3.8 Selection of decay parameter ($\sigma$) for DMHT. Note the presence of the early trail of motion in various images. $\sigma$ is set to 1, 3, 5 and 10 from left to right.
3.5 Experiments

3.5.2 Experimental Setup

As introduced in subsection 2.6.1, the MSR Action3D dataset contains 20 action categories performed by 10 subjects. These action categories are chosen in the context of interactions with game consoles. This work follows the same experimental settings as Li et al. [75] to split 20 categories into three subsets, each having eight actions as listed in Table 3.1. All the subsets (AS1, AS2 and AS3) are deliberately constructed so that similar actions are included within the same subset. For each subset, there are three different tests, i.e., Test One (T1), Test Two (T2), and Cross Subject Test (CST). In T1, 1/3 of the samples are used as training and the rest as testing; in T2, 2/3 samples are used as training and the rest as testing; in CST, subjects 1, 3, 5, 7, 9 are used for training and 2, 4, 6, 8, 10 are used for testing. As different subjects have their own styles to perform actions, there are large variations among training and testing actions in CST.

<table>
<thead>
<tr>
<th>Action Set 1 (AS1)</th>
<th>Action Set 2 (AS2)</th>
<th>Action Set 3 (AS3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>horizontal arm wave</td>
<td>high arm wave</td>
<td>high throw</td>
</tr>
<tr>
<td>hammer</td>
<td>hand catch</td>
<td>forward kick</td>
</tr>
<tr>
<td>forward punch</td>
<td>draw x</td>
<td>side kick</td>
</tr>
<tr>
<td>high throw</td>
<td>draw tick</td>
<td>jogging</td>
</tr>
<tr>
<td>hand clap</td>
<td>draw circle</td>
<td>tennis swing</td>
</tr>
<tr>
<td>bend</td>
<td>two hand wave</td>
<td>tennis serve</td>
</tr>
<tr>
<td>tennis serve</td>
<td>forward kick</td>
<td>golf swing</td>
</tr>
<tr>
<td>pickup &amp; throw</td>
<td>side boxing</td>
<td>pickup &amp; throw</td>
</tr>
</tbody>
</table>

3.5.3 Evaluation of DMHT-PHOG

In all experiments, three levels ($L = 2$) are selected for the DMHT-PHOG descriptor, as no improvement in accuracy is observed when more levels are used. The number of gradient orientation bins $K$ is set to 9 as mentioned in Section 3.3. Thus, the DMHT-PHOG generates a descriptor with a dimension of $3 \times 4 \times 9 \times \sum_{l=0}^{2} (2^l \times 2^l) = 2268$. 
In order to evaluate the performance of the proposed method, the experiments are first conducted using a different number of training samples. This method is compared with other methods on the three subsets, and the overall accuracies are also provided for each test. As shown in Table 3.2, the proposed DMHT-PHOG descriptor yields results comparable with the state-of-the-art methods on all tests. The bag of 3D points [75] and DMM-HOG [165] are the methods proposed for action recognition based on the original depth data, while histograms of 3D joints [160] and EigenJoints [163] rely on the estimation of joint positions. The results reflect the robustness of the proposed method and demonstrate that the DMHT-PHOG can encode distinctive features of human gestures. In particular, the proposed method outperforms methods [75], [160], and [163] by 7%∼15% on CST, and achieves comparable performance with [165]. Note that AS1 and AS2 contain samples within small inter-class variations, while AS3 contains samples with complex gestures which usually have large intra-class variations. Therefore, the proposed method is not only of strong discriminative capability when the inter-class variations are small, but also of good robustness when the intra-class variations are large.

Table 3.2 Recognition accuracy (%) comparison of the DMHT-PHOG with other methods on three subsets of MSR Action3D dataset.

<table>
<thead>
<tr>
<th>Tests</th>
<th>[75]</th>
<th>[160]</th>
<th>[163]</th>
<th>[165]</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1 AS1</td>
<td>89.50</td>
<td>98.47</td>
<td>94.70</td>
<td>97.30</td>
<td>96.56</td>
</tr>
<tr>
<td>AS2</td>
<td>89.00</td>
<td>96.67</td>
<td>95.40</td>
<td>92.20</td>
<td>94.77</td>
</tr>
<tr>
<td>AS3</td>
<td>96.30</td>
<td>93.47</td>
<td>97.30</td>
<td>98.00</td>
<td>97.30</td>
</tr>
<tr>
<td>Overall</td>
<td>91.60</td>
<td>96.20</td>
<td>95.80</td>
<td>95.83</td>
<td>96.21</td>
</tr>
<tr>
<td>T2 AS1</td>
<td>93.40</td>
<td>98.61</td>
<td>97.30</td>
<td>98.70</td>
<td>97.26</td>
</tr>
<tr>
<td>AS2</td>
<td>92.90</td>
<td>97.92</td>
<td>98.70</td>
<td>94.70</td>
<td>98.72</td>
</tr>
<tr>
<td>AS3</td>
<td>96.30</td>
<td>94.93</td>
<td>97.30</td>
<td>98.70</td>
<td>97.30</td>
</tr>
<tr>
<td>Overall</td>
<td>94.20</td>
<td>97.15</td>
<td>97.77</td>
<td>97.37</td>
<td>97.76</td>
</tr>
<tr>
<td>CST AS1</td>
<td>72.90</td>
<td>87.98</td>
<td>74.50</td>
<td>96.20</td>
<td>91.43</td>
</tr>
<tr>
<td>AS2</td>
<td>71.90</td>
<td>85.48</td>
<td>76.10</td>
<td>84.10</td>
<td>83.92</td>
</tr>
<tr>
<td>AS3</td>
<td>79.20</td>
<td>63.46</td>
<td>96.40</td>
<td>94.60</td>
<td>95.50</td>
</tr>
<tr>
<td>Overall</td>
<td>74.70</td>
<td>78.97</td>
<td>82.33</td>
<td>91.63</td>
<td>90.28</td>
</tr>
</tbody>
</table>
The proposed method is then compared with other methods which have previously been widely used for action recognition (such as Recurrent Neural Network [95], Dynamic Temporal Warping [102], and Hidden Markov Model [88]) on CST. CST is more challenging because of the considerable variations in actions performed by different subjects. Cross subjects generate much larger intra-class variance than non-cross subjects.

Table 3.3 Recognition accuracy (%) comparison of the DMHT-PHOG with other methods on the cross subject test of MSR Action3D dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recurrent Neural Network [95]</td>
<td>42.50</td>
</tr>
<tr>
<td>Dynamic Temporal Warping [102]</td>
<td>54.00</td>
</tr>
<tr>
<td>Hidden Markov Model [88]</td>
<td>63.00</td>
</tr>
<tr>
<td>Bag of 3D Points [75]</td>
<td>74.70</td>
</tr>
<tr>
<td>Histogram of 3D Joints [160]</td>
<td>79.00</td>
</tr>
<tr>
<td>Eigenjoints [163]</td>
<td>82.30</td>
</tr>
<tr>
<td>STOP Feature [142]</td>
<td>84.80</td>
</tr>
<tr>
<td>Random Occupancy Pattern [143]</td>
<td>86.20</td>
</tr>
<tr>
<td>Actionlet Ensemble [145]</td>
<td>88.20</td>
</tr>
<tr>
<td>DMM-HOG [165]</td>
<td>91.63</td>
</tr>
<tr>
<td>DMHT-PHOG</td>
<td>90.28</td>
</tr>
</tbody>
</table>

Table 3.3 shows the experimental results by various methods. The proposed method achieves a relatively high recognition accuracy of 90.28% on CST. Note that the Actionlet Ensemble [145] required a feature selection process from 3D joint features and a multiple kernel learning process based on the SVM classifier to achieve the accuracy of 88.2%, whereas the proposed method is based on the original depth data without relying on the estimation of 3D joint positions. Overall, considering the large intra-class variations in this dataset, the proposed framework is quite robust.

Furthermore, confusion matrices of the method on CST are shown in Fig. 3.9. Actions with high similarity could produce relatively low accuracies. In AS1, “high throw” is confused with “pick up & throw”, because these two actions are performed with similar arm movements. In AS2, “draw x”, “draw tick”, and “draw circle” are confused with each other, as they have highly similar movements. As actions in AS3
are significantly different, the recognition results are better than those in the other subsets.

### 3.6 Summary

In this chapter, a framework based on the proposed DMHT-PHOG descriptor is presented for gesture recognition in depth sequences. The proposed DMHT-PHOG better represents human gestures in a compact and discriminative way using only three planes instead of the whole gesture sequence. The DMHT is able to capture temporal motion information from the front, top, and side views. To encode the feature from the DMHT, PHOG is extracted from the DMHT in different degrees of detail according to the selected pyramid levels. The proposed DMHT-PHOG descriptor requires no edge or regions of interest extraction, which is often necessary in other methods. The experimental results from the MSR Action3D dataset demonstrate the effectiveness and robustness of the proposed DMHT-PHOG descriptor. The proposed method obtains relatively high recognition accuracies within different experimental settings, which is comparable to current state-of-the-art methods.
Fig. 3.9 Confusion matrices for CST of the MSR Action3D dataset using the DMHT-PHOG.
Spatio-Temporal Pyramid Cuboid Matching for Gesture Recognition Using Depth Maps

4.1 Introduction

Encouraged by the success of template matching methods for 2D video sequences, many gesture recognition approaches using depth maps have been proposed to transform the problem from 3D to 2D. Typically, the depth map of each frame is projected onto three orthogonal planes, and then recognition is performed on the projected 2D planes. In Chapter 3, a framework based on the DMHT-PHOG descriptor was proposed to perform human gesture recognition on depth sequences. In [165] and [20], human gestures were recognized using depth motion maps (DMMs) developed to capture the aggregated temporal motion energies. Meanwhile, Jetley and Cuzzolin [56] exploited both motion history templates (MHTs) and binary shape templates (BSTs) to recognize actions from projected planes that were temporally overlapped. Obviously, these methods are able to reduce the computational complexity, but the inherent 3D information could be lost when depth maps are projected onto 2D planes, which may limit the success of gesture recognition. To address this issue, this chapter presents a gesture recognition framework that is based on the depth motion history template (DMHT) and cuboid fusion scheme.

Fig. 4.1 shows the framework of the proposed approach, spatio-temporal pyramid cuboid matching (STPCM), which consists of two main stages: the training and classification stages. The goals of the training stage are to generate three-channel
dictionaries from hierarchical DMHT and to learn a gesture model. Specifically, training depth gestures are represented using hierarchical DMHT with various temporal strides to exploit speed invariance (see Section 4.2 for details). Three-channel dictionaries are learned from all the projected planes of hierarchical DMHT (see Section 4.3 for details). The gesture model is based on linear SVM supported by combined spatio-temporal features from the cuboid fusion scheme. In the classification stage, each frame of the unknown depth gesture sequence is projected onto three planes that present the front, top and side views. The whole gesture sequence is then partitioned into pyramidal sub-volumes and represented using pyramid DMHT that encodes the multi-scale temporal motion and shape information in 3D. Consequently 3D motion information is captured in different stages of performance. In order to enrich the information provided by pyramid DMHT, each projected plane from pyramid DMHT is subdivided into pyramid spatio-temporal grids, in which feature representation is performed. Then, a novel technique named cuboid fusion is introduced to group spatial-dependent grids from projected planes into pyramid cuboids to enhance the discriminant information. Consequently, a spatio-temporal pyramid cuboid representation of the gesture sequence is generated for the learned gesture model (see Section 4.4 for details). Finally the framework is evaluated following the standard experimental settings on the four public benchmark datasets: MSR Action3D [75], MSR Gesture3D [143], MSR Action Pairs [111] and ChaLearn multi-modal [33] datasets. The experimental results demonstrate that the proposed method is able to achieve better recognition accuracy than the state-of-the-art depth map-based methods.

4.2 Hierarchical DMHT

The problem of representing gesture sequences comprises multiple challenges as discussed in Section 1.2.1. One of the most important challenges so far has
4.2 Hierarchical DMHT

Training Depth Gestures

Three-channel Dictionaries

Gesture Model
Cuboid Fusion

Pyramid DMHT

Hierarchical DMHT

Unknown Depth Gesture
Predication

Training Classification

model learning

sparse coding

Fig. 4.1 Illustration of the spatio-temporal pyramid cuboid matching framework.
undoubtedly been the speed variation related to temporal variability caused by the speed of human movements. In order to effectively solve the problem of speed variation, this work extends the DMHT by using multiple temporal strides to generate hierarchical DMHT. In the training stage, each gesture sequence is represented by hierarchical DMHT through sampling various numbers of frames, as illustrated in Fig. 4.2.

```
1 2 3 4 5 T
1 3 5
1 w
12 w
(1 ( 1))w n T
(1 2( 1))nT
…
…
…
Temporal stride w
…
1
2
w
Sampled frame n
```

Fig. 4.2 Illustration of hierarchical DMHT.

Based on the proposed DMHT, for a depth map sequence with $T$ frames, hierarchical DMHT $H_{v,w}(P_v,t)$ is obtained by using various temporal strides $w$, where $t$ denotes the frame index of the whole depth sequences and $t = 1 + w(n - 1)$; $n$ represents the number of sampled frames when using corresponding temporal stride $w(w \geq 1)$. Through this hierarchical process based on the DMHT, more training samples with different performance speeds can be generated. Fig. 4.3 shows an example of front views of hierarchical DMHT generated from a gesture using five different temporal strides. It is evident that hierarchical DMHTs with different temporal strides encode the motion information with various performance speeds.
4.3 Depth Feature Extraction and Sparse Coding

The hierarchical DMHT represents gesture sequences of different speeds by using different sampled frames. Therefore, hierarchical DMHT is able to represent a depth gesture as a set of DMHTs using various temporal strides in order to cope with speed variation.

![Front views of hierarchical DMHT](image)

Fig. 4.3 Front views of hierarchical DMHT.

4.3 Depth Feature Extraction and Sparse Coding

Multiple descriptors are usually adopted to represent a local region, each of which corresponds to a certain aspect of visual patterns such as static appearance, motion, and motion boundary. To describe the image patterns, several hand-crafted features have been designed such as HOG [28], HOF [69], and MBH [29]. As described in Section 4.2, the proposed hierarchical DMHT is able to encode the motion history information in 3D. Thus only appearance-based features (HOG) are used in this work to capture the dynamic appearance in hierarchical DMHT, which reduces the computational complexity. HOG features are densely extracted from three projected planes of each temporal template in hierarchical DMHT. To elaborate, HOG features are extracted from $32 \times 32$ pixel patches, which are densely sampled from each projected plane on a dense grid with the step size of 16 pixels. For HOG feature extraction, each block has $2 \times 2$ cells, and the size of each cell is set to $8 \times 8$ pixels. The gradient vector for each pixel in the cell is calculated and the orientations of the gradient vectors are stored in a nine-bin histogram ranging from 0 to 180 degrees.
Consequently, each patch is represented by a 144-dimensional HOG feature vector. In this way, three sets of depth features are obtained from the front, top and side views. Then, this work proposes to use dictionary learning to generate three-channel dictionaries for the extracted depth features.

Compared to the codebook used in bag of visual words (BoVW) modeling, the learned overcomplete basis set in sparse coding is referred to as the “dictionary”, consisting of a set of representative vectors learned from a large number of features. The representative vectors are referred to as the “codewords” in BoVW or “atoms”. A considerable amount of quantization error is incurred by the approximation process by which each sample vector is assigned the nearest codeword of BoVW. This can be greatly improved by sparse coding, which allows a linear and sparse combination of atoms to be used in the approximation process. The calculated sparse codes for one feature represent the coefficients in the linear combination of atoms. Next, the sparse coding process is illustrated using the features extracted from the plane \( v \) \((v \in \{f, t, s\})\) of the hierarchical DMHT. Let \( X_v = [x_{v1}, \ldots, x_{vi}, \ldots, x_{vn}] \in \mathbb{R}^{m \times n_v} \) be a set of extracted depth features in \( \mathbb{R}^m \), where \( x_{vj}^v \) is the \( i \)-th HOG feature vector in the plane \( v \). Let \( A_v = [\alpha_{v1}, \ldots, \alpha_{vi}, \ldots, \alpha_{vnv}] \in \mathbb{R}^{p_v \times n_v} \) be the sparse codes for the corresponding feature vectors. The approximation process can be represented by sparse coding [162]:

\[
\{A_v, D_v\} = \underset{A_v, D_v}{\arg\min} \left\{ \sum_{i=1}^{n_v} \| x_{vi} - D_v \alpha_{vi} \|_2^2 + \lambda \| \alpha_{vi} \|_1 \right\} \\
\text{s.t.} \quad \| d_{vj}^v \|_2 \leq 1, \quad \text{for } \forall j = 1, \ldots, p_v
\] (4.1)

where \( D_v = [d_{v1}, \ldots, d_{vj}, \ldots, d_{vp_v}] \in \mathbb{R}^{m \times p_v} \) is the learned overcomplete dictionary with \( p_v \) atoms, and \( \alpha_{vi}^v \) is the calculated sparse code for feature vector \( x_{vi}^v \). The unite \( \ell_2 \)-norm constraint on \( d_{vj}^v \) is typically applied to avoid trivial solutions [162]. Note that the \( \ell_1 \) regularization on \( \alpha_{vi}^v \) enforces \( \alpha_{vi}^v \) to have a small number of nonzero elements.
other words, each feature vector is approximated by a linear and sparse combination
of atoms in the learned dictionary.

To solve Eq. (4.1), two steps are involved by iteratively optimizing $D_v$ or $A_v$ while
fixing the other. By fixing $D_v$, the optimization can be solved by:

$$\min_{\alpha_i^v} \|x_i^v - D_v \alpha_i^v\|_2^2 + \lambda \|\alpha_i^v\|_1$$  \hspace{1cm} (4.2)

This is known as Lasso [134] in the statistical literature and can be efficiently
solved by the feature-sign search algorithm [71]. By fixing $A_v$, the objective function
becomes a least square problem with quadratic constraints:

$$\min_{D_v} \|X_v - D_v A_v\|_F^2$$

s.t. $\|d_j^v\|_2 \leq 1$, for $\forall j = 1, \ldots, p_v$  \hspace{1cm} (4.3)

The Lagrange dual can be used to solve the optimization problem [71].

In sparse coding, the dictionary $D_v$ is initialized by employing the K-SVD
algorithm [4], and then learned in the training phase while collecting a large number
of features from training samples by iteratively optimizing Eqs. (4.2) and (4.3). In
the coding phase, the sparse codes are obtained by optimizing Eq. (4.2) given learned
$D_v$.

### 4.4 Spatio-Temporal Pyramid Cuboid Matching

In order to encode the multi-scale temporal motion and shape information from
gestures for model learning and classification, the whole gesture sequence is parti-
tioned into pyramidal sub-volumes and represented using pyramid DMHT. Three-D
motion gesture information is therefore captured in different temporal durations.
Furthermore, the cuboid fusion scheme is hypothesized to enable spatial-dependent
grids from projected planes of pyramid DMHT to be grouped into cuboids to enhance
the discriminant information for model learning. In this way, a spatio-temporal pyramid cuboid representation of the gesture sequence is generated for the learned gesture model. This section details the procedure of spatio-temporal pyramid cuboid matching, including pyramid DMHT and cuboid fusion.

### 4.4.1 Pyramid DMHT

The temporal pyramid was introduced by Laptev et al. [69] to consider the rough temporal order of a video. It was also used in depth sequences [111, 145] to take into account the temporal context. In those papers, a set of temporal segments were repeatedly and evenly subdivided from a video sequence, and descriptor-level statistics were pooled from the temporal segments. However different people could have varied motion patterns when performing the same gesture, it shows inflexibility in dealing with the variance, as a video sequence is evenly subdivided in a single scale. This work aims to partition each projected sequence into multi-scale sub-volumes, each capturing various short motion stages with respect to the depth gesture.

Eq. (3.2) is rewritten in a more general way by introducing temporal duration \([t_1, t_2]\].

\[
D_v(P_v, t) = |I_v(P_v, t) - I_v(P_v, t - 1)|
\]

is defined to calculate the differences between the \(t\)-th \((t \in [t_1, t_2], 1 < t_1, t_2 \leq T)\) frame and the previous frame, where \(t_1\) and \(t_2\) are the start and finish time of the corresponding sub-volume, and \(T\) is the length of a depth sequence. A single DMHT can be obtained in a recursive manner

\[
M_v(P_v, t) = \begin{cases} 
 t_2 - t_1 + 1 & \text{if } D_v(P_v, t) > \xi \\
 \max(0, M_v(P_v, t - 1) - \sigma) & \text{otherwise}
\end{cases}
\]

where \(\xi\) and \(\sigma\) are the values of threshold for motion detection and decay parameter, respectively. Then, let a sequence of sub-volumes at scales \(0, \ldots, s, \ldots S\) be constructed such that there are \(2^s\) non-overlapping sub-volumes at the \(s\)-th scale partition. Accordingly, based on Eq. (4.4) DMHTs \(\tilde{M}_{v,s}\) at the scale \(s\) within time period
$[t_1', t_2']$ can be represented as the following equation

\[
\tilde{M}_{v,s} = \begin{cases} 
\tilde{M}_v(P_v, t_2') & \quad t_1' = (i-1)\frac{T}{2^s} + 1, \\
\tilde{M}_v(t_2') & \quad t_2' = i\frac{T}{2^s} 
\end{cases} 
\]

where \(i\) is defined as the index of sub-volume at the scale \(s\). \(t_1'\) and \(t_2'\) are the start and finish time of the corresponding sub-volume, which can be calculated using \(i\) and \(s\).

Thus, the final pyramid DMHT is a set of all the DMHTs generated at pyramid temporal scales \(\{\tilde{M}_{v,0}, \ldots, \tilde{M}_{v,s}, \ldots, \tilde{M}_{v,S}\}\). As shown in Fig. 4.4, a three-scale \((S = 2)\) temporal pyramid is used to partition a depth sequence to generate the corresponding DMHT. The pyramid DMHT provides several advantages including that it well characterizes the motion and shape information of the multi-scale 3D action in the temporal direction. It is also a compact and informative representation of human gestures, and it can handle complex gestures due to its pyramid representation.

![Fig. 4.4 Illustration of pyramid DMHT. Left: a depth sequence that is partitioned into sub-volumes using a three-scale temporal pyramid. Right: the pyramid DMHT with respect to the left sub-volumes.](image-url)
4.4.2 Cuboid Fusion

Based on the features extracted from pyramid DMHT, this work proposes STPCM as the feature representation scheme. Samples are initially collected from the training dataset to learn the dictionary $D_v$ for each projected plane $v$ from hierarchical DMHT. Next, the sparse codes are obtained by solving Eq. (4.2) for every feature vector in each DMHT. After that, each feature vector can be represented by $p_v$-dimensional sparse codes. Since the spatial pyramid matching (SPM) [70] is widely applied in 2D image classification and shows promising results, it is proposed that each DMHT is subdivided into spatial pyramid cuboids, with the final representation then constructed using the proposed cuboid fusion scheme. Specifically, each view of DMHT is divided into $2^l$ by $2^l$ grids at the $l$-th level ($l = 0, \ldots, L$). Fig. 4.5 shows an example of a pyramid with three levels. From the coarser to the finer level of the pyramid, the 3D temporal template is recursively divided in a half. Let $S_{kl}$ be defined by the indices of sparse codes in $\{1, \ldots, n\}$ falling into grid number $k$ at level $l$. Then, a pooling function shall be applied to the sparse codes falling in each grid of the projected plane $v$ to yield the histogram representation $z^{vl}(k)$ separately. Note that different pooling functions create different image statistics. As shown in [162], max pooling produces better performance than other pooling methods. The max pooling procedure is well established by biophysical evidence in the visual cortex [122] and is empirically justified by many applications in image categorization. Therefore, this work adopts the max pooling function:

$$z_{ji}^{vl}(k) = \max_{i \in S_{kl}} \left| \alpha_{ji}^v \right|$$

(4.6)

where $z_{ji}^{vl}(k)$ is the $j$-th element of $z^{vl}(k) \in \mathbb{R}^{p_v}$, and $\alpha_{ji}^v$ is the matrix element at the $j$-th row and the $i$-th column of $A_v$. Then pooled feature $z^{vl}(k)$ is further normalized by $\ell_2$-normalization to form the grid representation $g^{vl}(k)$. 
4.4 Spatio-Temporal Pyramid Cuboid Matching

Fig. 4.5 Illustration of spatial pyramid cuboids. Depth features on each plane can be represented by sparse codes. Circles, triangles and squares denote sparse codes on front plane, top plane, and side plane, respectively. Multi-level spatial dependent sparse codes construct pyramid cuboids.

In order to generate the final representation of the temporal template, feature fusion is always used to combine grid representations from projected planes. Normal fusion is usually employed in this case, which easily concatenates features of every grid from the projected planes, as shown in Fig. 4.6(a). However, the spatial dependent information could be lost, which causes degradation of discriminative ability. To preserve 3D spatial information, a new cuboid fusion scheme is proposed in order to combine the spatial-dependent features from the grids on three planes, as shown in Fig. 4.6(b). Obviously, the number of cuboids at level \(l\) is \(2^l \times 2^l \times 2^l\), instead of \(3 \times 2^l \times 2^l\) grids using normal fusion. For example, the number of cuboids is 8 \((2^1 \times 2^1 \times 2^1)\) at the level 1 \((l = 1)\), which is illustrated in Fig. 4.6(b). For the normal fusion at level 1, the number of grids is 12 \((3 \times 2^1 \times 2^1)\), as shown in Fig. 4.6(a). In the case where only the level \(l = 0\) is used, the cuboid fusion is reduced to normal fusion scheme.

Fig. 4.6 Comparison between normal fusion and cuboid fusion at the spatial level \(l = 1\). Normal fusion simply concatenates representation of every grid. Cuboid fusion combines the spatial-dependent grid representation to construct cuboids.
For mathematical brevity, the generated grid representation $g_{vl}(k)$ can be rewritten as $g_{vl}(i, j)$ in a form of two-dimensional array, where $i, j$ are the indices of the grid and $k = (i - 1)2^l + j$. Note the index starts at the cuboid located at the origin point in the coordinate system, as the highlighted cuboid of level $l = 1$ shown in Fig. 4.5. Thus, the representation of DMHT on plane $v$ at level $l$ can be represented as:

$$
G_{vl} = \begin{pmatrix}
g_{vl}(1, 1) & \cdots & g_{vl}(1, 2^l) \\
\vdots & \ddots & \vdots \\
g_{vl}(2^l, 1) & \cdots & g_{vl}(2^l, 2^l)
\end{pmatrix} \quad (4.7)
$$

With the aim of obtaining a compact and complete representation for pyramid DMHT, the related grid representations from three planes are combined to construct a final cuboid representation, as shown in Fig. 4.6(b). Finally, cuboid representations from all the DMHTs are then concatenated as the depth sequence representation for the final recognition. The outline of the cuboid fusion algorithm is summarized in Algorithm 1. Therefore, the resulting spatio-temporal cuboid representation $C$ of the depth sequence is obtained by concatenation of all the multi-level cuboid representation. Compared to normal fusion, the cuboid fusion scheme takes into account the 3D spatial relationship of neighbors, which provides a stronger discriminative ability to capture the 3D motion context of temporal templates.

As spatial pyramid cuboids are constructed from the proposed temporal templates, i.e., pyramid DMHT, a depth sequence is represented using spatio-temporal pyramid cuboids. The proposed STPCM represents depth gestures using pyramid DMHT, maintaining the multi-scale 3D motion and shape information in the temporal direction. Feature fusion from projected planes using cuboid fusion scheme preserves 3D spatial locations. STPCM simultaneously captures the temporal and spatial context of depth gesture in a pyramid way, which is much more flexible and rigorous in dealing with complex gesture recognition problems.
Algorithm 1 Cuboid fusion

Input: $G^{fl}$, $G^{tl}$, $G^{sl}$
Output: cuboid representation $C_l$

1: $m \leftarrow 0$
2: for $i = 1$ to $2^l$ do
3:     for $j = 1$ to $2^l$ do
4:         for $k = 1$ to $2^l$ do
5:             $m \leftarrow m + 1$
6:             $c_m \leftarrow [g^{fl}(i,j); g^{tl}(k,j); g^{sl}(i,k)]$
7:         end for
8:     end for
9: end for
10: $C_l \leftarrow [c_1; \ldots; c_{2^l \times 2^l \times 2^l}]$

4.5 Experiments

The proposed framework of STPCM is evaluated on four public datasets: MSR Action3D [75], MSR Gesture3D [143], MSR Action Pairs [111], and ChaLearn multi-modal dataset [33]. A series of experiments are carried out to investigate the performance of the proposed framework. The method is compared with the existing approaches and extensively evaluated under different settings. Two feature fusion schemes are used in STPCM to generate the final representation of depth gestures. One is the normal fusion that concatenates all the representations of the grids from three planes. The other is the proposed cuboid fusion, which groups the spatial dependent grid representations as cuboids to preserve the spatial information.

For DMHT generation in Eq. (4.4), the value of threshold $\xi$ is set at 50, and the value of decay parameter $\sigma$ is set at 1. Extensive experiments are conducted for the evaluation of parameters, i.e., temporal strides for generating hierarchical DMHT and temporal scales and spatial levels in spatio-temporal pyramids for generating pyramid DMHT. To speed up the process of training and testing, linear SVM is used for classification, as introduced in Section 3.4. The proposed method is compared with the existing depth-based approaches. For fair comparison, the methods only based on skeleton data are not included in the experiments. In all experiments, the
proposed approach consistently outperforms the state-of-the-art depth map-based methods.

4.5.1 MSR Action3D Dataset

The size of each dictionary learned from the projected planes of hierarchical DMHT is empirically set to 500. For a fair comparison, the same experimental parameters as described in [75] are followed. Five subjects are used for training, and the remaining five subjects are used for testing; known as a cross-subject test. This presents a challenge, as subjects are free to perform actions in their own manner or style.

Table 4.1 Recognition accuracy(%) comparison of STPCM with other methods on the MSR Action3D dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bag of 3D points [75]</td>
<td>74.70</td>
</tr>
<tr>
<td>STOP [142]</td>
<td>84.80</td>
</tr>
<tr>
<td>ROP [143]</td>
<td>86.20</td>
</tr>
<tr>
<td>HON4D [111]</td>
<td>88.89</td>
</tr>
<tr>
<td>DSTIP [159]</td>
<td>89.30</td>
</tr>
<tr>
<td>DMHT-PHOG (Chapter 3)</td>
<td>90.30</td>
</tr>
<tr>
<td>DMM-HOG [165]</td>
<td>91.70</td>
</tr>
<tr>
<td>STPCM (normal fusion)</td>
<td>92.45</td>
</tr>
<tr>
<td>STPCM (cuboid fusion)</td>
<td>94.26</td>
</tr>
</tbody>
</table>

The method is compared with the state-of-the-art methods using the MSR Action3D dataset. The comparison of recognition accuracy is shown in Table 4.1. The bag of 3D points method [75] encodes the 3D body shape information using depth data, with the recognition accuracy only 74.70%. This is probably due to the small number of subjects and also the significant variations in the same action performed by different subjects. STOP [142] leverages the spatial and temporal contextual information while allowing for intra-class variations, and achieves an accuracy of 84.80%. Wang et al. [143] propose a 3D depth feature called ROP to deal with the noise and occlusion problems and obtain an accuracy of 86.20%. As depth data can provide 3D information, some methods utilize the property for more
robust action recognition. The recently proposed depth-based action recognition methods, including HON4D [111], DSTIP [159], and DMM-HOG [165], achieve accuracies of 88.89%, 89.30%, and 91.70%, respectively. The proposed STPCM with cuboid fusion obtains higher accuracy (94.26%) than other methods, with the accuracy decreasing to 92.45% if normal fusion is used. The proposed cuboid fusion scheme combines the spatial-dependent representations from the grids on three planes, enabling the 3D spatial information to be well preserved.

Another advantage of the proposed method is that it is unaffected by the number of gesture repetitions in gesture sequences. In order to evaluate this advantage, all the test gesture sequences are manually replicated twice and four times, and the proposed approach is applied to the new test samples. The recognition accuracies are 93.66% and 89.43% for two repetitions and four repetitions, respectively. If half of the test gesture sequences are repeated twice, and the other half of the test gesture sequences are repeated four times, the recognition accuracy is 87.31%. Note that the training gesture sequences are not replicated, which means the same generated dictionaries and learned model as previous experiments are used here for the new evaluation. This experiment shows that the proposed method is relatively insensitive to the number of gesture repetitions.

Moreover, with respect to the confusion matrix shown in Fig. 4.7, the proposed method works very well for most of the gestures. The recognition errors occur if two gestures are too similar to each other, such as “draw tick” and “draw x”.

### 4.5.2 MSR Gesture3D Dataset

In this dataset, the dictionary size is empirically set to 500 for sparse coding. The leave-one-out cross-validation scheme in [143] is used in this experiment. Comparison of performance with other methods is shown in Table 4.2. The proposed method performs significantly better than the action graph model that uses carefully
designed shape features [72]. The performance of STPCM with cuboid fusion obtains state-of-the-art accuracy of 97.80%, which also outperforms the previous depth-based methods, including ROP [143], DMM-HOG [165], HON4D [111] and SNV [17]. This demonstrates the flexibility of the proposed STPCM method, as it is able to recognize hand gestures even if skeleton data is not provided.

Table 4.2 Recognition accuracy(%) comparison of STPCM with other methods on the MSR Gesture3D dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action Graph on Occupancy [72]</td>
<td>80.50</td>
</tr>
<tr>
<td>Action Graph on Silhouette [72]</td>
<td>87.70</td>
</tr>
<tr>
<td>ROP [143]</td>
<td>88.50</td>
</tr>
<tr>
<td>DMM-HOG [165]</td>
<td>89.20</td>
</tr>
<tr>
<td>HON4D [111]</td>
<td>92.45</td>
</tr>
<tr>
<td>SNV [164]</td>
<td>94.74</td>
</tr>
<tr>
<td>STPCM (normal fusion)</td>
<td>95.27</td>
</tr>
<tr>
<td>STPCM (cuboid fusion)</td>
<td>97.80</td>
</tr>
</tbody>
</table>
The confusion matrix of STPCM on the MSR Gesture3D dataset is shown in Fig. 4.8. The proposed method performs reasonably well on most dynamic gestures. The highest level of confusion occurs in recognizing “past”, as some subjects use up and down arm movements when performing “past”, sharing a similar motion to “hungry”.

![Confusion Matrix](image)

Fig. 4.8 Confusion matrix for MSR Gesture3D dataset using STPCM.

### 4.5.3 MSR Action Pairs Dataset

The MSR Action Pairs dataset [111] selects pairs of actions so that within each pair the motion and the shape cues are similar, but their correlations are different. This dataset can be used to investigate how the temporal order affects the action recognition. Six pairs of actions are collected. Each action is performed three times by ten different subjects.

Following the experimental settings set out by Oreifej and Liu [111], the first five subjects in each action class are used for training, with the rest used for testing. The
Table 4.3 Recognition accuracy(%) comparison of STPCM with other methods on the MSR Action Pairs dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMM-HOG [165]</td>
<td>66.11</td>
</tr>
<tr>
<td>HON4D [111]</td>
<td>96.67</td>
</tr>
<tr>
<td>STPCM (normal fusion)</td>
<td>88.57</td>
</tr>
<tr>
<td>STPCM (cuboid fusion)</td>
<td>91.43</td>
</tr>
</tbody>
</table>

sizes of dictionaries are empirically set at 500. The detailed comparison to other approaches is demonstrated in Table 4.3. In DMM-HOG [165], depth sequences are collapsed onto three projected maps where temporal orders are eliminated. As no temporal information is encoded in this method, relatively lower accuracy is obtained. HON4D [111], meanwhile, operates in the 4D space of shape and motion using depth data. It captures motion and geometry cues jointly using a histogram of normal orientation. HON4D obtains the state-of-the-art accuracy of 96.67%, which outperforms the proposed STPCM. This is because of the presence of background (e.g., floor) in the sample frames shown in Fig. 2.5. This dataset is different to the first two datasets where the depth information of the background is also captured, which impacts on the generation of the DMHT.

The confusion matrix of STPCM on the MSR Action Pairs dataset is shown in Fig. 4.9. It can be seen that each pair of similar actions is confused with each other.

### 4.5.4 ChaLearn Multi-Modal Dataset

The ChaLearn multi-modal dataset [33] is based on Italian gestures. This dataset is focused on “multiple instances, user independent learning” of gestures. It is recorded with Kinect, containing RGB videos, depth videos, user mask videos and skeleton data. Fig. 4.10 shows an example of different available visual modalities. There are 20 Italian sign gestures. This dataset is challenging due to the high intra-class variability of gesture samples and low inter-class variability for some gesture categories.
4.5 Experiments

Fig. 4.9 Confusion matrix for MSR Action Pairs dataset using STPCM.

Fig. 4.10 An example from the ChaLearn multi-modal dataset (left to right): RGB video, depth video, user mask video, and skeleton data
The dataset is divided into three parts: training data, validation data, and test data. All the gesture sequences from the training data are used in dictionary learning. Validation data is used for parameter optimization. The sizes of dictionaries are empirically set to 1000. Gestures in the dataset are continuous and some of them are distractor gestures. In this experiment, the truth labels of start and end frames provided by the dataset are used to split continuous gestures into several individual gestures. The proposed STPCM is then performed on these individual gestures.

In order to alleviate the influence of background noise, it is necessary to segment gesture regions from the original depth maps. As user mask videos are available for each gesture sample, it is proposed that the gesture regions are segmented using depth videos and mask videos. Additionally, a median filter is applied to remove the noise in the videos. It is assumed that a binary user mask at frame $t$ is $M_t$, and the corresponding depth map is $D_t$. Gesture regions can then be segmented by aligning these two maps to find their intersection $R_t = M_t \cap D_t$, as shown in Fig. 4.11.

![Fig. 4.11 Gesture regions segmentation.](image)

The ChaLearn multi-modal dataset is collected for multi-modal gesture recognition, and most of the work uses multiple modalities to perform experiments. Comparisons of these work with multi-modal methods will be demonstrated in Chapter 7. Few published studies use only depth maps on this dataset. The experimental results are shown in Table 4.4. Research on 2DMTM [76] suggests that this model represent gesture motion along with static posture in 2D space to encode regional information of human gestures. The depth-based 2DMTM only obtains 76.99% accuracy, as subjects exhibit large variations when performing the same gesture in
this dataset, and the temporal information is not considered in 2DMTM. On the other hand, the proposed STPCM with cuboid fusion obtains 86.56% accuracy — higher than depth-based 2DMTM. Furthermore, with visualization of the confusion matrix illustrated in Fig. 4.12, it can be seen that the highest accuracy is 92%, and the lowest accuracy is 83%.

Table 4.4 Recognition accuracy(%) comparison of STPCM with other methods on the ChaLearn multi-modal dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>2DMTM [76]</td>
<td>76.99</td>
</tr>
<tr>
<td>STPCM (normal fusion)</td>
<td>81.85</td>
</tr>
<tr>
<td>STPCM (cuboid fusion)</td>
<td>87.31</td>
</tr>
</tbody>
</table>

Fig. 4.12 Confusion matrix for ChaLearn multi-modal dataset using STPCM.
4.5.5 Evaluation of Parameters

This subsection evaluates the impacts of parameters in the proposed approach, that are, temporal stride \( w \) for hierarchical DMHT, temporal scale \( S \) and spatial level \( L \) for STPCM. In these experiments, unless otherwise stated, the evaluation for each parameter is carried out one at a time, and the other ones are fixed to the default values, i.e., stride set for hierarchical DMHT \( W = \{1, 2, 3\} \), temporal scale \( S = 2 \), and spatial level \( L = 1 \).

**Temporal stride.** The results of choosing different temporal stride sets are shown in Fig. 4.13. From this representation, it is clear that using multiple temporal strides compared with a single temporal stride is beneficial for all the datasets. The main reason is because hierarchical DMHT with multiple temporal strides has the ability to handle the variation caused by the speed of human movement.

![Fig. 4.13 Evaluation of temporal stride parameter \( w \) in hierarchical DMHT.](image)

<table>
<thead>
<tr>
<th>Temporal Stride Settings</th>
<th>MSR Action 3D</th>
<th>MSR Gesture 3D</th>
<th>MSR Action Pairs</th>
<th>ChaLearn multi-modal</th>
</tr>
</thead>
<tbody>
<tr>
<td>( W_1 = {1} )</td>
<td>90.09009009</td>
<td>93.69085174</td>
<td>88</td>
<td>86.30098453</td>
</tr>
<tr>
<td>( W_2 = {1, 2} )</td>
<td>93.95770393</td>
<td>95.58359621</td>
<td>88.57142857</td>
<td>86.75105485</td>
</tr>
<tr>
<td>( W_3 = {1, 2, 3} )</td>
<td>94.26</td>
<td>97.80</td>
<td>89.71428571</td>
<td>87.0323488</td>
</tr>
<tr>
<td>( W_4 = {1, 2, 3, 4} )</td>
<td>94.26</td>
<td>97.80</td>
<td>91.42857143</td>
<td>87.31364276</td>
</tr>
<tr>
<td>( W_5 = {1, 2, 3, 4, 5} )</td>
<td>94.26</td>
<td>97.80</td>
<td>91.42857143</td>
<td>87.31364276</td>
</tr>
</tbody>
</table>

**Temporal scale and spatial level.** The temporal scale and spatial level are also evaluated on the four datasets with default temporal stride sets \( \{1, 2, 3\} \). For both temporal scale \( S \) and spatial level \( L \), lower values obtain fewer cuboids in STPCM, which negatively affects the recognition performance, as illustrated in Fig. 4.14. For example, when \( S = 2 \) and \( L = 1 \), there are \( \sum_{x=0}^{2} 2^x = 7 \) pyramid temporal templates in pyramid DMHT, each containing \( \sum_{l=0}^{1} (2^l \times 2^l \times 2^l) = 9 \) pyramid cuboids. These
4.6 Summary

In this chapter, a framework based on spatio-temporal pyramid cuboid matching (STPCM) has been proposed for gesture recognition. Based on the DMHT, the extended hierarchical DMHT is able to handle speed variation by using multiple temporal strides. The pyramid DMHT jointly encodes 3D motion and shape cues in pyramid temporal scales. A new cuboid fusion scheme is proposed by grouping spatially dependent grids from projected planes of pyramid DMHT to construct spatio-temporal pyramid cuboids without loss of discriminant spatial information. Based on pyramid DMHT and cuboid fusion, the STPCM framework is then designed.
to recognize human gestures from depth sequences. The proposed STPCM retains the 3D locations and motion information in both spatial and temporal directions. It has been extensively evaluated using four public benchmark datasets, and has shown its effectiveness and robustness.
Specificity and Latent Correlation Learning for Multi-View Gesture Recognition

5.1 Introduction

Human gestures are performed in real 3D environments; however, traditional cameras only capture the 2D projection of a scene. As a result, projection of gestures depends on the viewpoint. As visual appearances vary from different perspectives, a single view does not usually provide information sufficiently comprehensive to describe gestures. Compared with single-view representations, multiple views encode more information for human gesture recognition [166, 57, 39]. In single-view systems, the change of observation view and potential occlusions increase the difficulties of gesture recognition. In contrast, multi-view data can provide complementary information to overcome the limitations in single-view systems. Over the past decades, multi-view gesture recognition has been a hot research topic, and there have been many remarkable research achievements. In multi-view gesture recognition, it is difficult to discover the latent correlation between multiple views [42]. According to gesture-capture system settings, the state-of-the-art methods can be roughly divided into two categories. The first category is to perform gesture recognition based on multi-view data recorded by the multi-camera system. The second category is to synthesize multi-view data from a single depth sensor for gesture recognition.

Methods in the first category are required to construct 3D model based on multi-view system to perform gesture recognition. To overcome the difficulties in
correlation discovery between multiple views, Yilmaz and Shah [170] employed epipolar geometry for point correspondence between actions to impose fundamental matrix constraints for view-invariant action recognition. Rao et al. [119] showed that the maxima in space-time curvature of a 3D trajectory were preserved in 2D image trajectories, and therefore the 2D trajectories could be utilized to capture the view-independent representation of human actions. Li et al. [74] reconstructed a 3D model from multi-view inputs for action recognition. Another method implemented 3D reconstruction to take advantage of multi-view information [40]. Gao et al. [41] proposed an effective method for view selection that facilitated discriminative feature representation and model learning. Lv and Nevatia [89] constructed the action net model to represent spatial shapes of human action. Except for explicitly utilizing view-based knowledge for gesture representation, researchers pay more attention to view-specific feature learning, and then study their intrinsic relationships by view-specific feature transformation. Farhadi et al. [35] proposed to extract features from individual views, and then to utilize maximum margin clustering to generate split-based features and a predictor in one view. For the other view, split-based feature for action recognition was obtained by the learned predictor. Liu et al. [81] put forward bipartie graph partitioning to classify the bag-of-words representation of two views into visual-word clusters, and the view-specific features were transformed into the view-independent representation. In addition, some multi-view DL methods have been presented for gesture recognition. Gao et al. [42] explored the complementary properties between multiple views based on DL framework, and the latent common knowledge was discovered by joint learning. Jin et al. [58] designed the uncorrelated constraint for multi-view DL to reduce the redundancy among dictionaries learned from multiple views. The multi-camera systems have some drawbacks including: they require costly complicated setup of calibration; they contain high complexity and expensive computation which are not suitable for real-time application; they require frame synchronization.
In the second category, multiple views are synthesized from depth maps in the form of projected planes. In order to leverage depth information and reduce computational cost, many approaches have been proposed to transform the problem from 3D to 2D by projecting depth map onto predefined planes, as discussed in Section 2.2.1. Inspired by the three-view multi-view drawing in engineering and technology, in Chapters 3 and 4, depth frames were projected onto three orthogonal planes with the aim of providing additional information from different views. In order to preserve the 3D spatial information from the projected planes, a cuboid fusion scheme was proposed in Chapter 4. Furthermore, the STPCM framework based on the cuboid fusion scheme showed its effectiveness. However, it is still challenging to discover the latent correlation between multiple views when an arbitrary number of views is provided. For example, three views are enough to capture the shape and appearance information of simple gestures, but more views might be required to ensure sufficient description of more complex gestures. To solve this problem, this work further extends the previous methods, and focuses on multi-view gesture recognition in the second category, where multiple views are synthesized from depth maps. The work is different from the previous work in that, instead of only using three projected planes, it addresses gesture recognition in more general cases when an arbitrary number of projected planes are obtained from depth maps.

Dictionary learning (DL), as a particular sparse coding model, aims to learn a set of atoms that can be linearly combined to well approximate a given feature vector. Most DL methods have been addressed to solve single or two views based DL problems [80, 37, 90, 73]. Recently, multi-view DL has attracted a lot of research interests, because there exists more useful information for recognition in multiple views than that in a single view [50, 48, 58]. However, most multi-view DL methods mainly focus on the reconstruction accuracy, whereas the latent information between multiple views have not been investigated comprehensively and thoroughly. Motivated by this, different to the aforementioned methods that explicitly utilize
view-based knowledge, the focus of this work is on the gesture recognition based on DL framework using the synthetic multi-view data from depth maps. Within the general DL framework, a novel method is proposed whereby the view-specific dictionary and the latent dictionary among multiple views are learned for gesture recognition. The proposed method is able to learn a view-specific sub-dictionary for each view. The sub-dictionary for each view is called specificity in this work. Under empirical observation, the latent information between multiple views should be taken into consideration for better recognition. In this work, the latent information is encoded in a learned latent dictionary, named latent correlation. The specificity captures the most discriminative features of each view, while the latent correlation contributes the essential 3D information to all the views. Therefore, if only the sub-dictionaries learned independently from each view are used for classification without the latent information from multiple views, the performance of gesture recognition would be degraded. In this chapter, a novel method — specificity and latent correlation learning (SLCL) — is introduced to learn the specificity and latent correlation for multi-view gesture recognition. With the help of latent correlation between multiple views, the overall dictionary (consisting of specificity and latent correlation) can be more compact and more discriminative for classification.

5.2 Multiple Views from Depth Maps

In Chapter 3, three projected planes from a depth map were shown, representing three views. As multiple views usually provide additional information when representing complex gestures, this section introduces a more general way to obtain multiple views from a depth map.

According to Eq. (3.1), a pixel in a depth map can be converted to a 3D point \((x_w, y_w, z_w)\). The projections onto different planes can be obtained by rotating the calculated 3D point. The rotation of the 3D points can be performed equivalently by
assuming that a virtual camera moves around the subject from different viewpoints as illustrated in Fig. 5.1. If the virtual camera moves from position $p_0$ to $p_2$, the process can be decomposed into the following two steps: firstly it moves from $p_0$ to $p_1$ by an angle $\alpha$ about the $y$ axis, and then it moves from $p_1$ to $p_2$ by an angle $\beta$ about the $x$ axis. The corresponding coordinate $(x_{rw}, y_{rw}, z_{rw})$ of point clouds after rotation can be computed through multiplication by the transformation matrices $R_y(\alpha)$ and $R_x(\beta)$

$$
[x_{rw}, y_{rw}, z_{rw}, 1]^T = R_x(\beta)R_y(\alpha)[x_w, y_w, z_w, 1]^T
$$

(5.1)

where $R_y(\alpha)$ denotes the rotation about the $y$ axis, and $R_x(\beta)$ denotes the rotation about the $x$ axis. The transformation matrices can be expressed as

$$
R_y(\alpha) = \begin{bmatrix}
\cos \alpha & 0 & \sin \alpha & 0 \\
0 & 1 & 0 & 0 \\
-\sin \alpha & 0 & \cos \alpha & 0 \\
0 & 0 & 0 & 1
\end{bmatrix},
R_x(\beta) = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \cos \beta & -\sin \beta & 0 \\
0 & \sin \beta & \cos \beta & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
$$

(5.2)
The pixel values of the projection image are the number of 3D points with the same projected coordinates. Therefore, an arbitrary number of views can be obtained by rotating the virtual camera.

A depth camera captures 2.5D information because only the 3D structure of the point visible to the sensor is contained in the map; nothing is known about the other side of the object or scene. The rotation has to be within a range such that the projections still provide sufficient spatial information about the gestures. In other words, both the angles of $\alpha$ and $\beta$ have to be limited to a certain range. Fig. 5.2 shows an example of multiple views from a depth map. The depth map of a gesture sequence is at the center of the figure. It can be seen from the figure that each view captures the shape of the body from that viewpoint.

Fig. 5.2 Multiple views from a depth map.
5.3 Specificity and Latent Correlation Learning

For gesture representation, multiple views are used in this work for the DMHT generation instead of the three views used in previous chapters. Based on the STPCM framework in Chapter 4, this work represents training samples using hierarchical DMHT with multiple views. A method is proposed to address the aforementioned problems in Section 5.1, that is, how to learn compact and discriminative sub-dictionaries for each view, and how to learn the latent information from different views. This section elaborates on the proposed *specificity and latent correlation learning (SLCL)* to address the above issues.

The proposed method explicitly learns the latent information (latent correlation) that is represented by a dictionary, as well as the view-specific sub-dictionary (specificity) for each view. To derive the model, the conventional DL framework is first reviewed. Suppose there are \( n \) training samples denoted by \( x_i \in \mathbb{R}^m \), where \( i = 1, \ldots, n \), the DL framework learns a dictionary \( D \in \mathbb{R}^{m \times p} \) from them by alternatively minimizing the following objective function \( f \) over \( D \) and the coefficient matrix \( A = [\alpha_1, \ldots, \alpha_n] \in \mathbb{R}^{p \times n} \):

\[
\{A,D\} = \arg\min_{A,D} \left\{ f \triangleq \sum_{i=1}^{n} \|x_i - D\alpha_i\|_2^2 + \lambda \psi(\alpha_i) \right\}.
\]  

subject to \( \|d_j\|_2 \leq 1 \), for \( \forall j = 1, \ldots, p \) \hspace{1cm} (5.3)

The learned dictionary consists of representative vectors that are referred to as the “atoms”. The constraint on the Euclidean length for all the atoms guarantees the objective function convex, and it always exists for all the objective functions throughout this chapter. Therefore, for brevity it is omitted hereafter. It is worth noting that a similar objective function proposed by Wong [154] only encodes each sample datum over the data themselves and enforces the sparse coefficients non-negative. By comparison, Eq. (5.3) contains two types of variables, the dictionary...
and the sparse coefficients, and does not over-explore the non-negative property of
the coefficient. If $p > m$, then $D$ is called an overcomplete dictionary.

It is observed that the latent information in different views contributes to multi-
view representation, which is essential for reconstruction in DL. For this reason, to
improve the recognition performance, this work aims to explicitly learn the latent
information (latent correlation), which provides the inherent 3D spatial pattern bases
for all views. Suppose there are $V$ views. $X = [X_1, \ldots, X_v, \ldots, X_V] \in \mathbb{R}^{m \times n}$
denotes the features extracted from training gestures, wherein $X_v \in \mathbb{R}^{m \times n_v}$
represents the features of view $v$, and feature $x_i \in \mathbb{R}^m$ from this view is indexed
by $i \in I_v$. Denote the overall dictionary as $D = [D_1, \ldots, D_v, \ldots, D_V, D_{V+1}] \in \mathbb{R}^{m \times p}$,
where $D_v \in \mathbb{R}^{m \times p_v}$ stands for the specificity of the $v$-th view ($p = \sum_{v=1}^{V+1} p_v$), and $D_{V+1} \in \mathbb{R}^{m \times p_{V+1}}$ represents the latent correlation.

First of all, the learned overall dictionary $D$ should well represent every feature $x_i$,
i.e., $x_i \approx D \alpha_i$, where the coefficient $\alpha_i = [\delta_1(\alpha_i); \ldots; \delta_v(\alpha_i); \ldots; \delta_V(\alpha_i); \delta_{V+1}(\alpha_i)] \in \mathbb{R}^p$,
and $\delta_v(\alpha_i) \in \mathbb{R}^{p_v}$ is the part corresponding to the $v$-th sub-dictionary $D_v$. In addition
to the overall dictionary, it is also expected that the features from the $v$-th view can
be well represented by the cooperative efforts of the $v$-th specificity $D_v$ and the latent
correlation $D_{V+1}$, i.e., $x_i \approx D_v \delta_v(\alpha_i) + D_{V+1} \delta_{V+1}(\alpha_i)$. The Elastic-net regularization
is used in this model to cope with instability issues of estimators observed with the
$\ell_1$-regularization. Therefore, the objective function $f$ can be written as below:

$$
f \triangleq \sum_{v=1}^{V} \sum_{i \in I_v} \left\{ \frac{1}{2} ||x_i - D \alpha_i||^2 + \lambda_1 ||\alpha_i||_1 + \lambda_2 ||\alpha_i||_2^2 \right\}. 
$$  \hspace{1cm} (5.4)
5.3 Specificity and Latent Correlation Learning

For mathematical brevity, a coefficient selector $\mathbf{M}_v$ is introduced to select the corresponding coefficients:

$$
\mathbf{M}_v = \begin{bmatrix}
0, & \ldots, & 0, & \mathbf{I}_{p_v \times p_v}, & 0, & \ldots, & 0
\end{bmatrix} \in \mathbb{R}^{p_v \times p_v}
$$

(5.5)

where $\mathbf{I}$ is an identity matrix. Hereafter in this chapter, $\mathbf{I}$ is an identity matrix with appropriate size. Therefore, $\delta_v(\alpha_i) = \mathbf{M}_v \alpha_i$ and $\mathbf{D}_v = \mathbf{D} \mathbf{M}_v^T$ are derived. Then, Eq. (5.4) can be rewritten as

$$
f \overset{\Delta}{=} \sum_{v=1}^{V} \left\{ \left\| \mathbf{X}_v - \mathbf{D} \mathbf{A}_v \right\|_F^2 + \left\| \mathbf{X}_v - \mathbf{D} (\mathbf{M}_v^T \mathbf{M}_v + \mathbf{M}_{V+1}^T \mathbf{M}_{V+1}) \mathbf{A}_v \right\|_F^2 + \lambda_1 \sum_{i \in I_v} \left\| \alpha_i^v \right\|_1 + \lambda_2 \left\| \mathbf{A}_v \right\|_F^2 \right\}
$$

(5.6)

where $\mathbf{A}_v = [\alpha_1^v, \ldots, \alpha_i^v, \ldots, \alpha_n^v] \in \mathbb{R}^{p_v \times n_v}$ is the coefficient matrix of the $v$-th view. Assume $\tilde{\mathbf{M}}_v = \begin{bmatrix} \mathbf{M}_v \\ \mathbf{M}_{V+1} \end{bmatrix}$, then $\tilde{\mathbf{M}}_v^T \tilde{\mathbf{M}}_v = \mathbf{M}_v^T \mathbf{M}_v + \mathbf{M}_{V+1}^T \mathbf{M}_{V+1}$. Thus, Eq. (5.6) can be rewritten concisely, and the objective function can be obtained as

$$
f \overset{\Delta}{=} \sum_{v=1}^{V} \left\{ \left\| \mathbf{X}_v - \mathbf{D} \mathbf{A}_v \right\|_F^2 + \left\| \mathbf{X}_v - \mathbf{D} \tilde{\mathbf{M}}_v^T \tilde{\mathbf{M}}_v \mathbf{A}_v \right\|_F^2 + \left\| \alpha_i^v \right\|_1 + \lambda_2 \left\| \mathbf{A}_v \right\|_F^2 \right\}.
$$

(5.7)

From Eq. (5.7), it is evident that the representation power of the overall dictionary is guaranteed by the first term, and the specificity $\mathbf{D}_v$ in conjunction with the latent correlation $\mathbf{D}_{V+1}$ has the ability to well represent the data of its corresponding view through the second term. In the next section, the optimization of this model is presented.
5.4 Optimization of the Model

The optimization problem in Eq. (5.7) can be solved in an alternative fashion, as the objective function $f$ is convex when fixing the sub-dictionaries to update the coefficient matrices, or vice versa. The overall algorithm is first summarized in Algorithm 2, and the detailed steps are illustrated in the remainder of this section. The end of this section describes initialization of the sub-dictionaries.

Algorithm 2 Specificity and latent correlation learning

**Input:** features $X = [X_1, \ldots, X_V]$ from multiple views, size $p_v$'s of the $V+1$ sub-dictionaries, $\lambda_1$ and $\lambda_2$

**Output:** the learned dictionaries $D_1, \ldots, D_V, D_{V+1}$

1: initialize $D_v$ for $v = 1, \ldots, V, V + 1$

2: while stop criterion is not reached do

3: update the coefficient matrix $A_v$ by solving the Lasso problem in Eq. (5.12)

4: update the specificity $D_v$ atom-by-atom through Eq. (5.20) with normalization and correspondingly scale the related coefficients

5: update the latent correlation $D_{V+1}$ atom-by-atom through Eq. (5.27) with normalization and correspondingly scale the related coefficients

6: end while

5.4.1 Optimization Step — Coefficient Matrix $A_v$

Assuming that the overall dictionary $D$ is fixed, coefficient matrix $A_v$ can be updated one by one from the objective function $f$ in Eq. (5.7) as below

$$A_v = \arg\min_{A_v} \left\{ \frac{1}{2} \left\| X_v - DA_v \right\|_F^2 + \frac{1}{2} \left\| X_v - D\tilde{M}_v \tilde{M}_v A_v \right\|_F^2 + \lambda_1 \sum_{i \in I_v} \left\| \alpha^v_i \right\|_1 + \lambda_2 \left\| A_v \right\|_F^2 \right\}$$

$$= \arg\min_{A_v} \left\{ \frac{1}{2} \left\| X_v - DA_v \right\|_F^2 + \left\| X_v - D\tilde{M}_v \tilde{M}_v A_v \right\|_F^2 + \lambda_1 \sum_{i \in I_v} \left\| \alpha^v_i \right\|_1 \right\}.$$
Thus the coefficients for the $i$-th feature from the $v$-th view can be calculated by solving the following convex problem

$$
\alpha^v_i = \arg\min_{\alpha^v_i} \left\{ \left\| \bar{x}^v_i - \mathbf{D} \alpha^v_i \right\|_2^2 + \lambda_1 \sum_{i \in I^v} \| \alpha^v_i \|_1 \right\} \tag{5.9}
$$

where $I \in \mathbb{R}^{p \times p}$ is an identity matrix. Denote $\bar{x}^v_i = \left[ x^v_i, x^v_i; 0; \ldots; 0 \right]$ and $\mathbf{D}'^v = \left[ \mathbf{D}; \mathbf{D}^T \mathbf{M}_v ; \sqrt{\lambda_2} \mathbf{I} \right]$, so that Eq. (5.9) can be rewritten as

$$
\alpha^v_i = \arg\min_{\alpha^v_i} \left\{ \left\| \bar{x}^v_i - \mathbf{D}'^v \alpha^v_i \right\|_2^2 + \lambda_1 \sum_{i \in I^v} \| \alpha^v_i \|_1 \right\} . \tag{5.10}
$$

Given the analytical solution can be calculated for Eq. (5.10) if the sign of each element in $\alpha^v_i$ is known, the feature-sign method [71] can be adopted here to obtain the coefficients due to its speedup advantages and effectiveness. However, the matrix $\mathbf{D}'^v$ needs to be normalized before using the feature-sign search method. Through simple derivations, it is easy to show the Euclidean length of each column of $\mathbf{D}'^v$ is $\sqrt{\lambda_2 + 2}$, then

$$
\alpha^v_i = \arg\min_{\alpha^v_i} \left\{ \left\| \bar{x}^v_i - \frac{1}{\sqrt{\lambda_2 + 2}} \mathbf{D}'^v \sqrt{\lambda_2 + 2} \alpha^v_i \right\|_2^2 + \lambda_1 \sum_{i \in I^v} \| (\sqrt{\lambda_2 + 2}) \alpha^v_i \|_1 \right\} \tag{5.11}
$$

which is equivalent to

$$
\bar{\alpha}^v_i = \arg\min_{\bar{\alpha}^v_i} \left\{ \left\| \bar{x}^v_i - \bar{\mathbf{D}} \sqrt{\lambda_2 + 2} \alpha^v_i \right\|_2^2 + \frac{\lambda_1}{\sqrt{\lambda_2 + 2}} \sum_{i \in I^v} \| \bar{\alpha}^v_i \|_1 \right\} \tag{5.12}
$$
where \( \bar{\alpha}_v^y = \sqrt{\lambda_2 + 2\alpha_v^y} \), and \( \bar{D}_v' = \left( \frac{1}{\sqrt{\lambda_2 + 2}} \right) D' \). Therefore, the feature-sign search method can be applied to Eq. (5.12) to obtain \( \bar{\alpha}_v^y \), and the coefficients for input feature \( x_v^y \) should be \( \alpha_v^y = \left( \frac{1}{\sqrt{\lambda_2 + 2}} \right) \bar{\alpha}_v^y \).

### 5.4.2 Optimization Step — Overall Dictionary \( D \)

To update the overall dictionary \( D = [D_1, \ldots, D_V, D_{V+1}] \), an iterative approach is employed; updating \( D_v \) by fixing all the other \( D_i \)'s where \( i \neq v \). Note that in a different way from the specificity \( D_v \)'s, the latent correlation \( D_{V+1} \) always contributes to the representation of latent information of all the views. Therefore, there are differences in optimizing the latent correlation \( D_{V+1} \) and the specificity \( D_v \)'s. The optimization steps are illustrated in detail below.

**Update the specificity \( D_v \)**

Without loss of generality, this section concentrates on the optimization of the \( v \)-th view-specific dictionary \( D_v \) by fixing the latent correlation \( D_{V+1} \) and all the other view-specific dictionaries \( D_i \)'s where \( i \neq v \).

**Proposition 1.**

\[
D = \sum_{v=1}^{V+1} D_v M_v
\]

where \( M_v \) is the coefficient selector that was introduced in Section 5.3.

Proposition 1 is easy to prove, and the detailed proof can be found in Appendix A. This proposition indicates the relationship between \( D \) and \( D_v \) for \( v = 1, \ldots, V, V+1 \). Therefore, the \( v \)-th specificity is updated as below

\[
D_v = \arg\min_{D_v} \left\{ \left\| X_v - \sum_{i=1,i \neq v}^{V+1} D_i M_i A_v - D_v M_v A_v \right\|_F^2 + \right\},
\]

\[
D_v = \arg\min_{D_v} \left\{ \left\| X_v - D_v M_v A_v - D_{V+1} M_{V+1} A_v \right\|_F^2 \right\}.
\]

(5.13)
Denote $B_v^{(i)} = M_i A_v$ for $i = 1, \ldots, V + 1$. By dropping the unrelated terms of $D_v$, Eq. (5.13) can be rewritten as

$$D_v = \arg\min_{D_v} \left\{ \left\| X_v - \sum_{i=1,j\neq v}^{V+1} D_i B_v^{(i)} - D_v B_v^{(v)} \right\|_F^2 + \left\| X_v - D_v B_v^{(v)} - D_{V+1} B_v^{(V+1)} \right\|_F^2 \right\}. \quad (5.14)$$

Let $Q_v = X_v - \sum_{i=1,i\neq v}^{V+1} D_i B_v^{(i)}$, $R_v = X_v - D_{V+1} B_v^{(V+1)}$, then

$$D_v = \arg\min_{D_v} \left\{ \left\| Q_v - D_v B_v^{(v)} \right\|_F^2 + \left\| R_v - D_v B_v^{(v)} \right\|_F^2 \right\}. \quad (5.15)$$

This work proposes to update $D_v = \left[ d_1^{(v)}, \ldots, d_i^{(v)}, \ldots, d_p^{(v)} \right]$ atom by atom, that is, by updating $d_i^{(v)}$ while fixing the other columns. Specifically, denote $B_v^{(v)} = \left[ b_1^T; \ldots; b_i^T; \ldots; b_p^T \right]$, where $b_i^T$ is the $i$-th row of $B_v^{(v)}$. Since

$$D_v B_v^{(v)} = \sum_{j=1}^{p_v} d_j^{(v)} b_j^T = \sum_{j \neq i} d_j^{(v)} b_j^T + d_i^{(v)} b_i^T, \quad (5.16)$$

the column $d_i^{(v)}$ of $D_v$ in Eq. (5.15) is updated as below

$$d_i^{(v)} = \arg\min_{d_i^{(v)}} \left\{ \left\| Q_v - \sum_{j \neq i} d_j^{(v)} b_j^T - d_i^{(v)} b_i^T \right\|_F^2 + \left\| R_v - \sum_{j \neq i} d_j^{(v)} b_j^T - d_i^{(v)} b_i^T \right\|_F^2 \right\}. \quad (5.17)$$

Let $\tilde{Q}_v = Q_v - \sum_{j \neq i} d_j^{(v)} b_j^T$ and $\tilde{R}_v = R_v - \sum_{j \neq i} d_j^{(v)} b_j^T$, then

$$d_i^{(v)} = \arg\min_{d_i^{(v)}} \left\{ g(d_i^{(v)}) \triangleq \left\| \tilde{Q}_v - d_i^{(v)} b_i^T \right\|_F^2 + \left\| \tilde{R}_v - d_i^{(v)} b_i^T \right\|_F^2 \right\}. \quad (5.18)$$

The first deviation of $g(d_i^{(v)})$ w.r.t. $d_i^{(v)}$ can be obtained as

$$\frac{\partial g(d_i^{(v)})}{\partial d_i^{(v)}} = -2\tilde{Q}_v b_i - 2\tilde{R}_v b_i + 4\|b_i\|_2 d_i^{(v)}. \quad (5.19)$$
The detailed deviation is listed in Appendix A. Then, the updated atom $d_i^v$ can be calculated by setting Eq. (5.19) to zero, which is:

$$d_i^v = \frac{\bar{Q}_v + \bar{R}_v}{2\|b_i\|^2_2}. \quad (5.20)$$

Note that, as the visual word of a dictionary, $d_i^v$ should be normalized to satisfy the constraint in Eq. (5.3), i.e., $\hat{d}_i^v = d_i^v / \|d_i^v\|_2$. Along with this normalization process, the corresponding coefficient should be multiplied by $\|d_i^v\|_2$, i.e., $\hat{b}_i^T = \|d_i^v\|^2_2 b_i^T$.

**Update the latent correlation $D_{V+1}$**

By dropping the unrelated terms, the latent correlation $D_{V+1}$ can be updated as below:

$$D_{V+1} = \arg\min_{D_{V+1}} \sum_{v=1}^V \left\{ \left\| X_v - \sum_{i=1}^V D_i B_v^{(i)} - D_{V+1} B_v^{(V+1)} \right\|^2_F + \left\| X_v - D_v B_v^{(v)} - D_{V+1} B_v^{(V+1)} \right\|^2_F \right\} \quad (5.21)$$

where $B_v^{(i)} = M_i A_v$ as used in Eq. (5.14). Denote $Q_v = X_v - \sum_{i=1}^V D_i B_v^{(i)}$ and $R_v = X_v - D_v B_v^{(v)}$ (note $Q_v$ and $R_v$ here are different from those used in the optimization of $D_v$), then

$$D_{V+1} = \arg\min_{D_{V+1}} \sum_{v=1}^V \left\{ \left\| Q_v - D_{V+1} B_v^{(V+1)} \right\|^2_F + \left\| R_v - D_{V+1} B_v^{(V+1)} \right\|^2_F \right\} \quad (5.22)$$

$$= \arg\min_{D_{V+1}} \left\{ \left\| Q - D_{V+1} B^{(V+1)} \right\|^2_F + \left\| R - D_{V+1} B^{(V+1)} \right\|^2_F \right\}$$

where $Q = [Q_1, \ldots, Q_v]$, $R = [R_1, \ldots, R_v]$, and $B^{(V+1)} = [b_1^T, \ldots, b_{p_{V+1}}^T]$. Similarly to the optimization of $D_v$, $D_{V+1}$ is also updated column by column. Since

$$D_{V+1} B^{(V+1)} = \sum_{j \neq i} d_{V+1}^j b_j^T + d_{V+1}^i b_i^T, \quad (5.23)$$
5.4 Optimization of the Model

the column \( \mathbf{d}_{V+1}^i \) of \( \mathbf{D}_{V+1} \) in Eq. (5.22) is updated by

\[
\mathbf{d}^i_{V+1} = \text{argmin}_{\mathbf{d}^i_{V+1}} \left\{ \left\| \mathbf{Q} - \sum_{j \neq i} \mathbf{d}^j_{V+1} \mathbf{b}_j^T - \mathbf{d}_{V+1}^i \mathbf{b}_i^T \right\|_F^2 + \left\| \mathbf{R} - \sum_{j \neq i} \mathbf{d}^j_{V+1} \mathbf{b}_j^T - \mathbf{d}_{V+1}^i \mathbf{b}_i^T \right\|_F^2 \right\}. \tag{5.24}
\]

Denote \( \bar{\mathbf{Q}} = \mathbf{Q} - \sum_{j \neq i} \mathbf{d}^j_{V+1} \mathbf{b}_j^T \) and \( \bar{\mathbf{R}} = \mathbf{R} - \sum_{j \neq i} \mathbf{d}^j_{V+1} \mathbf{b}_j^T \), then

\[
\mathbf{d}^i_{V+1} = \text{argmin}_{\mathbf{d}^i_{V+1}} \left\{ h(\mathbf{d}^i_{V+1}) = \||\bar{\mathbf{Q}} - \mathbf{d}^i_{V+1} \mathbf{b}_i^T||_F^2 + ||\bar{\mathbf{R}} - \mathbf{d}^i_{V+1} \mathbf{b}_i^T||_F^2 \right\}. \tag{5.25}
\]

The first deviation of \( h(\mathbf{d}^i_{V+1}) \) can be derived as:

\[
\frac{\partial h(\mathbf{d}^i_{V+1})}{\partial \mathbf{d}^i_{V+1}} = -2\bar{\mathbf{Q}}\mathbf{b}_i - 2\bar{\mathbf{R}}\mathbf{b}_i + 4(||\mathbf{b}_i||_2^2 \mathbf{d}^i_{V+1}). \tag{5.26}
\]

The deviation calculation of \( h(\mathbf{d}^i_{V+1}) \) is similar to the deviation calculation of \( g(\mathbf{d}^i_{V}) \), which is illustrated in detail in Appendix A. By setting Eq. (5.26) to zero, the updated column \( \mathbf{d}^i_{V+1} \) can be obtained as

\[
\mathbf{d}^i_{V+1} = \frac{(\bar{\mathbf{Q}} + \bar{\mathbf{R}})\mathbf{b}_i}{2 ||\mathbf{b}_i||_2^2}. \tag{5.27}
\]

Similarly, \( \mathbf{d}^i_{V+1} \) is further normalized as \( \hat{\mathbf{d}}^i_{V+1} = \mathbf{d}^i_{V+1} / ||\mathbf{d}^i_{V+1}||_2 \), and the coefficient is updated as \( \hat{\mathbf{b}}_i^T = ||\mathbf{d}^i_{V+1}||_2 \mathbf{b}_i^T \).

The overall algorithm is summarized in Algorithm 2 at the beginning of this section. As the updated sub-dictionaries and coefficient matrices consistently decrease the objective value of \( f \) in two alternative optimizations, Algorithm 2 will converge to a local minima. The initialization of the sub-dictionaries is discussed in the next subsection.
5.4.3 Initialization of the Sub-Dictionaries

As Algorithm 2 converges to a local minima, the initialization of the sub-dictionaries is crucial for learning the desirable dictionary. This work employs the K-SVD algorithm [4] to initialize the specificity and the latent correlation. Examining this in more detail, to initialize the \( v \)-th sub-dictionary, K-SVD is performed among all the features from the \( v \)-th view, and while initializing the latent correlation, K-SVD is applied to the whole features from different views. Empirically, this simple initialization method always produces promising results.

5.5 Spatial-Temporal Pyramid Matching

In the proposed approach, gesture recognition is performed with the help of the spatial-temporal pyramid matching (STPM) framework. STPM is similar to STPCM (Chapter 4), except that it explicitly learns the latent correlation within multiple views rather than using the cuboid fusion scheme to preserve the spatial information of three views. Fig. 5.3 shows the framework of STPM, consisting of \textit{training} and \textit{classification} stages. The training stage includes overall dictionary learning and gesture model learning. Specifically, training samples are represented using hierarchical DMHT with multiple views. The overall dictionary (consisting of specificity and latent correlation) is then learned from multiple views of hierarchical DMHT using the proposed method as mentioned in Section 5.3. Features extracted from multiple views of pyramid DMHT are encoded over the learned dictionary, and the coefficients are fed to the classifier for gesture model learning. In the classification stage, unknown gestures are represented by pyramid DMHT, and coefficients are obtained through learned specificity and latent correlation. Finally the predicted class labels are determined by the learned gesture model.
Fig. 5.3 Illustration of the spatio-temporal pyramid cuboid matching framework.
5.6 **Experimental Validation**

In this section, a series of experiments is performed to evaluate the proposed model. Four benchmark datasets, MSR Action3D [75], MSR Gesture3D [143], MSR Action Pairs [111], and ChaLearn multi-modal dataset [33], are used for evaluation purposes. The effectiveness of the proposed method is compared with state-of-the-art depth map-based approaches. In all experiments, the proposed approach consistently achieves comparable results with the state-of-the-art methods.

5.6.1 **Parameter Setting**

In all experiments, $\alpha$ and $\beta$ in Eq. (5.1) are set in the range of $[-90^\circ, 45^\circ, 90^\circ]$ to obtain nine multiple views from a depth map. Examples are the images along the horizontal and vertical directions shown in Fig. 5.2. In the framework of STPM, temporal strides are set to $\{1, 2, 3\}$ for hierarchical DMHT generation, and temporal scale $S = 1$ and spatial level $L = 1$ are used for pyramid DMHT generation and matching.

Within STPM, the proposed model is adopted to learn the overall dictionary for sparse coding of each patch. In detail, each gesture sequence is partitioned into $2^s$ sub-volumes at the $s$-th scale partition, so there are $3 \left( \sum_{s=0}^{1} 2^s \right)$ DMHTs to represent a gesture sequence in pyramid DMHT generation. For feature extraction from multiple views of DMHTs, HOG features (144 in length) are first extracted from $32 \times 32$ pixel patches, which are densely sampled from each view on a dense grid with step size 16 pixels (for a detailed description go to Section 4.3). From the extracted HOG features of multiple views, the proposed model is used to learn an overall dictionary $D \in \mathbb{R}^{144 \times p}$ concatenated by the specificities and latent correlation. According the observations, the best performance is obtained when $\lambda_1 = 0.1 \sim 0.2$, and $\lambda_2 = 0.01 \sim 0.02$. Then, the features of an image are encoded over the learned dictionary. By dividing each view of DMHT into $2^l \times 2^l$ grids in different levels, max
pooling technique is used to pool the coefficients of each grid into a single vector. There are 5 \((\sum_{l=0}^{1}2^l \times 2^l)\) grids in each view of DMHT. Consequently, there are 135 \((9 \times 3 \times 5)\) pooled vectors, which are concatenated into the \(135 \times p\)-dimensional vector as the final representation of the gesture sequence. Linear SVMs in the framework of STPM are used for the classification of the calculated representations.

### 5.6.2 MSR Action3D Dataset

The proposed method is evaluated in terms of recognition accuracy and compared with the state-of-the-art approaches that have been applied to the MSR Action3D dataset. This experiment is performed using the same settings described in Subsection 4.5.1: five subjects are used for training, with the remaining five subjects used for testing.

Direct comparisons are listed in Table 5.1. In this dataset, the proposed method learns an overall dictionary of 195 visual words, with 20 visual words for each view, and the last 15 visual words acting as the latent correlation, i.e., \(9 \times 20 + 15 = 195\). Table 5.1 shows that the proposed method (SLCL + STPM) achieves the highest recognition accuracy compared with others. This proves that the proposed SLCL+STPM approach utilizes the features from multiple views, and therefore achieves better results.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bag of 3D points [75]</td>
<td>74.70</td>
</tr>
<tr>
<td>STOP [142]</td>
<td>84.80</td>
</tr>
<tr>
<td>ROP [143]</td>
<td>86.20</td>
</tr>
<tr>
<td>HON4D [111]</td>
<td>88.89</td>
</tr>
<tr>
<td>DSTIP [159]</td>
<td>89.30</td>
</tr>
<tr>
<td>DMHT-PHOG (Chapter 3)</td>
<td>90.30</td>
</tr>
<tr>
<td>DMM-HOG [165]</td>
<td>91.70</td>
</tr>
<tr>
<td>STPCM (Chapter 4)</td>
<td>94.26</td>
</tr>
<tr>
<td>SLCL + STPM</td>
<td>95.77</td>
</tr>
</tbody>
</table>
Furthermore, in order to study how the sizes of the view-specific dictionary and the latent correlation affect the recognition accuracy, the recognition accuracies on the MSR Action3D dataset are plotted with different dictionary sizes in Fig. 5.4. From this figure, it is evident that the proposed method saturates when the sizes of specificity and latent correlation reach 20 and 15, respectively. This indicates that with the help of latent correlation, only a relatively smaller view-specific dictionary is needed to be learned for each view to faithfully reconstruct the gesture representation.

![Fig. 5.4 Recognition accuracies with different dictionary sizes.](image)

Fig. 5.4 Recognition accuracies with different dictionary sizes.

Fig. 5.5 shows the confusion matrix of the SLCL + STPM. Gestures of high similarity get relatively low accuracies. For example, “tennis serve” tends to be confused with “high arm wave”.

### 5.6.3 MSR Gesture3D Dataset

The leave-one-out cross-validation scheme described by Wang et al. [143] is used in the evaluation. The proposed method (SLCL + STPM) obtains the state-of-the-art accuracy of 98.42%, outperforming the previous methods shown in Table 5.2. It is worth noting this method only trains 20 codewords for each view and 15 atoms as the latent correlation, whereas the STPCM put forward in Chapter 4 learns 500
5.6 Experimental Validation

Fig. 5.5 Confusion matrix for MSR Action3D dataset using SLCL+STPM.

atoms of dictionary for each view; even so its results are very close to that of SLCL + STPM. The confusion matrix for the proposed method is shown in Fig. 5.6. SLCL + STPM performs well on most of the dynamic gestures, such as “z”, “where”, and “store” gestures. Self-occlusions happen in many hand gestures, and some complex gestures require multiple views to capture. The proposed multi-view method not only encodes the view-specific information, but also discovers the latent correlation between multiple views. Thus, promising results have been achieved using the proposed SLCL+STPM.

5.6.4 MSR Action Pairs Dataset

In this experiment, evaluation setup as [111] is used. The size of the overall dictionary is set to $p = 255$. Specifically, SLCL learns 25 visual words for each view and 30 for the latent correlation, $9 \times 25 + 30 = 255$ in total. SLCL + STPM achieves an accuracy
Table 5.2 Recognition accuracy(%) comparison of SLCL+STPM with other methods on the MSR Gesture3D dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action Graph on Occupancy [72]</td>
<td>80.50</td>
</tr>
<tr>
<td>Action Graph on Silhouette [72]</td>
<td>87.70</td>
</tr>
<tr>
<td>ROP [143]</td>
<td>88.50</td>
</tr>
<tr>
<td>DMM-HOG [165]</td>
<td>89.20</td>
</tr>
<tr>
<td>HON4D [111]</td>
<td>92.45</td>
</tr>
<tr>
<td>SNV [164]</td>
<td>94.74</td>
</tr>
<tr>
<td>STPCM (Chapter 4)</td>
<td>97.80</td>
</tr>
<tr>
<td>SLCL + STPM</td>
<td>98.42</td>
</tr>
</tbody>
</table>

Fig. 5.6 Confusion matrix for MSR Gesture3D dataset using SLCL+STPM.
of 95.43% that is close to the state-of-the-art performance. The detailed comparison with other depth map-based approaches is demonstrated in Table 5.3. Note that SLCL + STPM is performed on this dataset without removing the background noise, so the generation of DMHT with multiple views could involve noise. This dataset is collected to investigate how the temporal order affects action recognition. It is therefore crucial to capture the spatio-temporal orders to distinguish the actions with similar motion and cues. In SLCL + STPM, the spatial information is encoded by learning the specificity of each view and the latent correlation between multiple views. Furthermore, the spatio-temporal orders are embedded in the framework of STPM that characterizes the spatio-temporal information at different scales and levels. The confusion matrix of SLCL + STPM performed on this dataset is shown in Fig. 5.7.

Table 5.3 Recognition accuracy(%) comparison of SLCL+STPM with other methods on the MSR Action Pairs dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMM-HOG [165]</td>
<td>66.11</td>
</tr>
<tr>
<td>HON4D [111]</td>
<td>96.67</td>
</tr>
<tr>
<td>STPCM (Chapter 4)</td>
<td>91.43</td>
</tr>
<tr>
<td>SLCL + STPM</td>
<td>95.43</td>
</tr>
</tbody>
</table>

5.6.5 ChaLearn Multi-Modal Dataset

This dataset is divided into three parts: training data, validation data, and test data. All the gesture sequences from the training data are used to learn the overall dictionary. Validation data is used for parameter optimization. As for SLCL, a 30-visual-word specificity is learned for each view, and the latent correlation consists of 50 atoms. The size of the concatenated dictionary is $9 \times 30 + 50 = 320$. In this experiment, the truth labels of start and end frames are used to split continuous gestures into several individual gestures. Furthermore, gesture regions are segmented from the original depth maps using depth videos and mask videos (for details refer to
Subsection 4.5.4). The proposed method is then applied to the individual gestures. The direct comparisons are shown in Table 5.4, and the confusion matrix is illustrated in Fig. 5.8. SLCL + STPM obtains the highest accuracy of 91.33%. This is a relatively high performance in “user independent” gesture recognition.

Table 5.4 Recognition accuracy(%) comparison of SLCL+STPM with other methods on the ChaLearn multi-modal dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>2DMTM [76]</td>
<td>76.99</td>
</tr>
<tr>
<td>STPCM (Chapter 4)</td>
<td>87.31</td>
</tr>
<tr>
<td>SLCL + STPM</td>
<td>91.33</td>
</tr>
</tbody>
</table>

5.7 Summary

This chapter presents a method (SLCL) that learns view-specific dictionaries and latent dictionary over multiple views for gesture recognition. Different views of a gesture usually encode some latent information that is essential for good representation.
Therefore, this work proposes a novel approach based on dictionary learning to explicitly learn the latent information (latent correlation), as well as the view-specific dictionaries (specificity). The combination of the specificity and the latent correlation faithfully represent the gesture from multiple views for classification. To evaluate the method, a series of experiments is conducted on four public depth gesture datasets. The experimental results demonstrate that the proposed method achieves very competitive performances on various datasets. However, it is unavoidable that noise occurs in the background or captured gesture data when depth maps are used. In this case, it is imperative to seek an alternative representation to compensate for the gesture representation.
Part-based Skeleton Representation Learning for Gesture Recognition

6.1 Introduction

A human body can be represented as an articulated system of rigid segments connected by joints, and a gesture can be considered to be a continuous evolution of the spatial configuration of these rigid segments [63]. Hence, if the human skeleton can be extracted reliably, gesture recognition can be performed by classifying the temporal evolution of the human skeleton. In addition to depth maps, skeletal joints are therefore also often adopted for gesture analysis and recognition. With the release of depth cameras and the associated SDK, 3D positions of skeleton joints are able to be ascertained in real time and with reasonable accuracy [43, 124, 130]. These recent advances have resulted in a renewed interest in skeleton-based human gesture recognition. As shown in Fig. 6.1, the gesture “pick up” from the MSR Action Pairs dataset can be well reflected from the extracted 3D joints by the joints “HAND_LEFT” and “WRIST_LEFT” (The types of joints can be found in Fig. 2.2(a)). However, it is not straightforward to distinguish the differences between these consecutive depth frames. Furthermore, background noises affect the ability of some depth map-based methods of gesture representation.

Existing skeleton-based gesture recognition approaches can be broadly grouped into two main categories: joint-based approaches and part-based approaches. Joint-based approaches consider the human skeleton simply as a set of joints. These approaches try to model the motion of either individual joints or combinations of joints.
using various features such as joint positions [54, 88], joint orientations with respect to a fixed coordinate axis [160], and pairwise relative joint positions [145, 163]. Luo et al. [87] proposed a sparse coding-based temporal pyramid matching approach (ScTPM) for action recognition. As different skeletal joints have distinctive features even for the same action, each joint was treated independently for the sparse coding problem. That is, multiple dictionaries were separately learned for the skeletal joints. Learned dictionaries were at the joint level in their approach. Therefore, if every variation is to be represented by a different dictionary, there must be a large number of dictionaries for a single gesture class in joint-based approaches. Furthermore, the relative geometry information between various joints provides a meaningful description, so this should be considered in skeleton-based gesture recognition. On the other hand, part-based approaches consider the skeletal joints as a connected set of rigid segments (body parts). These approaches either model the temporal evolution of individual body parts [161] or perform directly on connected pairs of body parts and model the temporal evolution of joint angles [109, 110]. In [19], a human skeleton was hierarchically divided into smaller parts, each part being
represented by certain bio-inspired shape features. The temporal evolutions of these bio-inspired features were modeled using linear dynamic systems. Zhao et al. [175] proposed a new effective and efficient feature extraction method, structured streaming skeleton (SSS), which used a dynamic matching approach to construct a feature vector for each frame. The SSS feature was extracted based on the learned templates, which were clustered separately on different human body parts. The human body is considered to be a combination of many small parts, and analysis is performed on these parts separately, which overlooks the relationships between different body parts.

Based on the above discussion, this work will introduce a novel representation for extracting skeleton features at the body-part level. This representation should be able to represent human motion characteristics of not only the body parts but also the inherent interrelationships of the body parts. The work in this chapter proposes to learn part-based skeleton representation for gesture recognition. As mentioned in Section 1.2.1, the performance of gesture recognition can be affected by four main sources of variation, that is, viewpoint, anthropometry, execution rate, and personal style. These factors can be dealt with using the proposed method, as follows:

- **Viewpoint and anthropometry variations.** Five body parts are used in this work, each of which includes multiple skeletal joints. Skeleton data are firstly pre-processed to generate pair-wise distances between skeletal joints. Furthermore, in each part the distances between each pair of joints are normalized by human body size. Therefore the proposed method is viewpoint invariant and anthropometry invariant.

- **Execution rate variations.** The execution rate variation problem is solved by using temporal pyramid matching (TPM). TPM divides the video sequence into several segments in a pyramid fashion in the temporal direction. Histograms generated from segments by max pooling are concatenated to form the final
Part-based Skeleton Representation Learning for Gesture Recognition

representation. The temporal information is well retained by the segments, and the number of segments is predefined. Therefore the method is execution rate invariant.

• Personal style variations. To solve this problem, learned dictionaries are used at a granularity of the body part level. Each sub-dictionary is constructed by a part movement; therefore, a gesture consists of multiple learned dictionaries. Different personal styles of gestures can be represented by combinations of multiple part movements. Hence, the proposed method can achieve personal style invariance.

6.2 Part-based Representation

This work represents a human as a set of body parts, each of which consists of multiple skeletal joints. The joints are manually grouped into five parts: “Head & Torso”, “Left Arm”, “Right Arm”, “Left Leg”, and “Right Leg”, as shown in Table 6.1 and Fig. 6.2 (The types of joints can be found in Fig. 2.2(a)).

<table>
<thead>
<tr>
<th>Part</th>
<th>Skeletal Joints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head &amp; Torso</td>
<td>HEAD, SHOULDER_CENTER, SPINE, HIP_CENTER</td>
</tr>
<tr>
<td>Left Arm</td>
<td>SHOULDER_LEFT, ELBOW_LEFT, WRIST_LEFT, HAND_LEFT</td>
</tr>
<tr>
<td>Right Arm</td>
<td>SHOULDER_RIGHT, ELBOW_RIGHT, WRIST_RIGHT, HAND_RIGHT</td>
</tr>
<tr>
<td>Left Leg</td>
<td>HIP_LEFT, KNEE_LEFT, ANKLE_LEFT, FOOT_LEFT</td>
</tr>
<tr>
<td>Right Leg</td>
<td>HIP_RIGHT, KNEE_RIGHT, ANKLE_RIGHT, FOOT_RIGHT</td>
</tr>
</tbody>
</table>

Table 6.1 Human body parts and their corresponding skeletal joints.

There are $n$ skeletal joints, and each of them is indexed by a number $i (i = 1, \ldots, n)$. The index set of the body part $s$ is denoted as $\mathcal{P}_s$. Thus, skeletal joints of the body
6.3 Part-based Skeleton Features

Simple and effective features are essential for accurate gesture recognition. In the skeleton feature extraction section, this work proposes part-based skeletal features that capture the spatial information between the joints within each part and the temporal information within each gesture. The feature extraction is performed independently within each body part.

The skeleton model obtained from the depth camera is constituted of joints represented by 3D coordinates, which contains raw information about posture. At frame $t$, the position of each skeleton joint $i$ in one body part is uniquely defined by three coordinates $\mathbf{p}_i(t) = (x_i(t), y_i(t), z_i(t))$ and can be represented as a three-element vector.

**Normalized 3D joint positions.** Compared with the depth map, the skeleton joint positions are much more compact as each frame can be represented by 20 joints while the corresponding depth map contains a large number of pixels. However,
the coordinate system varies in uncontrolled capture environments, which directly
influences the joint coordinate values. Moreover, the same gesture performed by
different subjects would have different coordinate values due to different body sizes.
Therefore, it is essential to normalize coordinate values of skeletal joints into a
common coordinate system. In this work, all 3D joint coordinates are transformed
from the world coordinate system to a person-centric coordinate system by placing a
reference joint at the origin. The joint “HIP_CENTER” is selected as a reference
joint, since it has relatively small motions for most actions. Let the position of
“HIP_CENTER” be \( \mathbf{p}_1(t) \), and \( L_t \) be the sum of the distances of the connected joints.
The normalized skeleton data at frame \( t \) is defined as:

\[
\overline{\mathbf{p}}(t) = \left\{ \frac{\mathbf{p}_i(t) - \mathbf{p}_1(t)}{L_t} \mid i = 1, \ldots, N \right\}
\] (6.1)

where \( N \) is the selected number of skeletal joints at frame \( t \). For the body part \( s \),
the dimensionality of normalized 3D joint positions is \( 3 \times |P_s| \), where \( |P_s| \) is the
cardinality of the set \( P_s \).

**Pair-wise joint distances.** In addition to 3D joint positions of skeletal joints,
pair-wise joint distances [175] are adopted to characterize relative distances between
joints. For each pair-wise joints \( i \) and \( j \), \( 1 \leq i < j \leq N \), their normalized distance \( d_{ij} \)
is calculated:

\[
d_{ij} = \frac{\|\overline{\mathbf{p}}_i - \overline{\mathbf{p}}_j\|_2}{r_{ij}}
\] (6.2)

where \( r_{ij} \) is the route between joints \( i \) and \( j \). As shown in Fig. 6.3, the black dotted
line is the Euclidean distance between “HAND_LEFT” and “HAND_RIGHT”, and
the green bold lines indicate their route. It is obvious that \( d_{ij} \) has no relationship
with the body position, body orientation or body size; that is, it is viewpoint
and anthropometry invariant. In the body part \( s \), the dimensionality of pair-wise
joint distances consists of two parts: (1) when \( i, j \in P_s \), the dimensionality is
|\mathcal{P}_s| \times ((|\mathcal{P}_s| - 1)/2); (2) when \(i \in \mathcal{P}_s, j \notin \mathcal{P}_s\) or \(i \notin \mathcal{P}_s, j \in \mathcal{P}_s\), the dimensionality is \((N - |\mathcal{P}_s|) \times |\mathcal{P}_s|\). Therefore, the final dimensionality of pair-wise joint distances in part \(s\) is \(|\mathcal{P}_s| \times ((|\mathcal{P}_s| - 1)/2 + (N - |\mathcal{P}_s|) \times |\mathcal{P}_s|\).

**Pair-wise positional differences.** Furthermore, pair-wise positional differences of joints [163] are used to capture posture features \(f_{t1}\), temporal motion features \(f_{t2}\) and offset features \(f_{t3}\). Specifically, to characterize the static posture information at frame \(t\), pair-wise joint positional differences are computed:

\[
f_{t1} = \{\hat{\mathbf{p}}_i(t) - \hat{\mathbf{p}}_j(t) | 1 \leq i < j \leq N\}
\]  (6.3)

Similar to the dimensionality of pair-wise joint distances, the dimensionality of \(f_{t1}\) for body part \(s\) is \(|\mathcal{P}_s| \times ((|\mathcal{P}_s| - 1)/2 + (N - |\mathcal{P}_s|) \times |\mathcal{P}_s|) \times 3\)

Additionally, the pair-wise positional differences of joints between frame \(t\) and its preceding frame \(t - 1\) are used to capture the temporal motion property within consecutive frames:

\[
f_{t2} = \{\hat{\mathbf{p}}_i(t) - \hat{\mathbf{p}}_j(t - 1) | \hat{\mathbf{p}}_i(t) \in \bar{\mathbf{p}}(t); \hat{\mathbf{p}}_j(t - 1) \in \bar{\mathbf{p}}(t - 1)\}
\]  (6.4)
To encode the overall dynamic information in frame $t$ and the first frame, the offset features are calculated:

$$f_{t3} = \{ \pmb{\bar{p}}_i(t) - \pmb{\bar{p}}_j(1) \, | \, \pmb{\bar{p}}_i(t) \in \pmb{\bar{p}}(t); \pmb{\bar{p}}_j(1) \in \pmb{\bar{p}}(1) \} \quad (6.5)$$

Temporal motion features $f_{t2}$ and offset features $f_{t3}$ in the body part $s$ have the same dimensionality of $N \times |\mathcal{P}_s| \times 3$

The final pair-wise positional differences of joints at frame $t$ are obtained by direct concatenation: $f_t = [f_{t1}, f_{t2}, f_{t3}]$. This results in a dimensionality of $3 \times ((|\mathcal{P}_s| \times (|\mathcal{P}_s| - 1)/2 + (N - |\mathcal{P}_s|) \times |\mathcal{P}_s| + 2 \times N \times |\mathcal{P}_s|)$ for the body part $s$.

**Dimensionality of features for body part $s$.** The dimensionalities of the skeleton features for body part $s$ described in this section are:

- normalized 3D joint positions: $3 \times |\mathcal{P}_s|$
- pair-wise joint distances: $|\mathcal{P}_s| \times (|\mathcal{P}_s| - 1)/2 + (N - |\mathcal{P}_s|) \times |\mathcal{P}_s|$
- pair-wise positional differences: $3 \times (|\mathcal{P}_s| \times (|\mathcal{P}_s| - 1)/2 + (N - |\mathcal{P}_s|) \times |\mathcal{P}_s| + 2 \times N \times |\mathcal{P}_s|)$

Therefore, the skeleton feature $f_s(t)$ of the body part $s$ at frame $t$ are concatenated by these three types of features, and the final dimensionality $m$ of the feature is the sum of these dimensionalities. For example, when $N = 20$ and $|\mathcal{P}_s| = 4$, $m = 772$.

### 6.4 Part-based Representation Learning

Following a similar dictionary learning framework to that in Section 5.3, this work proposes to explicitly learn the sub-dictionary for each body part, as well as the dictionary representing the correlation between these body parts. The general procedure of part-based representation learning is presented as follows. Suppose there are $S$ body parts and $F = [F_1, \ldots, F_s, \ldots, F_S] \in \mathbb{R}^{m \times n}$ is the extracted
skeleton features, wherein \( \mathbf{F}_s \in \mathbb{R}^{m \times n_s} \) represents the features in the body part \( s \), and feature \( f_i \in \mathbb{R}^m \) from this part is indexed by \( i \in I_s \). Denote the overall dictionary as \( \mathbf{D} = [\mathbf{D}_1, \ldots, \mathbf{D}_s, \ldots, \mathbf{D}_S, \mathbf{D}_{S+1}] \in \mathbb{R}^{m \times p} \), where \( \mathbf{D}_s \in \mathbb{R}^{m \times p_s} \) is the dictionary representing the \( s \)-th body part, and \( \mathbf{D}_{S+1} \in \mathbb{R}^{m \times p_{S+1}} \) is the dictionary representing the correlation of the \( S \) body parts.

Therefore, the objective function is defined as below to minimize the reconstruction error:

\[
 f \triangleq \sum_{s=1}^{S} \sum_{i \in I_s} \left\{ \| \mathbf{f}_i - \mathbf{D} \alpha_i \|_2^2 + \| \mathbf{f}_i - \mathbf{D}_s \delta_s(\alpha_i) - \mathbf{D}_{S+1} \delta_{S+1}(\alpha_i) \|_2^2 + \lambda_1 \| \alpha_i \|_1 + \lambda_2 \| \alpha_i \|_2^2 \right\}
\]  
(6.6)

where \( \alpha_i = [\delta_1(\alpha_i); \ldots; \delta_s(\alpha_i); \ldots; \delta_S(\alpha_i); \delta_{S+1}(\alpha_i)] \in \mathbb{R}^p \) is the coefficient, and \( \delta_s(\alpha_i) \in \mathbb{R}^{p_s} \) is the part corresponding to the \( s \)-th sub-dictionary \( \mathbf{D}_s \). Therefore, a learned overall dictionary \( \mathbf{D} \) is expected to well represent every feature \( \mathbf{f}_i \), i.e., \( \mathbf{f}_i \approx \mathbf{D} \alpha_i \). Furthermore, features of the body part \( s \) are expected to be well represented by the cooperative efforts of the \( s \)-th sub-dictionary \( \mathbf{D}_s \) and the correlation \( \mathbf{D}_{S+1} \), i.e., \( \mathbf{f}_i \approx \mathbf{D}_s \delta_s(\alpha_i) + \mathbf{D}_{S+1} \delta_{S+1}(\alpha_i) \). The Elastic-net regularization is added as the sparsity constraint, which is a combination of \( \ell_1 \) - and \( \ell_2 \) - norms.

A coefficient selector \( \mathbf{M}_s \) is introduced to select the corresponding coefficients:

\[
 \mathbf{M}_s = \begin{bmatrix} 0 & \cdots & 0 \end{bmatrix}_{p \times 1} P_{s} \times p_s, \begin{bmatrix} \mathbf{I} \end{bmatrix}_{p \times p_s}, \begin{bmatrix} 0 \cdots 0 \end{bmatrix}_{p \times 1}, \begin{bmatrix} \mathbf{I} \end{bmatrix}_{p \times 1} P_{s+1} \times p_{s+1}
\end{bmatrix} \in \mathbb{R}^{p \times p}
\]

where \( \mathbf{I} \) is an identity matrix. Thus, \( \delta_s(\alpha_i) = \mathbf{M}_s \alpha_i \) and \( \mathbf{D}_s = \mathbf{D} \mathbf{M}_s^T \). Eq. (6.6) can be rewritten concisely as below:

\[
 f \triangleq \sum_{s=1}^{S} \left\{ \| \mathbf{F}_s - \mathbf{D} \mathbf{A}_s \|_F^2 + \| \mathbf{F}_s - \mathbf{D} \mathbf{M}_s^T \mathbf{M}_s \mathbf{A}_s \|_F^2 + \lambda_1 \sum_{i \in I_s} \| \alpha_i^s \|_1 + \lambda_2 \| \mathbf{A}_s \|_F^2 \right\}
\]  
(6.7)
where $A_s = [\alpha_s^1, \ldots, \alpha_s^n] \in \mathbb{R}^{p \times n_s}$ is the coefficient matrix of the $s$-th body part, and $\tilde{M}_s = \begin{bmatrix} M_s \\ M_{S+1} \end{bmatrix}$. Thus, $\tilde{M}_s^T \tilde{M}_s = M_s^T M_s + M_{S+1}^T M_{S+1}$. Eq. (6.7) shows that the representation power of the overall dictionary is guaranteed by the first term, and the part-specific sub-dictionary $D_s$ in conjunction with the correlation $D_{S+1}$ have the ability to well represent the features of its corresponding body part through the second term.

The optimization problem in Eq. (6.7) can be solved via an alternative fashion, as the objective function $f$ is convex when fixing the sub-dictionaries to update the coefficient matrices, or vice versa. The overall algorithm is summarized in Algorithm 3.

**Algorithm 3** Part-based skeleton representation learning

**Input:** skeleton features $F = [F_1, \ldots, F_S]$ from $S$ body parts, size $p_s$’s of the $S + 1$ sub-dictionaries, $\lambda_1$ and $\lambda_2$

**Output:** the learned dictionaries $D_1, \ldots, D_S, D_{S+1}$

1: initialize $D_s$ for $s = 1, \ldots, S, S + 1$
2: while stop criterion is not reached do
3: update the coefficient matrix $A_s$ by solving the Lasso problem similar to Eq. (5.12)
4: update the specificity $D_s$ atom-by-atom by solving the problem similar to Eq. (5.20) with normalization and correspondingly scale the related coefficients
5: update the latent correlation $D_{S+1}$ atom-by-atom by solving the problem similar to Eq. (5.27) with normalization and correspondingly scale the related coefficients
6: end while

The optimization problem of Eq. (6.7) can be solved following the specificity and latent correlation learning in Chapter 5. Details can be found in Section 5.4. To initialize the $s$-th sub-dictionary, K-SVD is performed on all the features from the $s$-th body part, and while initializing the sub-dictionary $D_{S+1}$, K-SVD is performed on all the features from different body parts.
6.5 Temporal Pyramid Matching

Similar to the STPM framework (Section 5.5), temporal pyramid matching (TPM) based on max pooling is used to retain the temporal information during feature representation. TPM is based on max pooling to yield histogram representation for every skeleton sequence. TPM divides the video sequence into several segments in a pyramid method in the temporal direction. Given the coding coefficients of each temporal segment at each level, a pooling method is often used to obtain an holistic representation for the segment. Sum pooling [70] and max pooling [162] are two common pooling strategies. In [12], the authors presented a theoretical analysis of sum pooling and max pooling. Their results indicate sparse features may prefer max pooling. Therefore, max pooling is adopted in each temporal segment to generate the representation. The pooled feature is then further normalized by normalization methods. Generally, there are two common normalization techniques. In $\ell_1$-normalization [162], the feature $p$ is divided by its $\ell_1$ norm: $p = \frac{p}{\sum_{k=1}^{K} |p_k|}$.

In $\ell_2$-normalization [114], the feature $p$ is divided by its $\ell_2$ norm: $p = \frac{p}{\sqrt{\sum_{k=1}^{K} p_k^2}}$.

Based on the analysis in [149], the practical choice for max pooling is $\ell_2$-normalization, thus $\ell_2$-normalization is applied after max pooling to generate a histogram for each temporal segment. All the histograms generated from all the segments are concatenated to form a long histogram as the feature representation for the gesture. During the training stage, the overall dictionary $D$ can be learned using Algorithm 3. After constructing the overall dictionary, the coefficients for a given feature can be obtained.

The scheme of skeleton-based TPM is illustrated in Fig. 6.4. Instead of modeling the temporal evolution of features, TPM is focused on the distribution of representative skeleton features within a given time period. There are two advantages of using this temporal pyramid:

1. the temporal information is well kept by temporal segments;
2. it is not sensitive to the temporal shift or misalignment as the lower level of the pyramid retains less temporal information.

6.6 Experiments

Three benchmark datasets, MSR Action3D [75], MSR Action Pairs [111] and ChaLearn multi-modal dataset [33], are used to evaluate the proposed method for gesture recognition. The method is compared with the existing skeleton-based approaches and the proposed depth-based methods.

6.6.1 Parameter Setting

In the framework of TPM, the proposed method is adopted to learn the overall dictionary for sparse coding of each skeleton feature vector. In detail, each skeleton sequence is partitioned into $2^s$ segments at the $s$-th scale partition, so there are $3 \left( \sum_{s=0}^{1} 2^s \right)$ segments to represent a skeleton sequence. Then part-based skeleton features are extracted from multiple body parts with the dimensionality of $m$. Over the extracted skeleton features, the proposed method is used to learn an overall dictionary $D \in \mathbb{R}^{m \times p}$ concatenated by the part-based sub-dictionaries and the correlation. $\lambda_1 = 0.1 \sim 0.2$, and $\lambda_2 = 0.01 \sim 0.02$ are used as in Section 5.6. Max pooling technique is used to pool the coefficients in each segment into a single vector. Consequently, there are three pooled vectors, which are concatenated into the $3 \times p$-dimensional vector as the final representation of the skeleton sequence. Linear SVMs are used for the classification of the obtained representations.

6.6.2 MSR Action3D Dataset

The proposed method is compared with the state-of-the-art methods using cross-subject test setting, where the sequences of half the subjects are used for training,
Fig. 6.4 Illustration of skeleton-based temporal pyramid matching.
and the rest for testing. In this dataset, five body parts are used for part-based representation learning, each of which has four skeletal joints (see Fig. 6.2), i.e., $N = 20$ and $|P_s| = 4$. Thus, the dimensionality $m$ of features for each body part is 772 (covered in more detail in Section 6.3). The proposed method learns an overall dictionary of 2000 visual words, of which 300 visual words are for each body part, and the remaining 500 act as the correlation of these parts, i.e., $5 \times 300 + 500 = 2000$.

As Table 6.2 shows, the proposed approach achieves the highest recognition accuracy of 94.86% among skeleton-based approaches. In [160], human gestures are represented by histograms of 3D joint locations, and the temporal evolutions are modeled by discrete hidden Markov models (HMMs). The relatively low accuracy of 78.97% is due to the small number of training samples for HMM classification. The accuracy of EigenJoints [163] is 82.33%, because similar gestures are more sensitive to the large intra-class variations generated in cross-subject testing. The confusion matrix is illustrated in Fig. 6.5. For most gestures, the proposed method works well; however for similar gestures such as “hammer” and “forward punch”, there are some misclassifications.

Table 6.2 Recognition accuracy(%) comparison of the proposed skeleton method with other methods on the MSR Action3D dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOJ3D [160]</td>
<td>78.97</td>
</tr>
<tr>
<td>Eigenjoints [163]</td>
<td>82.33</td>
</tr>
<tr>
<td>DMHT-PHOG (Chapter 3)</td>
<td>90.30</td>
</tr>
<tr>
<td>STPCM (Chapter 4)</td>
<td>94.26</td>
</tr>
<tr>
<td>SLCL + STPM (Chapter 5)</td>
<td>96.98</td>
</tr>
<tr>
<td>Proposed</td>
<td>94.86</td>
</tr>
</tbody>
</table>

6.6.3 MSR Action Pairs Dataset

This experiment is performed using the same setup as [111]. The size of the overall dictionary is set as $p = 2000$. Specifically, the method learns 300 visual words for
each body part and 500 for the correlation. Most methods tested on this dataset use both depth maps and skeletal joints, and this will be presented in Chapter 7. The proposed method is compared with the bag-of-FLPs [148]. Table 6.3 shows a detailed comparison. The bag-of-FLPs [148] is a gesture representation built by the learned mid-level skeleton features. It obtains relatively lower accuracy of 75.56%, because not enough states are used for the different unit vectors [148]. The proposed method achieves an accuracy of 90.29%; superior to that of the bag-of-FLPs [148]. The results verify that the proposed method can distinguish temporal order in gestures. However, the proposed depth-based methods can obtain higher accuracies of 95.43% (Chapter 5) and 91.43% (Chapter 4). This is because gestures in this dataset are interacted with objects, for example, a box and chair. Skeleton data are not able to capture the information of interaction with objects, while depth maps can provide complementary information. The confusion matrix on this dataset is shown in Fig. 6.6.
Table 6.3 Recognition accuracy(%) comparison of the proposed skeleton method with other methods on the MSR Action Pairs dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>bag-of-FLPs [148]</td>
<td>75.56</td>
</tr>
<tr>
<td>STPCM (Chapter 4)</td>
<td>91.43</td>
</tr>
<tr>
<td>SLCL + STPM (Chapter 5)</td>
<td>95.43</td>
</tr>
<tr>
<td>Proposed</td>
<td>90.29</td>
</tr>
</tbody>
</table>

Fig. 6.6 Confusion matrix for MSR Action Pairs dataset using the proposed skeleton method.
6.6.4 ChaLearn Multi-Modal Dataset

Most of the gestures in this dataset are performed by upper-body movements, so only twelve skeletal joints of the upper body are used, and these are grouped into three body parts: Head & Torso, Left Arm and Right Arm, as illustrated in Table 6.1. In this dataset, 500 visual words are used to learn sub-dictionaries for each body part, and the correlation consists of 500 atoms. Thus, the size of the overall dictionary is $3 \times 500 + 500 = 2000$. In this experiment, the truth labels of start and end frames are used to split continuous gestures into several individual gestures. The proposed method is then performed on the individual gestures, and obtains an accuracy of 92.43%. The direct comparisons are shown in Table 6.4. The skeleton-based method performs better than the proposed depth-based methods. This dataset is focused on “multiple instances, user-independent learning” of gestures. Different subjects have different shape information in depth maps, which could affect recognition using depth maps. On the other hand, the proposed part-based skeleton features are user independent, giving promising results. Furthermore, the confusion matrix is illustrated in Fig. 6.7.

Table 6.4 Recognition accuracy(%) comparison of the proposed skeleton method with other methods on the Chalearn Multi-Modal dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>STPCM (Chapter 4)</td>
<td>87.31</td>
</tr>
<tr>
<td>SLCL + STPM (Chapter 5)</td>
<td>91.33</td>
</tr>
<tr>
<td>Proposed</td>
<td>92.43</td>
</tr>
</tbody>
</table>

6.7 Summary

In this chapter, a method is proposed to enable part-based skeleton representation to be learned for gesture recognition. The work represents a human as a set of body parts, each of which consists of multiple skeletal joints. Part-based skeleton
features of each body part are proposed, to deal with four types of variations, that is, viewpoint, anthropometry, execution rate, and personal style. Given the part-based features, this work proposes a dictionary learning approach to learn sub-dictionaries for each body part and the correlation between them. At last, the recognition of learned representation is performed under the temporal pyramid matching framework. A series of experiments has shown the effectiveness and robustness of the proposed method.

Depth maps and skeletal joints capture different properties of human gestures. Either of them alone is not sufficient to achieve reliable results in some cases. Therefore, it is necessary to adopt both to further improve the performance of gesture recognition that will be presented in the following chapter.
Chapter 7

Multi-Modal Gesture Recognition Using Depth Maps and Skeletal Joints

7.1 Introduction

Previous chapters have shown proposed methods for gesture recognition using one single modality, i.e., depth maps or skeletal joints. However, one single modality is usually insufficient to solve the problem of recognizing some complex human gestures. For example, two different gestures, “eating” and “drinking”, may exhibit very similar skeleton motions in the human-object interaction. Additional modalities, such as depth maps, are necessary to compensate the lost spatio-temporal context in this scenario. To address these issues, multi-modal data are expected to be exploited to represent and recognize human gestures. This work focuses on recognizing human gestures from depth camera, so two modalities — depth maps and skeletal joints — are adopted for this task.

This chapter is a continuation of the previous work. Rather than using only one modality, a multi-modal scheme is developed to combine both depth and skeleton modalities. The proposed framework consists of gesture representation, feature extraction and representation, and multi-modal fusion and classification, as illustrated in Fig. 7.1. For depth maps, specificity and latent correlation learning with spatio-temporal pyramid matching (STPM) is adopted to generate depth-based spatio-temporal pyramid representation (for details see Chapter 5). For skeletal joints, part-based representation learning with temporal pyramid matching (TPM)
is adopted to generate skeleton-based temporal pyramid representation (details are referred to Chapter 6).

As depth-based STPM and skeleton-based TPM capture different properties of human gestures, the combination of the two enables each to provide complementary information to the other. Two kinds of multi-modal fusion schemes are investigated in this work: representation-level and classifier-level fusion. Representation-level fusion concatenates the representations generated from two modalities to form a final representation as the input to the classifier; while classifier-level fusion combines the probability scores from two classifiers to generate the final result. Both representation-level and classifier-level fusion of two modalities are evaluated based on a fast and simple linear SVM classifier. Meanwhile, according to the analysis of fusion schemes, this work proposes a simple yet effective fusion scheme — weight-learning classifier-level fusion. The experimental results indicate that associated representations across multiple video modalities, i.e. depth maps and skeletal joints, lead to a remarkable boost in final recognition performance.

7.2 Multi-Modal Fusion

As different modalities have their own advantages, their fusion yields a multi-modal semantic representation and improves the performance of recognition. This section presents the fusion schemes used in the experiments.

7.2.1 Representation-Level Fusion

For representation-level fusion, the representations generated from two modalities are concatenated to form a final representation as the input to the classifier.

Let $h^d$ and $h^s$ denote representations of depth-based STPM and skeleton-based TPM, respectively. Their concatenation yields a final representation ($h = [h^d, h^s]$).
Fig. 7.1 General framework of the proposed approach.
for an SVM classifier, as shown in Fig. 7.2. The classification is then performed according to the process in Section 3.4.

Fig. 7.2 Representation-level fusion.

### 7.2.2 Classifier-Level Fusion

For classifier-level fusion, the representations of different modalities are used independently for classifier training. The final recognition score is obtained by fusing the scores from multiple classifiers, as shown in Fig. 7.3. To fuse the scores, arithmetic mean or geometric mean is often used.

Fig. 7.3 Classifier-level fusion.

To combine the classifier outputs of both the depth and skeleton modalities, the classifier outputs are first interpreted as the probability measures. For each test sample \( x \), given the degree of confidence that it belongs to class \( q \) can be measured by \( w_q^T x \), the probability representation of the classifier can be defined as:

\[
p(y = q | x) = \frac{e^{(w_q^T x)}}{\sum_{j=1}^{Q} e^{(w_j^T x)}}
\] (7.1)
where \( p(y = q | x) \) gives the probability of sample \( x \) belonging to class \( q \), and \( q \in \mathcal{Y} = \{1, \ldots, Q\} \).

Thus, given a depth representation \( h^d \), a vector of score probabilities \( p^d \) can be obtained:

\[
p^d = [p(y = 1 | h^d), \ldots, p(y = Q | h^d)]
\]  

Similarly, the probability vector \( p^s \) of the classification for the corresponding skeleton representation \( h^s \) is denoted as

\[
p^s = [p(y = 1 | h^s), \ldots, p(y = Q | h^s)]
\]

The goal of classifier-level fusion is estimating the probability of the label \( q \) of a gesture sequence given the score vectors \( p^d \) and \( p^s \). Widely used arithmetic mean and geometric mean are first applied as two baseline classifier-level fusion schemes. Arithmetic mean combines scores from various modalities by taking the arithmetic mean of the scores from multiple classifiers, which can be expressed as:

\[
\hat{y} = \arg\max_{i \in \mathcal{Y}} (p(y = i | h^d) + p(y = i | h^s))
\]  

Geometric mean computes the fused score as the geometric mean of all scores from different classifiers:

\[
\hat{y} = \arg\max_{i \in \mathcal{Y}} (p(y = i | h^d)p(y = i | h^s))
\]

These two methods are simple rule-based fusions that first normalize the scores to a comparable range and then treat each modality identically. These fusion methods are popular due to their inherent robustness to over-fitting [137].

In addition, with the aim of obtaining a stable and robust performance, a weight-learning classifier-level fusion method is proposed. Specifically, a logistic regression
with $\ell_2$ regularization [38] is adopted to learn the weights for the combination of scores. Logistic regression is a common approach for converting a vector of scores into a single value, the likelihood ratio, which can be used to make final decisions. Given two score vectors $p^d$ and $p^s$, a logit score vector $p$ is created by concatenating the logit scores of the two score vectors, where the logit function is defined as

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right)$$  \hspace{1cm} (7.6)

The logit expands the dynamic range of the exponentially distributed probabilistic scores. The resulting scores are close to normally distributed for both positives and negatives and behave better for logistic regression. The primal problem of logistic regression is

$$\min_w \frac{1}{2} w^T w + C \sum \log(1 + \exp(-y_i w^T p_i))$$  \hspace{1cm} (7.7)

where $C > 0$ is a penalty parameter, and $w$ is a matrix/vector with the model weights. Specifically, a set of probability scores are used to train logistic regression model. Logistic regression automatically takes care of correlation between the scores. Gradient descent [129] is usually adopted to find the optimal $w$. The result of training is a set of coefficient values (learned weights) that provide a linear weighting of the scores from the multiple modalities. After obtaining the optimal regression model from the training samples, the learned weights are then used to fuse the probability scores from different modalities. The learned weights for individual modalities are implicitly encoded by the weight vector $w$ in the logistic regression model.

The multi-modal modalities provide complementary properties, resulting in higher recognition accuracy. Particularly, in the context of human object interaction (e.g., MSR Action Pairs [111]), where skeletal joints do not cover object information and depth data have the information of human action as well as the object information.
The combination of skeleton data and depth data boosts the performance of action recognition.

### 7.3 Experiments

In this section, the multi-modal framework is evaluated using three public datasets: MSR Action3D [75], MSR Action Pairs [111] and ChaLearn multi-modal dataset [33]. The multi-modal framework is compared with the existing approaches using different modalities. The depth-based STPM follows the parameter settings in Subsection 5.6.1, and the skeleton-based TPM uses the parameter settings of Section 6.6.1.

As discussed in Section 7.2, the representations generated from two modalities can be fused at two levels: representation-level and classifier-level. For representation-level fusion, representations generated by the two modalities are concatenated to form a final representation that is used as the input to the classifier. For classifier-level fusion, the common fusion schemes — arithmetic mean (AM) and geometric mean (GM) — are used for score combination. In addition, logistic regression (LR) is adopted to learn the weights for combination at classifier level for a more stable performance. For each benchmark dataset, four multi-modal fusion schemes are investigated; representation-level fusion, classifier-level fusion based on AM, GM, and LR. Furthermore, for a more detailed comparison with single-modal methods (depth-based STPM and skeleton-based TPM) and multi-modal method (STPM+TPM), an $F_1$ score for each class is calculated and illustrated. To calculate the $F_1$ score for each class, precision and recall are calculated individually for each class (see Appendix B for details).
7.3.1 MSR Action3D Dataset

The proposed multi-modal framework is evaluated in terms of recognition accuracy and compared with the state-of-the-art methods that have been applied to the MSR Action3D dataset.

The multi-modal framework using the weight-learning classifier-level fusion scheme, STPM + TPM (LR), is compared with the state-of-the-art methods using the MSR Action3D dataset. The comparison of recognition accuracy is listed in Table 7.1. The actionlet ensemble [145] uses both depth features based on LOP and skeleton features based on relative joint positions to achieve an accuracy of 88.20%. [164] combines the depth-based SNVs aligned with all the joint trajectories as the final representation of a depth sequence, and obtains an accuracy of 93.09%. By adopting logistic regression fusion at the classifier level, the proposed multi-modal framework achieves a state-of-the-art performance with an accuracy of 97.28%. The depth-based STPM and skeleton-based TPM compensate for each other in characterizing different aspects of 3D human gestures and thus can be combined to achieve a better recognition performance.

Table 7.1 Recognition accuracy(%) comparison of the proposed multi-modal method with other methods on the MSR Action3D dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Modality</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actionlet Ensemble [145]</td>
<td>depth + skeleton</td>
<td>88.20</td>
</tr>
<tr>
<td>SNV + joint trajectories [164]</td>
<td>depth + skeleton</td>
<td>93.09</td>
</tr>
<tr>
<td>STPM (Chapter 5)</td>
<td>depth</td>
<td>95.77</td>
</tr>
<tr>
<td>TPM (Chapter 6)</td>
<td>skeleton</td>
<td>94.86</td>
</tr>
<tr>
<td>STPM + TPM (LR)</td>
<td>depth + skeleton</td>
<td>97.28</td>
</tr>
</tbody>
</table>

Furthermore, the $F_1$ score for each class is calculated and shown in Fig. 7.4. It is evident that the multi-modal method performs better than a single-modal method in most of the gesture classes, especially for similar gestures, such as, “draw x”, “draw tick” and “draw circle”.
7.3 Experiments

7.3.2 MSR Action Pairs Dataset

Using the same experimental settings as [111], the first five subjects in each gesture class are used for training, and the rest for testing. The multi-modal framework (STPM+TPM) achieves an accuracy of 98.29%. A detailed comparison with other approaches is demonstrated in Table 7.2. In actionlet ensemble [145], the skeleton feature only involves pair-wise differences of joint positions in each frame, and the LOP feature is used to characterize the depth appearance, which counts the number of cloud points within each spatial grid of a depth subvolumes. By adding temporal pyramid to the features in actionlet ensemble, the performance is improved to an accuracy of 82.22%. The proposed multi-modal framework combining these two modalities boosts the performance and yields better accuracy of 98.29%, which is very close to the state-of-the-art accuracy of 98.89% in work [164]. This is because an adaptive spatio-temporal pyramid was used in the work [164]. Different to the setting of evenly subdividing a video along the time axis, [164] proposed to build an adaptive temporal pyramid based on the motion energy, which is flexible to handle the inter-class variance. However, the proposed method obtains comparable performance because of the different aspects of STPM and TPM from human gestures. Certainly, in the context of human-object recognition, one modality alone is usually insufficient.

Fig. 7.4 $F_1$ score for each class on MSR Action3D dataset.
to encode human gesture motion information with an object. The missing motion information resulting from the occlusion of the object is compensated by skeleton data, while the missing object information in the skeleton data is recovered by the depth data. Multiple modalities recoup each other. The multi-modal framework combines the depth representation and skeleton representation, leading to the remarkable boost in recognition accuracy.

Table 7.2 Recognition accuracy(%) comparison of the proposed multi-modal method with other methods on the MSR Action Pairs Dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Modality</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actionlet Ensemble [145]</td>
<td>depth + skeleton</td>
<td>63.33</td>
</tr>
<tr>
<td>Actionlet Ensemble + Pyramid [145]</td>
<td>depth + skeleton</td>
<td>82.22</td>
</tr>
<tr>
<td>SNV + joint trajectories [164]</td>
<td>depth + skeleton</td>
<td>98.89</td>
</tr>
<tr>
<td>STPM (Chapter 5)</td>
<td>depth</td>
<td>95.43</td>
</tr>
<tr>
<td>TPM (Chapter 6)</td>
<td>skeleton</td>
<td>90.29</td>
</tr>
<tr>
<td>STPM + TPM (LR)</td>
<td>depth + skeleton</td>
<td>98.29</td>
</tr>
</tbody>
</table>

The comparison of $F_1$ scores for each class using single-modal and multi-modal methods is shown in Fig. 7.5. The multi-modal method yields consistently better performances than the single-modal methods in each gesture class.

![Fig. 7.5 $F_1$ score for each class on MSR Action Pairs dataset.](image_url)
7.3 Experiments

7.3.3 ChaLearn Multi-Modal Dataset

The performance of the multi-modal method using the ChaLearn multi-modal dataset is compared with the other three methods, and the results are shown in Table 7.3. The work of [76] proposes that 2DMTM could represent gesture motion along with static postures in 2D space to encode regional information about human gestures. Given that subjects show large variations when performing the same action in this dataset, the depth-based 2DMTM only obtains an accuracy of 76.99% (See Table 5.4). When skeleton data is used in addition to 2DMTM, the accuracy increases to 92.80%.

The work of [21] proposes that spatio-temporal features could be learned from four channels (grayscale, depth, gradient, and surface normal) of RGB-D video in an unsupervised way. It utilizes the random forest framework with discriminative decision trees to discover spatio-temporal blocks that are highly discriminative for gesture recognition. The depth-based STPM with cuboid fusion obtains a higher accuracy of 91.33% than that of depth-based 2DMTM, but it is still lower than skeleton-based TPM (92.43%). The multi-modal framework (STPM+TPM) obtains the best accuracy of 93.81% when combining dual modalities. This is a relatively high score in “user-independent” gesture recognition. It is obvious that multi-modal recognition improves performance significantly. Furthermore, the comparison of $F_1$ scores is shown in Fig. 7.6. The results indicate that STPM and TPM are complementary to each other.

Table 7.3 Recognition accuracy(%) comparison of the proposed multi-modal method with other methods on the Chalearn multi-modal dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Modality</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-modality Recognition [21]</td>
<td>RGB+depth+skeleton</td>
<td>90.30</td>
</tr>
<tr>
<td>Skeleton + 2DMTM [76]</td>
<td>skeleton + depth</td>
<td>92.80</td>
</tr>
<tr>
<td>STPM (Chapter 5)</td>
<td>depth</td>
<td>91.33</td>
</tr>
<tr>
<td>TPM (Chapter 6)</td>
<td>skeleton</td>
<td>92.43</td>
</tr>
<tr>
<td>STPM + TPM (LR)</td>
<td>depth + skeleton</td>
<td>93.81</td>
</tr>
</tbody>
</table>
7.3.4 Evaluation of Multi-Modal Fusion Schemes

This experiment investigates different multi-modal fusion schemes, including representation-level fusion, classifier-level fusion based on arithmetic mean (AM), geometric mean (GM), and logistic regression (LR). Fig. 7.7 shows the accuracies when using STPM and TPM separately, and various multi-modal fusion schemes for MSR Action3D dataset, MSR Action Pairs dataset and ChaLearn multi-modal dataset. Performance is improved by combining the two modalities even with the simple fusion schemes at classifier level, i.e., AM and GM. All fusion schemes can improve the performance of gesture recognition, but classifier-level fusion based on LR achieves a relatively stable performance. The experimental results demonstrate that classifier-level fusion using LR outperforms the other fusion schemes and obtains stable performances. The reasons are: (1) logistic regression is able to learn the adaptive weights for various modalities instead of using the simple combination of arithmetic mean and geometric mean; (2) classifier-level fusion performs better when the generated representations from different modalities are relatively independent, as bias in one representation of the modality may be corrected by others.
7.4 Summary

This chapter presents a multi-modal framework that characterizes the spatio-temporal structure of human gestures. It utilizes both depth and skeleton modalities. For the depth modality, a spatio-temporal pyramid matching (STPM) approach is adopted to retain the motion information of multiple views in spatial and temporal directions. In addition, a skeleton modality is used for skeleton-based gesture recognition with temporal pyramid matching (TPM). Part-based skeleton representation with TPM is able to capture the correlations and distribution of the joints of body parts within a given time period. As different modalities have their own properties, various multi-modal fusion schemes are evaluated at representation level and classifier level. The extensive experiments demonstrate the superior performance of the proposed multi-modal approach, compared with the state-of-the-art methods.

Fig. 7.7 Evaluation of various multi-modal fusion schemes.
Conclusions and Future Work

This thesis has presented and evaluated the three types of approaches proposed for gesture recognition with depth camera: depth map-based, skeleton-based, and multi-modal approaches. This chapter summarizes the study findings and discusses conclusions of this thesis in Section 8.1. Based on these conclusions, Section 8.2 then indicates possible future research directions in this field.

8.1 Conclusions

The objective of the thesis is to explore efficient methods for the task of human gesture recognition using depth cameras. This problem has been addressed from the perspectives of gesture representation, feature extraction, and feature representation.

- Gesture Representation. Depth motion history template (DMHT) and part-based skeleton representation are proposed for gestures using depth maps and skeletal joints, respectively.

- Feature Extraction. Pyramid histogram of oriented gradients (PHOG) and part-based skeleton features are identified to extract depth features from DMHT, and skeleton features from part-based skeleton representation, respectively.

- Feature Representation. In order to retain spatial dependent information from the projected planes of the DMHT, a spatio-temporal pyramid cuboid matching (STPCM) framework is developed. Next, the specificity and latent
correlation learning (SLCL) method is proposed for multi-view gesture recognition to learn the sub-dictionary of each view, and latent information among multiple views. For the extracted part-based skeleton features, dictionary learning takes place through the part-based skeleton representation learning method to encode the information of body parts, and the correlation between the body parts. Depth features and skeleton features are then represented through the learned dictionaries to generate the final representations of gestures.

Specifically, the following methods for human gesture recognition with depth camera have been developed.

To represent the spatio-temporal structure of the gesture from depth maps, the DMHT is developed to capture the temporal motion changes of depth maps projected onto three orthogonal planes. Different to DMM \[165\] that accumulates projected depth maps, the proposed DMHT not only encodes 3D motion distribution, but also captures motion changes in the temporal direction. For feature extraction from the DMHT, PHOG features are then extracted from the DMHT by employing spatial pyramid representation. Compared with the state-of-the-art methods on the MSR Action3D dataset, the framework based on DMHT-PHOG achieves a relatively high level of performance under different experimental settings.

Motivated by the success of the dictionary learning (DL) framework, a spatio-temporal pyramid cuboid matching (STPCM) framework is then proposed to recognize human gestures. In that framework, three-channel dictionaries are learned through the features of the projected planes of hierarchical DMHT that handles speed variations by using multiple temporal strides. In order to generate the final representation of the gesture sequence, features within the spatial-dependent grids of pyramid DMHT are grouped into spatio-temporal cuboids through the proposed cuboid fusion scheme. The cuboid fusion scheme retains the 3D locations and motion information both
in spatial and temporal domains. The evaluation of the method on public datasets demonstrates the effectiveness and robustness of the STPCM framework.

In order to address the challenge of complex gesture recognition, based on the DL framework, it is proposed that gestures can be recognized from multiple views generated from the depth map, that is, multiple-view gesture recognition. This method is developed to learn a view-specific sub-dictionary (i.e., specificity) for each view, and a sub-dictionary (i.e., latent correlation) to encode the latent information among multiple views. The specificity encodes the discriminative information of each view, and the latent correlation captures the essential 3D information for multiple views. The specificity and latent correlation learning (SLCL) method produces an overall dictionary that is constructed by specificity and latent correlation. The compact and discriminative dictionary is then used for feature representation of gestures. With the help of the spatio-temporal pyramid matching (STPM) framework, the proposed method (SLCL+STPM) obtains very competitive results on various datasets.

In addition to depth maps, skeletal joints — as another often-used modality from depth cameras — are also adopted for gesture recognition. For gesture representation using skeletal joints, a part-based skeleton representation is proposed, which regards the human body as a combination of parts. Skeleton features within each body part are then fed into the part-based representation learning method for feature representation. The part-based representation learning method explicitly learns the sub-dictionaries for body parts, and the sub-dictionary for the correlation between these body parts. In order to keep the temporal information, a temporal pyramid matching (TPM) framework is adopted for the classification. The experimental results outperform existing skeleton-based methods used on public datasets.

Given that the proposed depth-based and skeleton-based methods have shown their effectiveness for gesture recognition, it is intuitive to combine these two modalities to boost performance. In fact, there are advantages in using multiple modalities for gesture recognition, especially when the modalities are complementary to each other.
For instance, challenging problems such as intra-class similarity of gestures, and performance variation, which cannot be resolved easily using a single modality, can be more effectively solved by fusing multiple modalities. Therefore, depth-based STPM and skeleton-based TPM are combined as a multi-modal fusion for gesture recognition. Various multi-modal fusion schemes are investigated at representation level and classifier level. Furthermore, in order to achieve stable performance when fusing representations from depth maps and skeletal joints, a weight-learning fusion scheme at classifier level is proposed to learn weights that are used to fuse the scores from different modalities. The comparisons of the $F_1$ score for each class have shown that multi-modal fusion improves the performance significantly. In addition, the comparisons of performances of single-modal and multi-modal methods demonstrate that classifier-level fusion using weight-learning method outperforms the other fusion schemes and obtains stable performances.

### 8.2 Future Work

There is an excellent scope for the further development of the work in this thesis. The field of gesture recognition with depth camera is still in its infancy, but is certain to become prevalent in the future. Possible future developments, extensions, and improvements of the presented approaches in this thesis are as follows:

- **Background subtraction.** Clearly, there are some drawbacks to the proposed depth-map based approaches for human gesture recognition. One limitation is background cluttering. In fact, the proposed depth-map based gesture representation (i.e., DMHT) provides even better results if the body silhouette or motion regions are segmented from the background. Hence, in order to build a more robust system, a strong approach for dynamic segmentation independent of environment needs to be developed and be utilized in the proposed approaches.
• **Gesture spotting.** In this work, it is assumed that the beginning and end of a gesture are always known, and the final classification is performed on the whole duration of the isolated gesture. To achieve this condition, gesture spotting is required, which is another active research topic.

• **Real-time gesture recognition.** Another problem is how to continuously recognize multiple gestures from an incoming stream, which is known as real-time gesture recognition. Currently, the proposed methods have limitations when dealing with streaming data in real time. Real-time gesture recognition is an extremely challenging task because of the size of video data and the complexity of human gesture spatio-temporal patterns.

• **Deep learning.** It is expected that the deep learning techniques will also be explored to obtain more powerful and generic feature representations in large-scale video recognition. Deep learning techniques have been proven to outperform hand-crafted features in many computer vision applications. The spatio-temporal structure, which plays a crucial role in identifying human gestures, cannot be captured well in the existing convolutional neural network [66]. More sophisticated temporal models are needed, such as the recurrent neural network [46].

With the rapid development of the depth sensor, a growing number of research projects are being conducted using depth data. In future developments, depth cameras will have a higher resolution, less noise, and an extended sensing range. Furthermore, depth cameras accompanied by traditional cameras on laptops and cell phones will be released in the near future, which will provide broader computer vision research topics and applications. Understanding human gestures is fundamental to artificial intelligence systems. Only when machines understand human gestures can they assist the human beings more effectively. It is believed that further efforts to study these developments will make another step toward true artificial intelligence systems.
References


References


[73] L. Li, S. Li, and Y. Fu, “Discriminative dictionary learning with low-rank regularization for face recognition,” in Automatic Face and Gesture Recognition
References


Appendix A

Derivations

This appendix is related to specificity and latent correlation learning that is previously presented in Chapter 5. The proof of Proposition 1 and the deviation of \( g(d_v^i) \) in Eq. (5.18) are illustrated here.

A.1 Proof of Proposition 1

Proof.

\[
D = [D_1, \ldots, D_v, \ldots, D_V, D_{V+1}]
\]

\[
= \sum_{v=1}^{V+1} \begin{bmatrix}
0 & \ldots & 0 & D_v & 0 & \ldots & 0 \\
p_1 & p_{v-1} & p_v & p_{v+1} & p_{v+1}
\end{bmatrix}
\]

\[
= \sum_{v=1}^{V} D_v \begin{bmatrix}
0 & \ldots & 0 & I_{p_v \times p_v} & 0 & \ldots & 0 \\
p_1 & p_{v-1} & p_v & p_{v+1} & p_{v+1}
\end{bmatrix}
\]

\[
= \sum_{v=1}^{V+1} D_v M_v
\]

A.2 Deviation of \( g(d_v^i) \) in Eq. (5.18)

According to Eq. (5.18), \( g(d_v^i) \equiv \| \overline{Q}_v - d_v^i b_i^T \|_F^2 + \| \overline{R}_v - d_v^i b_i^T \|_F^2 \). \( g(d_v^i) \) can be expanded using the proposition below.

Proposition 2. Let \( A \) and \( B \) be \( m \times n \) matrices,

\[
\| A + B \|_F^2 = tr(AA^T) + 2tr(AB^T) + tr(BB^T)
\]
where \( \text{tr}(\cdot) \) returns the matrix trace.

The detailed proof of Proposition 2 is illustrated below.

**Proof.**

\[
\|A + B\|_F^2 = \text{tr}[(A + B)(A + B)^T] = \text{tr}[(A + B)(A^T + B^T)] = \text{tr}(AA^T + AB^T + BA^T + BB^T) = \text{tr}(AA^T) + \text{tr}(AB^T) + \text{tr}(BA^T) + \text{tr}(BB^T) = \text{tr}(AA^T) + 2\text{tr}(AB^T) + \text{tr}(BB^T)
\]

\[\square\]

Thus,

\[
g(d_i^j) \triangleq \|Q_v - d_i^j b_i^T\|_F^2 + \|R_v - d_i^j b_i^T\|_F^2 = \text{tr}(Q_v^T Q_v) + 2\text{tr}(d_i^j b_i^T Q_v^T) + \text{tr}(d_i^j b_i^T R_v^T) - 2\text{tr}(d_i^j b_i^T Q_v^T) + \text{tr}(d_i^j b_i^T R_v^T) = \text{tr}(Q_v^T Q_v) + 2\text{tr}(d_i^j b_i^T R_v^T) = \text{tr}(Q_v^T Q_v) + 2\text{tr}(b_i^T R_v^T d_i^j) + 2\|b_i\|^2 d_i^j d_i^j
\]

(A.1)

Since \( \text{tr}(ab^T C) = \text{tr}((C^T b)a^T) = (b^T C)a \), where \( a, b \) are vectors and \( C \) is a matrix, Eq. (A.1) is rewritten as below

\[
g(d_i^j) = \text{tr}(Q_v^T Q_v) + 2\text{tr}(b_i^T R_v^T) d_i^j + \|b_i\|^2 d_i^j d_i^j
\]

(A.2)

Then, the first deviation of \( g(d_i^j) \) w.r.t. \( d_i^j \) is

\[
\frac{\partial g(d_i^j)}{\partial d_i^j} = -2Q_v b_i - 2R_v b_i + 4\|b_i\|^2 d_i^j
\]
Let $\frac{\partial g(d_i)}{\partial d_i} = 0$, then

$$0 = -2\bar{Q}_i b_i - 2\bar{R}_i b_i + 4 \|b_i\|^2_2 d_i^t$$

$$\Rightarrow d_i^t = \frac{(\bar{Q}_i + \bar{R}_i)b_i}{2\|b_i\|^2_2}$$

Therefore, the $i$-th column of the dictionary $D_v \in \mathbb{R}^{m \times p_v}$ is updated as $d_i^t / \|d_i^t\|_2$, and the coefficient is rescaled $b_i^T \leftarrow \|d_i^t\|_2 b_i^T$. 
Review of F-measure

In statistical analysis of binary classification, the F-measure is a measure of test accuracy, combining both the precision and the recall of the test. Precision is the fraction of correct results from all the returned results (i.e., the number of correct results divided by the number of all returned results), and recall is the fraction of correct results from all the results that should be returned (i.e., the number of correct results divided by the number of results that should be returned) [171]. Note that both precision and recall have the same numerator.

The general formula for F-measure is

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$  \hspace{1cm} (B.1)

which is the weighted harmonic mean of precision and recall. The $F_1$ score is the most commonly used one where precision and recall have equal weight:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$  \hspace{1cm} (B.2)

Two other commonly used F measures are the $F_2$ measure that weights recall higher than precision, and the $F_{0.5}$ measure that emphasizes precision more than recall.